Guarantees for Self-Play in Multiplayer Games via Polymatrix Decomposability

Revan MacQueen
Department of Computing Science
University of Alberta

revan@ualberta.ca

James R. Wright
Department of Computing Science
University of Alberta

james.wright@ualberta.ca

Abstract

Self-play is a technique for machine learning in multi-agent systems where a learning algorithm learns by interacting with copies of itself. Self-play is useful for generating large quantities of data for learning, but has the drawback that the agents the learner will face post-training may have dramatically different behavior than the learner came to expect by interacting with itself. For the special case of two-player constant-sum games, self-play that reaches Nash equilibrium is guaranteed to produce strategies that perform well against any post-training opponent; however, no such guarantee exists for multi-player games. We show that in games that approximately decompose into a set of two-player constant-sum games (called polymatrix games) where global ϵ-Nash equilibria are boundedly far from Nash-equilibria in each subgame, any no-external-regret algorithm that learns by self-play will produce a strategy with bounded vulnerability. For the first time, our results identify a structural property of multi-player games that enable performance guarantees for the strategies produced by a broad class of self-play algorithms. We demonstrate our findings through experiments on Leduc poker.

1 Introduction

Self-play is one of the most commonly used approaches for machine learning in multi-agent systems. In self-play, a learner interacts with copies of itself to produce data that will be used for training. Some of the most noteworthy successes of AI in the past decade have been based on self-play; by employing the procedure, algorithms have been able to achieve super-human abilities in various games, including Poker (Moravčík et al., 2017; Brown & Sandholm, 2018, 2019), Go and Chess (Silver et al., 2016, 2018), Starcraft (Vinyals et al., 2019), Diplomacy (Paquette et al., 2019), and Stratego (Perolat et al., 2022).

Self-play has the desirable property that unbounded quantities of training data can be generated (assuming access to a simulator). But using self-play necessarily involves a choice of agents for the learner to train with: copies of itself. Strategies that perform well during training may perform poorly against new agents, whose behavior may differ dramatically from that of the agents that the learner trained against. The problem of learning strategies during training that perform well against new agents is a central challenge in algorithmic game theory and multi-agent reinforcement learning (MARL) (Matignon et al., 2012; Lanctot et al., 2017). In particular, Lanctot et al. (2017) show that interacting with agents from an independent self-play instance, differing only by random seed, can lead to dramatically worse performance.

There are special classes of environments where the strategies learned through self-play generalize well to new agents. In two-player, constant-sum games there exist strong theoretical results guaranteeing the performance of a strategy learned through self-play: Nash equilibrium strategies are

Preprint. Under review.

maxmin strategies, which will perform equally well against any optimal opponent and can guarantee the value of the game against any opponent.

We lose these guarantees outside of two-player constant-sum games. For example, consider the simple two-player coordination game of Figure 1. If both players choose the same action, both receive a utility of 1, otherwise they receive 0. Suppose the row player learned in self-play to choose a (which performs well against another a-player). Similarly, column learned to play b. If these two players played against each other, both agents would regret their actions. Upon the introduction of a new agent who did not train with a learner, despite a and b being optimal strategies during training, they fail to generalize to new agents. As this example demonstrates, equilibrium strategies in general are *vulnerable*: agents are not guaranteed the equilibrium's value against new agents–even if the new agent's also play an equilibrium strategy.

	a	b
a	$1, 1$	$0, 0$
b	$0, 0$	$1, 1$

Figure 1: A simple coordination game

A related problem can arise due to a loss of correlation post-training. Many algorithms converge to a *mediated equilibrium*, where a mediator recommends actions to each player (Von Stengel & Forges, 2008; Farina et al., 2019, 2020; Morrill et al., 2021b). The mediator can represent an external entity that makes explicit recommendations, such as traffic lights mediating traffic flows. More commonly in machine learning, correlation can arise through the shared history of actions of learning agents interacting with each other (Hart & Mas-Colell, 2000). In this second scenario, new agents would not have access to the actions taken by other agents during training, so players would no longer be able to correlate their actions. In fact, even if all agents play a decorrelated strategy from *the same* mediated equilibrium, the result may not be an equilibrium (please refer to the appendix for an example).

Despite the problems of equilibrium selection and loss of correlation, self-play has shown promising results outside of two-player constant-sum games. For example, algorithms based on self-play have outperformed professional poker players in multiplayer Texas hold-'em, despite the lack of theoretical guarantees (Brown & Sandholm, 2019)

We seek to understand what structure in multiplayer games allows for self-play to compute a good strategy. We show that any multiplayer game can be projected into a set of two-player constant-sum games between each pair of players, called *polymatrix games*. The closer a game is to this space, the less the problem of correlation affects the removal of a mediator. We additionally give sufficient conditions for strategies learned in these games to have the desirable properties of two-player constant-sum games, including the potential for any strategy learned in self-play to generalize to new agents. Throughout this work, we take an algorithm-agnostic approach by assuming only that self-play is performed by a regret minimizing algorithm. This is accomplished by analyzing directly the equilibria that no-regret algorithms converge to—namely coarse correlated equilibria. As a result, our analysis applies to a broad class of game-theoretically-inspired learning algorithms but also to MARL algorithms that converge to coarse correlated equilibria (Marris et al., 2021; Liu et al., 2021; Jin et al., 2021), since any policy can be transformed into a mixed strategy with Kuhn's Theorem (Kuhn, 1953).

Decomposition-based approaches have been used in prior work to show convergence of fictitious play to Nash equilibria in two-player games (Chen et al., 2022) and evolutionary dynamics (Tuyls et al., 2018). Cheung & Tao (2020) decompose games into zero-sum and cooperative parts to analyse the chaotic nature of Multiplicative Weights and Follow-the-Regularized-Leader. We focus less on converge of algorithms per se, and focus instead of the generalization of learned strategies to new agents and are, to the best of our knowledge, the first to do so.

After defining our structural properties and proving our main results, we conclude with experiments on Leduc poker to elucidate why self-play performs well in multi-player poker. Our results suggest that regret-minimization techniques converge to a subset of the game's strategy space that is approximately polymatrix.

2 Background

2.1 Normal Form Games

Normal Form Games A normal form game G is a 3 tuple $G = (N, \mathrm{P}, u)$ where N is a set of players, $\mathrm{P} = \bigtimes_{i \in N} \mathrm{P}_i$ is a joint pure strategy space and P_i is a set of *pure strategies* for player i.

Let $n = |N|$. Pure strategies are deterministic choices of actions in the game. We call $\rho \in \mathrm{P}$ a *pure strategy profile*. $u = (u_i)_{i \in N}$ is a set of *utility functions* where $u_i : \mathrm{P} \to \mathbb{R}$. A player i can randomize by playing a *mixed strategy*, a probability distribution s_i over i's pure strategies. Let $S_i = \Delta(\mathrm{P}_i)$ be the set of player i's mixed strategies (where $\Delta(X)$ denotes the set of probability distributions over a domain X), and let $S = \bigtimes_{i \in N} S_i$ be the set of mixed strategy profiles. We overload the definition of utility function to accept mixed strategies as follows: $u_i(s) = \sum_{\rho \in \mathrm{P}} \left(\prod_{i \in N} s_i(\rho_i) \right) u_i(\rho)$. We use ρ_{-i} and s_{-i} to denote a joint assignment of pure (resp. mixed) strategies to all players except for i, thus $s = (s_i, s_{-i})$.

Hindsight Rationality The hindsight rationality framework (Morrill et al., 2021b) conceptualizes the goal of an agent as finding a strategy that minimizes regret with respect to a set of deviations Φ. A deviation $\phi \in \Phi$ is a mapping $\phi : S_i \to S_i$ that transforms a learners strategy into some other strategy. Regret measures the amount the learner would prefer to deviate to $\phi(s_i)$: $u_i(\phi(s_i), s_{-i}) - u_i(s_i, s_{-i})$. An agent is hindsight rational with respect to a set of deviations Φ if the agent does not have positive regret with respect to any deviation in Φ, i.e. $\forall \phi \in \Phi, u_i(\phi(s_i), s_{-i}) - u_i(s_i, s_{-i}) \leq 0$. Let $\mu \in \Delta(\mathrm{P})$ be a distribution over pure strategy profiles and $(\Phi_i)_{i \in N}$ be a choice of deviation sets for each player.

Definition 2.1 (ϵ-Mediated Equilibrium (Morrill et al., 2021b))**.** We say $m = (\mu, (\Phi_i)_{i \in N})$ is an *ϵ-mediated equilibrium* if $\forall i \in N, \phi \in \Phi_i$ we have $\mathbb{E}_{\rho \sim \mu} \left[u_i(\phi(\rho_i), \rho_{-i}) - u_i(\rho) \right] \leq \epsilon$. A *mediated equilibrium* is a 0-mediated equilibrium.

Learning takes place in an online learning environment. At each iteration t, a learning agent i chooses a strategy s_i^t while all other agents choose a strategy profile s_{-i}^t. No-Φ-regret learning algorithms ensure that the maximum average positive regret tends to 0

$$\lim_{T \to \infty} \frac{1}{T} \left(\max_{\phi \in \Phi} \sum_{t=1}^{T} u_i(\phi(s_i^t), s_{-i}^t) - u_i(s_i^t, s_{-i}^t) \right) \to 0$$

If all agents use a no-regret learning algorithms w.r.t. a set of deviations Φ_i, the *empirical distribution of play* converges to a mediated equilibrium. Formally, let $\hat{\mu} \in \Delta(\mathrm{P})$ where $\hat{\mu}(\rho) \doteq \sum_{t=1}^{T} \left(\prod_{i \in N} s_i^t(\rho_i) \right)$ be the empirical distribution of play. As $T \to \infty$, $\hat{\mu}$ converges to μ of a mediated equilibrium $(\mu, (\Phi_i)_{i \in N})$.

For normal-form games, the set of all possible deviations, called swap deviations, is denoted Φ_{SW}; however the smaller set of internal deviations Φ_I, which exchanges a particular action recommended by the mediator with another, offers the same strategic power in hindsight (Foster & Vohra, 1999). The set of external deviations Φ_{EX} is even more restricted: $\phi \in \Phi_{EX}$ maps all (mixed) strategies to some particular pure strategy; i.e. $\Phi_{EX} = \{\phi \in \Phi_{SW} \mid \exists \rho_i, \forall s_i', \phi(s_i') = \rho_i\}$. The choice of $(\Phi_i)_{i \in N}$ determines the nature of the mediated equilibrium—provided the learning algorithm for player i is no-Φ_i-regret (Greenwald et al., 2011). For example, if all players are hindsight rational w.r.t. Φ_{EX}, then $\hat{\mu}$ converges to the well-known solution concept coarse correlated equilibrium (CCE) (Moulin & Vial, 1978) and if all players are hindsight rational w.r.t. Φ_I then $\hat{\mu}$ converges to a correlated equilibrium (Aumann, 1974).

A special case of mediated equilibria are *Nash equilibria*. If some mediated equilibrium $m = (\mu, (\Phi_i)_{i \in N})$ is a product distribution (i.e. $\mu = \bigotimes_{i \in N} s_i$ for $s_i \in S_i$) and $\Phi_i \supseteq \Phi_{EX}\ \forall i \in N$ then μ is a Nash equilibrium. Similarly an ϵ-mediated equilibrium is an ϵ-Nash equilibrium if μ is a product distribution and $\Phi_i \supseteq \Phi_{EX}\ \forall i \in N$.

In sequential decision making scenarios (often modelled as extensive form games), the set of deviations is even more rich (Morrill et al., 2021b). All of these deviation classes—with the exception of action deviations (Selten, 1988) (which are so weak they do not even imply Nash equilibria in two-player constant-sum games, see appendix)—are stronger than external deviations. This means that the equilibria of any algorithm that minimizes regret w.r.t. a stronger class of deviations than external deviations still inherit all the properties of CCE (for example Hart & Mas-Colell (2000); Zinkevich et al. (2008); Celli et al. (2020); Steinberger et al. (2020); Morrill et al. (2021a)). Thus, we focus on CCE since the analysis generalizes broadly. Moreover, CCE can be computed efficiently, either analytically (Jiang & Leyton-Brown, 2011) or by a learning algorithm. When we refer to CCE, we use the distribution μ to refer to the CCE, since Φ is implicit.

3 Self-Play and Generalization

The choice of other agents during learning affects the strategy that is learned. Choosing which agents make good "opponents" during training is an open research question (Lanctot et al., 2017; Marris et al., 2021). One common approach, *self-play*, is to have a learning algorithm train with copies of itself as the other agents. If the algorithm is a no-Φ-regret algorithm, the learned behavior will converge to a mediated equilibrium; this gives a nice characterization of the convergence behavior of the algorithm. For the remainder of this work, when we say "self-play" we are referring to self-play using a no-Φ-regret algorithm.

However, the strategies in a mediated equilibrium are correlated with each other. This means that in order to play a strategy learned in self-play, an agent must first extract it by marginalizing out other agent's strategies. This new *marginal strategy* can then be played against new agents with whom the agent did not train (and thus correlate).

Definition 3.1 (Marginal strategy)**.** Given some mediated equilibrium $(\mu, (\Phi_i)_{i=1}^N)$, let s_i^μ be the *marginal strategy* for i, where $s_i^\mu(\rho_i) \doteq \sum_{\rho_{-i} \in \mathrm{P}_{-i}} \mu(\rho_i, \rho_{-i})$. Let s^μ be a *marginal strategy profile*, where each $\forall i \in N$ plays s_i^μ.

Once a strategy has been extracted via marginalization, learning can either continue with the new agents (and potentially re-correlate), or the strategy can remain fixed. We focus on the case where the strategy remains fixed. In doing so we can guarantee the performance of this strategy if learning stops, but also the show guarantees about the initial performance of a strategy that continues to learn; this is especially important in safety-critical domains.

Given a marginal strategy s_i^μ, we can bound its underperformance against new agents that behave differently from the (decorrelated) training opponents by its vulnerability.

Definition 3.2 (Vulnerability)**.** The *vulnerability* of a strategy profile s for player i with respect to $S'_{-i} \subseteq S_{-i}$ is

$$\mathrm{Vul}_i\left(s, S'_{-i}\right) \doteq u_i(s) - \min_{s'_{-i} \in S'_{-i}} u_i(s_i, s'_{-i}).$$

Vulnerability gives a measure of how much worse s will perform with new agents compared to its training performance under pessimistic assumptions—that $-i$ play the strategy profile in S'_{-i} that is worst for i. We assume that $-i$ are not able to correlate their strategies.

Thus, if a marginal strategy profile s^μ is learned through self-play and $\mathrm{Vul}_i\left(s^\mu, S'_{-i}\right)$ is small, then s_i^μ performs roughly as well against new agents $-i$ playing some strategy profile in S'_{-i}. S'_{-i} is used to encode assumptions about the strategies of opponents. $S'_{-i} = S_{-i}$ means opponents could play *any* strategy, but we could also set S'_{-i} to be the set of strategies learnable through self-play if we believe that opponents would also be using self-play as a training procedure.

Some games have properties that make the vulnerability low. For example, in two-player constant-sum games the marginal strategies learned in self-play generalize well to new opponents since any Nash equilibrium strategy is also a maxmin strategy (von Neumann, 1928).

3.1 Polymatrix Games

Multiplayer games are fundamentally more complex than two-player constant-sum games (Daskalakis & Papadimitriou, 2005; Daskalakis et al., 2009). However, certain multiplayer games can be decomposed into a graph of two-player games, where a player's payoffs depend only on their actions and the actions of players who are neighbours in the graph (Bergman & Fokin, 1998). In these *polymatrix* games (a subset of graphical games (Kearns et al., 2013)) Nash equilibria can be computed efficiently if player's utilities sum to a constant (Cai & Daskalakis, 2011; Cai et al., 2016).

Definition 3.3 (Polymatrix game)**.** A *polymatrix game* $G = (N, E, \mathrm{P}, u)$ consists of a set N of players, a set of edges E between players, a set of pure strategy profiles P, and a set of utility functions $u = \{u_{ij}, u_{ji} \mid \forall (i,j) \in E\}$ where $u_{ij}, u_{ji} : \mathrm{P}_i \times \mathrm{P}_j \to \mathbb{R}$ are utility functions associated with the edge (i, j).

We refer to the normal-form *subgame* between (i, j) as $G_{ij} = (\{i, j\}, \mathrm{P}_i \times \mathrm{P}_j, (u_{ij}, u_{ji}))$. We use u_i to denote the *global utility function* $u_i : \mathrm{P} \to \mathbb{R}$ where $u_i(\rho) = \sum_{(i,j) \in E} u_{ij}(\rho_i, \rho_j)$ for each player. We use $E_i \subseteq E$ to denote the set of edges where i is a player.

Definition 3.4 (Constant-sum polymatrix)**.** We say a polymatrix game G is *constant-sum* if for some constant c we have that $\forall \rho \in \mathrm{P}, \sum_{i \in N} u_i(\rho) = c$.

Constant-sum polymatrix (CSP) games have the desirable property that all CCE factor into a product distribution; i.e., are Nash equilibria (Cai et al., 2016). We give a relaxed version:

Proposition 3.5. *If μ is an ϵ-CCE of a CSP game G, s^μ is an $n\epsilon$-Nash of G.*

This means no-external-regret learning algorithms will converge to Nash equilibria, and thus do not require a mediator to enable the equilibrium. However, they do not necessarily have the property of two-player constant-sum games that all (marginal) equilibrium strategies are maxmin strategies (Cai et al., 2016). Thus Nash equilibrium strategies in CSP games have no vulnerability guarantees. Cai et al. (2016) show that CSP games that are constant sum in each subgame are no more or less general than CSP games that are constant sum globally, since there exists a payoff preserving transformation between the two sets. For this reason we focus on CSP games that are constant sum in each subgame without loss of generality. Note that the constant need not be the same in each subgame.

4 Self-Play in Multiplayer Games

4.1 Subgame Stability

However, some polymatrix games *do* have have bounded vulnerability; we call these *subgame stable games*. In subgame stable games, global equilibria imply equilibria at each pairwise subgame.

Definition 4.1 (Subgame stable profile)**.** Let G be a polymatrix game with global utility functions $(u_i)_{i \in N}$. We say a strategy profile s is γ-*subgame stable* if $\forall (i,j) \in E$, we have (s_i, s_j) is a γ-Nash of G_{ij}; that is $u_{ij}(\rho_i, s_j) - u_{ij}(s_i, s_j) \leq \gamma \quad \forall \rho_i \in \mathrm{P}_i$ and $u_{ji}(\rho_j, s_i) - u_{ji}(s_j, s_i) \leq \gamma \quad \forall \rho_j \in \mathrm{P}_j$

Definition 4.2 (Subgame stable game)**.** Let G be a polymatrix game. We say G is (ϵ, γ)-*subgame stable* if for *any* ϵ-Nash equilibrium s of G, s is γ-subgame stable.

Subgame stability connects the global behavior of play (ϵ-Nash equilibrium in G) to local behavior in a subgame (γ-Nash in G_{ij}). If a polymatrix game is both constant-sum and is $(0, \gamma)$-subgame stable then we can bound the vulnerability of any marginal strategy.

Theorem 4.3. *Let G be a CSP game. If G is $(0, \gamma)$-subgame stable, then for any CCE μ of G, we have* $\mathrm{Vul}_i\left(s^\mu, S_{-i}\right) \leq |E_i|\gamma$.

Theorem 4.3 tells us that using self-play to compute a marginal strategy s^μ on constant-sum polymatrix games will have low vulnerability against worst-case opponents if γ is low. Thus, these are a set of multiplayer games where self-play is an effective training procedure.

Proof idea. Since G is a CSP game, s^μ is a Nash equilbrium. Since G is $(0, \gamma)$-subgame stable, (s_i^μ, s_j^μ) is a γ-Nash equilibrium in each subgame, which bounds the vulnerability (which coincides with exploitability in 2-player constant-sum games) in each subgame. This is because

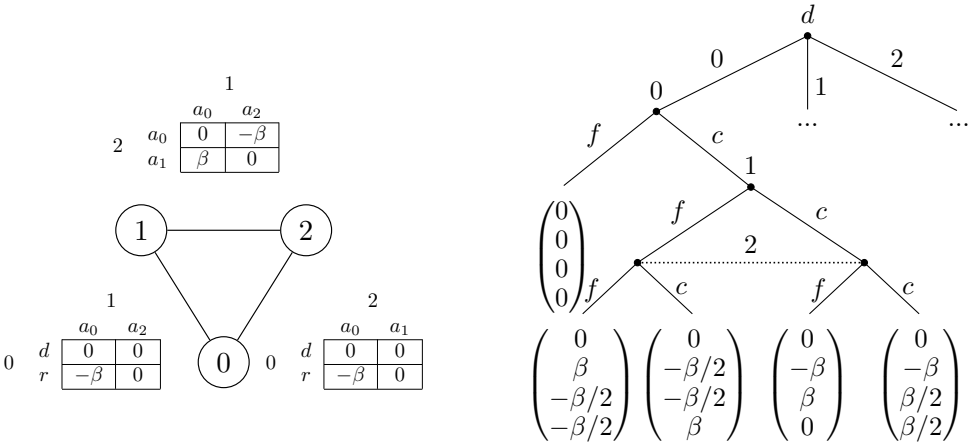
$$\min_{s_{-i} \in S_{-i}} u_i(s_i^\mu, s_{-i}) = \sum_{(i,j) \in E_i} \min_{s_j \in S_j} u_{ij}(s_i^\mu, s_j)$$

since players $j \neq i$ can minimize i's utility without coordinating, as G is a polymatrix game.

4.2 Vulnerability on a Simple Polymatrix Game

We next demonstrate the relationship between subgame stability and vulnerability on a simple 3-player game called Offense-Defense (Figure 2a). There are 3 players: 0, 1 and 2. Players 1 and 2 have the option to either attack 0 (a_0) or attack the other (e.g. a_1); player 0, on the other hand, may either relax (r) or defend (d). If either 1 or 2 attacks the other while the other is attacking 0, the attacker gets β and the other gets $-\beta$ in that subgame. If both 1 and 2 attack 0, 1 and 2 get 0 in their subgame and if they attack each other, their attacks cancel out and they get 0. If 0 plays d, they defend and will always get 0. If they relax, they get $-\beta$ if they are attacked and 0 otherwise. Offense-Defense is a CSP game, so any CCE is a Nash equilibrium.

Note that $\rho = (r, a_2, a_1)$ is a Nash equilibrium. Each $i \in \{1, 2\}$ are attacking the other $j \in \{1, 2\} \setminus \{i\}$, so has expected utility of 0. Deviating to attacking 0 would leave them open against



(a) (b)

Figure 2: (a) Offense-Defense, a simple CSP polymatrix game. We only show payoffs for the row player, column player payoffs are zero minus the row player's payoffs. (b) Bad Card: a game that is not overall polymatrix, but the subset of strategies learnable by self-play are. At the terminals, we show the dealers utility first, followed by players $0, 1$ and 2, respectively.

the other, so a_0 is not a profitable deviation, as it would also give utility 0. Additionally, 0 has no incentive to deviate to d, since this would also give them a utility of 0. However, ρ is β-subgame stable, since all $i \in \{1, 2\}$ has a profitable deviation in their subgame against 0. However, if 1 and 2 were to both deviate to a_0, and 0 continues to play their Nash equilibrium strategy of r, 0 would lose 2β utility from their equilibrium value; in other words, the vulnerability of player 0 is 2β.

4.3 Approximate Polymatrix Games

Most games are not factorizable into polymatrix games. However, we can take any game G and project it into the space of constant-sum polymatrix (CSP) games.

Definition 4.4 (δ-constant sum polymatrix)**.** A game G is *δ-constant sum polymatrix* (δ-CSP) if there exists a CSP game $\check{G}$ with global utility function $\check{u}$ such that $\forall i \in N, \rho \in \mathrm{P}$, $|u_i(\rho) - \check{u}_i(\rho)| \leq \delta$. We denote the set of such CSP games as $\mathrm{CSP}_\delta(G)$.

Proposition 4.5. *In a δ-CSP game G the following hold.*

1. *Any CCE of G is a 2δ-CCE of any $\check{G} \in \mathrm{CSP}_\delta(G)$.*
2. *The marginal strategy profile of any CCE of G is a $2n\delta$-Nash equilibrium of any $\check{G} \in \mathrm{CSP}_\delta(G)$.*
3. *The marginal strategy profile of any CCE of G is a $2(n+1)\delta$-Nash equilibrium of G.*

From (3) we have that the removal of the mediator impacts players utilities by a bounded amount in δ-CSP games. We give a linear program in the appendix that will find the minimum δ such that G is δ-CSP and returns a CSP game $\check{G} \in \mathrm{CSP}_\delta(G)$.

Combining δ-CSP with (ϵ, γ)-subgame stability lets us bound vulnerability in *any* game.

Theorem 4.6. *If G is δ-CSP and $\exists \check{G} \in \mathrm{CSP}_\delta(G)$ that is $(2n\delta, \gamma)$-subgame stable and μ is a CCE of G, then*

$$\mathrm{Vul}_i\left(s^\mu, S_{-i}\right) \leq |E_i|\gamma + 2\delta \leq (n-1)\gamma + 2\delta.$$

Theorem 4.6 shows that games that are close to the space of subgame stable CSP games are cases where the marginal strategies learned through self-play have bounded worst-case performance. This makes them suitable for any no-external-regret learning algorithm.

5 Vulnerability Against Other Self-Taught Agents

Theorem 4.6 bounds vulnerability in the worst-case scenarios, where $-i$ play any strategy profile to minimize i's utility. In reality, however, each player $j \in -i$ has their own interests and would only play a strategy that is reasonable under these own interests. In particular, what if each agent was also determining their own strategy via self-play in a separate training instance. How much utility can i guarantee themselves in this set up?

No-external-regret learning algorithms converge to the set of CCE. However, other assumptions can be made with additional information about the type of regret being minimized. No-external-regret learning algorithms will also play strictly dominated strategies with vanishing probability and CFR will play dominated actions with vanishing probability (Gibson, 2014). These refinements can also tighten our bounds, since the part of the game that no-regret learning algorithms converge to might be closer to a CSP game than the game overall.

Consider the game shown in Figure 2b, called "Bad Card". The game starts with each player putting $\beta/2$ into the pot. A dealer player d—who receives utility 0 regardless of the strategies of the other players—then selects a player from $\{0, 1, 2\}$ to receive a "bad card", while the other two players receive a "good card". The player who receives the bad card has an option to fold, after which the game ends and all players receive their ante back. Otherwise if this player calls, the other two players can either fold or call. The pot of β is divided among the players with good cards who call. If one player with a good card calls, they win the pot of β. If both good card players call then they split the pot. If both players with good cards fold, then the player with the bad card wins the pot.

Bad Card does not have a constant-sum polymatrix decomposition; in fact it does not have any polymatrix decomposition. Since Bad Card is an extensive-form game without chance[1], each pure strategy profile leads to a single terminal history. Let $\mathrm{P}(z)$ be the set of pure strategy profiles that play to a terminal z. In order for Bad Card to be polymatrix, we would need to find subgame utility functions such that $\forall \rho \in \mathrm{P}$, $u_0(\rho) = u_{0,d}(\rho_0, \rho_d) + u_{0,1}(\rho_0, \rho_1) + u_{0,2}(\rho_0, \rho_2)$, equivalently, we could write $\forall z \in Z, \rho \in \mathrm{P}(z), u_0(z) = u_{0,d}(\rho_0, \rho_d) + u_{0,1}(\rho_0, \rho_1) + u_{0,2}(\rho_0, \rho_2)$ where Z is the set of terminals. A subset of these constraints results in an infeasible system of equations.

Consider the terminals in the subtree shown in Figure 2b: $z^1 = (0, c, c, c)$, $z^2 = (0, c, c, f)$, $z^3 = (0, c, f, c)$ and $z^4 = (0, c, f, f)$. Let ρ_i^c be any pure strategy that plays c in this subtree and ρ_i^f be any strategy that plays f in this subtree for player i. In order for Bad Card to decompose into a polymatrix game we would need to solve the following infeasible system of linear equations:

$$
\begin{aligned}
u_0(z^1) &= u_{0,d}(\rho_0^c, 0) + u_{0,1}(\rho_0^c, \rho_1^c) + u_{0,2}(\rho_0^c, \rho_2^c) = -\beta \\
u_0(z^2) &= u_{0,d}(\rho_0^c, 0) + u_{0,1}(\rho_0^c, \rho_1^c) + u_{0,2}(\rho_0^c, \rho_1^f) = -\beta \\
u_0(z^3) &= u_{0,d}(\rho_0^c, 0) + u_{0,1}(\rho_0^c, \rho_1^f) + u_{0,2}(\rho_0^c, \rho_2^c) = -\beta \\
u_0(z^4) &= u_{0,d}(\rho_0^c, 0) + u_{0,1}(\rho_0^c, \rho_1^f) + u_{0,2}(\rho_0^c, \rho_2^f) = \beta
\end{aligned}
$$

Thus, Bad Card is not a constant-sum polymatrix game, although it is a β-CSP game. However, if we prune out dominated actions (i.e., those in which a player folds after receiving a good card), the resulting game is indeed a 0-CSP game.

Let $\mathcal{M}(\mathcal{A})$ be the set of mediated equilibria than an algorithm $\mathcal{A}$ converges to. For example, if $\mathcal{A}$ is a no-external-regret algorithm, $\mathcal{M}(\mathcal{A})$ is the set of CCE without strictly dominated strategies in their support. Likewise, if $\mathcal{A}$ is CFR, $\mathcal{M}(\mathcal{A})$ is the set of (CF)(CCE) (Morrill et al., 2021b). Let $S(\mathcal{A}_i) = \{s^\mu \mid (\mu, (\Phi_i)_{i \in N}) \in \mathcal{M}(\mathcal{A}_i)\}$ be the set of marginal strategy profiles of $\mathcal{M}(\mathcal{A}_i)$, and let $S_i(\mathcal{A}_i) = \{s_i \mid s \in S(\mathcal{A}_i)\}$ be the set of i's marginal strategies from $S(\mathcal{A}_i)$.

Now, consider if each player i learns with their own self-play algorithm $\mathcal{A}_i$. Let $\mathcal{A}^N = (\mathcal{A}_1, ...\mathcal{A}_n)$ be the profile of learning algorithms, then $S(\mathcal{A}^N) = \bigtimes_{i \in N} S_i(\mathcal{A}_i)$. Summarizing, if each player learns with a no-regret-learning algorithm $\mathcal{A}_i$, they will converge to the set of $\mathcal{M}(\mathcal{A}_i)$ equilibrium. The set of marginal strategies from this set of equilibria is $S_i(\mathcal{A}_i)$ and the set of marginal strategy profiles is $S(\mathcal{A}_i)$. If each player plays a (potentially) different learning algorithm, $S(\mathcal{A}^N)$ is the set of possible joint match-ups if each player plays a marginal strategy from their own algorithms set of equilibria.

[1] See the appendix for details on extensive-form games.

Definition 5.1. We say a game G is *δ-CSP in the neighborhood of $S' \subseteq S$* if there exists a CSP game $\check{G}$ such that $\forall s \in S'$ we have $|u_i(s) - \check{u}_i(s)| \leq \delta$. We denote the set of such CSP games as $\mathrm{CSP}_\delta(G, S')$.

Definition 5.2. We say a CSP game G is *γ-subgame stable in the neighborhood of S'* if $\forall s \in S', \forall (i,j) \in E$ we have that (s_i, s_j) is a γ-Nash of G_{ij}.

These definitions allow us to prove the following generalization of Theorem 4.6.

Theorem 5.3. *If G is δ-CSP in the neighborhood of $S(\mathcal{A}^N)$ and $\exists \check{G} \in \mathrm{CSP}_\delta(G, S(\mathcal{A}^N))$ that is γ-subgame stable in the the neighborhood of $S(\mathcal{A}_i)$ $\forall i \in N$, then for any $s \in S(\mathcal{A}_i)$*

$$\mathrm{Vul}_i\left(s, S(\mathcal{A}^N)\right) \leq |E_i|\gamma + 2\delta \leq (n-1)\gamma + 2\delta.$$

An implication of Theorem 5.3 is that if agents use self-play to compute a marginal strategy from some mediated equilibrium and there is a subgame stable CSP game that is close to the original game for these strategies, then this is sufficient to bound vulnerability against strategies learned in self-play.

6 Experiments on Leduc Poker

Approaches using regret-minimization in self-play have been shown to outperform expert human players in some multiplayer games, the most notable example being multiplayer no-limit Texas hold-'em (Brown & Sandholm, 2019), despite no formal guarantees.

Conjecture 6.1. Self-play with regret minimization performs well in multiplayer Texas hold-'em because "good" players (whether professional players or strategies learned by self-play) play in a part of the games' strategy space that is close to a subgame stable CSP game for some low values of γ, δ.

While multiplayer no-limit Texas hold-'em is too large to directly check the properties developed in this work, we use a smaller poker game, called Leduc poker (Southey et al., 2012), to suggest why regret-minimization "works" in multiplayer Texas hold-'em. Leduc poker was originally developed for two players but was extended to a 3-player variant by Abou Risk et al. (2010). The game has 8 cards with 4 ranks and 2 suits. At the start, each player antes 1 and receives one card privately before an initial round of betting commences. A public card is then revealed before the second round of betting. The bet values in the first and second round are 2 and 4, respectively.

We use CFR to compute a set of approximate marginal strategies.[2] Vanilla CFR is a deterministic algorithm, so we use different random initializations of CFR's initial strategy in order to generate a set of CCE. We train 30 random initializations of CFR for 10,000 iterations. CFR computes these marginal strategies by averaging its iterates.[3]

Making this formal, we use $\mathcal{A}_i = \mathrm{CFR}$ as a learning algorithm for each $i \in N$. Then $S(\mathrm{CFR})$ is the set of strategy profiles learned by CFR, with $S_i(\mathrm{CFR})$ be the set of mixed strategies for player i and $S(\mathrm{CFR}^N) = \bigtimes_{i \in N} S_i(\mathrm{CFR})$ is the set of match ups between agents trained in self-play with CFR.

We used the following algorithm to find a CSP game $\check{G}$ that minimizes our bounds, by iteratively adding constraints to a linear program until we find a CSP game that is subgame-stable in the neighborhood of $S(\mathrm{CFR})$, after which we output the δ so that $\check{G}$ is δ-CSP in the neighborhood of $S(\mathrm{CFR}^N)$. We start with CSP game $\check{G}^0$ with utility function $\check{u}^0_{ij}(\cdot) = 0 \; \forall i \neq j$. At each iteration $t \geq 1$, we compute a best response for each player i in each subgame G^{t-1}_{ij} against each CFR strategy of their opponent $s_j \in S_j(\mathrm{CFR})$: $s^t_{ij} \in \arg\max_{s'_i \in S_i} \check{u}^{t-1}_{ij}(s'_i, s_j)$. This strategy is added to a set of strategies $\mathrm{BR}^{t-1}_{ij}(s_j) = \{s^1_{ij}, \ldots s^{t-1}_{ij}\}$ to produce $\mathrm{BR}^t_{ij}(s_j) = \{s^1_{ij}, \ldots s^t_{ij}\}$. We then solve the

[2] We use the OpenSpiel implementation (Lanctot et al., 2019).

[3] Note that these average strategies do not converge to CCE themselves; computing a CCE requires additional machinery; for example, CFR-JR is a variant of CFR that does indeed converge to a CCE (Celli et al., 2019).

following linear program to produce $\check{G}^t$ with utility functions $\check{u}^t_{ij}(\cdot)$.

$$\begin{aligned}
\min_{\gamma^t, \delta^t, (\check{u}^t_{ij}(\cdot), c^t_{ij})_{\forall i,j}} \quad & (n-1)\gamma^t + 2\delta^t \\
\text{s.t.} \quad & \check{u}^t_{ij}(s'_i, s_j) - u^t_{ij}(s_i, s_j) \leq \gamma^t \quad \forall i \neq j \in N, s'_i \in \mathrm{BR}^t_{ij}(s_j), s \in S(\mathrm{CFR}) \\
& \sum_{(i,j)\in E_i} \check{u}^t_{ij}(s_i, s_j) - u_i(s) \leq \delta^t \quad \forall i \in N, s \in S(\mathrm{CFR}) \\
& u_i(s) - \sum_{(i,j)\in E_i} \check{u}^t_{ij}(s_i, s_j) \leq \delta^t \quad \forall i \in N, s \in S(\mathrm{CFR}) \\
& \check{u}^t_{ij}(\rho) + u^t_{ji}(\rho) = c^t_{ij} \quad \forall i \neq j \in N \times N, \rho \in \mathrm{P} \\
& \delta^t, \gamma^t \geq 0
\end{aligned}$$

The algorithm terminates when $\forall i, j, s_j \in S_j(\mathrm{CFR})$ we have $s^t_{ij} = s^{t-1}_{ij}$. At this point all s^t_{ij} are best responses w.r.t. $u^t_{ij}(\cdot)$, so $\check{G}^t$ is γ^t-subgame stable $\forall i \in N$. Lastly, we compute $\delta = \max_{s \in S(\mathrm{CFR}^N)} \left| u_i(s) - \sum_{(i,j)\in E_i} u^t_{ij}(s_i, s_j) \right|$. Note that we implemented this algorithm using the extensive-form representation of Leduc poker for compactness and efficiency. We express the algorithm here using mixed strategies for brevity; please refer to the appendix for more detail.

Using this algorithm, we found a CSP game $\check{G}$ that was 0-subgame stable in the neighborhood of the CFR strategy profiles (i.e. the CFR profiles are Nash equilibria in *every* subgame between players) and $\delta = 0.097$-CSP (i.e. we need to perturb the utility function by at most 0.097), implying that $\forall i \in N, s \in S(\mathrm{CFR})$ we have $\mathrm{Vul}_i\left(s, S(\mathrm{CFR}^N)\right) \leq 0.194$. For better intuition, consider normalizing our bounds by the range of the utility function to give a relative value $\mathrm{Vul}^{\mathrm{rel}} = 2(n-1)\gamma + 2\delta / \left(\max_{\rho\in\mathrm{P}} u_i(\rho) - \min_{\rho\in\mathrm{P}} u_i(\rho)\right)$. In Leduc poker, $\mathrm{Vul}^{\mathrm{rel}} = 0.194/12 \approx 0.016$. In other words, because CFR converges to a part of the game that is near polymatrix and subgame stable, players can lose at most 1.6% of the range of their utility functions against players from a separate training instance. How vulnerable are the CFR strategies in actuality? We computed the vulnerability with respect to the CFR-computed strategies by evaluating each strategy against each other and taking the maximum. We found that $\mathrm{Vul}^{\mathrm{rel}} = 0.009/12 \approx 0.0008$; our bounds thus do a good job of bounding vulnerability, but are not fully tight.

It was previously believed that CFR does not compute an ϵ-Nash equilibrium on 3-player Leduc for any reasonable value of ϵ. Abou Risk et al. (2010) found CFR computed a 0.130-Nash equilibrium after 10^8 iterations. However, we found that *all* of our strategies converged to an approximate Nash equilibrium with the maximum $\epsilon = 0.017$ after only 10^4 iterations.

Summarizing, we found that the part of the strategy space of Leduc poker that CFR converges to is close to a subgame stable CSP game. If new players have strategies within this part of Leduc poker (e.g., if they also learned with a different initialization of CFR), then the strategies learned by CFR bound the loss in utility against these new players.

7 Conclusion

Self-play has been incredibly successful in producing strategies that perform well against new opponents in two-player constant-sum games. Despite a lack of theoretical guarantees, self-play seems to also produce good strategies in some multiplayer games (Brown & Sandholm, 2019). We identify a structural property of multiplayer, general-sum game that allow us to establish guarantees on the performance of strategies learned via self-play against new opponents. We show that any game can be projected into the space of constant-sum polymatrix games, and if there exists a game with this set with high subgame stability (low γ), strategies learned through self-play have bounded loss of performance against new opponents.

We conjecture that Texas hold-'em is one such game. We investigate this claim on Leduc poker, and find that CFR plays strategies from a nearly subgame stable part of the strategy space within Leduc poker. This work lays the groundwork for guarantees for self-play in multiplayer games. However, there is room for algorithmic improvement and efficiency gains for checking these properties in very large extensive-form games.

References

Abou Risk, N., Szafron, D., et al. Using counterfactual regret minimization to create competitive multiplayer poker agents. *International Conference on Autonomous Agents and Multiagent Systems*, 2010.

Aumann, R. J. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1(1):67–96, 1974.

Bergman, L. and Fokin, I. On separable non-cooperative zero-sum games. *Optimization*, 44(1): 69–84, 1998.

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

Brown, N. and Sandholm, T. Superhuman AI for multiplayer poker. *Science*, 365(6456):885–890, 2019.

Cai, Y. and Daskalakis, C. On minmax theorems for multiplayer games. *ACM-SIAM Symposium on Discrete algorithms*, 2011.

Cai, Y., Candogan, O., Daskalakis, C., and Papadimitriou, C. Zero-sum polymatrix games: A generalization of minmax. *Mathematics of Operations Research*, 41(2):648–655, 2016.

Celli, A., Marchesi, A., Bianchi, T., and Gatti, N. Learning to correlate in multi-player general-sum sequential games. *Neural Information Processing Systems*, 2019.

Celli, A., Marchesi, A., Farina, G., and Gatti, N. No-regret learning dynamics for extensive-form correlated equilibrium. *Neural Information Processing Systems*, 2020.

Chen, Y., Deng, X., Li, C., Mguni, D., Wang, J., Yan, X., and Yang, Y. On the convergence of fictitious play: A decomposition approach. *International Joint Conferences on Artificial Intelligence*, 2022.

Cheung, Y. K. and Tao, Y. Chaos of learning beyond zero-sum and coordination via game decompositions. *arXiv preprint arXiv:2008.00540*, 2020.

Daskalakis, C. and Papadimitriou, C. H. Three-player games are hard. *Electron. Colloquium Comput. Complex.*, 2005.

Daskalakis, C., Goldberg, P. W., and Papadimitriou, C. H. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

Farina, G., Ling, C. K., Fang, F., and Sandholm, T. Correlation in extensive-form games: Saddle-point formulation and benchmarks. *Neural Information Processing Systems*, 2019.

Farina, G., Bianchi, T., and Sandholm, T. Coarse correlation in extensive-form games. *AAAI conference on Artificial Intelligence*, 2020.

Foster, D. P. and Vohra, R. Regret in the on-line decision problem. *Games and Economic Behavior*, 29(1):7–35, 1999.

Fudenberg, D. and Tirole, J. *Game Theory*. MIT Press Books. The MIT Press, 1991.

Gibson, R. G. Regret minimization in games and the development of champion multiplayer computer poker-playing agents. *Ph.D. Thesis*, 2014.

Greenwald, A., Jafari, A., and Marks, C. No-ϕ-regret: A connection between computational learning theory and game theory. *Games, Norms and Reasons: Logic at the Crossroads*, 2011.

Hart, S. and Mas-Colell, A. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000.

Jiang, A. X. and Leyton-Brown, K. A general framework for computing optimal correlated equilibria in compact games. *Internet and Network Economics: 7th International Workshop, WINE 2011*, 2011.

Jin, C., Liu, Q., Wang, Y., and Yu, T. V-learning–a simple, efficient, decentralized algorithm for multiagent rl. *arXiv preprint arXiv:2110.14555*, 2021.

Kearns, M., Littman, M. L., and Singh, S. Graphical models for game theory, 2013.

Kuhn, H. W. Extensive games and the problem of information. *Contributions to the Theory of Games*, 2(28):193–216, 1953.

Lanctot, M., Zambaldi, V., Gruslys, A., Lazaridou, A., Tuyls, K., Perolat, J., Silver, D., and Graepel, T. A unified game-theoretic approach to multiagent reinforcement learning. *Neural Information Processing Systems*, 2017.

Lanctot, M., Lockhart, E., Lespiau, J.-B., Zambaldi, V., Upadhyay, S., Pérolat, J., Srinivasan, S., Timbers, F., Tuyls, K., Omidshafiei, S., et al. Openspiel: A framework for reinforcement learning in games. *arXiv preprint arXiv:1908.09453*, 2019.

Liu, Q., Yu, T., Bai, Y., and Jin, C. A sharp analysis of model-based reinforcement learning with self-play. *International Conference on Machine Learning*, 2021.

Marris, L., Muller, P., Lanctot, M., Tuyls, K., and Graepel, T. Multi-agent training beyond zero-sum with correlated equilibrium meta-solvers. *International Conference on Machine Learning*, 2021.

Matignon, L., Laurent, G. J., and Le Fort-Piat, N. Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. *The Knowledge Engineering Review*, 27(1):1–31, 2012.

Moravčík, M., Schmid, M., Burch, N., Lisỳ, V., Morrill, D., Bard, N., Davis, T., Waugh, K., Johanson, M., and Bowling, M. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017.

Morrill, D., D'Orazio, R., Lanctot, M., Wright, J. R., Bowling, M., and Greenwald, A. R. Efficient deviation types and learning for hindsight rationality in extensive-form games. *International Conference on Machine Learning*, 2021a.

Morrill, D., D'Orazio, R., Sarfati, R., Lanctot, M., Wright, J. R., Greenwald, A. R., and Bowling, M. Hindsight and sequential rationality of correlated play. *AAAI Conference on Artificial Intelligence*, 2021b.

Moulin, H. and Vial, J. Strategically zero-sum games: The class of games whose completely mixed equilibria cannot be improved upon. *International Journal of Game Theory*, 7(3):201–221, 1978.

Paquette, P., Lu, Y., Bocco, S. S., Smith, M., O-G, S., Kummerfeld, J. K., Pineau, J., Singh, S., and Courville, A. C. No-press diplomacy: Modeling multi-agent gameplay. *Neural Information Processing Systems*, 2019.

Perolat, J., Vylder, B. D., Hennes, D., Tarassov, E., Strub, F., de Boer, V., Muller, P., Connor, J. T., Burch, N., Anthony, T., McAleer, S., Elie, R., Cen, S. H., Wang, Z., Gruslys, A., Malysheva, A., Khan, M., Ozair, S., Timbers, F., Pohlen, T., Eccles, T., Rowland, M., Lanctot, M., Lespiau, J.-B., Piot, B., Omidshafiei, S., Lockhart, E., Sifre, L., Beauguerlange, N., Munos, R., Silver, D., Singh, S., Hassabis, D., and Tuyls, K. Mastering the game of stratego with model-free multiagent reinforcement learning. *Science*, 378(6623):990–996, 2022.

Selten, R. Reexamination of the perfectness concept for equilibrium points in extensive games. In *Models of Strategic Rationality*, pp. 1–31. Springer, 1988.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., and Hassabis, D. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419): 1140–1144, 2018.

Southey, F., Bowling, M. P., Larson, B., Piccione, C., Burch, N., Billings, D., and Rayner, C. Bayes' bluff: Opponent modelling in poker. *arXiv preprint arXiv:1207.1411*, 2012.

Steinberger, E., Lerer, A., and Brown, N. Dream: Deep regret minimization with advantage baselines and model-free learning. *arXiv preprint arXiv:2006.10410*, 2020.

Tuyls, K., Pérolat, J., Lanctot, M., Ostrovski, G., Savani, R., Leibo, J. Z., Ord, T., Graepel, T., and Legg, S. Symmetric decomposition of asymmetric games. *Scientific reports*, 8(1):1–20, 2018.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T. L., Gülçehre, Ç., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T. P., Kavukcuoglu, K., Hassabis, D., Apps, C., and Silver, D. Grandmaster level in Starcraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

von Neumann, J. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

Von Stengel, B. and Forges, F. Extensive-form correlated equilibrium: Definition and computational complexity. *Mathematics of Operations Research*, 33(4):1002–1022, 2008.

Zinkevich, M., Johanson, M., Bowling, M., and Piccione, C. Regret minimization in games with incomplete information. *Neural Information Processing Systems*, 2008.

A Decorrelated Strategies From a CCE Are Not a CCE

	a	b
a	$1, 0$	$-1, -1$
b	$-1, -1$	$0, 1$

Figure 3

Consider a distribution $\mu \in \Delta(\mathrm{P})$ s.t. $\mu(a, a) = 0.5$ and $\mu(b, b) = 0.5$. μ is a CCE: $\mathbb{E}_{\rho\sim\mu}\left[u_i(\rho)\right] = 0.5$ for each player. If row (r) were to player a and column continues to play according to μ, row's utility is 0; if r plays b instead, their utility is now -0.5. Thus r has no profitable deviations from the CCE recommendations. Column does not either.

Row's marginal strategy s_r^μ plays a with probability 0.5 and b with probability 0.5, s_c^μ does likewise. $u_r(s_r^\mu, s_c^\mu) = u_c(s_r^\mu, s_c^\mu) = -0.25$. However, a is a profitable deviation for r now since $0 > -0.25$, thus the decorrelated strategies a CCE are also not a CCE.

B Hindsight Rationality w.r.t. Action Deviations Does Not Imply Nash

Here we show that hindsight rationality with respect to action deviations does not imply Nash equilibrium in 2 player constant-sum games. We show this with a 1 player game and assume the second player is a dummy player. Consider the agents strategy, shown in blue, which receives utility of 1. Deviating to $[I_1 : b, I_2 : a]$ will increase the player's utility to 2, so the blue strategy is not a Nash equilibrium. However, this would require two simultaneous action deviations, one at I_1 to b and one at I_2 to a. Neither of these deviations increases the player's utility on their own, so the player is hindsight rational w.r.t. action deviations.

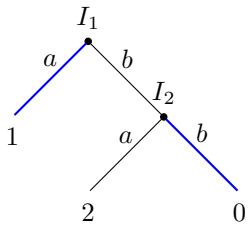


Figure 4

C Omitted Proofs

Proposition C.1 (Cai et al. (2016))**.** *In CSP games, for any CCE μ, if i deviates to s_i, then their expected utility if other players continue to play μ is equal to their utility if other player where to play the marginal strategy profile s_{-i}^μ:*

$$\mathbb{E}_{\rho\sim\mu}[u_i(s_i, \rho_{-i})] = u_i(s_i, s_{-i}^\mu)\ \forall s_i \in S_i$$

C.1 Proof of Proposition 3.5

Proposition 3.5. *If μ is an ϵ-CCE of a CSP game G, s^μ is an $n\epsilon$-Nash of G.*

Proof. Since μ is an ϵ-CCE, $\forall i \in N$, we have

$$\max_{\rho_i'\in\mathrm{P}_i} \mathbb{E}_{\rho\sim\mu}\left[u_i(\rho_i', \rho_{-i})\right] - \mathbb{E}_{\rho\sim\mu}\left[u_i(\rho)\right] \leq \epsilon$$

which implies (by Proposition C.1) that $\forall i \in N$

$$\max_{\rho_i'\in\mathrm{P}_i} u_i(\rho_i', s_{-i}^\mu) - \mathbb{E}_{\rho\sim\mu}\left[u_i(\rho)\right] \leq \epsilon$$

$$\implies \max_{\rho_i'\in\mathrm{P}_i} u_i(\rho_i', s_{-i}^\mu) \leq \epsilon + \mathbb{E}_{\rho\sim\mu}\left[u_i(\rho)\right]$$

Summing over N, we get

$$\sum_{i \in N} \max_{\rho_i' \in \mathrm{P}_i} u_i(\rho_i', s_{-i}^\mu) \leq \sum_{i \in N} \left(\epsilon + \mathbb{E}_{\rho \sim \mu}\left[u_i(\rho)\right]\right) \tag{1}$$

$$= \sum_{i \in N} \epsilon + \sum_{i \in N} \mathbb{E}_{\rho \sim \mu}\left[u_i(\rho)\right] \tag{2}$$

$$= n\epsilon + c \tag{3}$$

$$= n\epsilon + \sum_{i \in N} u_i(s^\mu) \tag{4}$$

Where (3) and (4) use the fact that $\forall \rho \in \mathrm{P}, \sum_{i \in N} u_i(\rho) = c$ for some constant. The above inequalities give us

$$\sum_{i \in N} \max_{\rho_i' \in \mathrm{P}_i} u_i(\rho_i', s_{-i}^\mu) \leq n\epsilon + \sum_{i \in N} u_i(s^\mu)$$

Rearranging, we get

$$\sum_{i \in N} \underbrace{\max_{\rho_i' \in \mathrm{P}_i} u_i(\rho_i', s_{-i}^\mu) - u_i(s^\mu)}_{\geq 0} \leq n\epsilon$$

All terms in the sum are are non-negative because ρ_i' is a best-response to s_{-i}^μ. Then any particular term in the summation is upper bounded by $n\epsilon$ and we are done. □

C.2 Proof of Theorem 4.3

Definition C.3 (Exploitability)**.** Let s be a strategy profile of a 2 player constant-sum game. The exploitability of s for player i is

$$u_i(s) - \min_{s_{-i}' \in S_{-i}} u_i(s_i, s_{-i}')$$

We say a strategy profile is ϵ-exploitable for player i if

$$u_i(s) - \min_{s_{-i}' \in S_{-i}} u_i(s_i, s_{-i}') \leq \epsilon$$

Proposition C.4 (Fudenberg & Tirole (1991))**.** *In two-player constant-sum games, the set of ϵ-Nash equilibria are ϵ-exploitable for each player.*

Theorem 4.3. *Let G be a CSP game. If G is $(0, \gamma)$-subgame stable, then for any CCE μ of G, we have* $\mathrm{Vul}_i\left(s^\mu, S_{-i}\right) \leq |E_i|\gamma$.

Proof. Any marginal strategy s^μ of a CCE μ is a Nash equilibrium of G (Cai et al., 2016). Then,

$$\begin{aligned}
\mathrm{Vul}_i\left(s^\mu, S_{-i}\right) &\doteq u_i(s^\mu) - \min_{s_{-i} \in S_{-i}} u_i(s_i^\mu, s_{-i}) \\
&= \sum_{(i,j) \in E_i} u_{ij}(s_i^\mu, s_j^\mu) - \min_{s_{-i} \in S_{-i}} \left(\sum_{(i,j) \in E_i} u_{ij}(s_i^\mu, s_j) \right) \\
&= \sum_{(i,j) \in E_i} u_{ij}(s_i^\mu, s_j^\mu) - \sum_{(i,j) \in E_i} \min_{s_j \in S_j} u_{ij}(s_i^\mu, s_j)
\end{aligned}$$

Where the last line uses the fact that G is polymatrix and $-i$ only care about minimizing $i's$ utility, so given s_i^μ can do so without coordinating. Continuing,

$$= \sum_{(i,j) \in E_i} \left(u_{ij}(s_i^\mu, s_j^\mu) - \min_{s_j \in S_j} u_i(s_i^\mu, s_j) \right) \tag{5}$$

$$\leq \sum_{(i,j) \in E_i} \gamma \tag{6}$$

$$\leq |E_i|\gamma \tag{7}$$

Where by $(0, \gamma)$-subgame stability of each G_{ij}, (s_i^μ, s_i^μ) is a γ-Nash of G_{ij}. By Proposition C.4, s^μ is γ-exploitable for i in each subgame since vulnerability coincides with exploitability for 2 player constant-sum games. □

C.3 Proof of Proposition 4.5

Proposition 4.5. *In a δ-CSP game G the following hold*

1. *Any CCE of G is a 2δ-CCE of any $\check{G} \in \mathrm{CSP}_\delta(G)$.*
2. *The marginalized strategy profile of any CCE of G is a $2n\delta$-Nash equilibrium of any $\check{G} \in \mathrm{CSP}_\delta(G)$.*
3. *The marginalized strategy profile of any CCE is a $2(n+1)\delta$-Nash equilibrium of G*

Proof. First we prove claim 1. Let $\check{u}_i$ denote the utility function of i in $\check{G}$. Note that $\forall \rho \in \mathrm{P}$ we have $|\check{u}_i(\rho) - u_i(\rho)| \leq \delta \; \forall i \in N$. Then,

$$\begin{aligned}
&\mathbb{E}_{\rho\sim\mu}\left[u_i(\rho_i', \rho_{-i}) - u_i(\rho)\right] \leq 0 \quad \forall i \in N, \rho_i' \in \mathrm{P}_i \\
\implies &\mathbb{E}_{\rho\sim\mu}\left[\check{u}_i(\rho_i', \rho_{-i}) - \delta - (\check{u}_i(\rho) + \delta)\right] \leq 0 \quad \forall i \in N, \rho_i' \in \mathrm{P}_i \\
\implies &\mathbb{E}_{\rho\sim\mu}\left[\check{u}_i(\rho_i', \rho_{-i}) - \check{u}_i(\rho)\right] \leq 2\delta \quad \forall i \in N, \rho_i' \in \mathrm{P}_i
\end{aligned}$$

Next, claim 2 is an immediate corollary of claim 1 and Proposition 3.5. Lastly, we show claim 3. By claim 2, we have the marginalized strategy of μ, s^μ is a $2n\delta$-Nash equilibrium of $\check{G} \in \mathrm{CSP}_\delta(G)$. That is,

$$\check{u}_i(\rho_i, s_{-i}^\mu) - \check{u}_i(s_i^\mu, s_{-i}^\mu) \leq 2n\delta \quad \forall i \in N, \rho_i \in \mathrm{P}_i$$

However, since G is δ-CSP:

$$\big(u_i(\rho_i, s_{-i}^\mu) - \delta\big) - \big(u_i(s_i^\mu, s_{-i}^\mu) + \delta)\big) \leq \check{u}_i(\rho_i, s_{-i}^\mu) - \check{u}_i(s_i^\mu, s_{-i}^\mu) \leq 2n\delta \quad \forall i \in N, \rho_i \in \mathrm{P}_i$$

Which gives us

$$u_i(\rho_i, s_{-i}^\mu) - u_i(s_i^\mu, s_{-i}^\mu) \leq 2n\delta + 2\delta = 2(n+1)\delta \quad \forall i \in N, \rho_i \in \mathrm{P}_i$$

□

C.4 Proof of Theorem 4.6

Theorem 4.6. *If G is δ-CSP and $\exists \check{G} \in \mathrm{CSP}_\delta(G)$ that is $(2n\delta, \gamma)$-subgame stable and μ is a CCE of G, then*

$$\mathrm{Vul}_i\left(s^\mu, S_{-i}\right) \leq |E_i|\gamma + 2\delta \leq (n-1)\gamma + 2\delta.$$

Proof. Let $\check{G} = (N, E, \mathrm{P}, \check{u})$ be a polymatrix games that is $(2n\delta, \gamma)$-subgame stable such that $\check{G} \in \mathrm{CSP}_\delta(G)$. Let $\check{u}_i$ denote the utility function of i in $\check{G}$. By Proposition 4.5, μ is a $2n\delta$-Nash equilibrium of $\check{G}$. Then

$$\begin{aligned}
\mathrm{Vul}_i\left(s^\mu, S_{-i}\right) &\doteq u_i(s^\mu) - \min_{s_{-i}\in S_{-i}} u_i(s_i^\mu, s_{-i}) \\
&\leq \check{u}_i(s^\mu) - \min_{s_{-i}\in S_{-i}} \check{u}_i(s_i^\mu, s_{-i}) + 2\delta
\end{aligned}$$

Since G is δ-CSP. Then

$$= \sum_{(i,j)\in E_i} \check{u}_{ij}(s_i^\mu, s_j^\mu) - \min_{s_{-i}\in S_{-i}} \sum_{(i,j)\in E_i} u_i(s_i^\mu, s_j) + 2\delta \tag{8}$$

$$= \sum_{(i,j)\in E_i} \check{u}_{ij}(s_i^\mu, s_j^\mu) - \sum_{(i,j)\in E_i} \min_{s_j\in S_j} u_i(s_i^\mu, s_j) + 2\delta \tag{9}$$

Where, as in Theorem 4.3, the last line uses the fact that $\check{G}$ is polymatrix, G_{ij} is constant-sum and $-i$ minimize $i's$ utility and can do so by without coordinating. Continuing, we have

$$\sum_{(i,j)\in E_i} \check{u}_{ij}(s_i^\mu, s_j^\mu) - \sum_{(i,j)\in E_i} \min_{s_j\in S_j} u_i(s_i^\mu, s_j) + 2\delta \tag{10}$$

$$= \sum_{(i,j)\in E_i} \left(\check{u}_{ij}(s_i^\mu, s_j^\mu) - \min_{s_j\in S_j} u_i(s_i^\mu, s_j) \right) + 2\delta \tag{11}$$

$$\leq \sum_{(i,j)\in E_i} \gamma + 2\delta \tag{12}$$

$$= |E_i|\gamma + 2\delta \tag{13}$$

$$\leq (n-1)\gamma + 2\delta \tag{14}$$

Where by $(2n\delta, \gamma)$-subgame stability of each G_{ij}, (s_i^μ, s_i^μ) is a γ-Nash of G_{ij}. By Proposition C.4, s^μ is γ-exploitable for i in each subgame since vulnerability coincides with exploitability for 2 player constant-sum games. □

C.5 Proof of Theorem 5.3

Theorem 5.3. *If G is δ-CSP in the neighborhood of $S(\mathcal{A}^N)$ and $\exists \check{G} \in \mathrm{CSP}_\delta(G, S(\mathcal{A}^N))$ that is γ-subgame stable in the the neighborhood of $S(\mathcal{A}_i)$ $\forall i \in N$, then for any $s \in S(\mathcal{A}_i)$*

$$\mathrm{Vul}_i\left(s, S(\mathcal{A}^N)\right) \leq |E_i|\gamma + 2\delta \leq (n-1)\gamma + 2\delta.$$

Proof.

$$\mathrm{Vul}_i\left(s, S(\mathcal{A})\right) \doteq u_i(s) - \min_{s'_{-i}\in S_{-i}(\mathcal{A}^N)} u_i(s, s'_{-i})$$

since G is δ-CSP in the neighborhood of $S(\mathcal{A}^N)$

$$\leq \check{u}_i(s) - \min_{s'_{-i}\in S_{-i}(\mathcal{A}^N)} \check{u}_i(s_i, s'_{-i}) + 2\delta$$

Since $\check{G}$ is a polymatrix game:

$$\begin{aligned}
&\check{u}_i(s) - \min_{s'_{-i}\in S_{-i}(\mathcal{A}^N)} \check{u}_i(s_i, s'_{-i}) + 2\delta \\
&= \sum_{(i,j)\in E_i} \check{u}_{ij}(s_i, s_j) - \min_{s'_{-i}\in S_{-i}(\mathcal{A}^N)} \sum_{(i,j)\in E_i} \check{u}_{ij}(s_i, s_j) + 2\delta \\
&= \left(\sum_{(i,j)\in E_i} \check{u}_{ij}(s_i, s_j) - \min_{s'_j\in S_j(\mathcal{A}_j)} \check{u}_{ij}(s_i, s'_j) \right) + 2\delta
\end{aligned}$$

Where, as in Theorem 4.3 and Theorem 4.6, the last line uses the fact that $\check{G}$ is polymatrix, G_{ij} is constant-sum and $-i$ minimize $i's$ utility and can do so by without coordinating.

Since $\check{G}$ is γ-subgame stable in the neighborhood of $S(\mathcal{A}_i)$ and $s \in S(\mathcal{A}_i)$, then means (s_i, s_j) is a γ-Nash for each subgame $\check{G}_{ij}$, so has bounded vulnerability within that subgame.

$$\begin{aligned}
&\leq \left(\sum_{(i,j)\in E_i} \gamma \right) + 2\delta \\
&\leq |E_i|\gamma + 2\delta \\
&\leq (n-1)\gamma + 2\delta
\end{aligned}$$

□

D Normal-Form Algorithms

D.1 Finding Constant-Sum Polymatrix Decomposition

LP 2 Below we give a linear program that takes as input a normal-form game $G = (N, \mathrm{P}, u)$ and (1) computes the minimum δ such that G is a δ-CSP game and (2) constant-sum polymatrix game $\check{G} = (N, E, \mathrm{P}, \check{u})$ such that $|u_i(\rho) - \check{u}_i(\rho)| \leq \delta$ for all $\rho \in \mathrm{P}$. The decision variables are the values of $\check{u}_{ij}(\rho)$ for all $i \neq j \in N, \rho \in \mathrm{P}$, δ and constants for each subgame c_{ij} for all $i \neq j$.

$$\min_{\check{u}_{ij}(\cdot), \delta, c_{ij}} \quad \delta \tag{15}$$

$$u_i(\rho) - \sum_{j \in -i} \check{u}_{ij}(\rho_i, \rho_j) \leq \delta \quad \forall i \in N, \rho \in \mathrm{P} \tag{16}$$

$$u_i(\rho) - \sum_{j \in -i} \check{u}_{ij}(\rho_i, \rho_j) \geq -\delta \quad \forall i \in N, \rho \in \mathrm{P} \tag{17}$$

$$\check{u}_{ij}(\rho_i, \rho_j) + \check{u}_{ji}(\rho_i, \rho_j) = c_{ij} \quad \forall i \neq j \in N, (\rho_i, \rho_j) \in \mathrm{P}_{ij}, \tag{18}$$

D.2 Computing γ for $(0, \gamma)$-Subgame Stability

Given a constant-sum polymatrix game, we can compute the minimum γ such that G is $(0, \gamma)$-subgame stable using Algorithm 1.

Algorithm 1 Compute γ

Input: $G = (N, E, A, u)$, a polymatrix game
$\gamma \leftarrow -\infty$
for $i \in N$ **do**
  for $j \neq i \in N$ **do**
    for $\bar{\rho}_i \in \mathrm{P}_i$ **do**
      $\gamma' \leftarrow \text{LP3}(i, j, \bar{\rho}_i)$
      $\gamma \leftarrow \max(\gamma, \gamma')$
    end for
  end for
end for

Where LP3$(i, j, \bar{\rho}_i)$ is a linear program that computes a Nash equilibrium of a CSP game G that maximizes the incentive to deviate to some candidate strategy $\bar{\rho}_i$ in the subgame between i and j. This linear program is given as follows. Let $a_i(\rho_i, \mu)$ be the advantage of deviating to ρ_i from a joint distribution over pure strategies:

$$a_i(\rho_i, \mu) \doteq \sum_{(i,j) \in E_i} \underbrace{u_{i,j}(\rho_i, s_j^\mu)}_{(a)} - \underbrace{\mathbb{E}_{\rho \sim \mu} \left[\sum_{(i,j) \in E_i} u_{i,j}(\rho_i, \rho_j) \right]}_{(b)} \tag{19}$$

Note that (a) is a linear function of μ, since s_j^μ is a marginal strategy. (b) is also a linear function of μ, and so $a_i(\rho_i, \mu)$ is a linear function of μ. Likewise, let

$$a_{i,j}(\rho_i, \mu) \doteq u_{i,j}(\rho_i, s_j^\mu) - \mathbb{E}_{\rho \sim \mu} \left[u_{i,j}(\rho_i, \rho_j) \right] \tag{20}$$

be the advantage of ρ_i restricted to the subgame between i and j. LP3$(i, j, \bar{\rho}_i)$ is given below. The decision variables are the weights of μ for each $\rho \in \mathrm{P}$ and γ

LP 3

$$\max_{\mu(\cdot),\gamma} \quad \gamma \tag{21}$$

$$\text{s.t.} \quad a_i(\rho_i, \mu) \leq 0 \quad \forall i \in N, \rho_i \in \mathrm{P}_i \tag{22}$$

$$a_{i,j}(\rho_i, \mu) \geq \gamma \tag{23}$$

$$\sum_{\rho \in \mathrm{P}} \mu(\rho) = 1 \tag{24}$$

$$\mu(\rho) \in [0, 1] \quad \forall \rho \in \mathrm{P} \tag{25}$$

We can get away with computing a CCE rather than targeting Nash equilibria is because the marginals of any CCE are Nash equilibria in constant-sum polymatrix games (Cai et al., 2016).

E Extensive-Form Games

Here we detail the algorithms and methods we used for computing $\check{G}$ for Leduc Poker. Before giving the full algorithm, we give some background on extensive-form games (EFG).

E.1 Background

We use the imperfect information extensive form game as a model for strategic multi-agent decision situations. An imperfect information extensive form game is a 10-tuple $(N, \mathcal{A}, H, Z, A, P, u, \mathcal{I}, c, \pi_c)$ where N is a set of players, $\mathcal{A}$ is a set of actions, H is a set of sequences of actions, the possible histories, $Z \subseteq H$ is a set of terminal histories, $A : H \to \mathcal{A}$ is a function that maps a history to available actions, $P : H \to N$ is the player function, which assigns a player to choose an action at each non-terminal history, u = $\{u_i\}_{i \in N}$ is a set of utility functions where $u_i : Z \to \mathbb{R}$ is the utility function for player i, $\mathcal{I} = \{\mathcal{I}_i\}_{i \in N}$ where $\mathcal{I}_i$ is a partition of the set $\{h \in H : P(h) = i\}$ such that if $h, h' \in I \in \mathcal{I}_i$ then $A(h) = A(h')$. We call an element $I \in \mathcal{I}_i$ an information set. The chance player c, who has a function $\pi_c(h, a)$ $\forall h : P(h) = c$ which returns the probability of random nature events $a \in \mathcal{A}$. Let $N_c = N \cup \{c\}$.

For some history h, the jth action in h is written h_j. A sub-history of h from the jth to kth actions is denoted $h_{j:k}$ and we use $h_{:k}$ as a short-hand for $h_{0:k}$. If a history h' is a prefix of history h, we write $h' \sqsubseteq h'$ and if h' is a proper prefix of h, we write $h' \sqsubset h'$.

We assume extensive-Form games have perfect recall.

Strategies We use $\rho_i : \mathcal{I}_i \to 2^{\mathcal{A}}$ to denote a *pure strategy* of player i, and the set of all pure strategies as P_i. $s_i \in \Delta(\mathrm{P}_i) = S_i$ is a *mixed strategy*, where $\Delta(X)$ denotes the set of probability distributions over a domain X. We use $\pi_i \in \Pi_i$ to denote a *behavior strategy* of player i, which, given $I \in \mathcal{I}_i$ returns $\Delta(A(I))$, a probability distribution over actions available at I. ρ, s and π are then used to denote pure, mixed and behavior strategy profiles, respectively. Note that P is a subset of both Π and S.

Given a behavior strategy profile, let

$$p_i^{\pi_i}(z) \doteq \prod_{z_{:k} \sqsubseteq z : P(z_{:k}) = i} \pi_i(z_{k+1}; z_{:k})$$

be the contribution of π_i to reaching z. Let $p^{\pi}(z) \doteq \prod_{i \in N_c} p_i^{\pi_i}(z)$ be the reach probabilities over $z \in Z$ induced by π.

We over load the utility function to accept behavior strategies:

$$u_i(\pi) \doteq \mathbb{E}_{z \sim (\pi)}\left[u_i(z)\right] = \sum_{z \in Z} p^{\pi}(z) u_i(z) = \sum_{z \in Z} \left(\prod_{i \in N_c} p_i^{\pi_i}(z) \right) u_i(z)$$

E.2 Poly-EFGs

We can straight-forwardly generalize our definitions to extensive-form games.

Definition E.1 (Poly-EFG)**.** A poly-EFG $(N, E, \mathcal{G})$ is defined by a graph with nodes N, one for each player, a set of edges E and a set of games $\mathcal{G} = \{G_{ij} \mid \forall (i,j) \in E\}$ where $G_{ij} \in \mathcal{G}$ is a two player EFG between i and j. We require that each player plays the same strategy in each game they take part in.

We assume that each G_{ij} has perfect recall.

Let $\check{u}_{ij}$ be the utility function of i against j

$$\check{u}_{ij}(\pi_i, \pi_j) = \mathbb{E}_{z \sim (\pi_i, \pi_j, \pi_c)} \left[\check{u}_{ij}(z) \right]$$

Then,

$$\check{u}_i(\pi) = \sum_{(i,j) \in E_i} \check{u}_{ij}(\pi_i, \pi_j)$$

Definition E.2 (Constant-sum (polymatrix))**.** We say a poly-EFG G is constant-sum if for some constant c we have that $\forall G_{ij} \in \mathcal{G}, z \in Z_{ij}$, $\check{u}_{ij}(z) + \check{u}_{ji}(z) = c$ where Z_{ij} is the set of terminal histories of G_{ij}.

Definitions of subgame stability and δ-CSP also generalize.

Definition E.3 (Subgame stable (profile))**.** Let G be a poly-EFG. We say a strategy profile π is γ-subgame stable if $\forall (i,j) \in E$, we have (π_i, π_j) is a γ-Nash of G_{ij}.

Definition E.4 (Subgame stable (game))**.** Let G be a poly-EFG e. We say G is (ϵ, γ)-subgame stable if for *any* ϵ-Nash equilibrium π of G, π is γ-subgame stable.

Definition E.5 (δ-constant sum polymatrix)**.** An EFG G is δ-constant sum polymatrix (δ-CSP) if there exists a constant sum poly-EFG $\check{G}$ with global utility functions $\check{u}$ such that $\forall i \in N, \pi \in \Pi$, $|u_i(\pi) - \check{u}_i(\pi)| \leq \delta$. We denote the set of such CSP games as $\mathrm{CSP}_\delta(G)$. We assume that $\forall (i,j), G_{ij}$ has the same strategy space as G for i and j.

Definition E.6. Given a poly-EFG $G = (N, E, \mathcal{G})$, the *induced normal-form polymatrix game* is a polymatrix game $G' = (N, E, \mathrm{P}, u')$ such that P_i is the same as i's set of pure strategies in all G_{ij} and $u'_{ij}(\rho_i, \rho_j) = u_{ij}(\rho_i, \rho_j)$.

In games of perfect recall every behavior strategy π_i has an equivalent mixed strategy s_i^π (Kuhn, 1953). This means for some EFG G, we can use the poly-EFG representation instead of turning G into a normal-form game then using a normal-form polymatrix game.

Corollary E.7. *For an extensive-form game G, if G is δ-constant sum polymatrix in the sense of Definition E.5 then the induced normal form of G, G' is δ-constant sum polymatrix in the sense of Definition 4.4 and if π is γ subgame stable for a poly-EFG $\check{G}$, then s^π is γ subgame stable in the induced normal-form polymatrix game of $\check{G}$.*

This is an immediate corollary of Kuhn's Theorem (Kuhn, 1953) and the assumption that each $G_{ij} \in \mathcal{G}$ has perfect recall.

Definition E.8. We say an EFG game G is δ-CSP in the neighborhood of $\Pi' \subseteq \Pi$ if there exists a constant-sum poly-EFG $\check{G}$ such that $\forall \pi \in \Pi'$ we have $|u_i(\pi) - \check{u}_i(\pi)| \leq \delta$. We denote the set of such CSP games as $\mathrm{CSP}_\delta(G, \Pi')$.

Definition E.9. We say a constant-sum poly-EFG is γ-subgame stable in the neighborhood of Π' if $\forall \pi \in \Pi', \forall (i,j) \in E$ we have that (π_i, π_j) is a γ-Nash of G_{ij}.

E.3 Finding a Poly-EFG for Leduc Poker

Here we show how to find a poly-EFG for Leduc Poker. The structure of each $G_{ij} \in \mathcal{G}$ needs to be such that i and j have the same strategy space as the original game. To do so, we define G_{ij} to have the same structure as Leduc poker, except we replace all information sets belonging to to players $k \in N \setminus \{i, j\}$ with chance nodes, where chance plays uniformly at random. Thus, we only need to compute $\check{u}_{ij}(z)$ $\forall z \in Z$ for each subgame.

Algorithm 2 Compute $\check{G}$

Input: G, an extensive-form game; Π', a set of behavior strategy profiles.
$\check{u}_{ij}(z) \leftarrow 0 \quad \forall i \neq j \in N, z \in Z$
$\mathrm{BR}^0_{ij}(\pi_j) \leftarrow \{\} \quad i \neq j \in N \times N, \pi_j \in \Pi'_j$
converged $\leftarrow$ **false**
$t \leftarrow 0$
while not converged **do**
  $t \leftarrow t + 1$
  {Compute best-responses to each subgame.}
  for $i \neq j \in N \times N$ **do**
    for $\pi_j \in \Pi'_j$ **do**
      compute $\pi^t_{ij} \in \arg\max_{\pi'_i \in \Pi_i} \check{u}^{t-1}_{ij}(\pi'_i, \pi_j)$
      $\mathrm{BR}^t_{ij}(\pi_j) = \mathrm{BR}^{t-1}_{ij}(\pi_j) \cup \{\pi^t_{ij}\}$
    end for
  end for
  $\check{u}^t \leftarrow$ UpdateUtility()
  {Next, check convergence.}
  converged $\leftarrow$ **true**
  for $i \neq j \in N \times N$ **do**
    for $\pi_j \in \Pi'_j$ **do**
      if $\pi^t_{ij} \neq \pi^{t-1}_{ij}$ for $\pi^t_{ij}, \pi^{t-1}_{ij} \in \mathrm{BR}^t_{ij}(\pi_j)$ **then**
        converged $\leftarrow$ **false**
      end if
    end for
  end for
end while
{Last, output δ.}
$\Pi^\times \leftarrow \bigtimes_{i \in N} \Pi'_i$
$\delta \leftarrow \max_{\pi \in \Pi^\times} \left| u_i(\pi) - \sum_{j \in -i} u^t_{ij}(\pi_i, \pi_j) \right|$
return $\check{u}^t(\cdot), \gamma^t, \delta$

E.4 Algorithm For Computing $\check{G}$ in Large games

Here we give the full algorithm that we described in Section 6. We describe it in more general terms: instead of $S(\mathrm{CFR})$, we write Π' and $\Pi^\times$ as $S(\mathrm{CFR}^N)$ since this algorithm need not necessariliy used CFR-computed strategies.

We use sequence-form linear programming to compute $\arg\max_{\pi'_i \in \Pi_i} \check{u}^{t-1}_{ij}(\pi'_i, \pi_j)$.

UpdateUtility() The decision variables are γ^t, δ^t, $\check{u}^t_{ij}(z)\ \forall i \neq j \in N \times N, z \in Z_{ij}$ and $c^t_{ij}\ \forall i \neq j \in N \times N$.

$$
\begin{aligned}
\min_{\gamma^t, \delta^t, (\check{u}^t_{ij}(\cdot))_{\forall i,j}, (\tilde{u}^t_{ij}(\cdot))_{\forall i,j}} \quad & (n-1)\gamma^t + 2\delta^t \\
\text{s.t.} \quad & \check{u}^t_{ij}(\pi'_i, \pi_j) - u^t_{ij}(\pi_i, \pi_j) \leq \gamma^t \quad \forall i \neq j \in N \times N, \pi'_i \in \mathrm{BR}^t_{ij}(\pi_j), \pi \in \Pi' \\
& \sum_{j \in -i} \check{u}^t_{ij}(\pi_i, \pi_j) - u_i(\pi) \leq \delta^t \quad \forall i \in N, \pi \in \Pi' \\
& u_i(\pi) - \sum_{j \in -i} \check{u}^t_{ij}(\pi_i, \pi_j) \leq \delta^t \quad \forall i \in N, \pi \in \Pi' \\
& \check{u}^t_{ij}(z) + u^t_{ji}(z) = c^t_{ij} \quad \forall i \neq j \in N \times N, z \in Z_{ij} \\
& \delta^t, \gamma^t \geq 0
\end{aligned}
$$

E.5 Experimental Details

We used the OpenSpiel library to compute our CFR strategies. We used vanilla CFR (Zinkevich et al., 2008) without any additional modifications (e.g. linear averaging or alternating updates) with the exception of allowing for random initializations. It took roughtly 100 hours to train each of the 30 instances of CFR on leduc poker for 10,000 iterations. Running Algorithm 2 took approximately 17,000s on a server with 100gb of memory. We used Gurobi as an LP solver.

E.6 Nash Equilibrium in Leduc Poker

Figure 5 shows the values of $\epsilon = \max_{s_i'} u_i(s_i', s_{-i}) - u_i(s)$ for each of the CFR strategies s computed by CFR in Leduc Poker. Each column is for one of the players and each point is one of the random seeds. We see the maximum value of ϵ after 10,000 iterations is 0.017.

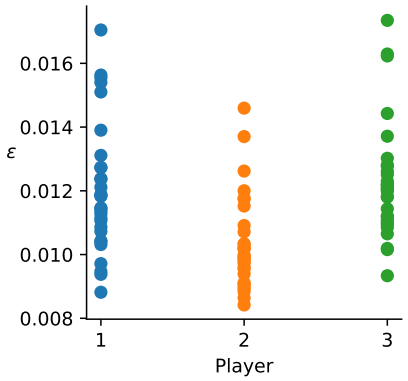


Figure 5: ϵ-Nash in Leduc Poker