

# Variance Reduction in Monte Carlo Counterfactual Regret Minimization (VR-MCCFR) for Extensive Form Games Using Baselines

Martin Schmid,<sup>1</sup> Neil Burch,<sup>1</sup> Marc Lanctot,<sup>1</sup> Matej Moravčík,<sup>1</sup> Rudolf Kadlec,<sup>1</sup> Michael Bowling<sup>1,2</sup>

DeepMind<sup>1</sup>

University of Alberta<sup>2</sup>

{mschmid,burchn,lanctot,moravcik,rudolfkadlec,bowling}@google.com

## Abstract

Learning strategies for imperfect information games from samples of interaction is a challenging problem. A common method for this setting, Monte Carlo Counterfactual Regret Minimization (MCCFR), can have slow long-term convergence rates due to high variance. In this paper, we introduce a variance reduction technique (VR-MCCFR) that applies to any sampling variant of MCCFR. Using this technique, per-iteration estimated values and updates are reformulated as a function of sampled values and state-action baselines, similar to their use in policy gradient reinforcement learning. The new formulation allows estimates to be bootstrapped from other estimates within the same episode, propagating the benefits of baselines along the sampled trajectory; the estimates remain unbiased even when bootstrapping from other estimates. Finally, we show that given a perfect baseline, the variance of the value estimates can be reduced to zero. Experimental evaluation shows that VR-MCCFR brings an order of magnitude speedup, while the empirical variance decreases by three orders of magnitude. The decreased variance allows for the first time CFR+ to be used with sampling, increasing the speedup to two orders of magnitude.

## Introduction

Policy gradient algorithms have shown remarkable success in single-agent reinforcement learning (RL) (Mnih et al. 2016; Schulman et al. 2017). While there has been evidence of empirical success in multiagent problems (Foerster et al. 2017; Bansal et al. 2018), the assumptions made by RL methods generally do not hold in multiagent partially-observable environments. Hence, they are not guaranteed to find an optimal policy, even with tabular representations in two-player zero-sum (competitive) games (Littman 1994). As a result, policy iteration algorithms based on computational game theory and regret minimization have been the preferred formalism in this setting. Counterfactual regret minimization (Zinkevich et al. 2008) has been a core component of this progress in Poker AI, leading to solving Heads-Up Limit Texas Hold'em (Bowling et al. 2015) and defeating professional poker players in No-Limit (Moravčík et al. 2017; Brown and Sandholm 2017).

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

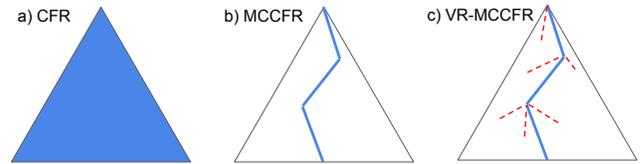


Figure 1: High-level overview of Variance Reduction MC-CFR (VR-MCCFR) and related methods. a) CFR traverses the entire tree on every iteration. b) MCCFR samples trajectories and computes the values only for the sampled actions, while the off-trajectory actions are treated as zero-valued. While MCCFR uses importance sampling weight to ensure the values are unbiased, the sampling introduces high variance. c) VR-MCCFR follows the same sampling framework as MCCFR, but uses baseline values for both sampled actions (in blue) as well as the off-trajectory actions (in red). These baselines use control variates and send up bootstrapped estimates to decrease the per-iteration variance thus speeding up the convergence.

The two fields of RL and computational game theory have largely grown independently. However, there has been recent work that relates approaches within these two communities. Fictitious self-play uses RL to compute approximate best responses and supervised learning to combine responses (Heinrich, Lanctot, and Silver 2015). This idea is extended to a unified training framework that can produce more general policies by regularizing over generated response oracles (Lanctot et al. 2017). RL-style regressors were first used to compress regrets in game theoretic algorithms (Waugh et al. 2015). DeepStack introduced deep neural networks as generalized value-function approximators for online planning in imperfect information games (Moravčík et al. 2017). These value functions operate on a belief-space over all possible states consistent with the players' observations.

This paper similarly unites concepts from both fields, proposing an unbiased variance reduction technique for Monte Carlo counterfactual regret minimization using an analog of state-action baselines from actor-critic RL methods. While policy gradient methods typically involve Monte Carlo estimates, the analog in imperfect information settings is Monte Carlo Counterfactual Regret Minimization (MC-

CFR) (Lanctot et al. 2009). Policy gradient estimates based on a single sample of an episode suffer significantly from variance. A common technique to decrease the variance is a state or state-action dependent baseline value that is subtracted from the observed return. These methods can drastically improve the convergence speed. However, no such methods are known for MCCFR.

MCCFR is a sample based algorithm in imperfect information settings, which approximates counterfactual regret minimization (CFR) by estimating regret quantities necessary for updating the policy. While MCCFR can offer faster short-term convergence than original CFR in large games, it suffers from high variance which leads to slower long-term convergence.

CFR+ provides significantly faster empirical performance and made solving Heads-Up Limit Texas Hold'em possible (Bowling et al. 2015). Unfortunately, CFR+ has so far did not outperform CFR in Monte Carlo settings (Burch 2017) (also see appendix for an experiment).

In this work, we reformulate the value estimates using a control variate and a state-action baseline. The new formulation includes any approximation of the counterfactual values, which allows for a range of different ways to insert domain-specific knowledge (if available) but also to design values that are learned online.

Our experiments show two orders of magnitude improvement over MCCFR. For the common testbed imperfect information game – Leduc Poker – VR-MCCFR with a state-action baseline needs 250 times fewer iterations than MC-CFR to reach the same solution quality. In contrast to RL algorithms in perfect information settings, where state-action baselines bring little to no improvement over state baselines (Tucker et al. 2018), state-action baselines lead to significant improvement over state baselines in multiagent partially-observable settings. We suspect this is due to variance from the environment and different dynamics of the policies during the computation.

## Related Work

There are standard variance reduction techniques for Monte Carlo sampling methods (Owen 2013) and the use of control variates in these settings has a long history (Boyle 1977). Reducing variance is particularly important when estimating gradients from sample trajectories. Consequentially, the use of a control variates using baseline has become standard practice in policy gradient methods (Williams 1992; Sutton and Barto 2017). In RL, action-dependent baselines have recently shown promise (Wu et al. 2018; Liu et al. 2018) but the degree to which variance is indeed reduced remains unclear (Tucker et al. 2018). We show that in our setting of MCCFR in imperfect information multiplayer games, action-dependent baselines necessarily influence the variance of the estimates, and we confirm the reduction empirically. This is important because lower-variance estimates lead to better regret bounds (Gibson et al. 2012).

There have been a few uses of variance reduction techniques in multiplayer games, within Monte Carlo tree search (MCTS). In MCTS, control variates have used to augment the reward along a trajectory using a property of the state

before and after a transition (Veness, Lanctot, and Bowling 2011) and to augment the outcome of a rollout from its length or some pre-determined quality of the states visited (Pepels et al. 2014).

Our baseline-improved estimates are similar to the ones used in AIVAT (Burch et al. 2018). AIVAT defines estimates of expected values using heuristic values of states as baselines in practice. Unlike this work, AIVAT was only used for evaluation of strategies.

To the best of our knowledge, there has been two applications of variance reduction in Monte Carlo CFR: by manipulating the chance node distribution (Lanctot 2013, Section 7.5) and by sampling (“probing”) more trajectories for more estimates of the underlying values (Gibson et al. 2012). The variance reduction (and resulting drop in convergence rate) is modest in both cases, whereas we show more than a two order of magnitude speed-up in convergence using our method.

## Background

We start with the formal background necessary to understand our method. For details, see (Shoham and Leyton-Brown 2009; Sutton and Barto 2017).

A two player **extensive-form game** is tuple  $(\mathcal{N}, \mathcal{A}, \mathcal{H}, \mathcal{Z}, \tau, u, \mathcal{I})$ .

$\mathcal{N} = \{1, 2, c\}$  is a finite set of players, where  $c$  is a special player called chance.  $\mathcal{A}$  is a finite set of actions. Players take turns choosing actions, which are composed into sequences called *histories*; the set of all valid histories is  $\mathcal{H}$ , and the set of all terminal histories (games) is  $\mathcal{Z} \subseteq \mathcal{H}$ . We use the notation  $h' \sqsubseteq h$  to mean that  $h'$  is a prefix sequence or equal to  $h$ . Given a nonterminal history  $h$ , the player function  $\tau : \mathcal{H} \setminus \mathcal{Z} \rightarrow \mathcal{N}$  determines who acts at  $h$ . The utility function  $u : (\mathcal{N} \setminus \{c\}) \times \mathcal{Z} \rightarrow [u_{\min}, u_{\max}] \subset \mathbb{R}$  assigns a payoff to each player for each terminal history  $z \in \mathcal{Z}$ .

The notion of a state in imperfect information games requires groupings of histories:  $\mathcal{I}_i$  for some player  $i \in \mathcal{N}$  is a partition of  $\{h \in \mathcal{H} \mid \tau(h) = i\}$  into parts  $I \in \mathcal{I}_i$  such that  $h, h' \in I$  if player  $i$  cannot distinguish  $h$  from  $h'$  given the information known to player  $i$  at the two histories. We call these **information sets**. For example, in Texas Hold'em poker, for all  $I \in \mathcal{I}_i$ , the (public) actions are the same for all  $h, h' \in I$ , and  $h$  only differs from  $h'$  in cards dealt to the opponents (actions chosen by chance). For convenience, we refer to  $I(h)$  as the information state that contains  $h$ .

At any  $I$ , there is a subset of legal actions  $A(I) \subseteq \mathcal{A}$ . To choose actions, each player  $i$  uses a **strategy**  $\sigma_i : I \rightarrow \Delta(A(I))$ , where  $\Delta(X)$  refers to the set of probability distributions over  $X$ . We use the shorthand  $\sigma(h, a)$  to refer to  $\sigma(I(h), a)$ . Given some history  $h$ , we define the **reach probability**  $\pi^\sigma(h) = \prod_{h' \sqsubset h} \sigma_{\tau(h')}(I(h'), a)$  to be the product of all action probabilities leading up to  $h$ . This reach probability contains all players’ actions, but can be separated  $\pi^\sigma(h) = \pi_i^\sigma(h) \pi_{-i}^\sigma(h)$  into player  $i$ ’s actions’ contribution and the contribution of the opponents’ of player  $i$  (including chance).

Finally, it is often useful to consider the **augmented information sets** (Burch, Johanson, and Bowling 2014).

While an information set  $I$  groups histories  $h$  that player  $i = \tau(h)$  cannot distinguish, an augmented information set groups histories that player  $i$  can not distinguish, including those where  $\tau(h) \neq i$ . For a history  $h$ , we denote an augmented information set of player  $i$  as  $I_i(h)$ . Note that the if  $\tau(h) = i$  then  $I_i(h) = I(h)$  and  $I(h) = I_{\tau(h)}(h)$ .

### Counterfactual Regret Minimization

Counterfactual Regret (CFR) Minimization is an iterative algorithm that produces a sequence of strategies  $\sigma^0, \sigma^1, \dots, \sigma^T$ , whose average strategy  $\bar{\sigma}^T$  converges to an approximate Nash equilibrium as  $T \rightarrow \infty$  in two-player zero-sum games (Zinkevich et al. 2008). Specifically, on iteration  $t$ , for each  $I$ , it computes **counterfactual values**. Define  $\mathcal{Z}_I = \{(h, z) \in \mathcal{H} \times \mathcal{Z} \mid h \in I, h \sqsubseteq z\}$ , and  $u_i^{\sigma^t}(h, z) = \pi^{\sigma^t}(h, z)u_i(z)$ . We will also sometimes use the short form  $u_i^\sigma(h) = \sum_{z \in \mathcal{Z}, h \sqsubseteq z} u_i^\sigma(h, z)$ . A counterfactual value is:

$$v_i(\sigma^t, I) = \sum_{(h, z) \in \mathcal{Z}_I} \pi_{-i}^{\sigma^t}(h) u_i^{\sigma^t}(h, z). \quad (1)$$

We also define an action-dependent counterfactual value,

$$v_i(\sigma, I, a) = \sum_{(h, z) \in \mathcal{Z}_I} \pi_{-i}^\sigma(ha) u^\sigma(ha, z), \quad (2)$$

where  $ha$  is the sequence  $h$  followed by the action  $a$ . The values are analogous to the difference in  $Q$ -values and  $V$ -values in RL, and indeed we have  $v_i(\sigma, I) = \sum_a \sigma(I, a) v_i(\sigma, I, a)$ . CFR then computes a **counterfactual regret** for *not taking*  $a$  at  $I$ :

$$r^t(I, a) = v_i(\sigma^t, I, a) - v_i(\sigma^t, I), \quad (3)$$

This regret is then accumulated  $R^T(I, a) = \sum_{t=1}^T r^t(I, a)$ , which is used to update the strategies using **regret-matching** (Hart and Mas-Colell 2000):

$$\sigma^{T+1}(I, a) = \frac{(R^T(I, a))^+}{\sum_{a \in A(I)} (R^T(I, a))^+}, \quad (4)$$

where  $(x)^+ = \max(x, 0)$ , or to the uniform strategy if  $\sum_a (R^T(I, a))^+ = 0$ . CFR+ works by thresholding the quantity at each round (Tammelin et al. 2015): define  $Q^0(I, a) = 0$  and  $Q^T(I, a) = (Q^{T-1} + r^T(I, a))^+$ ; CFR+ updates the policy by replacing  $R^T$  by  $Q^T$  in equation 4. In addition, it always alternates the regret updates of the players (whereas some variants of CFR update both players), and the average strategy places more (linearly increasing) weight on more recent iterations.

If for player  $i$  we denote  $u(\sigma) = u_i(\sigma_i, \sigma_{-i})$ , and run CFR for  $T$  iterations, then we can define the **overall regret** of the strategies produced as:

$$R_i^T = \max_{\sigma'_i} \sum_{t=1}^T (v_i(\sigma'_i, \sigma_{-i}^t) - v_i(\sigma^t)).$$

CFR ensures that  $R_i^T/T \rightarrow 0$  as  $T \rightarrow \infty$ . When two players minimize regret, the folk theorem then guarantees a bound on the distance to a Nash equilibrium as a function of  $R_i^T/T$ .

To compute  $v_i$  precisely, each iteration requires traversing over subtrees under each  $a \in A(I)$  at each  $I$ . Next, we describe variants that allow sampling parts of the trees and using estimates of these quantities.

### Monte Carlo CFR

Monte Carlo CFR (MCCFR) introduces sample estimates of the counterfactual values, by visiting and updating quantities over only part of the entire tree. MCCFR is a general family of algorithms: each instance defined by a specific sampling policy. For ease of exposition and to show the similarity to RL, we focus on **outcome sampling** (Lanctot et al. 2009); however, our baseline-enhanced estimates can be used in all MCCFR variants. A **sampling policy**  $\xi$  is defined in the same way as a strategy (a distribution over  $A(I)$  for all  $I$ ) with a restriction that  $\xi(h, a) > 0$  for all histories and actions. Given a terminal history sampled with probability  $q(z) = \pi^\xi(z)$ , a **sampled counterfactual value**  $\tilde{v}_i(\sigma, I|z)$

$$= \tilde{v}_i(\sigma, h|z) = \frac{\pi_{-i}^\sigma(h) u_i^\sigma(h, z)}{q(z)}, \text{ for } h \in I, h \sqsubseteq z, \quad (5)$$

and 0 for histories that were not played,  $h \not\sqsubseteq z$ . The estimate is unbiased:  $\mathbb{E}_{z \sim \xi}[\tilde{v}_i(\sigma, I|z)] = v_i(\sigma, I)$ , by (Lanctot et al. 2009, Lemma 1). As a result,  $\tilde{v}_i$  can be used in Equation 3 to accumulate estimated regrets  $\tilde{r}^t(I, a) = \tilde{v}_i(\sigma^t, I, a) - \tilde{v}_i(\sigma^t, I)$  instead. The regret bound requires an additional term  $\frac{1}{\min_{z \in \mathcal{Z}} q(z)}$ , which is exponential in the length of  $z$  and similar observations have been made in RL (Arjona-Medina et al. 2018). The main problem with the sampling variants is that they introduce variance that can have a significant effect on long-term convergence (Gibson et al. 2012).

### Control Variates

Suppose one is trying to estimate a statistic of a random variable,  $X$ , such as its mean, from samples  $\mathbf{X} = (X_1, X_2, \dots, X_n)$ . A crude Monte Carlo estimator is defined to be  $\hat{X}^{mc} = \frac{1}{n} \sum_{i=1}^n X_i$ . A **control variate** is a random variable  $Y$  with a known mean  $\mu_Y = \mathbb{E}[Y]$ , that is paired with the original variable, such that samples are instead of the form  $(\mathbf{X}, \mathbf{Y})$  (Owen 2013). A new random variable is then defined,  $Z_i = X_i + c(Y_i - \mu_Y)$ . An estimator  $\hat{Z}^{cv} = \frac{1}{n} \sum_{i=1}^n Z_i$ . Since  $\mathbb{E}[Z_i] = \mathbb{E}[X_i]$  for any value of  $c$ ,  $\hat{Z}^{cv}$  can be used in place of  $\hat{X}^{mc}$ . with variance  $\text{Var}[Z_i] = \text{Var}[X_i] + c^2 \text{Var}[Y_i] + 2c \text{Cov}[X_i, Y_i]$ . So when  $X$  and  $Y$  are positively correlated and  $c < 0$ , variance is reduced when  $\text{Cov}[X, Y] > \frac{c^2}{2} \text{Var}[Y]$ .

### Reinforcement Learning Mapping

There are several analogies to make between Monte Carlo CFR in imperfect information games and reinforcement learning. Since our technique builds on ideas that have been widely used in RL, we end the background by providing a small discussion of the links.

First, dynamics of an imperfect information game are similar to a partially-observable episodic MDP without any cycles. Policies and strategies are identically defined, but in imperfect information games a deterministic optimal (Nash)

strategy may not exist causing most of the RL methods to fail to converge. The search for a minmax-optimal strategy with several players is the main reason CFR is used instead of, for example, value iteration. However, both operate by defining values of states which are analogous (counterfactual values versus expected values) since they are both functions of the strategy/policy; therefore, can be viewed as a kind of policy iteration which computes the values and from which a policy is derived. However, the iterates  $\sigma^t$  are not guaranteed to converge, only the average strategy  $\bar{\sigma}^t$  does.

Monte Carlo CFR is an off-policy Monte Carlo analog. The value estimates are unbiased specifically because they are corrected by importance sampling. Most applications of MCCFR have operated with tabular representations, but this is mostly due to the differences in objectives. Function approximation methods have been proposed for CFR (Waugh et al. 2015) but the variance from pure Monte Carlo methods may prevent such techniques in MCCFR. The use of baselines has been widely successful in policy gradient methods, so reducing the variance could enable the practical use of function approximation in MCCFR.

It was recently shown that policy gradient and actor-critic algorithms implement a form of *on-policy* Monte Carlo CFR in zero-sum games, inspiring regret-based policy update rules (Srinivasan et al. 2018); we suspect that the variance reduction techniques proposed in this paper could apply to these model-free RL algorithms as well.

## Monte Carlo CFR with Baselines

We now introduce our technique: MCCFR with baselines. While the baselines are analogous to those from policy gradient methods (using counterfactual values), there are slight differences in their construction.

Our technique constructs value estimates using control variates. Note that MCCFR is using sampled estimates of counterfactual values  $\tilde{v}_i(\sigma, I)$  whose expected value is the counterfactual value  $v_i(\sigma, I)$ . First, we introduce an **estimated counterfactual** value  $\hat{v}_i(\sigma, I)$  to be any estimator of the counterfactual value (not necessarily  $\tilde{v}_i$  as defined above, but this is one possibility).

We now define an action-dependent **baseline**  $b_i(I, a)$  that, as in RL, serves as a basis for the sampled values. The intent is to define a baseline function to approximate or be correlated with  $\mathbb{E}[\hat{v}_i(\sigma, I, a)]$ . We also define a sampled baseline  $\hat{b}_i(I, a)$  as an estimator such that  $\mathbb{E}[\hat{b}_i(I, a)] = b_i(I, a)$ . From this, we construct a new baseline-enhanced estimate for the counterfactual values:

$$\hat{v}_i^b(\sigma, I, a) = \hat{v}_i(\sigma, I, a) - \hat{b}_i(\sigma, I, a) + b_i(\sigma, I, a) \quad (6)$$

First, note that  $\hat{b}_i$  is a control variate with  $c = -1$ . Therefore, it is important that  $\hat{b}_i$  be correlated with  $\hat{v}_i$ . The main idea of our technique is to replace  $\tilde{v}_i(\sigma, I, a)$  with  $\hat{v}_i^b(\sigma, I, a)$ . A key property is that by doing so, the expectation remains unchanged.

**Lemma 1.** *For any  $i \in \mathcal{N} - \{c\}$ ,  $\sigma_i, I \in \mathcal{I}, a \in A(I)$ , if  $\mathbb{E}[\hat{b}_i(I, a)] = b_i(I, a)$  and  $\mathbb{E}[\hat{v}_i(\sigma, I, a)] = v_i(\sigma, I, a)$ , then  $\mathbb{E}[\hat{v}_i^b(\sigma, I, a)] = v_i(\sigma, I, a)$ .*

The proof is in the appendix. As a result, any baseline whose expectation is known can be used and the baseline-enhanced estimates are consistent. However, not all baselines will decrease variance. For example, if  $\text{Cov}[\hat{v}_i, \hat{b}_i]$  is too low, then the  $\text{Var}[\hat{b}_i]$  term could dominate and actually increase the variance.

## Recursive Bootstrapping

Consider the individual computation (1) for all the information sets on the path to a sampled terminal history  $z$ . Given that the counterfactual values up the tree can be computed from the counterfactual values down the tree, it is natural to consider propagating the already baseline-enhanced counterfactual values (6) rather than the original noisy sampled values - thus propagating the benefits up the tree. The Lemma (2) then shows that by doing so, the updates remain unbiased. Our experimental section shows that such bootstrapping a crucial component for the proper performance of the method.

To properly formalize this bootstrapping computation, we must first recursively define the **expected value**:

$$\hat{u}_i(\sigma, h, a|z) = \begin{cases} \hat{u}_i(\sigma, ha|z)/\xi(h, a) & \text{if } ha \sqsubseteq z \\ 0 & \text{otherwise} \end{cases}, \quad (7)$$

and

$$\hat{u}_i(\sigma, h|z) = \begin{cases} u_i(h) & \text{if } h = z \\ \sum_a \sigma(h, a) \hat{u}_i(\sigma, h, a|z) & \text{if } h \sqsubset z \\ 0 & \text{otherwise} \end{cases}. \quad (8)$$

Next, we define a baseline-enhanced version of the expected value. Note that the baseline  $b_i(I, a)$  can be arbitrary, but we discuss a particular choice and update of the baseline in the later section. For every action, given a specific sampled trajectory  $z$ , then  $\hat{u}_i^b(\sigma, h, a|z) =$

$$\begin{cases} b_i(I_i(h), a) + \frac{\hat{u}_i^b(\sigma, ha|z) - b_i(I_i(h), a)}{\xi(h, a)} & \text{if } ha \sqsubseteq z \\ b_i(I_i(h), a) & \text{if } h \sqsubset z, ha \not\sqsubseteq z \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

and

$$\hat{u}_i^b(\sigma, h|z) = \begin{cases} u_i(h) & \text{if } h = z \\ \sum_a \sigma(h, a) \hat{u}_i^b(\sigma, h, a|z) & \text{if } h \sqsubset z \\ 0 & \text{otherwise} \end{cases}. \quad (10)$$

These are the values that are bootstrapped. We estimate counterfactual values needed for the regret updates using these values as:

$$\hat{v}_i^b(\sigma, I(h), a|z) = \hat{v}_i^b(\sigma, h, a|z) = \frac{\pi_i^\sigma(h)}{q(h)} \hat{u}_i^b(\sigma, h, a|z). \quad (11)$$

We can now formally state that the bootstrapping keeps the counterfactual values unbiased:

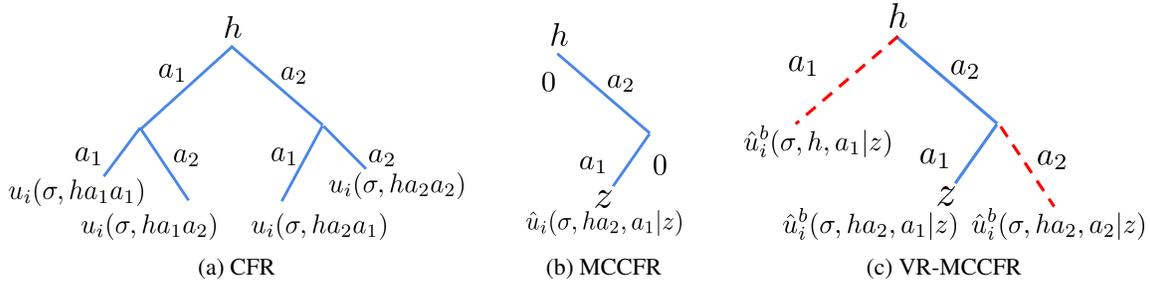


Figure 2: Values and updates for the discussed methods: (a) CFR updates the full tree and thus uses the exact values for all the actions, (b) MCCFR updates only a single path, and uses the sampled values for the sampled actions and zero values for the off-trajectory actions, (c) VR-MCCFR also updates only a single path, but uses the bootstrapped baseline-enhanced values for the sampled actions and baseline-enhanced values for the off-trajectory actions.

**Lemma 2.** Let  $\hat{v}_i^b$  be defined as in Equation 11. Then, for any  $i \in \mathcal{N} - \{c\}, \sigma_i, I \in \mathcal{I}, a \in A(I)$ , it holds that  $\mathbb{E}_z[\hat{v}_i^b(\sigma, I, a|z)] = v_i(\sigma, I, a)$ .

The proof is in the appendix. Since each estimate builds on other estimates, the benefit of the reduction in variance can be propagated up through the tree.

Another key result is that there exists a perfect baseline that leads to zero-variance estimates at the updated information sets.

**Lemma 3.** There exists a perfect baseline  $b^*$  and optimal unbiased estimator  $\hat{v}_i^*(\sigma, h, a)$  such that under a specific update scheme:  $\forall a_{h,z} \sim \xi, h \in I, h \sqsubseteq z [\hat{v}_i^*(\sigma, h, a|z)] = 0$ .

The proof and description of the update scheme are in the appendix. We will refer to  $b^*$  as the **oracle baseline**. Note that even when using the oracle baseline, the convergence rate of MCCFR is still not identical to CFR because each iteration applies regret updates to a portion of the tree, whereas CFR updates the entire tree.

Finally, using unbiased estimates to tabulate regrets  $\hat{r}(I, a)$  for each  $I$  and  $a$  leads to a probabilistic regret bound:

**Theorem 1.** (Gibson et al. 2012, Theorem 2) For some unbiased estimator of the counterfactual values  $\hat{v}_i$  and a bound on the difference in its value  $\hat{\Delta}_i = |\hat{v}_i(\sigma, I, a) - \hat{v}_i(\sigma, I, a')|$ , with probability  $1-p, \frac{R_i^T}{T}$

$$\leq \left( \hat{\Delta}_i + \frac{\sqrt{\max_{t,I,a} \text{Var}[r_i^t(I, a) - \hat{r}_i^t(I, a)]}}{\sqrt{p}} \right) \frac{|\mathcal{I}_i| |\mathcal{A}_i|}{\sqrt{T}}.$$

### Choice of Baselines

How does one choose a baseline, given that we want these to be good estimates of the individual counterfactual values? A common choice of the baseline in policy gradient algorithms is the mean value of the state, which is learned online (Mnih et al. 2016). Inspired by this, we choose a similar quantity: the average expected value  $\hat{v}_i(I_i, a)$ . That is, in addition to accumulating regret for each  $I$ , average expected values are also tracked.

While a direct average can be tracked, we found that an exponentially-decaying average that places heavier weight

on more recent samples to be more effective in practice. On the  $k^{\text{th}}$  visit to  $I$  at iteration  $t$ ,

$$\bar{u}_i^k(I_i, a) = \begin{cases} 0 & \text{if } k = 0, \text{ otherwise:} \\ (1 - \alpha) \bar{u}_i^{k-1}(I_i, a) + \alpha \hat{u}_i^b(\sigma^t, I_i, a) \end{cases}$$

We then define the baseline  $b_i(I_i, a) = \bar{u}_i(I_i, a)$ , and

$$\hat{b}_i(I_i, a|z) = \begin{cases} b_i(I_i, a)/\xi(I_i, a) & \text{if } ha \sqsubseteq z, h \in I_i \\ 0 & \text{otherwise.} \end{cases}$$

The baseline can therefore be thought as *local* to  $I_i$  since it depends only on quantities defined and tracked at  $I_i$ . Note that  $\mathbb{E}_{a \sim \xi(I_i)}[\hat{b}_i(I_i, a|z)] = b_i(I_i, a)$  as required.

### Summary of the Full Algorithm

We now summarize the technique developed above. One iteration of the algorithm consists of:

1. Repeat the steps below for each  $i \in \mathcal{N} - \{c\}$ .
2. Sample a trajectory  $z \sim \xi$ .
3. For each history  $h \sqsubseteq z$  in reverse order (longest first):
  - (a) If  $h$  is terminal, simply return  $u_i(h)$
  - (b) Obtain current strategy  $\sigma(I)$  from Eq. 4 using cumulative regrets  $R(I, a)$  where  $h \in I$ .
  - (c) Use the child value  $\hat{u}_i^b(\sigma, ha)$  to compute  $\hat{u}_i^b(\sigma, h)$  as in Eq. 9.
  - (d) If  $\tau(h) = i$  then for  $a \in A(I)$ , compute  $\hat{v}_i^b(\sigma, I, a) = \frac{\pi_{-i}(h)}{q(h)} \hat{u}_i^b(\sigma, ha)$  and accumulate regrets  $R(I, a) \leftarrow R(I, a) + \hat{v}_i^b(\sigma, I, a) - \hat{v}_i^b(\sigma, I)$ .
  - (e) Update  $\hat{u}_i(\sigma, I_i, a)$ .
  - (f) Finally, return  $\hat{u}_i^b(\sigma, h)$ .

Note that the original outcome sampling is an instance of this algorithm. Specifically, when  $b_i(I_i, a) = 0$ , then  $\hat{v}_i^b(\sigma, I, a) = \hat{v}_i(\sigma, I, a)$ . Step by step example of the computation is in the appendix.

### Experimental Results

We evaluate the performance of our method on **Leduc poker** (Southey et al. 2005), a commonly used benchmark poker game. Players have an unlimited number of chips,

and the deck has six cards, divided into two suits of three identically-ranked cards. There are two rounds of betting; after the first round a single public card is revealed from the deck. Each player antes 1 chip to play, receiving one private card. There are at most two bet or raise actions per round, with a fixed size of 2 chips in the first round, and 4 chips in the second round.

For the experiments, we use a vectorized form of CFR that applies regret updates to each information set consistent with the public information. The first vector variants were introduced in (Johanson et al. 2012), and have been used in DeepStack and Libratus (Moravčík et al. 2017; Brown and Sandholm 2017). See the appendix for more detail on the implementation. Baseline average values  $\hat{u}_i^b(I, a)$  used a decay factor of  $\alpha = 0.5$ . We used a uniform sampling in all our experiments,  $\xi(I, a) = \frac{1}{|A(I)|}$ .

We also consider the best case performance of our algorithm by using the oracle baseline. It uses baseline values of the true counterfactual values. We also experiment with and without CFR+, demonstrating that our technique allows the CFR+ to be for the first time efficiently used with sampling.

### Convergence

We compared MCCFR, MCCFR+, VR-MCCFR, VR-MCCFR+, and VR-MCCFR+ with the oracle baseline, see Fig. 3. The variance-reduced VR-MCCFR and VR-MCCFR+ variants converge significantly faster than plain MCCFR. Moreover, the speedup grows as the baseline improves during the computation. A similar trend is shown by both VR-MCCFR and VR-MCCFR+, see Fig. 4. MCCFR needs hundreds of millions of iterations to reach the same exploitability as VR-MCCFR+ achieves in one million iterations: a 250-times speedup. VR-MCCFR+ with the oracle baseline significantly outperforms VR-MCCFR+ at the start of the computation, but as time progresses and the learned baseline improves, the difference shrinks. After one million iterations, exploitability of VR-MCCFR+ with a learned baseline approaches the exploitability of VR-MCCFR+ with the oracle baseline. This oracle baseline result gives a bound on the gains we can get by constructing better learned baselines.

### Observed Variance

To verify that the observed speedup of the technique is due to variance reduction, we experimentally observed variance of counterfactual value estimates for MCCFR+ and MCCFR, see Fig. 5. We did that by sampling 1000 alternative trajectories for all visited information sets, with each trajectory sampling a different estimate of the counterfactual value. While the variance of value estimates in the plain algorithm seems to be more or less constant, the variance of VR-MCCFR and VR-MCCFR+ value estimates is lower, and continues to decrease as more iterations are run. This confirms that the combination of baseline and bootstrapping is reducing variance, which implies better performance given the connection between variance and MCCFR’s performance (Theorem 1).

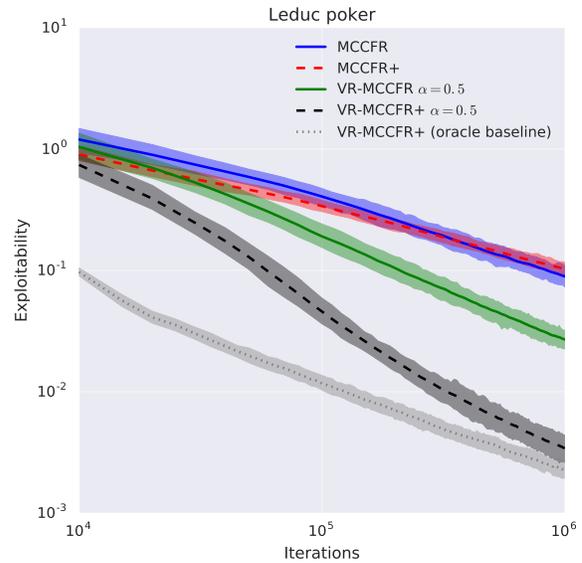


Figure 3: Convergence of exploitability for different MCCFR variants on logarithmic scale. VR-MCCFR converges substantially faster than plain MCCFR. VR-MCCFR+ bring roughly two orders of magnitude speedup. VR-MCCFR+ with oracle baseline (actual true values are used as baselines) is used as a bound for VR-MCCFR’s performance to show possible room for improvement. When run for  $10^6$  iterations VR-MCCFR+ approaches performance of the oracle version. The ribbons show 5th and 95th percentile over 100 runs.

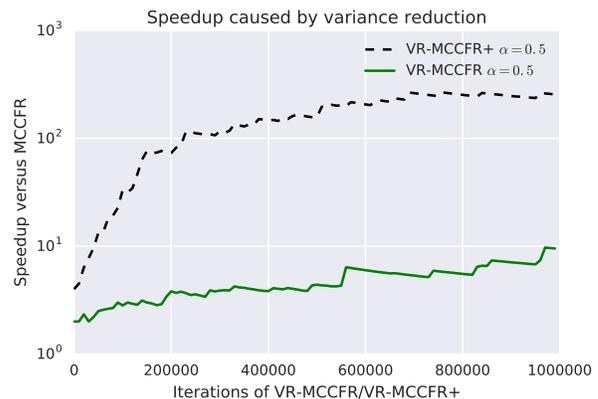


Figure 4: Speedup of VR-MCCFR and VR-MCCFR+ compared to plain MCCFR. Y-axis show how many times more iterations are required by MCCFR to reach the same exploitability as VR-MCCFR or VR-MCCFR+.

### Evaluation of Bootstrapping and Baseline Dependence on Actions

Recent work that evaluates action-dependent baselines in RL (Tucker et al. 2018), shows that there is often no real advan-

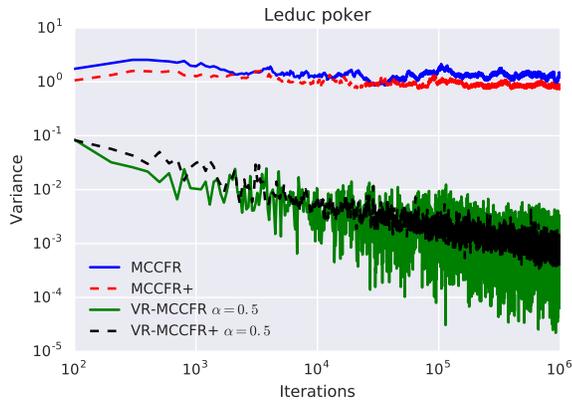


Figure 5: Variance of counterfactual values in VR-MCCFR and plain MCCFR with both regret matching and regret matching+. The curves were smoothed by computing moving average over a sliding window of 100 iterations.

tage compared to baselines that depend just on the state. It is also not common to bootstrap the value estimates in RL. Since VR-MCCFR uses both of these techniques it is natural to explore the contribution of each idea. We compared four VR-MCCFR+ variants: with or without bootstrapping and with baseline that is state or state-action dependant, see Fig. 6. The conclusion is that the improvement in the performance is very small unless we use both bootstrapping and an action-dependant baseline.

## Conclusions

We have presented a new technique for variance reduction for Monte Carlo counterfactual regret minimization. This technique has close connections to existing RL methods of state and state-action baselines. In contrast to RL environments, our experiments in imperfect information games suggest that state-action baselines are superior to state baselines. Using this technique, we show that empirical variance is indeed reduced, speeding up the convergence by an order of magnitude. The decreased variance allows for the first time CFR+ to be used with sampling, bringing the speedup to two orders of magnitude. Finally, the technique requires only a relatively small computational overhead. In the experiments on Leduc using our non-optimized implementation, we observed a factor of 2 per-iteration slowdown.

## References

Arjona-Medina, J. A.; Gillhofer, M.; Widrich, M.; Unterthiner, T.; and Hochreiter, S. 2018. Rudder: Return decomposition for delayed rewards. *CoRR* abs/1806.07857.

Bansal, T.; Pachocki, J.; Sidor, S.; Sutskever, I.; and Mordatch, I. 2018. Emergent complexity via multi-agent competition. In *Proceedings of the Sixth International Conference on Learning Representations*.

Bowling, M.; Burch, N.; Johanson, M.; and Tammelin, O. 2015. Heads-up Limit Hold'em Poker is solved. *Science* 347(6218):145–149.

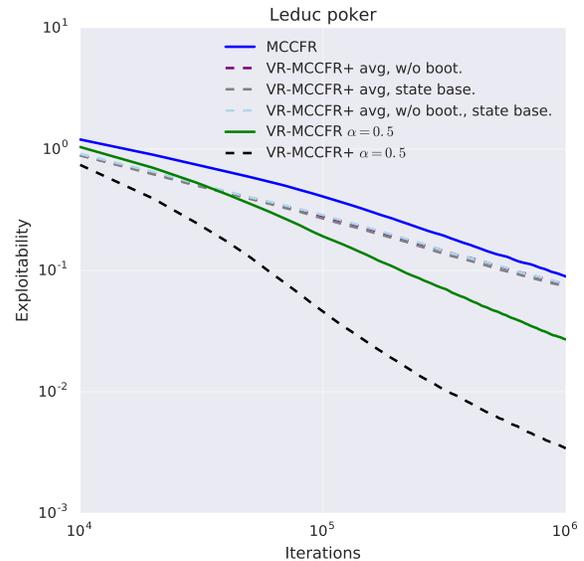


Figure 6: Detailed comparison of different VR-MCCFR variants on logarithmic scale. The curves for MCCFR, VR-MCCFR and VR-MCCFR+ are the same as in the previous plot, the other lines show how the algorithm performs when using state baselines instead of state-action baselines, and without bootstrapping. All of these reduced variants perform better than plain MCCFR, however they are worse than full VR-MCCFR. This ablation study shows that the combination of all VR-MCCFR features is important for final performance.

Boyle, P. P. 1977. Options: A monte carlo approach. *Journal of financial economics* 4(3):323–338.

Brown, N., and Sandholm, T. 2017. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science* 360(6385).

Burch, N.; Schmid, M.; Moravcik, M.; Morill, D.; and Bowling, M. 2018. Aivat: A new variance reduction technique for agent evaluation in imperfect information games. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Burch, N.; Johanson, M.; and Bowling, M. 2014. Solving imperfect information games using decomposition. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI)*.

Burch, N. 2017. *Time and Space: Why Imperfect Information Games are Hard*. Ph.D. Dissertation, University of Alberta.

Foerster, J. N.; Chen, R. Y.; Al-Shedivat, M.; Whiteson, S.; Abbeel, P.; and Mordatch, I. 2017. Learning with opponent-learning awareness. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Gibson, R.; Lanctot, M.; Burch, N.; Szafron, D.; and Bowling, M. 2012. Generalized sampling and variance in

- counterfactual regret minimization. In *Proceedings of the Twenty-Sixth Conference on Artificial Intelligence (AAAI-12)*, 1355–1361.
- Hart, S., and Mas-Colell, A. 2000. A simple adaptive procedure leading to correlated equilibrium. *Econometrica* 68(5):1127–1150.
- Heinrich, J.; Lanctot, M.; and Silver, D. 2015. Fictitious self-play in extensive-form games. In *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*.
- Johanson, M.; Bard, N.; Lanctot, M.; Gibson, R.; and Bowling, M. 2012. Efficient nash equilibrium approximation through Monte Carlo counterfactual regret minimization. In *Proceedings of the Eleventh International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.
- Lanctot, M.; Waugh, K.; Zinkevich, M.; and Bowling, M. 2009. Monte Carlo sampling for regret minimization in extensive games. In Bengio, Y.; Schuurmans, D.; Lafferty, J.; Williams, C. K. I.; and Culotta, A., eds., *Advances in Neural Information Processing Systems 22*, 1078–1086.
- Lanctot, M.; Zambaldi, V.; Gruslys, A.; Lazaridou, A.; Tuyls, K.; Perolat, J.; Silver, D.; and Graepel, T. 2017. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems*.
- Lanctot, M. 2013. *Monte Carlo Sampling and Regret Minimization for Equilibrium Computation and Decision-Making in Large Extensive Form Games*. Ph.D. Dissertation, University of Alberta, University of Alberta, Computing Science, 116 St. and 85 Ave., Edmonton, Alberta T6G 2R3.
- Littman, M. L. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, 157–163. Morgan Kaufmann.
- Liu, H.; Feng, Y.; Mao, Y.; Zhou, D.; Peng, J.; and Liu, Q. 2018. Action-dependent control variates for policy optimization via stein identity.
- Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T. P.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 1928–1937.
- Moravčík, M.; Schmid, M.; Burch, N.; Lisý, V.; Morrill, D.; Bard, N.; Davis, T.; Waugh, K.; Johanson, M.; and Bowling, M. 2017. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science* 358(6362).
- Owen, A. B. 2013. *Monte Carlo theory, methods and examples*.
- Pepels, T.; Tak, M. J.; Lanctot, M.; and Winands, M. H. 2014. Quality-based rewards for Monte-Carlo tree search simulations. In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI)*.
- Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Shoham, Y., and Leyton-Brown, K. 2009. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press.
- Southey, F.; Bowling, M. H.; Larson, B.; Piccione, C.; Burch, N.; Billings, D.; and Rayner, D. C. 2005. Bayes’ bluff: Opponent modelling in poker. In *UAI ’05, Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence*, 550–558.
- Srinivasan, S.; Lanctot, M.; Zambaldi, V.; Pérolat, J.; Tuyls, K.; Munos, R.; and Bowling, M. 2018. Actor-critic policy optimization in partially observable multiagent environments. In *Advances in Neural Information Processing Systems (NIPS) 31*.
- Sutton, R., and Barto, A. 2017. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition. Draft, in progress.
- Tammelin, O.; Burch, N.; Johanson, M.; and Bowling, M. 2015. Solving heads-up limit texas hold’em. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*.
- Tucker, G.; Bhupatiraju, S.; Gu, S.; Turner, R. E.; Ghahramani, Z.; and Levine, S. 2018. The mirage of action-dependent baselines in reinforcement learning. *arXiv preprint arXiv:1802.10031*.
- Veness, J.; Lanctot, M.; and Bowling, M. 2011. Variance reduction in Monte-Carlo tree search. In Shawe-Taylor, J.; Zemel, R.; Bartlett, P.; Pereira, F.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems 24*, 1836–1844.
- Waugh, K.; Morrill, D.; Bagnell, J. A.; and Bowling, M. 2015. Solving games with functional regret estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Williams, R. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* 8(3):229–256.
- Wu, C.; Rajeswaran, A.; Duan, Y.; Kumar, V.; Bayen, A. M.; Kakade, S.; Mordatch, I.; and Abbeel, P. 2018. Variance reduction for policy gradient with action-dependent factorized baselines. *CoRR* abs/1803.07246.
- Zinkevich, M.; Johanson, M.; Bowling, M.; and Piccione, C. 2008. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems 20 (NIPS 2007)*.