Low-Variance and Zero-Variance Baselines for Extensive-Form Games

Trevor Davis [1 †] **Martin Schmid** [2] **Michael Bowling** [2 1]

Abstract

Extensive-form games (EFGs) are a common model of multi-agent interactions with imperfect information. State-of-the-art algorithms for solving these games typically perform full walks of the game tree that can prove prohibitively slow in large games. Alternatively, sampling-based methods such as Monte Carlo Counterfactual Regret Minimization walk one or more trajectories through the tree, touching only a fraction of the nodes on each iteration, at the expense of requiring more iterations to converge due to the variance of sampled values. In this paper, we extend recent work that uses baseline estimates to reduce this variance. We introduce a framework of baseline-corrected values in EFGs that generalizes the previous work. Within our framework, we propose new baseline functions that result in significantly reduced variance compared to existing techniques. We show that one particular choice of such a function — predictive baseline — is provably optimal under certain sampling schemes. This allows for efficient computation of zero-variance value estimates even along sampled trajectories.

1. Introduction

Multi-agent strategic interactions are often modeled as *extensive-form games (EFGs)*, a game tree representation that allows for hidden information, stochastic outcomes, and sequential interactions. Research on solving EFGs has been driven by the experimental domain of poker games, in which the *Counterfactual Regret Minimization (CFR)* algorithm (Zinkevich et al., 2008) has been the basis of several breakthroughs. Approaches incorporating CFR have been used to essentially solve one nontrivial poker game (Bowling et al., 2015), and to beat human professionals in another (Moravčík et al., 2017; Brown & Sandholm, 2018).

CFR is in essence a policy improvement algorithm that iteratively evaluates and improves a strategy for playing an EFG. As part of this process, it must walk the entire game tree on every iteration. However, many games have prohibitively large trees when represented as EFGs. For example, many commonly played poker games have more possible game states than there are atoms in the universe (Johanson, 2013). In such cases, performing even a single iteration of traditional CFR is impossible.

The prohibitive cost of CFR iterations is the motivation for *Monte Carlo Counterfactual Regret Minimization (MCCFR)*, which samples trajectories to walk through the tree to allow for significantly faster iterations (Lanctot et al., 2009). Additionally, while CFR spends equal time updating every game state, the sampling scheme of MCCFR can be altered to target updates to parts of the game that are more critical or more difficult to learn (Gibson et al., 2012b;a). As a trade-off for these benefits, MCCFR requires more iterations to converge due to the variance of sampled values.

In the Reinforcement Learning (RL) community, variance reduction in sampling algorithms has been extensively studied. In particular, baseline functions that estimate state values are typically used within policy gradient methods to decrease the variance of value estimates along sampled trajectories (Williams, 1992; Greensmith et al., 2004; Bhatnagar et al., 2009; Schulman et al., 2016). Recent work by Schmid et al. (2019) has adapted these ideas to reduce variance in MCCFR, resulting in the VR-MCCFR algorithm.

In this work, we generalize and extend the ideas of Schmid et al. In Section 3, we introduce a framework for variance reduction of sampled values in EFGs by use of state-action baselines. In Section 4, we show that VR-MCCFR is a specific application of our baseline framework that unnecessarily generalizes across dissimilar states. We introduce alternative baseline functions that take advantage of our access to the full hidden state during training, avoiding this generalization. We improve on prior theoretical analysis of baseline performance, demonstrating that our baselines are more directly tailored to reducing error. In Section 5, we show empirically that our new baselines result in significantly reduced variance and faster convergence.

[†]Work done during an internship at DeepMind. [1]Alberta Machine Intelligence Institute, Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada [2]DeepMind, Edmonton, Alberta, Canada. Correspondence to: Trevor Davis <trdavis1@ualberta.ca>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

The variance of MCCFR updates can be further reduced through careful choice of sampling scheme. In Section 6 we examine how our baselines perform in a "vectorized" form of MCCFR introduced by Schmid et al. We show that with this sampling scheme, our proposed *predictive baseline* exactly tracks the expected utility of the current strategies, thus provably computing zero-variance sampled values. Using this baseline thus allows for practical computation of the theoretically optimal "oracle baseline" discussed by Schmid et al. For the first time, this allows for exact CFR updates to be performed along sampled trajectories.

Finally, we demonstrate that our framework is also effective in online game play. *Monte Carlo continual re-solving (MCCR)* (Šustr et al., 2019) uses MCCFR to solve subgames, allowing an agent to decide how to act only for situations encountered during execution while still guaranteeing approximate equilibrium play. In Section 7, we show that our baselines improve the convergence speed of subgame solving, allowing an MCCR agent to make faster decisions.

2. Background

An *extensive-form game (EFG)* (Osborne & Rubinstein, 1994) is a game tree, formally defined by a tuple $\langle N, H, P, \sigma_c, u, \mathcal{I}\rangle$. N is a finite set of players. H is a set of *histories*, where each history is a sequence of *actions* and corresponds to a vertex of the tree. For $h, h' \in H$, we write $h \sqsubseteq h'$ if h is a prefix of h'. The set of actions available at $h \in H$ that lead to a successor history $(ha) \in H$ is denoted $A(h)$. Histories with no successors are *terminal histories* $Z \subseteq H$. $P\colon H \setminus Z \to N \cup \{c\}$ maps each history to the player that chooses the next action, where c is the *chance* player that acts according to the defined distribution $\sigma_c(h) \in \Delta_{A(h)}$, where $\Delta_{A(h)}$ is the set of probability distributions over $A(h)$. The *utility function* $u\colon N \times Z \to \mathbb{R}$ assigns a value to each terminal history for each player.

For each player $i \in N$, the collection of *(augmented) information sets* $\mathcal{I}_i \in \mathcal{I}$ is a partition of the histories H.[1] Player i does not observe the true history h, but only the information set $I_i(h)$. Necessarily, this means that $A(h) = A(h')$ if $I_{P(h)}(h) = I_{P(h)}(h')$, which we then denote $A(I)$. For brevity, we will omit the subscript from $I_{P(h)}(h)$ when considering the acting player's information set.

Each player selects actions according to a *(behavioral) strategy* that maps each information set $I \in \mathcal{I}_i$ where $P(I) = i$ to a distribution over actions, $\sigma_i(I) \in \Delta_{A(I)}$. The probability of taking a specific action at a history is $\sigma_{P(h)}(h, a) = \sigma_{P(h)}(I(h), a)$. A *strategy profile*, $\sigma = \{\sigma_i | i \in N\}$, specifies a strategy for each player. The *reach probability* of a history h is $\pi^\sigma(h) = \prod_{(h'a)\sqsubseteq h} \sigma_{P(h')}(h', a)$. This product can be decomposed as $\pi^\sigma(h) = \pi_i^{\sigma_i}(h)\pi_{-i}^{\sigma_{-i}}(h)$, where the first term contains the actions of player i, and the second contains the actions of other players and chance. We also write $\pi^\sigma(h, h')$ for the probability of reaching h' from h, defined to be 0 if $h \not\sqsubseteq h'$. A strategy profile defines an expected utility for each player as $u_i(\sigma) = \sum_{z\in Z} \pi^\sigma(z)u_i(z)$.

In this work, we consider two-player zero-sum EFGs, in which $N = \{1, 2\}$ and $u(z) := u_i(z) = -u_{-i}(z)$. We also assume that the information sets satisfy *perfect recall*, which requires that players do not forget any information that they once observed. Mathematically, this means that two histories in the same information set I_i must have the same sequence of past information sets and actions for player i. All games played by humans exhibit perfect recall. We write $I_i \sqsubseteq h$ if there is any history $h' \in I_i$ such that $h' \sqsubseteq h$, and we denote that history (unique by perfect recall) by $I_i[h]$.

2.1. Solving EFGs

A common solution concept for EFGs is a *Nash equilibrium*, in which no player has an incentive to deviate from their specified strategy. We evaluate strategy profiles by their distance from equilibrium as measured by *exploitability*, the average expected loss against worst-case opponents:

$$\text{exploit}(\sigma) = {}^1\!/_2 \max_{\sigma' \in \Sigma}(u_2(\sigma_1, \sigma'_2) + u_1(\sigma'_1, \sigma_2)).$$

Counterfactual Regret Minimization (CFR) is an algorithm for learning Nash equilibria in EFGs through iterative self play (Zinkevich et al., 2008). For any $h \in H$, let $Z[h] = \{z \in Z \mid h \sqsubseteq z\}$ be the set of terminal histories reachable from h, and define the history's expected utility as $u(h|\sigma) = \sum_{z\in Z[h]} \pi^\sigma(h, z)u(z)$. For each information set I and action $a \in A(I)$, the counterfactual regret is

$$r^t(I, a) = \sum_{h\in I} \pi^{\sigma^t}_{-P(h)}(h) \left(u((ha)|\sigma) - u(h|\sigma)\right). \quad (1)$$

CFR accumulates these regrets over time as $R^T(I, a) = \sum_{t=1}^T r^t(I, a)$. The next strategy profile is then selected with *regret matching*, which sets probabilities proportional to the positive regrets: $\sigma^{T+1}(I, a) \propto \max(R^T(I, a), 0)$. Defining the average strategy $\overline{\sigma}^T$ such that $\overline{\sigma}^T(h, a) \propto \sum_{t=1}^T \pi_i^{\sigma^t}(h)\sigma_i^t(h, a)$, CFR guarantees that exploit$(\overline{\sigma}^T) \to 0$ as $T \to \infty$, thus converging to a Nash equilibrium.

The empirical convergence rate of CFR is greatly improved by the *CFR$^+$* variant, which greedily zeroes all negative regrets on every iteration, replacing R^t with an accumulant Q^t recursively defined with $Q^0(I, a) = 0$, $Q^t(I, a) = \max(Q^{t-1}(I, a) + r^t(I, a), 0)$ (Tammelin et al., 2015). It also alternates updates for each player, and uses linear averaging, which gives greater weight to more recent strategies.

[1] Augmented information sets were introduced by Burch et al. (2014).

CFR($^+$) requires a full walk of the game tree on each iteration, which can be a very costly operation on large games. *Monte Carlo Counterfactual Regret Minimization (MCCFR)* avoids this cost by only updating along sampled trajectories. For simplicity, we focus on the *outcome sampling (OS)* variant of MCCFR (Lanctot et al., 2009), though all results in this paper can be trivially extended to other MCCFR variants. On each iteration t, a sampling strategy $q^t \in \Sigma$ is used to sample a single terminal history $z^t \sim \pi^{q^t}$. A sampled utility is then calculated recursively for each prefix of z^t as

$$\begin{aligned}\hat{u}(h,a|\sigma^t,z^t) &= \frac{\mathbb{1}((ha) \sqsubseteq z^t)}{q^t(h,a)}\hat{u}((ha)|\sigma^t,z^t)\\ \hat{u}(h|\sigma^t,z^t) &= \sum_{a\in A(h)} \sigma^t(h,a)\hat{u}(h,a|\sigma^t,z^t)\end{aligned} \tag{2}$$

where $\mathbb{1}$ is the indicator function and $\hat{u}(z^t|\sigma^t,z^t) = u(z^t)$. For any $h \sqsubseteq z^t$, the sampled value $\hat{u}(h,a|\sigma^t,z^t)$ is an unbiased estimate of the expected utility $u((ha)|\sigma^t)$, whether a is sampled or not. These sampled values are used to calculate a sample of the counterfactual regret $\hat{r}^t(I,a|z^t) =$

$$\sum_{h\in I} \frac{\pi^{\sigma^t}_{-P(h)}(h)}{\pi^{q^t}(h)}\left(\hat{u}(h,a|\sigma^t,z^t) - \hat{u}(h|\sigma^t,z^t)\right) \tag{3}$$

This gives an unbiased sample of the counterfactual regret $r^t(I,a)$, which is then used to perform unbiased CFR updates. As long as as the sampling strategies satisfy $\pi^{q^t}(z) > 0$ for all $z \in Z$, MCCFR guarantees that exploit$(\overline{\sigma}^T) \to 0$ with high probability, thus converging to a Nash equilibrium. However, the rate of convergence depends on the variance of $\hat{r}^t(I,a|z^t)$ (Gibson et al., 2012b).

3. Baseline framework for EFGs

We now introduce a method for calculating unbiased estimates of utilities in EFGs that has lower variance than the sampled utilities $\hat{u}(h,a|\sigma^t,z^t)$ defined above. We do this using *baseline functions*, which estimate the expected utility of actions in the game. We will describe specific examples of such functions in Section 4; for now, we assume the existence of some function $b^t\colon H \times A \to \mathbb{R}$ such that $b^t(h,a)$ in some way approximates $u((ha)|\sigma^t)$. We define a baseline-corrected sampled utility as

$$\begin{aligned}&\hat{u}_b(h,a|\sigma^t,z^t)\\ &= \frac{\mathbb{1}((ha) \sqsubseteq z^t)}{q^t(h,a)}\left(\hat{u}_b((ha)|\sigma^t,z^t) - b^t(h,a)\right) + b^t(h,a)\\ &\hat{u}_b(h|\sigma^t,z^t) = \sum_{a\in A(h)} \sigma^t(h,a)\hat{u}_b(h,a|\sigma^t,z^t)\end{aligned} \tag{4}$$

Equation (4) comes from the application of a *control variate*, in which we lower the variance of a random variable ($X = \frac{\mathbb{1}((ha)\sqsubseteq z^t)}{q^t(h,a)}\hat{u}_b((ha)|\sigma^t,z^t)$) by subtracting another random variable ($Y = \frac{\mathbb{1}((ha)\sqsubseteq z^t)}{q^t(h,a)}b^t(h,a)$) and adding its known expectation ($\mathbb{E}[Y] = b^t(h,a)$), thus keeping the resulting estimate unbiased. If X and Y are positively correlated, then this estimate will have lower variance than X itself. Because $\hat{u}_b((ha)|\sigma^t,z^t)$ is defined recursively, its computation includes the application of independent control variates at every action taken between h and z^t.

These estimates are unbiased, and their variance is directly proportional to a measure of aggregate squared error in the baseline function:

Theorem 1. *For any $h \sqsubseteq z^t$ and any $a \in A(h)$, the baseline-corrected utilities satisfy*

$$\begin{aligned}\mathbb{E}_{z^t}\left[\hat{u}_b(h,a|\sigma^t,z^t)|z^t \sqsupseteq h\right] &= u((ha)|\sigma^t)\\ \mathbb{E}_{z^t}\left[\hat{u}_b(h|\sigma^t,z^t)|z^t \sqsupseteq h\right] &= u(h|\sigma^t)\end{aligned}$$

Theorem 2. *For any $h \sqsubseteq z^t$ and any $a \in A(h)$, the baseline-corrected utilities satisfy*

$$\begin{aligned}&\mathrm{Var}_{z^t}\left[\hat{u}_b(h,a|\sigma^t,z^t)|z^t \sqsupseteq h\right]\\ &\leq \sum_{(h'a')\sqsupseteq(ha)}\left(\frac{(\pi^{\sigma^t}((ha),(h'a')))^2}{\pi^{q^t}(h,(h'a'))}\right.\\ &\qquad\left.\cdot\left(u((h'a')|\sigma^t) - b^t(h',a')\right)^2\right)\end{aligned}$$

Full proofs are given in the supplementary materials. Theorem 1 show that we can use $\hat{u}_b(h,a|\sigma^t,z^t)$ in place of $\hat{u}(h,a|\sigma^t,z^t)$ in equation 3 and maintain the convergence guarantees of MCCFR. Theorem 2 shows that the variance of the resulting estimates depends only on the squared difference between the baseline estimates and the true expected utilities. In particular, using an ideal baseline eliminates all variance in the MCCFR update. By choosing our baseline well, we decrease the MCCFR variance and speed up its convergence. Pseudocode for MCCFR with baseline-corrected values is given in the supplementary materials.

Although we focus on using our baseline-corrected samples in MCCFR, nothing in the value definition is particular to that algorithm. In fact, a lower variance estimate of sampled utilities is useful in any algorithm that performs iterative training using sampled trajectories, such as policy gradient methods (Srinivasan et al., 2018) or stochastic first-order methods (Kroer et al., 2015).

4. Baselines for EFGs

In this section we examine specific baseline functions that can be used during iterative training. We show how MC-

CFR (without a baseline) and VR-MCCFR can be be reconstructed in our baseline framework, and we propose three novel baseline functions. Theorem 2 shows that we can minimize variance by choosing a baseline function b^t such that $b^t(h,a) \approx u((ha)|\sigma^t)$.

No baseline (MCCFR). We begin by examining MCCFR under its original definition, where no baseline function is used. We note that when we run baseline-corrected MCCFR with a static choice of $b^t(h,a) = 0$ for all h, a, the operation of the algorithm is identical to MCCFR. Thus, opting to not use a baseline is, in itself, a choice of a very particular baseline.

Using $b^t(h,a) = 0$ might seem like a reasonable choice when we expect the game's payouts to be balanced between the players. However, even when the overall expected utility $u(\sigma)$ is very close to 0, there will usually be particular histories with high magnitude expected utility $u(h|\sigma)$. For example, in poker games, the expected utility of a history is heavily biased toward the player who has been dealt better cards, even if these biases cancel out when considered across all histories. In fact, often there is no strategy profile at all that satisfies $u((ha)|\sigma) = 0$, which makes $b^t(h,a) = 0$ a poor choice in regards to the ideal criteria $b^t(h,a) \approx u((ha)|\sigma^t)$. An example game where a zero baseline performs very poorly is explored in Section 5.

Static strategy baseline. The simplest way to ensure that the baseline function does correspond to an actual strategy is to choose a static, known strategy profile $\sigma^b \in \Sigma$ and let $b^t(h,a) = u((ha)|\sigma^b)$ for each time t. Once the strategy is chosen, the baseline values only need to be computed once and stored. In general this requires a full walk of the game tree, but it is sometimes possible to take advantage of the structure of the game to greatly reduce this cost. For an example, see Section 5.

Learned history baseline. Using a static strategy for our baseline ensures that it corresponds to some expected utility, but it fails to take advantage of the iterative nature of MCCFR. In particular, when attempting to estimate $u((ha)|\sigma^t)$, we have access to past samples $\hat{u}_b((ha)|\sigma^\tau, z^\tau)$ for $\tau < t$. Because the strategy is changed incrementally, we might expect the expected utility to change slowly and for these to be reasonable samples of the utility at time t as well.

Define $\mathcal{T}^{ha}(t) = \{\tau < t \mid (ha) \sqsubseteq z^\tau\}$ to be the set of timesteps on which (ha) was sampled, and denote the jth such timestep as τ_j. The *learned history baseline* sets

$$b^t(h,a) = \sum_{j=1}^{|\mathcal{T}^{ha}(t)|} w_j \hat{u}_b((ha)|\sigma^{\tau_j}, z^{\tau_j}) \tag{5}$$

where $(w_j)_{j=1}^{|\mathcal{T}^{ha}(t)|}$ is a sequence of nonnegative weights with $\sum_{j=1}^{|\mathcal{T}^{ha}(t)} w_j \leq 1$. Possible weighting choices include simple averaging, where $w_j = 1/|\mathcal{T}^{ha}(t)|$, and exponentially-decaying averaging, where $w_j = \alpha(1-\alpha)^{|\mathcal{T}^{ha}(t)|-j}$ for some $\alpha \in (0,1]$.

Learned infoset baseline (VR-MCCFR). The learned history baseline is similar to the VR-MCCFR baseline defined by Schmid et al. (2019). The principle difference is that the VR-MCCFR baseline tracks values for each information set, rather than for each history; we thus refer to it as the *learned infoset baseline*. This baseline also updates values for each player separately, based on their own information sets. This can be accomplished by tracking separate values for each player throughout the tree walk, or by running MCCFR with alternating updates, where only one player's regrets are updated on each tree walk. The learned infoset baseline is defined as $b^t(h,a) = b^t(I_i(h),a)$ where

$$b^t(I_i,a) = \sum_{j=1}^{|\mathcal{T}^{I_i a}(t)|} w_j \hat{u}_b((I_i[z^{\tau_j}]a)|\sigma^{\tau_j}, z^{\tau_j}) \tag{6}$$

where i is the player being updated, $\mathcal{T}^{I_i a}(t)$ is the set of timesteps on which $(h'a)$ was sampled for any $h' \in I_i$, and τ_j is jth such timestep. Following Schmid et al. we consider both simple averaging and exponentially-decaying averaging for selecting the weights w_j. Running MCCFR with this baseline exactly reproduces VR-MCCFR.

Predictive baseline. When we use one of the learned baselines with MCCFR, on iteration t we generate sampled values which are used both to update the baseline estimates b^{t+1} and to update the strategies σ^{t+1} (via regret matching). Thus b^{t+1} is updated with regards to σ^t, despite the theory that the optimal value for b^{t+1} depends on σ^{t+1}. As an alternative, we can first update the strategy, then use the new strategy in the baseline update. In particular, the *predictive baseline* recursively defines the baseline values as estimated values with respect to the newly computed strategy. Formally, it updates

$$b^{t+1}(h,a) = \begin{cases} u(z^t) & \text{if } (ha) = z^t \\ \sum_{a'} \sigma^{t+1}((ha),a') b^{t+1}((ha),a') & \text{if } (ha) \sqsubset z^t \\ b^t(h,a) & \text{otherwise.} \end{cases} \tag{7}$$

The recursive nature of this definition is possible due to the depth-first order of MCCFR updates. Examining the recursive value definition (4), it can be seen that (7) is equivalent to the definition $b^{t+1}(h,a) = \hat{u}_b((ha)|\sigma^{t+1}, z^t)$ for sampled (ha).[2] Thus we set $b^{t+1}(h,a)$ to an unbiased estimate of $u((ha)|\sigma^{t+1})$, aiming to minimize the error term in Theorem 2.

[2] See Lemma 3 in the supplementary materials.

5. Experimental comparison

We run our experiments using a commodity desktop machine in Leduc hold'em (Southey et al., 2005), a small poker game commonly used as a benchmark in games research[3]. We compare the effect of the various baselines on the MCCFR convergence rate. Our experiments use the regret zeroing and linear averaging of CFR$^+$, as these improve convergence when combined with any of the nonzero baselines examined in this work. For the static strategy baseline, we use the "always call" strategy, which matches the opponent's bets and makes no bets of its own. Expected utility under this strategy is determined by the current size of the pot, which is measurable at run time, and the winning chance of each player's cards. Before training, we measure and store these odds for all possible sets of cards, which is significantly smaller than the size of the full game. For both of the learned baselines, we use simple averaging as it performed best in preliminary experiments.

We run experiments with two sampling strategies. The first is uniform sampling, in which $q^t(h,a) = 1/|A(h)|$. The second is opponent on-policy sampling, which depends on the player i being updated: we sample uniformly ($q^t(h,a) = 1/|A(h)|$) at histories h where $P(h) = i$, and sample on-policy ($q^t(h,a) = \sigma^t(h,a)$) otherwise. For consistency, we use alternating updates for both schemes.

Figures 1a and 1b show the convergence of MCCFR with the various baselines, as measured by exploitability (recall that exploitability converges to zero). All results in this paper are averaged over 20 runs, with 95% confidence intervals shown as error bands (often too narrow to be visible). We compare algorithms in terms of number of iterations in order to avoid implementation details. In our implementation we found the choice of baseline to have no noticeable impact on iteration time. With either sampling scheme, the learned infoset (VR-MCCFR) baseline improves on using no baseline at all, while the other three baselines achieve a significant improvement on top of that.

Many true expected values in Leduc are very close to zero, making MCCFR without a baseline (i.e. $b^t(h,a) = 0$) better than it might otherwise be. To demonstrate the necessity of a baseline in some games, we ran MCCFR in a modified Leduc game where player 2 always transfers 100 chips to player 1 after every game. This utility change is independent of the player's actions, so it doesn't strategically change the game. However, it means that 0 is now an extremely inaccurate value estimate for all histories. Figure 1c shows convergence in Leduc with shifted utilities. Here we see that using any baseline at all provides a significant advantage over not using a baseline, due to the ability to adapt to the shifted utilities. The always call baseline converges quickest because its baseline values are initialized with the shift, avoiding the need to learn the shift during training. In fact, the exploitability values for always call are identical to those in the original Leduc experiment because all utilities and baseline values are shifted by the same amount, and these shifts cancel out when calculating counterfactual regret. The shift causes expected utilities to be strongly correlated between histories, making generalization in more effective when learning a baseline; this effect is demonstrated by the learned infoset baseline outperforming the other learned baselines early on.

[3] An open source implementation of CFR$^+$ and Leduc hold'em is available from the University of Alberta (http://webdocs.cs.ualberta.ca/~games/poker/cfr_plus.html).

6. Public Outcome Sampling

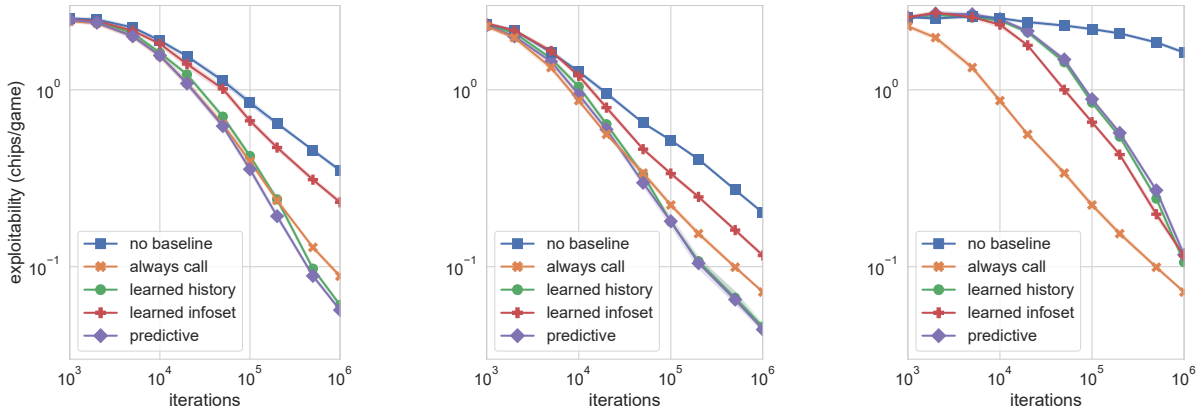
So far we have shown that using a baseline improves the convergence rate in outcome sampling MCCFR. The baselines effectively reduce the variance of estimating expected values from sampled trajectories. The baseline framework, however, does not address an additional source of variance, in which the counterfactual regret for an information set is estimated from a single sampled history in the information set. Schmid et al. (2019) avoided this variance in their VR-MCCFR experiments by using a "vectorized" form of MCCFR, which evaluates every possible history consistent with the players' shared observations along a sampled trajectory. Schmid et al. do not formally define their algorithm. We refer to it as *Public Outcome Sampling (POS)* as the algorithm samples any actions that are publicly visible to both players, while exhaustively considering all possible private states. We give a full formal definition of POS in the supplementary materials. In this section we examine how our baselines perform when combined with the reduced variance of POS sampling. In particular, we show that this setting allows strong theoretical guarantees for the predictive baseline.

6.1. Baselines in POS

In MCCFR with POS, we still use action baselines $b^t(h,a)$ with the ideal baseline values being $b^t(h,a) = u((ha)|\sigma^t)$. Thus the baselines in Section 4 apply to this setting as well.

For the learned infoset baseline, we have more information available to us than in the OS case. This is because when POS samples some history-action pair h, a, it also samples every pair h', a for $h' \in I(h)$. Thus, rather, than using one sampled history value to update the baseline, we use a weighted sample of all of the history values. Following Schmid et al., we weight the baseline values $b^t(I_i, a) =$

$$\sum_{j=1}^{|\mathcal{T}^{I_i a}(t)|} w_j \frac{\sum_{h' \in I_i} \pi_{-i}^{\sigma^{\tau_j}}(h') \hat{u}_b((h'a)|\sigma^{\tau_j}, z^{\tau_j})}{\sum_{h' \in I_i} \pi_{-i}^{\sigma^{\tau_j}}(h')}.$$



(a) Uniform sampling (b) Opponent on-policy sampling (c) Shifted utilities

Figure 1: Log-log plots of convergence of MCCFR strategies with various baselines. **(a)** and **(b)** Leduc with different MCCFR sampling schemes. **(c)** Leduc with utilities shifted by 100 and opponent on-policy sampling.

This is the same relative weighting given to each history when calculating the counterfactual regret.

Zero-variance baseline. POS also has implications for the predictive baseline. In fact, we can guarantee that after every outcome of the game has been sampled, the predictive baseline will have learned the true value of the current strategy. For time t, let Z^t be the set of sampled terminal histories (consistent with a public outcome), and let $\text{samp}^t(h)$ be the event that h is sampled on way to Z^t.

Theorem 3. *If each of the terminal states $Z[h]$ reachable from history $h \in H$ has been sampled at least once under public outcome sampling ($Z[h] \subseteq \bigcup_{\tau<t} Z^\tau$), then for all $a \in A(h)$ the predictive baseline satisfies*

$$b^t(h,a) = u((ha)|\sigma^t)$$
$$\text{Var}_{Z^t}\left[\hat{u}_b(h|\sigma^t, Z^t)|\text{samp}^t(h)\right] = 0.$$

The proof is in the supplementary materials. In order for the theorem to hold everywhere in the tree, all outcomes must be sampled, which could take a large number of iterations. An alternative is to guarantee that all outcomes are sampled during the early iterations of MCCFR. For example, one could do a full CFR tree walk on the very first iteration, and then sample on subsequent iterations. Alternatively, we can ensure the theorem always holds with smart initialization of the baseline. When there are no regrets accumulated, MCCFR uses an arbitrary strategy. If we have some strategy with known expected values throughout the tree, we can use this strategy as the default MCCFR strategy and initialize the baseline values to the strategy's expected values. Either option guarantees that all regret updates will use zero-variance values.

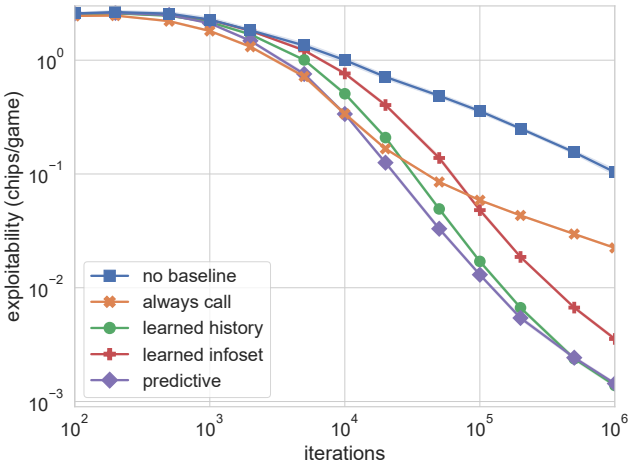


Figure 2: Log-log plot of MCCFR convergence with POS.

6.2. POS results

As in Section 5, we run experiments in Leduc and use CFR$^+$ updates. For the learned baselines, we use exponentially-decaying averaging with $\alpha = 0.5$, which preliminary experiments found to outperform simple averaging when combined with POS. For simplicity and consistency with the experiments of Schmid et al. (2019), we use uniform sampling and simultaneous updates.

Figure 2 compares the baselines' effects on POS MCCFR. We find that using any baseline provides a significant improvement on using no baseline. The always call baseline performs well early but tales off as it doesn't learn during training. Even with POS, where we always see an entire information set at a time, the learned infoset baseline (VR-MCCFR) is significantly outperformed by the learned history and predictive baselines. This is likely because the

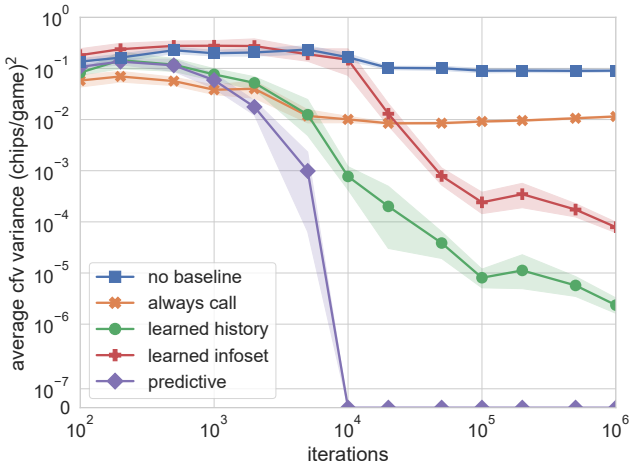


Figure 3: Log-log plot of average variance of counterfactual values during MCCFR solving with POS.

learned infoset baseline has to learn the relative weighting between histories in an infoset, while the other baselines always use the current strategy to weight the learned values. Finally, we observe that the predictive baseline has a small, but statistically significant, advantage over the learned history baseline in early iterations.

In addition, we compare the baselines by directly measuring their variance. To do this we first run MCCFR for the specified number of iterations, then freeze the strategy. We then walk every information set-action pair in the game tree, and for each such pair we run a large number of sampled trajectories originating at the pair. These trajectories are walked as if we were running MCCFR with POS, but we do not update the strategy. Instead, we only calculate the sampled counterfactual value $\sum_{h \in I} \pi^{\sigma^t}_{-P(h)}(h)\hat{u}_b((ha)|\sigma^t, z^t)$ at the initial I, a pair. From these samples, we compute an estimate of the true variance of the counterfactual value. Finally, we average these variance estimates across all information set-action pairs in the game.

Results are shown in Figure 3. We see that using no baseline results in high and relatively steady variance of counterfactual values. Using the always call baselines also results in steady variance, as nothing is learned, but at approximately an order of magnitude lower than no baseline. Variance with the other baselines improves over time, as the baseline becomes more accurate. The learned history baseline mirrors the learned infoset baseline, but with more than an order of magnitude reduction in variance. The predictive baseline is best of all, and in fact we see Theorem 3 in action as the variance drops to zero.

Finally, we examine how the baselines scale with game size and with the number of histories in information sets, using two versions of Generic Poker (r,4,4,1) with $r = 6$ and $r = 13$ (Lisý et al., 2015). These games respectively use decks containing 24 and 52 cards, compared to 6 for Leduc hold'em, and allow 4 raise actions per betting round instead of 2. The game sizes are compared in Table 1. Figure 4 shows the convergence of POS MCCFR with various baselines in these game. The results in the larger games are consistent with the Leduc results, showing that baselines offer considerable improvement in convergence speed, with the predictive baseline performing best. For the learned baselines, we used exponentially decaying averages with $\alpha = 0.5$ because we found this to work well in Leduc hold'em—it is possible that a different weighting scheme would perform better in these games.

Table 1: Sizes of poker games used in experiments.

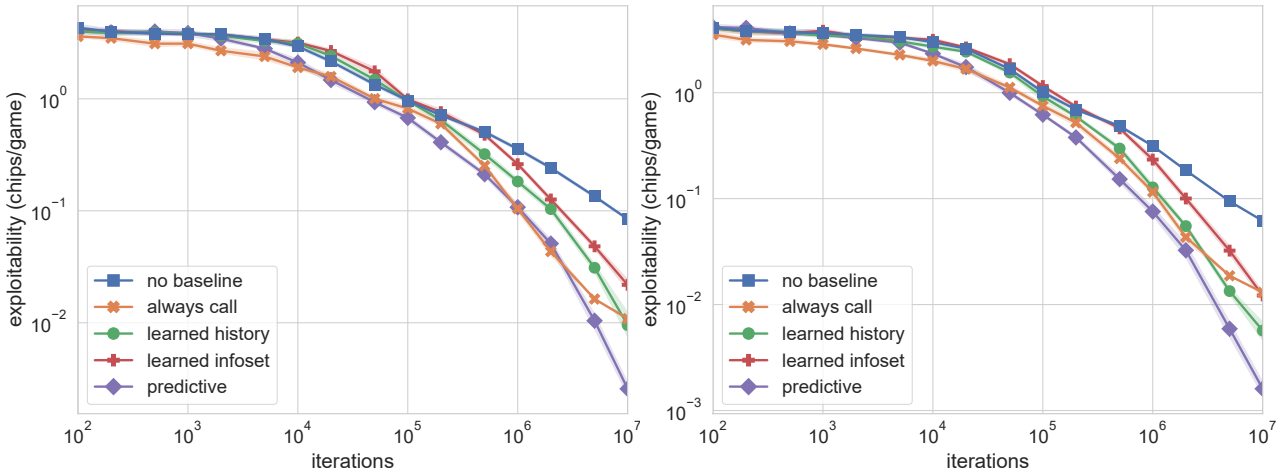
GAME	$\lvert H\rvert$	$\lvert\mathcal{I}\rvert$	$\max \lvert I\rvert$
LEDUC HOLD'EM	9450	936	5
GP(6,4,4,1)	$\approx 3 \cdot 10^6$	$\approx 5 \cdot 10^4$	23
GP(13,4,4,1)	$\approx 3 \cdot 10^7$	$\approx 2 \cdot 10^5$	51

7. Baselines in Monte Carlo continual re-solving

So far in this work we have concentrated on game solving, in which offline training is used to find a fixed strategy for playing a game. In contrast, strong agents in perfect information games often decide how to act only in situations encountered during online play. Traditionally, this approach was intractable in games with imperfect information because there was no way to guarantee that individually computed decisions would fit together into a cohesive equilibrium strategy. Recently, however, techniques have been developed for safe and efficient online computation of strategies in imperfect information games (Moravčík et al., 2017; Brown & Sandholm, 2017; Brown et al., 2018).

A particular example of this new paradigm, *continual re-solving*, was used in the DeepStack agent which defeated poker professionals (Moravčík et al., 2017). Each time a continual re-solving agent must select an action, CFR$^+$ is used to solve a relatively small subgame. Šustr et al. (2019) replaced the CFR$^+$ solver with MCCFR, creating *Monte Carlo continual re-solving (MCCR)*. It is straightforward to use our baseline framework within MCCR.

We conducted an experiment examining MCCR and baselines in Leduc hold'em. We measure the exploitability of strategy profiles that are constructed by independently re-solving each decision point as if they were encountered during online self-play. For each decision, we solve a subgame of depth three (i.e. looking a maximum of three actions into the future). After three actions, we approximately evaluate histories by running 100 iterations of CFR$^+$ on the subtree



(a) GP(6,4,4,4,1) (b) GP(13,4,4,4,1)

Figure 4: Log-log plots of POS MCCFR convergence in Generic Poker games.

rooted at the histories and calculating expected utilities under the resulting strategy.[4] At each decision point, we run re-solving until we have performed a maximum number of evaluations, either at terminal histories or depth-limited evaluations. We use this as an implementation-independent way of comparing algorithms, as evaluations take the vast majority of computation time in continual re-solving.

We compare MCCR with and without baselines. We use CFR$^+$ updates, which we've found to decrease variance when used with any nonzero baseline, and public outcome sampling with uniform sampling. Because of the inexact nature of the evaluation function, Theorem 3 does not hold in this setting, and we found the learned history baseline to perform best.[5] We also compare to (deterministic) continual re-solving, with both CFR and CFR$^+$ update rules.

Results are shown in Figure 5. The inclusion of a baseline significantly decreases the exploitability of MCCR strategies. Without a baseline, MCCR is not competitive with deterministic continual re-solving. With a baseline, it is able to clearly outperform continual resolving with CFR updates, and slightly outperform continual re-solving with CFR$^+$ updates. This is especially notable because there is still plenty of room to improve the technique, such as by tuning the baseline and by refining the sampling strategy.

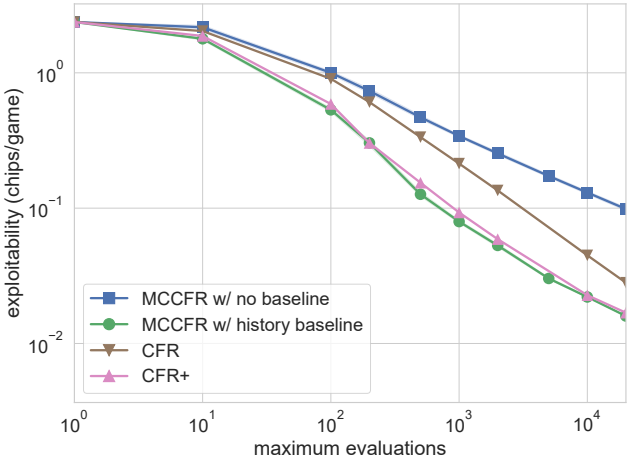


Figure 5: Exploitability of continual re-solving strategies based on the maximum number of evaluations per resolve.

8. Related Work

As discussed in the introduction, the use of baseline functions has a long history in RL. Typically these approaches have used state value baselines, with some recent exceptions (Liu et al., 2018; Wu et al., 2018). Tucker et al. (2018) suggest an explanation for this by isolating the variance terms that come from sampling an immediate action and from sampling the rest of a trajectory. Typical RL baselines only reduce the action variance, so the additional benefit from using a state-action baseline is insignificant when compared to the trajectory variance. In our work, we apply a recursive baseline to reduce both the action and trajectory variances, meaning state-action baselines give a noticeable benefit.

In multiagent RL, MADDPG (Lowe et al., 2017) and COMA (Foerster et al., 2018) are actor-critic methods that use a critic that evaluates the true game state (rather than the acting player's observations) to reduce variance during training. This is analogous to our baseline functions that

[4] This strategy, which contains errors because of the low number of iterations, approximates using a neural net for evaluation.

[5] Results for other baselines are shown in the supplementary materials.

evaluate histories rather than information sets.

In RL, the doubly-robust estimator (Jiang & Li, 2016) has been used to reduce variance settings by the recursive application of control variates (Thomas & Brunskill, 2016). Similarly, variance reduction in EFGs via recursive control variates is the basis of the advantage sum estimator (Zinkevich et al., 2008) and AIVAT (Burch et al., 2018). All of these techniques construct control variates by evaluating a static policy or strategy, either on the true game or on a static model. In this sense they are equivalent to our static strategy baseline. However, to the best of our knowledge, these techniques have only been used for evaluation of static strategies, rather than for variance reduction during training. Our work extends the EFG techniques to the training domain; we believe that similar ideas can be used in RL, and this is an interesting avenue of future research.

Concurrent to this work, Zhou et al. (2018) also suggested tracking true values of histories in a CFR variant, analogous to our predictive baseline. They use these values for truncating full tree walks, rather than for variance reduction along sampled trajectories. As such, they always initialize their values with a full tree walk, and don't examine gradually learning the values during training.

9. Conclusion and Future Work

In this work we introduced a new framework for variance reduction in EFGs through the application of a baseline value function. We demonstrated that the existing VR-MCCFR baseline can be described in our framework with a specific baseline function, and we introduced other baseline functions that significantly outperform it in practice. In addition, we introduced a predictive baseline and showed that it gives provably optimal performance under a sampling scheme that we formally define.

There are three sources of variance when performing sampled updates in EFGs. The first is from sampling trajectory values, the second from sampling individual histories within an information set that is being updated, and the third from sampling which information sets will be updated on a given iteration. By introducing MCCFR with POS, we provably eliminate the first two sources of variance: the first because we have a zero-variance baseline, and the second because we consider all histories within the information set. For the first time, this allows us to select the MCCFR sampling strategy q^t entirely on the basis of minimizing the third source of variance, by choosing the "best" information sets to update. Doing this in a principled way is an exciting avenue for future research.

Finally, we close by discussing function approximation. All of the baselines introduced in this paper require an amount of memory that scales with the size of the game tree. In contrast, baseline functions in RL typically use function approximation, requiring a much smaller number of parameters. Additionally, these functions generalize across states, which can allow for learning an accurate baseline function more quickly. The framework that we introduce in this work is completely compatible with function approximation, and combining the two is an area for future research.

References

Bhatnagar, S., Sutton, R. S., Ghavamzadeh, M., and Lee, M. Natural actor–critic algorithms. *Automatica*, 45(11): 2471–2482, 2009.

Bowling, M., Burch, N., Johanson, M., and Tammelin, O. Heads-up limit hold'em poker is solved. *Science*, 347 (6218):145–149, January 2015.

Brown, N. and Sandholm, T. Safe and nested subgame solving for imperfect-information games. In *Advances in Neural Information Processing Systems 30*, 2017.

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

Brown, N., Sandholm, T., and Amos, B. Depth-limited solving for imperfect-information games. In *Advances in Neural Information Processing Systems 31*, 2018.

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

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

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Gibson, R., Burch, N., Lanctot, M., and Szafron, D. Efficient Monte Carlo counterfactual regret minimization in games with many player actions. In *Proceedings of the Twenty-Sixth Conference on Advances in Neural Information Processing Systems*, 2012a.

Gibson, R., Lanctot, M., Burch, N., Szafron, D., and Bowling, M. Generalized sampling and variance in counterfactual regret minimization. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012b.

Greensmith, E., Bartlett, P. L., and Baxter, J. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov): 1471–1530, 2004.

Jiang, N. and Li, L. Doubly robust off-policy value evaluation for reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, 2016.

Johanson, M. Measuring the size of large no-limit poker games. Technical Report TR13-01, Department of Computing Science, University of Alberta, 2013.

Kroer, C., Waugh, K., Kilinç-Karzan, F., and Sandholm, T. Faster first-order methods for extensive-form game solving. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, 2015.

Lanctot, M., Waugh, K., Zinkevich, M., and Bowling, M. Monte Carlo sampling for regret minimization in extensive games. In *Advances in Neural Information Processing Systems 22*, 2009.

Lisý, V., Lanctot, M., and Bowling, M. Online Monte Carlo counterfactual regret minimization for search in imperfect information games. In *Proceedings of the 14th International Conference on Autonomous Agents and Multi-Agent Systems*, 2015.

Liu, H., Feng, Y., Mao, Y., Zhou, D., Peng, J., and Liu, Q. Action-dependent control variates for policy optimization via Stein identity. In *International Conference on Learning Representations*, 2018.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems 30*, 2017.

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

Osborne, M. J. and Rubinstein, A. *A Course in Game Theory*. The MIT Press, 1994. ISBN 0-262-65040-1.

Schmid, M., Burch, N., Lanctot, M., Moravcik, M., Kadlec, R., and Bowling, M. Variance reduction in monte carlo counterfactual regret minimization (VR-MCCFR) for extensive form games using baselines. In *Proceedings of the The Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

Southey, F., Bowling, M., Larson, B., Piccione, C., Burch, N., Billings, D., and Rayner, C. Bayes' bluff: Opponent modelling in poker. In *Proceedings of the Twenty-First Conference on Uncertainty in Artficial Intelligence*, 2005.

Srinivasan, S., Lanctot, M., Zambaldi, V., Pérolat, J., Tuyls, K., Munos, R., and Bowling, M. Actor-critic policy optimization in partially observable multiagent environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Tammelin, O., Burch, N., Johanson, M., and Bowling, M. Solving heads-up limit Texas hold'em. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 2015.

Thomas, P. S. and Brunskill, E. Data-efficient off-policy policy evaluation for reinforcement learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, 2016.

Tucker, G., Bhupatiraju, S., Gu, S., Turner, R., Ghahramani, Z., and Levine, S. The mirage of action-dependent baselines in reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Šustr, M., Kovařík, V., and Lisý, V. Monte Carlo continual resolving for online strategy computation in imperfect information games. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 2019.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Wu, C., Rajeswaran, A., Duan, Y., Kumar, V., Bayen, A. M., Kakade, S. M., Mordatch, I., and Abbeel, P. Variance reduction for policy gradient with action-dependent factorized baselines. In *International Conference on Learning Representations*, 2018.

Zhou, Y., Ren, T., Li, J., Yan, D., and Zhu, J. Lazy-CFR: a fast regret minimization algorithm for extensive games with imperfect information. *CoRR*, abs/1810.04433, 2018. URL http://arxiv.org/abs/1810.04433.

Zinkevich, M., Johanson, M., Bowling, M., and Piccione, C. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems 20*, 2008.