1 Supplementary Material

1.1 MCCFR with baseline-corrected values

Pseudocode for MCCFR with baseline-corrected values is given in Algorithm 1. Quantities of the form $\sigma^t(h, \cdot)$ refer to the vector of all quantities $\sigma^t(h, a)$ for $a \in A(h)$. The regrets and baseline values are initialized to 0, and the strategies are initialized arbitrarily (e.g. to uniform random). The inputs to the UPDATEBASELINE procedure depend on the particular baseline function used; for example, the learned baselines use the current sampled value $\hat{u}_b(h, a|\sigma^t, z^t)$ while the predictive baseline uses the newly computed strategy σ^{t+1} along with the baseline values of successor states. This algorithm has the same worst-case iteration complexity as MCCFR without baselines, namely $\mathcal{O}(d|A_{\text{max}}|)$ where d is the tree's depth and $|A_{\text{max}}| = \max_h |A(h)|$.

Algorithm 1 MCCFR w/ baseline

```
function MCCFR($h$)
    if $h \in Z$ then return $u(h)$
    $\overline{\sigma}^t(h, \cdot) \leftarrow \frac{t-1}{t}\overline{\sigma}^{t-1}(h, \cdot) + \frac{1}{t}\sigma^t$
    sample action $a \sim q^t(h, \cdot)$
    $\hat{u}_b((ha)|\sigma^t, z^t) \leftarrow \text{MCCFR}((ha))$
    $\hat{u}_b(h, a'|\sigma^t, z^t) \leftarrow b^t(h, a') \qquad \forall a' \neq a$
    $\hat{u}_b(h, a|\sigma^t, z^t) \leftarrow b^t(h, a) + \frac{1}{q^t(h,a)}\left(\hat{u}_b((ha)|\sigma^t, z^t) - b^t(h, a)\right)$
    $\hat{u}_b(h|\sigma^t, z^t) \leftarrow \sum_{a'} \sigma^t(h, a')\hat{u}_b(h, a'|\sigma^t, z^t)$
    if $P(h) = 1$ then
        $r^t(I(h), a) \leftarrow \frac{\pi_2^{\sigma^t}(h)}{\pi^{q^t}(h)}\left(\hat{u}_b(h, \cdot|\sigma^t, z^t) - \hat{u}_b(h|\sigma^t, z^t)\right)$
    else if $P(h) = 2$ then
        $r^t(I(h), a) \leftarrow \frac{\pi_1^{\sigma^t}(h)}{\pi^{q^t}(h)}\left(-\hat{u}_b(h, \cdot|\sigma^t, z^t) + \hat{u}_b(h|\sigma^t, z^t)\right)$
    end if
    $R^t(I(h), \cdot) \leftarrow R^{t-1}(I(h), \cdot) + r^t(I(h), \cdot)$
    $\sigma^{t+1}(h, \cdot) \leftarrow$ REGRETMATCHING($R^t(I(h), \cdot)$)
    $b^{t+1}(h, a) \leftarrow$ UPDATEBASELINE($\cdot$)
    return $\hat{u}_b(h|\sigma^t, z^t)$
end function
```

1.2 Proof of Theorem 1

This proof is a simplified version of the proof of Lemma 5 in Schmid et al. [4].

We directly analyze the expectation of the baseline-corrected utility:

$$
\begin{aligned}
&\mathbb{E}_{z^t}\left[\hat{u}_b(h, a|\sigma^t, z^t) \mid z^t \sqsupseteq h\right] \\
&= \Pr\left[(ha) \sqsubseteq z^t \mid h \sqsubseteq z^t\right]\left(\frac{1}{q^t(h, a)}\left(\mathbb{E}_{z^t}\left[\hat{u}_b((ha)|\sigma^t, z^t) \mid z^t \sqsupseteq (ha)\right] - b^t(h, a)\right) + b^t(h, a)\right) \\
&\quad + \Pr\left[(ha) \not\sqsubseteq z^t \mid h \sqsubseteq z^t\right]\left(b^t(h, a)\right) \\
&= q^t(h, a)\left(\frac{1}{q^t(h, a)}\left(\mathbb{E}_{z^t}\left[\hat{u}_b((ha)|\sigma^t, z^t) \mid z^t \sqsupseteq (ha)\right] - b^t(h, a)\right) + b^t(h, a)\right) \\
&\quad + (1 - q^t(h, a))(b^t(h, a)) \\
&= \mathbb{E}_{z^t}\left[\hat{u}_b((ha)|\sigma^t, z^t) \mid z^t \sqsupseteq (ha)\right]
\end{aligned}
$$

We now proceed by induction on the height of (ha) in the tree. If (ha) has height 0, then $(ha) \in Z$ and $\mathbb{E}_{z^t}\left[\hat{u}_b(h, a|\sigma^t, z^t) \mid z^t \sqsupseteq h\right] = \mathbb{E}_{z^t}\left[\hat{u}_b((ha)|\sigma^t, z^t) \mid z^t \sqsupseteq (ha)\right] = u((ha))$ by definition.

For the inductive step, consider arbitrary h, a such that (ha) has height more than 0. We assume that $\mathbb{E}_{z^t}\left[\hat{u}_b(h', a'|\sigma^t, z^t) \mid z^t \sqsupseteq h'\right] = u((h'a')|\sigma^t)$ for all h', a' such that $(h'a')$ has smaller height than (ha). We then have

$$\begin{aligned}
&\mathbb{E}_{z^t}\left[\hat{u}_b(h, a|\sigma^t, z^t) \mid z^t \sqsupseteq h\right] \\
&\quad = \mathbb{E}_{z^t}\left[\hat{u}_b((ha)|\sigma^t, z^t) \mid z^t \sqsupseteq (ha)\right] \\
&\quad = \sum_{a' \in A((ha))} \sigma^t((ha), a')\mathbb{E}_{z^t}\left[\hat{u}_b((ha), a'|\sigma^t, z^t) \mid z^t \sqsupseteq (ha)\right] \\
&\quad = \sum_{a' \in A((ha))} \sigma^t((ha), a')u((haa')|\sigma^t) && \text{by inductive hypothesis} \\
&\quad = u((ha)|\sigma^t) && \text{by definition}
\end{aligned}$$

We are able to apply the inductive hypothesis because (haa') is a suffix of (ha) and thus must have smaller height. The proof follows by induction. □

1.3 Proof of Theorem 2

Before proving Theorem 2, we first examine how the full (trajectory) variance can be decomposed into contributions from individual actions.

Lemma 1. *For any baseline function b^t and any $h \in H$*

$$\begin{aligned}
\mathrm{Var}_{z^t}\left[\hat{u}_b(h|\sigma^t, z^t)|z^t \sqsupseteq h\right] = &\sum_{a \in A(h)} \frac{\left(\sigma^t(h,a)\right)^2}{q^t(h,a)} \mathrm{Var}_{z^t}\left[\hat{u}_b((ha)|\sigma^t, z^t)\right] \\
&+ \mathrm{Var}_a\left[\frac{\sigma^t(h,a)}{q^t(h,a)}\left(u((ha)|\sigma^t) - b^t(h,a)\right)\right]
\end{aligned}$$

Proof. We use the law of total variance, conditioning on which a is sampled at h. This gives us

$$\begin{aligned}
&\mathrm{Var}_{z^t}\left[\hat{u}_b(h|\sigma^t, z^t)|z^t \sqsupseteq h\right] \\
&\quad = \mathbb{E}_a\left[\mathrm{Var}_{z^t}\left[\hat{u}_b(h|\sigma^t, z^t)\big|z^t \sqsupseteq (ha)\right]\right] + \mathrm{Var}_a\left[\mathbb{E}_{z^t}\left[\hat{u}_b(h|\sigma^t, z^t)\big|z^t \sqsupseteq (ha)\right]\right] \qquad (1)
\end{aligned}$$

We analyze each of these terms separately.

First, to analyze the left summand in (1), we note that if $ha \sqsubset z^t$ then by the recursive definition of baseline-corrected values

$$\hat{u}_b(h|\sigma^t, z^t) = \frac{\sigma^t(h,a)}{q^t(h,a)}\hat{u}_b((ha)|\sigma^t, z^t) - \frac{\sigma^t(h,a)}{q^t(h,a)}b^t(h,a) + \sum_{a' \in A(h)} \sigma^t(h,a')b^t(h,a')$$

Only the first term depends on the sampled trajectory z^t, and thus

$$\begin{aligned}
\mathbb{E}_a\left[\mathrm{Var}_{z^t}\left[\hat{u}_b(h|\sigma^t, z^t)\big|z^t \sqsupseteq (ha)\right]\right] &= \mathbb{E}_a\left[\mathrm{Var}_{z^t}\left[\frac{\sigma^t(h,a)}{q^t(h,a)}\hat{u}_b((ha)|\sigma^t, z^t)\bigg|z^t \sqsupseteq (ha)\right]\right] \\
&= \mathbb{E}_a\left[\left(\frac{\sigma^t(h,a)}{q^t(h,a)}\right)^2 \mathrm{Var}_{z^t}\left[\hat{u}_b((ha)|\sigma^t, z^t)\big|z^t \sqsupseteq (ha)\right]\right] \\
&= \sum_{a \in A(h)} \frac{\left(\sigma^t(h,a)\right)^2}{q^t(h,a)} \mathrm{Var}_{z^t}\left[\hat{u}_b((ha)|\sigma^t, z^t)\big|z^t \sqsupseteq (ha)\right]
\end{aligned} \qquad (2)$$

Next, we analyze the inner expectation of the right summand of (1)

$$\begin{aligned}
&\mathbb{E}_{z^t}\left[\hat{u}_b(h|\sigma^t, z^t)\middle|z^t \sqsupseteq (ha)\right] \\
&\quad= \sum_{a'} \sigma^t(h,a')b^t(h,a') + \frac{\sigma^t(h,a)}{q^t(h,a)}\left(\mathbb{E}_{z^t}\left[\hat{u}_b((ha)|\sigma^t, z^t)\right] - b^t(h,a)\right) \\
&\quad= \sum_{a'} \sigma^t(h,a')b^t(h,a') + \frac{\sigma^t(h,a)}{q^t(h,a)}\left(u((ha)|\sigma^t) - b^t(h,a)\right)
\end{aligned}$$

The first term here doesn't depend on the sampled a, giving us

$$\mathrm{Var}_a\left[\mathbb{E}_{z^t}\left[\hat{u}_b(h|\sigma^t, z^t)\middle|(ha) \sqsubseteq z^t\right]\right] = \mathrm{Var}_a\left[\frac{\sigma^t(h,a)}{q^t(h,a)}\left(u((ha)|\sigma^t) - b^t(h,a)\right)\right] \tag{3}$$

Combining (1), (2), and (3) completes the proof. □

Lemma 1 decomposes the variance into a part from the immediately sampled action, and a part from the remainder of the sampled trajectory. We extend this to completely decompose the trajectory variance.

Lemma 2. *For any baseline function* b^t *and any* h, a

$$\mathrm{Var}_{z^t}\left[\hat{u}_b(h|\sigma^t, z^t)|z^t \sqsupseteq h\right] = \sum_{h' \sqsupseteq h} \frac{(\pi^{\sigma^t}(h,h'))^2}{\pi^{q^t}(h,h')}\mathrm{Var}_{a'}\left[\frac{\sigma^t(h',a')}{q^t(h',a')}\left(u((h'a')|\sigma^t) - b^t(h',a')\right)\right]$$

Proof. We proceed by induction on the height of h in the tree. If h has height 0, then $A(h) = \emptyset$, and $\mathrm{Var}_{z^t}\left[\hat{u}_b(h|\sigma^t, z^t)|z^t \sqsupseteq h\right] = 0$. Otherwise, we begin from Lemma 1 and apply the inductive hypothesis for h' with height less than that of h. This gives

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

The lemma follows by induction. □

Proof of Theorem 2. Starting from Lemma 2, we first bound the variance of history values

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

$$
\begin{aligned}
&\leq \sum_{h' \sqsupseteq h} \frac{(\pi^{\sigma^t}(h,h'))^2}{\pi^{q^t}(h,h')} \mathbb{E}_{a'} \left[\left(\frac{\sigma^t(h',a')}{q^t(h',a')} \big(u((h'a')|\sigma^t) - b^t(h',a') \big) \right)^2 \right] \\
&= \sum_{h' \sqsupseteq h} \frac{(\pi^{\sigma^t}(h,h'))^2}{\pi^{q^t}(h,h')} \sum_{a' \in A(h')} \frac{\left(\sigma^t(h',a')\right)^2}{q^t(h',a')} \big(u((h'a')|\sigma^t) - b^t(h',a') \big)^2 \\
&= \sum_{\substack{h' \sqsupseteq h \\ a' \in A(h')}} \frac{(\pi^{\sigma^t}(h,(h'a')))^2}{\pi^{q^t}(h,(h'a'))} \big(u((h'a')|\sigma^t) - b^t(h',a') \big)^2 \qquad (4)
\end{aligned}
$$

We then reformulate the variance of the history action value $\hat{u}_b(h,a|\sigma^t,z^t)$ in terms of the variance of the succeeding history value $\hat{u}_b((ha)|\sigma^t,z^t)$. To do this, we apply the law of total variance conditioning on the random variable $\mathbb{1}((ha) \sqsubseteq z^t)$ which indicates whether a is sampled at h.

$$
\begin{aligned}
&\mathrm{Var}_{z^t}\left[\hat{u}_b(h,a|\sigma^t,z^t) \middle| z^t \sqsupseteq h \right] \\
&\quad = \mathrm{Var}_{z^t}\left[\frac{\mathbb{1}((ha) \sqsubseteq z^t)}{q^t(h,a)} \big(\hat{u}_b((ha)|\sigma^t,z^t) - b^t(h,a) \big) + b^t(h,a) \middle| z^t \sqsupseteq h \right] \\
&\quad = \mathrm{Var}_{z^t}\left[\frac{\mathbb{1}((ha) \sqsubseteq z^t)}{q^t(h,a)} \big(\hat{u}_b((ha)|\sigma^t,z^t) - b^t(h,a) \big) \middle| z^t \sqsupseteq h \right] \\
&\quad = \mathbb{E}\left[\mathrm{Var}_{z^t}\left[\frac{\mathbb{1}((ha) \sqsubseteq z^t)}{q^t(h,a)} \big(\hat{u}_b((ha)|\sigma^t,z^t) - b^t(h,a) \big) \middle| \mathbb{1}((ha) \sqsubseteq z^t) \right] \right] \\
&\qquad\quad + \mathrm{Var}\left[\mathbb{E}_{z^t}\left[\frac{\mathbb{1}((ha) \sqsubseteq z^t)}{q^t(h,a)} \big(\hat{u}_b((ha)|\sigma^t,z^t) - b^t(h,a) \big) \middle| \mathbb{1}((ha) \sqsubseteq z^t) \right] \right] \\
&\quad = \mathbb{E}\left[\frac{\mathbb{1}((ha) \sqsubseteq z^t)}{(q^t(h,a))^2} \mathrm{Var}_{z^t}\left[\hat{u}_b((ha)|\sigma^t,z^t) \middle| \mathbb{1}((ha) \sqsubseteq z^t) \right] \right] \\
&\qquad\quad + \mathrm{Var}\left[\frac{\mathbb{1}((ha) \sqsubseteq z^t)}{q^t(h,a)} \big(u((ha)|\sigma^t) - b^t(h,a) \big) \right] \\
&\quad = \frac{1}{q^t(h,a)} \mathrm{Var}_{z^t}\left[\hat{u}_b((ha)|\sigma^t,z^t) \middle| z^t \sqsupseteq (ha) \right] \\
&\qquad\quad + \frac{1}{(q^t(h,a))^2} \big(u((ha)|\sigma^t) - b^t(h,a) \big)^2 \mathrm{Var}\left[\mathbb{1}((ha) \sqsubseteq z^t) \right] \\
&\quad = \frac{1}{q^t(h,a)} \mathrm{Var}_{z^t}\left[\hat{u}_b((ha)|\sigma^t,z^t) \middle| z^t \sqsupseteq (ha) \right] + \frac{1 - q^t(h,a)}{q^t(h,a)} \big(u((ha)|\sigma^t) - b^t(h,a) \big)^2 \\
&\quad \leq \frac{1}{q^t(h,a)} \Big(\mathrm{Var}_{z^t}\left[\hat{u}_b((ha)|\sigma^t,z^t) \middle| z^t \sqsupseteq (ha) \right] + \big(u((ha)|\sigma^t) - b^t(h,a) \big)^2 \Big) \qquad (5)
\end{aligned}
$$

Combining (4) and (5), we get

$$
\begin{aligned}
&\mathrm{Var}_{z^t}\left[\hat{u}_b(h,a|\sigma^t,z^t) \middle| z^t \sqsupseteq h \right] \\
&\quad \leq \frac{1}{q^t(h,a)} \Bigg(\sum_{\substack{h' \sqsupseteq (ha) \\ a' \in A(h')}} \frac{(\pi^{\sigma^t}((ha),(h'a')))^2}{\pi^{q^t}((ha),(h'a'))} \big(u((h'a')|\sigma^t) - b^t(h',a') \big)^2 \\
&\qquad\qquad + \big(u((ha)|\sigma^t) - b^t(h,a) \big)^2 \Bigg) \\
&\quad = \frac{1}{q^t(h,a)} \sum_{(h'a') \sqsupseteq (ha)} \frac{(\pi^{\sigma^t}((ha),(h'a')))^2}{\pi^{q^t}((ha),(h'a'))} \big(u((h'a')|\sigma^t) - b^t(h',a') \big)^2 \\
&\quad = \sum_{(h'a') \sqsupseteq (ha)} \frac{(\pi^{\sigma^t}((ha),(h'a')))^2}{\pi^{q^t}(h,(h'a'))} \big(u((h'a')|\sigma^t) - b^t(h',a') \big)^2
\end{aligned}
$$

□

1.4 Public trees

There are multiple sources of variance when computing the regret at an information set in MCCFR. One form of variance comes from sampling actions (and recursively, trajectories) from the information set, rather than walking the full subtree. A second form of variance comes from sampling only one of the histories in the information set itself. Our baseline framework reduces the first kind of variance, but does not take the second form of variance into account.

One approach to combating this single-history variance could be to extend the use of the baseline; analogous to how we created a control variate from using $b^t(h, a)$ to evaluate unsampled actions a, we could also create a control variate that uses $b^t(h', a)$ to evaluate all unsampled $h' \in I(h)$. However, this requires evaluating alternate histories along every step of the sampled trajectory, meaning that a single iteration of MCCFR goes from complexity $\mathcal{O}(d|A_{\max}|)$ to $\mathcal{O}(d|A_{\max}||I_{\max})|$.

A second approach, and the one we present in this section, is to change the sampling method used. Rather than using a baseline to consider each alternate history in the information set, we directly evaluate all such histories. Intuitively, this can be done by only sampling actions that are publicly observable, and walking all actions that change the game's hidden state. This approach was used by Schmid et al. [4], but was never formalized. We formalize the algorithm here, after presenting some additional assumptions and definitions.

We assume that the EFG is *timeable* [1], which informally means that no player can gain additional information by tracking how much time elapses while they are not acting. Formally, this means that we can assign a value time(h) to every $h \in H$ such that time$(h) =$ time(h') for any $h' \in I_{P(h)}(h)$, and time$(h) <$ time(h') for any $h' \sqsupseteq h$. Every game played by humans must be timeable, or else the human could distinguish histories in the same information set by tracking elapsed time. If a game is timeable, players always observe the timing when they are acting, so there must be some strategically identical game where they observe the timing even when not acting. Thus we will assume that our games satisfy this requirement.

We now introduce the concept of a *public state* [2], which groups histories based on information available to all players, or informally, based on whether they distinguishable to an outside observer. Formally, a public state is a set of histories that is (minimally) closed under the information set relation for all players. Let $\mathcal{S}$ be the set of public states (which partitions H), and $S(h) \in \mathcal{S}$ be the public state that h belongs to. By assumption that all players observe the game's timing, necessarily time$(h) =$ time(h') if $S(h) = S(h')$. In turn, this means that if $h \sqsubseteq h'$, then $S(h) \neq S(h')$. We also assume for simplicity that if $S(h) = S(h')$, then $P(h) = P(h')$. If necessary, this can be made true for any timeable game by splitting information sets and adding dummy actions, without strategically changing the game.

We define $\mathcal{T}(S)$ to be the set of successor public states to $\mathcal{S}$: $S' \in \mathcal{T}(S)$ if there is some $h \in S$, $a \in A(h)$, and $h' \in S'$ such that $(ha) = h'$. The successor relation defines the edges of a *public tree*, where the public states are nodes. It should be noted that more than one action can lead to the same successor public state when some player doesn't observe the action, and that one action can lead to more than one successor public state if some previously private information becomes public.

In the statement of Theorem 3, we used samp$^t(h)$ to notate whether h was sampled on iteration t. With the notation introduced here, we can formalize this by defining samp$^t(h)$ to occur if and only if $h' \sqsubseteq z^t$ for some $h' \in S(h)$ and some $z^t \in Z^t$. For clarity, we thus symbolize this relation as $S(h) \sqsubseteq Z^t$.

1.4.1 Public Outcome Sampling

We now define our MCCFR variant, which we call *Public Outcome Sampling (POS)*. Instead of walking trajectories through the EFG tree by sampling actions, POS walks trajectories through the public tree by sampling successor public states.

For public state S, let $\mathcal{I}_i(S) \subseteq \mathcal{I}_i$ be the collection of player i information sets contained within S. While walking down the tree, POS keeps track of reach probabilities $\pi_i^{\sigma^t}(I_i)$ for each $I_i \in \mathcal{I}_i(S)$ and each player i at public state S. To recurse, it samples some successor $S' \in \mathcal{T}(S)$ using a probability distribution $q^t(S) \in \Delta_{\mathcal{T}(S)}$. It updates the reach probabilities to $\pi_i^{\sigma^t}(I_i)$ for each $I_i \in \mathcal{I}_i(S')$, using the current strategy σ^t. Ultimately, the recursion reaches a public state which only contains terminal nodes (as the end of the game is publicly observable). This public state, which defines the sampled trajectory in the public tree, we label Z^t. The terminal histories are evaluated as $u(z)$ for each $z \in Z^t$.

Walking back up the tree, at each recursion step we pass back the utilities $\hat{u}_b(h'|\sigma^t, Z^t)$ for each $h' \in S'$. From these, we apply a baseline and recursively calculate utilities as

$$\hat{u}_b(h, a|\sigma^t, Z^t) = \frac{\mathbb{1}(S((ha)) \sqsubseteq Z^t)}{q^t(S(h), S((ha))} \left(\hat{u}_b((ha)|\sigma^t, Z^t) - b^t(h, a)\right) + b^t(h, a) \tag{6}$$

$$\hat{u}_b(h|\sigma^t, Z^t) = \sum_{a \in A(h)} \sigma^t(h, a)\hat{u}_b(h, a|\sigma^t, Z^t) \tag{7}$$

for each $h \in S$ and $a \in A(h)$. We then use these values to calculate regrets $r^t(I, a)$ for each $I \in \mathcal{I}_i(S)$ and update the saved regrets.

Algorithm 2 gives pseudocode for MCCFR with POS.

Updating a public state S with this algorithm requires walking through all of the possible histories in the public state, as well as all of the actions possible at each history, giving a complexity $\mathcal{O}(|S||A_{\text{max}}|)$, or equivalently $\mathcal{O}(|\mathcal{I}_i(S)||\mathcal{I}_{-i}(S)||A_{\text{max}}|)$. However, the computations for each information set $I \in \mathcal{I}_i(S)$ with acting player i can be done completely independently, allowing for easy parallelization (e.g. on a GPU) to achieve complexity $\mathcal{O}(|\mathcal{I}_{-i}(S)||A_{\text{max}}|)$. This approach was taken with the non-sampling algorithm used in DeepStack [3].

1.5 Proof of Theorem 3

We begin by proving the equivalence of the two possible definitions for the predictive baseline. This lemma is independent of sampling scheme.

Lemma 3. *Let h, a be such that $(ha) \sqsubseteq z^t$, and define b^{t+1} according to the predictive baseline update (equation (7) in the main paper). Then we have that*

$$b^{t+1}(h, a) = \hat{u}_b((ha)|\sigma^{t+1}, z^t).$$

Proof. We prove this by induction on the tree. Our base case is that $(ha) = z^t$, in which case by definition $b^{t+1}(h, a) = u(z^t) = \hat{u}_b((ha)|\sigma^{t+1}, z^t)$.

For the inductive step, assume that the statement holds for all $(h'a') \sqsupset (ha)$ such that $(h'a') \sqsubseteq z^t$. Consider some arbitrary $(h'a') \sqsupset (ha)$. If $(h'a') \sqsubseteq z^t$, then the inductive hypothesis holds and we have

$$\begin{aligned}
\hat{u}_b(h', a'|\sigma^{t+1}, z^t) &= \frac{\mathbb{1}((h'a') \sqsubseteq z^t)}{q^t(h', a')} \left(\hat{u}_b((h'a')|\sigma^{t+1}, z^t) - b^{t+1}(h', a')\right) + b^{t+1}(h', a') \\
&= \frac{\mathbb{1}((h'a') \sqsubseteq z^t)}{q^t(h', a')} \left(b^{t+1}(h', a') - b^{t+1}(h', a')\right) + b^{t+1}(h', a') \\
&= b^{t+1}(h', a')
\end{aligned}$$

On the other hand, if $(h'a') \not\sqsubseteq z^t$, then

$$\begin{aligned}
\hat{u}_b(h', a'|\sigma^{t+1}, z^t) &= \frac{\mathbb{1}((h'a') \sqsubseteq z^t)}{q^t(h', a')} \left(\hat{u}_b((h'a')|\sigma^{t+1}, z^t) - b^{t+1}(h', a')\right) + b^{t+1}(h', a') \\
&= \frac{0}{q^t(h', a')} \left(\hat{u}_b((h'a')|\sigma^{t+1}, z^t) - b^{t+1}(h', a')\right) + b^{t+1}(h', a') \\
&= b^{t+1}(h', a')
\end{aligned}$$

Thus, either way $\hat{u}_b(h', a'|\sigma^{t+1}, z^t) = b^{t+1}(h', a')$ for any $(h'a') \sqsupset (ha)$, which gives us

$$\begin{aligned}
b^{t+1}(h, a) &= \sum_{a' \in A((h,a))} \sigma^{t+1}((ha), a') b^{t+1}((ha), a') \\
&= \sum_{a' \in A((h,a))} \sigma^{t+1}((ha), a') \hat{u}_b((ha), a'|\sigma^{t+1}, z^t) \\
&= \hat{u}_b((ha)|\sigma^{t+1}, z^t)
\end{aligned}$$

□

Next, we show that the recursive definition of the predictive baseline is maintained as an invariant under POS.

Lemma 4. *After t iterations of POS updates, for any non-terminal h, a the predictive baseline satisfies*

$$b^{t+1}(h, a) = \sum_{a'} \sigma^{t+1}((ha), a') b^{t+1}((ha), a') \tag{8}$$

Proof. If (ha) is sampled on iteration t, then the statement holds trivially from the predictive baseline definition. We thus assume that $S((ha)) \not\sqsubseteq Z^t$, and proceed by induction on time.

For our base case, $t = 0$, we have that $b^1(h, a) = 0$ for all h, a.

For the inductive step, we assume that the lemma holds for time t, and we show that it then follows for time $t + 1$. Because we assume $S((ha)) \not\sqsubseteq Z^t$, we have that $b^{t+1}(h, a) = b^t(h, a)$ by definition of the predictive baseline. In addition, we must have that $S((haa')) \not\sqsubseteq Z^t$ for all a' because $S((ha)) \sqsubseteq S((haa'))$, so $b^{t+1}((ha), a') = b^t((ha), a')$. Thus

$$\begin{aligned}
b^{t+1}(h, a) &= b^t(h, a) && \text{by predictive baseline definition} \\
&= \sum_{a' \in A((ha)} \sigma^t((ha), a') b^t((ha), a') && \text{by inductive hypothesis} \\
&= \sum_{a' \in A((ha)} \sigma^t((ha), a') b^{t+1}((ha), a')
\end{aligned}$$

This completes the inductive step and the lemma follows. □

We now introduce the following definition that tracks sampled values of terminal histories:

$$\tilde{u}^t(z) = \begin{cases} u(z) & \text{if } z \in Z^\tau \text{ for any } \tau < t \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

We show that the invariant maintained under POS is an expectation over these values.

Lemma 5. *After t iterations of POS updates, for all h, a the predictive baseline satisfies*

$$b^{t+1}(h, a) = \sum_{z \in Z[(ha)]} \pi^{\sigma^{t+1}}((ha), z) \tilde{u}^{t+1}(z) \tag{10}$$

Proof. We proceed by induction on the tree. For base case, we have $(ha) = z$ for some $z \in Z$. If z has been sampled, then $b^{t+1}(h, a) = u(z) = \tilde{u}^{t+1}(z)$ by definition of the predictive baseline. If it hasn't been sampled, then $b^{t+1}(h, a) = 0 = \tilde{u}^{t+1}(z)$.

For the inductive step, we assume the lemma holds for all $(h'a') \sqsupset (ha)$. Then we have that

$$\begin{aligned}
&b^{t+1}(h,a) \\
&\quad= \sum_{a' \in A((ha))} \sigma^{t+1}((ha),a') b^{t+1}((ha),a') && \text{by Lemma 4} \\
&\quad= \sum_{a' \in A((ha))} \sigma^{t+1}((ha),a') \sum_{z \in Z[(haa')]} \pi^{\sigma^{t+1}}((haa'),z) \tilde{u}^{t+1}(z) && \text{by inductive hypothesis} \\
&\quad= \sum_{a' \in A((ha))} \sum_{z \in Z[(haa')]} \pi^{\sigma^{t+1}}((ha),z) \tilde{u}^{t+1}(z) \\
&\quad= \sum_{z \in Z[(ha)]} \pi^{\sigma^{t+1}}((ha),z) \tilde{u}^{t+1}(z)
\end{aligned}$$

□

To prove Theorem 3, we note that if $Z[h] \subseteq \bigcup_{\tau < t} Z^\tau$, then by definition $\tilde{u}^t(z) = u(z)$ for any $z \in Z[h]$. Thus we have

$$\begin{aligned}
b^t(h,a) &= \sum_{z \in Z[(ha)]} \pi^{\sigma^t}((ha),z) \tilde{u}^t(z) && \text{by Lemma 5} \\
&= \sum_{z \in Z[(ha)]} \pi^{\sigma^t}((ha),z) u(z) \\
&= u((ha)|\sigma^t) && \text{by definition}
\end{aligned}$$

□

1.6 Baselines in Monte Carlo continual resolving

Figure 1 is an expanded version of Figure 5 from the main paper, showing results for MCCR with additional baseline functions.

References

[1] Sune K. Jakobsen, Troels B. Sørensen, and Vincent Conitzer. Timeability of extensive-form games. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, 2016.

[2] Michael Johanson, Kevin Waugh, Michael Bowling, , and Martin Zinkevich. Accelerating best response calculation in large extensive games. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[3] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael H. Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356 6337:508–513, 2017.

[4] Martin Schmid, Neil Burch, Marc Lanctot, Matej Moravcik, Rudolf Kadlec, and Michael Bowling. 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.

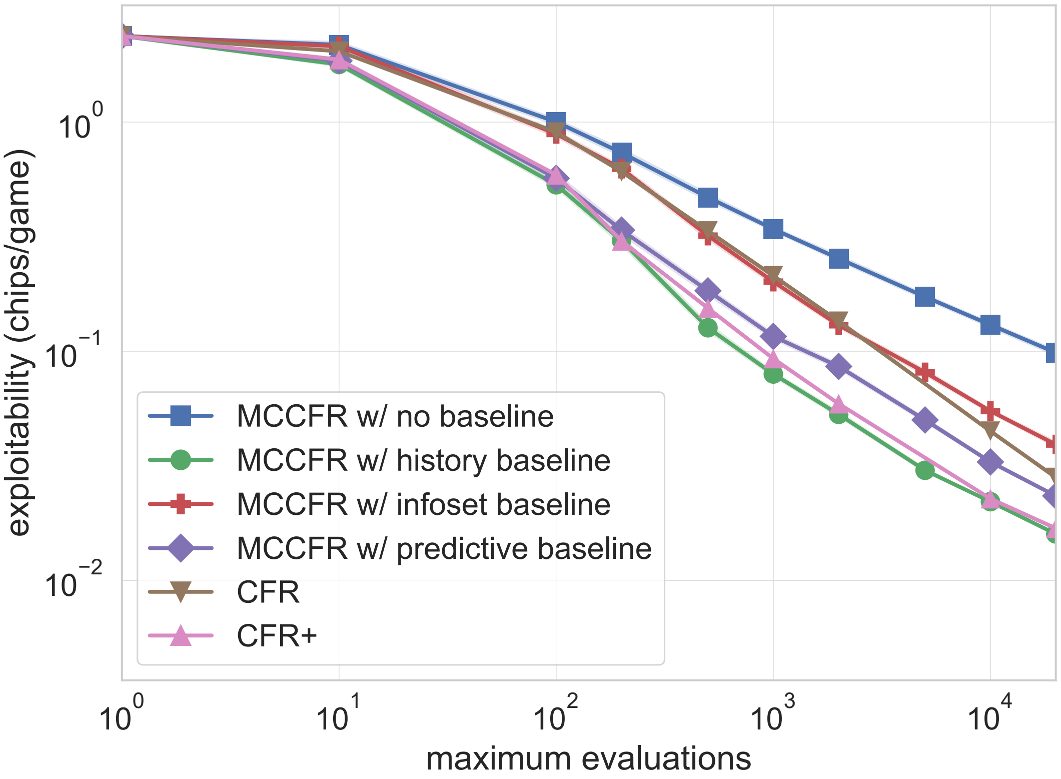

Figure 1: Exploitability of continual resolving strategies based on the maximum number of evaluations allowed per resolve.

Algorithm 2 MCCFR w/ POS and baseline

```
function POS-MCCFR($S$)
    if $S \subseteq Z$ then return $\{u(h) \mid \forall h \in S\}$
    for $I \in \mathcal{I}_{P(S)}(S)$ do
        $\overline{\sigma}^t(I,\cdot) \leftarrow \frac{t-1}{t}\overline{\sigma}^{t-1}(I,\cdot) + \frac{1}{t}\sigma^t$
    end for
    sample successor $S' \sim q^t(S,\cdot)$
    $\{\hat{u}_b(h'|\sigma^t,Z^t) \mid \forall h' \in S'\} \leftarrow$ POS-MCCFR($S'$)
    for $I \in \mathcal{I}_{P(S)}$ do
        for $h \in I$ do
            for $a \in A(h)$ do
                if $(ha) \in S'$ then
                    $\hat{u}_b(h,a|\sigma^t,Z^t) \leftarrow b^t(h,a) + \frac{1}{q^t(S,S')}\left(\hat{u}_b((ha)|\sigma^t,Z^t) - b^t(h,a)\right)$
                else
                    $\hat{u}_b(h,a|\sigma^t,Z^t) \leftarrow b^t(h,a)$
                end if
            end for
            $\hat{u}_b(h|\sigma^t,Z^t) \leftarrow \sum_{a'}\sigma^t(h,a')\hat{u}_b(h,a'|\sigma^t,Z^t)$
        end for
        if $P(h) = 1$ then
            $r^t(I,a) \leftarrow \frac{1}{\pi^{q^t}(S)}\sum_{h \in I}\pi_2^{\sigma^t}(h)\left(\hat{u}_b(h,\cdot|\sigma^t,Z^t) - \hat{u}_b(h|\sigma^t,Z^t)\right)$
        else if $P(h) = 2$ then
            $r^t(I,a) \leftarrow \frac{1}{\pi^{q^t}(S)}\sum_{h \in I}\pi_1^{\sigma^t}(h)\left(-\hat{u}_b(h,\cdot|\sigma^t,Z^t) + \hat{u}_b(h|\sigma^t,Z^t)\right)$
        end if
        $R^t(I,\cdot) \leftarrow R^{t-1}(I,\cdot) + r^t(I,\cdot)$
    end for
    for $I \in \mathcal{I}_{P(S)}(S)$ do
        $\sigma^{t+1}(I,\cdot) \leftarrow \textsc{RegretMatching}(R^t(I,\cdot))$
    end for
    for $I \in \mathcal{I}_{P(S)}$ do
        for $h \in I$ do
            for $a \in A(h)$ do
                if $(ha) \in S'$ then
                    $b^{t+1}(h,a) \leftarrow \textsc{UpdateBaseline}(\cdot)$
                end if
            end for
        end for
    end for
    return $\{\hat{u}_b(h|\sigma^t,Z^t) \mid \forall h \in S\}$
end function
```