A. Rules for Heads-Up Limit Texas Hold'em and Flop Hold'em Poker

Heads-up limit Texas hold'em is a two-player zero-sum game. There are two players and the position of the two players alternate after each hand. On each betting round, each player can choose to either fold, call, or raise. Folding results in the player losing and the money in the pot being awarded to the other player. Calling means the player places a number of chips in the pot equal to the opponent's share. Raising means that player adds more chips to the pot than the opponent's share. A round ends when a player calls (if both players have acted). There cannot be more than three raises in the first or second betting round or more than four raises in the third or fourth betting round, so there is a limited number of actions in the game. Raises in the first two rounds are \$100 and raises in the second two rounds are \$200.

At the start of each hand of HULH, both players are dealt two private cards from a standard 52-card deck. P_1 must place \$50 in the pot and P_2 must place \$100 in the pot. A round of betting then occurs starting with P_1. When the round ends, three *community* cards are dealt face up that both players can ultimately use in their final hands. Another round of betting occurs, starting with P_2 this time. Afterward another community card is dealt face up and another betting round occurs. Then a final card is dealt face up and a final betting round occurs. At the end of the betting round, unless a player has folded, the player with the best five-card poker hand constructed from their two private cards and the five community cards wins the pot. In the case of a tie, the pot is split evenly.

Flop Hold'em Poker is identical to HULH except there are only the first two betting rounds.

B. Proofs of Theorems

B.1. Review of MCCFR

We begin by reviewing the derivation of convergence bounds for external sampling MCCFR from Lanctot et al. 2009.

An MCCFR scheme is completely specified by a set of *blocks* $\mathcal{Q} = \{Q_i\}$ which each comprise a subset of all terminal histories Z. On each iteration MCCFR samples one of these blocks, and only considers terminal histories within that block. Let $q_j > 0$ be the probability of considering block Q_j in an iteration.

Let Z_I be the set of terminal nodes that contain a prefix in I, and let $z[I]$ be that prefix. Define $\pi^\sigma(h \to z)$ as the probability of playing to z given that player p is at node h with both players playing σ.

$$\pi^\sigma(h \to z) = \sum_{z \in Z_I} \frac{\pi^\sigma(z[I])}{\pi^\sigma(I)} \pi^\sigma(z).$$

$\pi^\sigma(I \to z)$ is undefined when $\pi(I) = 0$.

Let $q(z) = \sum_{j: z \in Q_j} q_j$ be the probability that terminal history z is sampled in an iteration of MCCFR. For external sampling MCCFR, $q(z) = \pi^\sigma_{-i}(z)$.

The *sampled value* $\tilde{v}^\sigma_i(I|j)$ when sampling block j is

$$\tilde{v}^\sigma_p(I|j) = \sum_{z \in Q_j \cap Z_I} \frac{1}{q(z)} u_p(z) \pi^\sigma_{-p}(z[I]) \pi^\sigma(z[I] \to z) \tag{6}$$

For external sampling, the sampled value reduces to

$$\tilde{v}^\sigma_p(I|j) = \sum_{z \in Q_j \cap Z_I} u_p(z) \pi^\sigma_p(z[I] \to z) \tag{7}$$

The sampled value is an unbiased estimator of the true value $v_p(I)$. Therefore the *sampled instantaneous regret* $\tilde{r}^t(I, a) = \tilde{v}^{\sigma^t}_p(I, a) - \tilde{v}^{\sigma^t}_p(I)$ is an unbiased estimator of $r^t(I, a)$.

The *sampled regret* is calculated as $\tilde{R}^T(I, a) = \sum_{t=1}^T \tilde{r}^t(I, a)$.

We first state the general bound shown in (Lanctot, 2013), Theorem 3.

Lanctot 2013 defines $\mathcal{B}_p$ to be a set with one element per distinct action sequence $\vec{a}$ played by p, containing all infosets that may arise when p plays $\vec{a}$. M_p is then defined by $\sum_{B \in \mathcal{B}_p} |B|$. Let Δ be the difference between the maximum and minimum payoffs in the game.

Theorem 2. *(Lanctot 2013, Theorem 3) For any $p \in (0, 1]$, when using any algorithm in the MCCFR family such that for all $Q \in \mathcal{Q}$ and $B \in \mathcal{B}_p$,*

$$\sum_{I \in B} \left(\sum_{z \in Q \cap Z_I} \frac{\pi^\sigma(z[I] \to z) \pi^\sigma_{-p}(z[I])}{q(z)} \right)^2 \leq \frac{1}{\delta^2} \tag{8}$$

where $\delta \leq 1$, then with probability at least $1 - \rho$, total regret is bounded by

$$R_p^T \leq \left(M_p + \frac{\sqrt{2|\mathcal{I}_p||\mathcal{B}_p|}}{\sqrt{\rho}} \right) \left(\frac{1}{\delta} \right) \Delta \sqrt{|A|T} \tag{9}$$

.

For the case of external sampling MCCFR, $q(z) = \pi^\sigma_{-i}(z)$. Lanctot et al. 2009, Theorem 9 shows that for external sampling, for which $q(z) = \pi^\sigma_{-i}(z)$, the inequality in (8) holds for $\delta = 1$, and thus the bound implied by (9) is

$$\bar{R}_p^T \leq \left(M_p + \frac{\sqrt{2|\mathcal{I}_p||\mathcal{B}_p|}}{\sqrt{\rho}} \right) \Delta \frac{\sqrt{|A|}}{\sqrt{T}} \tag{10}$$

$$\leq \left(1 + \frac{\sqrt{2}}{\sqrt{\rho K}} \right) \Delta |\mathcal{I}_p| \frac{\sqrt{|A|}}{\sqrt{T}} \qquad \text{because } |\mathcal{B}_p| \leq M_p \leq |\mathcal{I}_p| \tag{11}$$

.

B.2. Proof of Lemma 1

We show

$$\mathbb{E}_{Q_j \sim \mathcal{Q}} \left[\tilde{v}_p^{\sigma^t}(I) \middle| Z_I \cap Q_j \neq \emptyset \right] = v^{\sigma^t}(I) / \pi^{\sigma^t}_{-p}(I).$$

Let $q_j = P(Q_j)$.

$$\begin{aligned}
\mathbb{E}_{Q_j \sim \mathcal{Q}} \left[\tilde{v}_p^{\sigma^t}(I) \middle| Z_I \cap Q \neq \emptyset \right] &= \frac{\mathbb{E}_{Q_j \sim \mathcal{Q}} \left[\tilde{v}_p^{\sigma^t}(I) \right]}{P_{Q_j \sim \mathcal{Q}}(Z_I \cap Q_j \neq \emptyset)} \\
&= \frac{\sum_{Q_j \in \mathcal{Q}} q_j \sum_{z \in Z_I \cap Q_j} u_p(z) \pi^{\sigma^t}_{-p}(z[I]) \pi^{\sigma^t}(z[I] \to z) / q(z)}{\pi^{\sigma^t}_{-p}(I)} \\
&= \frac{\sum_{z \in Z_I \cap Q_j} \left(\sum_{Q_j : z \in Q_j} q_j \right) u_p(z) \pi^{\sigma^t}_{-p}(z[I]) \pi^{\sigma^t}(z[I] \to z) / q(z)}{\pi^{\sigma^t}_{-p}(I)} \\
&= \frac{\sum_{z \in Z_I} q(z) u_p(z) \pi^{\sigma^t}_{-p}(z[I]) \pi^{\sigma^t}(z[I] \to z) / q(z)}{\pi^{\sigma^t}_{-p}(I)} \qquad \text{By definition of } q(z) \\
&= \frac{v^{\sigma^t}(I)}{\pi^{\sigma^t}_{-p}(I)}
\end{aligned}$$

The result now follows directly.

B.3. K-external sampling

We first show that performing MCCFR with K external sampling traversals per iteration (K-ES) shares a similar convergence bound with standard external sampling (i.e. 1-ES). We will refer to this result in the next section when we consider the full

Deep CFR algorithm. This convergence bound is rather obvious and the derivation pedantic, so the reader is welcome to skip this section.

We model T rounds of K-external sampling as $T \times K$ rounds of external sampling, where at each round $t \cdot K + d$ (for integer $t \geq 0$ and integer $0 \leq d < K$) we play

$$\sigma_{tK+d}(a) = \begin{cases} \frac{R^+_{tK}(a)}{R^+_{\Sigma,tK}} \text{ if } R^+_{\Sigma,tK} > 0 \\ \text{arbitrary, otherwise} \end{cases} \tag{12}$$

In prior work, σ is typically defined to play $\frac{1}{|A|}$ when $R^+_{\Sigma,T}(a) \leq 0$, but in fact the convergence bounds do not constraint σ's play in these situations, which we will demonstrate explicitly here. We need this fact because minimizing the loss $\mathcal{L}(V)$ is defined only over the samples of (visited) infosets and thus does not constrain the strategy in unvisited infosets.

Lemma 2. *If regret matching is used in K-ES, then for $0 \leq d < K$*

$$\sum_{a \in A} R^+_{tK}(a) r_{tK+d}(a) \leq 0 \tag{13}$$

Proof. If $R^+_{\Sigma,tK} \leq 0$, then $R^+_{tK}(a) = 0$ for all a and the result follows directly. For $R^+_{\Sigma,tK} > 0$,

$$\begin{aligned}
\sum_{a \in A} R^+_{tK}(a) r_{tK+d}(a) &= \sum_{a \in A} R^+_T(a)(u_{tK+d}(a) - u_{tK+d}(\sigma_{tK})) && (14)\\
&= \left(\sum_{a \in A} R^+_{tK}(a) u_{tK+d}(a)\right) - \left(u_{tK+d}(\sigma_{tK}) \sum_{a \in A} R^+_{tK}(a)\right) && (15)\\
&= \left(\sum_{a \in A} R^+_{tK}(a) u_{tK+d}(a)\right) - \left(\sum_{a \in A} \sigma_{tK+d}(a) u_{tK+d}(a)\right) R^+_{\Sigma,tK}(a) && (16)\\
&= \left(\sum_{a \in A} R^+_{tK}(a) u_{tK+d}(a)\right) - \left(\sum_{a \in A} \frac{R^+_{tK}(a)}{R^+_{\Sigma,tK}(a)} u_{tK+d}(a)\right) R^+_{\Sigma,tK}(a) && (17)\\
&= \left(\sum_{a \in A} R^+_{tK}(a) u_{tK+d}(a)\right) - \left(\sum_{a \in A} R^+_{tK}(a)(a) u_{tK+d}(a)\right) && (18)\\
&= 0 && (19)
\end{aligned}$$

□

Theorem 3. *Playing according to Equation 12 guarantees the following bound on total regret*

$$\sum_{a \in A} (R^+_{TK}(a))^2 \leq |A| \Delta^2 K^2 T \tag{20}$$

Proof. We prove by recursion on T.

$$\begin{aligned}
\sum_{a \in A} (R^+_{TK}(a))^2 &\leq \sum_{a \in A} \left(R^+_{(T-1)K}(a) + \sum_{d=0}^{K-1} r_{tK-d}(a)\right)^2 && (21)\\
&= \sum_{a \in A} \left(R^+_{(T-1)K}(a)^2 + 2 \sum_{d=0}^{K-1} r_d(a) R^+_{(T-1)K}(a) + \sum_{d=0}^{K-1} \sum_{d'=0}^{K-1} r_{TK-d}(a) r_{TK-d'}(a)\right) && (22)
\end{aligned}$$

By Lemma 2,

$$\sum_{a\in A}(R^+_{TK}(a))^2 \leq \sum_{a\in A}(R^+_{(T-1)K}(a))^2 + \sum_{a\in A}\sum_{d=0}^{K-1}\sum_{d'=0}^{K-1} r_{TK-d}(a)r_{TK-d'}(a) \tag{23}$$

By induction,

$$\sum_{a\in A}(R^+_{(T-1)K}(a))^2 \leq |A|\Delta^2(T-1) \tag{24}$$

From the definition, $|r_{TK-d}(a)| \leq \Delta$

$$\sum_{a\in A}(R^+_{TK}(a))^2 \leq |A|\Delta^2(T-1) + K^2|A|\Delta^2 = |A|\Delta^2K^2T \tag{25}$$

□

Theorem 4. *(Lanctot 2013, Theorem 3 & Theorem 5) After T iterations of K-ES, average regret is bounded by*

$$\bar{R}^{TK}_p \leq \left(1 + \frac{\sqrt{2}}{\sqrt{\rho K}}\right)|\mathcal{I}_p|\Delta\frac{\sqrt{|A|}}{\sqrt{T}} \tag{26}$$

with probability $1-\rho$.

Proof. The proof follows Lanctot 2013, Theorem 3. Note that K-ES is only different from ES in terms of the choice of σ_T, and the proof in Lanctot 2013 only makes use of σ_T via the bound on $(\sum_a R^T_+(a))^2$ that we showed in Theorem 3. Therefore, we can apply the same reasoning to arrive at

$$\tilde{R}^{TK}_p \leq \frac{\Delta M_p\sqrt{|A|T}K}{\delta} \tag{27}$$

(Lanctot 2013, Eq. (4.30)).

Lanctot et al. 2009 then shows that $\tilde{R}^{TK}_p$ and R^{TK}_p are similar with high probability, leading to

$$\mathbb{E}\left[\left(\sum_{I\in\mathcal{I}_p}(R^{TK}_p(I) - \tilde{R}^{TK}_p(I))\right)^2\right] \leq \frac{2|\mathcal{I}_p||\mathcal{B}_p||A|TK\Delta^2}{\delta^2} \tag{28}$$

(Lanctot 2013, Eq. (4.33), substituting $T \to TK$).

Therefore, by Markov's inequality, with probability at least $1-\rho$,

$$R^{TK}_p \leq \frac{\sqrt{2|\mathcal{I}_p||\mathcal{B}_p||A|TK}\Delta}{\delta\sqrt{\rho}} + \frac{\Delta M\sqrt{|A|T}K}{\delta} \tag{29}$$

, where external sampling permits $\delta = 1$ (Lanctot, 2013).

Using the fact that $M \leq |\mathcal{I}_p|$ and $|\mathcal{B}_p| < |\mathcal{I}_p|$ and dividing through by KT leads to the simplified form

$$\bar{R}^{TK}_p \leq \left(1 + \frac{\sqrt{2}}{\sqrt{\rho K}}\right)\Delta|\mathcal{I}_p|\frac{\sqrt{|A|}}{\sqrt{T}} \tag{30}$$

with probability $1-\rho$.

□

We point out that the convergence of K-ES is faster as K increases (up to a point), but it still requires the same order of iterations as ES.

B.4. Proof of Theorem 1

Proof. Assume that an online learning scheme plays

$$\sigma^t(I,a) = \begin{cases} \frac{y_+^t(I,a)}{\sum_a y_+^t(I,a)} \text{ if } \sum_a y_+^t(I,a) > 0 \\ \text{arbitrary, otherwise} \end{cases} . \tag{31}$$

Morrill 2016, Corollary 3.0.6 provides the following bound on the total regret as a function of the L2 distance between y_t^+ and $R^{T,+}$ at each infoset.

$$\max_{a\in A}(R^T(I,a))^2 \leq |A|\Delta^2 T + 4\Delta|A| \sum_{t=1}^{T}\sum_{a\in A}\sqrt{(R_+^t(I,a) - y_+^t(I,a))^2} \tag{32}$$

$$\leq |A|\Delta^2 T + 4\Delta|A| \sum_{t=1}^{T}\sum_{a\in A}\sqrt{(R^t(I,a) - y^t(I,a))^2} \tag{33}$$

Since $\sigma^t(I,a)$ from Eq. 31 is invariant to rescaling across all actions at an infoset, it's also the case that for any $C(I) > 0$

$$\max_{a\in A}(R^T(I,a))^2 \leq |A|\Delta^2 T + 4\Delta|A| \sum_{t=1}^{T}\sum_{a\in A}\sqrt{(R^t(I,a) - C(I)y^t(I,a))^2} \tag{34}$$

Let $x^t(I)$ be an indicator variable that is 1 if I was traversed on iteration t. If I was traversed then $\tilde{r}^t(I)$ was stored in $M_{V,p}$, otherwise $\tilde{r}^t(I) = 0$. Assume for now that $\mathcal{M}_{V,p}$ is not full, so all sampled regrets are stored in the memory.

Let $\Pi^t(I)$ be the fraction of iterations on which $x^t(I) = 1$, and let

$$\epsilon^t(I) = \left\| \mathbb{E}_t\left[\tilde{r}^t(I) | x^t(I) = 1\right] - V(I, a|\theta^t) \right\|_2 .$$

Inserting canceling factors of $\sum_{t'=1}^{t} x^{t'}(I)$ and setting $C(I) = \sum_{t'=1}^{t} x^{t'}(I)$,[7]

$$\max_{a\in A}(\tilde{R}^T(I,a))^2 \leq |A|\Delta^2 T + 4\Delta|A| \sum_{t=1}^{T}\left(\sum_{t'=1}^{t} x^{t'}(I)\right)\sum_{a\in A}\sqrt{\left(\frac{\tilde{R}^t(I,a)}{\sum_{t'=1}^{t} x^{t'}(I)} - y^t(I,a)\right)^2} \tag{35}$$

$$= |A|\Delta^2 T + 4\Delta|A| \sum_{t=1}^{T}\left(\sum_{t'=1}^{t} x^{t'}(I)\right)\left\| \mathbb{E}_t\left[\tilde{r}^t(I) | x^t(I) = 1\right] - V(I, a|\theta^t) \right\|_2 \tag{36}$$

$$= |A|\Delta^2 T + 4\Delta|A| \sum_{t=1}^{T} t\Pi^t(I)\epsilon^t(I) \quad \text{by definition} \tag{37}$$

$$\leq |A|\Delta^2 T + 4\Delta|A| T \sum_{t=1}^{T} \Pi^t(I)\epsilon^t(I) \tag{38}$$

$$\tag{39}$$

The first term of this expression is the same as Theorem 3, while the second term accounts for the approximation error.

[7] The careful reader may note that $C(I) = 0$ for unvisited infosets, but $\sigma^t(I,a)$ can play an arbitrary strategy at these infosets so it's okay.

In the case of K-external sampling, the same derivation as shown in Theorem 3 leads to

$$\max_{a \in A} (\tilde{R}^T(I,a))^2 \leq |A|\Delta^2 T K^2 + 4\Delta\sqrt{|A|}TK^2 \sum_{t=1}^{T} \Pi^t(I)\epsilon^t(I) \tag{40}$$

in this case. We elide the proof.

The new regret bound in Eq. (40) can be plugged into Lanctot 2013, Theorem 3 as we do for Theorem 4, leading to

$$\bar{R}_p^T \leq \sum_{I \in \mathcal{I}_p} \left(\left(1 + \frac{\sqrt{2}}{\sqrt{\rho K}}\right) \Delta \frac{\sqrt{|A|}}{\sqrt{T}} + \frac{4}{\sqrt{T}} \sqrt{|A|\Delta \sum_{t=1}^{T} \Pi^t(I)\epsilon^t(I)} \right) \tag{41}$$

Simplifying the first term and rearranging,

$$\bar{R}_p^T \leq \left(1 + \frac{\sqrt{2}}{\sqrt{\rho K}}\right) \Delta |\mathcal{I}_p| \frac{\sqrt{|A|}}{\sqrt{T}} + \frac{4\sqrt{|A|\Delta}}{\sqrt{T}} \sum_{I \in \mathcal{I}_p} \sqrt{\sum_{t=1}^{T} \Pi^t(I)\epsilon^t(I)} \tag{42}$$

$$\bar{R}_p^T \leq \left(1 + \frac{\sqrt{2}}{\sqrt{\rho K}}\right) \Delta |\mathcal{I}_p| \frac{\sqrt{|A|}}{\sqrt{T}} + \frac{4\sqrt{|A|\Delta}}{\sqrt{T}} |\mathcal{I}_p| \frac{\sum_{I \in \mathcal{I}_p}}{|\mathcal{I}_p|} \sqrt{\sum_{t=1}^{T} \Pi^t(I)\epsilon^t(I)} \qquad \text{Adding canceling factors} \tag{43}$$

$$\leq \left(1 + \frac{\sqrt{2}}{\sqrt{\rho K}}\right) \Delta |\mathcal{I}_p| \frac{\sqrt{|A|}}{\sqrt{T}} + \frac{4\sqrt{|A|\Delta|\mathcal{I}_p|}}{\sqrt{T}} \sqrt{\sum_{t=1}^{T} \sum_{I \in \mathcal{I}_p} \Pi^t(I)\epsilon^t(I)} \qquad \text{by Jensen's inequality} \tag{44}$$

Now, lets consider the average MSE loss $\mathcal{L}_V^T(\mathcal{M}^T)$ at time T over the samples in memory $\mathcal{M}^{\mathcal{T}}$.

We start by stating two well-known lemmas:

Lemma 3. *The MSE can be decomposed into bias and variance components*

$$\mathbb{E}_x[(x-\theta)^2] = (\theta - \mathbb{E}[x])^2 + \mathrm{Var}(\theta) \tag{45}$$

Lemma 4. *The mean of a random variable minimizes the MSE loss*

$$\operatorname*{argmin}_{\theta} \mathbb{E}_x[(x-\theta)^2] = \mathbb{E}[x] \tag{46}$$

and the value of the loss at when $\theta = \mathbb{E}[x]$ *is* $\mathrm{Var}(x)$.

$$\mathcal{L}_V^T = \frac{1}{\sum_{I \in \mathcal{I}_p} \sum_{t=1}^{T} x^t(I)} \sum_{I \in \mathcal{I}_p} \sum_{t=1}^{T} x^t(I) \left\| \tilde{r}^t(I) - V(I|\theta^T) \right\|_2^2 \tag{47}$$

$$\geq \frac{1}{|\mathcal{I}_p|T} \sum_{I \in \mathcal{I}_p} \sum_{t=1}^{T} x^t(I) \left\| \tilde{r}^t(I) - V(I|\theta^T) \right\|_2^2 \tag{48}$$

$$= \frac{1}{|\mathcal{I}_p|} \sum_{I \in \mathcal{I}_p} \Pi^T(I)\, \mathbb{E}_t \left[\left\| \tilde{r}^t(I) - V(I|\theta^T) \right\|_2^2 \middle| x^t(I) = 1 \right] \tag{49}$$

Let V^* be the model that minimizes $\mathcal{L}^T$ on $\mathcal{M}_T$. Using Lemmas 3 and 4,

$$\mathcal{L}_V^T \geq \frac{1}{|\mathcal{I}_p|T} \sum_{I \in \mathcal{I}_p} \Pi^T(I) \left(\left\| V(I|\theta^T) - \mathbb{E}_t \left[\tilde{r}^t(I) | x^t(I) = 1 \right] \right\|_2^2 + \mathcal{L}_{V^*}^T \right) \tag{50}$$

So,

$$\mathcal{L}_V^T - \mathcal{L}_{V^*}^T \geq \frac{1}{|\mathcal{I}_p|} \sum_{I \in \mathcal{I}_p} \Pi^T(I)\, \epsilon^T(I) \tag{51}$$

$$\sum_{I \in \mathcal{I}_p} \Pi^T(I)\, \epsilon^T(I) \leq |\mathcal{I}_p|(\mathcal{L}_V^T - \mathcal{L}_{V^*}^T) \tag{52}$$

Plugging this into Eq. 42, we arrive at

$$\bar{R}_p^T \leq \left(1 + \frac{\sqrt{2}}{\sqrt{\rho K}}\right) \Delta |\mathcal{I}_p| \frac{\sqrt{|A|}}{\sqrt{T}} + \frac{4\sqrt{|A|\Delta|\mathcal{I}_p|}}{\sqrt{T}} \sqrt{|\mathcal{I}_p| \sum_{t=1}^{T} (\mathcal{L}_V^t - \mathcal{L}_{V^*}^t)} \tag{53}$$

$$\leq \left(1 + \frac{\sqrt{2}}{\sqrt{\rho K}}\right) \Delta |\mathcal{I}_p| \frac{\sqrt{|A|}}{\sqrt{T}} + 4|\mathcal{I}_p| \sqrt{|A| \Delta \epsilon_{\mathcal{L}}} \tag{54}$$

So far we have assumed that $\mathcal{M}_V$ contains all sampled regrets. The number of samples in the memory at iteration t is bounded by $K \cdot |\mathcal{I}_p| \cdot t$. Therefore, if $K \cdot |\mathcal{I}_p| \cdot T < |\mathcal{M}_V|$ then the memory will never be full, and we can make this assumption.[8] □

B.5. Proof of Corollary 1

Proof. Let $\rho = T^{-1/4}$.

$$P\left(\bar{R}_p^T > \left(1 + \frac{\sqrt{2}}{\sqrt{K}}\right) \Delta |\mathcal{I}_p| \frac{\sqrt{|A|}}{T^{-1/4}} + 4|\mathcal{I}_p| \sqrt{|A| \Delta \epsilon_{\mathcal{L}}} \right) < T^{-1/4} \tag{55}$$

Therefore, for any $\epsilon > 0$,

$$\lim_{T \to \infty} P\left(\bar{R}_p^T - 4|\mathcal{I}_p| \sqrt{|A| \Delta \epsilon_{\mathcal{L}}} > \epsilon \right) = 0. \tag{56}$$

□

[8] We do not formally handle the case where the memories become full in this work. Intuitively, reservoir sampling should work well because it keeps an 'unbiased' sample of previous iterations' regrets. We observe empirically in Figure 4 that reservoir sampling performs well while using a sliding window does not.

C. Network Architecture

In order to clarify the network architecture used in this work, we provide a PyTorch (Paszke et al., 2017) implementation below.

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class CardEmbedding(nn.Module):
    def __init__(self, dim):
        super(CardEmbedding, self).__init__()
        self.rank = nn.Embedding(13, dim)
        self.suit = nn.Embedding(4, dim)
        self.card = nn.Embedding(52, dim)

    def forward(self, input):
        B, num_cards = input.shape
        x = input.view(-1)

        valid = x.ge(0).float()  # -1 means 'no card'
        x = x.clamp(min=0)

        embs = self.card(x) + self.rank(x // 4) + self.suit(x % 4)
        embs = embs * valid.unsqueeze(1)  # zero out 'no card' embeddings

        # sum across the cards in the hole/board
        return embs.view(B, num_cards, -1).sum(1)

class DeepCFRModel(nn.Module):
    def __init__(self, ncardtypes, nbets, nactions, dim=256):
        super(DeepCFRModel, self).__init__()

        self.card_embeddings = nn.ModuleList(
            [CardEmbedding(dim) for _ in range(ncardtypes)])

        self.card1 = nn.Linear(dim * ncardtypes, dim)
        self.card2 = nn.Linear(dim, dim)
        self.card3 = nn.Linear(dim, dim)

        self.bet1 = nn.Linear(nbets * 2, dim)
        self.bet2 = nn.Linear(dim, dim)

        self.comb1 = nn.Linear(2 * dim, dim)
        self.comb2 = nn.Linear(dim, dim)
        self.comb3 = nn.Linear(dim, dim)

        self.action_head = nn.Linear(dim, nactions)

    def forward(self, cards, bets):

        """
        cards: ((N x 2), (N x 3)[, (N x 1), (N x 1)])  # (hole, board, [turn, river])
        bets: N x nbet_feats
        """

        # 1. card branch
        # embed hole, flop, and optionally turn and river
        card_embs = []
        for embedding, card_group in zip(self.card_embeddings, cards):
            card_embs.append(embedding(card_group))
        card_embs = torch.cat(card_embs, dim=1)

        x = F.relu(self.card1(card_embs))
        x = F.relu(self.card2(x))
```

```
x = F.relu(self.card3(x))

# 1. bet branch
bet_size = bets.clamp(0, 1e6)
bet_occurred = bets.ge(0)
bet_feats = torch.cat([bet_size, bet_occurred.float()], dim=1)
y = F.relu(self.bet1(bet_feats))
y = F.relu(self.bet2(y) + y)

# 3. combined trunk
z = torch.cat([x, y], dim=1)
z = F.relu(self.comb1(z))
z = F.relu(self.comb2(z) + z)
z = F.relu(self.comb3(z) + z)

z = normalize(z)  # (z - mean) / std
return self.action_head(z)
```