7 Supplementary Material

Alternative Definitions of Filtering

We analyze history filtering in terms of the public state and joint range for simplicity, but some algorithms compute value functions using player ranges and beliefs instead. Definitions 4 and 5 alternatively describe filtering histories from a player's infostate according to their beliefs and filtering a player's infostates from the public state according to their range, respectively.

Definition 4. ***(FILTER-BELIEF)*** *For any FOSG G with finite $\mathcal{W}, \mathcal{A}$ and joint policy π, let* `FILTER-BELIEF`$(G, \pi) := \{(S_i, h) \in \Sigma^* \times \Sigma^* : s_i(h) = S_i, P^\pi(h|S_i) > 0\}$.

Definition 5. ***(FILTER-RANGE)*** *For any FOSG G with finite $\mathcal{W}, \mathcal{A}$ and joint policy π, let* `FILTER-RANGE`$(G, \pi) := \{(S, S_i) \in \Sigma^* \times \Sigma^* : S_i \in \mathcal{S}_i(S), P^\pi(S_i|S) > 0\}$.

The key difference in Definition 4 is that the input is a player i's infostate S_i instead of a public state S. Definition 5 consists of public state S as input and i's infostates as output. Both require that the output history is reachable (with non-zero probability) according to the joint policy π.

Lemma 3. *`FILTER-BELIEF` and `FILTER` are mutually polynomial-time reducible.*

Proof. Given an instance S of `FILTER-BELIEF`(G, π) with FOSG G and player i with infostate tree $\mathcal{S}_i$ in G, there exists an FOSG G' with public tree $\mathcal{S}' = \mathcal{S}_i$. Given G, we need to construct G' in polynomial-time with respect to the encoding length for the input instance S, using only the functions that define G. To do so, we include evaluations of i's private observation function s_i in the public observation function s'_{pub} of G'. In other words, we set $s'_{pub} = (s_i, s_{pub})$ in G', which increases the cost of evaluating $s'_{pub}(h)$ by a factor of 2. Since each call to the observation function is polynomial-time with respect to the encoding of w, a, w', our transformation is also polynomial-time.

A basic fact from this construction is $h \in S_i \iff h \in S'$. Moreover, there exists a joint policy π' such that $P^{\pi'}(h) = P^\pi(h)$ for each such h. Joint policy π' is constructed by mapping $\pi'_p((s'_p(g), s'_{pub}(g)), a) = \pi_p((s_p(g), s_{pub}(g)), a)$ for all $g \cdot a \sqsubseteq h$ for each player $p \in \mathcal{N}$. Each player $p \neq i \in G'$ ignores i's private information, reconstructing an information set on which π_p is valid and contributes the desired reach probability on h. Evaluating π' therefore requires just a single call to π, so the transformation from π to π' is clearly polynomial-time with respect to the encoding length of S.

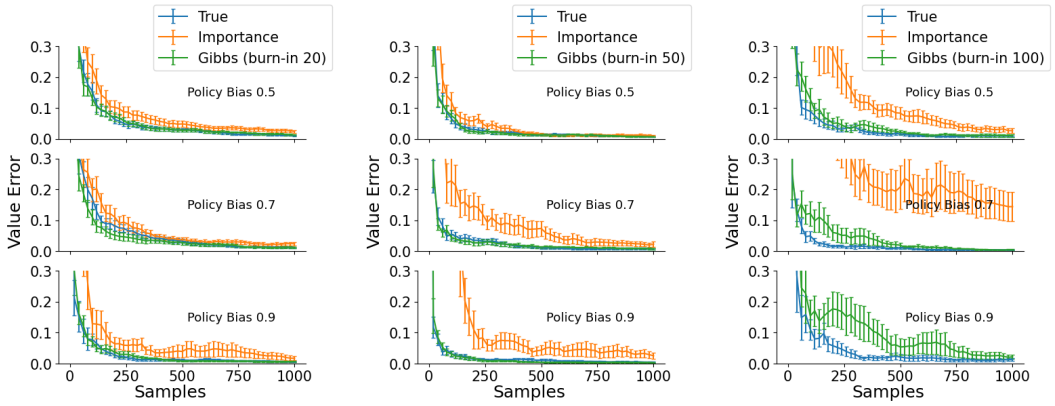
Since $S = S'$, we have $P^\pi(h|S_i) = P^{\pi'}(h|S')$. It follows that $(S_i, h) \in$ `FILTER-BELIEF`$(G, \pi) \iff (S', h) \in$ `FILTER`(G', π'), where G' and π' can be constructed in polynomial time given G and π.

For the other direction, given an instance of `FILTER`(G, π) with FOSG G, construct G' by adding a player i (who takes no actions) and setting $s'_i = s_{pub}$, $s'_{pub} = \emptyset$, and $s'_j = (s_{pub}, s_j)$ for every $j \in -i$, and similarly constructing a new joint policy π' that copies π at every infostate for players in $-i$. It follows that $(S, h) \in$ `FILTER`$(G, \pi) \iff (S'_i, h) \in$ `FILTER-BELIEF`(G', π'). □

Lemma 4. *`FILTER` $\leq_p$ `FILTER-RANGE`.*

Proof. Given an instance S of `FILTER`(G, π), we construct game G' by adding a player i that has perfect information but takes no actions. This transformation is computed in polynomial-time with respect to the encoding length of S by creating an observation function that returns the current history to i. Thus, for any $S \in \mathcal{S}$ in the original game G, there exists an S'_i in i's infostate tree which corresponds to exactly one $h \in S$. Since i takes no actions, the policy is unchanged. This means we have $P^\pi(h|S) = P^\pi(S'_i|S)$ by the definition of infostate reach probability. Therefore, $(S, S'_i) \in$ `FILTER-RANGE`$(G', \pi) \implies (S, h) \in$ `FILTER`(G, π). □

Lemmas 3 and 4 show that `FILTER` is polynomial-time reducible to these alternative definitions, and unsurprisingly, that filtering a history from a player's beliefs is equivalent to filtering a history from the joint range.



(a) 192 Histories (b) 12,960 Histories (c) 544,320 Histories

Figure 6: Value estimation error of TTCG Gibbs Sampler with specified burn-in and baselines on PBS of various sizes. Error bars show one standard error of the mean over 100 runs. Joint policies are generated randomly according to *policy bias*.

Experiment Parameters

Table 1 shows the game parameters used to generate Figures 4 and 6. Experiments are repeated 100 times and runtimes range from seconds to 12 hours on a single core of an AMD Ryzen 9 © CPU.

Table 1: Value estimation and mixing time experiment parameters.

Size	Players	Suits	Ranks	Tricks	Tricks Played
192	3	2	4	2	1
12,960	3	3	4	3	2
544,320	3	3	4	3	1

Additional Experiments

In addition to the value estimation experiments using policies learned via reinforcement learning in the main paper, we conducted similar experiments using random policies to increase empirical coverage of the policy space. At any information state, the policies place *policy bias* probability on a randomly selected action and distribute the remaining probability mass uniformly across the other actions. Figure 6 shows that these experiments yielded similar results to the RL policies.

Table 2 shows the entropy and variance of the tested public states, organized by the policy bias used to generate them. Policy bias clearly has an effect on both the reach probabilities of the histories in the PBS, as well as on the variance of the PBS value. The medium size exhibits proportionally higher entropy and lower variance because an extra trick has been played—meaning play is closer to the end of the game.

Table 2: Mean variance and entropy of generated public belief states of different sizes.

	Entropy			Variance		
Bias	0.5	0.7	0.9	0.5	0.7	0.9
196	6.84 ± 0.05	6.13 ± 0.07	4.89 ± 0.07	12.98 ± 1.12	10.11 ± 0.95	5.36 ± 0.93
12,960	12.04 ± 0.06	10.99 ± 0.08	9.30 ± 0.08	8.77 ± 1.02	6.01 ± 0.86	2.38 ± 0.58
544,320	16.21 ± 0.25	14.06 ± 0.45	12.36 ± 0.17	10.55 ± 1.17	11.44 ± 2.40	8.77 ± 2.01

Proofs of Theorems

Lemma 1. For any FOSG G with finite $\mathcal{W}, \mathcal{A}$ and arbitrary joint policy π, $\texttt{FILTER}(G, \pi)$ is polynomially balanced and polynomial-time verifiable.

Proof. We need to show that there exists a polynomial p such that for any $(S, h) \in \texttt{FILTER}(G, \pi)$, $|h| \leq p(|S|)$ and that the predicate $(S, h) \in \texttt{FILTER}(G, \pi)$ can be verified in polynomial time. $\mathcal{W}$ being finite implies there exists an encoding for all $w \in \mathcal{W}$ and a constant $c_{\mathcal{W}}$, where $|w| \leq c_{\mathcal{W}}$. Likewise, $\mathcal{A}$ being finite implies the existence of an encoding for all $a \in \mathcal{A}$ and a constant $c_{\mathcal{A}}$ such that $|a| \leq c_{\mathcal{A}}$. $h = (w^0, a^0, w^1, a^1, ..., w^t)$, so $|h| \leq t(c_{\mathcal{W}} + c_{\mathcal{A}}) + c_{\mathcal{W}} \leq ct$ for all $t > 0$ and some constant $c > 0$. For any encoding of input observations $S := (o^1, o^2, ..., o^t)$ with $|o^i| \geq 1$, $|S| \geq t$, so $|h| \leq c|S|$. Lastly, predicate $(S, h) \in \texttt{FILTER}(G, \pi)$ is easily verified in polynomial time in the encoding length of S and h by checking that $O(w^k, a^k, w^{k+1}) = o^{k+1}$ for $0 \leq k \leq t - 1$ and $P^\pi(h) > 0$. □

Theorem 1. There exists a joint policy π for which the construction problem associated with $\texttt{FILTER}(\texttt{3-FSAT-GAME}, \pi)$ is FNP-complete.

Proof. A construction problem is in the class FNP if its associated relation is both polynomially-balanced and polynomial-time verifiable (Bellare and Goldwasser [1994]). A construction problem is FNP-complete if and only if it belongs to FNP and all other problems in FNP are polynomial-time reducible to it.

First, Lemma 1 implies that the problem is in FNP. Next, we need to show that 3-FSAT-GAME, an FNP-complete problem, can be reduced to $\texttt{FILTER}(\texttt{3-FSAT-GAME}, \pi)$.

Given TM M, which computes construction for $\texttt{FILTER}(\texttt{3-FSAT-GAME}, \pi)$ and any 3-FSAT instance $\phi = (c_1, ..., c_k)$ containing variables $y_1, ..., y_m$ in $\texttt{DTIME}(p(|x|))$ for some time-constructible polynomial p, map ϕ to the equivalent public state $x = \phi$ in *3-FSAT-GAME* and run $h = M'(x)$, where M' simulates M with a time bound of $p(|x|)$ in $DTIME(p(|x|)^2)$.

Let $\pi_1(w^0)$ be the uniform policy, which assigns equal action probability to each of the 2^m actions available at w^0 Since action a is the singular joint action at any $w \neq w^0$, $\pi_i(s, a) = 1$ for any $s \in S_i(s_{pub})$ for all players i—implying that any h output by M' satisfies $P^\pi(h) > 0$. If M' returns $h = (w^0, a^0, w, a^1, w, a^2, ..., w)$, then w must be a satisfying assignment to ϕ. If M' returns NO, then there is no $h, s_{pub}(h) = S$ and ϕ is unsatisfiable. □

Theorem 2. For any FOSG G with finite $\mathcal{W}, \mathcal{A}$, the enumeration problem associated with $\texttt{FILTER}(G, \pi)$ can be solved in polynomial time if and only if G's public tree is sparse.

Proof. Suppose G's public tree is sparse. To enumerate $\mathcal{H}_S$, consider a basic breadth-first search which, at time k, only searches actions that satisfy $O_{pub}(w^k, a^k, w^{k+1}) = O^{k+1}$, where $a^k \in \mathcal{A}(w^k)$ and w^{k+1} is in the support of $\mathcal{T}(w^k, a^k)$. Let the set $\mathcal{H}_S^k = \{h' \in \mathcal{H} : h' \sqsubseteq h, h \in \mathcal{H}_S, |h'| = k\}$ contain the length k prefix histories of all histories in $\mathcal{H}_S$. All histories in $\mathcal{H}_S^k$ must produce the same public observation sequence and therefore must correspond to a unique public state. Since G's public tree is sparse, this implies there can be at most $p(k)$ histories at depth k of the search. This means we must evaluate $O_{pub}(\cdot)$ at most $|\mathcal{A} \times \mathcal{W}|p(k)$ times at depth k and therefore $O(|h|p(|h|))$ times overall.

Conversely, suppose there exists a polynomial-time TM M for solving the enumeration problem associated with $\texttt{FILTER}(G, \pi)$. Assume there exists a dense public state S in G. Then for input instance S, $M(S)$ computes $\mathcal{H}_S$ in $p(t)$-time for some polynomial p. This is impossible since, for any polynomial p', $|\mathcal{H}_S| > p'(t)$ by assumption. □

Lemma 2 For TTCG G and joint policy π with full support, $\texttt{FILTER}(G, \pi)$ can be solved in polynomial-time using a maximum flow computation.

Proof. Given public state instance S, let $N = (V, E)$ be the flow network constructed as follows. The network contains a source connected via edges to a set of *suit* vertices (one vertex for every suit, each with an edge from the source). Each suit vertex is connected to any number of vertices from the

set of *player* vertices (one for each player). A suit vertex is connected via a directed edge to a player vertex if and only if the player can hold that suit in their hand. Finally, each suit vertex is connected to the sink via a directed edge. For the edges connected to source s, capacity $c(s, u)$ is the number of cards remaining in suit u. Likewise, for edges connected to the sink t, $c(v, t)$ is the number of cards that must be assigned to player v. Edges connecting suits to players have infinite capacity.

Let $\mathcal{F}$ be the set of flows over N, and $\phi : \mathcal{H} \to \mathcal{F}$ be the following transformation from history to flow: for edges from source to suit and player to sink, the flow is equal to the capacity, for edges from suit to player, the flow is equal to the number of cards from that suit dealt to that player in h.

Suppose there exists some $h \in \mathcal{H}_S$, then by construction $|\phi(h)| = \sum_u c(u, t) = \sum_u c(s, u)$ and is an integer because the number of cards dealt to each player is known. To show that $|\phi(h)|$ is a maximum flow, assume for contradiction, that there exists some $f \in \mathcal{F}$ such that $|f| > |\phi(h)|$. This implies that $|f| > \sum_u c(s, u)$ which is impossible, so $|\phi(h)| \geq |f|$ for all $f \in \mathcal{F}$.

Now suppose we have some maximum flow $f^* \in \mathcal{F}$; we can construct $h \in \mathcal{H}_S$ in the following way. If $|f*| < \sum_u c(s, u)$, then at least one player or suit cannot have the appropriate number of cards allocated to it, so we return that $\mathcal{H}_S$ is empty. Otherwise, for each suit u and player v, we assign $f^*(u, v)$ arbitrary cards (which have not been revealed by play) from suit u to player v in the deal. □

Theorem 3. The TTCG Gibbs sampler is aperiodic and irreducible.

Proof. Self-transitions in the chain imply aperiodicity. We prove irreducibility by showing that there exists a path between any two consistent suit length assignments that can be generated via multiple iterations of our neighbor generation algorithm. This proves the chain's irreducibility because the neighbor set is the union of all history subsets that correspond to neighboring suit length assignments.

Given two suit length assignment matrices A and B (see description in Section 5.3, we prove that B is reachable from A by showing that the L_1-norm

$$||B - A||_1 = \sum_{j=1}^{n} \sum_{i=1}^{m} |b_{i,j} - a_{i,j}|$$

decreases to zero for a sequence of iterations of our neighbor generation algorithm, and that each iteration makes at most $\max\{n, m\}$ *swaps* (where a player gains a card of one suit and loses a card of another) for an $n \times m$ matrix.

First, we show that applying a sequence of `RingSwap` calls to A decreases $||B - A||_1$ to zero. Let $C = B - A$, and denote a swap that adds one to $c_{i,j}$ and subtracts one from $c_{i,k}$ as $\delta = (i; j, k)$. Since all columns and rows of C sum to zero, if $||C||_1 > 0$, there must be some element $c_{i,j} < 0$ and another $c_{i,k} > 0$. Since swap $\delta_0 = (i; j, k)$ decreases $||C||_1$ by 2, and such a swap is guaranteed to exist whenever $||C||_1 > 0$, repeating this process decreases $||C||_1$ to zero.

After applying δ_0, $\sum_b C_{a,b} = 0$ for all rows a, but columns $\sum_a C_{a,k} = -1$ and $\sum_a C_{a,j} = 1$, so the result does not correspond to a valid suit assignment. To generate a neighbor, the algorithm performs a sequence of swaps until the column sums are corrected. Since, prior to δ_0, $c_{i,k} > 0$, there exists $c_{l,k} < 0$; likewise there must be some $c_{l,z} > 0$. Perform swap $(l; k, z)$, now $\sum_a C_{a,k} = 0$. If $z = j$, then $\sum_a C_{a,j} = 0$ and we are done. Otherwise, $\sum_a C_{a,z} = 1$, and the above process can be repeated using column z instead of l.

Now, we show that given any starting swap, we can reach a valid suit length assignment using at most n swaps. Assume $n \geq m$, otherwise transpose A and B. As we have shown above, given an arbitrary starting swap $(i; a, b)$, all rows and columns will sum to zero once a swap of the form $(j; b, c)$ adds a unit back to column b. Consider the sequence of swaps made until the $z = j$ stopping condition is reached and suppose that a row has to be repeated in this sequence (i.e. there is a subsequence $(k; d, e)$,..., $(k; f, g)$). This means the remainder of the sequence begins by with a swap from column g and ends with a swap into column b, and leads to the condition that all rows and columns sum to zero. But before the first swap in the subsequence, $(k; d, e)$, only columns d and b have non-zero sum. This implies we can remove this subsequence, and replace it with $(k; d, g)$ and proceed until b is reached. So, each row must only be visited once in the sequence of swaps, implying that there exists a sequence with length at most n which turns A to A' where all rows and columns of $A' - B$ sum to

zero and $||A' - B||_1 < ||A - B||$. This implies a path between any two suit length assignments can be generated using multiple ring swaps of length $<= n$. □

Theorem 4. The stationary distribution of the TTCG Gibbs sampler with input π is P^π.

Proof. A sufficient condition for some μ to be the stationary distribution of a Markov chain is that the chain is reversible with respect to μ (Häggström and others [2002]). So if we can show that $P_i^\pi Q_{i,j} = P_j^\pi Q_{j,i}$, the theorem follows. This is the standard approach which derives the Metropolis-Hastings algorithm ([Metropolis *et al.*, 1953]).

First, suppose $P_i^\pi|\Omega_j| > P_j^\pi|\Omega_i|$. Then, by inspecting lines 1-5 in the algorithm we see that

$$\begin{aligned} P_i^\pi Q_{i,j} &= P_i^\pi \frac{1}{|\Omega_i|} \frac{\bar{P}_j^\pi|\Omega_i|}{\bar{P}_i^\pi|\Omega_j|} \\ &= \frac{P_j^\pi}{|\Omega_j|} = P_j^\pi \frac{1}{|\Omega_j|} \\ &= P_j^\pi Q_{j,i} \end{aligned}$$

where the last equation holds because $P_i^\pi|\Omega_j| > P_j^\pi|\Omega_i|$ implies $z = 1$ for the transition from j to i. The same applies to the case where $P_i^\pi|\Omega_j| < P_j^\pi|\Omega_i|$. Finally, suppose $P_i^\pi|\Omega_j| = P_j^\pi|\Omega_i|$. Then,

$$\begin{aligned} & P_i^\pi|\Omega_j| = P_j^\pi|\Omega_i| \\ \implies & \frac{P_i^\pi}{|\Omega_i|} = \frac{P_j^\pi}{|\Omega_j|} \\ \implies & P_j^\pi Q_{j,i} = P_i^\pi Q_{i,j} \end{aligned}$$

So the TTCG Gibbs sampler is reversible with respect to P^π. □