GPU-ACCELERATED COUNTERFACTUAL REGRET MINIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Counterfactual regret minimization is a family of algorithms of no-regret learning dynamics capable of solving large-scale imperfect information games. We propose implementing this algorithm as a series of dense and sparse matrix and vector operations, thereby making it highly parallelizable for a graphical processing unit, at a cost of higher memory usage. Our experiments show that our implementation performs up to about 352.5 times faster than OpenSpiel's Python implementation and up to about 22.2 times faster than OpenSpiel's C++ implementation and the speedup becomes more pronounced as the size of the game being solved grows.

1 INTRODUCTION

Counterfactual regret minimization (CFR) (Zinkevich et al., 2007) and its variants dominated the development of AI agents for large imperfect information games like *Poker* (Tammelin et al., 2015; Moravčík et al., 2017; Brown & Sandholm, 2018; 2019b) and *The Resistance: Avalon* (Serrino et al., 2019) and were components of ReBeL (Brown et al., 2020) and student of games (Schmid et al., 2023). Notable variants of CFR are as follows: CFR+ by Tammelin (2014) (optionally) eliminates the averaging step while improving the convergence rate; Sampling variants (Lanctot et al., 2009) makes a complete recursive tree traversal unnecessary; Burch et al. (2014) proposes CFR-D in which games are decomposed into subgames; Brown & Sandholm (2019a) explores modifying CFR such as to explore alternate weighted averaging (and discounting) schemes; Xu et al. (2024) learns a discounting technique from smaller games to be used in larger games.

We propose implementing this algorithm as a series of dense and sparse matrix and vector operations, thereby making it parallelizable for a graphical processing unit (GPU) at a cost of higher memory usage. We analyze the runtimes of our implementation with both computer processing unit (CPU) and GPU backends and compare them to Google DeepMind's OpenSpiel (Lanctot et al., 2020) implementations in Python and C++ on 8 games of differing sizes.

Our experiments show that, compared to Google DeepMind OpenSpiel's (Lanctot et al., 2020) Python implementation, our GPU implementation performs about 2.7 times slower for small games but is up to about 352.5 times faster for large games. Against their C++ implementation, our performance with a GPU is up to about 75.7 times slower for small games, but is up to about 22.2 times faster for large games. Even without a GPU (i.e. with a CPU backend), our implementation shows speedups compared to the OpenSpiel baselines (from about 1.7 to 56.4 times faster than their Python implementation and from 17.0 times slower to 5.6 times faster than theirs in C++). In general, We see that the speedup becomes more pronounced as the size of the game being solved grows.

2 BACKGROUND

The background of our work and the notations we use throughout this paper is introduced below.

2.1 FINITE EXTENSIVE-FORM GAMES

An extensive-form game is a powerful representation of games that allow the specification of the rules of the game, information sets, actions, actors (players and the nature), chances, and payoffs.

Definition 1 Formally, a **finite extensive-form game** (Osborne & Rubinstein, 1994) is a structure $\mathcal{G} = \langle \mathcal{T}, \mathbb{H}, f_h, \mathbb{A}, f_a, \mathbb{I}, f_i, \sigma_0, u \rangle$ where:

- $\mathcal{T} = \langle \mathbb{V}, v_0, \mathbb{T}, f_{Pa} \rangle$ is a **finite game tree** with a **finite set of nodes** (i.e. vertices) $\mathbb{V}$, a unique **initial node** (i.e. a root) $v_0 \in \mathbb{V}$, a **finite set of terminal nodes** (i.e. leaves) $\mathbb{T} \subseteq \mathbb{V}$, and a **parent function** $f_{Pa} : \mathbb{V}_+ \to \mathbb{D}$ that maps a non-initial node (i.e. a non-root) $v_+ \in \mathbb{V}_+$ to an immediate predecessor (i.e. a parent) $d \in \mathbb{D}$, with $\mathbb{V}_+ = \mathbb{V} \backslash \{v_0\}$ the finite set of non-initial nodes (i.e. non-roots) and $\mathbb{D} = \mathbb{V} \backslash \mathbb{T}$ the finite set of decision nodes (i.e. internal vertices),
- $\mathbb{H}$ is a **finite set of information sets**, $f_h : \mathbb{D} \to \mathbb{H}$ is an **information partition** of $\mathbb{D}$ associating each decision node $d \in \mathbb{D}$ to an information set $h \in \mathbb{H}$,
- $\mathbb{A}$ is a **finite set of actions**, $f_a : \mathbb{V}_+ \to \mathbb{A}$ is an **action partition** of $\mathbb{V}_+$ associating each non-initial node $v_+ \in \mathbb{V}_+$ to an action $a \in \mathbb{A}$ such that $\forall d \in \mathbb{D}$ the restriction $f_{a,d} : S(d) \to A(f_h(d))$ is a bijection, with $S(d \in \mathbb{D}) = \{v_+ \in \mathbb{V}_+ : f_{Pa}(v_+) = d\}$ the finite set of immediate successors (i.e. children) of a node $d \in \mathbb{D}$ and $A(h \in \mathbb{H}) = \{a \in \mathbb{A} : [\exists v_+ \in \mathbb{V}_+](f_h(f_{Pa}(v_+)) = h \wedge f_a(v_+) = a)\}$ the finite set of available actions at an information set $h \in \mathbb{H}$,
- $\mathbb{I}$ is a **finite set of (rational) players and, optionally, the nature** (i.e. chance) $i_0 \in \mathbb{I}$, $f_i : \mathbb{H} \to \mathbb{I}$ is a **player partition** of $\mathbb{H}$ associating each information set $h \in \mathbb{H}$ to a player $i \in \mathbb{I}$,
- $\sigma_0 : \mathbb{Q}_0 \to [0, 1]$ is a **chance probabilities function** that associates each pair of a nature information set and an available action $(h_0, a) \in \mathbb{Q}_0$ to an independent probability value, with $\mathbb{Q}_j = \{(h, a) \in \mathbb{Q} : h \in \mathbb{H}_j\}$ the finite set of pairs of a player information set $h_j \in \mathbb{H}_j$ and an available action $a \in A(h_j)$, $\mathbb{Q} = \{(h, a) \in \mathbb{H} \times \mathbb{A} : a \in A(h)\}$ the finite set of pairs of an information set $h \in \mathbb{H}$ and an available action $a \in A(h)$, and $\mathbb{H}_j = \{h \in \mathbb{H} : f_i(h) = i_j\}$ the finite set of information sets associated with a player $i_j \in \mathbb{I}$, and
- $u : \mathbb{T} \times \mathbb{I}_+ \to \mathbb{R}$ is a **utility function** that associates each pair of a terminal node $t \in \mathbb{T}$ and a (rational) player $i_+ \in \mathbb{I}_+$ to a real payoff value. $\mathbb{I}_+ = \mathbb{I} \backslash \{i_0\}$ is the finite set of (rational) players.

2.2 Nash Equilibrium

Each player $i_j \in \mathbb{I}$ selects a **player strategy** $\sigma_j : \mathbb{Q}_j \to [0, 1]$ from a **set of player strategies** Σ_j. A player strategy $\sigma_j \in \Sigma_j$ associates, for each player information set $h_j \in \mathbb{H}_j$, a probability distribution over a finite set of available actions $A(h_j)$. A **strategy profile** $\sigma : \mathbb{Q} \to [0, 1]$ is a direct sum of the strategies of each player $\sigma = \bigoplus_{i_j \in \mathbb{I}} \sigma_j$ which, for each information set $h \in \mathbb{H}$, gives a probability distribution over a finite set of available actions $A(h)$. Σ is a set of strategy profiles. $\sigma_{-j} = \bigoplus_{i_k \in \mathbb{I} \backslash \{i_j\}} \sigma_k$ is a direct sum of all player strategies in σ except σ_j (i.e. that of player $i_j \in \mathbb{I}$).

Let $\pi : \Sigma \times \mathbb{V} \to \mathbb{R}$ be a probability of reaching a vertex $v \in \mathbb{V}$ following a strategy profile $\sigma \in \Sigma$.

$$\pi(\sigma \in \Sigma, v \in \mathbb{V}) = \begin{cases} \sigma(f_h(f_{Pa}(v)), f_a(v))\pi(\sigma, f_{Pa}(v)) & v \in \mathbb{V}_+ \\ 1 & v = v_0 \end{cases}$$

Then, define $\hat{u} : \Sigma \times \mathbb{I} \to \mathbb{R}$ to be an expected payoff of a (rational) player $i_+ \in \mathbb{I}_+$, following a strategy profile $\sigma \in \Sigma$.

$$\hat{u}(\sigma \in \Sigma, i_+ \in \mathbb{I}_+) = \sum_{t \in \mathbb{T}} \pi(\sigma, t) u(t, i_+)$$

A strategy profile $\sigma^* \in \Sigma$ is a **Nash equilibrium**, a traditional solution concept for non-cooperative games, if no player stands to gain by deviating from the strategy profile.

$$\forall i_{+,j} \in \mathbb{I}_+ \quad \hat{u}(\sigma^*, i_{+,j}) \geqslant \max_{\sigma'_j \in \Sigma_j} \hat{u}(\sigma'_j \oplus \sigma^*_{-j}, i_{+,j})$$

A strategy profile that approximates a Nash equilibrium σ^* is an **ϵ-Nash equilibrium** $\sigma^{*,\epsilon} \in \Sigma$ if

$$\forall i_{+,j} \in \mathbb{I}_+ \quad \hat{u}(\sigma^{*,\epsilon}, i_{+,j}) + \epsilon \geqslant \max_{\sigma'_j \in \Sigma_j} \hat{u}(\sigma'_j \oplus \sigma^{*,\epsilon}_{-j}, i_{+,j})$$

2.3 Counterfactual Regret Minimization

Define $\check{u} : \Sigma \times \mathbb{V} \times \mathbb{I}_+ \to \mathbb{R}$ as an expected payoff of a (rational) player $i_+ \in \mathbb{I}_+$ at a node $v \in \mathbb{V}$, following a strategy profile $\sigma \in \Sigma$.

$$\check{u}(\sigma \in \Sigma, v \in \mathbb{V}, i_+ \in \mathbb{I}_+) = \begin{cases} \sum_{s \in S(v)} \sigma(f_h(v), f_a(s)) \check{u}(\sigma, s, i_+) & v \in \mathbb{D} \\ u(v, i_+) & v \in \mathbb{T} \end{cases} \tag{1}$$

Let $\check{\pi} : \Sigma \times \mathbb{V} \times \mathbb{I} \to \mathbb{R}$ be a probability of reaching a vertex $v \in \mathbb{V}$ following a strategy profile $\sigma \in \Sigma$ while ignoring a strategy of a player $i \in \mathbb{I}$.

$$\check{\pi}(\sigma \in \Sigma, v \in \mathbb{V}, i \in \mathbb{I}) = \begin{cases} \check{\pi}(\sigma, f_{Pa}(v), i) \begin{cases} \sigma(f_h(f_{Pa}(v)), f_a(v)) & f_i(f_h(f_{Pa}(v))) \neq i \\ 1 & f_i(f_h(f_{Pa}(v))) = i \end{cases} & v \in \mathbb{V}_+ \\ 1 & v = v_0 \end{cases} \tag{2}$$

Below definition shows a "counterfactual" reach probability $\tilde{\pi} : \Sigma \times \mathbb{H} \to \mathbb{R}$.

$$\tilde{\pi}(\sigma \in \Sigma, h \in \mathbb{H}) = \sum_{d \in \mathbb{D} : f_h(d) = h} \check{\pi}(\sigma, d, f_i(h)) \tag{3}$$

Now, let $\tilde{u} : \Sigma \times \mathbb{H}_+ \to \mathbb{R}$ be a counterfactual utility, with $\mathbb{H}_+ = \mathbb{H} \backslash \mathbb{H}_0$ the finite set of information sets associated with (rational) players.

$$\tilde{u}(\sigma \in \Sigma, h_+ \in \mathbb{H}_+) = \frac{\sum_{d \in \mathbb{D} : f_h(d) = h_+} \check{\pi}(\sigma, d, f_i(h_+)) \check{u}(\sigma, d, f_i(h_+))}{\tilde{\pi}(\sigma, h_+)} \tag{4}$$

$\sigma|_{h \to a} \in \Sigma$ is an overrided strategy profile of σ where an action $a \in A(h)$ is always taken at an information set $h \in \mathbb{H}$.

$$\sigma|_{h \to a}((h', a') \in \mathbb{Q}) = \begin{cases} \mathbf{1}_{a = a'} & h = h' \\ \sigma(h', a') & h \neq h' \end{cases}$$

$\tilde{r} : \Sigma \times \mathbb{Q}_+ \to \mathbb{R}$ is the instantaneous counterfactual regret, with $\mathbb{Q}_+ = \mathbb{Q} \backslash \mathbb{Q}_0$ the finite set of pairs of a (rational) player information set $h_+ \in \mathbb{H}_+$ and an available action $a \in A(h_+)$.

$$\tilde{r}(\sigma \in \Sigma, (h_+, a) \in \mathbb{Q}_+) = \tilde{\pi}(\sigma, h_+)(\tilde{u}(\sigma|_{h_+ \to a}, h_+) - \tilde{u}(\sigma, h_+)) \tag{5}$$

$\bar{r}^{(T)} : \mathbb{Q}_+ \to \mathbb{R}$ is the average counterfactual regret at an iteration T. $\sigma^{(\tau)} \in \Sigma$ is the strategy played at an iteration τ.

$$\bar{r}^{(T)}(q_+ \in \mathbb{Q}_+) = \frac{1}{T} \sum_{\tau=1}^{T} \tilde{r}(\sigma^{(\tau)}, q_+) \tag{6}$$

The strategy profile for the next iteration $T + 1$ is $\sigma^{(T+1)} \in \Sigma$.

$$\sigma^{(T+1)}((h, a) \in \mathbb{Q}) = \begin{cases} \begin{cases} \frac{(\bar{r}^{(T)}(h,a))^+}{\sum_{a' \in A(h)} (\bar{r}^{(T)}(h,a'))^+} & \sum_{a' \in A(h)} (\bar{r}^{(T)}(h, a'))^+ > 0 \\ \frac{1}{|A(h)|} & \sum_{a' \in A(h)} (\bar{r}^{(T)}(h, a'))^+ = 0 \end{cases} & (h, a) \in \mathbb{Q}_+ \\ \sigma_0(h, a) & (h, a) \in \mathbb{Q}_0 \end{cases} \tag{7}$$

Counterfactual regret minimization (Zinkevich et al., 2007) is an algorithm that iteratively approximates a coarse correlated equilibrium $\bar{\sigma}^{(T)} : \mathbb{Q} \to \mathbb{R}$ (Hart & Mas-Colell, 2000).

$$\bar{\sigma}^{(T)}((h, a) \in \mathbb{Q}) = \frac{\sum_{\tau=1}^{T} \tilde{\pi}(\sigma^{(\tau)}, h)\sigma^{(\tau)}(h, a)}{\sum_{\tau=1}^{T} \tilde{\pi}(\sigma^{(\tau)}, h)} \tag{8}$$

Define $r^{(T)} : \mathbb{I}_+ \to \mathbb{R}$ as the average overall regret of a (rational) player $i_{+,j} \in \mathbb{I}_+$ at an iteration T.

$$r^{(T)}(i_{+,j} \in \mathbb{I}_+) = \frac{1}{T} \max_{\sigma'_j \in \Sigma_j} \sum_{\tau=1}^{T} (\hat{u}(\sigma'_j \oplus \sigma_{-j}^{(\tau)}, i_{+,j}) - \hat{u}(\sigma^{(\tau)}, i_{+,j}))$$

In 2-player ($|\mathbb{I}_+| = 2$) zero-sum games, if $\forall i_+ \in \mathbb{I}_+ \ r^{(T)}(i_+) \leqslant \epsilon$, the average strategy $\bar{\sigma}^{(T)}$ (at an iteration T) is also a 2ϵ-Nash equilibrium $\sigma^{*,2\epsilon} \in \Sigma$ (Zinkevich et al., 2007).

2.4 Prior Usages of GPUs for CFR

In the mainstream literature, algorithms inspired by CFR or using CFR as a subcomponent like DeepStack (Moravčík et al., 2017), Student of Games (Schmid et al., 2023), and ReBeL (Brown et al., 2020) only perform a limited lookahead instead of a complete game tree traversal. A neural network-based value function is typically used to evaluate the heuristic value of a node – GPUs can be utilized for the evaluation of these networks. Besides the fact that the vanilla CFR considers the entire game tree and does not use a value function, our approach differs significantly in that we use the GPU to parallelize CFR at every step of the process.

Reis (2015) and Rudolf (2021) have directly implemented CFR directly on CUDA and found orders of magnitude improvements in performance. However, in Rudolf's implementation, every thread assigned to each node moves up the game tree (toward the root), thus resulting in a quadratic number of visits to the game tree per iteration in the worst case. Reis's implementation is superior in that only one visit is made at each node per iteration by doing level-by-level updates (an approach we also use). However, aside from several reproducibility issues with the work by Reis (2015)[1], both approaches require each thread to perform a "large number of control flow statements" – a limitation mentioned by Reis (2015) – and require more generalized kernel instructions.

Our approach addresses these issues by framing this problem as a series of matrix/vector operations, and the utilization of GPUs for this task is an extremely well-studied problem in the field of systems, and can take advantage of optimized opcodes for these operations. Our implementation is also compatible with discrete games in OpenSpiel, which are commonly used as benchmarks for evaluating newly proposed CFR variants, unlike the work by Reis (2015) whose compatible games are limited to customized poker variants. In addition, our open-source pure Python code is available to the public.

3 Implementation

In order to highly parallelize the execution of CFR, we implement the algorithm as a series of dense and sparse matrix operations and avoid costly recursive game tree traversals. Due to space constraints, expanded forms of equations throughout this section had to be relegated to Appendix F.

3.1 Setup

Calculating expected payoffs of (rational) players $\breve{u} : \Sigma \times \mathbb{V} \times \mathbb{I}_+ \to \mathbb{R}$ in Equation 1 and "excepted" reach probability $\breve{\pi} : \Sigma \times \mathbb{V} \times \mathbb{I} \to \mathbb{R}$ in Equation 2 are classical problems of dynamic programming on trees. To calculate these values with matrix operations, we represent the game tree $\mathcal{T}$ as an adjacency matrix $\boldsymbol{G} \in \mathbb{R}^{\mathbb{V}^2}$ and the level graphs of the game tree $\mathcal{T}$ as adjacency matrices

[1] See Appendix G for more details

$\boldsymbol{L}^{(1)}, \boldsymbol{L}^{(2)}, \ldots, \boldsymbol{L}^{(D)} \in \mathbb{R}^{\mathbb{V}^2}$, with $D = \max_{t \in \mathbb{T}} d_{\mathcal{T}}(t)$ the maximum depth of any (terminal) node in the game tree $\mathcal{T}$ and $d_{\mathcal{T}} : \mathbb{V} \to \mathbb{Z}$ the depth of a vertex $v \in \mathbb{V}$ in the game tree $\mathcal{T}$ from the root v_0.

$$d_{\mathcal{T}}(v \in \mathbb{V}) = \begin{cases} 1 + d_{\mathcal{T}}(f_{Pa}(v)) & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}$$

$$\boldsymbol{G} = \left(\begin{cases} \mathbf{1}_{v = f_{Pa}(v')} & v \in \mathbb{D} \wedge v' \in \mathbb{V}_+ \\ 0 & v \in \mathbb{T} \vee v' = v_0 \end{cases} \right)_{(v,v') \in \mathbb{V}^2} \tag{9}$$

$$\forall l \in [1, D] \cap \mathbb{Z} \quad \boldsymbol{L}^{(l)} = \left(\begin{cases} \mathbf{1}_{v = f_{Pa}(v') \wedge d_{\mathcal{T}}(v') = l} & v \in \mathbb{D} \wedge v' \in \mathbb{V}_+ \\ 0 & v \in \mathbb{T} \vee v' = v_0 \end{cases} \right)_{(v,v') \in \mathbb{V}^2} \tag{10}$$

We also define matrices $\boldsymbol{M}^{(Q_+,V)} \in \mathbb{R}^{\mathbb{Q}_+ \times \mathbb{V}}, \boldsymbol{M}^{(H_+,Q_+)} \in \mathbb{R}^{\mathbb{H}_+ \times \mathbb{Q}_+}, \boldsymbol{M}^{(V,I_+)} \in \mathbb{R}^{\mathbb{V} \times \mathbb{I}_+}$ to represent the game $\mathcal{G}$. Matrix $\boldsymbol{M}^{(Q_+,V)}$ describes whether a node $v \in \mathbb{V}$ is a result of an action from a (rational) player information set $(h_+, a) \in \mathbb{Q}_+$. Matrix $\boldsymbol{M}^{(H_+,Q_+)}$ describes whether a (rational) player information set $h_+ \in \mathbb{H}_+$ is the first element of the corresponding (rational) player information set-action pair $(h_+, a) \in \mathbb{Q}_+$. Finally, matrix $\boldsymbol{M}^{(V,I_+)}$ describes whether a node $v \in \mathbb{V}$ has a parent whose associated information set's associated player is $i_+ \in \mathbb{I}_+$ (i.e. which player $i_+ \in \mathbb{I}_+$ acted to reach a node $v \in \mathbb{V}$). Note that we omit the nature player i_0 and related information sets $\mathbb{H}_0$ and information set-action pairs $\mathbb{Q}_0$ as only the strategies of (rational) players are updated by the algorithm. These mask-like matrices are later used to "select" the values associated with a player, action, node, or information set during the iteration.

$$\boldsymbol{M}^{(Q_+,V)} = \left(\begin{cases} \mathbf{1}_{q_+ = (f_h(f_{Pa}(v)), f_a(v))} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases} \right)_{(q_+,v) \in \mathbb{Q}_+ \times \mathbb{V}} \tag{11}$$

$$\boldsymbol{M}^{(H_+,Q_+)} = \left(\mathbf{1}_{h_+ = h'_+} \right)_{(h_+,(h'_+,a)) \in \mathbb{H}_+ \times \mathbb{Q}_+} \tag{12}$$

$$\boldsymbol{M}^{(V,I_+)} = \left(\begin{cases} \mathbf{1}_{f_i(f_h(f_{Pa}(v))) = i_+} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases} \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} \tag{13}$$

$\boldsymbol{G}, \boldsymbol{L}^{(1)}, \boldsymbol{L}^{(2)}, \ldots, \boldsymbol{L}^{(D)}, \boldsymbol{M}^{(Q_+,V)}, \boldsymbol{M}^{(H_+,Q_+)}, \boldsymbol{M}^{(V,I_+)}$ are constant matrices. In the games we experiment on, all aforesaid matrices except $\boldsymbol{M}^{(V,I_+)}$ are highly sparse (as demonstrated in Appendix A).[2] As such, they are implemented as sparse matrices in a compressed sparse row (CSR) format. Matrix $\boldsymbol{M}^{(V,I_+)}$ and all other defined matrices and vectors are implemented as dense.

Define a dense vector $\boldsymbol{s}^{(\sigma_0)}$ representing the probabilities of nature information set-action pairs $\mathbb{Q}_0$.

$$\boldsymbol{s}^{(\sigma_0)} = \left(\begin{cases} \begin{cases} \sigma_0(f_h(f_{Pa}(v)), f_a(v)) & f_h(f_{Pa}(v)) \in \mathbb{H}_0 \\ 0 & f_h(f_{Pa}(v)) \in \mathbb{H}_+ \end{cases} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases} \right)_{v \in \mathbb{V}} \tag{14}$$

$\boldsymbol{\sigma} \in \mathbb{R}^{\mathbb{Q}_+}$ is the strategy over (rational) player information set-action pairs $\mathbb{Q}_+$ at an iteration T.

$$\boldsymbol{\sigma} = \left(\sigma^{(T)}(q_+) \right)_{q_+ \in \mathbb{Q}_+} \tag{15}$$

A dense vector $\boldsymbol{\sigma}^{(T=1)} \in \mathbb{R}^{\mathbb{Q}_+}$ representing the initial strategy profile (i.e. at $T = 1$) is shown below.

[2]The sparsity of $\boldsymbol{M}^{(V,I_+)}$ depends on the number of (rational) players. For games with many players, it may be more efficient to implement this as sparse as well.

$$\boldsymbol{\sigma}^{(T=1)} = \left(\sigma^{(1)}(q_+)\right)_{q_+ \in \mathbb{Q}_+} = \left(\frac{1}{|A(h_+)|}\right)_{(h_+,a)\in\mathbb{Q}_+} = \mathbf{1}_{|\mathbb{Q}_+|} \oslash \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right)^\top \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right)\mathbf{1}_{|\mathbb{Q}_+|}\right)\right) \quad (16)$$

On each iteration, the strategy at the next iteration $\boldsymbol{\sigma}' = \left(\sigma^{(T+1)}(q_+)\right)_{q_+\in\mathbb{Q}_+}$ is calculated using $\boldsymbol{\sigma}$.

3.2 ITERATION

3.2.1 TREE TRAVERSAL

Let a dense vector $\boldsymbol{s} \in \mathbb{R}^{\mathbb{V}}$ represent the probabilities of taking an action that reaches a node $v \in \mathbb{V}$ at an iteration T. This value is irrelevant for the unique initial node v_0.

$$\boldsymbol{s} = \left(\begin{cases}\sigma^{(T)}(f_h(f_{Pa}(v)), f_a(v)) & v \in \mathbb{V}_+ \\ 0 & v = v_0\end{cases}\right)_{v\in\mathbb{V}} = \left(\boldsymbol{M}^{(Q_+,V)}\right)^\top \boldsymbol{\sigma} + \boldsymbol{s}^{(\sigma_0)} \quad (17)$$

For later use, we also broadcast the vector $\boldsymbol{s}$ to be a matrix $\boldsymbol{S} \in \mathbb{R}^{\mathbb{V}^2}$. This is defined only for notational convenience and, in our implementation, this matrix is not actually stored in memory.

$$\boldsymbol{S} = (\boldsymbol{s}_{v'})_{(v,v')\in\mathbb{V}^2} \quad (18)$$

The recurrence relations of the expected payoffs of (rational) players $\breve{u} : \Sigma \times \mathbb{V} \times \mathbb{I}_+ \to \mathbb{R}$ (see Equation 1) is expresssed with matrices. Define the dense matrices $\breve{\boldsymbol{U}}^{(1)}, \breve{\boldsymbol{U}}^{(2)}, \ldots, \breve{\boldsymbol{U}}^{(D+1)} \in \mathbb{R}^{\mathbb{V}\times\mathbb{I}_+}$.

$$\forall l \in [1, D+1] \cap \mathbb{Z} \quad \breve{\boldsymbol{U}}^{(l)} = \left(\begin{cases}\breve{u}(\sigma^{(T)}, v, i_+) & d_{\mathcal{T}}(v) \geqslant l-1 \lor v \in \mathbb{T} \\ 0 & d_{\mathcal{T}}(v) < l-1 \land v \in \mathbb{D}\end{cases}\right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} \quad (19)$$

$$\breve{\boldsymbol{U}}^{(D+1)} = \left(\begin{cases}u(v, i_+) & v \in \mathbb{T} \\ 0 & v \in \mathbb{D}\end{cases}\right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} \quad (20)$$

$$\forall l \in [1, D] \cap \mathbb{Z} \quad \breve{\boldsymbol{U}}^{(l)} = \left(\boldsymbol{L}^{(l)} \odot \boldsymbol{S}\right)\breve{\boldsymbol{U}}^{(l+1)} + \breve{\boldsymbol{U}}^{(l+1)} \quad (21)$$

Let a dense matrix $\breve{\boldsymbol{U}} \in \mathbb{R}^{\mathbb{V}\times\mathbb{I}_+}$ represent $\breve{u} : \Sigma \times \mathbb{V} \times \mathbb{I}_+ \to \mathbb{R}$.

$$\breve{\boldsymbol{U}} = \left(\breve{u}(\sigma^{(T)}, v, i_+)\right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} = \breve{\boldsymbol{U}}^{(1)} \quad (22)$$

Let $\breve{\boldsymbol{S}} \in \mathbb{R}^{\mathbb{V}\times\mathbb{I}_+}$ be a dense matrix to be used in a later calculation.

$$\breve{\boldsymbol{S}} = \left(\begin{cases}\boldsymbol{s}_v & \left(\boldsymbol{M}^{(V,I_+)}\right)_{v,i_+} = 0 \\ 1 & \left(\boldsymbol{M}^{(V,I_+)}\right)_{v,i_+} = 1\end{cases}\right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} \quad (23)$$

In order to represent a restriction (ignoring nature) of the "excepted" reach probabilities (defined in Equation 2) $\breve{\pi} : \Sigma \times \mathbb{V} \times \mathbb{I} \to \mathbb{R}$ with matrices, we, again, express the recurrence relations with matrices. We therefore define the following dense matrices: $\breve{\boldsymbol{\Pi}}^{(0)}, \breve{\boldsymbol{\Pi}}^{(1)}, \breve{\boldsymbol{\Pi}}^{(2)}, \ldots, \breve{\boldsymbol{\Pi}}^{(D)} \in \mathbb{R}^{\mathbb{V}\times\mathbb{I}_+}$.

$$\forall l \in [0, D] \cap \mathbb{Z} \quad \breve{\boldsymbol{\Pi}}^{(l)} = \left(\begin{cases}\breve{\pi}(\sigma^{(T)}, v, i_+) & d_{\mathcal{T}}(v) \leqslant l \\ 0 & d_{\mathcal{T}}(v) > l\end{cases}\right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} \quad (24)$$

$$\breve{\boldsymbol{\Pi}}^{(0)} = \left(\mathbf{1}_{v=v_0}\right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} \quad (25)$$

$$\forall l \in [1, D] \cap \mathbb{Z} \quad \breve{\boldsymbol{\Pi}}^{(l)} = \left(\left(\boldsymbol{L}^{(l)}\right)^\top \breve{\boldsymbol{\Pi}}^{(l-1)}\right) \odot \breve{\boldsymbol{S}} + \breve{\boldsymbol{\Pi}}^{(l-1)} \tag{26}$$

Let a dense vector $\breve{\boldsymbol{\pi}} \in \mathbb{R}^{\mathbb{V}}$ be the terms in Equation 3 for "counterfactual" reach probabilities $\tilde{\pi} : \Sigma \times \mathbb{H} \to \mathbb{R}$.

$$\breve{\boldsymbol{\pi}} = \left(\begin{cases} \begin{cases} \tilde{\pi}(\sigma^{(T)}, v, f_i(f_h(f_{Pa}(v)))) & f_h(f_{Pa}(v)) \in \mathbb{H}_+ \\ 0 & f_h(f_{Pa}(v)) \in \mathbb{H}_0 \end{cases} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\right)_{v \in \mathbb{V}} = \left(\boldsymbol{M}^{(V,I_+)} \odot \breve{\boldsymbol{\Pi}}^{(D)}\right) \mathbf{1}_{|\mathbb{I}_+|} \tag{27}$$

3.2.2 Average Strategy Profile

The average strategy profile $\bar{\sigma}^{(T)} : \mathbb{Q} \to \mathbb{R}$ at an iteration T, formulated in Equation 8 and represented as a dense vector $\bar{\boldsymbol{\sigma}} \in \mathbb{R}^{\mathbb{Q}_+}$, can be updated from the previous iteration's $\bar{\sigma}^{(T-1)} : \mathbb{Q} \to \mathbb{R}$, represented as a dense vector $\bar{\boldsymbol{\sigma}}' \in \mathbb{R}^{\mathbb{Q}_+}$. For this, the "counterfactual" reach probabilities $\tilde{\pi} : \Sigma \times \mathbb{H} \to \mathbb{R}$ (Equation 3), a restriction of which is represented by a dense vector $\tilde{\boldsymbol{\pi}} \in \mathbb{R}^{\mathbb{H}_+}$, and their sums, a restriction of which is represented by a dense vector $\tilde{\boldsymbol{\pi}}^{(\Sigma)} \in \mathbb{R}^{\mathbb{H}_+}$, must be calculated. The previous sums of "counterfactual" reach probabilities is denoted as a dense vector $\tilde{\boldsymbol{\pi}}^{(\Sigma)\prime} \in \mathbb{R}^{\mathbb{H}_+}$.

$$\tilde{\boldsymbol{\pi}} = \left(\tilde{\pi}(\sigma^{(T)}, h_+)\right)_{h_+ \in \mathbb{H}_+} = \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right)\left(\boldsymbol{M}^{(Q_+,V)}\right)\breve{\boldsymbol{\pi}}\right) \oslash \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right)\mathbf{1}_{|\mathbb{Q}_+|}\right) \tag{28}$$

$$\tilde{\boldsymbol{\pi}}^{(\Sigma)} = \left(\sum_{\tau=1}^{T} \tilde{\pi}(\sigma^{(\tau)}, h_+)\right)_{h_+ \in \mathbb{H}_+} = \tilde{\boldsymbol{\pi}}^{(\Sigma)\prime} + \tilde{\boldsymbol{\pi}} \tag{29}$$

$$\bar{\boldsymbol{\sigma}} = \left(\bar{\sigma}^{(T)}(q_+)\right)_{q_+ \in \mathbb{Q}_+} = \bar{\boldsymbol{\sigma}}' + \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right)^\top \left(\tilde{\boldsymbol{\pi}} \oslash \tilde{\boldsymbol{\pi}}^{(\Sigma)}\right)\right) \odot \left(\boldsymbol{\sigma} - \bar{\boldsymbol{\sigma}}'\right) \tag{30}$$

3.2.3 Next Strategy Profile

Let a dense vector $\tilde{\boldsymbol{r}} \in \mathbb{R}^{\mathbb{Q}_+}$ represent instantaneous counterfactual regrets $\tilde{r} : \Sigma \times \mathbb{Q}_+ \to \mathbb{R}$, defined in Equation 5, for a strategy profile $\sigma^{(T)}$ at an iteration T.

$$\tilde{\boldsymbol{r}} = \left(\tilde{r}(\sigma^{(T)}, q_+)\right)_{q_+ \in \mathbb{Q}_+} = \left(\boldsymbol{M}^{(Q_+,V)}\right)\left(\breve{\boldsymbol{\pi}} \odot \left(\left(\left(\boldsymbol{M}^{(V,I_+)}\right) \odot \left(\breve{\boldsymbol{U}} - \boldsymbol{G}^\top \breve{\boldsymbol{U}}\right)\right)\mathbf{1}_{|\mathbb{I}_+|}\right)\right) \tag{31}$$

Average counterfactual regrets $\bar{r}^{(T)} : \mathbb{Q}_+ \to \mathbb{R}$ in Equation 6 can be represented with a dense vector $\bar{\boldsymbol{r}} \in \mathbb{R}^{\mathbb{Q}_+}$. Let a dense vector $\bar{\boldsymbol{r}}' \in \mathbb{R}^{\mathbb{Q}_+}$ be the average counterfactual regrets at the previous iteration $\bar{r}^{(T-1)} : \mathbb{Q}_+ \to \mathbb{R}$.

$$\bar{\boldsymbol{r}} = \left(\bar{r}^{(T)}(q_+)\right)_{q_+ \in \mathbb{Q}_+} = \bar{\boldsymbol{r}}' + \frac{1}{T}\left(\tilde{\boldsymbol{r}} - \bar{\boldsymbol{r}}'\right) \tag{32}$$

The clipped regrets are normalized to get a restriction of the next strategy profile $\sigma^{(T+1)} : \mathbb{Q}_+ \to \mathbb{R}$ from Equation 7 for (rational) player information set-action pairs, represented as a dense vector $\boldsymbol{\sigma}'$.

$$\bar{\boldsymbol{r}}^{(+,\Sigma)} = \left(\sum_{a' \in A(h_+)} \left(\bar{r}^{(T)}(h_+, a')\right)^+\right)_{(h_+,a) \in \mathbb{Q}_+} = \left(\boldsymbol{M}^{(H_+,Q_+)}\right)^\top \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right)\bar{\boldsymbol{r}}^+\right) \tag{33}$$

Game (in OpenSpiel)	Average CFR Iteration Runtime (milliseconds)			
	OpenSpiel		Ours	
	Python	C++	CPU	GPU
tiny_hanabi	0.851 (0.00)	**0.035 (0.00)**	0.514 (0.00)	2.269 (0.01)
kuhn_poker	1.011 (0.00)	**0.042 (0.00)**	0.614 (0.00)	2.706 (0.01)
kuhn_poker(players=3)	15.224 (0.01)	**0.725 (0.00)**	1.016 (0.00)	3.828 (0.01)
first_sealed_auction	81.226 (0.02)	3.696 (0.01)	**1.435 (0.00)**	2.829 (0.10)
leduc_poker	153.731 (0.19)	15.444 (0.02)	**2.772 (0.00)**	4.673 (0.01)
tiny_bridge_2p	640.783 (1.57)	37.524 (0.25)	19.355 (0.03)	**4.796 (0.01)**
liars_dice	1351.281 (8.39)	98.109 (0.79)	78.017 (0.08)	**7.660 (0.02)**
tic_tac_toe	2629.924 (11.04)	165.389 (0.78)	119.713 (0.14)	**7.458 (0.01)**

Table 1: The average per-iteration runtimes (and the standard errors of the means, in brackets) of CFR implementations: reference OpenSpiel's (Lanctot et al., 2020) and ours (with a CPU or a GPU). The performances of the fastest implementation for each game are bolded. The games are sorted by the number of nodes in the game tree and their names in the first column correspond exactly to the game name in Deepmind's OpenSpiel (Lanctot et al., 2020) library. A similar table showing speedups or slowdowns are shown in Appendix C.

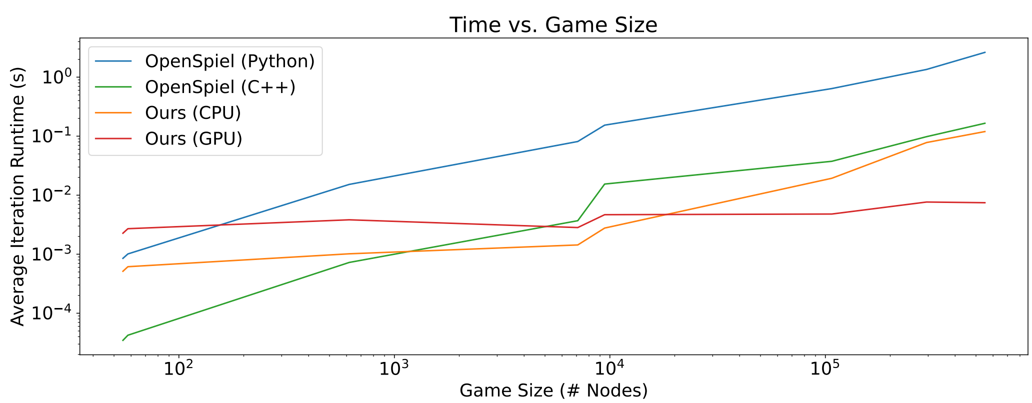

Figure 1: A log-log graph showing the average CFR iteration runtime with respect to the game size. The four lines show the runtimes of Deepmind's OpenSpiel (Lanctot et al., 2020) CFR implementation in Python and C++ and our implementation with a CPU or GPU backend.

$$\boldsymbol{\sigma}' = \left(\sigma^{(T+1)}(q_+)\right)_{q_+\in\mathbb{Q}_+} = \left(\begin{cases}\left(\overline{\boldsymbol{r}}^+ \oslash \overline{\boldsymbol{r}}^{(+,\Sigma)}\right)_{q_+} & \left(\overline{\boldsymbol{r}}^{(+,\Sigma)}\right)_{q_+} > 0\\ \left(\boldsymbol{\sigma}^{(T=1)}\right)_{q_+} & \left(\overline{\boldsymbol{r}}^{(+,\Sigma)}\right)_{q_+} = 0\end{cases}\right)_{q_+\in\mathbb{Q}_+} \tag{34}$$

4 BENCHMARKS

We run 1,000 CFR iterations on 8 games of varying sizes implemented in Google DeepMind's OpenSpiel (Lanctot et al., 2020) (see Appendix A for more details) using their Python and C++ CFR implementations and our implementations (with a CPU or GPU backend). The games represent a diverse range of sizes from small (tiny Hanabi and Kuhn poker), medium (Kuhn poker (3-player), first sealed auction, and Leduc poker), to large (tiny bridge (2-player), liar's dice, and tic-tac-toe).

In our GPU implementation (written in Python), we use CuPy (Okuta et al., 2017) for GPU-accelerated matrix and vector operations. For parity with OpenSpiel (Lanctot et al., 2020), our implementation uses double-precision floating point numbers (64-bit float) and do not leverage tensor cores. We also simply run our implementation with NumPy (Harris et al., 2020) and SciPy (Virtanen et al., 2020) (i.e. without a GPU) which we refer to as our CPU implementation. Our testbench computer contains an AMD Ryzen 9 3900X 12-core, 24-thread desktop processor, 128 GB memory, and Nvidia GeForce RTX 4090 24 GB VRAM graphics card.

The results are tabulated in Table 1 and plotted in Appenix B. The results vary depending on the size of the game being played. The relationship between the game sizes and the runtimes of each implementation is shown more clearly in the log-log graph in Figure 1. Note that our GPU implementation clearly scales better than both OpenSpiel's (Lanctot et al., 2020) and our CPU implementation.

4.1 Small Games: Tiny Hanabi and Kuhn Poker

In small games like tiny Hanabi (55 nodes) and Kuhn poker (58 nodes), our CPU implementation shows modest gains over the OpenSpiel's (Lanctot et al., 2020) Python baseline (about 1.7 times faster for both). However, our GPU implementation is actually about 2.7 times slower for both compared to OpenSpiel's Python baseline. OpenSpiel's C++ baseline vastly outperforms all others by at least an order of magnitude. This suggests the overheads from GPU and Python make our implementation impractical for games of similarly small sizes.

4.2 Medium Games: Kuhn Poker (3-Player), First Sealed Auction, and Leduc Poker

In medium-sized games like Kuhn poker (3-player) (617 nodes), first sealed auction (7,096 nodes), and Leduc poker (9,457 nodes), performance gains compared to OpenSpiel's (Lanctot et al., 2020) Python implementation can be observed for both our CPU (about 14.9, 56.4, and 55.5 times faster, respectively) and GPU implementation (about 4.0, 28.7, and 32.9 times faster, respectively). However, comparisons with OpenSpiel's C++ implementation is mixed. For Kuhn poker (3-player), OpenSpiel's C++ implementation is about 1.4 times faster than our CPU implementation and 5.3 times faster than our GPU implementation. But, for first sealed auction and Leduc poker, our CPU implementation is about 2.6 and 5.6 times faster, respectively, and our GPU implementation is about 1.3 and 3.3 times faster, respectively, than their C++ baseline. Here, while we begin to see our implementations outperform OpenSpiel's baselines, we see that our CPU implementation is faster than our GPU implementation. This suggests that, while the efficiency of our implementation overcomes the Python overhead, the remaining GPU overhead makes using a GPU less preferable than not.

4.3 Large Games: Tiny Bridge (2-Player), Liar's Dice, and Tic-Tac-Toe

In games like tiny bridge (2-player) (107,129 nodes), liar's dice (294,883 nodes), and tic-tac-toe (549,946 nodes), noticeable performance gains over OpenSpiel's (Lanctot et al., 2020) Python implementation can be observed for both our CPU (about 33.1, 17.3, and 22.0 times faster, respectively) and GPU implementation (about 133.5, 176.4, and 352.5 times faster, respectively). The same can be said for OpenSpiel's C++ implementation to a lesser degree: our CPU implementation is about 1.9, 1.3, and 1.4 times faster, respectively, and our GPU implementation is about 7.8, 12.8, and 22.2 times faster, respectively. Here, the performance benefits of utilizing a GPU is clear, and we predict that the differences will be even more pronounced for games of sizes larger than the ones explored.

The total allocated CUDA memory by our GPU implementation to solve each game is plotted in Figure 2, and the peak memory usages of the benchmark scripts are shown in Table 2. Note that this is not a fair comparison, as, in our implementations, we unnecessarily store the object representations of all states. By not doing so, further reduction in process memory usage would be possible.

5 Discussion

In this work, we only explore parallelizing the vanilla CFR algorithm, as proposed by Zinkevich et al. (2007). Later variants of CFR show improvements, namely in convergence speeds, which modify various aspects of the algorithm. The discounting techniques proposed by Brown and Sandholm can trivially be applied by altering Equation 32 and Equation 30. However, pruning techniques (Brown & Sandholm, 2015) would require non-trivial manipulations on the game-related matrices – possibly between iterations – problematic since updating certain types of sparse matrices like the CSR format we use is computationally expensive.

Unlike sampling variants of CFR (Lanctot et al., 2009), on each iteration, our implementation deals with the entire game tree and stores values for every node – impractical for extremely large games. In traditional implementations of CFR, while a complete recursive game tree traversal is carried

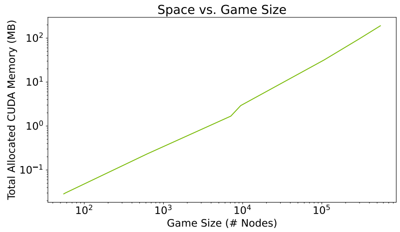

Figure 2: A log-log graph showing the total allocated CUDA memory by our GPU implementation with respect to the game size. The values are tabulated in Appendix D.

Implementation			Peak Memory Usage (GB)
OpenSpiel	Python		0.894
	C++		0.145
Ours	CPU		2.759
	GPU	Process	3.064
		CUDA	0.240

Table 2: The peak memory usages of the benchmark scripts of the 4 CFR implementations: OpenSpiel baselines (Lanctot et al., 2020) (in Python and C++) and ours (with a CPU or a GPU backend). For our GPU implementation, we show both the peak memory usage of the benchmark process and the total memory allocated by CuPy (Okuta et al., 2017) in the CUDA memory pool.

out, counterfactual values are typically not stored for each node but instead for each information set-actions. We demonstrate that it is possible to achieve a significant parallelization (and hence speedup) at a cost of higher memory usage. Intuitively, the root-to-leaf paths can be partitioned to construct subgraphs of which separate adjacency and submask matrices can be loaded and applied as necessary – a similar approach can be used for alternating player updates (Burch et al., 2019).

Our approach provides an alternate way for CFR to be run on supercomputers. During the development of Cepheus (Tammelin et al., 2015), the game tree was chunked into a trunk and many subtrees, each of which was assigned to a compute node to be traversed independently. This introduced a bottleneck in the trunk as the subtree nodes (which depend on the trunk's results) must wait for the trunk calculation to complete during the downward pass, and wait again while the trunk uses the values returned by the subtrees during the upward pass. Our approach is simply a series of matrix/vector operations, and distributing this is a well-studied problem in systems.

In our GPU implementation, we used CuPy (Okuta et al., 2017) without any customizations in configurations and did not profile or probe into resource usages. A careful analysis of these for further optimizations will most likely yield further performance improvements.

6 Conclusion

We introduced our CFR implementation, designed to be highly parallelized by computing each iteration as dense and sparse matrix and vector operations and eliminating costly recursive tree traversal. While our goal was to run the algorithm on a GPU, the tight nature of our code also allows for a vastly more efficient computation even when a GPU is not leveraged. Our experiments on solving 8 games of differing sizes show that, in larger games, our implementation achieves orders of magnitude performance improvements over Google DeepMind's OpenSpiel (Lanctot et al., 2020) baselines in Python and C++, and predict that the performance benefit will be even more pronounced for games of sizes larger than those we tested. Addressing the memory inefficiency and incorporating the use of a GPU with non-vanilla CFR variants remains a promising avenue for future research.

References

URL `https://docs.nvidia.com/cuda/cusparse/index.html`.

Noam Brown and Tuomas Sandholm. Regret-based pruning in extensive-form games. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL `https://proceedings.neurips.cc/paper_files/paper/2015/file/c54e7837e0cd0ced286cb5995327d1ab-Paper.pdf`.

Noam Brown and Tuomas Sandholm. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018. doi: 10.1126/science.aao1733. URL `https://www.science.org/doi/abs/10.1126/science.aao1733`.

Noam Brown and Tuomas Sandholm. Solving imperfect-information games via discounted regret minimization. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019a. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33011829. URL https://doi.org/10.1609/aaai.v33i01.33011829.

Noam Brown and Tuomas Sandholm. Superhuman ai for multiplayer poker. *Science*, 365(6456): 885–890, 2019b. doi: 10.1126/science.aay2400. URL https://www.science.org/doi/abs/10.1126/science.aay2400.

Noam Brown, Anton Bakhtin, Adam Lerer, and Qucheng Gong. Combining deep reinforcement learning and search for imperfect-information games. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 17057–17069. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/c61f571dbd2fb949d3fe5ae1608dd48b-Paper.pdf.

Neil Burch, Michael Johanson, and Michael Bowling. Solving imperfect information games using decomposition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 28(1), Jun. 2014. doi: 10.1609/aaai.v28i1.8810. URL https://ojs.aaai.org/index.php/AAAI/article/view/8810.

Neil Burch, Matej Moravcik, and Martin Schmid. Revisiting cfr+ and alternating updates. *J. Artif. Int. Res.*, 64(1):429–443, jan 2019. ISSN 1076-9757. doi: 10.1613/jair.1.11370. URL https://doi.org/10.1613/jair.1.11370.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/2999445.

Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. Monte carlo sampling for regret minimization in extensive games. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta (eds.), *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper_files/paper/2009/file/00411460f7c92d2124a67ea0f4cb5f85-Paper.pdf.

Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, Daniel Hennes, Dustin Morrill, Paul Muller, Timo Ewalds, Ryan Faulkner, János Kramár, Bart De Vylder, Brennan Saeta, James Bradbury, David Ding, Sebastian Borgeaud, Matthew Lai, Julian Schrittwieser, Thomas Anthony, Edward Hughes, Ivo Danihelka, and Jonah Ryan-Davis. Openspiel: A framework for reinforcement learning in games, 2020. URL https://arxiv.org/abs/1908.09453.

Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017. doi: 10.1126/science.aam6960. URL https://www.science.org/doi/abs/10.1126/science.aam6960.

John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with cuda: Is cuda the parallel programming model that application developers have been waiting for? *Queue*, 6(2):40–53, mar 2008. ISSN 1542-7730. doi: 10.1145/1365490.1365500. URL https://doi.org/10.1145/1365490.1365500.

Ryosuke Okuta, Yuya Unno, Daisuke Nishino, Shohei Hido, and Crissman Loomis. Cupy: A numpy-compatible library for nvidia gpu calculations. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*, 2017. URL http://learningsys.org/nips17/assets/papers/paper_16.pdf.

M.J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994. ISBN 9780262650403. URL https://books.google.ca/books?id=PuSMEAAAQBAJ.

João Reis. *A GPU implementation of Counterfactual Regret Minimization*. PhD thesis, Master Thesis, University of Porto, 2015. URL https://repositorio-aberto.up.pt/handle/10216/83517.

Jan Rudolf. *Counterfactual Regret Minimization on GPU*. PhD thesis, Czech Technical University in Prague, Jan. 2021. URL https://cent.felk.cvut.cz/courses/GPU/archives/2020-2021/W/rudolja1/.

Martin Schmid, Matej Moravčík, Neil Burch, Rudolf Kadlec, Josh Davidson, Kevin Waugh, Nolan Bard, Finbarr Timbers, Marc Lanctot, G. Zacharias Holland, Elnaz Davoodi, Alden Christianson, and Michael Bowling. Student of games: A unified learning algorithm for both perfect and imperfect information games. *Science Advances*, 9(46):eadg3256, 2023. doi: 10.1126/sciadv.adg3256. URL https://www.science.org/doi/abs/10.1126/sciadv.adg3256.

Jack Serrino, Max Kleiman-Weiner, David C Parkes, and Josh Tenenbaum. Finding friend and foe in multi-agent games. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/912d2b1c7b2826caf99687388d2e8f7c-Paper.pdf.

Oskari Tammelin. Solving large imperfect information games using cfr+, 2014. URL https://arxiv.org/abs/1407.5042.

Oskari Tammelin, Neil Burch, Michael Johanson, and Michael Bowling. Solving heads-up limit texas hold'em. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pp. 645–652. AAAI Press, 2015. ISBN 9781577357384.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Hang Xu, Kai Li, Haobo Fu, QIANG FU, Junliang Xing, and Jian Cheng. Dynamic discounted counterfactual regret minimization. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=6PbvbLyqT6.

Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In J. Platt, D. Koller, Y. Singer, and S. Roweis (eds.), *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL https://proceedings.neurips.cc/paper_files/paper/2007/file/08d98638c6fcd194a4b1e6992063e944-Paper.pdf.

A GAME PROPERTIES

Table 3 give details (e.g. number of nodes, terminal nodes, information sets, actions, and players) about the games we solve during our experimentation, and Table 4 shows the sparsities of the mask matrices when the discrete games we explore are converted into our desired format.

Game (in OpenSpiel)	# Nodes	# Terminals	# Infosets	# Actions	# Players
`tiny_hanabi`	55	36	8	3	2
`kuhn_poker`	58	30	12	3	2
`kuhn_poker(players=3)`	617	312	48	4	3
`first_sealed_auction`	7,096	3,410	20	11	2
`leduc_poker`	9,457	5,520	936	6	2
`tiny_bridge_2p`	107,129	53,340	3,584	28	2
`liars_dice`	294,883	147,420	24,576	13	2
`tic_tac_toe`	549,946	255,168	294,778	9	2

Table 3: The 8 games tested in our benchmark and relevant statistics: number of nodes, terminal nodes, information sets, actions, and (rational) players. The games are sorted by the number of nodes in the game tree and their names in the first column correspond exactly to the game name in Deepmind's OpenSpiel (Lanctot et al., 2020) library.

Game (in OpenSpiel)	Sparsities (%)			
	$\boldsymbol{M}^{(Q_+,V)}$	$\boldsymbol{M}^{(H_+,Q_+)}$	$\boldsymbol{L}^{(l)}$ (Average)	$\boldsymbol{G}$
`tiny_hanabi`	96.4	87.5	99.6	98.2
`kuhn_poker`	96.6	91.7	99.7	98.3
`kuhn_poker(players=3)`	99.0	97.9	99.9+	99.8
`first_sealed_auction`	99.5	95.0	99.9+	99.9+
`leduc_poker`	99.9+	99.9	99.9+	99.9+
`tiny_bridge_2p`	99.9+	99.9+	99.9+	99.9+
`liars_dice`	99.9+	99.9+	99.9+	99.9+
`tic_tac_toe`	99.9+	99.9+	99.9+	99.9+

Table 4: The sparsities of sparse matrix constants in our implementation. The entries in the leftmost column correspond exactly to the game name in Deepmind's OpenSpiel (Lanctot et al., 2020) library. CUDA's (Nickolls et al., 2008) cuSPARSE "library targets matrices with sparsity ratios in the range between 70%-99.9%" (cuS). Our values fall under this recommended range. We project that the matrices for games not tested in our work will typically have similar sparsity values as those we test.

B Plots of Runtimes of the Tested CFR Implementations

The pairs of plots for each game tested showing the runtimes for up to 1,000 iterations and a bar graph showing the average runtimes per iteration for the four implementations tested are shown in Figure 3.

C Speedups and slowdowns

The speedups or slowdowns of our implementation (with a GPU or a CPU backend) compared to OpenSpiel's baselines (Python or C++ implementation) is tabulated in Table 5.

D Total Allocated CUDA Memory

The total allocated CUDA memory by CuPy (Okuta et al., 2017) in our GPU implementation to solve each game through CFR is tabulated in Table 6.

E Game Tree Setup

In order to use our implementation, the game tree must be transformed into sparse matrices encoding the game rules, which requires a single complete game tree traversal. Note that this is a one-time operation performed prior to running CFR. Table 7 shows the time it takes to serialize each discrete game from OpenSpiel Lanctot et al. (2020).

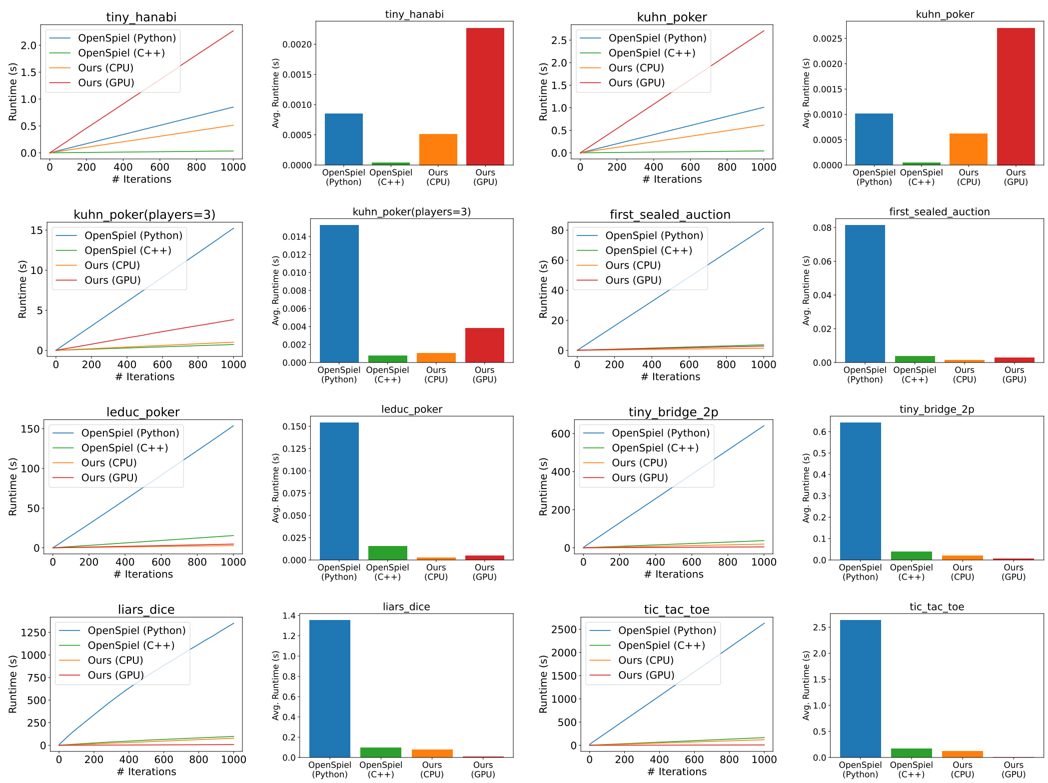

Figure 3: Pairs of plots for each game tested showing the runtimes for up to 1,000 iterations and a bar graph showing the average runtimes per iteration for four implementations of CFR: Deepmind's OpenSpiel (Lanctot et al., 2020) CFR implementation in Python and C++ and our implementation with a CPU or GPU backend.

Game (in OpenSpiel)	Average Speedup or Slowdown (times)			
	OpenSpiel's Python		OpenSpiel's C++	
	Our CPU	Our GPU	Our CPU	Our GPU
`tiny_hanabi`	1.7	-2.7	-17.0	-75.7
`kuhn_poker`	1.7	-2.7	-15.2	-67.8
`kuhn_poker(players=3)`	14.9	4.0	-1.4	-5.3
`first_sealed_auction`	56.4	28.7	2.6	1.3
`leduc_poker`	55.5	32.9	5.6	3.3
`tiny_bridge_2p`	33.1	133.5	1.9	7.8
`liars_dice`	17.3	176.4	1.3	12.8
`tic_tac_toe`	22.0	352.5	1.4	22.2

Table 5: The average per-iteration speedups or slowdowns in runtimes of our CFR implementations over reference OpenSpiel's (Lanctot et al., 2020). The positive values represent speedups and the negative values represent the slowdowns. The games are sorted by the number of nodes in the game tree and their names in the first column correspond exactly to the game name in Deepmind's OpenSpiel (Lanctot et al., 2020) library. A similar table showing the original raw runtime values are shown in Table 1.

Game (in OpenSpiel)	Total Allocated CUDA Memory (MB)
`tiny_hanabi`	0.029
`kuhn_poker`	0.030
`kuhn_poker(players=3)`	0.231
`first_sealed_auction`	1.683
`leduc_poker`	2.920
`tiny_bridge_2p`	31.899
`liars_dice`	95.294
`tic_tac_toe`	190.865

Table 6: The total allocated CUDA memory during CFR iterations for each game experimented on. The games are sorted by the number of nodes in the game tree and their names in the first column correspond exactly to the game name in Deepmind's OpenSpiel (Lanctot et al., 2020) library. A similar table showing the original raw runtime values are shown in Table 1.

F EXPANDED EQUATIONS

Subsections G.1, G.2, G.3, G.4, G.5, G.6, G.7, G.8, G.9, G.10, G.11, and G.12 show expanded forms of equations shown in Section 3.

G REPRODUCIBILITY OF REIS'S MASTER'S THESIS

Reis's thesis (Reis, 2015) contains screenshots of his code as figures that cannot compile due to syntax errors. For example, we point out the missing semicolon in Line 4 of Figure 12 and the mismatched square brace in Line 8 of Figure 18. Aside from the obvious errors, the thesis also omits details about the calculations of counterfactual regrets, strategy profiles, and counterfactual reach probabilities, and does not handle chance nodes, decision nodes, and terminal nodes separately. We doubt that his work can be reproduced to work in practice without significant work.

G.1 INITIAL STRATEGY PROFILE

An expanded form of Equation 16 is shown below.

$$\boldsymbol{\sigma}^{(T=1)} = \left(\sigma^{(1)}(q_+)\right)_{q_+ \in \mathbb{Q}_+}$$

Game (in OpenSpiel)	Setup Time (seconds)
`tiny_hanabi`	0.183
`kuhn_poker`	0.016
`kuhn_poker(players=3)`	0.074
`first_sealed_auction`	1.199
`leduc_poker`	1.110
`tiny_bridge_2p`	14.855
`liars_dice`	40.835
`tic_tac_toe`	73.037

Table 7: The time it took to convert OpenSpiel's (Lanctot et al., 2020) discrete game into sparse matrices.

$$
\begin{aligned}
&= \left(\frac{1}{|A(h_+)|}\right)_{(h_+,a)\in\mathbb{Q}_+} \\
&= \mathbf{1}_{|\mathbb{Q}_+|} \oslash \left(|A(h_+)|\right)_{(h_+,a)\in\mathbb{Q}_+} \\
&= \mathbf{1}_{|\mathbb{Q}_+|} \oslash \left(\sum_{h'_+\in\mathbb{H}_+} \left(\mathbf{1}_{h_+=h'_+}\right) |A(h'_+)|\right)_{(h_+,a)\in\mathbb{Q}_+} \\
&= \mathbf{1}_{|\mathbb{Q}_+|} \oslash \left(\left(\left(\mathbf{1}_{h_+=h'_+}\right)_{((h_+,a),h'_+)\in\mathbb{Q}_+\times\mathbb{H}_+}\right) \left(|A(h_+)|\right)_{h_+\in\mathbb{H}_+}\right) \\
&= \mathbf{1}_{|\mathbb{Q}_+|} \oslash \left(\left(\left(\mathbf{1}_{h_+=h'_+}\right)_{(h_+,(h'_+,a))\in\mathbb{H}_+\times\mathbb{Q}_+}\right)^\top \left(|A(h_+)|\right)_{h_+\in\mathbb{H}_+}\right)
\end{aligned}
$$

Using Equation 12

$$
\begin{aligned}
&= \mathbf{1}_{|\mathbb{Q}_+|} \oslash \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right)^\top \left(|A(h_+)|\right)_{h_+\in\mathbb{H}_+}\right) \\
&= \mathbf{1}_{|\mathbb{Q}_+|} \oslash \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right)^\top \left(\sum_{(h'_+,a)\in\mathbb{Q}_+} \mathbf{1}_{h_+=h'_+}\right)_{h_+\in\mathbb{H}_+}\right) \\
&= \mathbf{1}_{|\mathbb{Q}_+|} \oslash \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right)^\top \left(\left(\mathbf{1}_{h_+=h'_+}\right)_{(h_+,(h'_+,a))\in\mathbb{H}_+\times\mathbb{Q}_+}\right) \mathbf{1}_{|\mathbb{Q}_+|}\right)
\end{aligned}
$$

Using Equation 12

$$
= \mathbf{1}_{|\mathbb{Q}_+|} \oslash \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right)^\top \left(\boldsymbol{M}^{(H_+,Q_+)}\right) \mathbf{1}_{|\mathbb{Q}_+|}\right)
$$

To take advantage of the sparsity of $\boldsymbol{M}^{(H_+,Q_+)}$ (see Table 4)

$$
= \mathbf{1}_{|\mathbb{Q}_+|} \oslash \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right)^\top \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right) \mathbf{1}_{|\mathbb{Q}_+|}\right)\right)
$$

G.2 Strategies

An expanded form of Equation 17 is shown below.

$$\boldsymbol{s} = \left(\begin{cases} \sigma^{(T)}(f_h(f_{Pa}(v)), f_a(v)) & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases} \right)_{v \in \mathbb{V}}$$
$$= \left(\begin{cases} \begin{cases} \sigma^{(T)}(f_h(f_{Pa}(v)), f_a(v)) & f_h(f_{Pa}(v)) \in \mathbb{H}_+ \\ 0 & f_h(f_{Pa}(v)) \in \mathbb{H}_0 \end{cases} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases} \right)_{v \in \mathbb{V}}$$
$$+ \left(\begin{cases} \begin{cases} \sigma_0(f_h(f_{Pa}(v)), f_a(v)) & f_h(f_{Pa}(v)) \in \mathbb{H}_0 \\ 0 & f_h(f_{Pa}(v)) \in \mathbb{H}_+ \end{cases} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases} \right)_{v \in \mathbb{V}}$$

Using Equation 14

$$= \left(\begin{cases} \begin{cases} \sigma^{(T)}(f_h(f_{Pa}(v)), f_a(v)) & f_h(f_{Pa}(v)) \in \mathbb{H}_+ \\ 0 & f_h(f_{Pa}(v)) \in \mathbb{H}_0 \end{cases} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases} \right)_{v \in \mathbb{V}} + \boldsymbol{s}^{(\sigma_0)}$$
$$= \left(\begin{cases} \sum_{h_+ \in \mathbb{H}_+} \left(\mathbf{1}_{h_+ = f_h(f_{Pa}(v))} \right) \sigma^{(T)}(h_+, f_a(v)) & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases} \right)_{v \in \mathbb{V}} + \boldsymbol{s}^{(\sigma_0)}$$
$$= \left(\begin{cases} \sum_{q_+ \in \mathbb{Q}_+} \left(\mathbf{1}_{q_+ = (f_h(f_{Pa}(v)), f_a(v))} \right) \sigma^{(T)}(q_+) & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases} \right)_{v \in \mathbb{V}} + \boldsymbol{s}^{(\sigma_0)}$$
$$= \left(\left(\begin{cases} \mathbf{1}_{q_+ = (f_h(f_{Pa}(v)), f_a(v))} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases} \right)_{(v, q_+) \in \mathbb{V} \times \mathbb{Q}_+} \right) \left(\sigma^{(T)}(q_+) \right)_{q_+ \in \mathbb{Q}_+} + \boldsymbol{s}^{(\sigma_0)}$$
$$= \left(\left(\begin{cases} \mathbf{1}_{q_+ = (f_h(f_{Pa}(v)), f_a(v))} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases} \right)_{(q_+, v) \in \mathbb{Q}_+ \times \mathbb{V}} \right)^\top \left(\sigma^{(T)}(q_+) \right)_{q_+ \in \mathbb{Q}_+} + \boldsymbol{s}^{(\sigma_0)}$$

Using Equation 11 and Equation 15

$$= \left(\boldsymbol{M}^{(Q_+, V)} \right)^\top \boldsymbol{\sigma} + \boldsymbol{s}^{(\sigma_0)}$$

G.3 Expected Payoffs

G.3.1 Initial Condition

An expanded form of Equation 20 is shown below.

$$\breve{\boldsymbol{U}}^{(D+1)} = \left(\left(\begin{cases} \breve{u}(\sigma^{(T)}, v, i_+) & d_{\mathcal{T}}(v) \geqslant l - 1 \lor v \in \mathbb{T} \\ 0 & d_{\mathcal{T}}(v) < l - 1 \land v \in \mathbb{D} \end{cases} \right)_{(v, i_+) \in \mathbb{V} \times \mathbb{I}_+} \right) \Bigg|_{l = D+1}$$
$$= \left(\begin{cases} \breve{u}(\sigma^{(T)}, v, i_+) & d_{\mathcal{T}}(v) \geqslant D \lor v \in \mathbb{T} \\ 0 & d_{\mathcal{T}}(v) < D \land v \in \mathbb{D} \end{cases} \right)_{(v, i_+) \in \mathbb{V} \times \mathbb{I}_+}$$

Since $\forall d \in \mathbb{D} \quad d_{\mathcal{T}}(d) < D = \max_{t \in \mathbb{T}} d_{\mathcal{T}}(t)$

$$= \left(\begin{cases} \check{u}(\sigma^{(T)}, v, i_+) & v \in \mathbb{T} \\ 0 & v \in \mathbb{D} \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+}$$

Using Equation 1

$$= \left(\begin{cases} \begin{cases} \sum_{s\in S(v)} \sigma^{(T)}(f_h(v), f_a(s))\check{u}(\sigma^{(T)}, s, i_+) & v \in \mathbb{D} \\ u(v, i_+) & v \in \mathbb{T} \end{cases} & v \in \mathbb{T} \\ 0 & v \in \mathbb{D} \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+}$$

$$= \left(\begin{cases} u(v, i_+) & v \in \mathbb{T} \\ 0 & v \in \mathbb{D} \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+}$$

G.3.2 RECURRENCE

An expanded form of Equation 21 is shown below.

$\forall l \in [1, D] \cap \mathbb{Z}$

$$\begin{aligned}
\check{\boldsymbol{U}}^{(l)} &= \left(\begin{cases} \check{u}(\sigma^{(T)}, v, i_+) & d_{\mathcal{T}}(v) \geqslant l-1 \vee v \in \mathbb{T} \\ 0 & d_{\mathcal{T}}(v) < l-1 \wedge v \in \mathbb{D} \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} \\
&= \left(\begin{cases} \check{u}(\sigma^{(T)}, v, i_+) & d_{\mathcal{T}}(v) = l-1 \wedge v \in \mathbb{D} \\ 0 & d_{\mathcal{T}}(v) \neq l-1 \vee v \in \mathbb{T} \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} \\
&\quad + \left(\begin{cases} \check{u}(\sigma^{(T)}, v, i_+) & (d_{\mathcal{T}}(v) \neq l-1 \vee v \in \mathbb{T}) \wedge (d_{\mathcal{T}}(v) \geqslant l-1 \vee v \in \mathbb{T}) \\ 0 & (d_{\mathcal{T}}(v) = l-1 \wedge v \in \mathbb{D}) \vee (d_{\mathcal{T}}(v) < l-1 \wedge v \in \mathbb{D}) \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} \\
&= \left(\begin{cases} \check{u}(\sigma^{(T)}, v, i_+) & d_{\mathcal{T}}(v) = l-1 \wedge v \in \mathbb{D} \\ 0 & d_{\mathcal{T}}(v) \neq l-1 \vee v \in \mathbb{T} \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} \\
&\quad + \left(\begin{cases} \check{u}(\sigma^{(T)}, v, i_+) & d_{\mathcal{T}}(v) \geqslant l \vee v \in \mathbb{T} \\ 0 & d_{\mathcal{T}}(v) < l \wedge v \in \mathbb{D} \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+}
\end{aligned}$$

Using Equation 19

$$= \left(\begin{cases} \check{u}(\sigma^{(T)}, v, i_+) & d_{\mathcal{T}}(v) = l-1 \wedge v \in \mathbb{D} \\ 0 & d_{\mathcal{T}}(v) \neq l-1 \vee v \in \mathbb{T} \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} + \check{\boldsymbol{U}}^{(l+1)}$$

Using Equation 1

$$\begin{aligned}
&= \left(\begin{cases} \begin{cases} \sum_{s\in S(v)} \sigma^{(T)}(f_h(v), f_a(s))\check{u}(\sigma^{(T)}, s, i_+) & v \in \mathbb{D} \\ u(v, i_+) & v \in \mathbb{T} \end{cases} & d_{\mathcal{T}}(v) = l-1 \wedge v \in \mathbb{D} \\ 0 & d_{\mathcal{T}}(v) \neq l-1 \vee v \in \mathbb{T} \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} \\
&\quad + \check{\boldsymbol{U}}^{(l+1)}
\end{aligned}$$

$$
\begin{aligned}
&= \left(\begin{cases} \sum_{s \in S(v)} \sigma^{(T)}(f_h(v), f_a(s)) \check{u}(\sigma^{(T)}, s, i_+) & d_{\mathcal{T}}(v) = l-1 \wedge v \in \mathbb{D} \\ 0 & d_{\mathcal{T}}(v) \neq l-1 \vee v \in \mathbb{T} \end{cases} \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} \\
&\quad + \check{\boldsymbol{U}}^{(l+1)} \\
&= \left(\begin{cases} \sum_{s \in S(v)} \left(\mathbf{1}_{d_{\mathcal{T}}(s)=l}\right) \sigma^{(T)}(f_h(v), f_a(s)) \check{u}(\sigma^{(T)}, s, i_+) & v \in \mathbb{D} \\ 0 & v \in \mathbb{T} \end{cases} \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} + \check{\boldsymbol{U}}^{(l+1)} \\
&= \left(\sum_{v' \in \mathbb{V}} \begin{cases} \left(\mathbf{1}_{v=f_{Pa}(v') \wedge d_{\mathcal{T}}(v')=l}\right) \sigma^{(T)}(f_h(v), f_a(v')) \check{u}(\sigma^{(T)}, v', i_+) & v \in \mathbb{D} \wedge v' \in \mathbb{V}_+ \\ 0 & v \in \mathbb{T} \vee v' = v_0 \end{cases} \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} \\
&\quad + \check{\boldsymbol{U}}^{(l+1)} \\
&= \left(\sum_{v' \in \mathbb{V}} \left(\begin{cases} \left(\mathbf{1}_{v=f_{Pa}(v') \wedge d_{\mathcal{T}}(v')=l}\right) \sigma^{(T)}(f_h(v), f_a(v')) & v \in \mathbb{D} \wedge v' \in \mathbb{V}_+ \\ 0 & v \in \mathbb{T} \vee v' = v_0 \end{cases} \right) \right. \\
&\qquad \left. \left(\begin{cases} \check{u}(\sigma^{(T)}, v', i_+) & d_{\mathcal{T}}(v') \geqslant l \vee v' \in \mathbb{T} \\ 0 & d_{\mathcal{T}}(v') < l \wedge v' \in \mathbb{D} \end{cases} \right) \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} + \check{\boldsymbol{U}}^{(l+1)} \\
&= \left(\begin{cases} \left(\mathbf{1}_{v=f_{Pa}(v') \wedge d_{\mathcal{T}}(v')=l}\right) \sigma^{(T)}(f_h(v), f_a(v')) & v \in \mathbb{D} \wedge v' \in \mathbb{V}_+ \\ 0 & v \in \mathbb{T} \vee v' = v_0 \end{cases} \right)_{(v,v') \in \mathbb{V}^2} \\
&\qquad \left(\begin{cases} \check{u}(\sigma^{(T)}, v, i_+) & d_{\mathcal{T}}(v) \geqslant l \vee v \in \mathbb{T} \\ 0 & d_{\mathcal{T}}(v) < l \wedge v \in \mathbb{D} \end{cases} \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} + \check{\boldsymbol{U}}^{(l+1)}
\end{aligned}
$$

Using Equation 19

$$
\begin{aligned}
&= \left(\left(\begin{cases} \left(\mathbf{1}_{v=f_{Pa}(v') \wedge d_{\mathcal{T}}(v')=l}\right) \sigma^{(T)}(f_h(v), f_a(v')) & v \in \mathbb{D} \wedge v' \in \mathbb{V}_+ \\ 0 & v \in \mathbb{T} \vee v' = v_0 \end{cases} \right)_{(v,v') \in \mathbb{V}^2} \right) \check{\boldsymbol{U}}^{(l+1)} + \check{\boldsymbol{U}}^{(l+1)} \\
&= \left(\left(\begin{cases} \left(\mathbf{1}_{v=f_{Pa}(v') \wedge d_{\mathcal{T}}(v')=l}\right) \sigma^{(T)}(f_h(f_{Pa}(v')), f_a(v')) & v \in \mathbb{D} \wedge v' \in \mathbb{V}_+ \\ 0 & v \in \mathbb{T} \vee v' = v_0 \end{cases} \right)_{(v,v') \in \mathbb{V}^2} \right) \check{\boldsymbol{U}}^{(l+1)} \\
&\quad + \check{\boldsymbol{U}}^{(l+1)}
\end{aligned}
$$

Using Equation 17

$$
\begin{aligned}
&= \left(\left(\begin{cases} \left(\mathbf{1}_{v=f_{Pa}(v') \wedge d_{\mathcal{T}}(v')=l}\right) \boldsymbol{s}_{v'} & v \in \mathbb{D} \wedge v' \in \mathbb{V}_+ \\ 0 & v \in \mathbb{T} \vee v' = v_0 \end{cases} \right)_{(v,v') \in \mathbb{V}^2} \right) \check{\boldsymbol{U}}^{(l+1)} + \check{\boldsymbol{U}}^{(l+1)} \\
&= \left(\left(\left(\begin{cases} \left(\mathbf{1}_{v=f_{Pa}(v') \wedge d_{\mathcal{T}}(v')=l}\right) & v \in \mathbb{D} \wedge v' \in \mathbb{V}_+ \\ 0 & v \in \mathbb{T} \vee v' = v_0 \end{cases} \right) \boldsymbol{s}_{v'} \right)_{(v,v') \in \mathbb{V}^2} \right) \check{\boldsymbol{U}}^{(l+1)} + \check{\boldsymbol{U}}^{(l+1)} \\
&= \left(\left(\left(\begin{cases} \left(\mathbf{1}_{v=f_{Pa}(v') \wedge d_{\mathcal{T}}(v')=l}\right) & v \in \mathbb{D} \wedge v' \in \mathbb{V}_+ \\ 0 & v \in \mathbb{T} \vee v' = v_0 \end{cases} \right)_{(v,v') \in \mathbb{V}^2} \right) \odot \left(\boldsymbol{s}_{v'}\right)_{(v,v') \in \mathbb{V}^2} \right) \check{\boldsymbol{U}}^{(l+1)} + \check{\boldsymbol{U}}^{(l+1)}
\end{aligned}
$$

Using Equation 10 and Equation 18

$$
= \left(\boldsymbol{L}^{(l)} \odot \boldsymbol{S}\right) \check{\boldsymbol{U}}^{(l+1)} + \check{\boldsymbol{U}}^{(l+1)}
$$

G.4 "EXCEPTED" REACH PROBABILITIES

G.4.1 INITIAL CONDITION

An expanded form of Equation 25 is shown below.

$$
\begin{aligned}
\breve{\mathbf{\Pi}}^{(0)} &= \left(\left(\begin{cases} \breve{\pi}(\sigma^{(T)}, v, i_+) & d_{\mathcal{T}}(v) \leqslant l \\ 0 & d_{\mathcal{T}}(v) > l \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} \right) \Bigg|_{l=0} \\
&= \left(\begin{cases} \breve{\pi}(\sigma^{(T)}, v, i_+) & d_{\mathcal{T}}(v) \leqslant 0 \\ 0 & d_{\mathcal{T}}(v) > 0 \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} \\
&= \left(\begin{cases} \breve{\pi}(\sigma^{(T)}, v, i_+) & v = v_0 \\ 0 & v \in \mathbb{V}_+ \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+}
\end{aligned}
$$

Using Equation 2

$$
\begin{aligned}
&= \left(\begin{cases} \begin{cases} \ldots & v \in \mathbb{V}_+ \\ 1 & v = v_0 \end{cases} & v = v_0 \\ 0 & v \in \mathbb{V}_+ \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} \\
&= \left(\begin{cases} 1 & v = v_0 \\ 0 & v \in \mathbb{V}_+ \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} \\
&= \left(\mathbf{1}_{v=v_0} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+}
\end{aligned}
$$

G.4.2 RECURRENCE

An expanded form of Equation 26 is shown below.

$$\forall l \in [1, D] \cap \mathbb{Z}$$

$$
\begin{aligned}
\breve{\mathbf{\Pi}}^{(l)} &= \left(\begin{cases} \breve{\pi}(\sigma^{(T)}, v, i_+) & d_{\mathcal{T}}(v) \leqslant l \\ 0 & d_{\mathcal{T}}(v) > l \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} \\
&= \left(\begin{cases} \breve{\pi}(\sigma^{(T)}, v, i_+) & d_{\mathcal{T}}(v) = l \\ 0 & d_{\mathcal{T}}(v) \neq l \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} + \left(\begin{cases} \breve{\pi}(\sigma^{(T)}, v, i_+) & d_{\mathcal{T}}(v) \leqslant l-1 \\ 0 & d_{\mathcal{T}}(v) > l-1 \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+}
\end{aligned}
$$

Using Equation 24

$$
= \left(\begin{cases} \breve{\pi}(\sigma^{(T)}, v, i_+) & d_{\mathcal{T}}(v) = l \\ 0 & d_{\mathcal{T}}(v) \neq l \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} + \breve{\mathbf{\Pi}}^{(l-1)}
$$

Using Equation 2

$$
\begin{aligned}
&= \left(\begin{cases} \begin{cases} \breve{\pi}(\sigma^{(T)}, f_{Pa}(v), i_+) \begin{cases} \sigma^{(T)}(f_h(f_{Pa}(v)), f_a(v)) & f_i(f_h(f_{Pa}(v))) \neq i_+ \\ 1 & f_i(f_h(f_{Pa}(v))) = i_+ \end{cases} & v \in \mathbb{V}_+ \\ 1 & v = v_0 \end{cases} & d_{\mathcal{T}}(v) = l \\ 0 & d_{\mathcal{T}}(v) \neq l \end{cases} \right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+} \\
&\quad + \breve{\mathbf{\Pi}}^{(l-1)}
\end{aligned}
$$

Since $l \neq 0$ and therefore $v \neq v_0$

$$
= \left(\begin{cases} \breve{\pi}(\sigma^{(T)}, f_{Pa}(v), i_+) \begin{cases} \sigma^{(T)}(f_h(f_{Pa}(v)), f_a(v)) & f_i(f_h(f_{Pa}(v))) \neq i_+ \\ 1 & f_i(f_h(f_{Pa}(v))) = i_+ \end{cases} & d_{\mathcal{T}}(v) = l \\ 0 & d_{\mathcal{T}}(v) \neq l \end{cases} \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} + \breve{\boldsymbol{\Pi}}^{(l-1)}
$$

$$
= \left(\left(\begin{cases} \breve{\pi}(\sigma^{(T)}, f_{Pa}(v), i_+) & d_{\mathcal{T}}(v) = l \\ 0 & d_{\mathcal{T}}(v) \neq l \end{cases} \right) \left(\begin{cases} \begin{cases} \sigma^{(T)}(f_h(f_{Pa}(v)), f_a(v)) & f_i(f_h(f_{Pa}(v))) \neq i_+ \\ 1 & f_i(f_h(f_{Pa}(v))) = i_+ \end{cases} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases} \right) \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+}
$$

$$
+ \breve{\boldsymbol{\Pi}}^{(l-1)}
$$

$$
= \left(\begin{cases} \breve{\pi}(\sigma^{(T)}, f_{Pa}(v), i_+) & d_{\mathcal{T}}(v) = l \\ 0 & d_{\mathcal{T}}(v) \neq l \end{cases} \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+}
$$

$$
\odot \left(\begin{cases} \begin{cases} \sigma^{(T)}(f_h(f_{Pa}(v)), f_a(v)) & f_i(f_h(f_{Pa}(v))) \neq i_+ \\ 1 & f_i(f_h(f_{Pa}(v))) = i_+ \end{cases} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases} \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} + \breve{\boldsymbol{\Pi}}^{(l-1)}
$$

Using Equation 17 and Equation 13

$$
= \left(\left(\begin{cases} \breve{\pi}(\sigma^{(T)}, f_{Pa}(v), i_+) & d_{\mathcal{T}}(v) = l \\ 0 & d_{\mathcal{T}}(v) \neq l \end{cases} \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} \right) \odot \left(\begin{cases} \begin{cases} \boldsymbol{s}_v & \left(\boldsymbol{M}^{(V,I_+)}\right)_{v,i_+} = 0 \\ 1 & \left(\boldsymbol{M}^{(V,I_+)}\right)_{v,i_+} = 1 \end{cases} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases} \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+}
$$

$$
+ \breve{\boldsymbol{\Pi}}^{(l-1)}
$$

Using Equation 17

$$
= \left(\left(\begin{cases} \breve{\pi}(\sigma^{(T)}, f_{Pa}(v), i_+) & d_{\mathcal{T}}(v) = l \\ 0 & d_{\mathcal{T}}(v) \neq l \end{cases} \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} \right) \odot \left(\begin{cases} \begin{cases} \boldsymbol{s}_v & \left(\boldsymbol{M}^{(V,I_+)}\right)_{v,i_+} = 0 \\ 1 & \left(\boldsymbol{M}^{(V,I_+)}\right)_{v,i_+} = 1 \end{cases} & v \in \mathbb{V}_+ \\ \boldsymbol{s}_v & v = v_0 \end{cases} \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+}
$$

$$
+ \breve{\boldsymbol{\Pi}}^{(l-1)}
$$

Using Equation 13

$$
= \left(\left(\begin{cases} \breve{\pi}(\sigma^{(T)}, f_{Pa}(v), i_+) & d_{\mathcal{T}}(v) = l \\ 0 & d_{\mathcal{T}}(v) \neq l \end{cases} \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} \right) \odot \left(\begin{cases} \boldsymbol{s}_v & \left(\boldsymbol{M}^{(V,I_+)}\right)_{v,i_+} = 0 \\ 1 & \left(\boldsymbol{M}^{(V,I_+)}\right)_{v,i_+} = 1 \end{cases} \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} + \breve{\boldsymbol{\Pi}}^{(l-1)}
$$

Using Equation 23

$$
= \left(\left(\begin{cases} \breve{\pi}(\sigma^{(T)}, f_{Pa}(v), i_+) & d_{\mathcal{T}}(v) = l \\ 0 & d_{\mathcal{T}}(v) \neq l \end{cases} \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} \right) \odot \breve{\boldsymbol{S}} + \breve{\boldsymbol{\Pi}}^{(l-1)}
$$

$$
= \left(\left(\begin{cases} \left(\mathbf{1}_{d_{\mathcal{T}}(v)=l}\right) \breve{\pi}(\sigma^{(T)}, f_{Pa}(v), i_+) & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases} \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} \right) \odot \breve{\boldsymbol{S}} + \breve{\boldsymbol{\Pi}}^{(l-1)}
$$

$$
= \left(\left(\sum_{v' \in \mathbb{V}} \begin{cases} \left(\mathbf{1}_{v'=f_{Pa}(v) \wedge d_{\mathcal{T}}(v)=l}\right) \breve{\pi}(\sigma^{(T)}, v', i_+) & v' \in \mathbb{D} \wedge v \in \mathbb{V}_+ \\ 0 & v' \in \mathbb{T} \vee v = v_0 \end{cases} \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} \right) \odot \breve{\boldsymbol{S}} + \breve{\boldsymbol{\Pi}}^{(l-1)}
$$

$$
= \left(\left(\sum_{v' \in \mathbb{V}} \left(\begin{cases} \mathbf{1}_{v'=f_{Pa}(v) \wedge d_{\mathcal{T}}(v)=l} & v' \in \mathbb{D} \wedge v \in \mathbb{V}_+ \\ 0 & v' \in \mathbb{T} \vee v = v_0 \end{cases} \right) \left(\begin{cases} \breve{\pi}(\sigma^{(T)}, v', i_+) & d_{\mathcal{T}}(v') \leqslant l-1 \\ 0 & d_{\mathcal{T}}(v') > l-1 \end{cases} \right) \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} \right) \odot \breve{\boldsymbol{S}}
$$

$$
+ \breve{\boldsymbol{\Pi}}^{(l-1)}
$$

$$
= \left(\left(\left(\begin{cases} \mathbf{1}_{v'=f_{Pa}(v) \wedge d_{\mathcal{T}}(v)=l} & v' \in \mathbb{D} \wedge v \in \mathbb{V}_+ \\ 0 & v' \in \mathbb{T} \vee v = v_0 \end{cases} \right)_{(v,v') \in \mathbb{V}^2} \right) \left(\begin{cases} \breve{\pi}(\sigma^{(T)}, v, i_+) & d_{\mathcal{T}}(v) \leqslant l-1 \\ 0 & d_{\mathcal{T}}(v) > l-1 \end{cases} \right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} \right) \odot \breve{\boldsymbol{S}}
$$

$$
+ \breve{\boldsymbol{\Pi}}^{(l-1)}
$$

Using Equation 24

$$= \left(\left(\left(\left(\begin{cases} \mathbf{1}_{v'=f_{Pa}(v) \wedge d_{\mathcal{T}}(v)=l} & v' \in \mathbb{D} \wedge v \in \mathbb{V}_+ \\ 0 & v' \in \mathbb{T} \vee v = v_0 \end{cases}\right)_{(v,v') \in \mathbb{V}^2}\right) \breve{\boldsymbol{\Pi}}^{(l-1)}\right) \odot \check{\boldsymbol{S}} + \breve{\boldsymbol{\Pi}}^{(l-1)}$$

$$= \left(\left(\left(\left(\begin{cases} \mathbf{1}_{v=f_{Pa}(v') \wedge d_{\mathcal{T}}(v')=l} & v \in \mathbb{D} \wedge v' \in \mathbb{V}_+ \\ 0 & v \in \mathbb{T} \vee v' = v_0 \end{cases}\right)_{(v,v') \in \mathbb{V}^2}\right)^{\top} \breve{\boldsymbol{\Pi}}^{(l-1)}\right) \odot \check{\boldsymbol{S}} + \breve{\boldsymbol{\Pi}}^{(l-1)}$$

Using Equation 10

$$= \left(\left(\boldsymbol{L}^{(l)}\right)^{\top} \breve{\boldsymbol{\Pi}}^{(l-1)}\right) \odot \check{\boldsymbol{S}} + \breve{\boldsymbol{\Pi}}^{(l-1)}$$

G.5 "Counterfactual" Reach Probability Terms

An expanded form of Equation 27 is shown below.

$$\breve{\boldsymbol{\pi}} = \left(\begin{cases} \begin{cases} \breve{\pi}(\sigma^{(T)}, v, f_i(f_h(f_{Pa}(v)))) & f_h(f_{Pa}(v)) \in \mathbb{H}_+ \\ 0 & f_h(f_{Pa}(v)) \in \mathbb{H}_0 \end{cases} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\right)_{v \in \mathbb{V}}$$

$$= \left(\begin{cases} \begin{cases} \sum_{i_+ \in \mathbb{I}_+} \left(\mathbf{1}_{f_i(f_h(f_{Pa}(v)))=i_+}\right) \breve{\pi}(\sigma^{(T)}, v, i_+) & f_h(f_{Pa}(v)) \in \mathbb{H}_+ \\ 0 & f_h(f_{Pa}(v)) \in \mathbb{H}_0 \end{cases} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\right)_{v \in \mathbb{V}}$$

$$= \left(\begin{cases} \begin{cases} \left(\mathbf{1}_{f_i(f_h(f_{Pa}(v)))=i_+}\right) \breve{\pi}(\sigma^{(T)}, v, i_+) & f_h(f_{Pa}(v)) \in \mathbb{H}_+ \\ 0 & f_h(f_{Pa}(v)) \in \mathbb{H}_0 \end{cases} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} \mathbf{1}_{|\mathbb{I}_+|}$$

$$= \left(\begin{cases} \left(\mathbf{1}_{f_i(f_h(f_{Pa}(v)))=i_+}\right) \breve{\pi}(\sigma^{(T)}, v, i_+) & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} \mathbf{1}_{|\mathbb{I}_+|}$$

$$= \left(\left(\begin{cases} \mathbf{1}_{f_i(f_h(f_{Pa}(v)))=i_+} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\right) \breve{\pi}(\sigma^{(T)}, v, i_+)\right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+} \mathbf{1}_{|\mathbb{I}_+|}$$

$$= \left(\left(\left(\begin{cases} \mathbf{1}_{f_i(f_h(f_{Pa}(v)))=i_+} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+}\right) \odot \left(\breve{\pi}(\sigma^{(T)}, v, i_+)\right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+}\right) \mathbf{1}_{|\mathbb{I}_+|}$$

Using Equation 13

$$= \left(\left(\boldsymbol{M}^{(V,I_+)}\right) \odot \left(\breve{\pi}(\sigma^{(T)}, v, i_+)\right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+}\right) \mathbf{1}_{|\mathbb{I}_+|}$$

$$= \left(\left(\boldsymbol{M}^{(V,I_+)}\right) \odot \left(\begin{cases} \breve{\pi}(\sigma^{(T)}, v, i_+) & d_{\mathcal{T}}(v) \leqslant D \\ 0 & d_{\mathcal{T}}(v) > D \end{cases}\right)_{(v,i_+) \in \mathbb{V} \times \mathbb{I}_+}\right) \mathbf{1}_{|\mathbb{I}_+|}$$

Using Equation 24

$$= \left(\left(\boldsymbol{M}^{(V,I_+)}\right) \odot \breve{\boldsymbol{\Pi}}^{(D)}\right) \mathbf{1}_{|\mathbb{I}_+|}$$

G.6 "Counterfactual" Reach Probabilities

An expanded form of Equation 28 is shown below.

$$\tilde{\boldsymbol{\pi}} = \left(\tilde{\pi}(\sigma^{(T)}, h_+)\right)_{h_+ \in \mathbb{H}_+}$$

Using Equation 3

$$= \left(\sum_{d \in \mathbb{D}: f_h(d) = h_+} \breve{\pi}(\sigma^{(T)}, d, f_i(h_+))\right)_{h_+ \in \mathbb{H}_+}$$

$$= \left(\frac{|A(h_+)|}{|A(h_+)|} \sum_{d \in \mathbb{D}: f_h(d) = h_+} \breve{\pi}(\sigma^{(T)}, d, f_i(h_+))\right)_{h_+ \in \mathbb{H}_+}$$

$$= \left(\left(\sum_{d \in \mathbb{D}: f_h(d) = h_+} |A(h_+)| \breve{\pi}(\sigma^{(T)}, d, f_i(h_+))\right)_{h_+ \in \mathbb{H}_+}\right) \oslash \left(|A(h_+)|\right)_{h_+ \in \mathbb{H}_+}$$

$$= \left(\sum_{v \in \mathbb{V}} \begin{cases} \left(\mathbf{1}_{h_+ = f_h(f_{Pa}(v))}\right) \breve{\pi}(\sigma^{(T)}, f_{Pa}(v), f_i(f_h(f_{Pa}(v)))) & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\right)_{h_+ \in \mathbb{H}_+}$$

$$\oslash \left(\sum_{(h'_+, a) \in \mathbb{Q}_+} \mathbf{1}_{h_+ = h'_+}\right)_{h_+ \in \mathbb{H}_+}$$

Using Equation 2

$$= \left(\sum_{v \in \mathbb{V}} \begin{cases} \left(\mathbf{1}_{h_+ = f_h(f_{Pa}(v))}\right) \breve{\pi}(\sigma^{(T)}, v, f_i(f_h(f_{Pa}(v)))) & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\right)_{h_+ \in \mathbb{H}_+}$$

$$\oslash \left(\sum_{(h'_+, a) \in \mathbb{Q}_+} \mathbf{1}_{h_+ = h'_+}\right)_{h_+ \in \mathbb{H}_+}$$

$$= \Bigg(\left(\begin{cases} \mathbf{1}_{h_+ = f_h(f_{Pa}(v))} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\right)_{(h_+, v) \in \mathbb{H}_+ \times \mathbb{V}}$$

$$\left(\begin{cases} \begin{cases} \breve{\pi}(\sigma^{(T)}, v, f_i(f_h(f_{Pa}(v)))) & f_h(f_{Pa}(v)) \in \mathbb{H}_+ \\ 0 & f_h(f_{Pa}(v)) \in \mathbb{H}_0 \end{cases} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\right)_{v \in \mathbb{V}}\Bigg)$$

$$\oslash \left(\left(\left(\mathbf{1}_{h_+ = h'_+}\right)_{(h_+, (h'_+, a)) \in \mathbb{H}_+ \times \mathbb{Q}_+}\right) \mathbf{1}_{|\mathbb{Q}_+|}\right)$$

Using Equation 12

$$= \Bigg(\left(\begin{cases} \mathbf{1}_{h_+ = f_h(f_{Pa}(v))} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\right)_{(h_+, v) \in \mathbb{H}_+ \times \mathbb{V}}$$

$$\left(\left(\begin{cases}\begin{cases}\breve{\pi}(\sigma^{(T)}, v, f_i(f_h(f_{Pa}(v)))) & f_h(f_{Pa}(v)) \in \mathbb{H}_+ \\ 0 & f_h(f_{Pa}(v)) \in \mathbb{H}_0\end{cases} & v \in \mathbb{V}_+ \\ 0 & v = v_0\end{cases}\right)_{v\in\mathbb{V}}\right) \oslash \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right)\mathbf{1}_{|\mathbb{Q}_+|}\right)$$

Using Equation 27

$$= \left(\left(\left(\begin{cases}\mathbf{1}_{h_+=f_h(f_{Pa}(v))} & v \in \mathbb{V}_+ \\ 0 & v = v_0\end{cases}\right)_{(h_+,v)\in\mathbb{H}_+\times\mathbb{V}}\right)\breve{\boldsymbol{\pi}}\right) \oslash \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right)\mathbf{1}_{|\mathbb{Q}_+|}\right)$$

$$= \left(\left(\left(\begin{cases}\mathbf{1}_{(h_+,f_a(v))=(f_h(f_{Pa}(v)),f_a(v))} & v \in \mathbb{V}_+ \\ 0 & v = v_0\end{cases}\right)_{(h_+,v)\in\mathbb{H}_+\times\mathbb{V}}\right)\breve{\boldsymbol{\pi}}\right) \oslash \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right)\mathbf{1}_{|\mathbb{Q}_+|}\right)$$

$$= \left(\left(\left(\sum_{(h'_+,a)\in\mathbb{Q}_+}\mathbf{1}_{h_+=h'_+}\begin{cases}\mathbf{1}_{(h'_+,a)=(f_h(f_{Pa}(v)),f_a(v))} & v \in \mathbb{V}_+ \\ 0 & v = v_0\end{cases}\right)_{(h_+,v)\in\mathbb{H}_+\times\mathbb{V}}\right)\breve{\boldsymbol{\pi}}\right)$$
$$\oslash \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right)\mathbf{1}_{|\mathbb{Q}_+|}\right)$$

$$= \left(\left(\left(\mathbf{1}_{h_+=h'_+}\right)_{(h_+,(h'_+,a))\in\mathbb{H}_+\times\mathbb{Q}_+}\right)\left(\left(\begin{cases}\mathbf{1}_{q_+=(f_h(f_{Pa}(v)),f_a(v))} & v \in \mathbb{V}_+ \\ 0 & v = v_0\end{cases}\right)_{(q_+,v)\in\mathbb{Q}_+\times\mathbb{V}}\right)\breve{\boldsymbol{\pi}}\right)$$
$$\oslash \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right)\mathbf{1}_{|\mathbb{Q}_+|}\right)$$

Using Equation 12 and Equation 11

$$= \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right)\left(\boldsymbol{M}^{(Q_+,V)}\right)\breve{\boldsymbol{\pi}}\right) \oslash \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right)\mathbf{1}_{|\mathbb{Q}_+|}\right)$$

G.7 Counterfactual Reach Probability Sums

An expanded form of Equation 29 is shown below.

$$\begin{aligned}\tilde{\boldsymbol{\pi}}^{(\Sigma)} &= \left(\sum_{\tau=1}^{T}\tilde{\pi}(\sigma^{(\tau)}, h_+)\right)_{h_+\in\mathbb{H}_+} \\ &= \left(\left(\sum_{\tau=1}^{T-1}\tilde{\pi}(\sigma^{(\tau)}, h_+)\right) + \tilde{\pi}(\sigma^{(T)}, h_+)\right)_{h_+\in\mathbb{H}_+} \\ &= \left(\sum_{\tau=1}^{T-1}\tilde{\pi}(\sigma^{(\tau)}, h_+)\right)_{h_+\in\mathbb{H}_+} + \left(\tilde{\pi}(\sigma^{(T)}, h_+)\right)_{h_+\in\mathbb{H}_+}\end{aligned}$$

Using Equation 29 and Equation 28

$$= \tilde{\boldsymbol{\pi}}^{(\Sigma)\prime} + \tilde{\boldsymbol{\pi}}$$

G.8 Average Strategy Profile

An expanded form of Equation 30 is shown below.

$$\bar{\boldsymbol{\sigma}} = \left(\bar{\sigma}^{(T)}(q_+)\right)_{q_+ \in \mathbb{Q}_+}$$

Using Equation 8

$$\begin{aligned}
&= \left(\frac{\sum_{\tau=1}^{T} \tilde{\pi}(\sigma^{(\tau)}, h_+)\sigma^{(\tau)}(h_+, a)}{\sum_{\tau=1}^{T} \tilde{\pi}(\sigma^{(\tau)}, h_+)}\right)_{(h_+,a)\in\mathbb{Q}_+} \\
&= \left(\frac{\sum_{\tau=1}^{T-1} \tilde{\pi}(\sigma^{(\tau)}, h_+)\sigma^{(\tau)}(h_+, a)}{\sum_{\tau=1}^{T} \tilde{\pi}(\sigma^{(\tau)}, h_+)} + \frac{\tilde{\pi}(\sigma^{(T)}, h_+)\sigma^{(T)}(h_+, a)}{\sum_{\tau=1}^{T} \tilde{\pi}(\sigma^{(\tau)}, h_+)}\right)_{(h_+,a)\in\mathbb{Q}_+} \\
&= \left(\left(\frac{\sum_{\tau=1}^{T-1} \tilde{\pi}(\sigma^{(\tau)}, h_+)}{\sum_{\tau=1}^{T} \tilde{\pi}(\sigma^{(\tau)}, h_+)}\right)\left(\frac{\sum_{\tau=1}^{T-1} \tilde{\pi}(\sigma^{(\tau)}, h_+)\sigma^{(\tau)}(h_+, a)}{\sum_{\tau=1}^{T-1} \tilde{\pi}(\sigma^{(\tau)}, h_+)}\right) + \right. \\
&\qquad \left. + \left(\frac{\tilde{\pi}(\sigma^{(T)}, h_+)}{\sum_{\tau=1}^{T} \tilde{\pi}(\sigma^{(\tau)}, h_+)}\right)\sigma^{(T)}(h_+, a)\right)_{(h_+,a)\in\mathbb{Q}_+}
\end{aligned}$$

Using Equation 8

$$\begin{aligned}
&= \left(\left(\frac{\sum_{\tau=1}^{T-1} \tilde{\pi}(\sigma^{(\tau)}, h_+)}{\sum_{\tau=1}^{T} \tilde{\pi}(\sigma^{(\tau)}, h_+)}\right)\bar{\sigma}^{(T-1)}(h_+, a) + \left(\frac{\tilde{\pi}(\sigma^{(T)}, h_+)}{\sum_{\tau=1}^{T} \tilde{\pi}(\sigma^{(\tau)}, h_+)}\right)\sigma^{(T)}(h_+, a)\right)_{(h_+,a)\in\mathbb{Q}_+} \\
&= \left(\left(1 - \frac{\tilde{\pi}(\sigma^{(T)}, h_+)}{\sum_{\tau=1}^{T} \tilde{\pi}(\sigma^{(\tau)}, h_+)}\right)\bar{\sigma}^{(T-1)}(h_+, a) + \right. \\
&\qquad \left. + \left(\frac{\tilde{\pi}(\sigma^{(T)}, h_+)}{\sum_{\tau=1}^{T} \tilde{\pi}(\sigma^{(\tau)}, h_+)}\right)\sigma^{(T)}(h_+, a)\right)_{(h_+,a)\in\mathbb{Q}_+} \\
&= \left(\bar{\sigma}^{(T-1)}(h_+, a) + \left(\frac{\tilde{\pi}(\sigma^{(T)}, h_+)}{\sum_{\tau=1}^{T} \tilde{\pi}(\sigma^{(\tau)}, h_+)}\right)\left(\sigma^{(T)}(h_+, a) - \bar{\sigma}^{(T-1)}(h_+, a)\right)\right)_{(h_+,a)\in\mathbb{Q}_+} \\
&= \left(\bar{\sigma}^{(T-1)}(q_+)\right)_{q_+\in\mathbb{Q}_+} \\
&\qquad + \left(\frac{\tilde{\pi}(\sigma^{(T)}, h_+)}{\sum_{\tau=1}^{T} \tilde{\pi}(\sigma^{(\tau)}, h_+)}\right)_{(h_+,a)\in\mathbb{Q}_+} \\
&\qquad\quad \odot \left(\left(\left(\sigma^{(T)}(q_+)\right)_{q_+\in\mathbb{Q}_+}\right) - \left(\bar{\sigma}^{(T-1)}(q_+)\right)_{q_+\in\mathbb{Q}_+}\right)
\end{aligned}$$

Using Equation 30 and Equation 15

$$= \bar{\boldsymbol{\sigma}}' + \left(\left(\frac{\tilde{\pi}(\sigma^{(T)}, h_+)}{\sum_{\tau=1}^{T} \tilde{\pi}(\sigma^{(\tau)}, h_+)}\right)_{(h_+,a)\in\mathbb{Q}_+}\right) \odot \left(\boldsymbol{\sigma} - \bar{\boldsymbol{\sigma}}'\right)$$

$$
\begin{aligned}
&= \bar{\boldsymbol{\sigma}}' + \left(\left(\sum_{h'_+ \in \mathbb{H}_+} \left(\mathbf{1}_{h_+ = h'_+} \right) \frac{\tilde{\pi}(\sigma^{(T)}, h'_+)}{\sum_{\tau=1}^{T} \tilde{\pi}(\sigma^{(\tau)}, h'_+)} \right)_{(h_+, a) \in \mathbb{Q}_+} \right) \odot (\boldsymbol{\sigma} - \bar{\boldsymbol{\sigma}}') \\
&= \bar{\boldsymbol{\sigma}}' + \left(\left(\left(\mathbf{1}_{h_+ = h'_+} \right)_{((h_+, a), h'_+) \in \mathbb{Q}_+ \times \mathbb{H}_+} \right) \left(\frac{\tilde{\pi}(\sigma^{(T)}, h_+)}{\sum_{\tau=1}^{T} \tilde{\pi}(\sigma^{(\tau)}, h_+)} \right)_{h_+ \in \mathbb{H}_+} \right) \odot (\boldsymbol{\sigma} - \bar{\boldsymbol{\sigma}}') \\
&= \bar{\boldsymbol{\sigma}}' \\
&\quad + \left(\left(\left(\mathbf{1}_{h_+ = h'_+} \right)_{(h_+, (h'_+, a)) \in \mathbb{H}_+ \times \mathbb{Q}_+} \right)^{\top} \right. \\
&\qquad \left. \left(\left(\left(\tilde{\pi}(\sigma^{(T)}, h_+) \right)_{h_+ \in \mathbb{H}_+} \right) \oslash \left(\sum_{\tau=1}^{T} \tilde{\pi}(\sigma^{(\tau)}, h_+) \right)_{h_+ \in \mathbb{H}_+} \right) \right) \odot (\boldsymbol{\sigma} - \bar{\boldsymbol{\sigma}}')
\end{aligned}
$$

Using Equation 12, Equation 28, and Equation 29

$$
= \bar{\boldsymbol{\sigma}}' + \left(\left(\boldsymbol{M}^{(H_+, Q_+)} \right)^{\top} \left(\tilde{\boldsymbol{\pi}} \oslash \tilde{\boldsymbol{\pi}}^{(\Sigma)} \right) \right) \odot (\boldsymbol{\sigma} - \bar{\boldsymbol{\sigma}}')
$$

G.9 Instantaneous Counterfactual Regrets

An expanded form of Equation 31 is shown below.

First, define a dense vector $\boldsymbol{\rho} \in \mathbb{R}^{\mathbb{V}}$ for intermediate values.

$$
\begin{aligned}
\boldsymbol{\rho} &= \left(\begin{cases} \begin{cases} \check{u}(\sigma^{(T)}, v, f_i(f_h(f_{Pa}(v)))) - \check{u}(\sigma^{(T)}, f_{Pa}(v), f_i(f_h(f_{Pa}(v)))) & f_h(f_{Pa}(v)) \in \mathbb{H}_+ \\ 0 & f_h(f_{Pa}(v)) \in \mathbb{H}_0 \end{cases} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases} \right)_{v \in \mathbb{V}} \\
&= \left(\left(\boldsymbol{M}^{(V, I_+)} \right) \odot \left(\breve{\boldsymbol{U}} - \boldsymbol{G}^{\top} \breve{\boldsymbol{U}} \right) \right) \mathbf{1}_{|\mathbb{I}_+|}
\end{aligned}
\tag{35}
$$

Then,

$$
\tilde{\boldsymbol{r}} = \left(\tilde{r}(\sigma^{(T)}, q_+) \right)_{q_+ \in \mathbb{Q}_+}
$$

Using Equation 5

$$
= \left(\tilde{\pi}(\sigma^{(T)}, h_+)(\tilde{u}(\sigma^{(T)}|_{h_+ \to a}, h_+) - \tilde{u}(\sigma^{(T)}, h_+)) \right)_{(h_+, a) \in \mathbb{Q}_+}
$$

Using Equation 4

$$
\begin{aligned}
= &\left(\tilde{\pi}(\sigma^{(T)}, h_+) \right. \\
&\quad \left(\frac{\sum_{d \in \mathbb{D}: f_h(d) = h_+} \check{\pi}(\sigma^{(T)}|_{h_+ \to a}, d, f_i(h_+)) \check{u}(\sigma^{(T)}|_{h_+ \to a}, d, f_i(h_+))}{\tilde{\pi}(\sigma^{(T)}|_{h_+ \to a}, h_+)} \right. \\
&\qquad \left. \left. - \frac{\sum_{d \in \mathbb{D}: f_h(d) = h_+} \check{\pi}(\sigma^{(T)}, d, f_i(h_+)) \check{u}(\sigma^{(T)}, d, f_i(h_+))}{\tilde{\pi}(\sigma^{(T)}, h_+)} \right) \right)_{(h_+, a) \in \mathbb{Q}_+}
\end{aligned}
$$

$$
= \left(\tilde{\pi}(\sigma^{(T)}, h_+) \right.
$$

$$\left(\frac{\sum_{d\in\mathbb{D}:f_h(d)=h_+}\breve{\pi}(\sigma^{(T)},d,f_i(h_+))\breve{u}(\sigma^{(T)}|_{h_+\to a},d,f_i(h_+))}{\breve{\pi}(\sigma^{(T)},h_+)}\right.$$
$$\left.\left.-\frac{\sum_{d\in\mathbb{D}:f_h(d)=h_+}\breve{\pi}(\sigma^{(T)},d,f_i(h_+))\breve{u}(\sigma^{(T)},d,f_i(h_+))}{\breve{\pi}(\sigma^{(T)},h_+)}\right)\right)_{(h_+,a)\in\mathbb{Q}_+}$$

$$=\left(\sum_{d\in\mathbb{D}:f_h(d)=h_+}\breve{\pi}(\sigma^{(T)},d,f_i(h_+))\breve{u}(\sigma^{(T)}|_{h_+\to a},d,f_i(h_+))\right.$$
$$\left.-\sum_{d\in\mathbb{D}:f_h(d)=h_+}\breve{\pi}(\sigma^{(T)},d,f_i(h_+))\breve{u}(\sigma^{(T)},d,f_i(h_+))\right)_{(h_+,a)\in\mathbb{Q}_+}$$

$$=\left(\sum_{d\in\mathbb{D}:f_h(d)=h_+}\breve{\pi}(\sigma^{(T)},d,f_i(h_+))\left(\breve{u}(\sigma^{(T)}|_{h_+\to a},d,f_i(h_+))-\breve{u}(\sigma^{(T)},d,f_i(h_+))\right)\right)_{(h_+,a)\in\mathbb{Q}_+}$$

$$=\left(\sum_{v\in\mathbb{V}}\left(\begin{cases}\mathbf{1}_{q_+=(f_h(f_{Pa}(v)),f_a(v))} & v\in\mathbb{V}_+\\ 0 & v=v_0\end{cases}\right)\right.$$
$$\left(\begin{cases}\begin{cases}\breve{\pi}(\sigma^{(T)},f_{Pa}(v),f_i(f_h(f_{Pa}(v)))) & f_h(f_{Pa}(v))\in\mathbb{H}_+\\ 0 & f_h(f_{Pa}(v))\in\mathbb{H}_0\end{cases} & v\in\mathbb{V}_+\\ 0 & v=v_0\end{cases}\right)$$
$$\left.\left(\begin{cases}\begin{cases}\left(\begin{matrix}\breve{u}(\sigma^{(T)}|_{f_h(f_{Pa}(v))\to a},f_{Pa}(v),f_i(f_h(f_{Pa}(v))))\\ -\breve{u}(\sigma^{(T)},f_{Pa}(v),f_i(f_h(f_{Pa}(v))))\end{matrix}\right) & f_h(f_{Pa}(v))\in\mathbb{H}_+\\ 0 & f_h(f_{Pa}(v))\in\mathbb{H}_0\end{cases} & v\in\mathbb{V}_+\\ 0 & v=v_0\end{cases}\right)\right)_{q_+\in\mathbb{Q}_+}$$

$$=\left(\sum_{v\in\mathbb{V}}\left(\begin{cases}\mathbf{1}_{q_+=(f_h(f_{Pa}(v)),f_a(v))} & v\in\mathbb{V}_+\\ 0 & v=v_0\end{cases}\right)\right.$$
$$\left(\begin{cases}\begin{cases}\breve{\pi}(\sigma^{(T)},v,f_i(f_h(f_{Pa}(v)))) & f_h(f_{Pa}(v))\in\mathbb{H}_+\\ 0 & f_h(f_{Pa}(v))\in\mathbb{H}_0\end{cases} & v\in\mathbb{V}_+\\ 0 & v=v_0\end{cases}\right)$$
$$\left.\left(\begin{cases}\begin{cases}\breve{u}(\sigma^{(T)},v,f_i(f_h(f_{Pa}(v))))-\breve{u}(\sigma^{(T)},f_{Pa}(v),f_i(f_h(f_{Pa}(v)))) & f_h(f_{Pa}(v))\in\mathbb{H}_+\\ 0 & f_h(f_{Pa}(v))\in\mathbb{H}_0\end{cases} & v\in\mathbb{V}_+\\ 0 & v=v_0\end{cases}\right)\right)_{q_+\in\mathbb{Q}_+}$$

$$=\left(\begin{cases}\mathbf{1}_{q_+=(f_h(f_{Pa}(v)),f_a(v))} & v\in\mathbb{V}_+\\ 0 & v=v_0\end{cases}\right)_{(q_+,v)\in\mathbb{Q}_+\times\mathbb{V}}$$
$$\left(\left(\begin{cases}\begin{cases}\breve{\pi}(\sigma^{(T)},v,f_i(f_h(f_{Pa}(v)))) & f_h(f_{Pa}(v))\in\mathbb{H}_+\\ 0 & f_h(f_{Pa}(v))\in\mathbb{H}_0\end{cases} & v\in\mathbb{V}_+\\ 0 & v=v_0\end{cases}\right)_{v\in\mathbb{V}}\right.$$
$$\left.\odot\left(\begin{cases}\begin{cases}\breve{u}(\sigma^{(T)},v,f_i(f_h(f_{Pa}(v))))-\breve{u}(\sigma^{(T)},f_{Pa}(v),f_i(f_h(f_{Pa}(v)))) & f_h(f_{Pa}(v))\in\mathbb{H}_+\\ 0 & f_h(f_{Pa}(v))\in\mathbb{H}_0\end{cases} & v\in\mathbb{V}_+\\ 0 & v=v_0\end{cases}\right)_{v\in\mathbb{V}}\right)$$

Using Equation 11, Equation 27, and Equation 35

$$=\left(\boldsymbol{M}^{(Q_+,V)}\right)(\breve{\boldsymbol{\pi}}\odot\boldsymbol{\rho})$$

Using Equation 35

$$=\left(\boldsymbol{M}^{(Q_+,V)}\right)\left(\breve{\boldsymbol{\pi}}\odot\left(\left(\left(\boldsymbol{M}^{(V,I_+)}\right)\odot\left(\breve{\boldsymbol{U}}-\boldsymbol{G}^\top\breve{\boldsymbol{U}}\right)\right)\mathbf{1}_{|\mathbb{I}_+|}\right)\right)$$

G.9.1 Intermediate Values

An expanded form of Equation 35 is shown below.

$$\boldsymbol{\rho}=\left(\begin{cases}\begin{cases}\breve{u}(\sigma^{(T)},v,f_i(f_h(f_{Pa}(v))))-\breve{u}(\sigma^{(T)},f_{Pa}(v),f_i(f_h(f_{Pa}(v)))) & f_h(f_{Pa}(v))\in\mathbb{H}_+\\ 0 & f_h(f_{Pa}(v))\in\mathbb{H}_0\end{cases} & v\in\mathbb{V}_+\\ 0 & v=v_0\end{cases}\right)_{v\in\mathbb{V}}$$
$$=\left(\left(\sum_{i_+\in\mathbb{I}_+}\begin{cases}\left(\mathbf{1}_{i_+=f_i(f_h(f_{Pa}(v)))}\right)\begin{cases}\breve{u}(\sigma^{(T)},v,i_+)-\breve{u}(\sigma^{(T)},f_{Pa}(v),i_+) & f_h(f_{Pa}(v))\in\mathbb{H}_+\\ 0 & f_h(f_{Pa}(v))\in\mathbb{H}_0\end{cases} & v\in\mathbb{V}_+\\ 0 & v=v_0\end{cases}\right)_{v\in\mathbb{V}}\right)$$

$$= \Bigg(\bigg(\bigg(\begin{cases} \left(\mathbf{1}_{i_+ = f_i(f_h(f_{Pa}(v)))}\right) \begin{cases} \breve{u}(\sigma^{(T)}, v, i_+) - \breve{u}(\sigma^{(T)}, f_{Pa}(v), i_+) & f_h(f_{Pa}(v)) \in \mathbb{H}_+ \\ 0 & f_h(f_{Pa}(v)) \in \mathbb{H}_0 \end{cases} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\bigg)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+}\Bigg)\mathbf{1}_{|\mathbb{I}_+|}$$

$$= \Bigg(\bigg(\begin{cases} \left(\mathbf{1}_{i_+ = f_i(f_h(f_{Pa}(v)))}\right)\left(\breve{u}(\sigma^{(T)}, v, i_+) - \breve{u}(\sigma^{(T)}, f_{Pa}(v), i_+)\right) & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\bigg)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+}\Bigg)\mathbf{1}_{|\mathbb{I}_+|}$$

$$= \Bigg(\bigg(\bigg(\begin{cases} \mathbf{1}_{i_+ = f_i(f_h(f_{Pa}(v)))} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\bigg)\bigg(\breve{u}(\sigma^{(T)}, v, i_+) - \begin{cases} \breve{u}(\sigma^{(T)}, f_{Pa}(v), i_+) & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\bigg)\bigg)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+}\Bigg)\mathbf{1}_{|\mathbb{I}_+|}$$

$$= \Bigg(\bigg(\begin{cases} \mathbf{1}_{f_i(f_h(f_{Pa}(v)))=i_+} & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\bigg)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+}$$

$$\odot\bigg(\breve{u}(\sigma^{(T)}, v, i_+) - \begin{cases} \breve{u}(\sigma^{(T)}, f_{Pa}(v), i_+) & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\bigg)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+}\Bigg)\mathbf{1}_{|\mathbb{I}_+|}$$

Using Equation 13

$$= \Bigg(\left(\boldsymbol{M}^{(V,I_+)}\right)\odot\bigg(\breve{u}(\sigma^{(T)}, v, i_+) - \begin{cases} \breve{u}(\sigma^{(T)}, f_{Pa}(v), i_+) & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\bigg)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+}\Bigg)\mathbf{1}_{|\mathbb{I}_+|}$$

$$= \Bigg(\left(\boldsymbol{M}^{(V,I_+)}\right)\odot\Bigg(\bigg(\left(\breve{u}(\sigma^{(T)}, v, i_+)\right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+}\bigg) - \bigg(\begin{cases} \breve{u}(\sigma^{(T)}, f_{Pa}(v), i_+) & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\bigg)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+}\Bigg)\Bigg)\mathbf{1}_{|\mathbb{I}_+|}$$

Using Equation 22

$$= \Bigg(\left(\boldsymbol{M}^{(V,I_+)}\right)\odot\Bigg(\breve{\boldsymbol{U}} - \bigg(\begin{cases} \breve{u}(\sigma^{(T)}, f_{Pa}(v), i_+) & v \in \mathbb{V}_+ \\ 0 & v = v_0 \end{cases}\bigg)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+}\Bigg)\Bigg)\mathbf{1}_{|\mathbb{I}_+|}$$

$$= \Bigg(\left(\boldsymbol{M}^{(V,I_+)}\right)\odot\Bigg(\breve{\boldsymbol{U}} - \bigg(\sum_{v'\in\mathbb{V}}\bigg(\begin{cases} \mathbf{1}_{v'=f_{Pa}(v)} & v' \in \mathbb{D} \wedge v \in \mathbb{V}_+ \\ 0 & v' \in \mathbb{T} \vee v = v_0 \end{cases}\bigg)\breve{u}(\sigma^{(T)}, v', i_+)\bigg)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+}\Bigg)\Bigg)\mathbf{1}_{|\mathbb{I}_+|}$$

$$= \Bigg(\left(\boldsymbol{M}^{(V,I_+)}\right)\odot\Bigg(\breve{\boldsymbol{U}} - \bigg(\bigg(\begin{cases} \mathbf{1}_{v'=f_{Pa}(v)} & v' \in \mathbb{D} \wedge v \in \mathbb{V}_+ \\ 0 & v' \in \mathbb{T} \vee v = v_0 \end{cases}\bigg)_{(v,v')\in\mathbb{V}^2}\bigg)\left(\breve{u}(\sigma^{(T)}, v, i_+)\right)_{(v,i_+)\in\mathbb{V}\times\mathbb{I}_+}\Bigg)\Bigg)\mathbf{1}_{|\mathbb{I}_+|}$$

Using Equation 22

$$= \Bigg(\left(\boldsymbol{M}^{(V,I_+)}\right)\odot\Bigg(\breve{\boldsymbol{U}} - \bigg(\bigg(\begin{cases} \mathbf{1}_{v'=f_{Pa}(v)} & v' \in \mathbb{D} \wedge v \in \mathbb{V}_+ \\ 0 & v' \in \mathbb{T} \vee v = v_0 \end{cases}\bigg)_{(v,v')\in\mathbb{V}^2}\bigg)\breve{\boldsymbol{U}}\Bigg)\Bigg)\mathbf{1}_{|\mathbb{I}_+|}$$

$$= \Bigg(\left(\boldsymbol{M}^{(V,I_+)}\right)\odot\Bigg(\breve{\boldsymbol{U}} - \bigg(\bigg(\begin{cases} \mathbf{1}_{v=f_{Pa}(v')} & v \in \mathbb{D} \wedge v' \in \mathbb{V}_+ \\ 0 & v \in \mathbb{T} \vee v' = v_0 \end{cases}\bigg)_{(v,v')\in\mathbb{V}^2}\bigg)^{\top}\breve{\boldsymbol{U}}\Bigg)\Bigg)\mathbf{1}_{|\mathbb{I}_+|}$$

Using Equation 9

$$= \left(\left(\boldsymbol{M}^{(V,I_+)}\right)\odot\left(\breve{\boldsymbol{U}} - \boldsymbol{G}^{\top}\breve{\boldsymbol{U}}\right)\right)\mathbf{1}_{|\mathbb{I}_+|}$$

G.10 Average Counterfactual Regrets

An expanded form of Equation 32 is shown below.

$$\bar{\boldsymbol{r}} = \left(\bar{r}^{(T)}(q_+)\right)_{q_+\in\mathbb{Q}_+}$$

Using Equation 6

$$= \left(\frac{1}{T} \sum_{\tau=1}^{T} \tilde{r}(\sigma^{(\tau)}, q_+) \right)_{q_+ \in \mathbb{Q}_+}$$

$$= \left(\frac{1}{T} \left(\left(\sum_{\tau=1}^{T-1} \tilde{r}(\sigma^{(\tau)}, q_+) \right) + \tilde{r}(\sigma^{(T)}, q_+) \right) \right)_{q_+ \in \mathbb{Q}_+}$$

$$= \left(\left(\frac{1}{T} \sum_{\tau=1}^{T-1} \tilde{r}(\sigma^{(\tau)}, q_+) \right) + \left(\frac{1}{T} \right) \tilde{r}(\sigma^{(T)}, q_+) \right)_{q_+ \in \mathbb{Q}_+}$$

$$= \left(\left(\frac{(T-1)}{T} \right) \left(\frac{1}{(T-1)} \sum_{\tau=1}^{T-1} \tilde{r}(\sigma^{(\tau)}, q_+) \right) + \left(\frac{1}{T} \right) \tilde{r}(\sigma^{(T)}, q_+) \right)_{q_+ \in \mathbb{Q}_+}$$

Using Equation 6

$$= \left(\left(\frac{(T-1)}{T} \right) \bar{r}^{(T-1)}(q_+) + \left(\frac{1}{T} \right) \tilde{r}(\sigma^{(T)}, q_+) \right)_{q_+ \in \mathbb{Q}_+}$$

$$= \left(\left(1 - \frac{1}{T} \right) \bar{r}^{(T-1)}(q_+) + \left(\frac{1}{T} \right) \tilde{r}(\sigma^{(T)}, q_+) \right)_{q_+ \in \mathbb{Q}_+}$$

$$= \left(\bar{r}^{(T-1)}(q_+) + \frac{1}{T} \left(\tilde{r}(\sigma^{(T)}, q_+) - \bar{r}^{(T-1)}(q_+) \right) \right)_{q_+ \in \mathbb{Q}_+}$$

$$= \left(\bar{r}^{(T-1)}(q_+) \right)_{q_+ \in \mathbb{Q}_+} + \frac{1}{T} \left(\left(\tilde{r}(\sigma^{(T)}, q_+) \right)_{q_+ \in \mathbb{Q}_+} - \left(\bar{r}^{(T-1)}(q_+) \right)_{q_+ \in \mathbb{Q}_+} \right)$$

Using Equation 32 and Equation 31

$$= \bar{\boldsymbol{r}}' + \frac{1}{T} \left(\tilde{\boldsymbol{r}} - \bar{\boldsymbol{r}}' \right)$$

G.11 Regret Normalizers

An expanded form of Equation 33 is shown below.

$$\bar{\boldsymbol{r}}^{(+,\Sigma)} = \left(\sum_{a' \in A(h_+)} \left(\bar{r}^{(T)}(h_+, a') \right)^+ \right)_{(h_+, a) \in \mathbb{Q}_+}$$

$$= \left(\sum_{(h'_+, a') \in \mathbb{Q}_+} \left(\mathbf{1}_{h_+ = h'_+} \right) \left(\bar{r}^{(T)}(h'_+, a') \right)^+ \right)_{(h_+, a) \in \mathbb{Q}_+}$$

$$= \left(\left(\mathbf{1}_{h_+ = h'_+} \right)_{((h_+, a), (h'_+, a')) \in \mathbb{Q}_+^2} \right) \left(\left(\bar{r}^{(T)}(q_+) \right)^+ \right)_{q_+ \in \mathbb{Q}_+}$$

$$= \left(\left(\sum_{h''_+ \in \mathbb{H}_+} \left(\mathbf{1}_{h_+ = h''_+} \right) \left(\mathbf{1}_{h'_+ = h''_+} \right) \right)_{((h_+, a), (h'_+, a')) \in \mathbb{Q}_+^2} \right) \left(\left(\bar{r}^{(T)}(q_+) \right)_{q_+ \in \mathbb{Q}_+} \right)^+$$

Using Equation 32

$$= \left(\left(\sum_{h''_+ \in \mathbb{H}_+} \left(\mathbf{1}_{h_+ = h''_+}\right)\left(\mathbf{1}_{h'_+ = h''_+}\right)\right)_{((h_+,a),(h'_+,a')) \in \mathbb{Q}_+^2}\right)\bar{\boldsymbol{r}}^+$$
$$= \left(\left(\mathbf{1}_{h_+ = h'_+}\right)_{((h_+,a),h'_+) \in \mathbb{Q}_+ \times \mathbb{H}_+}\right)\left(\left(\mathbf{1}_{h_+ = h'_+}\right)_{(h_+,(h'_+,a)) \in \mathbb{H}_+ \times \mathbb{Q}_+}\right)\bar{\boldsymbol{r}}^+$$
$$= \left(\left(\mathbf{1}_{h_+ = h'_+}\right)_{(h_+,(h'_+,a)) \in \mathbb{H}_+ \times \mathbb{Q}_+}\right)^\top \left(\left(\mathbf{1}_{h_+ = h'_+}\right)_{(h_+,(h'_+,a)) \in \mathbb{H}_+ \times \mathbb{Q}_+}\right)\bar{\boldsymbol{r}}^+$$

Using Equation 12

$$= \left(\boldsymbol{M}^{(H_+,Q_+)}\right)^\top \left(\boldsymbol{M}^{(H_+,Q_+)}\right)\bar{\boldsymbol{r}}^+$$

To take advantage of the sparsity of $\boldsymbol{M}^{(H_+,Q_+)}$ (see Table 4)

$$= \left(\boldsymbol{M}^{(H_+,Q_+)}\right)^\top \left(\left(\boldsymbol{M}^{(H_+,Q_+)}\right)\bar{\boldsymbol{r}}^+\right)$$

G.12 Next Strategy Profile

An expanded form of Equation 34 is shown below.

$$\boldsymbol{\sigma}' = \left(\sigma^{(T+1)}(q_+)\right)_{q_+ \in \mathbb{Q}_+}$$

Using Equation 7

$$= \left(\begin{cases} \begin{cases} \frac{(\bar{r}^{(T)}(h_+,a))^+}{\sum_{a' \in A(h_+)} (\bar{r}^{(T)}(h_+,a'))^+} & \sum_{a' \in A(h_+)} (\bar{r}^{(T)}(h_+,a'))^+ > 0 \\ \frac{1}{|A(h_+)|} & \sum_{a' \in A(h_+)} (\bar{r}^{(T)}(h_+,a'))^+ = 0 \end{cases} & (h_+,a) \in \mathbb{Q}_+ \\ \sigma_0(h_+,a) & (h_+,a) \in \mathbb{Q}_0 \end{cases}\right)_{(h_+,a) \in \mathbb{Q}_+}$$
$$= \left(\begin{cases} \frac{(\bar{r}^{(T)}(h_+,a))^+}{\sum_{a' \in A(h_+)} (\bar{r}^{(T)}(h_+,a'))^+} & \sum_{a' \in A(h_+)} (\bar{r}^{(T)}(h_+,a'))^+ > 0 \\ \frac{1}{|A(h_+)|} & \sum_{a' \in A(h_+)} (\bar{r}^{(T)}(h_+,a'))^+ = 0 \end{cases}\right)_{(h_+,a) \in \mathbb{Q}_+}$$

Using Equation 32, Equation 33, and Equation 16

$$= \left(\begin{cases} \left(\bar{\boldsymbol{r}}^+ \oslash \bar{\boldsymbol{r}}^{(+,\Sigma)}\right)_{q_+} & \left(\bar{\boldsymbol{r}}^{(+,\Sigma)}\right)_{q_+} > 0 \\ \left(\boldsymbol{\sigma}^{(T=1)}\right)_{q_+} & \left(\bar{\boldsymbol{r}}^{(+,\Sigma)}\right)_{q_+} = 0 \end{cases}\right)_{q_+ \in \mathbb{Q}_+}$$