Information-Theoretic Approach to Detect Collusion in Multi-Agent Games: Supplementary Material

Trevor Bonjour[1] **Vaneet Aggarwal**[2] **Bharat Bhargava**[1]

[1]Department of Computer Science, Purdue University, West Lafayette, Indiana, USA
[2]School of Industrial Engineering, Purdue University, West Lafayette, Indiana, USA

A STATE SPACE SPECIFICATIONS

A.1 ROCK, PAPER, SCISSORS

In iterated rock, paper, scissors we represent the state space as a combination of 'action history' and 'opponent action distribution'. We bound the action history to the last 20 actions taken by the players. The opponent action distribution is represented as a 3-dimensional vector with each element depicting the probability of taking a given action.

A.2 LEDUC HOLD'EM

For Leduc Hold'em, we represent the state space as a combination of the following attributes with the number of possible values shown in brackets: Current player position (3), current player hole card (6), current player raised (2), opponent position (3x2), opponent folded (2x2), opponent raised (2x2), board card (6), round (2), pot value (36), and action history (9x9).

B COLLUSION DETECTION ALGORITHM

Algorithm 1 gives the procedure for the collusion detection method for multi-agent games.

C ADDITIONAL EXPERIMENT: 4 PLAYER ROCK PAPER SCISSORS

We run an additional experiment for a 4 player version of rock paper scissors. We have the following players:

1. **Player A** : Primary colluding agent.
2. **Player B** : Assistant colluding agent.
3. **Player C** : Non-colluding agent.
4. **Player D** : Non-colluding agent.

Player A, C and D choose an action at random. Player B chooses an action that guarantees A a point with some *collusion probability*, and a random action otherwise.

We run an experiment similar to Section 5.1 Experiment 1 where we try to answer how the collusion strength affects the swiftness or sample complexity of the detection algorithm. We use data generated from games played between Players A, B, C and D (manual collusion). We run multiple simulations for a different number of games (sample size) and varying levels of collusion probability values. We plot the calculated net influence for different settings in Figure 1. Each graph in the figure is generated for the different CP values. The y-axis gives the net influence values and the x-axis gives the number of

Accepted for the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022).

Algorithm 1 Collusion Detection for Multi-Agent Games

```
Input: Game m record tuples <a, s, r> for n agents and collusion threshold α
Output: Colluding pair if any.
Initialize n × n matrix I                         ▷ for pair-wise individual influence
Initialize n × n matrix N                         ▷ for pair-wise net influence
Initialize count ← 0                              ▷ count of colluding pairs
Initialize c1, c2 ← −1                            ▷ index of colluding agents
for i = 1 : n do                                  ▷ Construct policy matrix for each agent
    Construct π^i                                 ▷ from Equation (12)
end for
for i = 1 : n do                                  ▷ Construct joint policy matrix pair of agents
    for j = 1 : n do
        if i ≠ j then
            Construct π^{ij}, π^{ji}              ▷ from Equation (13)
            I[i, j] ← γ(i; j)                     ▷ from Equation (3)
            I[j, i] ← γ(i; j)                     ▷ from Equation (3)
        end if
    end for
end for
for i = 1 : n do                                  ▷ Calculate net influence
    for j = 1 : n do
        if i ≠ j then
            N[i, j] ← Γ(i; j)                     ▷ from Equation (4)
            N[j, i] ← Γ(i; j)                     ▷ from Equation (4)
        end if
    end for
end for
for i = 1 : n do
    for j = 1 : n do
        if N[i, j] ≥ α and N[j, i] ≥ α then
            count ← count + 1
            c1 ← i
            c2 ← j
            if count > 1 then
                return No Collusion
            end if
        end if
    end for
end for
if count == 1 then
    return c1, c2
else
    return No Collusion
end if
```

games used to calculate the net influence values. The dashed horizontal line in each graph depicts the collusion threshold α which is set at 0.05. Note that, as the CP values go higher, our algorithm can detect collusion using data from fewer games. The results are similar to what we get for the 3 player version of Rock, Paper, Scissors.

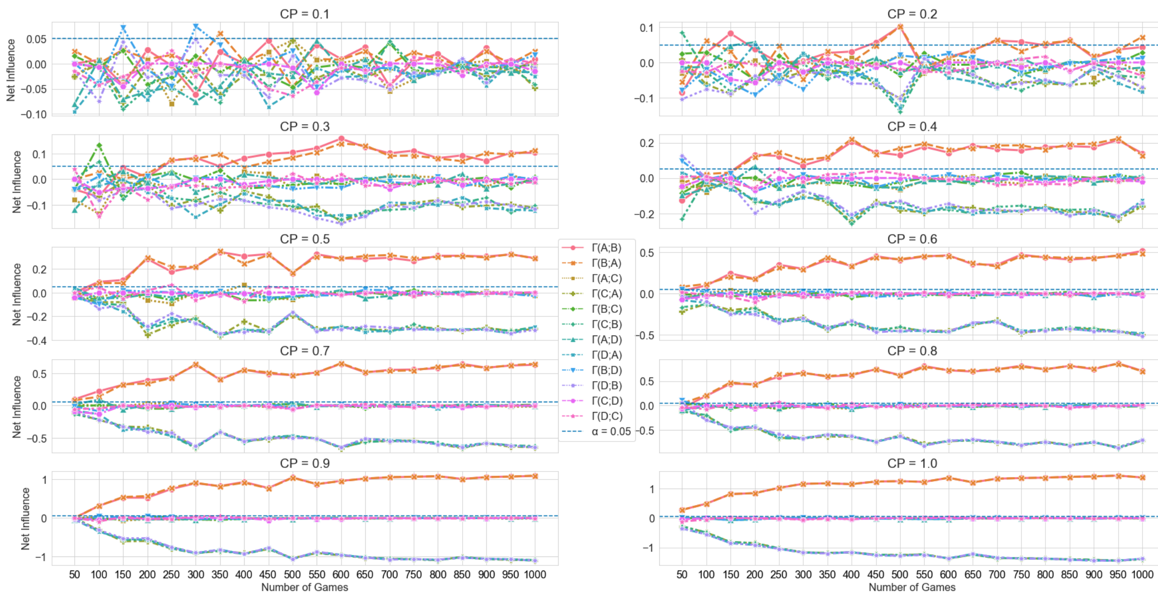

Figure 1: Net Influence Calculated for a Different Number of Games for Varying Values of Collusion Probability (CP) for Rock Paper Scissors.