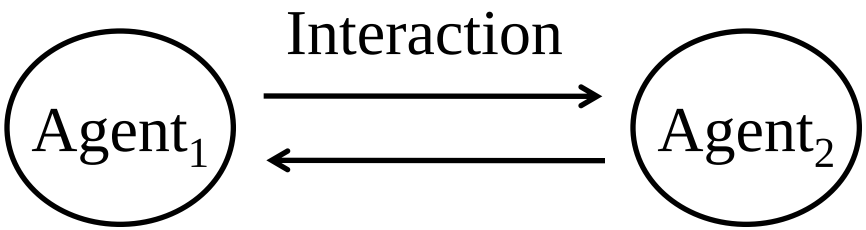
Accident and Intention: That is the Question (in the Noisy Iterated Prisoner's Dilemma)

Tsz-Chiu Au
University of Maryland, College Park
Department of Computer Science

Joint work with Prof. Dana Nau and Prof. Sarit Kraus

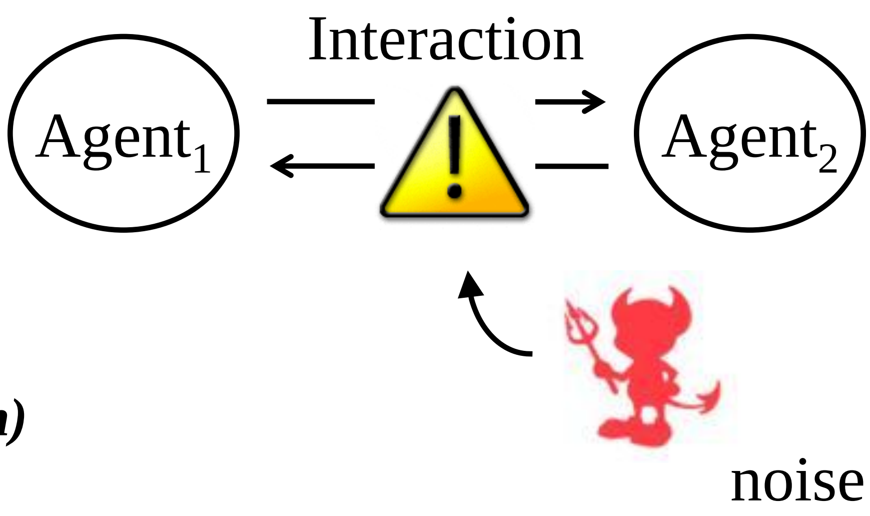
Social Games

- We are interested in modeling several *self-interested* parties (i.e., agents) that repeatedly interact with each other
  - » Governments
  - » Terrorist groups
  - » Crime organizations
  - » Civilians



- Performance of agents depends on how well they interact with each other
- Objective: decide how to interact with different parties so as to maximize the performance of an agent

Limitations of Existing Techniques



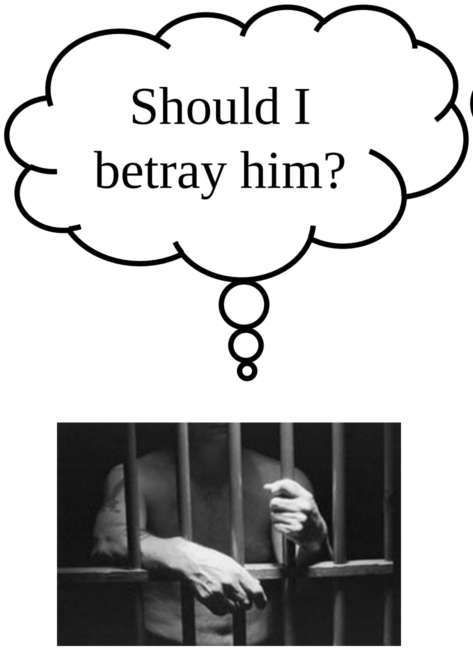
- Most existing game-theoretic research
  - » Studies problems at an abstract mathematical level
  - » Neglects many features that can occur in the real world
- This talk focuses mainly on one such feature: ***noise (i.e., errors in interaction)***
- Two kinds of noise:
  - » *Accidents*: An agent intends to do *X*, but accidentally does *Y* instead
    - e.g., slippery hands, car accidents, sabotage by "invisible" spies
  - » *Misinterpretations*: One agent thinks another agent's action was different from what the other agent actually did
    - e.g., miscommunication due to poorly-written email, dirt on eye-glasses
- Noise can cause serious problems
- How to deal with noise?

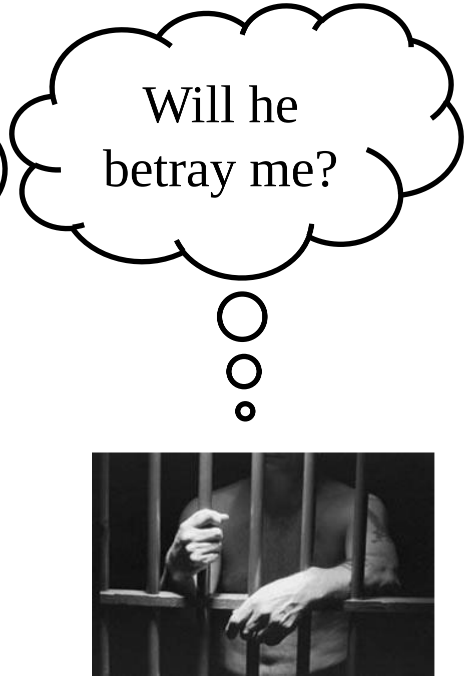
I Will Discuss

- Iterated Prisoner's Dilemma (IPD)
  - » A game that is widely used for studying cooperation among agents.
- Noisy Iterated Prisoner's Dilemma
  - » IPD with error in the interaction
- Symbolic Noise Detection
  - » Ways to
    - Build model of the other player, based on their past behavior
    - Use the model to detect noise
- Derived Belief Strategy (DBS)
  - » A strategy for playing IPD in noisy environment
    - Minimize the effect of noise in the learning of opponent model using symbolic noise detection
    - More accurate opponent model $\Rightarrow$ higher quality of decision
- Performance
  - » 20th Anniversary IPD Competition
- Generalizations
  - » Other games similar to IPD

Prisoner's Dilemma

- Two suspects are arrested by the police (and become prisoners)
- But the police does not have sufficient evidence for a conviction.
- The police want the prisoners to betray each other.





Prisoner$_2$ / *Prisoner*$_1$	Cooperate (Not Betray)	Defect (Betray)
Cooperate (Not betray)		
Defect (Betray)		

Prisoner's Dilemma (cont.)

- What will the prisoners do?
  - Sadly, both prisoners betray each other
    - because they are self-interested and they do not completely trust each other.

No matter what the other prisoner does, I better defect

Prisoner$_2$ / *Prisoner*$_1$	Cooperate	Defect
Cooperate	1, 1	5, 0
Defect	0, 5	3, 3

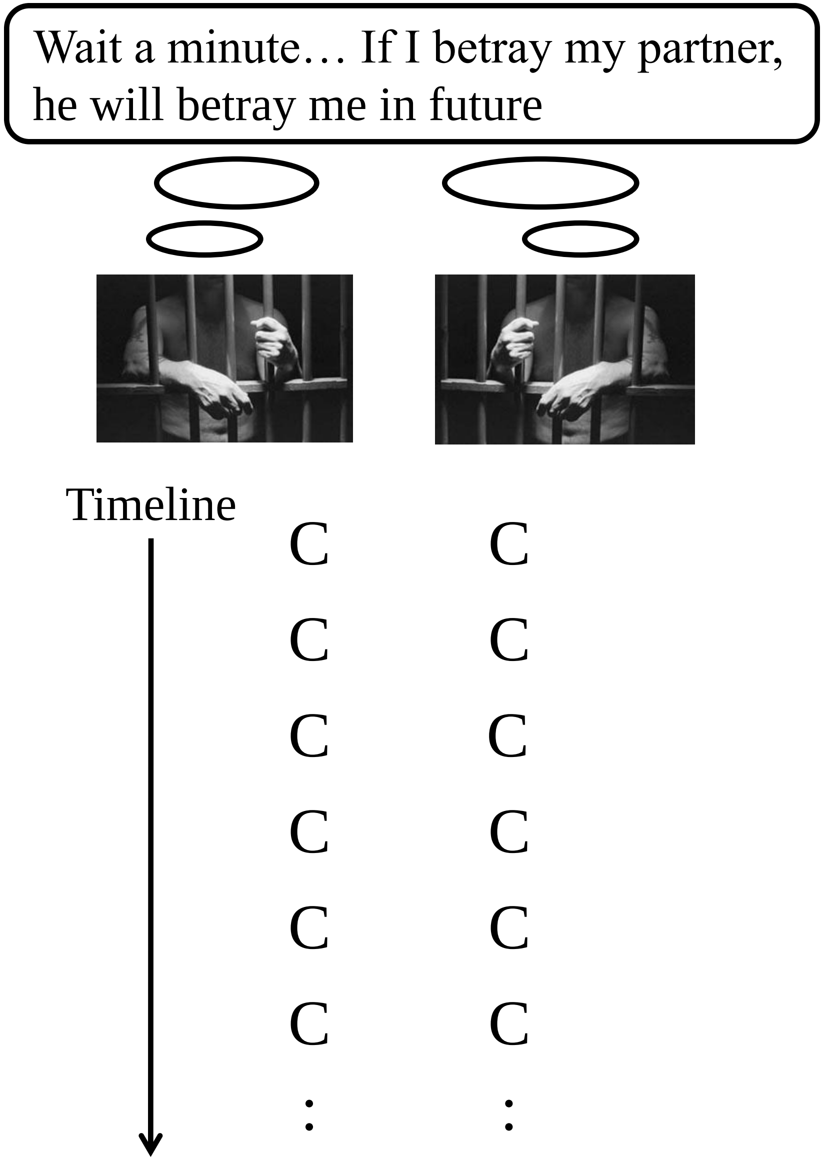
Nash Equilibrium

Iterated Prisoner's Dilemma

- But this game is *not* a zero-sum game.
  - » The prisoners can think "win-win"
- How can the prisoners cooperate with each other?
- Solution: if the same prisoners play the game many many times…
  - » "punish" the other prisoner if he does not cooperate in the past

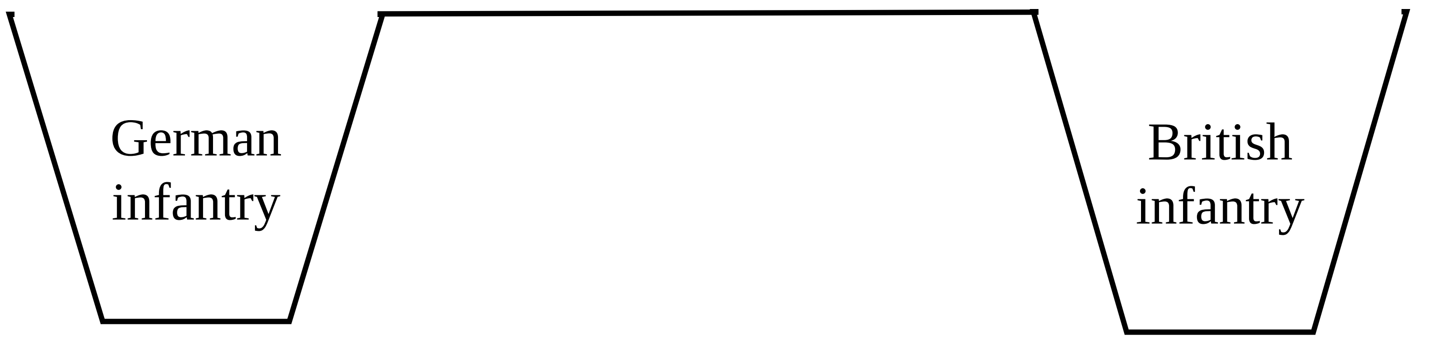
» Due to this threat of punishment, cooperation *can* (not must) emerge

- The new game is called ***Iterated Prisoner's Dilemma*** **(IPD)**



Example:

- A real-world example of the IPD, described in Axelrod (1997):
  - » Trench warfare in World War I



- Incentive to cooperate:
  - » If I attack the other side, then they'll retaliate and I'll get hurt
  - » If I don't attack, maybe they won't either
- Result: emergence of cooperation
  - » Even though the two infantries were *supposed* to be enemies, they avoided attacking each other
- This was one of the reasons why World War I lasted so long

Evolution of Cooperation

- **Question:** In general, under what conditions will cooperation emerge in a world of egoists without central authority?
- **Answer:** hard to tell
- IPD is the "fruit fly" in the study of evolution of cooperation
  - » *"The Selfish Gene"* by Richard Dawkins (1976)
  - » *"The Evolution of Cooperation"* by Robert Axelrod (1984)
    - His paper on IPD was cited 2000 times!
- A cross-disciplinary topic
  - » political science (foreign policy)
  - » economics and business (trading, auction)
  - » biology (evolution, animal behaviors),
  - » social science (group behaviors)
  - » and lately, computer science (peer-to-peer computing, bitTorrent)
  - » How about AI? (multi-agent system, computer games)

Strategies for the IPD

- But the threat of punishment (or the shadow of the future) does not necessarily result in cooperation.
  - It still depends on how the other players treat you.
- There are many different strategies for playing IPD:
  - ***ALLC***: always cooperate
  - ***ALLD***: always defect
  - ***GRIM***: cooperate until the other player defects, then defect forever
  - ***TESTER***: defect every other move if the other player does not retaliate for detection.
- Question: which strategy is the best?
  - In general, there is no optimal strategy for IPD
    - Performance depends on the other player's strategy, which is not known beforehand
- A famous political scientist called Axelrod proposed to use computer simulation to study which strategy can usually perform well.
  - The Iterated Prisoner's Dilemma Tournament
    - Ask people to submit programs to the tournament.
    - Every program plays with every other program.
    - Winning program is the one with the highest average score.

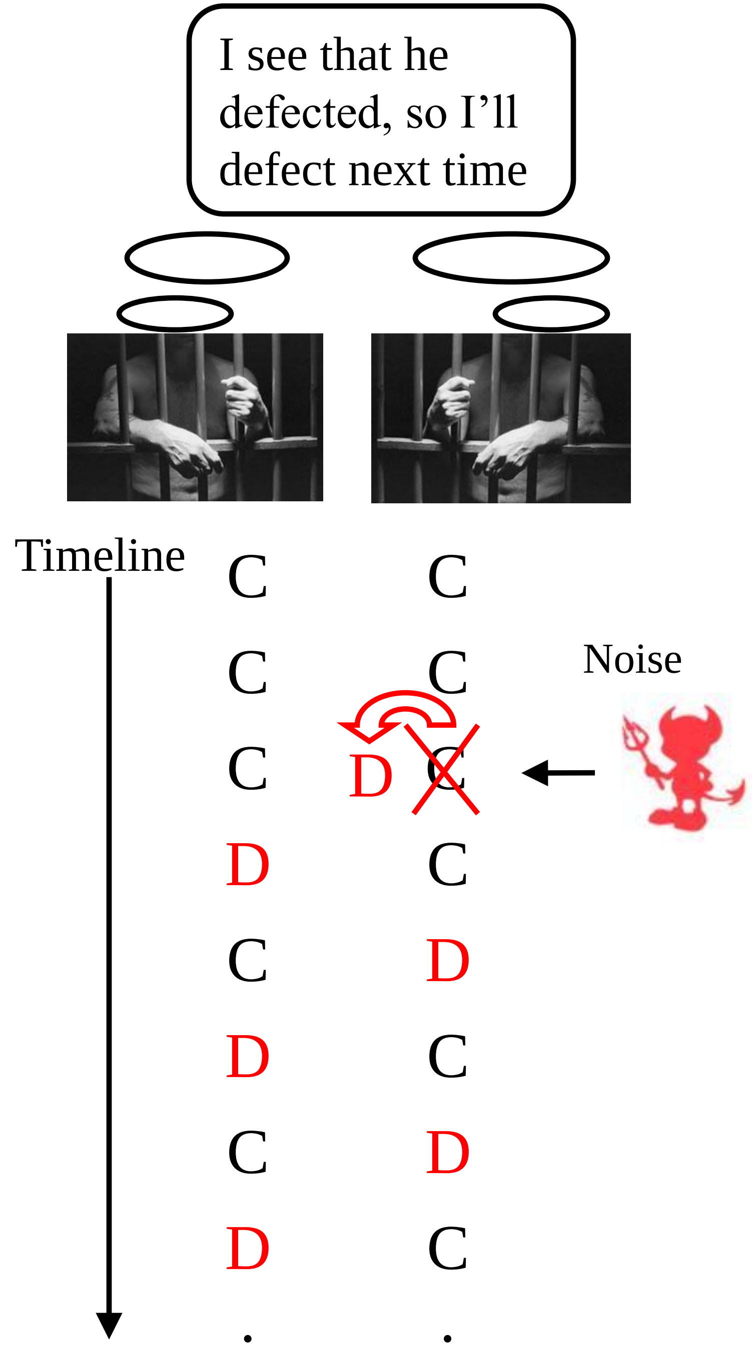
Results of the Axelrod's IPD tournaments

- A strategy called ***Tit-for-Tat (TFT)*** has been shown to cooperate extremely well with a wide variety of opponents
  - » Cooperate on the first move. On the *n*th move, repeat the other player's (*n*–1)-th move
    - It could establish and maintain cooperation with many other players
    - It could prevent malicious players from taking advantage of it
  - » TFT won several open IPD tournaments.
- Extremely influential: TFT becomes a key principle in foreign policy making

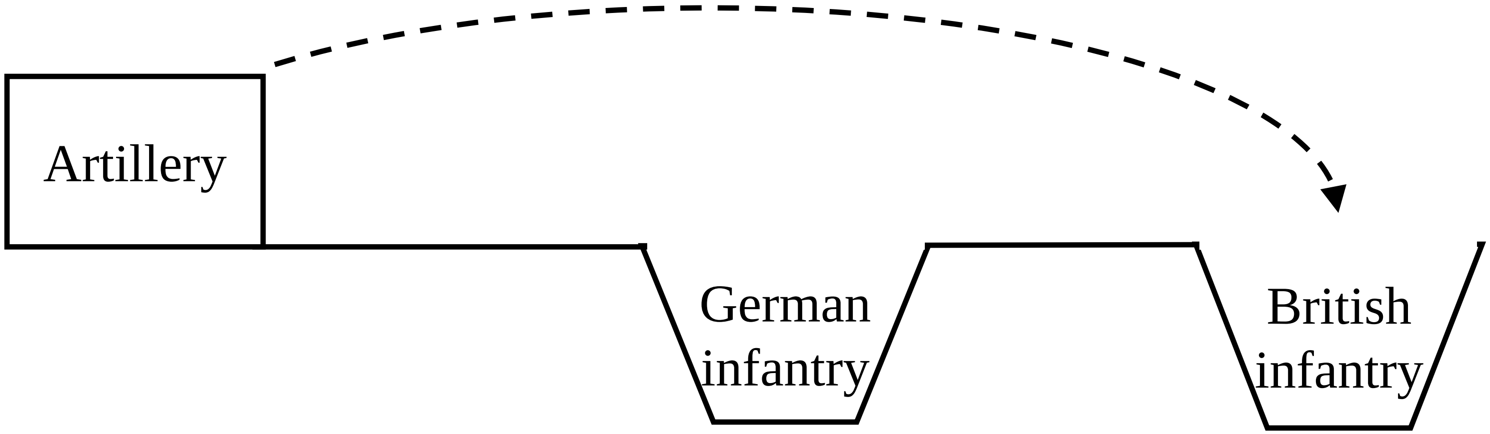
TFT	*AllC*	*TFT*	*AllD*	*TFT*	*Grim*	*TFT*	*TFT*	*TFT*	*Tester*
C	C	C	*D*	C	C	C	C	C	*D*
C	C	*D*	*D*	C	C	C	C	*D*	C
C	C	*D*	*D*	C	C	C	C	C	C
C	C	*D*	*D*	C	C	C	C	C	C
C	C	*D*	*D*	C	C	C	C	C	C
C	C	*D*	*D*	C	C	C	C	C	C
:	:	:	:	:	:	:	:	:	:

IPD with Noise

- But there is a serious problem with TFT…
- In noisy environments,
  - » There is a nonzero probability that *Cooperate* (C) will become *Defect* (D), and vice versa
  - » e.g., 10% chance that a "noise gremlin" will change some of the actions
- Noise causes big problems for strategies like Tit-for-Tat
  - » Mutual defection occurs
- How can we survive if all foreign policies are based on TFT?



Example of Noise



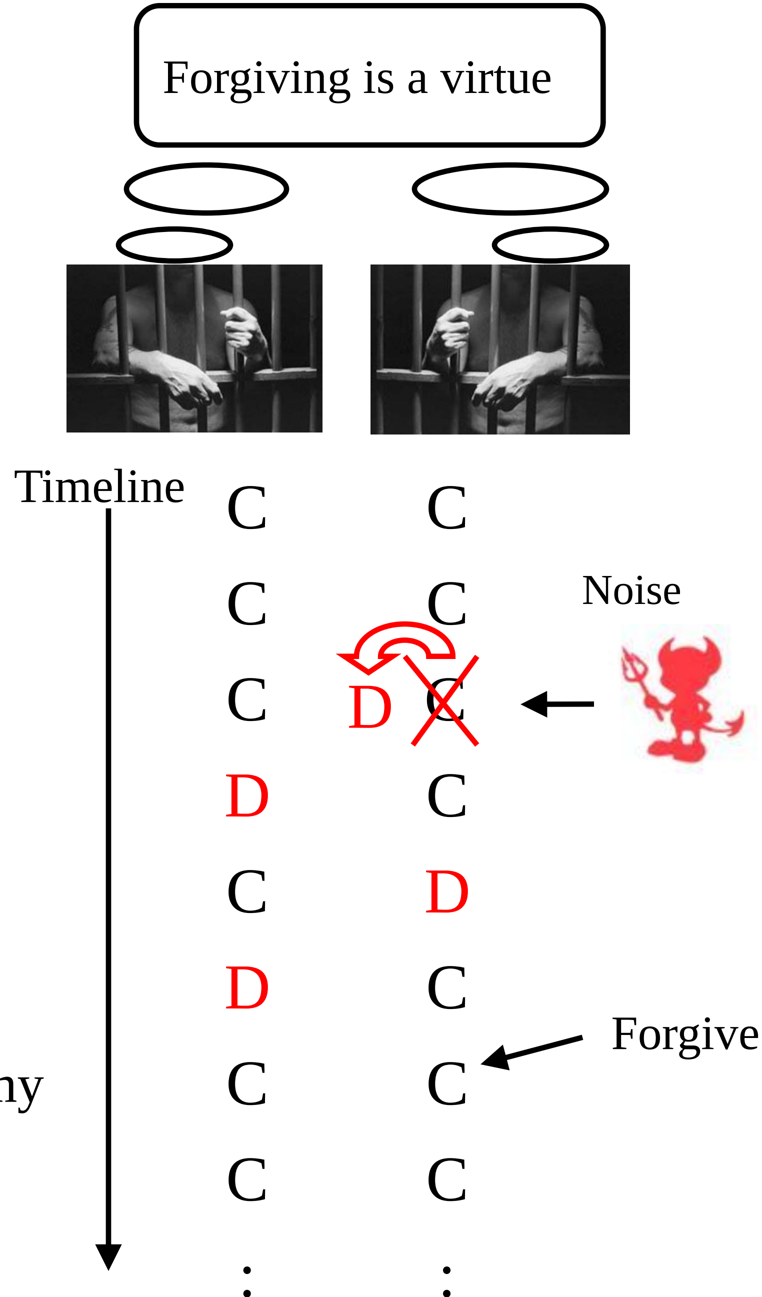
- Story from a British army officer in World War I:

"I was having tea with a company when we heard a lot of shouting and went out to investigate. We found our men and the Germans standing on their respective parapets. Suddenly a bomb arrived but did no damage. Naturally both sides got down and our men started swearing at the Germans,…."

- If you were the British army officer, what would you do?
  - Fight back (consider the bombing is intentional)
  - Don't fight back (consider the bombing is just an accident)

Existing Approach for Coping with Noise

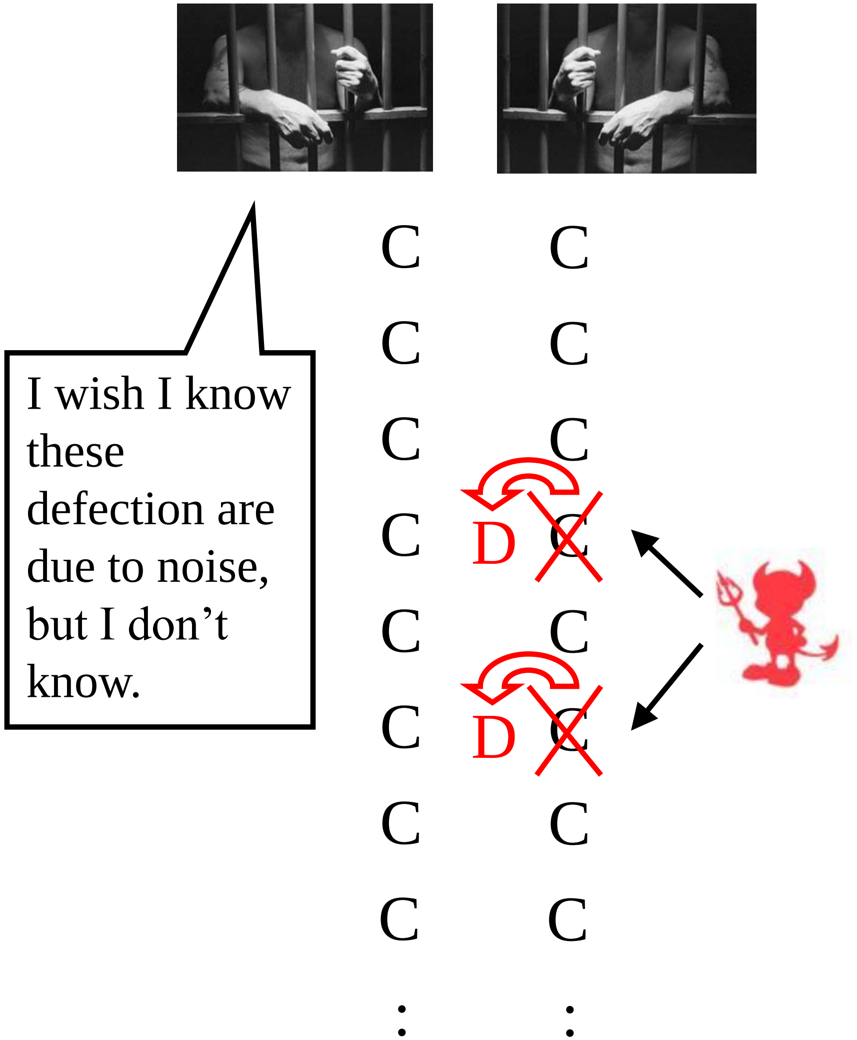
- **Principle: More forgiving, don’t retaliate**
- Tit-For-Two-Tats (TFTT)
  - » Retaliates only when it receives two defections in two previous iterations
  - » But susceptible to exploitation of its generosity
- Generous Tit-For-Tat (GTFT)
  - » Forgive randomly: randomly choose to cooperate with a small probability if the other player defects
    - avoided exploitation
- Pavlov
  - » Repeat previous move if I earn a lot of points in the previous iteration
  - » If the other player defects continuously, my agents will defect alternatively.
    - Punish less for mutual defections



Forgiving at will

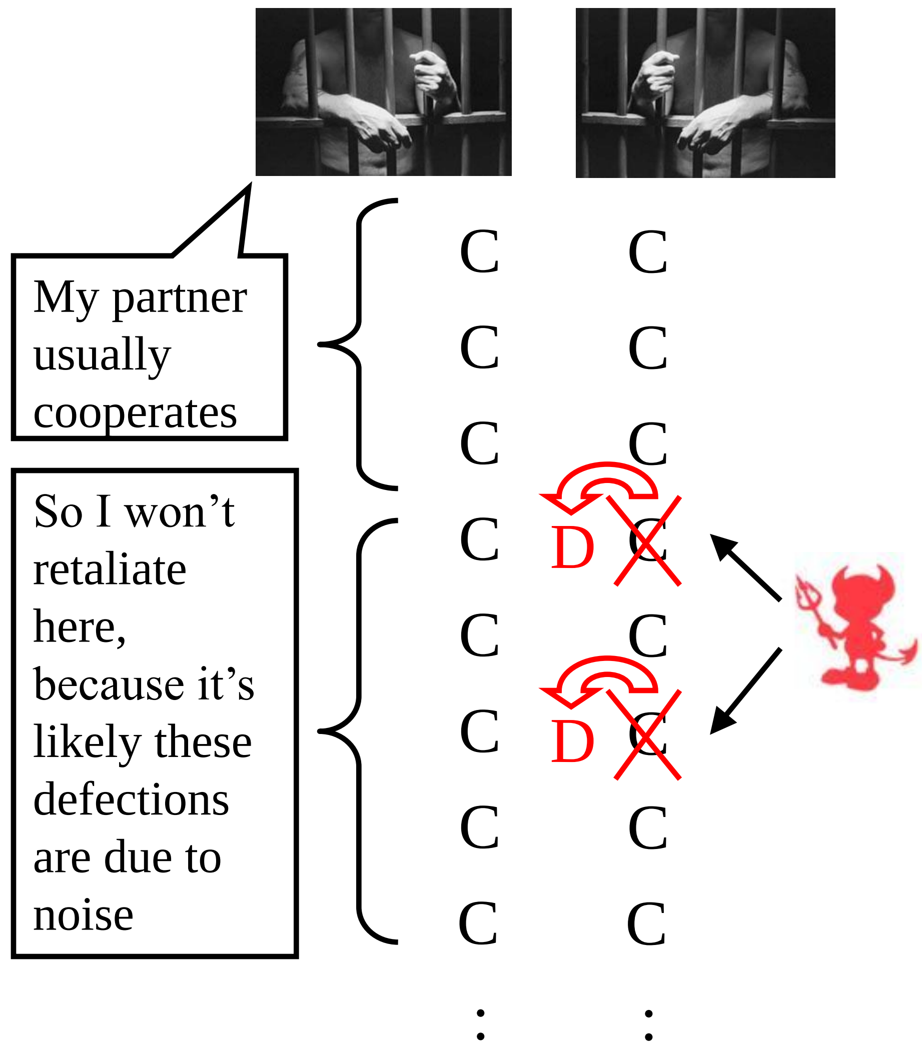
- The existing approaches are not very effective.
  - » Forgiving at will
  - » Sometimes they forgive too often, sometimes, they forgive too little.
- We think a better approach should avoid forgiving too often or too little.

- Observation: if the players knew exactly which actions were affected by noise, the conflict could be avoided.
- Unfortunately, players can never be sure about which actions were actually affected by noise
- But what if players can actually make good guesses of which actions have been affected by noise?

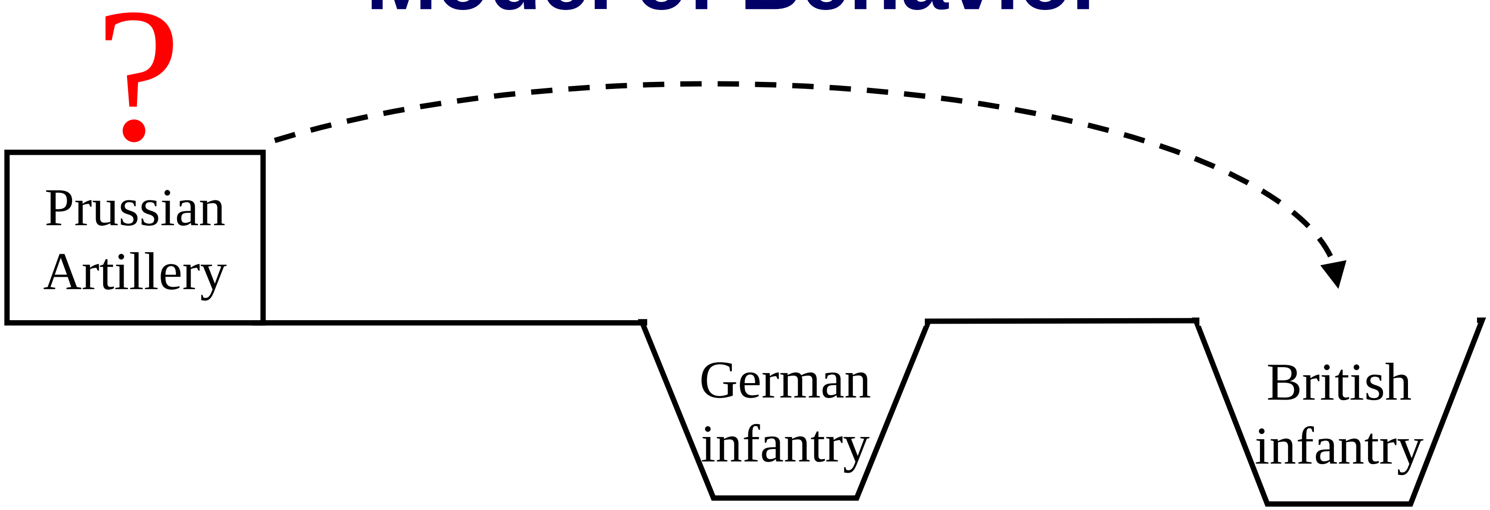


A New Approach for Coping with Noise

- Under what condition players can make good guesses about which actions have been affected by noise?
  - The other player plays *predictably*.
  - For example, if the other player used to cooperate in the past, there is a high chance that a sudden detection is due to noise.
- Our Approach: use a *deterministic model of behavior* to tell whether an action has been affected by noise
  - For highly predictable actions, build a deterministic model to predict them
  - Use it to tell whether an action has been affected by noise



Noise Detection based on Deterministic Model of Behavior



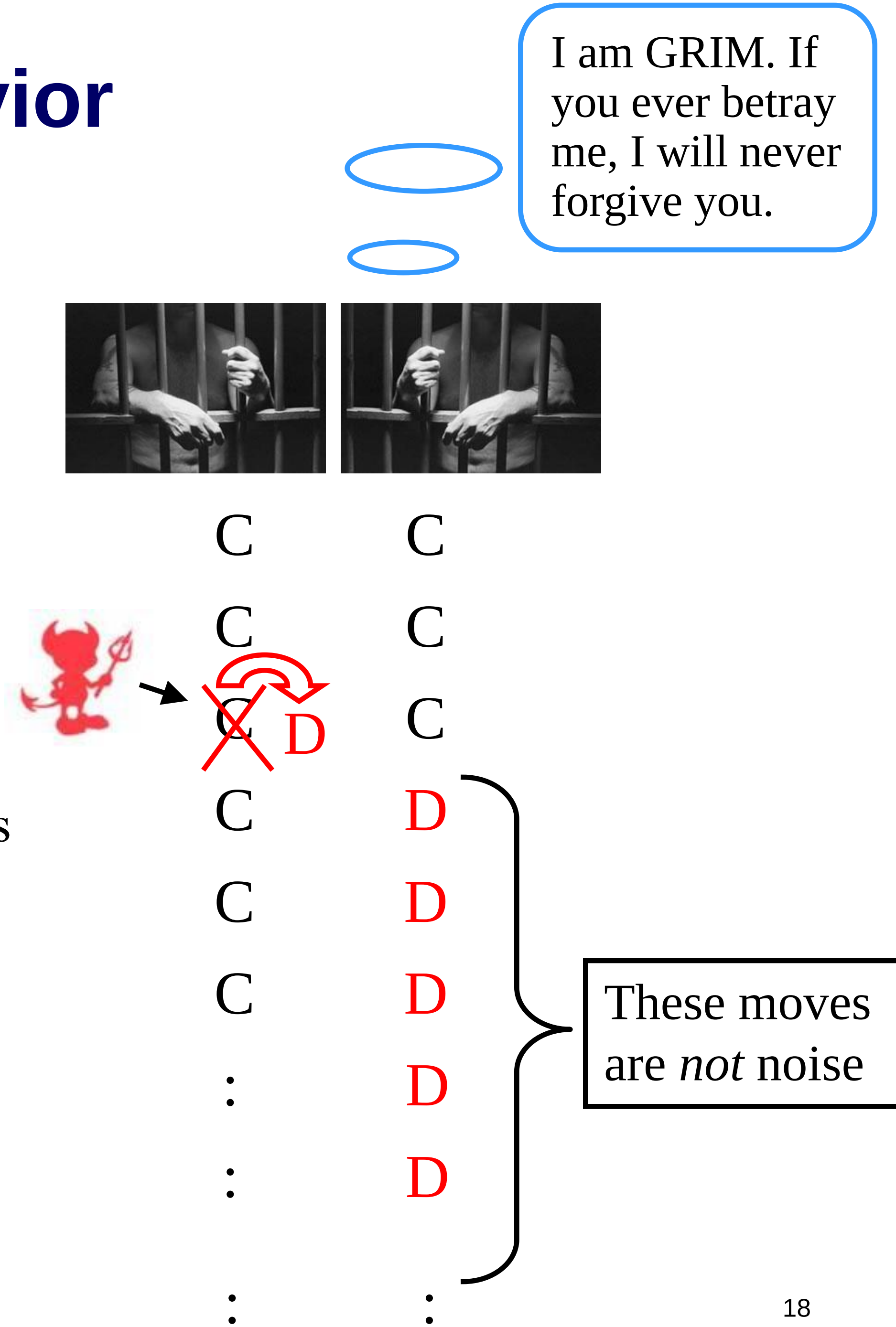
- Then the story went on:

"… our men started swearing at the Germans, when all at once a brave German got onto his parapet and shouted out: "We are very sorry about that; we hope no one was hurt. It is not our fault. It is that damned Prussian artillery""

- If you were the British army officer, did you believe the brave German?
  - Do you believe the brave German if the bombing is not consistent with the German infantry's past behavior?

Change of Behavior

- However, there is a problem..
  - » What if German really attack us?
- If the other player genuinely changes its behavior, the deterministic model learnt from the past is no longer valid.
  - » Cannot be used to detect noise
- Changes of behavior occur because
  - » The player can change its strategy anytime
  - » E.g., if noise affects one of Player 1’s actions, this may trigger a change in Player 2’s behavior
    - Player 1 does not know this
- How to distinguish noise from a real change of behavior?
  - » Is an anomaly of observed behavior accidental or intentional?



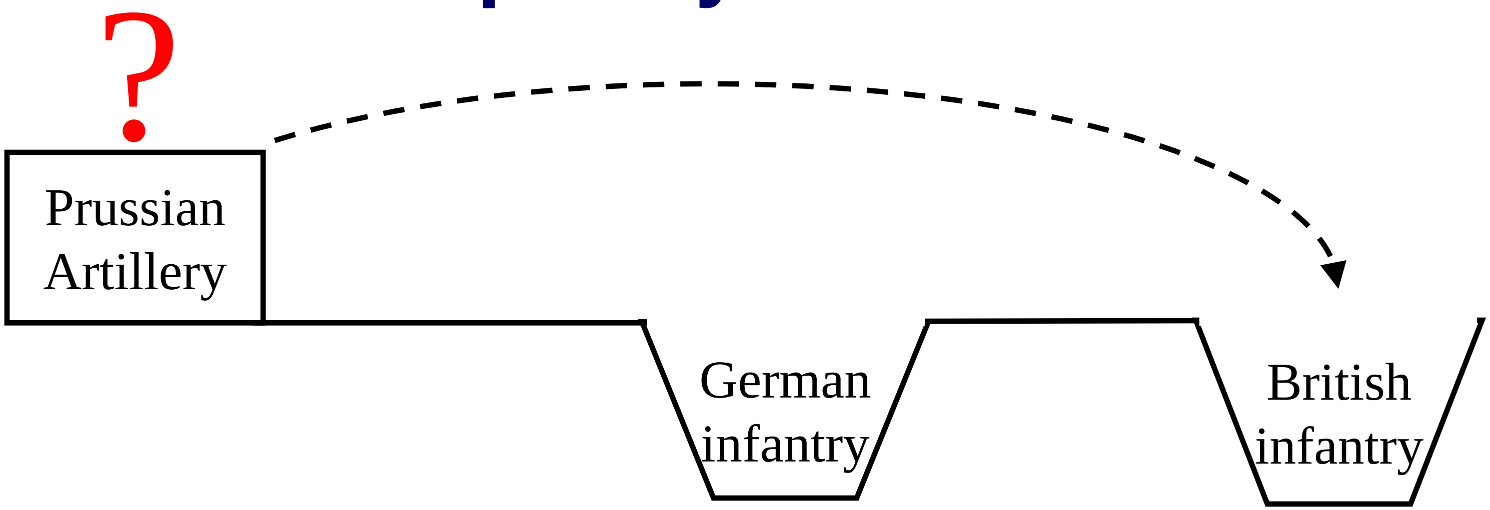
Detection of a Change of Behavior

Temporary Tolerance:

- When we observe unexpected behavior from the other player,
  - » Don't immediately decide whether it's noise or a real change of behavior
  - » Instead, defer judgment for a few iterations
- Why?
  - » We can often distinguish noise from a change of behavior by observing moves in the next few iterations and gathering enough evidence


My partner usually cooperates, so I will too	C	C
	C	C
	C	C
The defections might be accidents, so I shouldn't lose my temper too soon	C	*D*
	C	*D*
	C	*D*
I think his behavior has really changed, so I'll change mine too	*D*	*D*
	D	*D*
	:	:

Temporary Tolerance



- If you were the British army officer, do you believe the brave German?
  - Don’t believe *immediately* when the bombing occurs, even if this is not consistent with the German infantry’s past behavior.
  - Temporary tolerate the bombing, as if it did not occur
  - After a short period of time,
    - If the German infantry’s behavior continues to consistent with their past behavior, we consider this is an accident, as suggested by the brave German.
    - Otherwise, this means war.

Symbolic Noise Detection (SND)

A summary of SND:

- Build a deterministic model of how the other player behaves
- Watch for any *deviation* from the *deterministic behavior* predicted by the model
- If a deviation occurs, check to see if the inconsistency persists in the next few iterations
  - » If the inconsistency does not persist, assume the deviation is due to noise
  - » If the inconsistency persists, assume there is a change in the behavior

Why does SND be effective in IPD?

- Deterministic behavior is abundant in IPD
  - » In IPD, it is useful to cooperate with other players
    - ○ Most players are cooperative
  - » Clarity of behavior is an important ingredient of long-term cooperation
    - ○ To cooperate effectively, you need to give the other player a good idea how you will react
    - ○ As observed by Axelrod in his analysis of TFT
- Therefore, we can often build a deterministic behavioral model of the other player easily.
- Thus, noise are often detectable by using SND.

Derived Belief Strategy (DBS)

- *Derived Belief Strategy (DBS)* is a strategy based on symbolic noise detection
  - » Learn and maintain a model π of the other player's behavior
    - π is partly deterministic, partly probabilistic
    - Initially, π is assumed to be TFT
    - π is updated as the game progresses
  - » π has two primary functions:
    - Use the deterministic part of π to do symbolic noise detection
    - Use π to predict the other player's future moves, for decision-making based on game-tree search
- Basic idea: use symbolic noise detection to detect and isolate noise so as to keep π as accurate as possible for decision-making
  - » High accurate π ⇒ High quality decisions.

DBS Example

Us

Them

Opponent model π = TFT predicts C, so we do C too

Us	Them
C	C
C	C
C	C

These moves are what π = TFT predicts, so don't change π

Opponent model still is π = TFT, so we still do C

Us	Them
C	*D*
C	*D*
C	*D*

These Ds are not what π predicted. Temporarily tolerate them, as if they were C instead

Here, we decide the deviations are a change of behavior. Compute a new opponent model, π'

The new opponent model π' predicts D, so we do D too

Us	Them
D	*D*
D	*D*
:	:

Representing the Opponent Model

- We model the opponent's behavior as a *policy* represented as a set of *rules* of such as

  (*my last move*, *their last move*) $\rightarrow$ *p*

Moves in the previous iteration

Probability that the opponent will cooperate in this iteration

  » A rule is *deterministic* if *p* is 0.0 or 1.0
  » Otherwise the rule is *probabilistic*

- TFT can be modeled using deterministic rules:
  » $\pi = \{(C,C) \rightarrow 1.0,$
  $(C,D) \rightarrow 1.0,$
  $(D,C) \rightarrow 0.0,$
  $(D,D) \rightarrow 0.0\}$

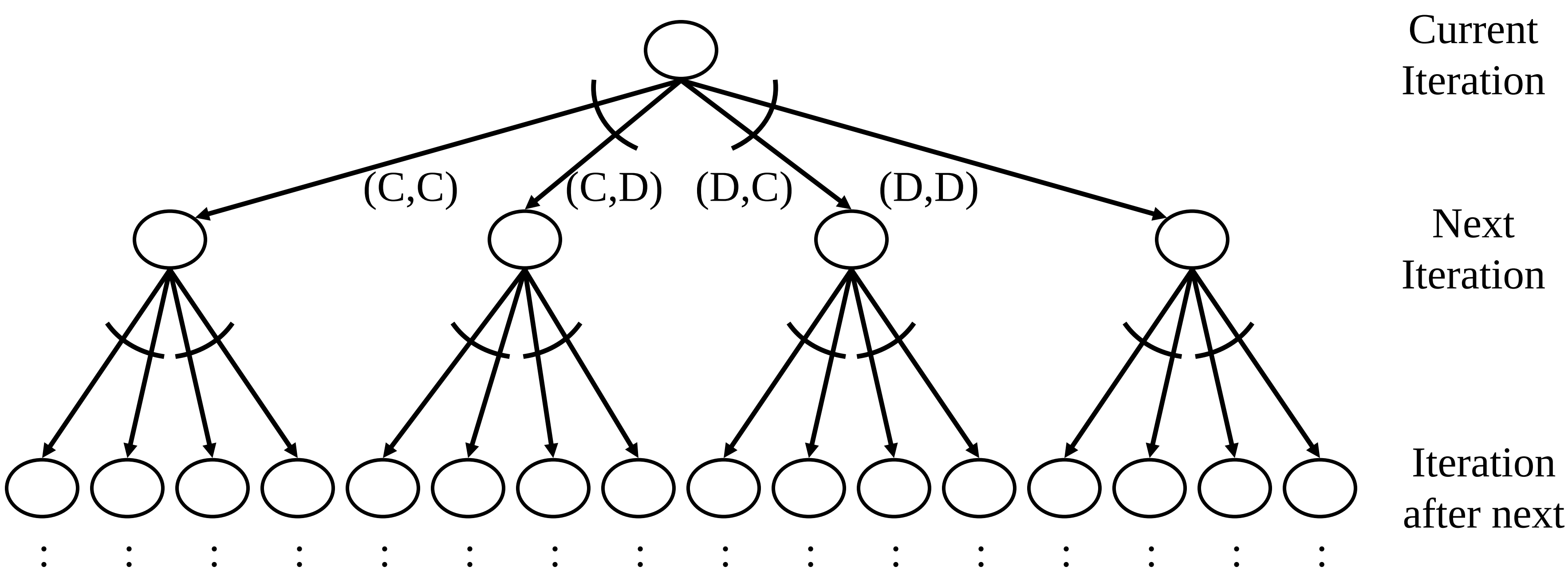
Suppose we have the rule
(C,C) $\rightarrow$ 0.7


C	C
C	*D*
D	C
C	C
	??

The rule predicts
$P(C) = 0.7$, $P(D) = 0.3$

Move Generator in DBS

- Basic idea: game-tree search
  - » Use the policy π to predict probabilities of the other player's moves
  - » Compute expected utility EU(x) for move x
    - EU(x) is the estimated score after the player choose the move x
  - » Choose a move with the highest expected utility



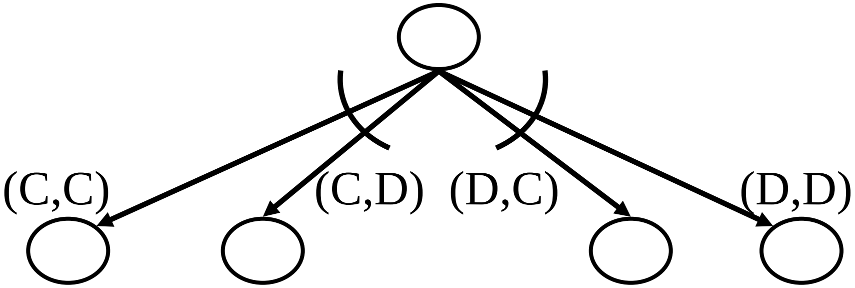
Suppose we have the rules

1. (C,C) → 0.7
2. (C,D) → 0.4
3. (D,C) → 0.1
4. (D,D) → 0.1

C	C
C	D
D	C
C	C
	??

Rule 1 predicts
$P(C) = 0.7$, $P(D) = 0.3$

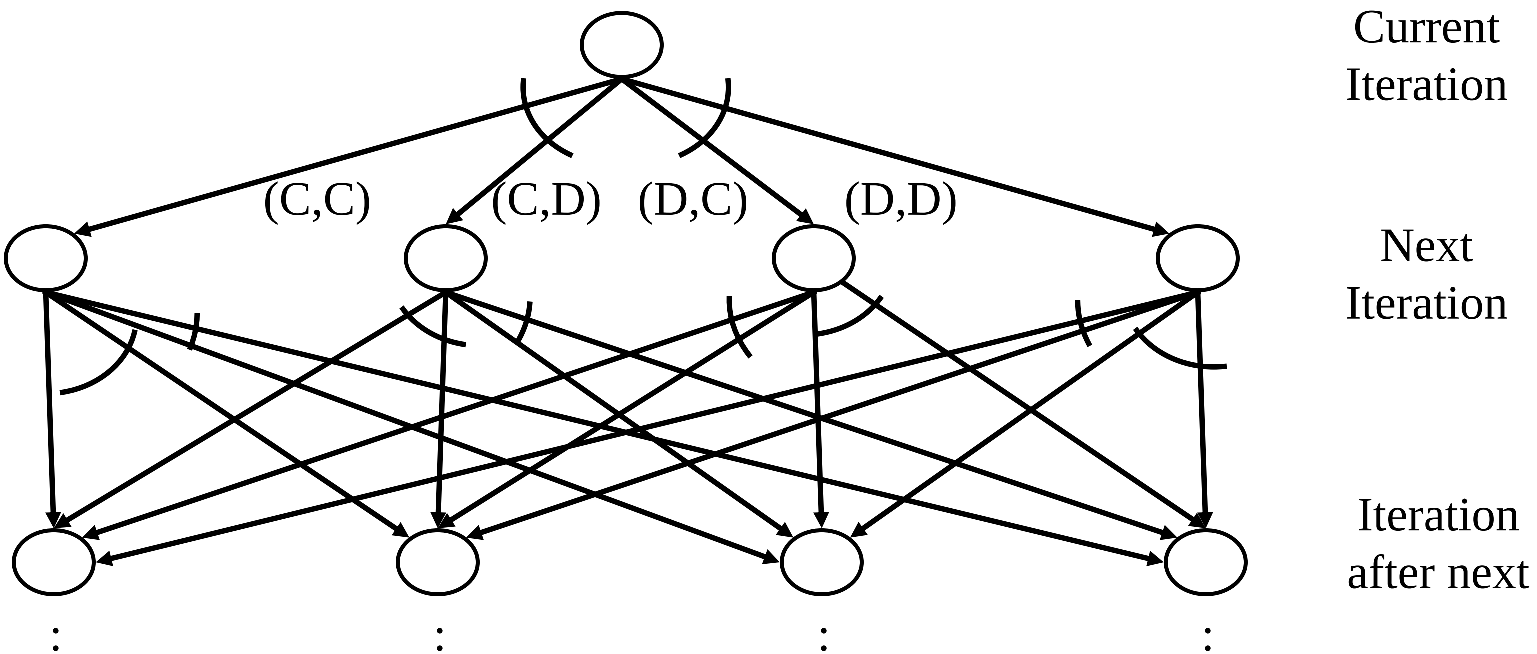
Example



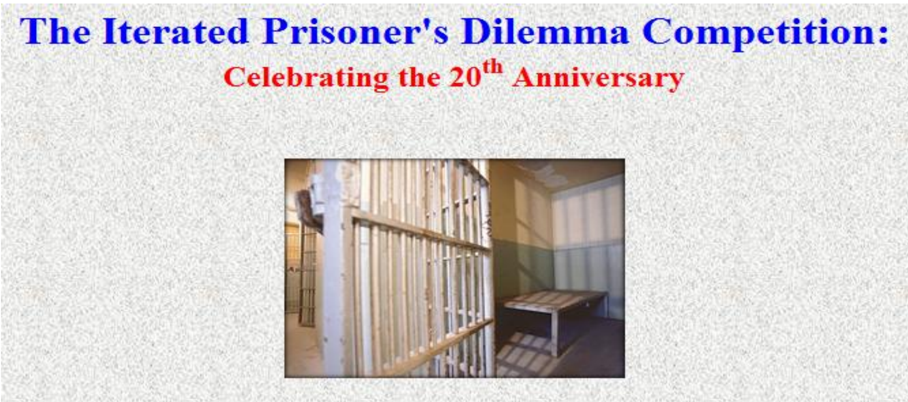
- Suppose we search to depth 1
  - » $EU(\mathrm{C}) = U(\mathrm{C,C}) \times 0.7 + U(\mathrm{C,D}) \times 0.3$
    $= 3 \times 0.7 + 0 \times 0.3 = 2.1$
  - » $EU(\mathrm{D}) = U(\mathrm{D,C}) \times 0.7 + U(\mathrm{D,D}) \times 0.3$
    $= 5 \times 0.7 + 1 \times 0.3 = 3.8$
  - » So D looks better
- Unwise to choose D
  - » On the move at depth 2, the opponent will retaliate with P=0.9
- If we search deeper, we'll see this and we'll choose C instead

How to Search Deeper

- Game trees grow exponentially with search depth
  - » How to search to the tree deeply?
- Assumption: the other player's future behavior will continue to be π
  - » Our opponent's decisions depend only on the previous moves as described in the policy π
- Then we can use dynamic programming
  - » Makes the search polynomial in the search depth
  - » Can easily search to depth 60
  - » This generates fairly good moves



The 20th-Anniversary IPD Competition



http://www.prisoners-dilemma.com

- Category 2: Noisy IPD
  - » Play each program against all programs, including itself
  - » Noise level = 10%
- 165 programs participated

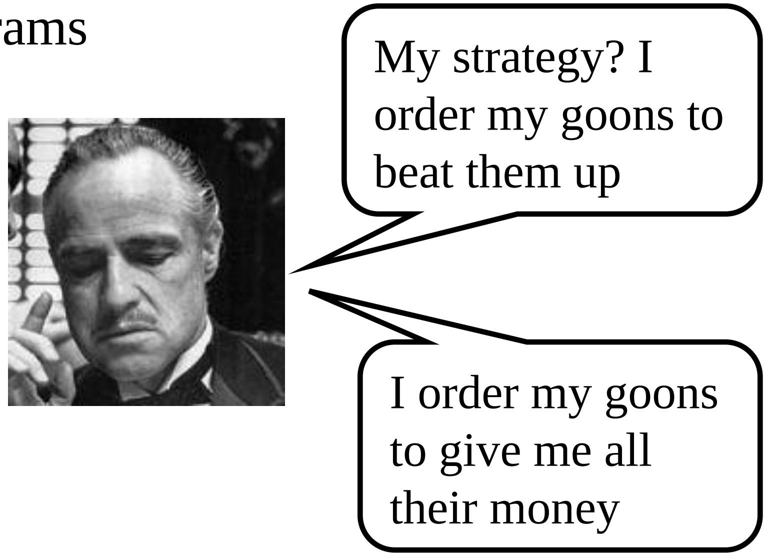
Rank	Program	Avg. score
1	BWIN	433.8
2	IMM01	414.1
3	DBSz	408.0
4	DBSy	408.0
5	DBSpl	407.5
6	DBSx	406.6
7	DBSf	402.0
8	DBStft	401.8
9	DBSd	400.9
10	lowESTFT_classic	397.2
11	TFTIm	397.0
12	Mod	396.9
13	TFTIz	395.5
14	TFTIc	393.7
15	DBSe	393.7
16	TTFT	393.4
17	TFTIa	393.3
18	TFTIb	393.1
19	TFTIx	393.0
20	mediumESTFT_classic	392.9

DBS dominated the top ten places, but ...

DBS was beaten by two other programs: BWIN and IMM01

How BWIN and IMM01 worked

- Each participant could submit up to 20 programs
- Some participants submitted 20 programs that worked as a team
    - 1 *master* + 19 *slaves*
  - » When slaves play with master
    - they cooperate and master detects
    - master gets all the points
  - » When slaves play with anyone not in their team, they defect
- Analysis
  - » The average score of each master-and-slaves team was much lower than DBSz’s average score
  - » If BWIN and IMM01 each had ≤ 10 slaves, DBS would have placed 1st
  - » If BWIN and IMM01 had no slaves, they would have done badly



DBS cooperates, not coerces

- Unlike BWIN and IMM01, DBS had *no* slaves
  - » None of the DBS programs even knew the others were there
- DBS worked by establishing cooperation with *many* other agents
- DBS could do this *despite* the noise, because it could filter out the noise

Summary

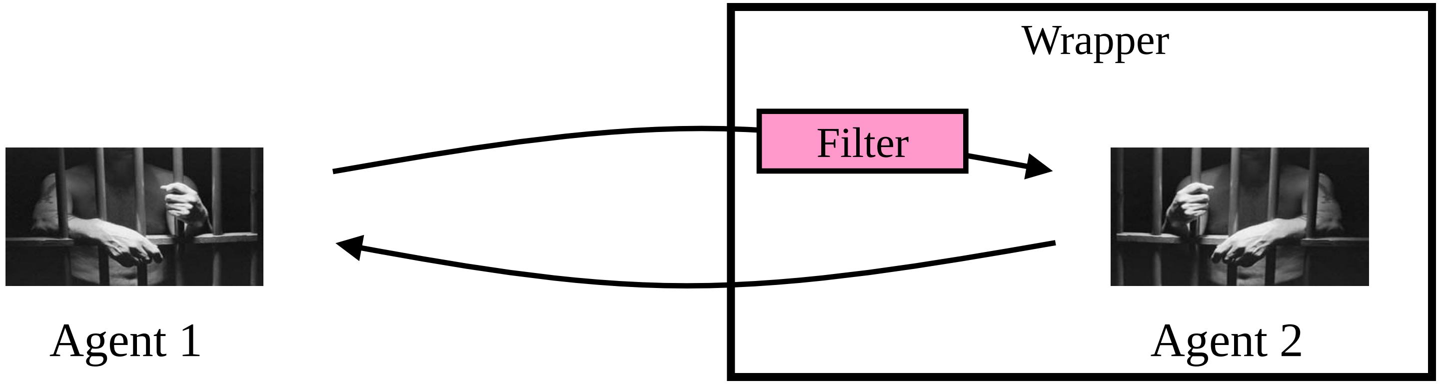
- Noisy Environments
  - » Actions can be altered by noise
  - » Noise can induce changes of behavior
- Important question in maintaining cooperation in noisy environment
  - » Is an anomaly of observed behavior accidental or intentional?
- Our solution: symbolic noise detection
  - » Use deterministic behavior to detect noise
  - » Defer judgments until there is enough evidence
- DBS performed extremely well in the 20th-Anniversary IPD competition
- Observation: deterministic behavior is abundant in IPD
  - » That’s why symbolic noise detection is effective

Would SND be effective in Other Games?

- Clarity of behavior is observed in some non-zero-sum games such IPD
  - » But what about other games?
- Consider zero-sum games such as chess and RoShamBo (i.e., Rock, Paper, Scissors)
  - » There is no intention to play deterministically or predictably.
- Questions:
  1. In what type of games deterministic behavior can be abundant?
  2. Would SND be effective in those games?
- We believe:
  - » SND will be most effective in games in which strategies are likely to exhibit deterministic behavior.

SND-based Wrappers

- To study this question, we proposed a **wrapper** that can be put around any strategy
  - » Examine the input using symbolic noise detection
  - » Modify the input as follows
    - When we don't think there's any noise, make no change
    - When we think we've detected noise, replace it with what we think the other player actually did
  - » Output: whatever action that the strategy chooses for this modified input

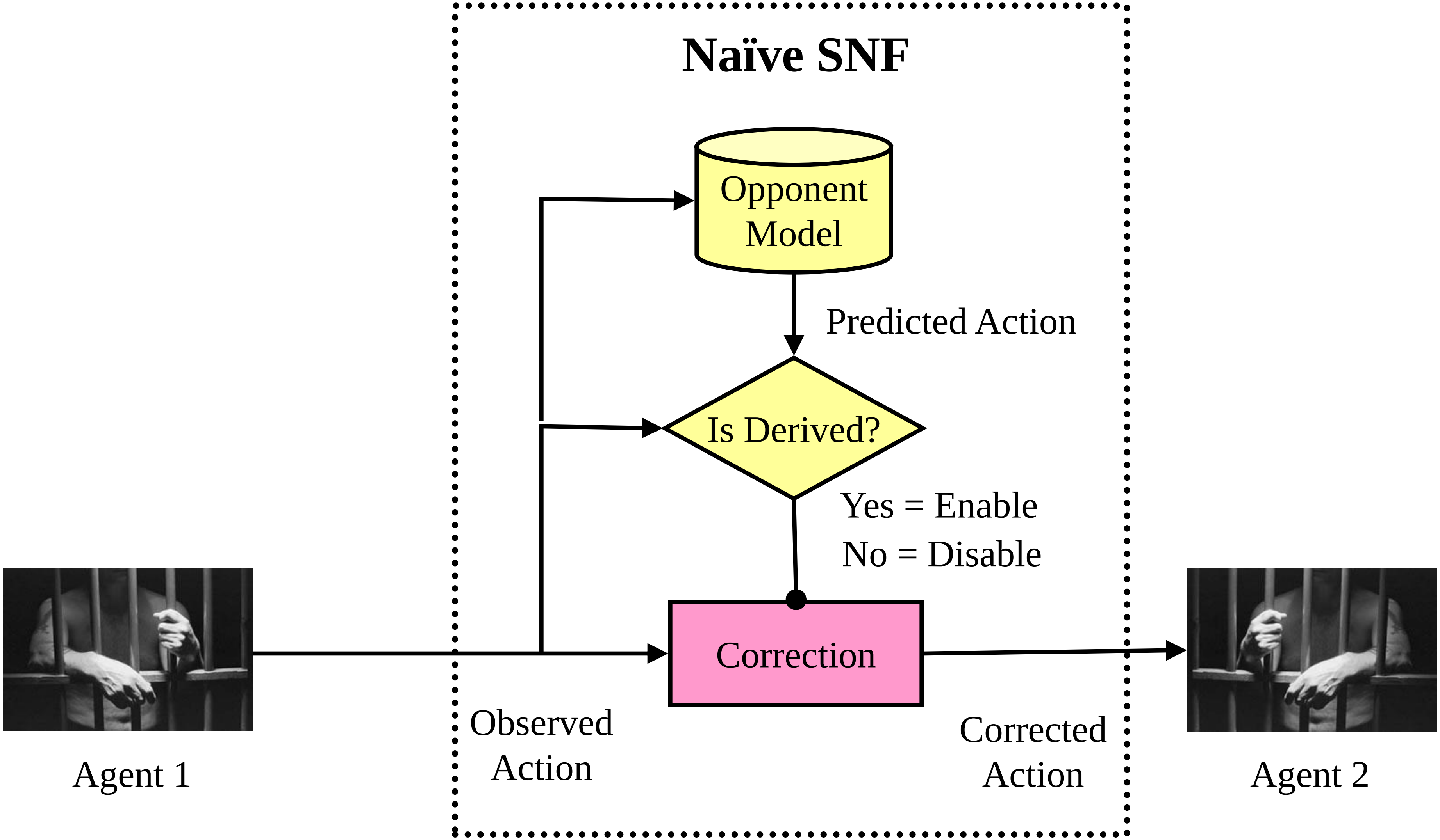


- Advantages of the wrapper approach:
  - » can place around *any* existing strategy
  - » enable us to compare different strategies

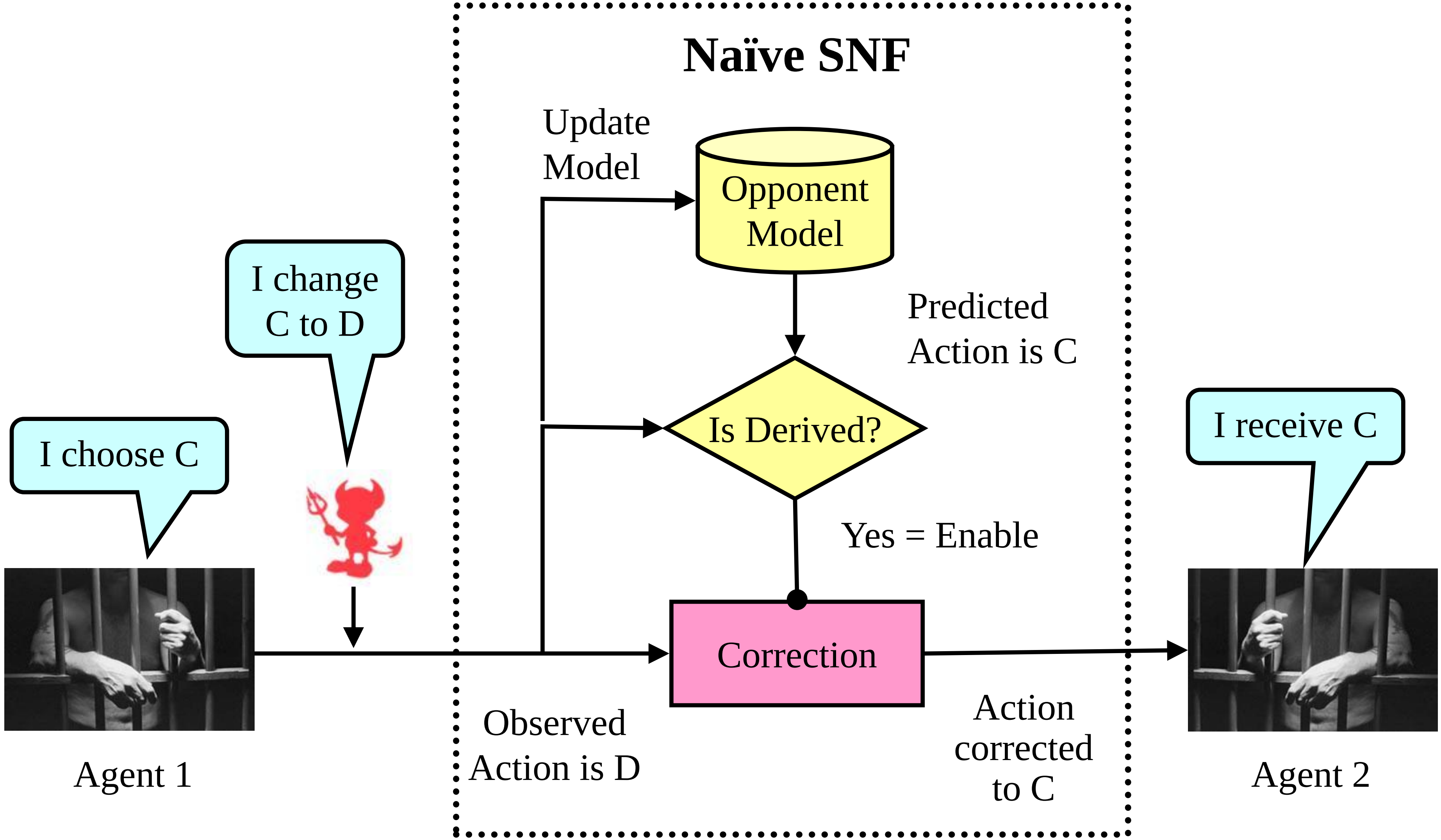
Naïve Symbolic Noise Filter

- Naïve Symbolic Noise Filter (NSNF)
  - A simple wrapper that can be easily implemented
  - Based on SND, but doesn't defer judgment about whether an anomaly is due to noise or to a change of behavior
- NSNF has only one parameter: the promotion threshold T
  - In each iteration,
    - If a behavior has occurred $\geq$ T times without contradiction, assume it's a deterministic behavior and record it
  - At the first violation of a deterministic behavior
    1. Correct the action
       - assume the observed input (e.g, Defect) is due to noise
       - change Defect to Cooperate or Cooperate to Defect before the input is fed into the strategy
    2. Drop the deterministic behavior
       - assume the behavior has changed
       - should identify new behavior or re-identify the same behavior in future

The Skeleton of Naïve Symbolic Noise Filter



How does Naïve SNF work?

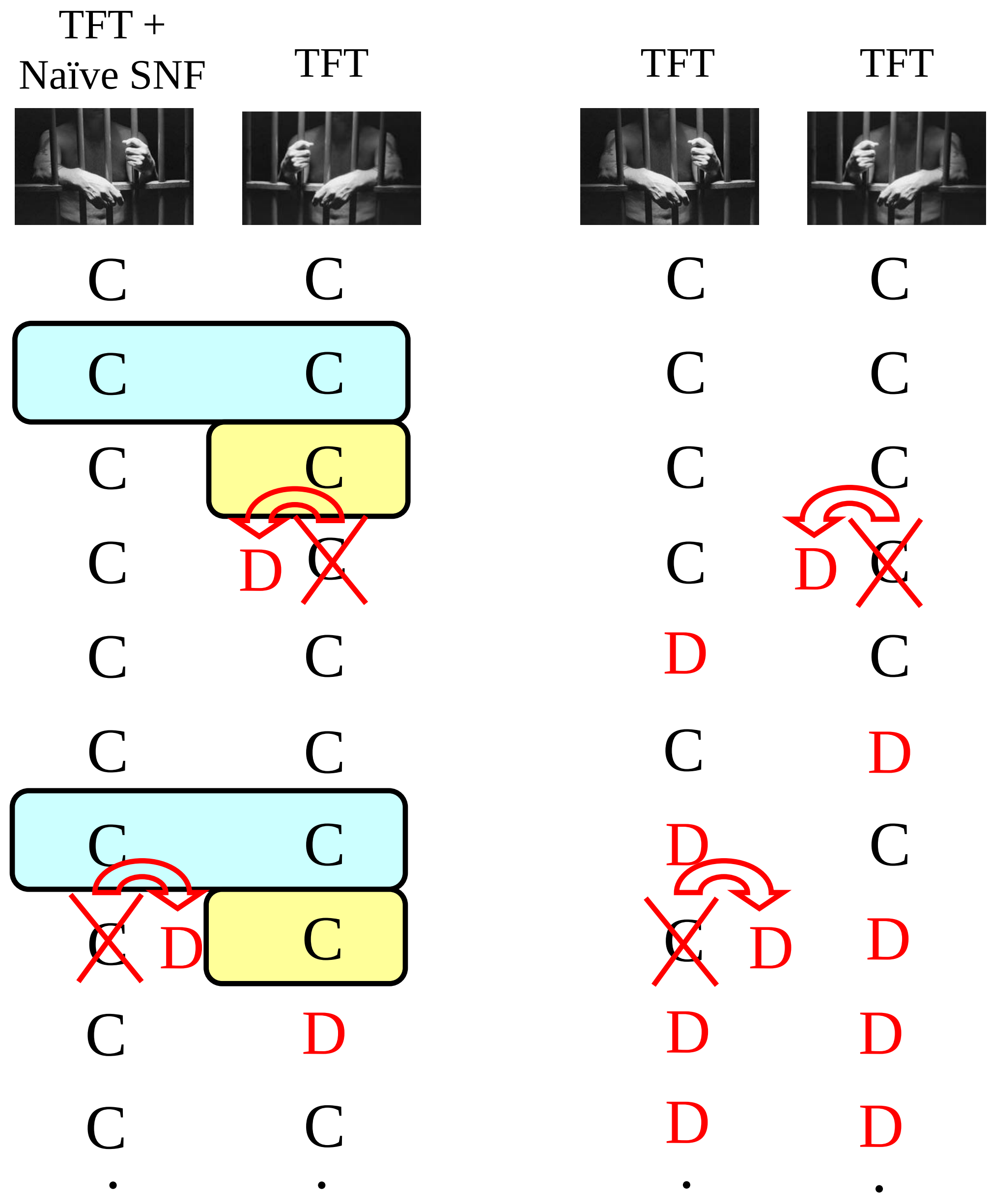


Naïve SNF vs Full-strength SND

- Naïve SNF does not defer judgment about whether a derivation (i.e., anomaly of behavior) is due to noise.
  - » All derivations from the opponent model, once observed, are considered due to noise.
- In contrast, DBS uses deferred judgment to increase the accuracy of noise detection
  - » If the opponent's behavior before and after a derivation are the same, the derivation is due to noise.
- The difference:
  - » Naïve SNF utilizes information before a derivation only
  - » *Full-strength SND* utilizes information before and after a derivation
- Full-strength SND is more accurate than Naïve SNF

Why Naïve SNF, not Full-Strength SND?

- Simple implementation is available
  - » no need to maintain the opponent model
    - Just look at the history to identify noise
  - » no need to “rewind” the state of the underlying strategy
    - suitable to our wrapper approach
- As an example:
  - » promotion threshold = 2



Chicken Game

- To evaluate Naïve SNF, we use two games as the test beds
  - » Iterated Chicken Game
  - » Iterated Battles of the Sexes
- Chicken Game:
  - » Two players compete for a resource
  - » If neither player concedes, both of them will suffer
- Example
  - » Two groups need to divide a piece of land between them
  - » If they can't agree upon how to divide it, they'll fight
- Nash equilibrium strategies (with no iteration):
  - » Do the opposite of what the other player is doing
- **Iterated** Chicken Game (ICG)
  - » Mutual cooperation does not emerge
    - Each player wants to establish him/herself as the defector

Payoff matrix:

Player$_2$ / *Player*$_1$	Cooperate	Defect
Cooperate	4, 4	3, 5
Defect	5, 3	0, 0

Battle of the Sexes

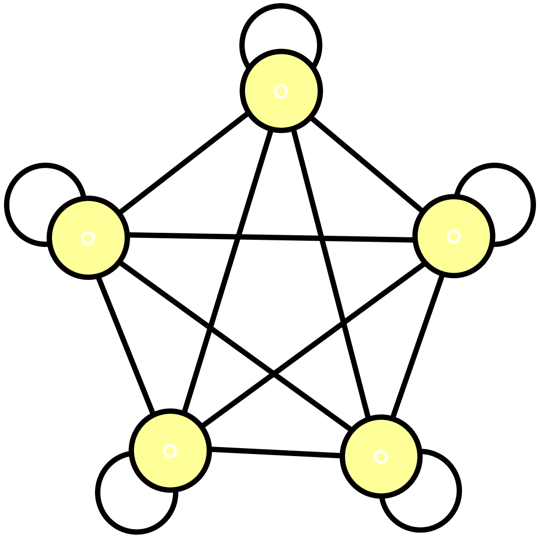
- Battle of the Sexes:
  - » Two players need to coordinate their actions to achieve some goal
  - » They each favor different actions
- Example
  - » Two nations must act together to win against a common enemy
  - » Each of them wants the other nation to make the biggest effort
- Nash equilibrium strategies (with no iteration):
  - » Do what the other player is doing
- **Iterated** Battle of the Sexes (IBS)
  - » Mutual cooperation does not emerge
  - » Each player wants to take advantage of each other

Payoff matrix:

Husband / *Wife*	Opera	Football
Opera	2, 1	0, 0
Football	0, 0	1, 2

Evaluation: Noisy Tournaments

- We organized two tournaments:
  - » The Noisy ICG tournament and The Noisy IBS tournament
- Agents:
  - » 37 agents for each tournament
    - written by 37 students in an advanced-level AI class
- What students knows:
  - » Payoff matrices
  - » Number of iterations in each game ≥ 50
  - » Noise level is 10%
  - » Robin-robin tournament
    - Every agent will play 1000 games with every other agents (including a copy of itself)
  - » The "internal memory" of the agents is "reset" at the beginning of every game



Experiments

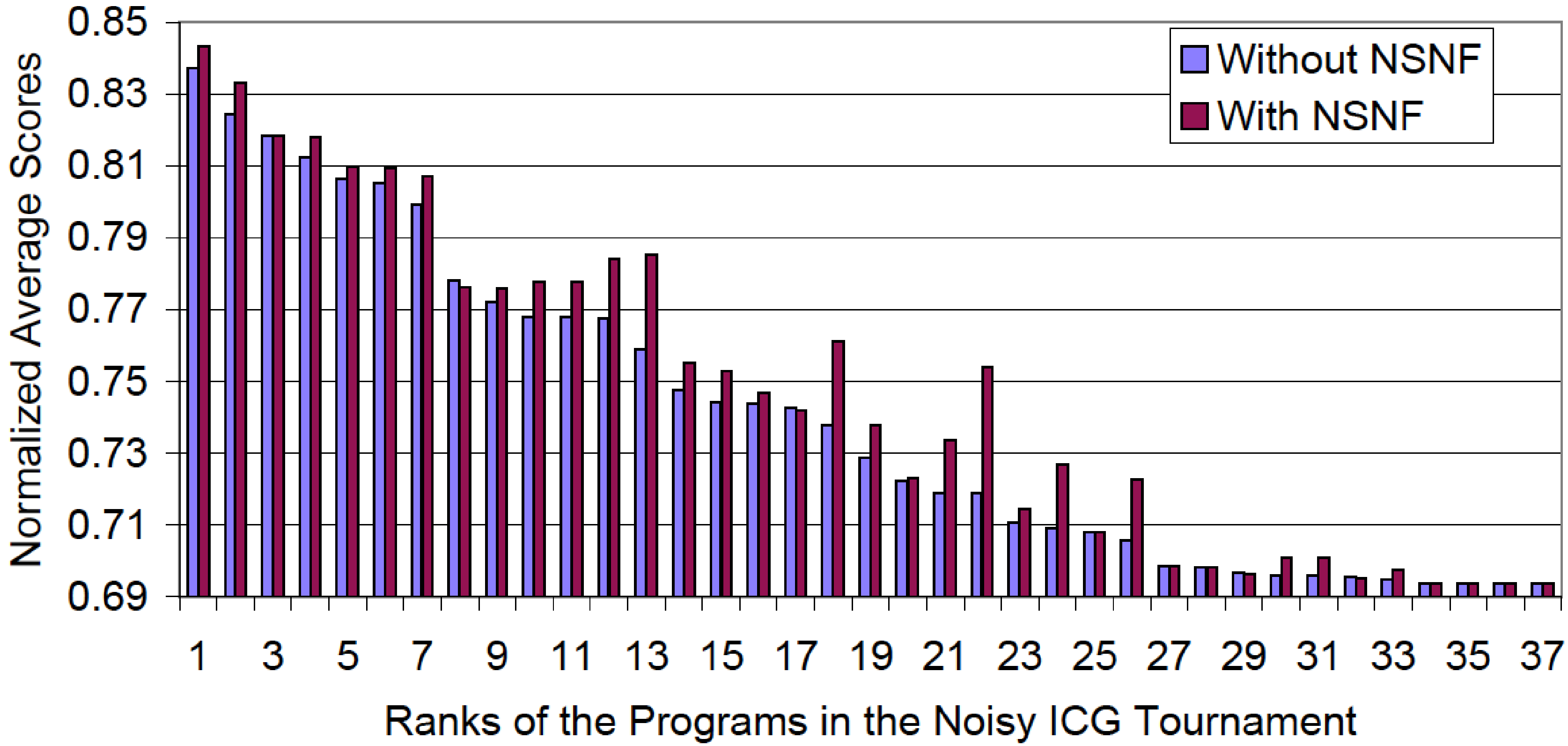
- For each program *P*, we ran two sets of tournaments
  - *P* against the others
  - *An enhanced version of P* against the others
    - The enhancement: use Naïve SNF to detect noise and correct action affected by noise.

Accuracy of Naïve SNF

- Naïve SNF is highly accurate in predicting a player's next move
  - 96% in ICG, 93% in IBS
- When NSNF makes a correction, how accurate is the correction?
  - **True positive**: corrected action was affected by noise
  - **False positive**: corrected action wasn't affected by noise
- At 300 iterations with noise level of 10%,
  - noise affects about 30 actions/game
- In ICG, about 25.4 corrections/game
  - 71% were true positives
- In IBS, about 27.1 corrections/game
  - 59% were true positives
- Baseline: random correction
  - % of true positives = 10 %
  - So Naïve SNF is much better than random correction

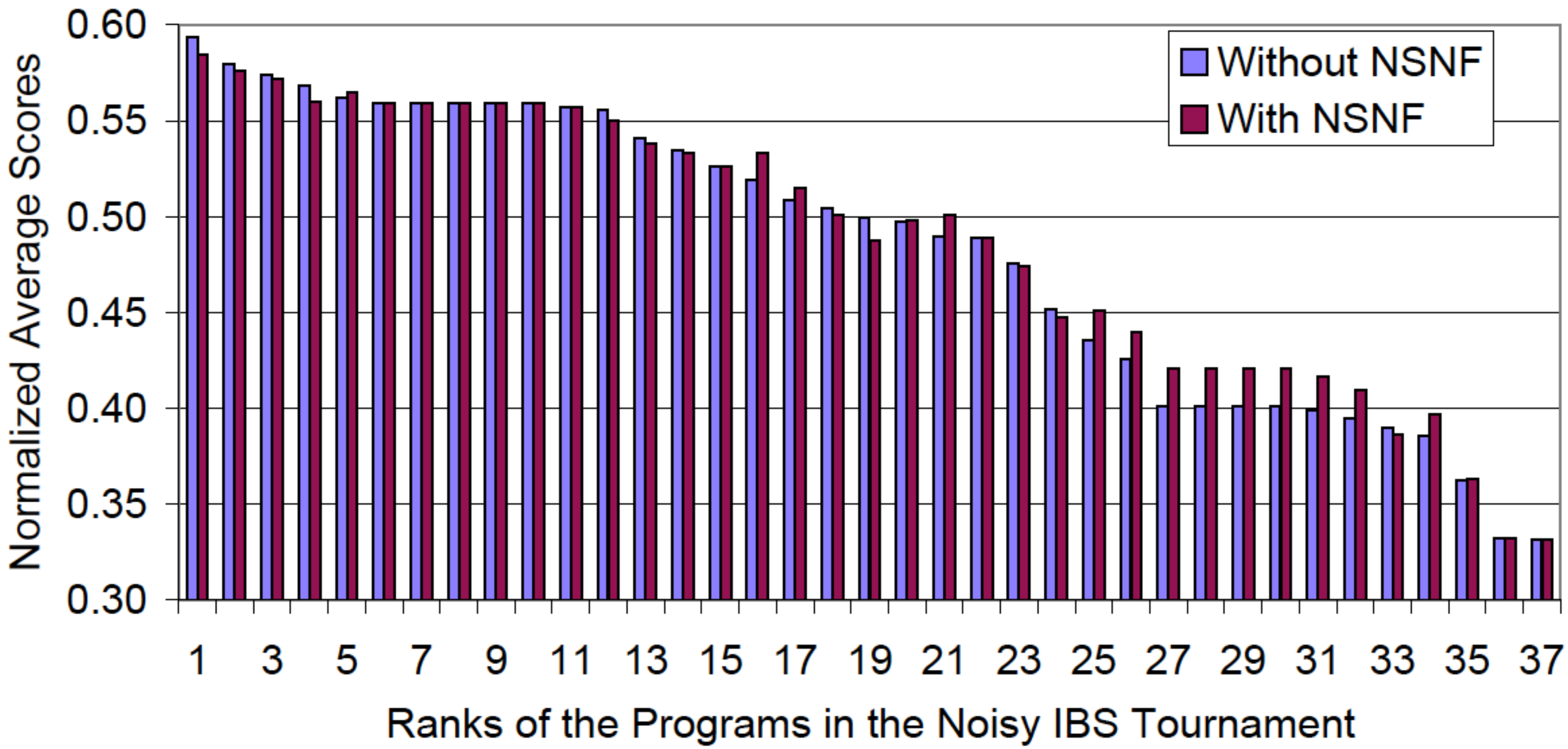
	ICG	IBS
Accuracy of predictions	96%	93%
Number of corrections	25.4 out of 300 moves	27.1 out of 300 moves
True pos.	71%	59%
False pos.	29%	41%

Naïve SNF in the Iterated Chicken Game



- On average:
  - » Naïve SNF improved each strategy's score by 0.9% (or 0.0068 in normalized score)
  - » Naïve SNF improved each strategy's ranking 1.51
- Difference in score of the best and worst strategies is only 0.144
  - » Thus a program's ranking can be changed by a small change in its score

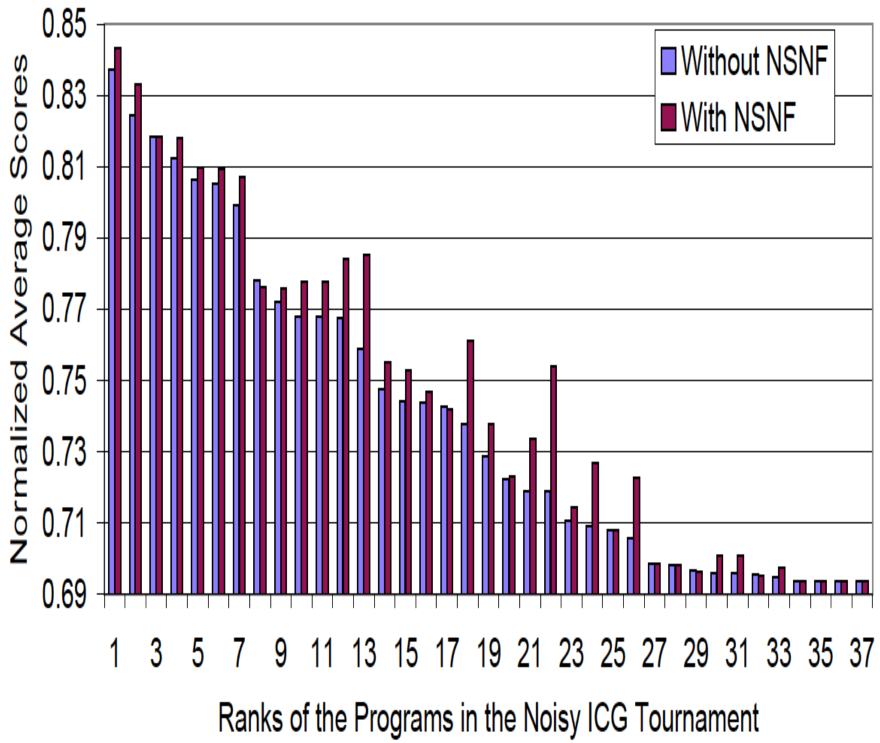
Naïve SNF in the Iterated Battle of the Sexes

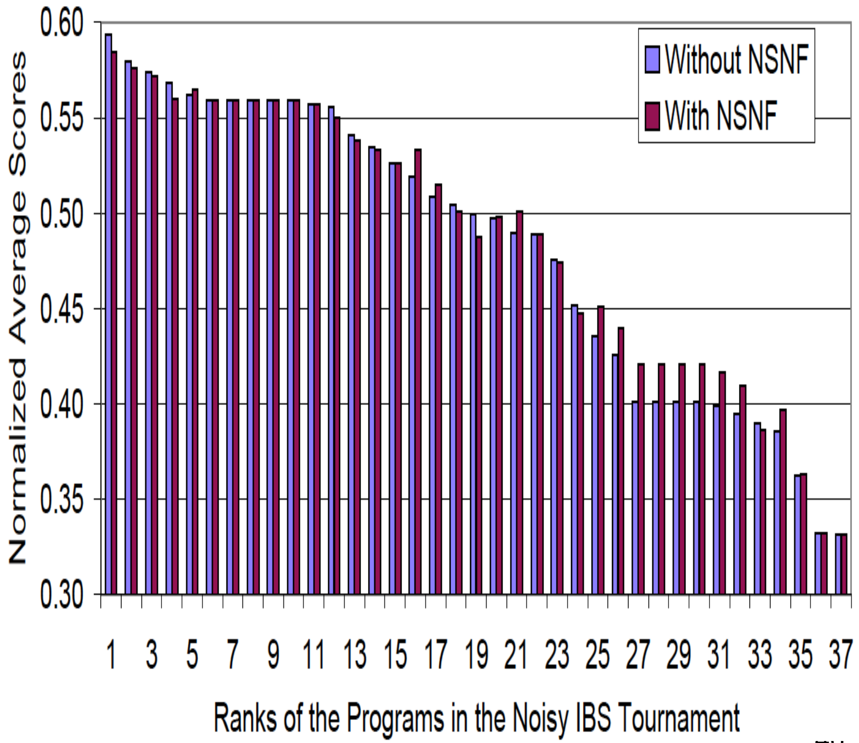


- On average, NSNF improved each strategy's score by 1.2% (or 0.0035 in normalized score)
  - » larger than ICG (0.9%), but doesn't change the rank as much
- Difference in score of the best and worst strategies is 0.262 (in ICG, it is 0.144)
  - » To change a program's ranking, need to change its score by about twice as much as in ICG

Naïve SNF on Strong, Average, and Weak Strategies

- But the benefit of SNF varies among different strategies
  - » In ICG, strong and average strategies benefit from Naïve SNF, but not weak strategies
  - » In IBS, average and weak strategies benefit from Naïve, but not the strong strategies
    - Some strategies perform even worse after using Naïve SNF

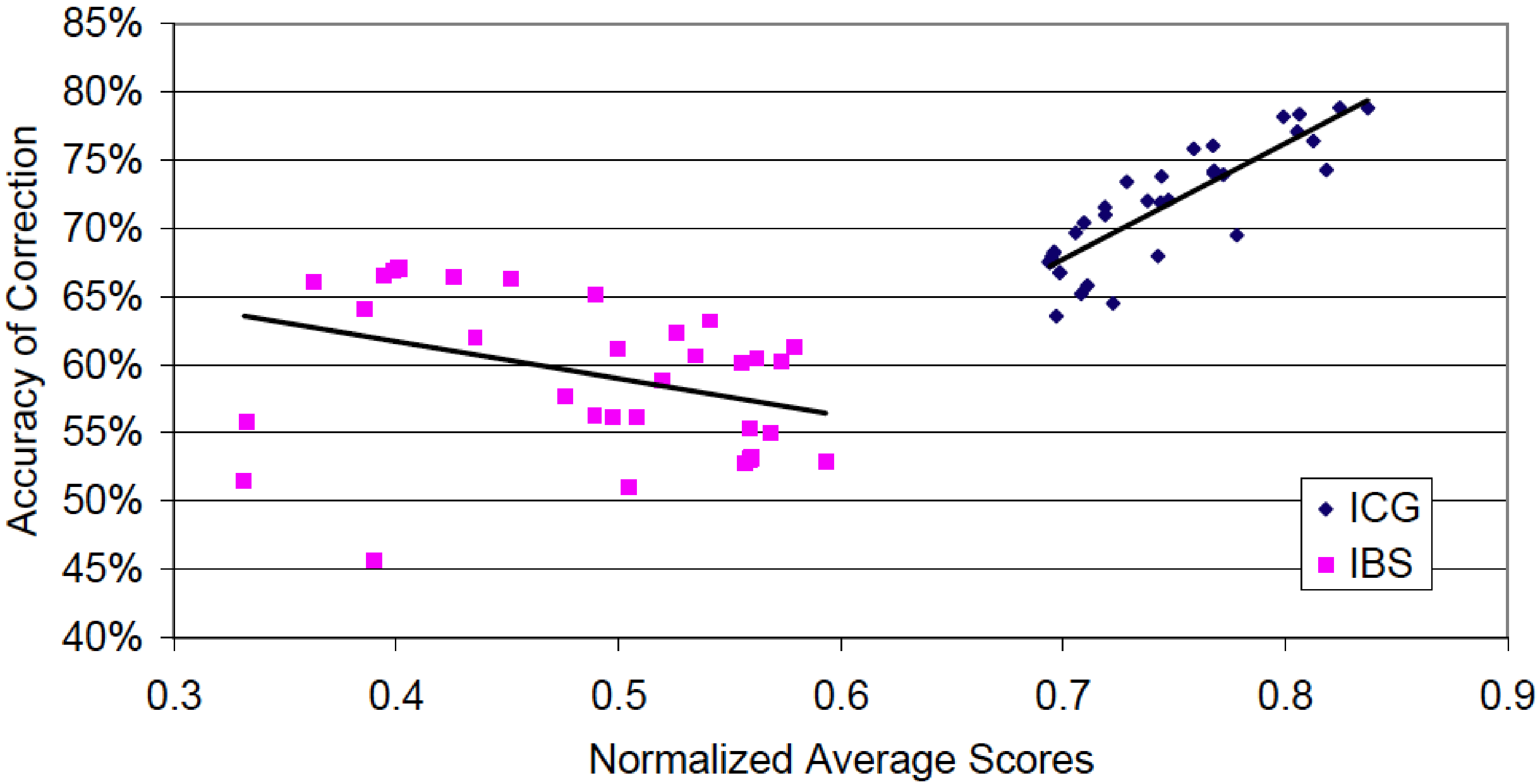




Three Key Variables and Their Relationships

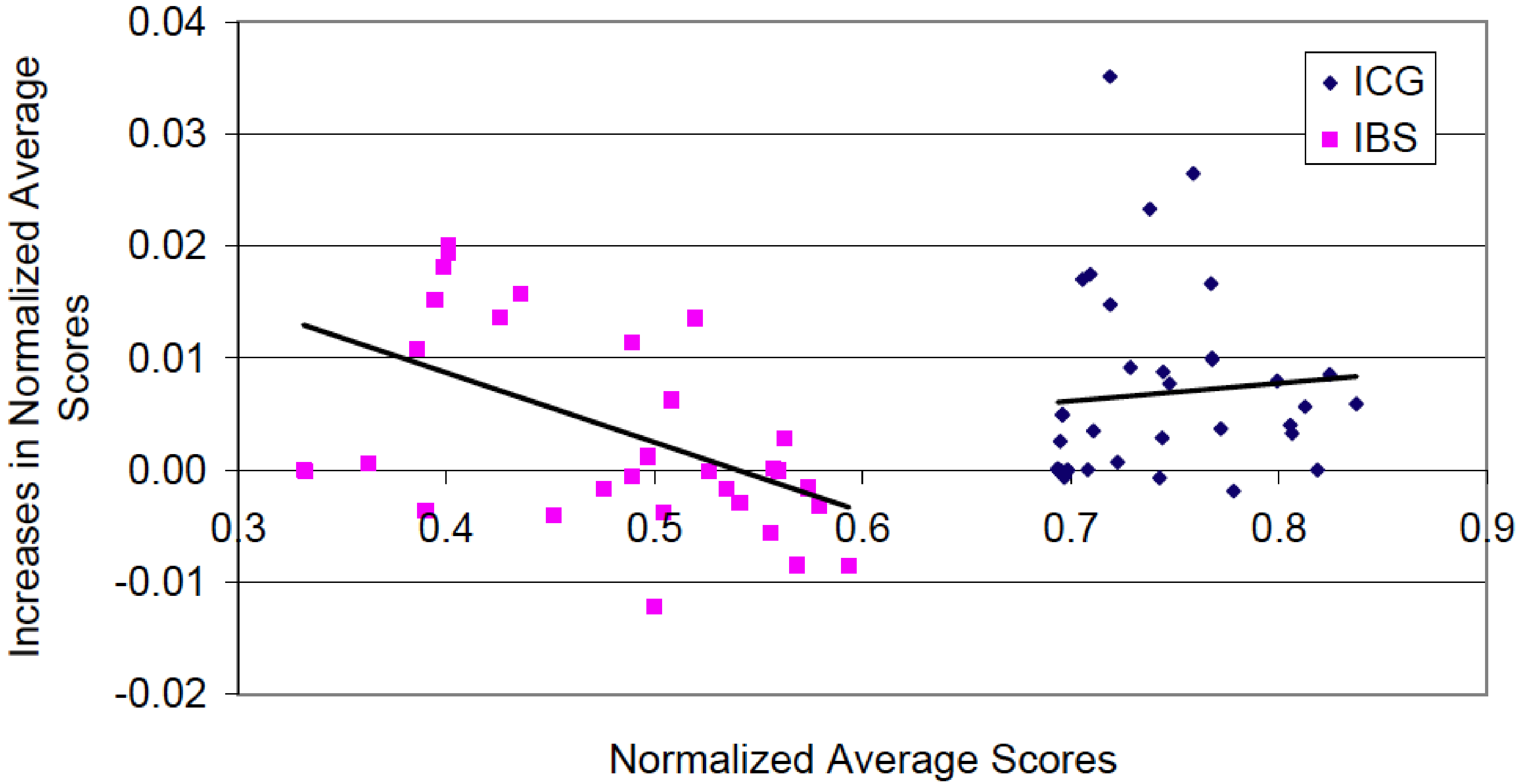
- To explain the difference among strong, average, and weak strategies, we identify three key variables and compare them
  - » Average score (X)
    - normalized to the range [0,1]
  - » Accuracy of correction (Δ)
  - » Increase in average scores (γ)
- Objective: identify the factors that would affect the effectiveness of Naïve SNF.
  - » Under what conditions would Naïve SNF be effective?
  - » The explanations can help us to determine when we should use Naïve SNF, and when we should not use Naïve SNF

Average Scores X vs Accuracy of Correction γ



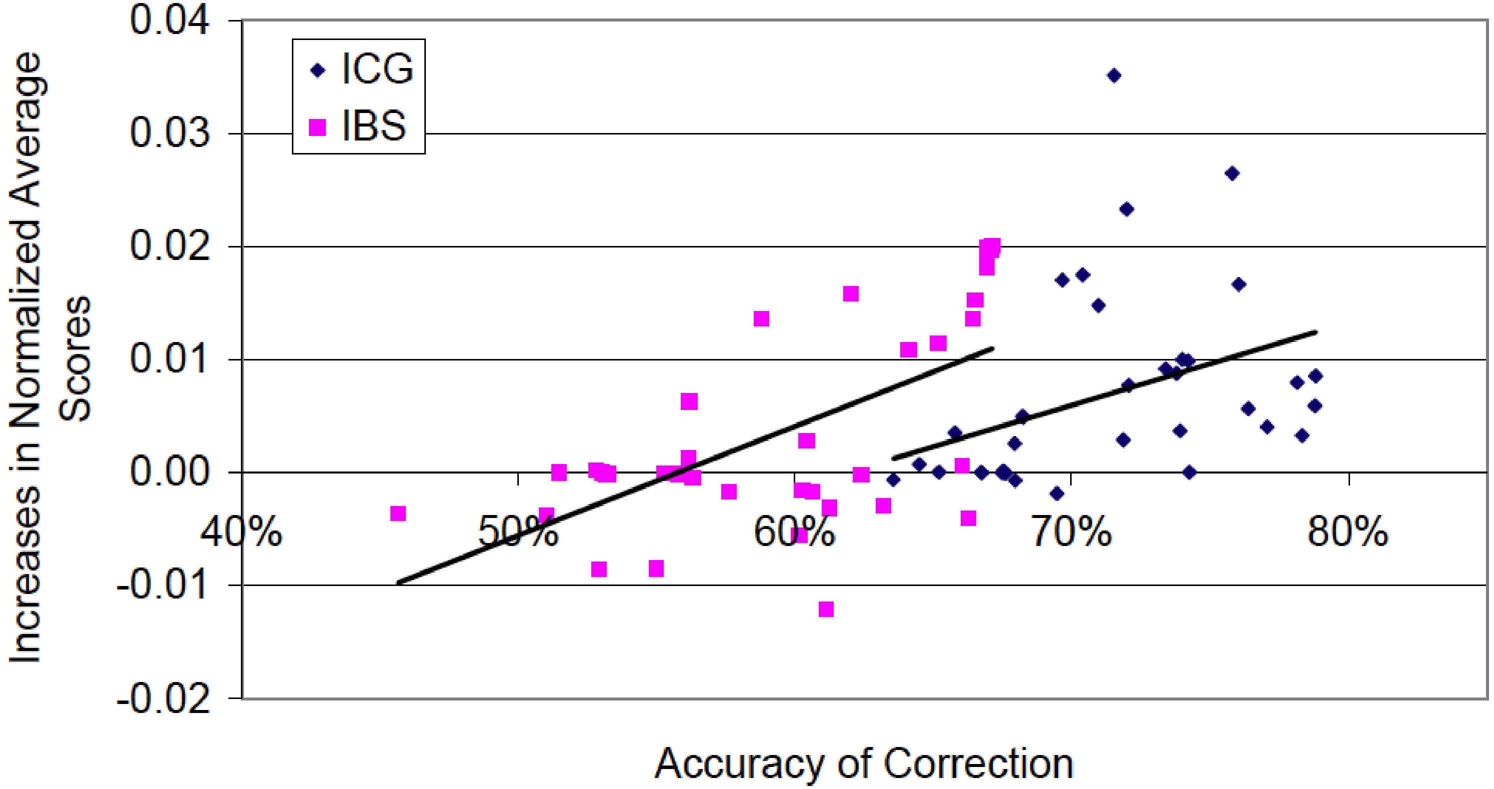
- ICG: strong positive linear correlation
- IBS: weak negative linear correlation (with outliners)
- Correction Accuracy in ICG > Correction Accuracy in IBS

Average Scores X vs Increases in Avg. Scores Δ



- ICG: No obvious correlation; but most strategies have an increase in average scores after using Naïve SNF
- IBS: weak negative linear correlation (with outliners); about half of the strategies have an decrease in average scores after using Naïve SNF.

Accuracy of Correction γ vs Increase in Avg. Scores Δ

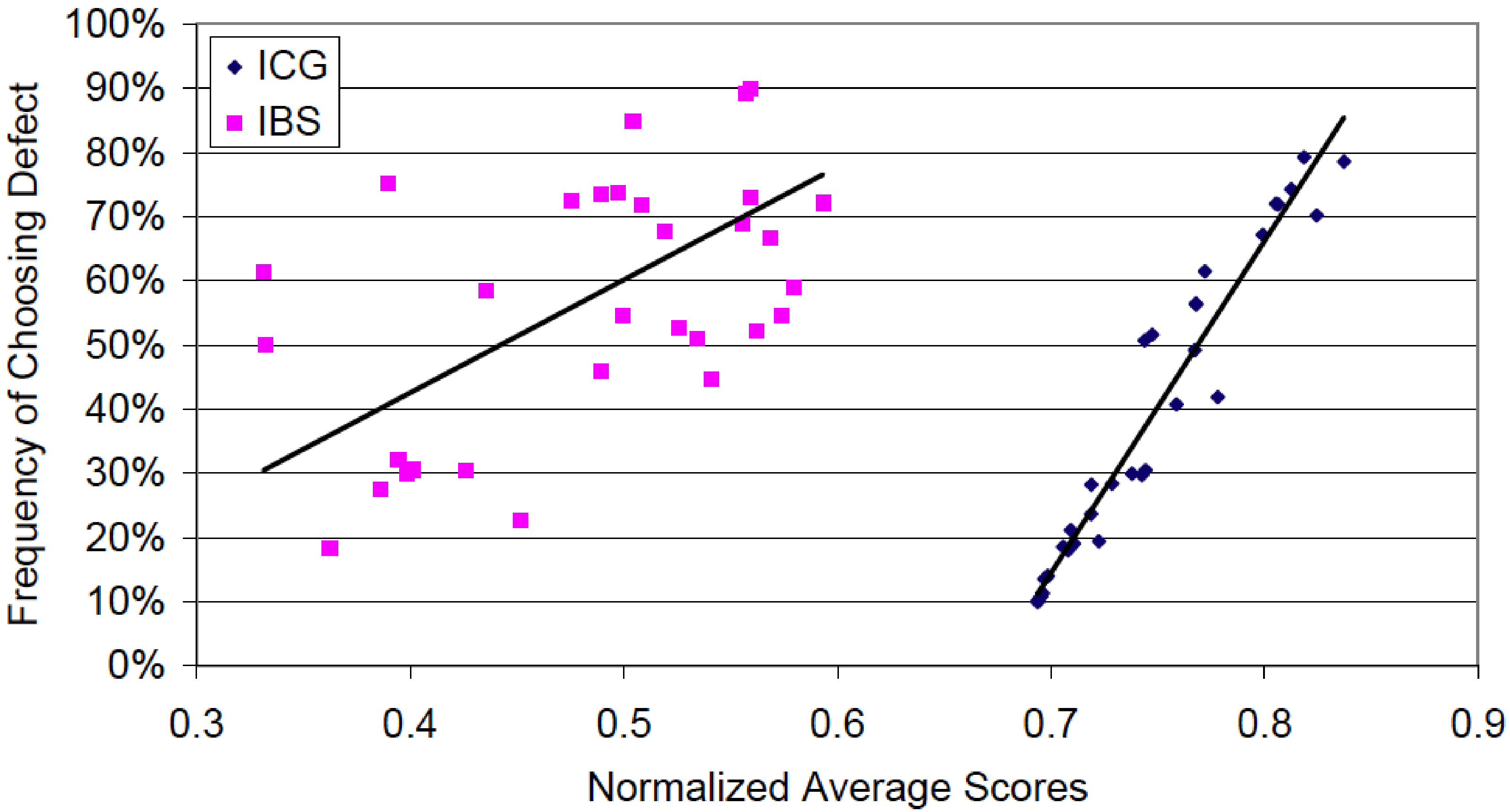


- ICG and IBS: no obvious correlation
- Low correction accuracy → decrease or no increase in average score

Experimental Explanation

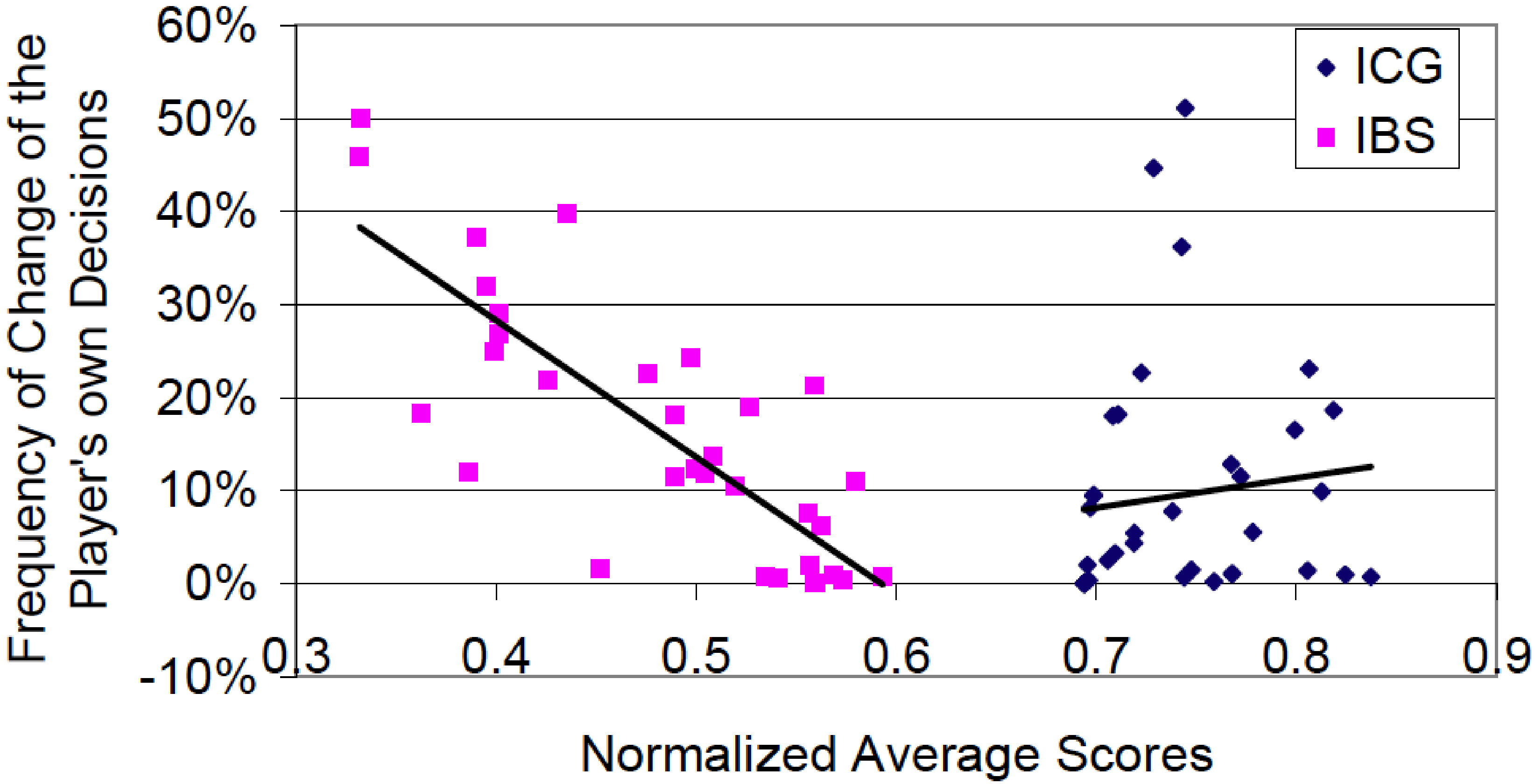
- We explain the relationships among the variables experimentally, via two variables concerning the decisions made by players
  - » The frequency of choosing defect (D)
  - » The frequency of “flip-flop”--- changing decisions (from cooperate to defect or vice versa) between two consecutive iterations. (F)
- We will explain the relationships between
  - » average scores and accuracy of correction
  - » average scores and increases in average scores
- We can deduce the relationship between the accuracy of correction and increases in average scores

Average Scores X vs Frequency of Defection D



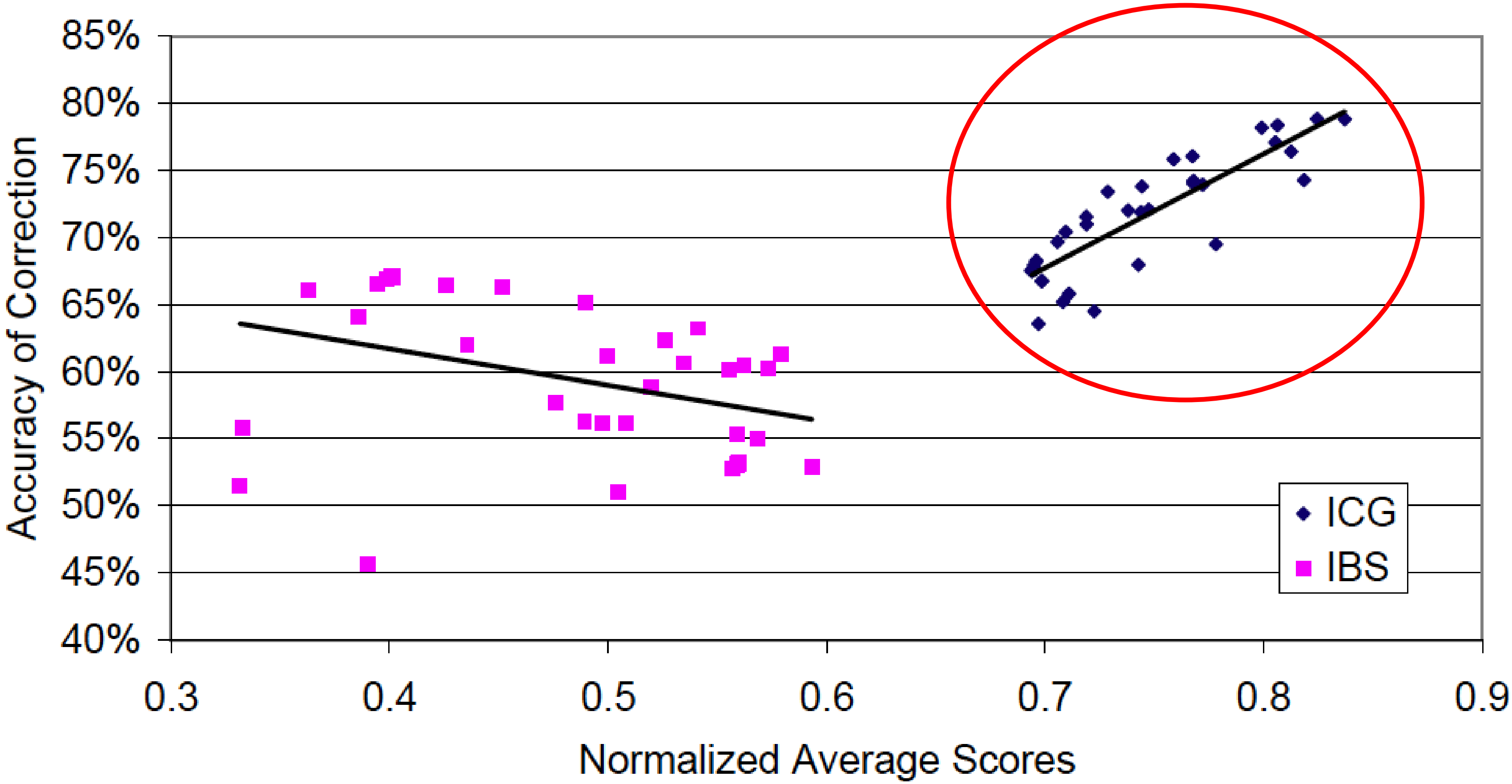
- ICG: Strong positive linear correlation (i.e., top strategies often defects)
- IBS: no obvious correlation

Average Scores X vs Freq. of Flip-Flop F



- ICG: no obvious correlation
- IBS: strong negative linear correlation (with outliners) (i.e., top strategies does not change decisions frequently)

Average Scores vs Accuracy of Correction



- First, let us explain why there is a strong positive linear correlation in ICG
  - Top players have higher accuracy of correction

Average Scores vs Accuracy of Correction

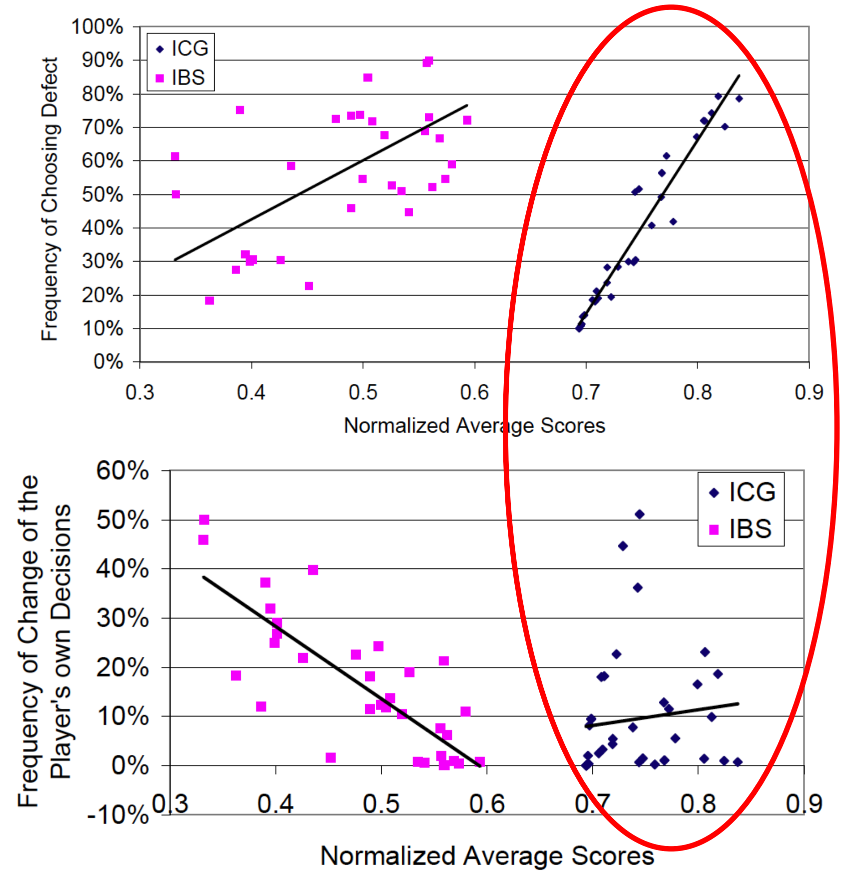
- In ICG
  - » strong players often defect
  - » weak players often cooperate
  - » average players often flip-flop (changing cooperate to defect, or vice versa)
- When facing strong players, it is better to cooperate to earn 3 points instead of defect to earn 0 point
  - » strong players force the average players to cooperate

    → average players often exhibit a clear deterministic behavior

    → High correction accuracy
- When facing weak players who cooperates, there is little difference between cooperate and defect (4 points ≅ 5 points)
  - » Weak players fails to force the average players to cooperate

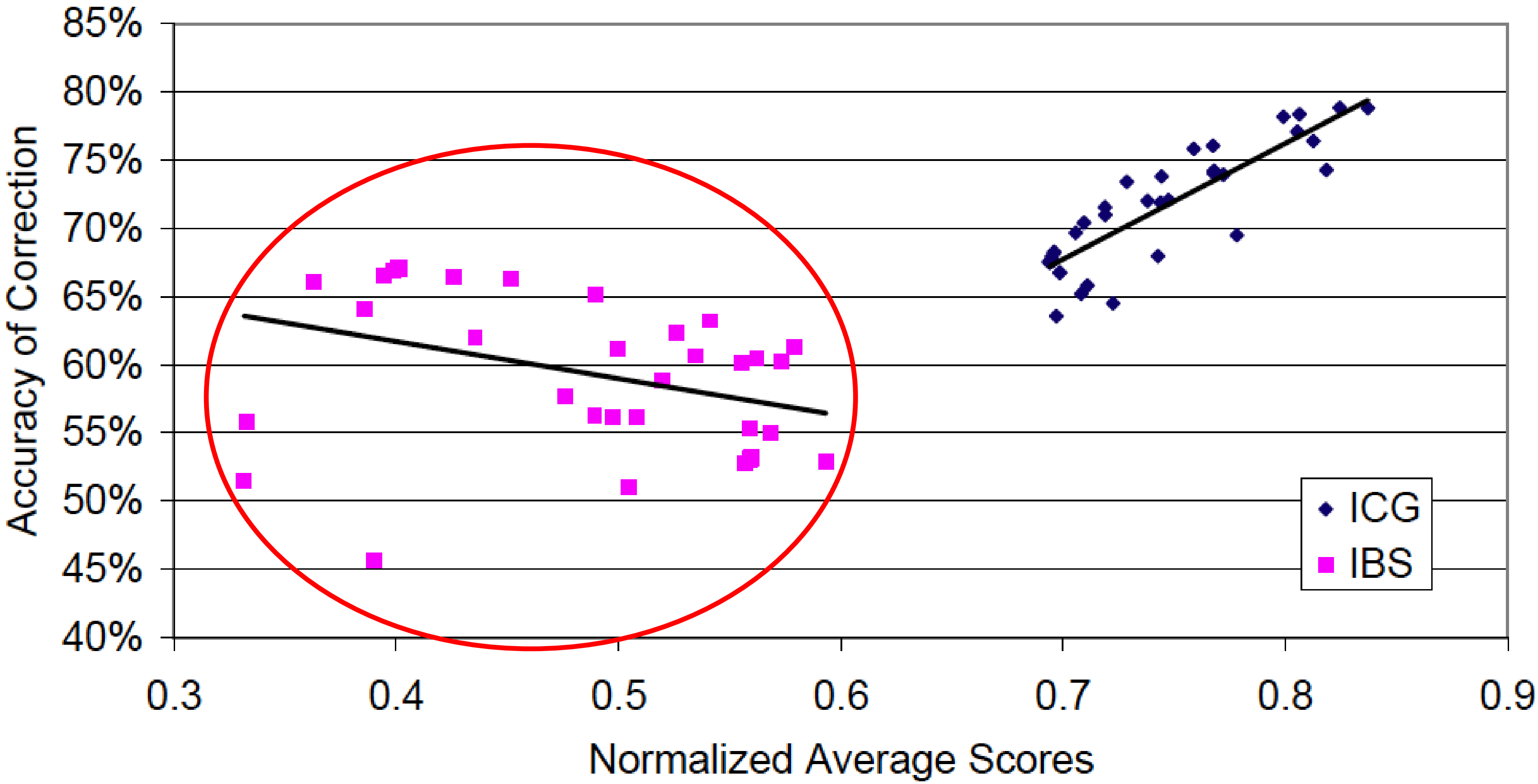
    → Average players continue to flip-flop

    → Low correction accuracy



$Player_2$ \ $Player_1$	Cooperate	Defect
Cooperate	4, 4	3, 5
Defect	5, 3	0, 0

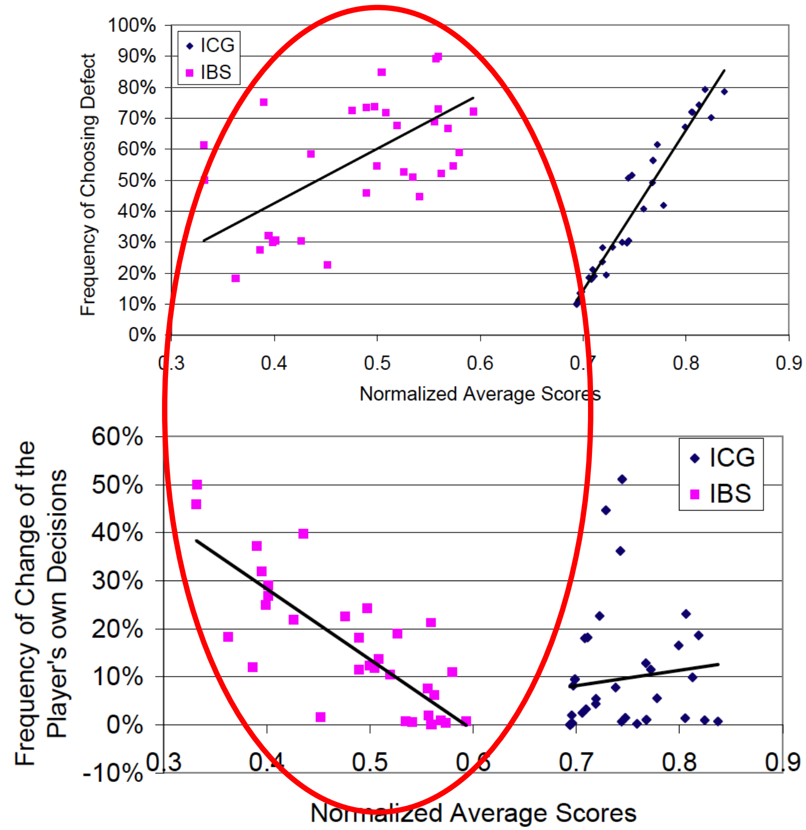
Average Scores vs Accuracy of Correction



- Second, we explain why there is a weak negative linear correlation in IBS

Average Scores vs Accuracy of Correction

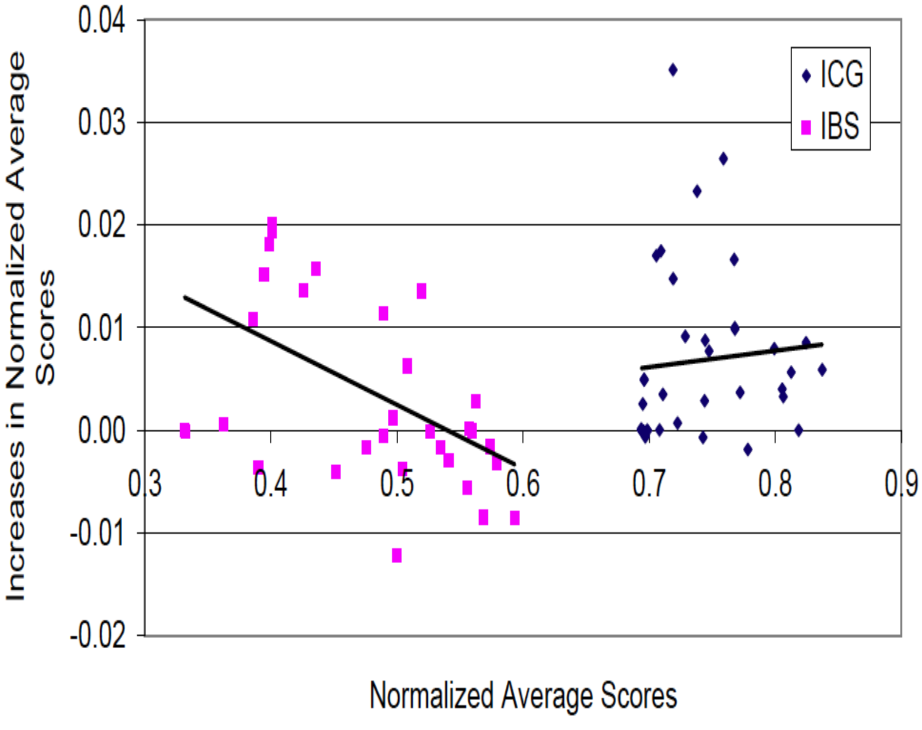
- In IBS
  - » strong players often defect
  - » weak players often flip-flop
- When facing strong players, there is little difference between cooperate and defect (1 points ≅ 0 points)
  - » Strong players less successfully in forcing other players to cooperate
    - → Other Players exhibit less clear deterministic behavior
    - → Lower correction accuracy
- When facing weak players who flip-flop, it is better to defect than to cooperate in long run
  - » Weak players causes other players to defect more often
    - → Higher correction accuracy

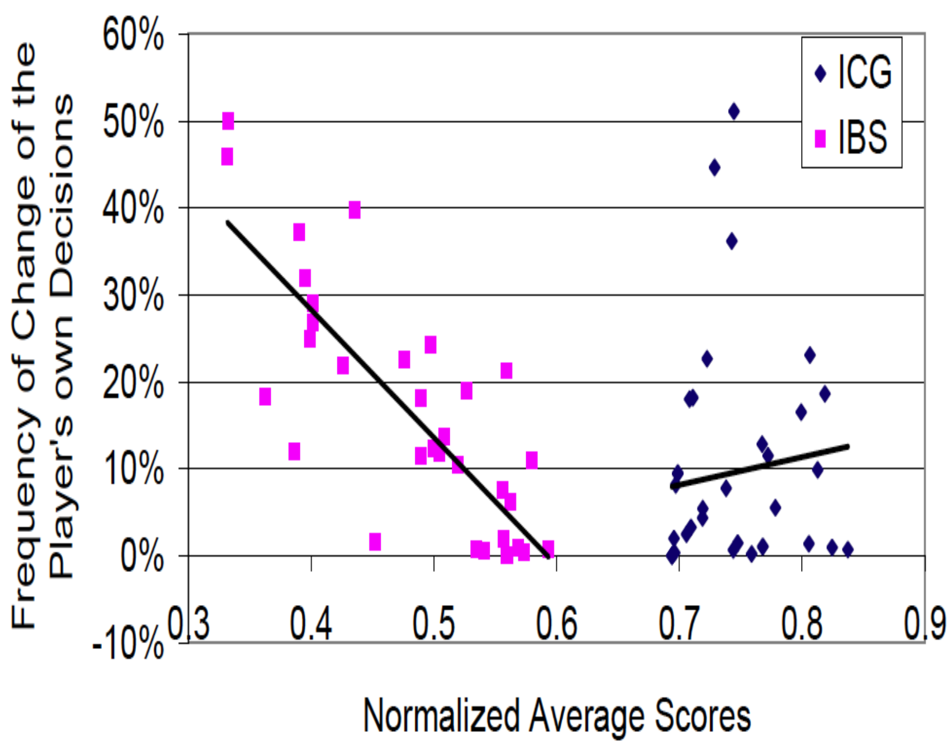


$Player_2$ / $Player_1$	Cooperate	Defect
Cooperate	0, 0	1, 2
Defect	2, 1	0, 0

Average Scores X vs Increases in Avg. Scores Δ

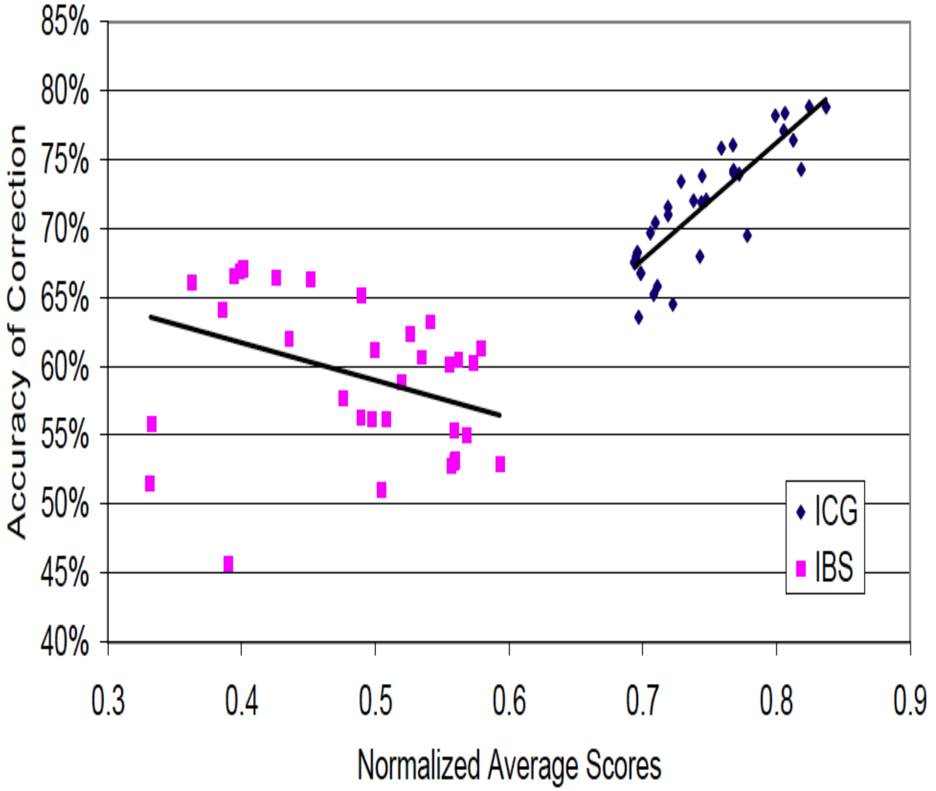
- Third, we explain the relationship between the increase in average score and the average scores





- Very similar!
- The larger the amount of flip-flops, the higher the increases in average scores after using Naïve SNF

Average Scores X vs Increases in Avg. Scores γ



- The increases in average scores seems to be less dependent to the accuracy of correction.

Observation and Prediction

- What strategies do not get a better performance when using Naïve SNF?
  - » Strategies which are insensitive to the other player’s moves
    - e.g., almost always choose defects no matter what.
- What strategies can potentially get a performance boost when using Naïve SNF?
  - » Strategies which are sensitive to the other player’s moves.
    - e.g, TFT
    - e.g., ICG strategies which often change their decisions (flip-flops).
- The increase of average scores seems to be less dependent of the accuracy of correction.
  - » But there are still some effects

- For strategies that works with Naïve SNF, our prediction:

$$\text{Change of average scores} = K \times (\alpha - \beta) \times (\gamma - 50\%)$$

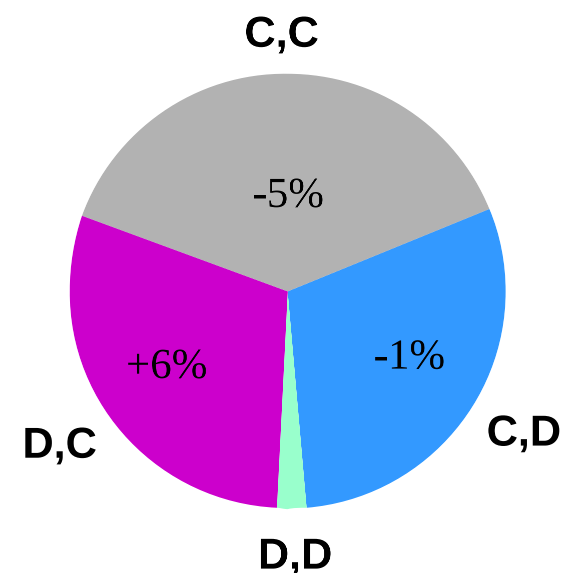
Where

- α = average score in noise-free environment
- β = average score in noisy environment
- γ = accuracy of correction
- K is a constant

Other Results: Distribution of Decision Pairs

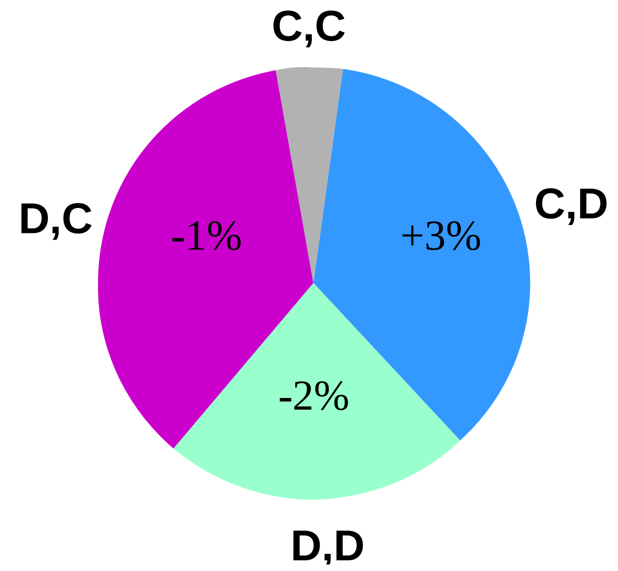
ICG:

$Player_2$ / $Player_1$	Cooperate	Defect
Cooperate	4, 4	3, 5
Defect	5, 3	0, 0



IBS:

$Player_2$ / $Player_1$	Cooperate	Defect
Cooperate	0, 0	1, 2
Defect	2, 1	0, 0



Summary

- Noisy Environments
  - » Actions can be altered by noise
  - » Noise can induce changes of behavior
- Important question in maintaining cooperation in noisy environments
  - » Is an anomaly in observed behavior accidental or intentional?
- Our solution: symbolic noise detection
  - » Use deterministic behavior to detect noise
  - » Defer judgments until there is enough evidence
- DBS performed extremely well in the 20th-Anniversary IPD competition

- We propose to use SND as a general technique for detection and correction of errors in interaction among agents in multi-agent systems
  - » Wrapper-based approach to equip existing strategies with symbolic noise detection
  - » Naïve Symbolic Noise Filter
    - can be put around any strategy for 2×2 normal-form games

Summary (cont.)

- Hypothesis: SND is effective in situations where strategies often exhibit deterministic behavior
  - » We have experimentally verified this hypothesis in
    - The Iterated Chicken Game (ICG)
    - The Iterated Battles of the Sexes (IBS)

- SND is effective *in noise detection* in ICG and IBS
  - » Accuracy of prediction: 96% in ICG, 93% in IBS
  - » Accuracy of correction: 71% in ICG, 59% in IBS
- Overall scores of all strategies increased in both games after using SND
  - » +0.0068 in ICG, +0.0035 in IBS
- Our hypothesis seems valid:
  - » SND is indeed effective in situations where strategies often exhibit deterministic behavior
    - But in ICG and IBS, this deterministic behavior is often a threat rather than an expression of cooperation

Summary (cont.)

- But if we look at strategies individually
  - » Most programs benefited from noise detection, but not all
  - » High accuracy in noise detection ≠ high score
  - » Some strategies, especially in IBS, did better with uncorrected inputs
- To see why, we conducts experiments to explain the relationships between
  - » Average scores
  - » Increases in average scores
  - » Accuracy of noise correction
- Conclusion:
  - » Strategies that are insensitive to the other player's moves do not get a better performance when using Naïve SNF
    - e.g., almost always choose defects no matter what (e.g., strong players in both ICG and IBS)
  - » Strategies that are sensitive to the other player's moves potentially get a performance boost when using Naïve SNF
    - e.g, TFT
    - e.g., ICG strategies which often change their decisions (flip-flops).
  - » The increase of average scores seems to be less dependent of the accuracy of correction.
  - » Change of average scores = $K \times (\alpha - \beta) \times (\gamma - 50\%)$

Future Work

- Noise can occur in nearly all kind of multi-agent systems in which agents has to interact as with each other.
  - » How to deal with noise is an important issue, especially in situations where the success of agents greatly depends on the robustness of interaction (i.e., IPD-like situations)
  - » Can SND be used to deal with errors in interaction in those real-world situations?
    - Need to modify SND to handle more actions, more agents, and more complicated situations.
- Open questions:
  - » Which type of games can benefit from using SND?
  - » Which type of programs can benefit from using SND?

Conclusion

Noise is less problematic when the incentive for cooperation is strong

Thank you!

Any Questions?