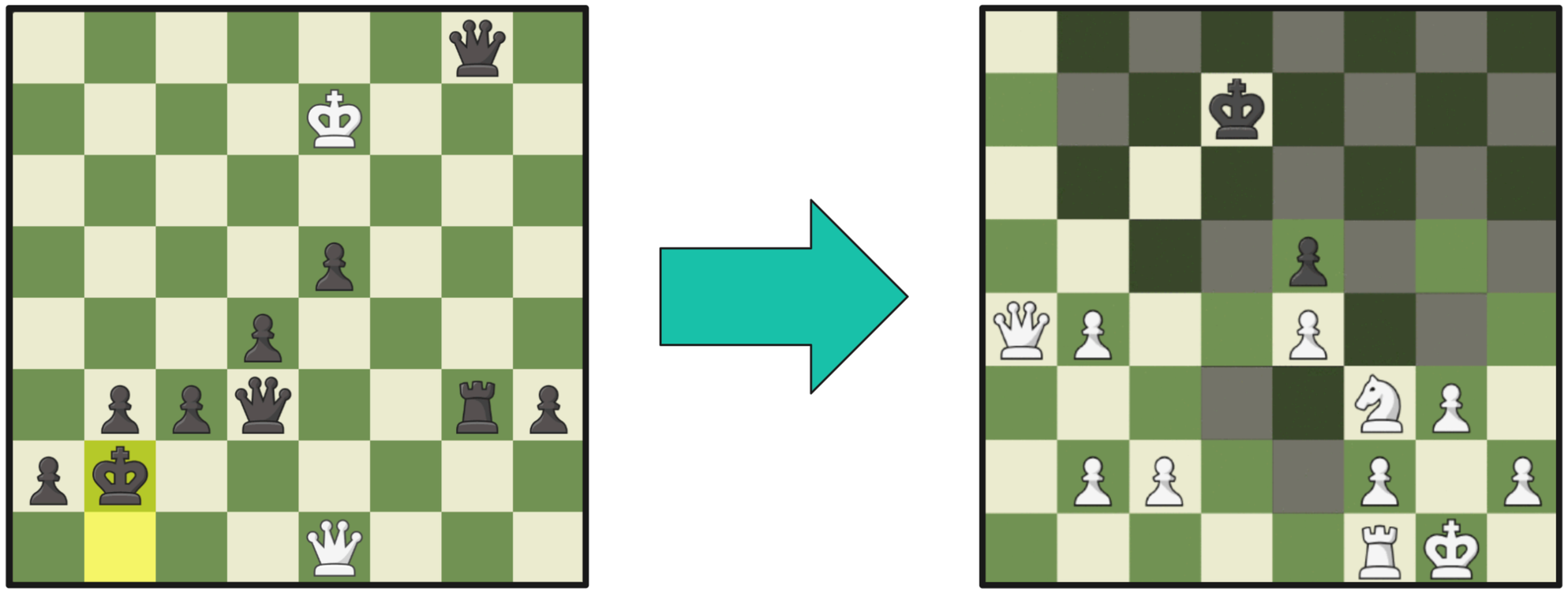
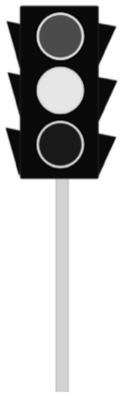
Going beyond Games with Perfect Information

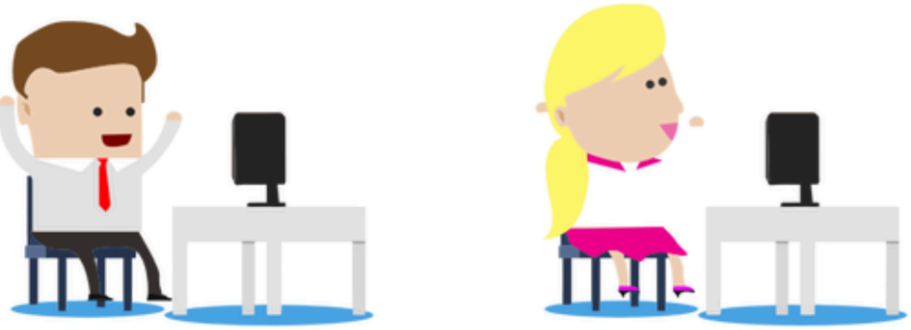
Seminar in Deep Neural Networks - 23.03.2021
Presented by Rafael Sterzinger

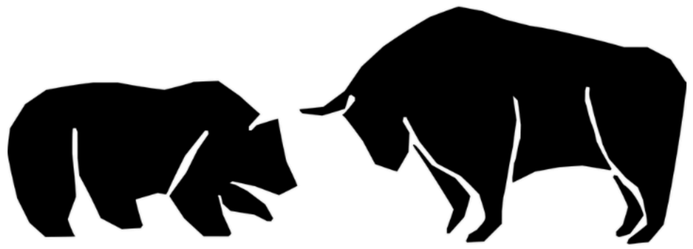
Perfect vs Imperfect Information Games



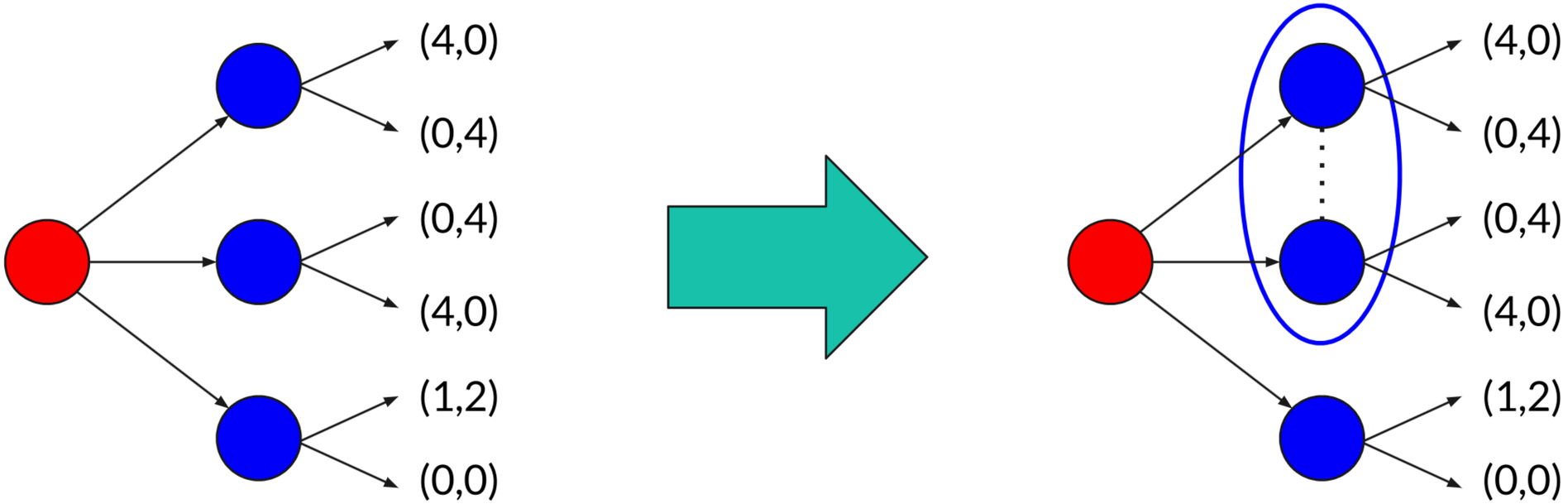
Why to study Games with Imperfect Information?



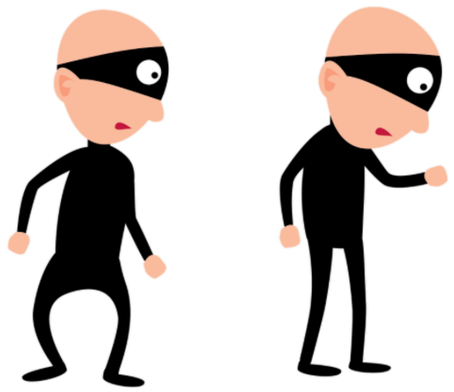




Example of Changing Strategies

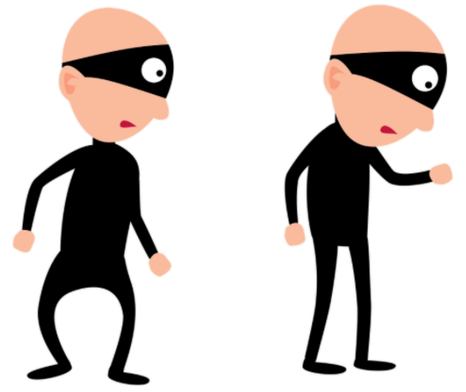


What is a Nash equilibrium?



"No player has anything to gain by changing only their own strategy"

What is a Nash equilibrium?



A \ B	B stays silent	B betrays
A stays silent	-1 / -1	0 / -3
A betrays	-3 / 0	-2 / -2

Why is seeking a Nash equilibrium the goal?

Seeking a Nash Equilibrium

?

?

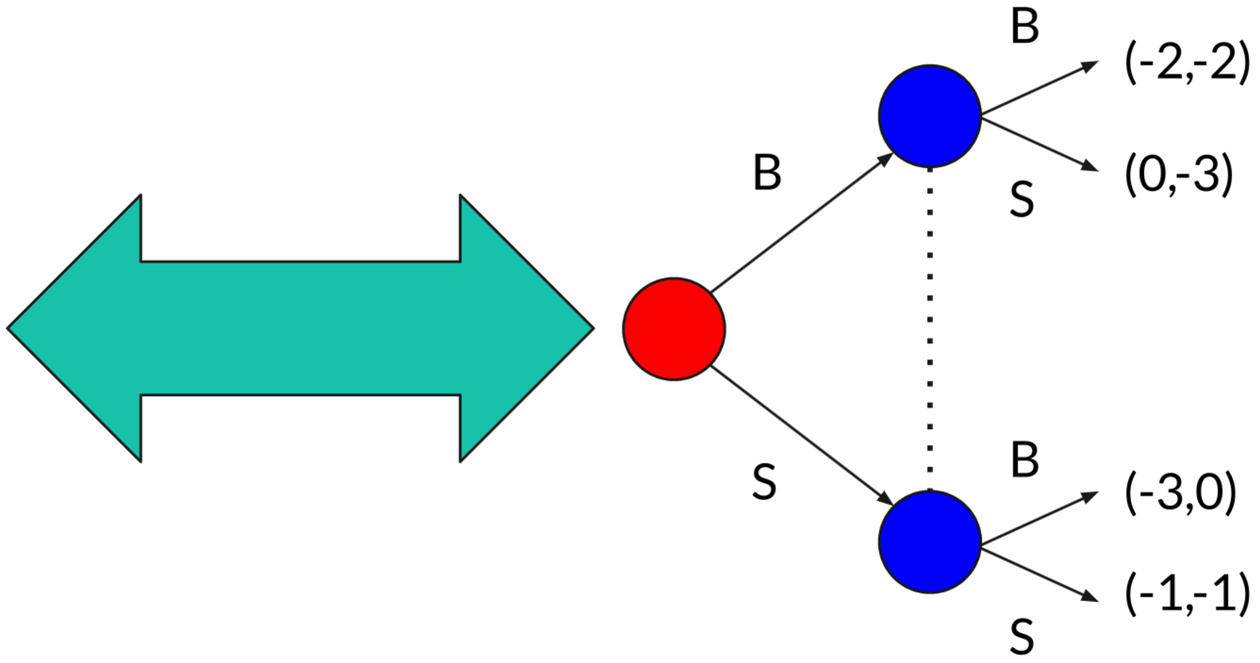
Study the opponent's strategy

Why is seeking a Nash equilibrium the goal?

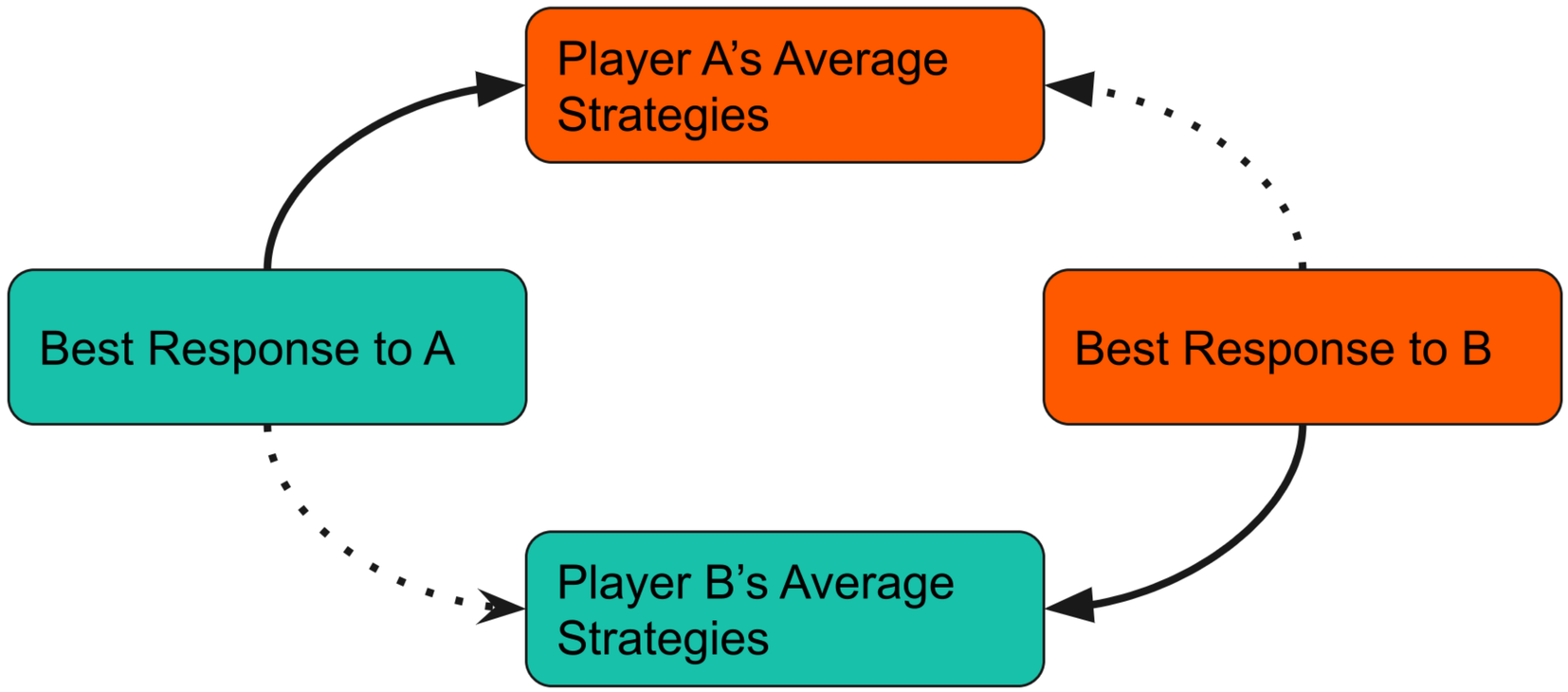
Seeking a Nash Equilibrium

Normal-Form Games vs Extensive-Form Games

A \ B	B stays silent	B betrays
A stays silent	-1, -1	0, -3
A betrays	-3, 0	-2, -2



Fictitious Play

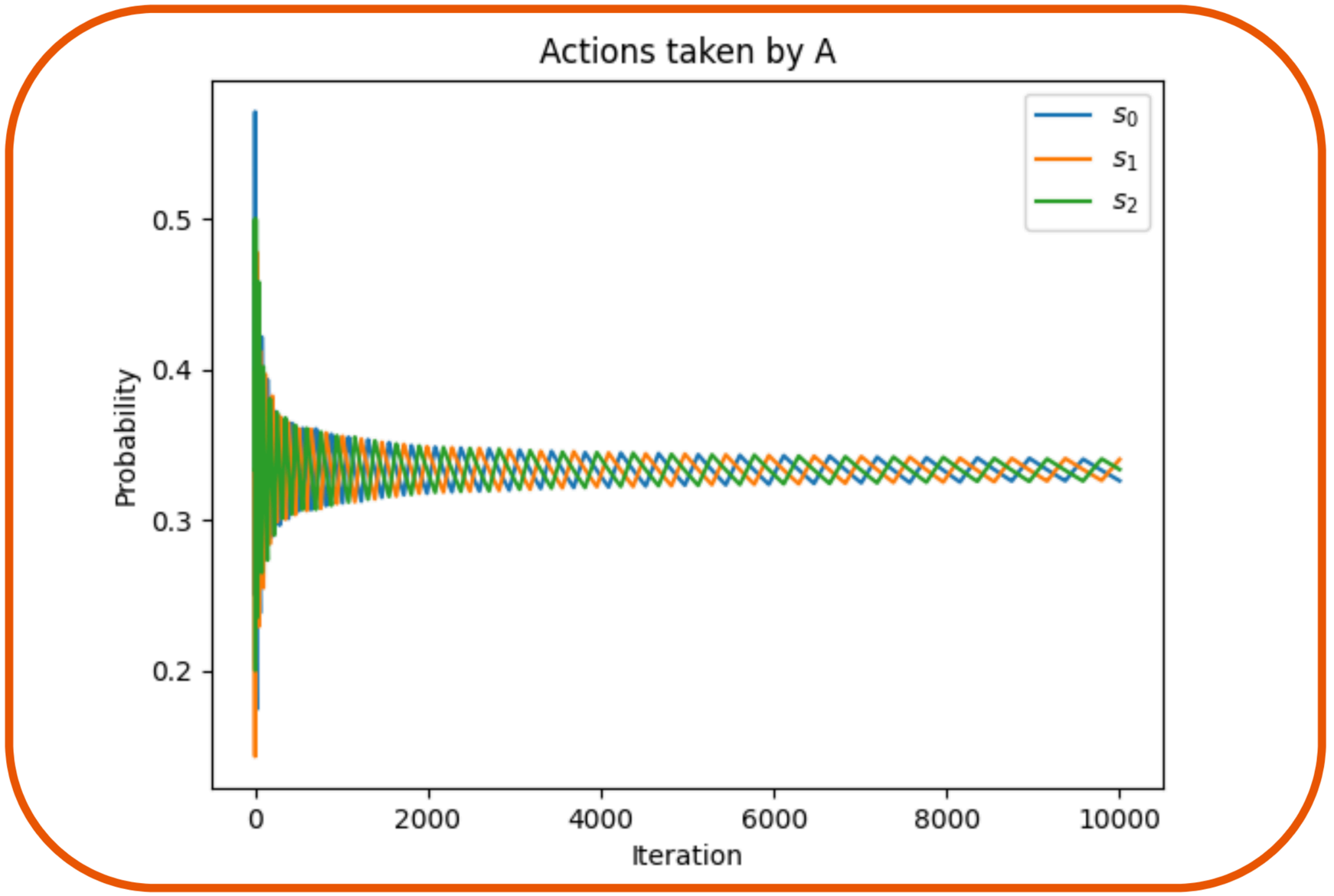


Fictitious Play

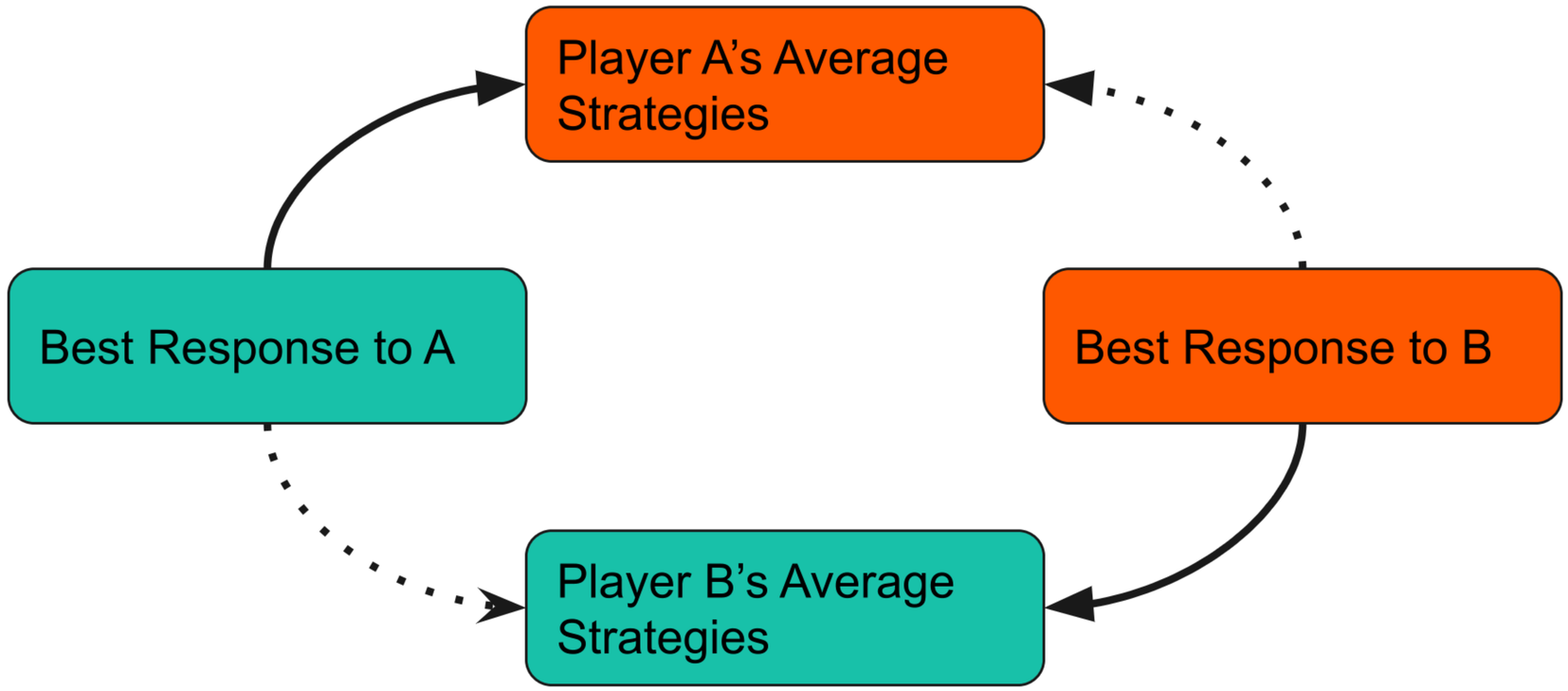
"The average strategy profile of fictitious players converges to a Nash equilibrium in certain classes of games, e.g. two-player zero-sum [...] games."

Example of Fictitious Play

Round	Player A's action	Player B's action	Player A's count	Player B's count
1	Rock	Paper	(1,0,0)	(0,1,0)
2	Scissors	Paper	(1,0,1)	(0,2,0)
3	Scissors	Rock	(1,0,2)	(1,2,0)
4	Paper	Rock	(1,1,2)	(2,2,0)


Actions taken by A
s_0
s_1
s_2
0.5
0.4
0.3
0.2
Probability
0
2000
4000
6000
8000
10000
Iteration

Problems of Fictitious Play



Problems of Fictitious Play

BR ComputeBRs(S)

Computation needs to be performed at all states

S UpdateAvgStrategies(S,BR)

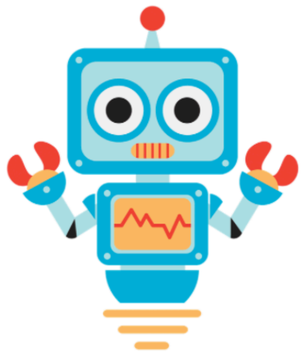
Agents are prevented from generalising between states

Proposed Solution: (Neural) Fictitious Self-Play

BR ComputeBRs(S)

Replace

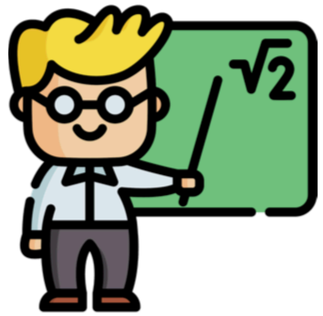
Reinforcement Learning



S UpdateAvgStrategies(S,BR)

Replace

Supervised Learning



Overview of NFSP

Best Response Network: $Q(s, a|\theta^Q)$

$(s_t, a_t, r_{t+1}, s_{t+1})$

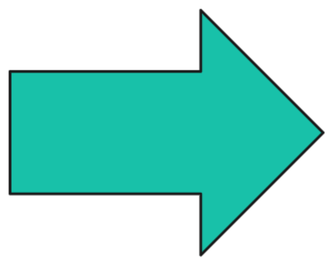
Average Strategy Network: $\Pi(s, a|\theta^\Pi)$

(s_t, a_t)

Algorithmic Changes for Sampling

BR ComputeBRs(S)

S UpdateAvgStrategies(S,BR)



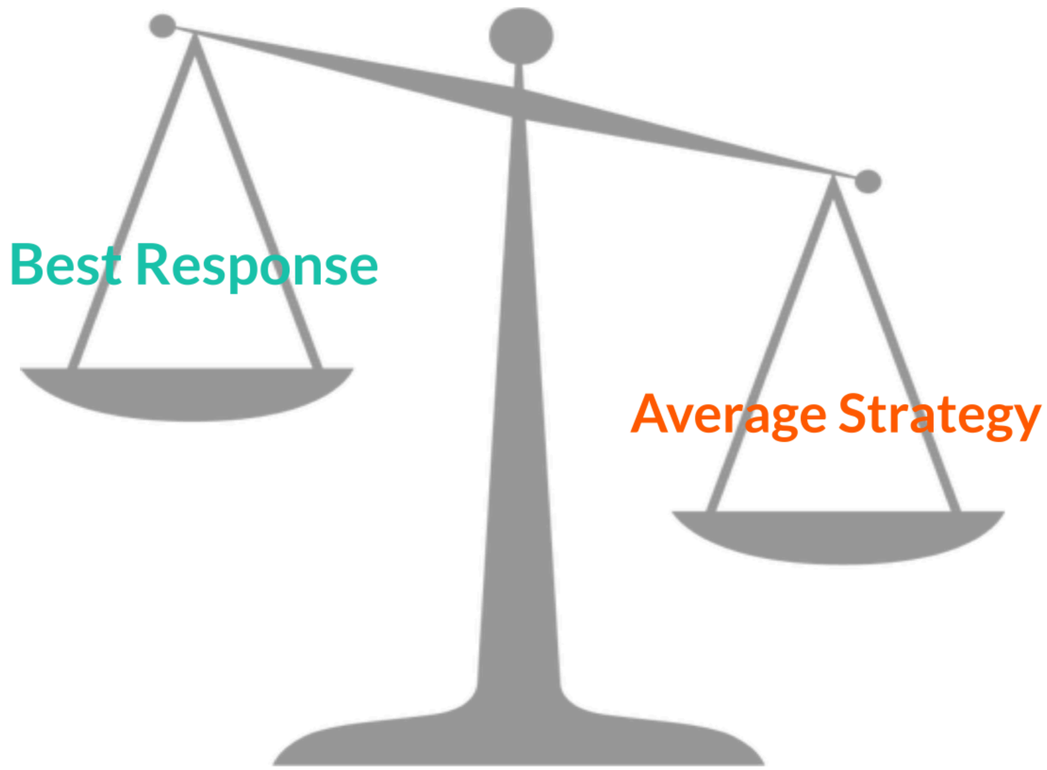
Sample Simultaneous Experience

Update Approximate Best Response Strategies

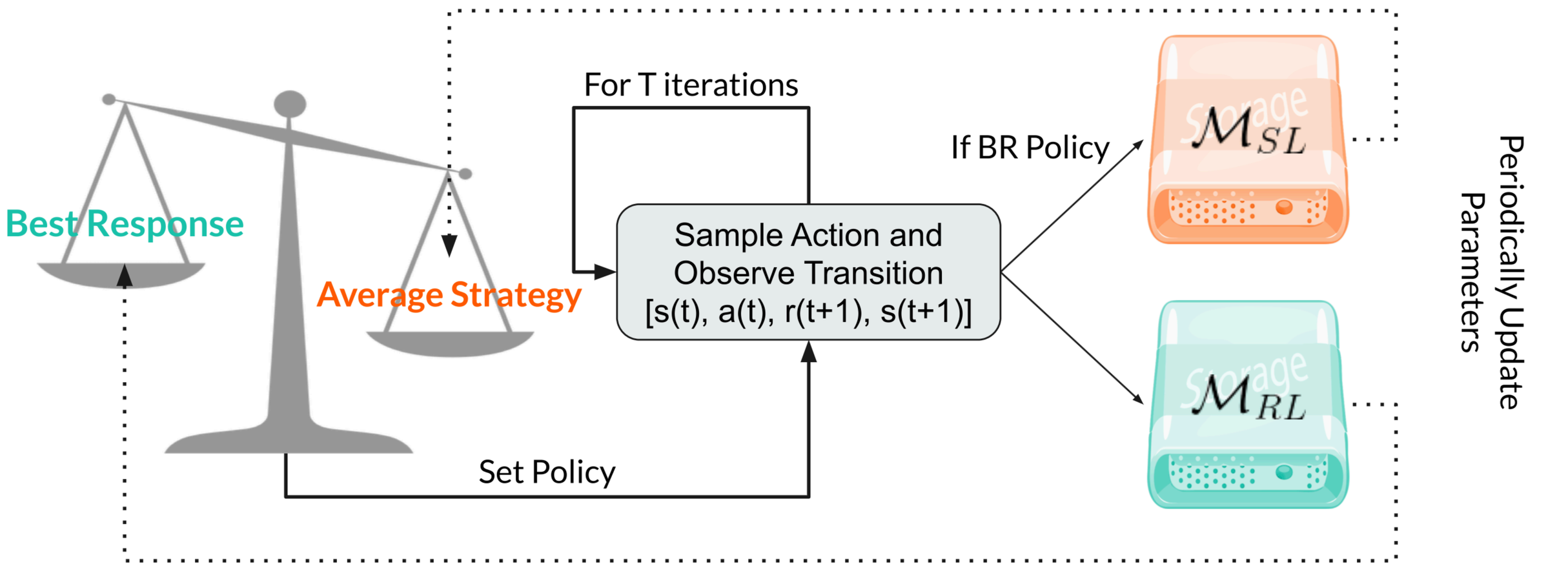
Sample Alternating Experience

Update Average Strategies

Algorithmic Changes for Sampling



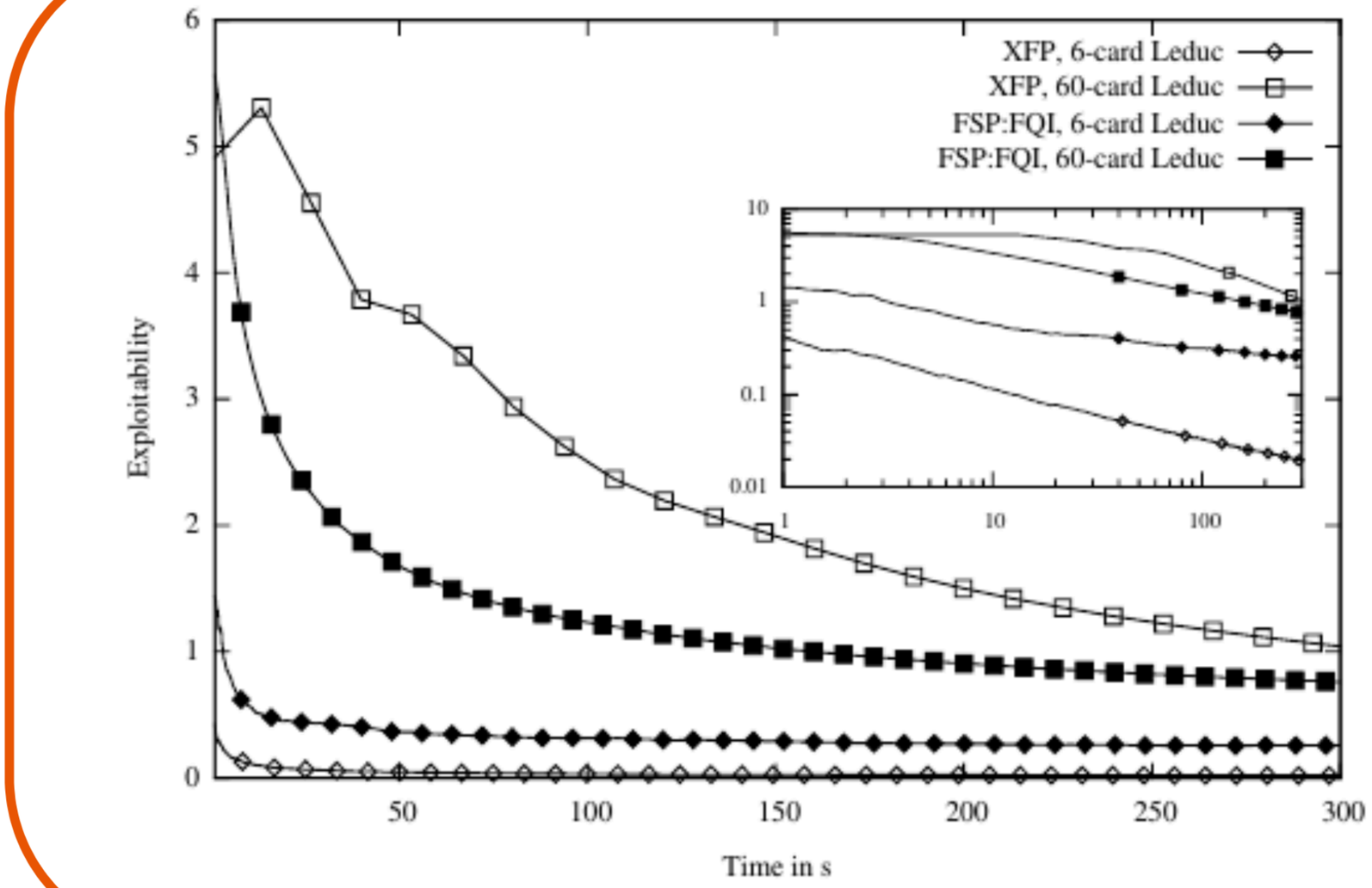
One Episode of NFSP

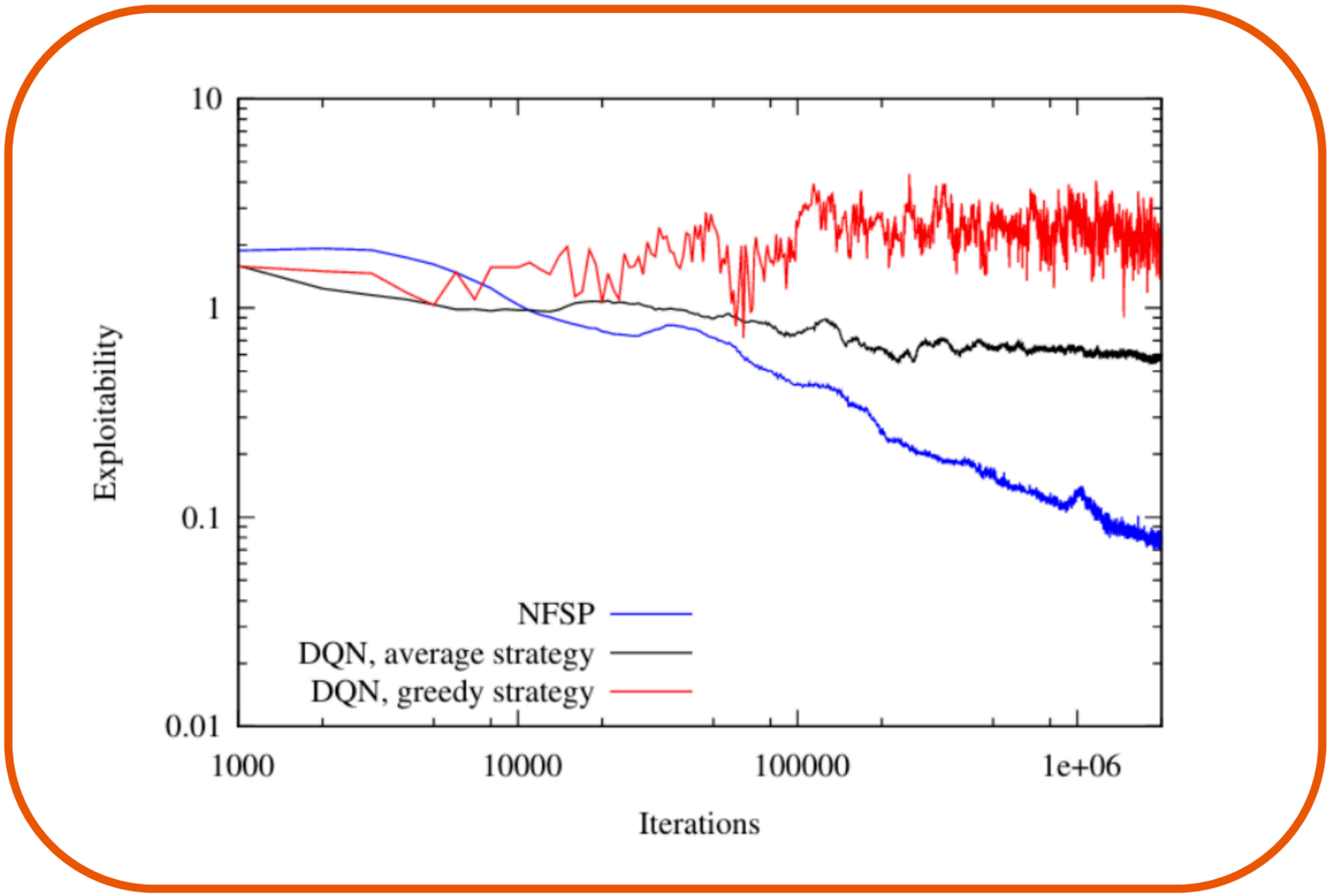


How to measure performance?

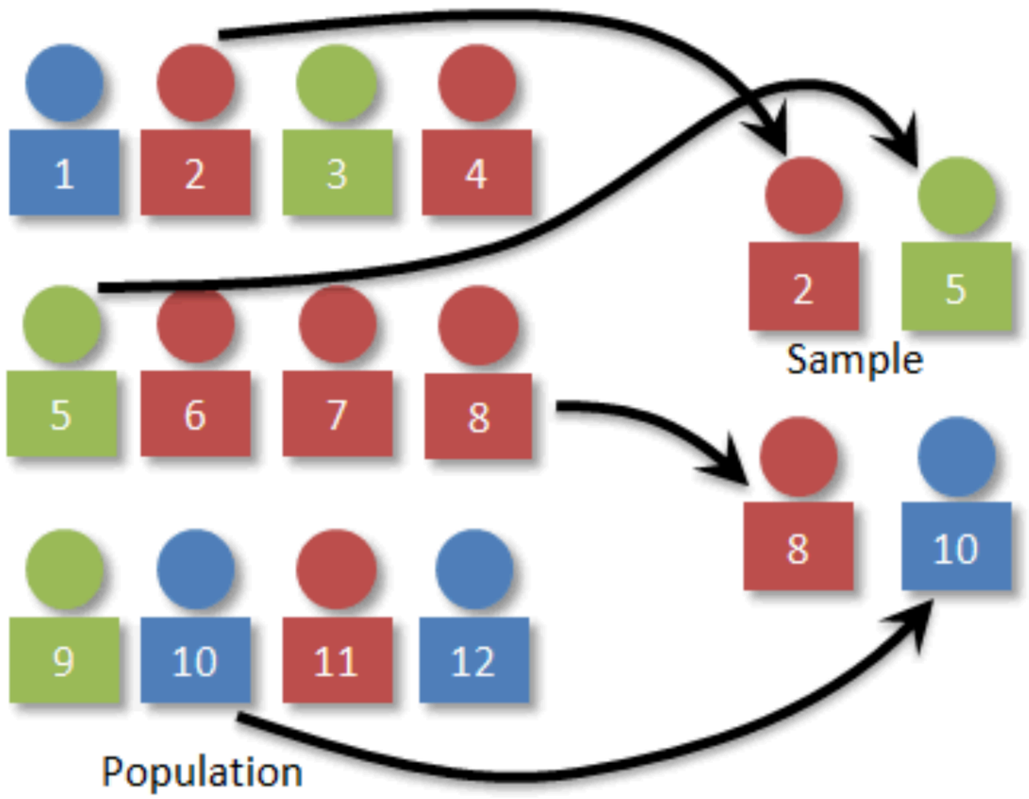
- **Exploitability in General** is the average reward of all players playing best responses to their opponents' strategy.

- **Exploitability in Poker Variants** is measured in milli-big blinds won per game/hand.

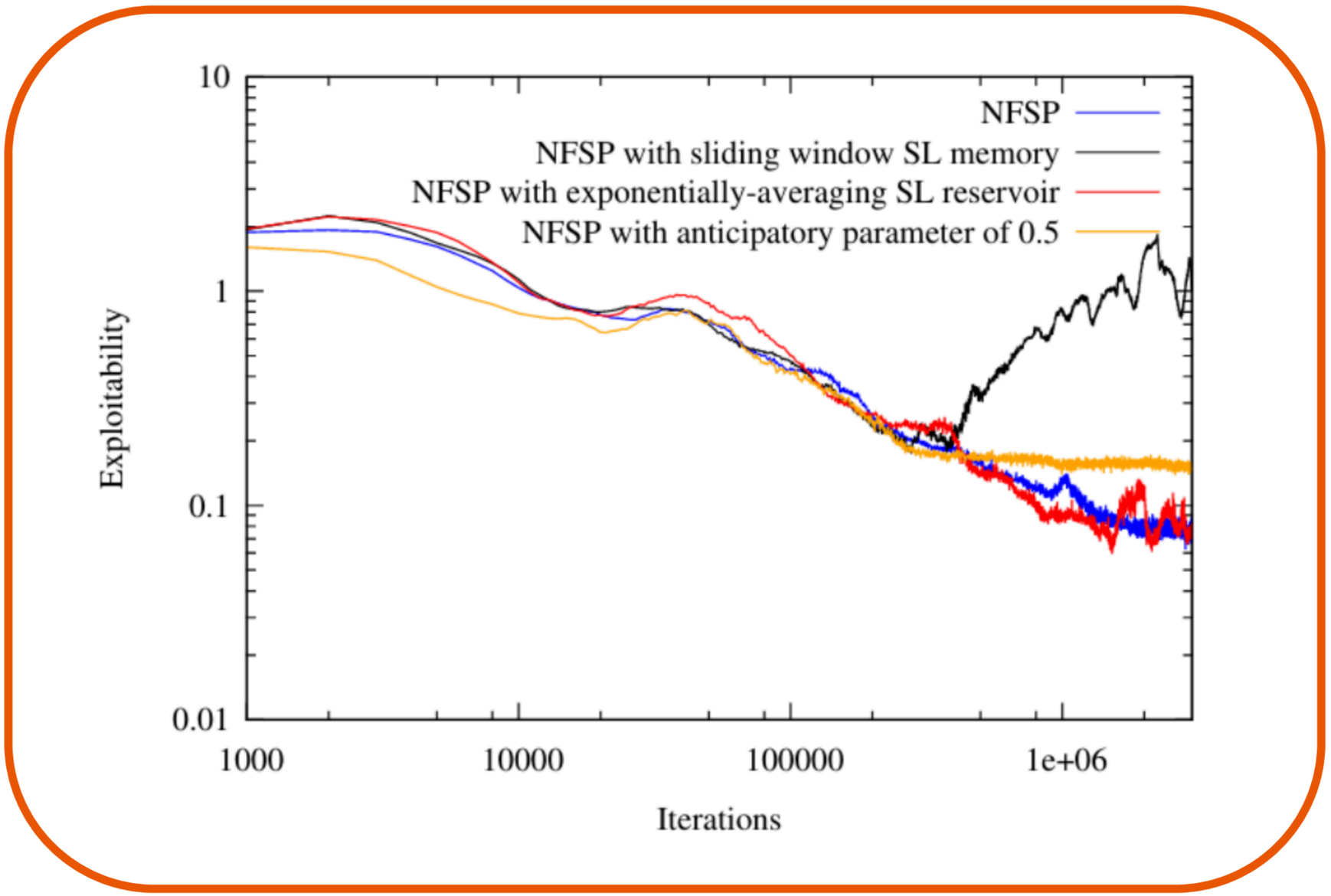

XFP, 6-card Leduc
XFP, 60-card Leduc
FSP:FQI, 6-card Leduc
FSP:FQI, 60-card Leduc
Exploitability
Time in s
0
1
2
3
4
5
6
50
100
150
200
250
300
0.01
0.1
1
10
1
10
100

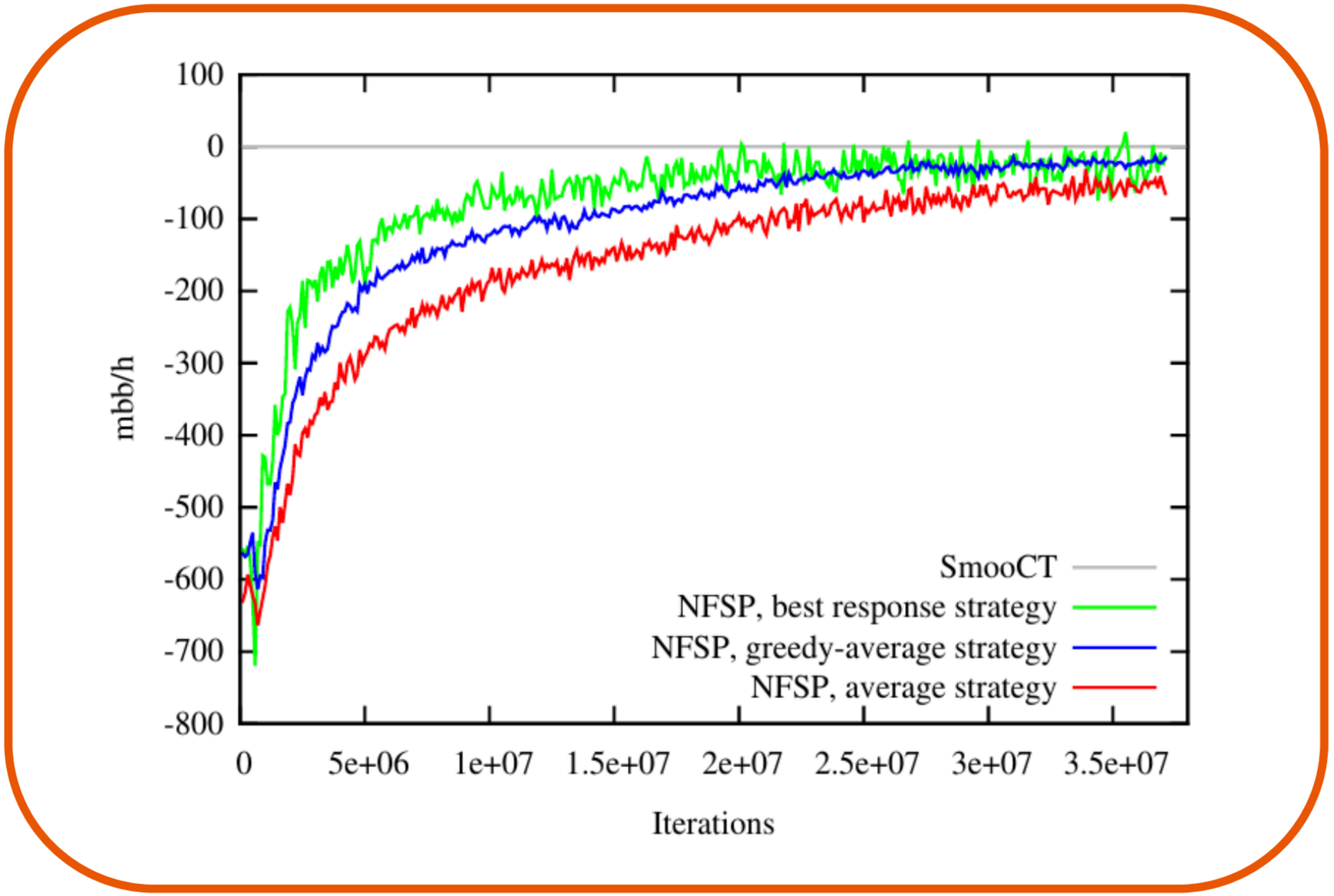

10
1
0.1
0.01
Exploitability
1000
10000
100000
1e+06
Iterations
NFSP
DQN, average strategy
DQN, greedy strategy

Additional Improvements besides NNs



1. Fill Reservoir
2. Now **one by one** consider all items from *(k+1)*th item to *n*th item.
    a. Generate a **random number from 0 to *i*** where *i* is the index of the current item in *stream[]*. Let the generated random **number is *j***.
    b. **If** *j* is **in range** 0 to *k-1*, **replace *reservoir[j]* with stream***[i]*


10
1
0.1
0.01
1000
10000
100000
1e+06
Exploitability
Iterations
NFSP
NFSP with sliding window SL memory
NFSP with exponentially-averaging SL reservoir
NFSP with anticipatory parameter of 0.5


100
0
-100
-200
-300
-400
-500
-600
-700
-800
mbb/h
0
5e+06
1e+07
1.5e+07
2e+07
2.5e+07
3e+07
3.5e+07
Iterations
SmooCT
NFSP, best response strategy
NFSP, greedy-average strategy
NFSP, average strategy

Comparison to State-of-the-Art Approaches

Match-up	Win rate (mbb/h)
escabeche	-52.1 ± 8.5
SmooCT	-17.4 ± 9.0
Hyperborean	-13.6 ± 9.2

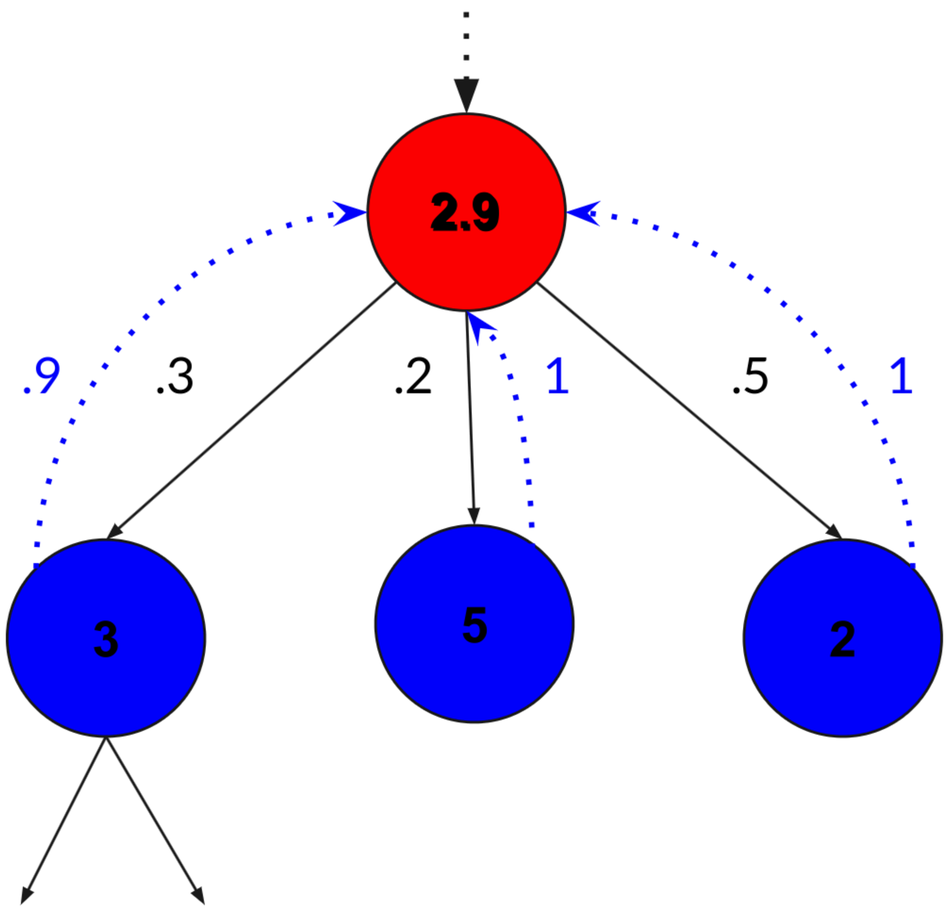
Counterfactual Regret Minimization

Counterfactual: “If I had known the value of the next state…”

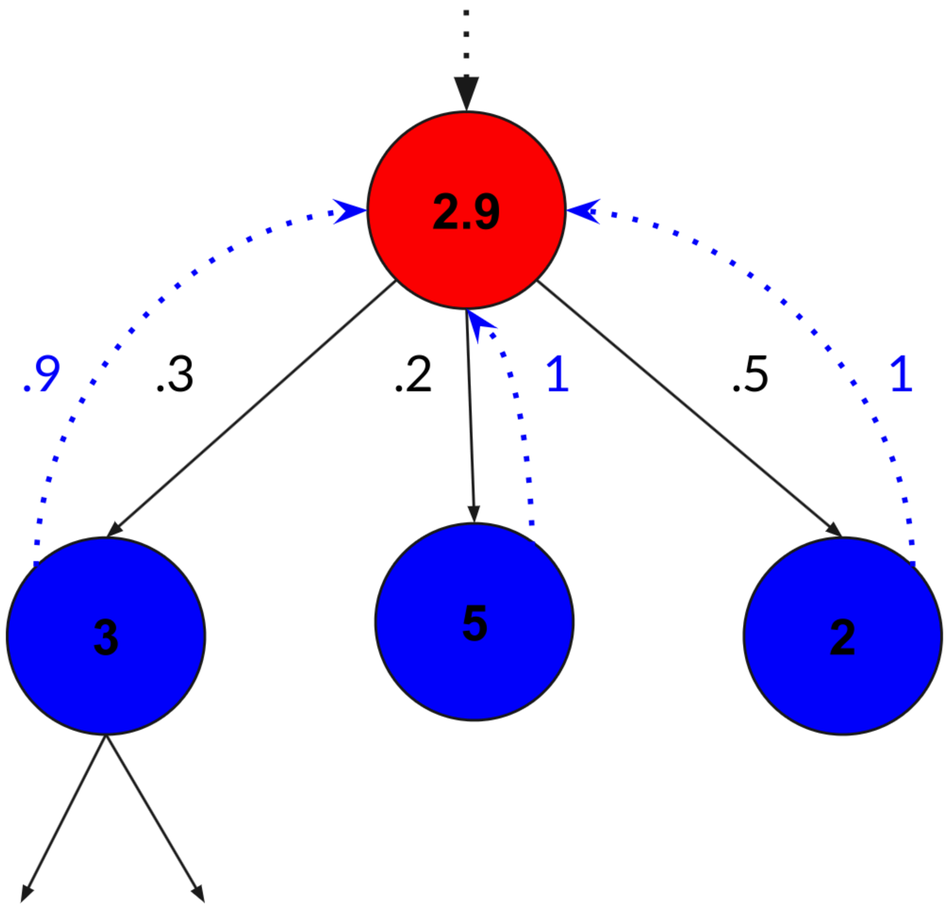
Regret: “How much better would I have done if I did something else instead?”

Minimization: “What strategy minimizes my overall regret?”

Example of Counterfactual Regret



Example of Counterfactual Regret



Value of Parent: 3*0.3 + 5*0.2 + 2*0.5 = 2.9

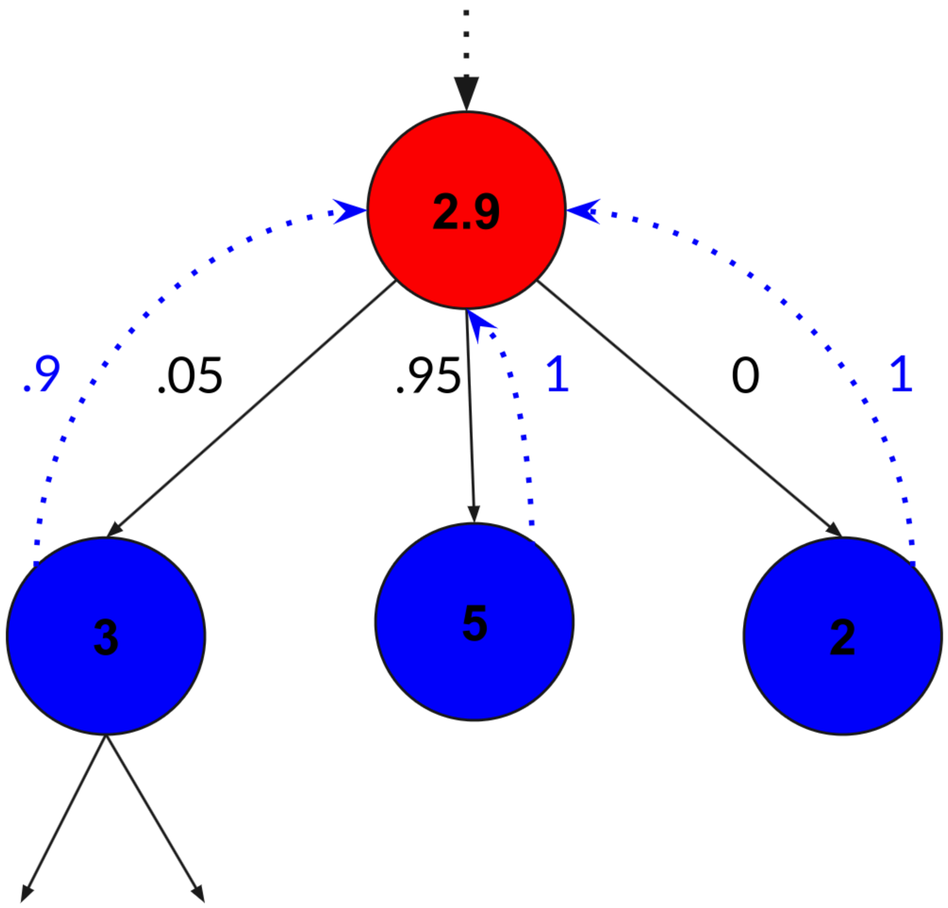
Regret of Children:

- 3 - 2.9 = 0.1
- **5 - 2.9 = 2.1**
- 2 - 2.9 = -0.9

Strategy Update:

- 0.1 / 2.2 = 0.05
- 2.1 / 2.2 = 0.95
- 0

Example of Counterfactual Regret



Value of Parent: 3*0.3 + 5*0.2 + 2*0.5 = 2.9

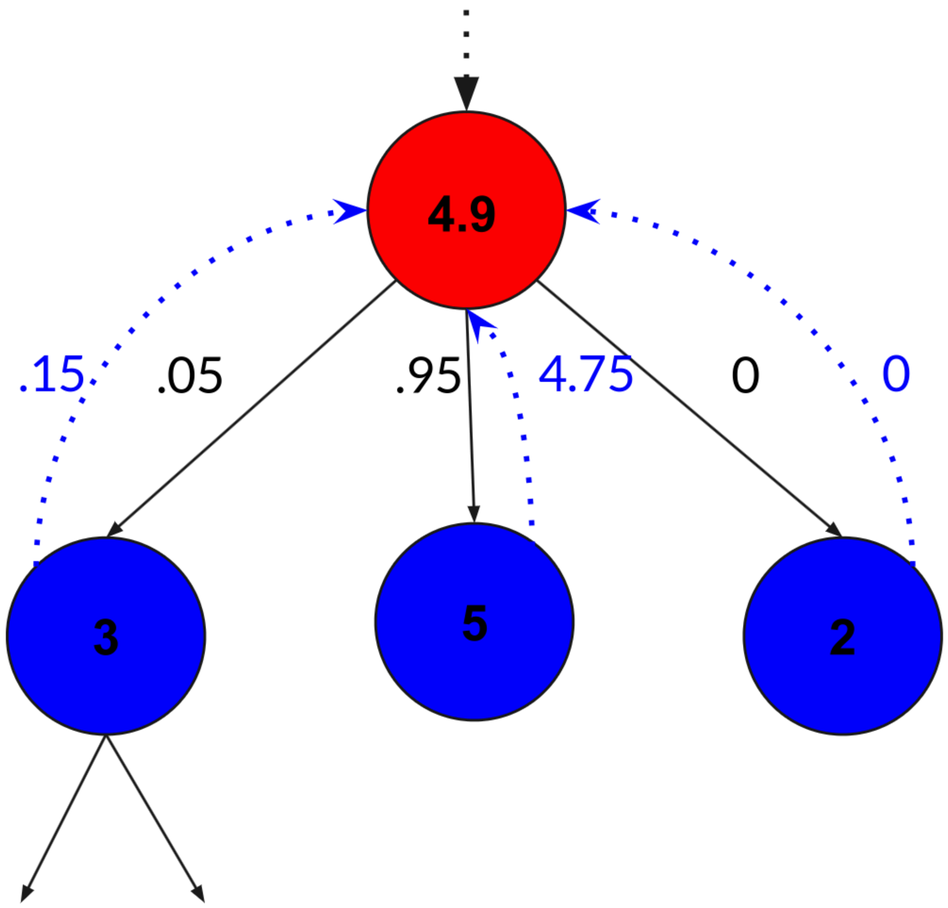
Regret of Children:

- 3 - 2.9 = 0.1
- **5 - 2.9 = 2.1**
- 2 - 2.9 = -0.9

Strategy Update:

- 0.1 / 2.2 = 0.05
- 2.1 / 2.2 = 0.95
- 0

Example of Counterfactual Regret



Value of Parent: 3*0.3 + 5*0.2 + 2*0.5 = 2.9

Regret of Children:

- 3 - 2.9 = 0.1
- **5 - 2.9 = 2.1**
- 2 - 2.9 = -0.9

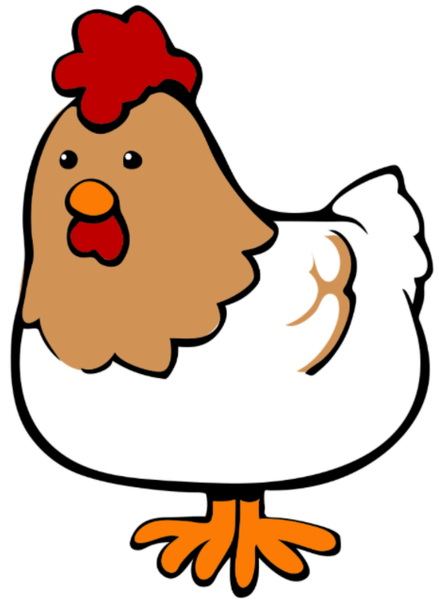
Strategy Update:

- 0.1 / 2.2 = 0.05
- 2.1 / 2.2 = 0.95
- 0

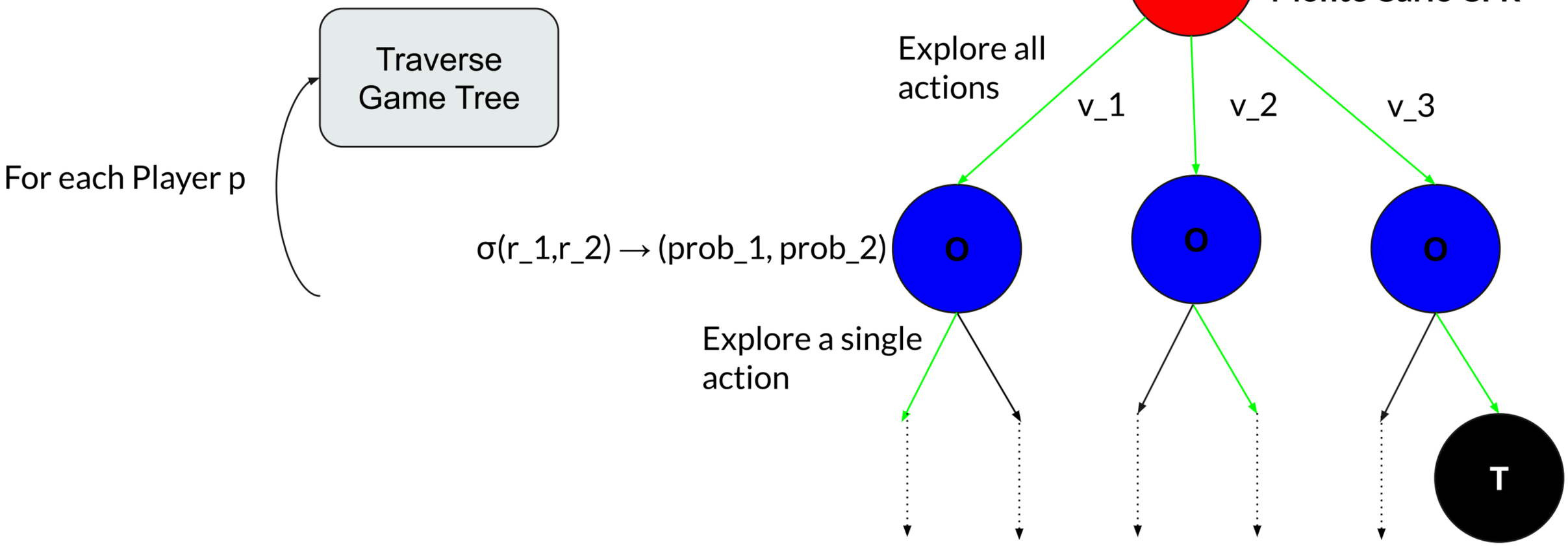
Problems with State-of-the-Art Approaches

- Manual and domain-specific abstractions are needed to scale to large games

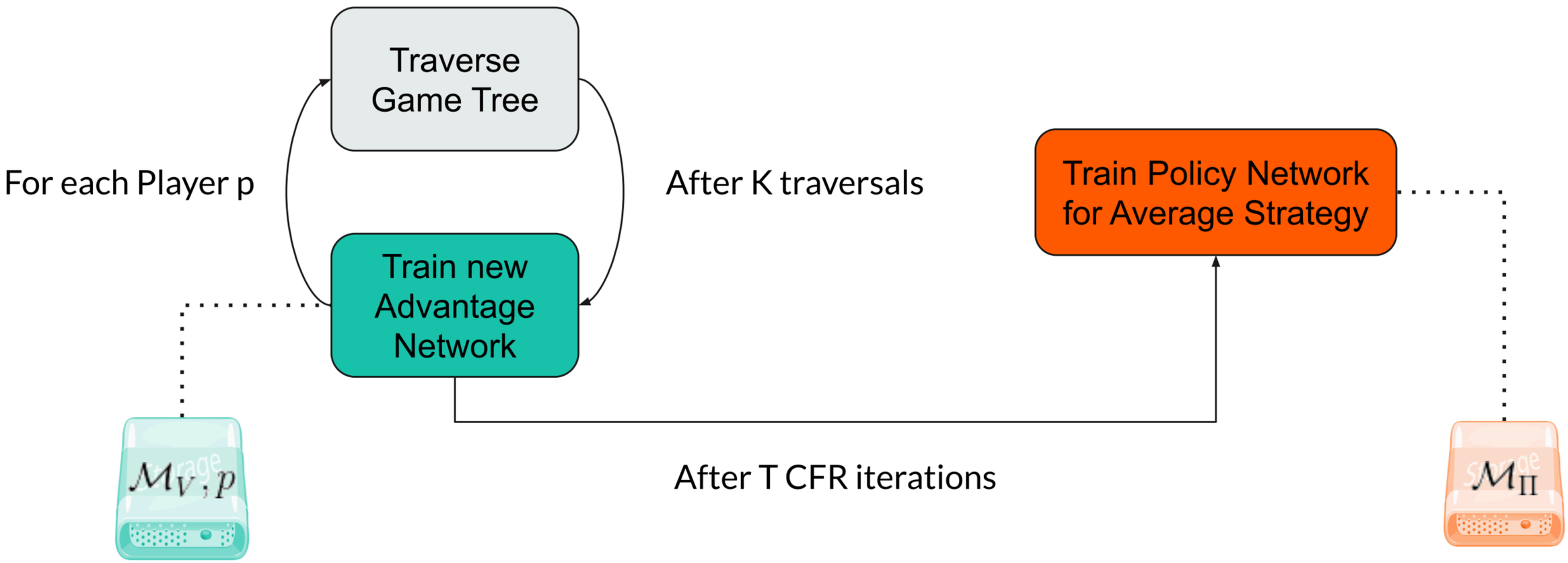
- Chicken-and-egg problem as a good abstraction requires knowing the equilibrium beforehand

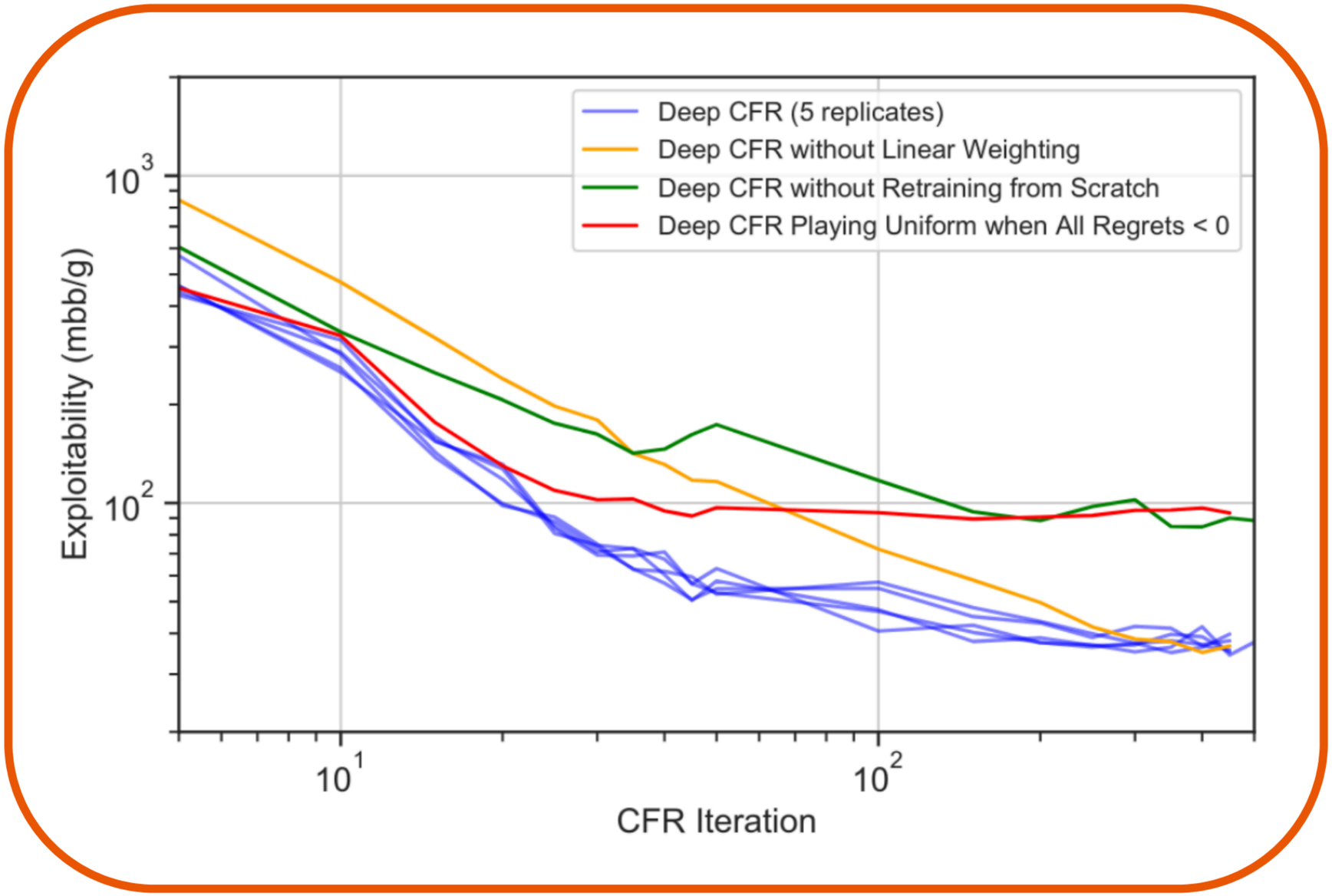


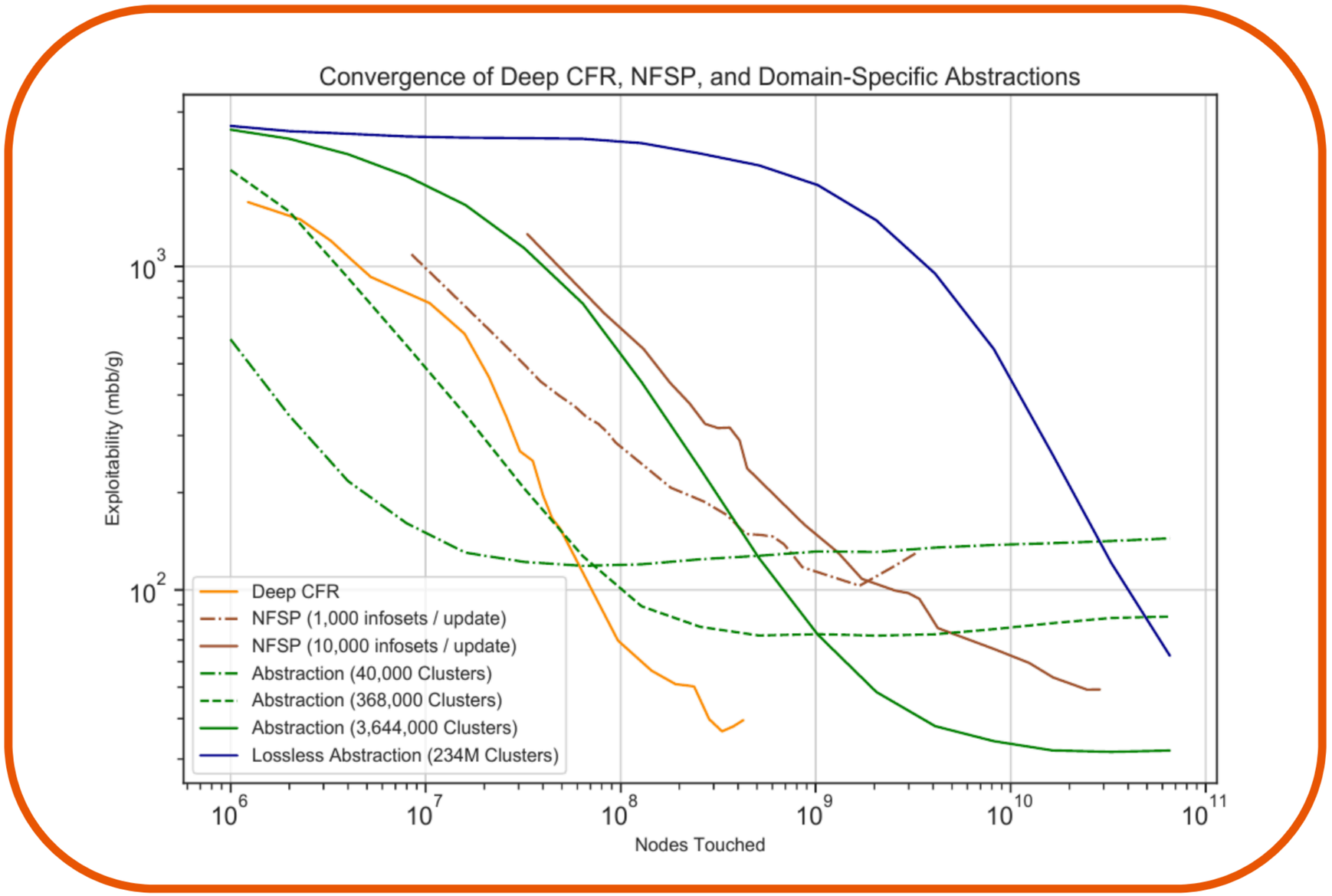
One Episode of Deep CFR Minimization

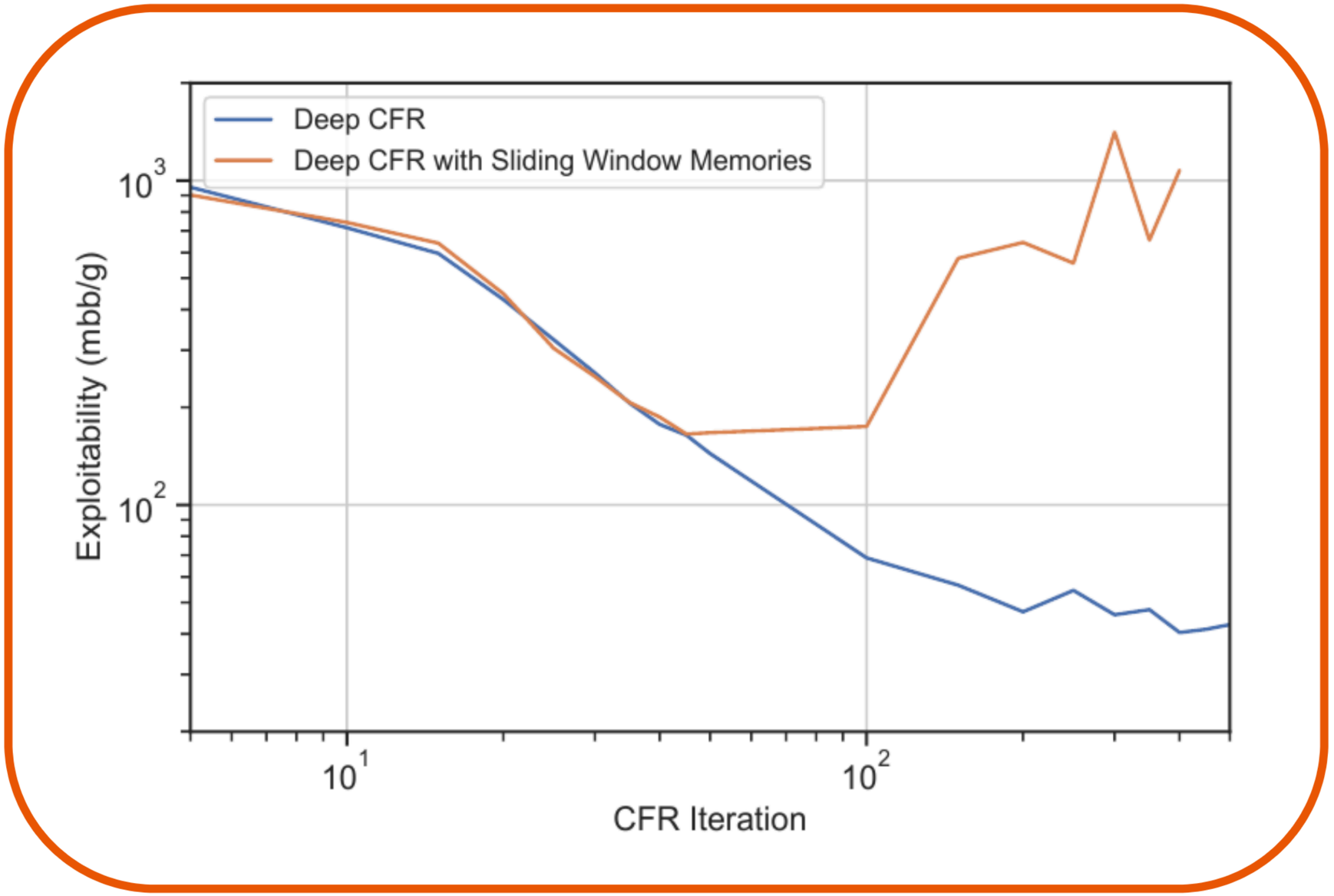


One Episode of Deep CFR Minimization

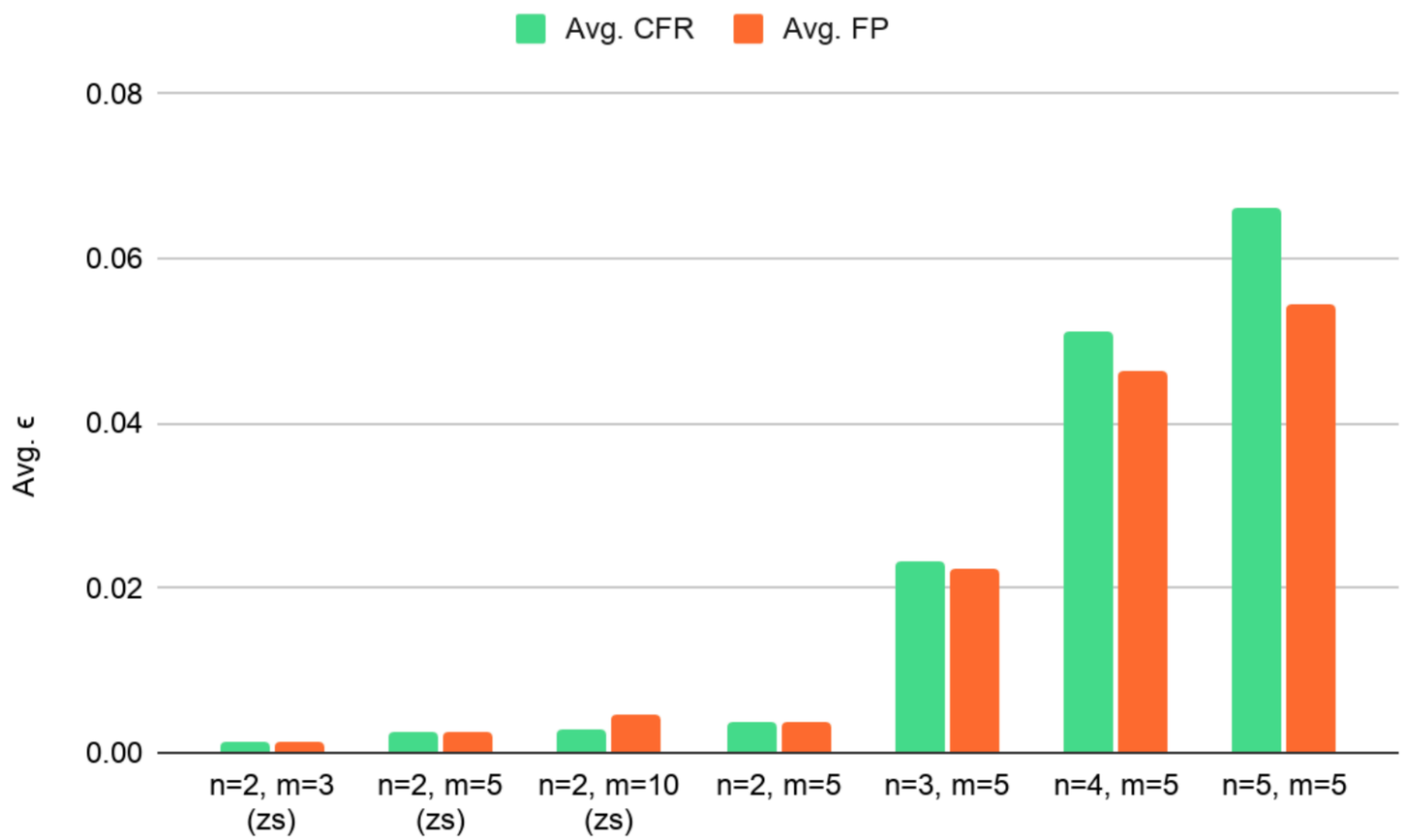



Deep CFR (5 replicates)
Deep CFR without Linear Weighting
Deep CFR without Retraining from Scratch
Deep CFR Playing Uniform when All Regrets < 0
Exploitability (mbb/g)
10^3
10^2
10^1
10^2
CFR Iteration

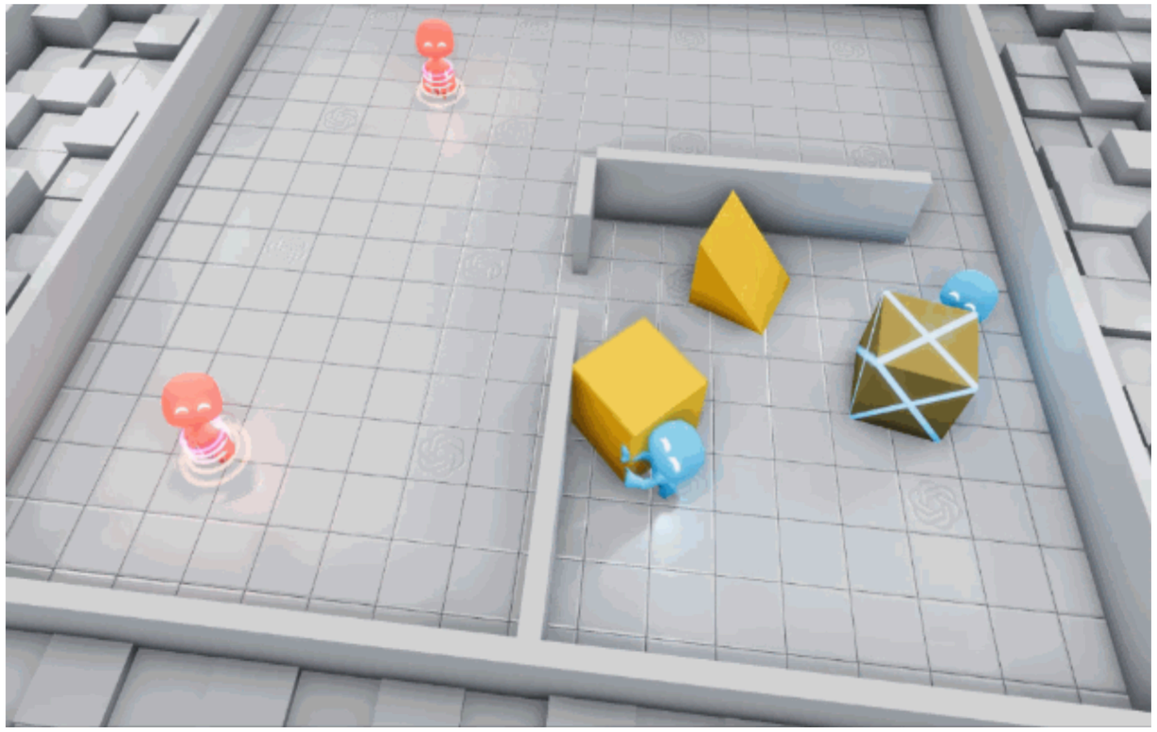

Convergence of Deep CFR, NFSP, and Domain-Specific Abstractions
Exploitability (mbb/g)
10^3
10^2
10^6
10^7
10^8
10^9
10^{10}
10^{11}
Nodes Touched
Deep CFR
NFSP (1,000 infosets / update)
NFSP (10,000 infosets / update)
Abstraction (40,000 Clusters)
Abstraction (368,000 Clusters)
Abstraction (3,644,000 Clusters)
Lossless Abstraction (234M Clusters)

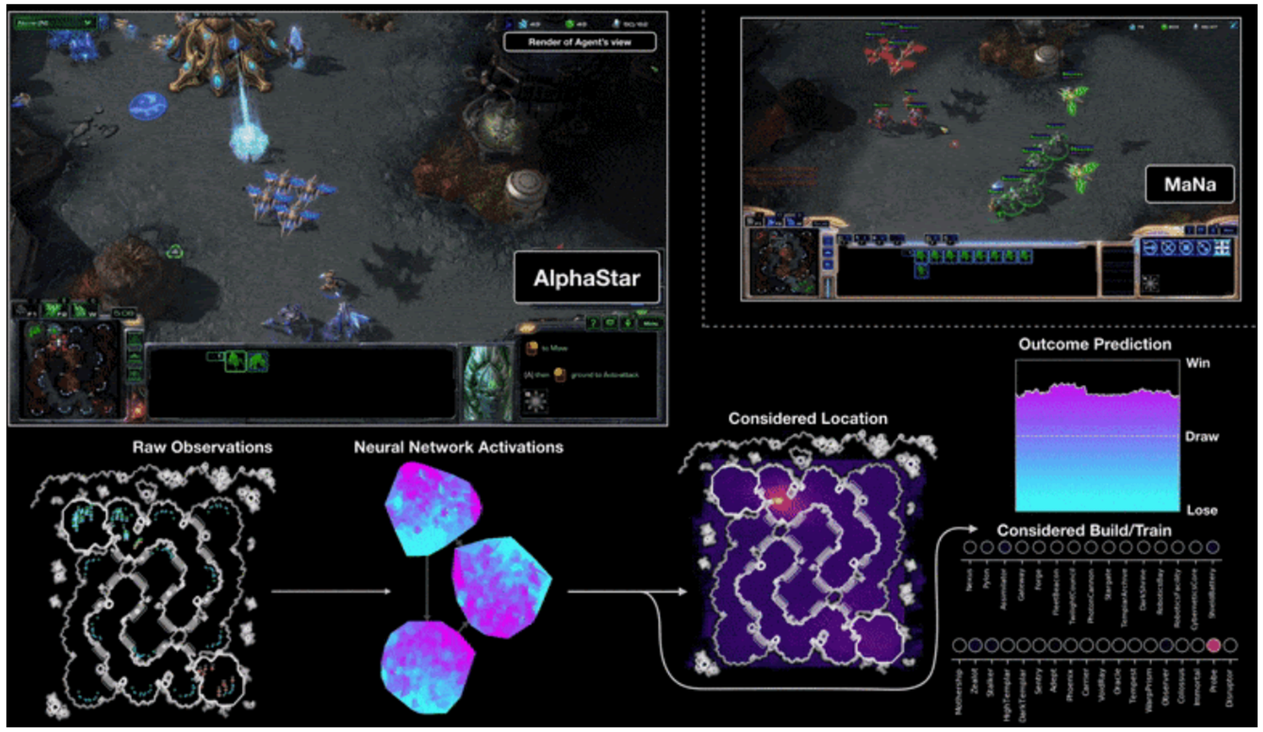

Deep CFR
Deep CFR with Sliding Window Memories
10^3
10^2
Exploitability (mbb/g)
10^1
10^2
CFR Iteration

Which one to use? NFSP or Deep CFR Minimization?



Astonishing Results





Imperfect-information games are games where some information is hidden

Fictitious Play converges to a Nash equilibrium in two-player zero-sum games

NFSP uses function approximation via neural networks to calculate approximate Nash-equilibria

Previous State-of-the-Art approaches massively relied on domain-knowledge

An alternative to NFSP is Deep CFR Minimization

Astonishing breakthroughs can be achieved with self-play and multi-agent reinforcement learning

References

Online Articles:
https://en.wikipedia.org/wiki/Prisoner%27s_dilemma
https://towardsdatascience.com/introduction-to-fictitious-play-12a8bc4ed1bb
https://towardsdatascience.com/fictitious-self-play-30b76e30ec6a
https://towardsdatascience.com/neural-fictitious-self-play-800612b4a53f
https://towardsdatascience.com/neural-fictitious-self-play-in-practice-132836b69bf5
https://towardsdatascience.com/counterfactual-regret-minimization-ff4204bf4205
https://openai.com/blog/emergent-tool-use/
https://deepmind.com/research/publications/AlphaStar-Grandmaster-level-in-StarCraft-II-using-multi-agent-reinforcement-learning

Papers:
“Fictitious Self-Play in Extensive-Form Games” (2015)
“Deep Reinforcement Learning from Self-Play in Imperfect-Information Games” (2016)
“Reinforcement Learning from Self-Play in Imperfect-Information Games” (2017)
“Deep Counterfactual Regret Minimization” (2019)
“Fictitious Play Outperforms Counterfactual Regret Minimization” (2020)

Python Library:
https://nashpy.readthedocs.io/en/stable/