

Aishik Pyne
Harshavardhan Abhichandani
Niharika Shrivastava

Deceptive Poker

Team ID: 8

Agents learn to cheat via collaboration
in a competitive multi-agent RL environment



Introduction

Problem Statement

Introduction

Problem Statement

Texas Hold'em Poker



A popular skill and luck based gambling game

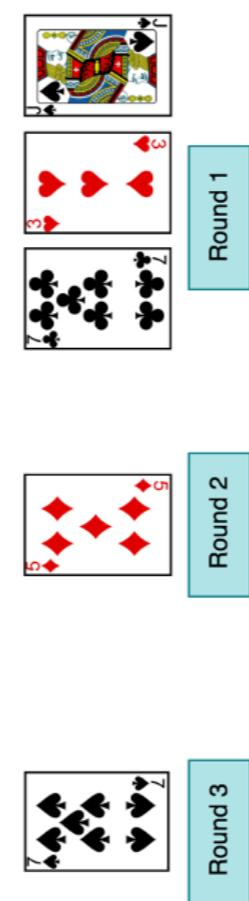
Introduction

Problem Statement

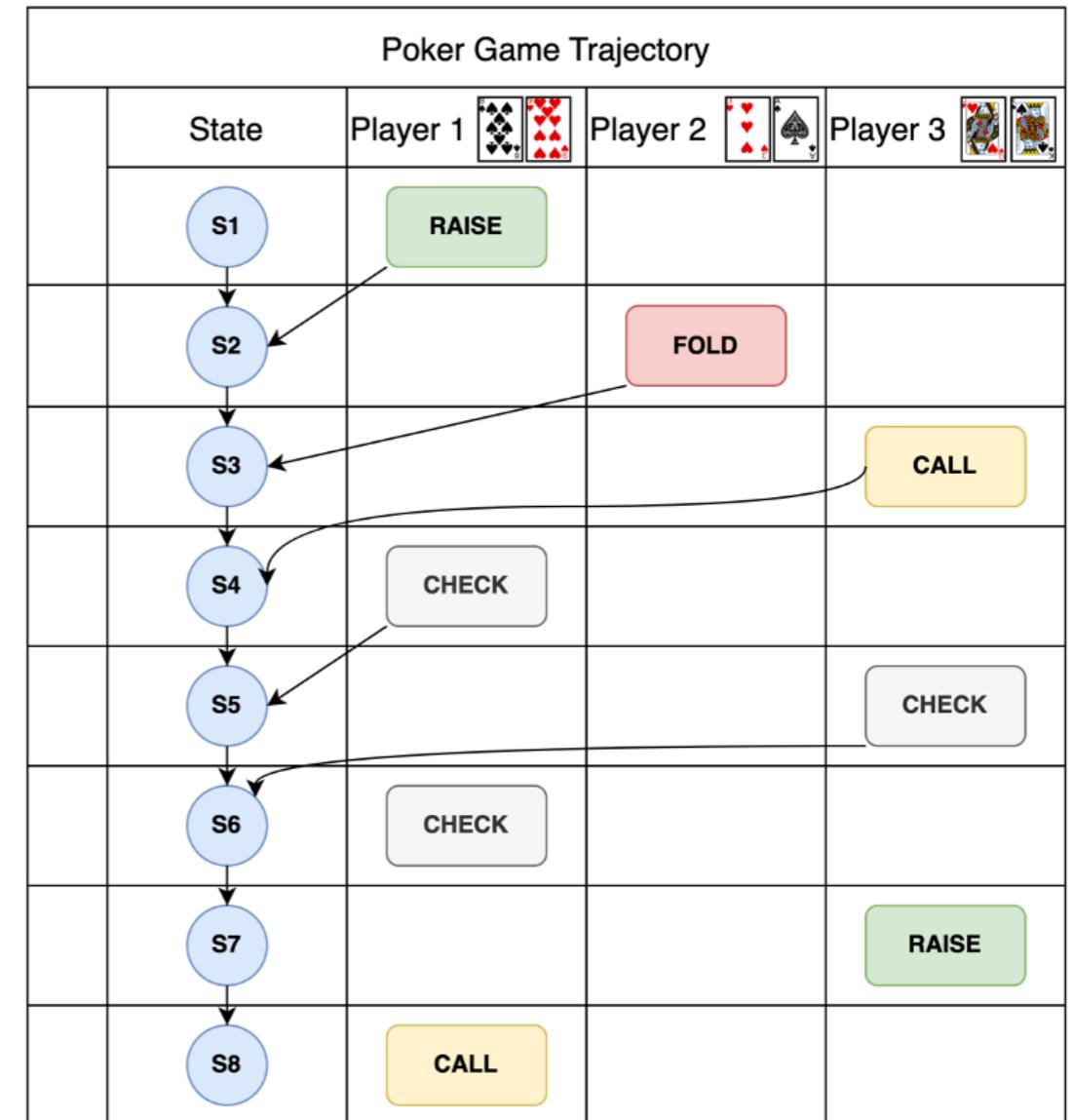
Texas Hold'em Poker



A popular skill and luck based gambling game



One Game of Poker



Winner gets all chips

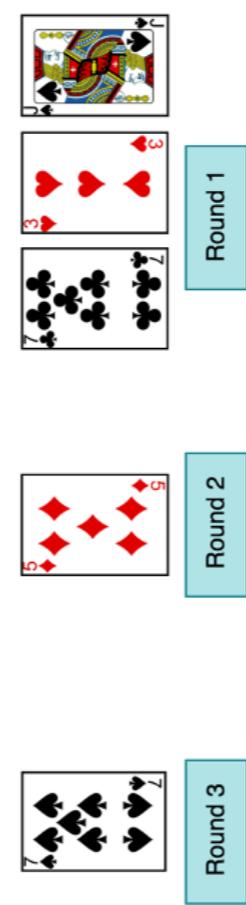
Introduction

Problem Statement

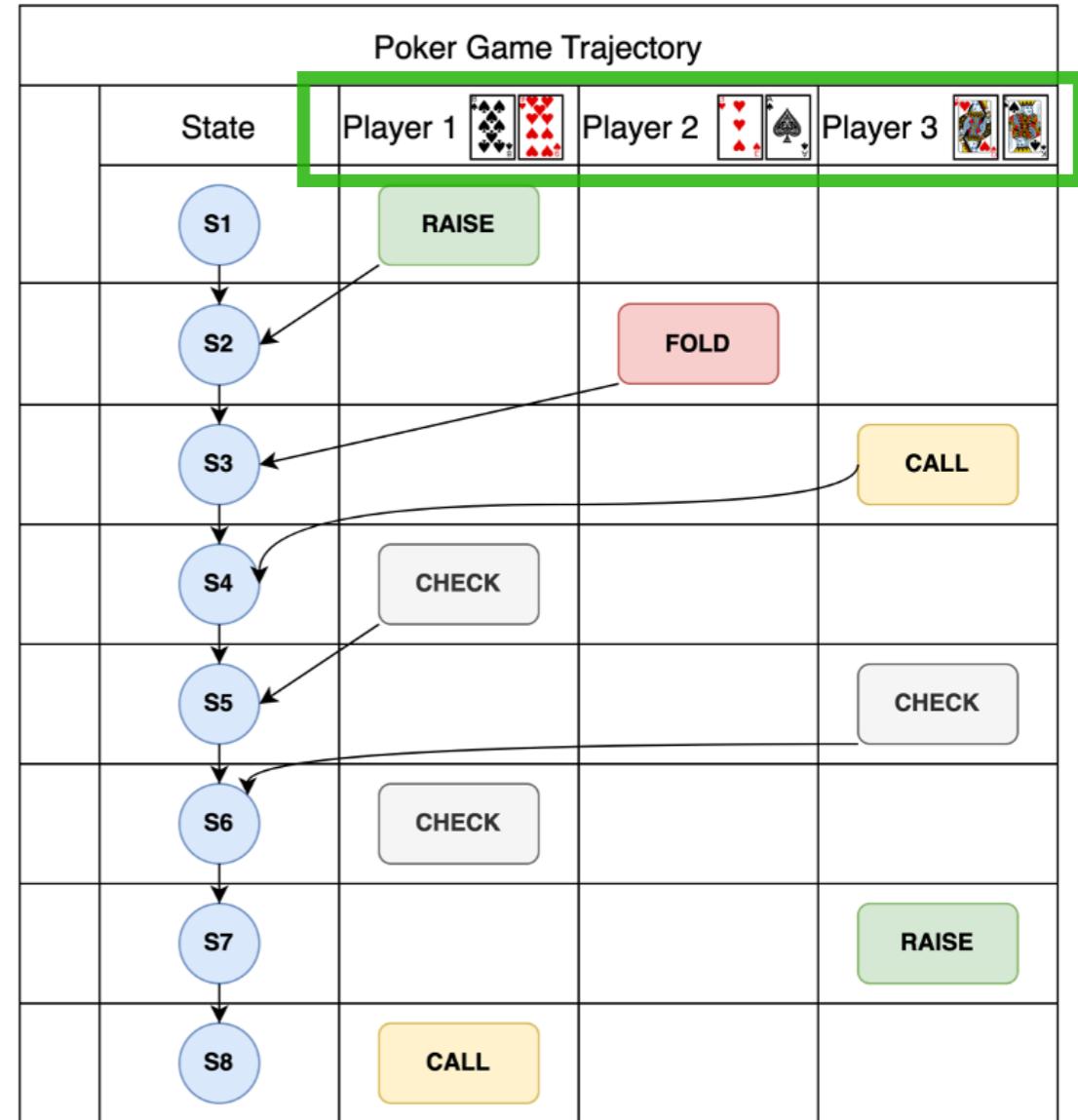
Texas Hold'em Poker



A popular skill and luck based gambling game



One Game of Poker



Winner gets all chips

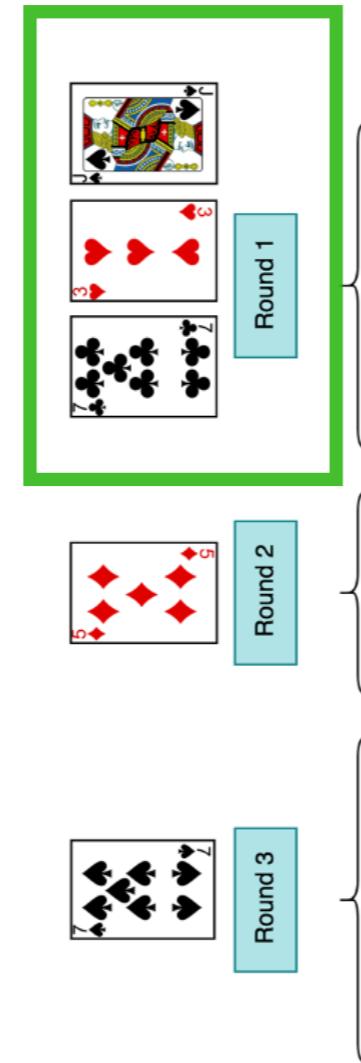
Introduction

Problem Statement

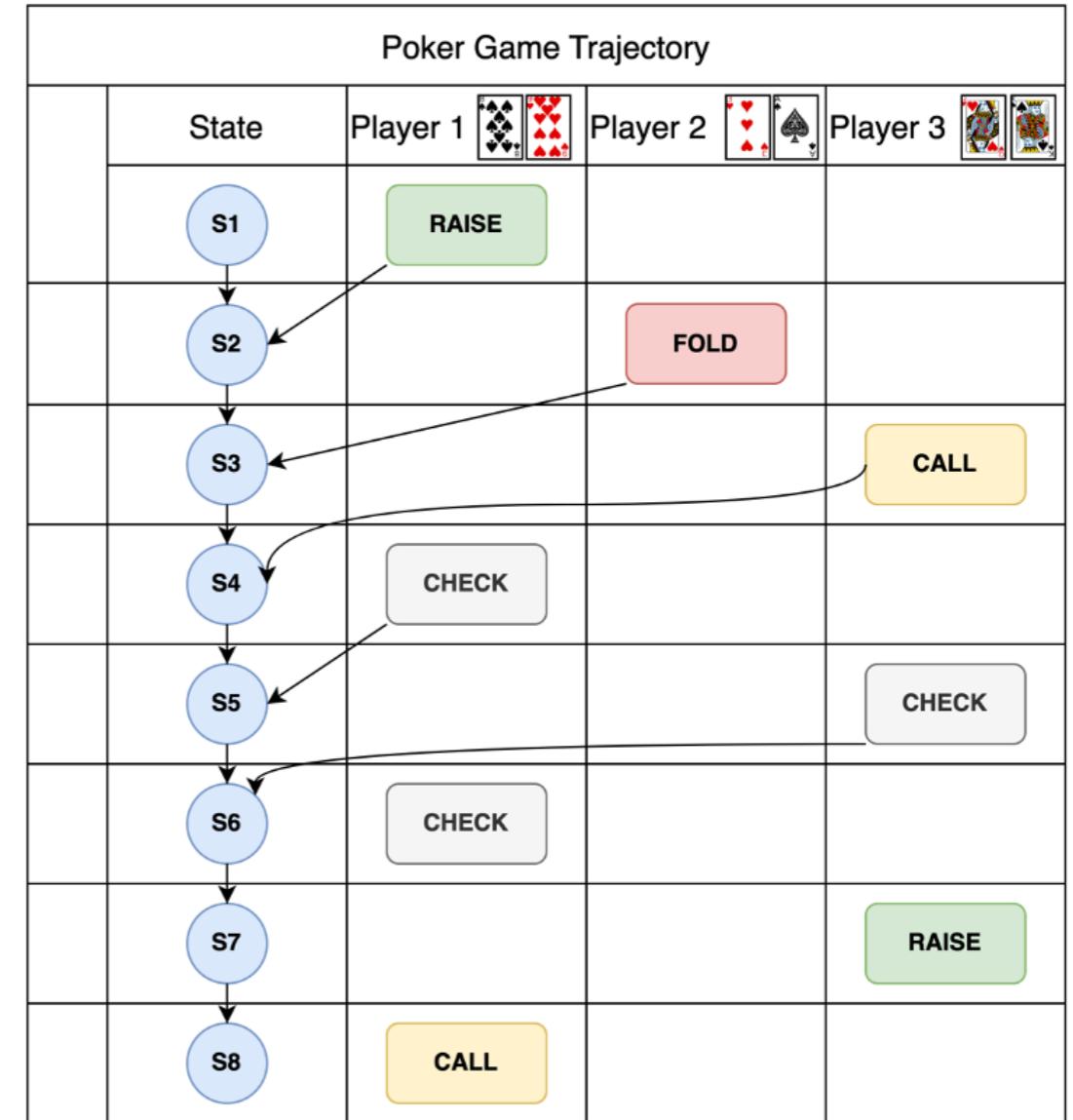
Texas Hold'em Poker



A popular skill and luck based gambling game



One Game of Poker



Winner gets all chips

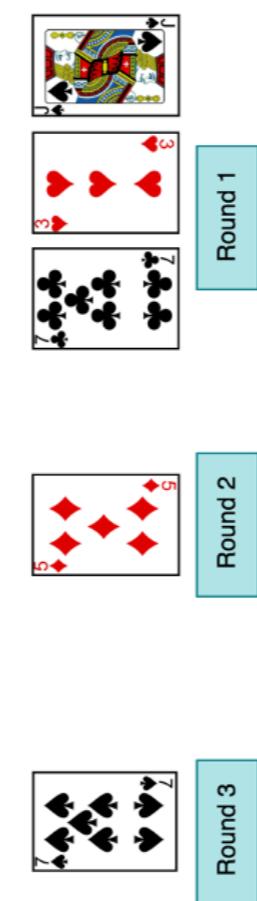
Introduction

Problem Statement

Texas Hold'em Poker

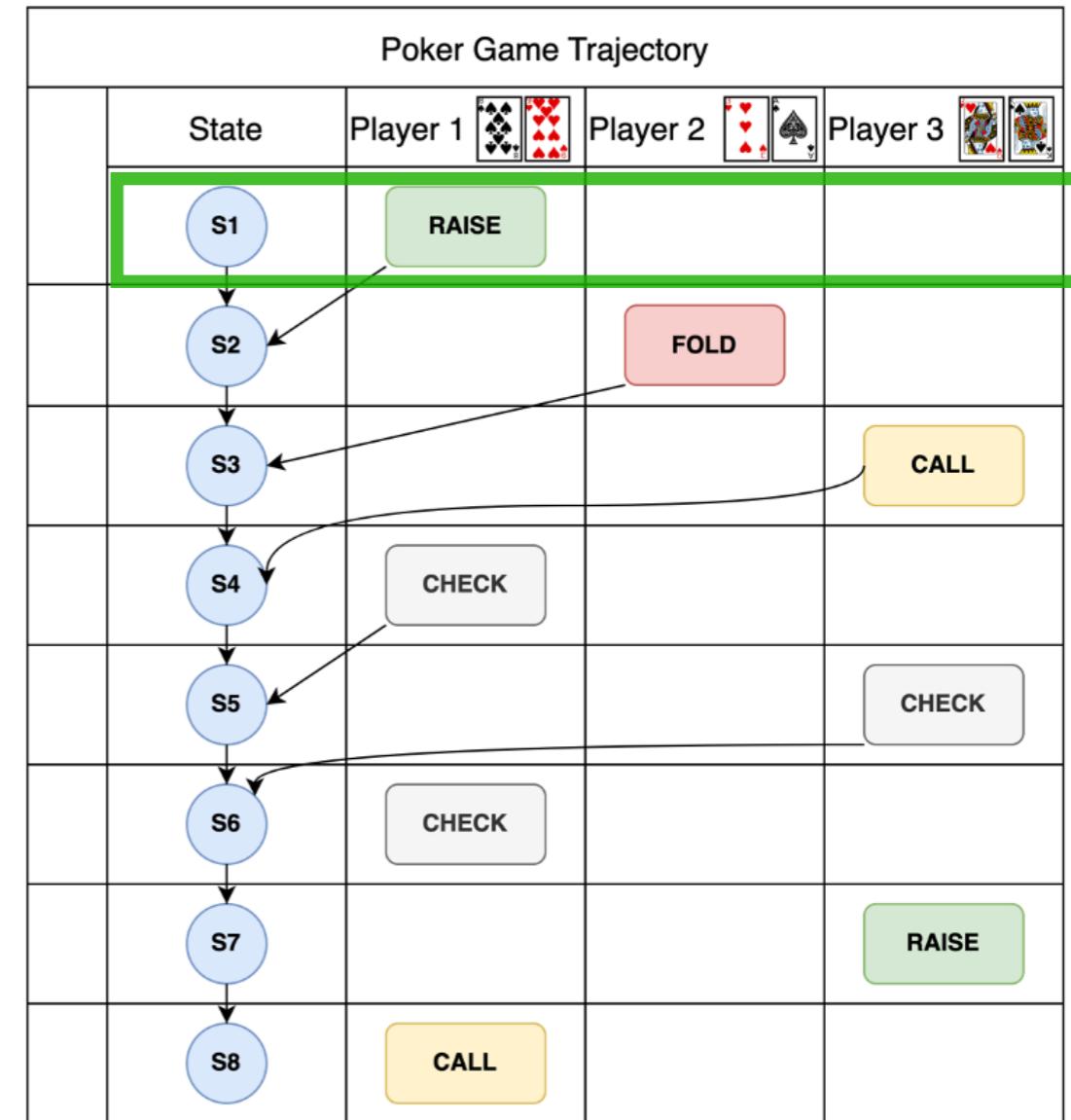


A popular skill and luck based gambling game



CALL RAISE CHECK FOLD

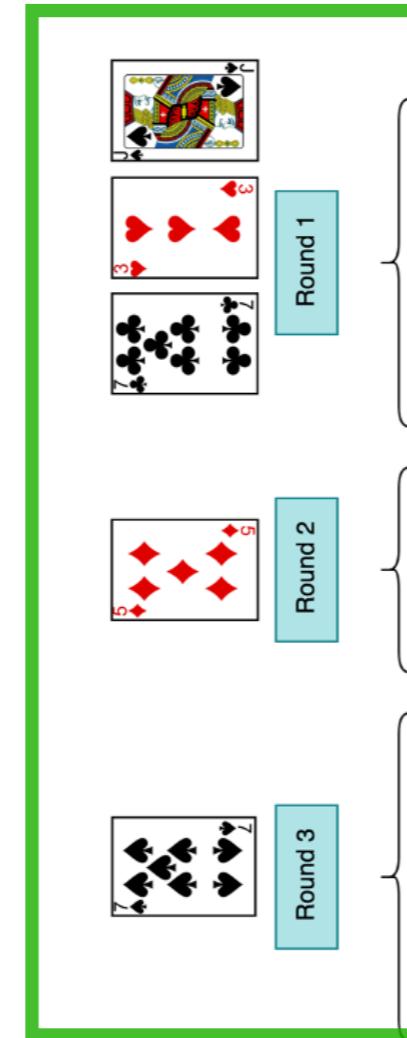
One Game of Poker



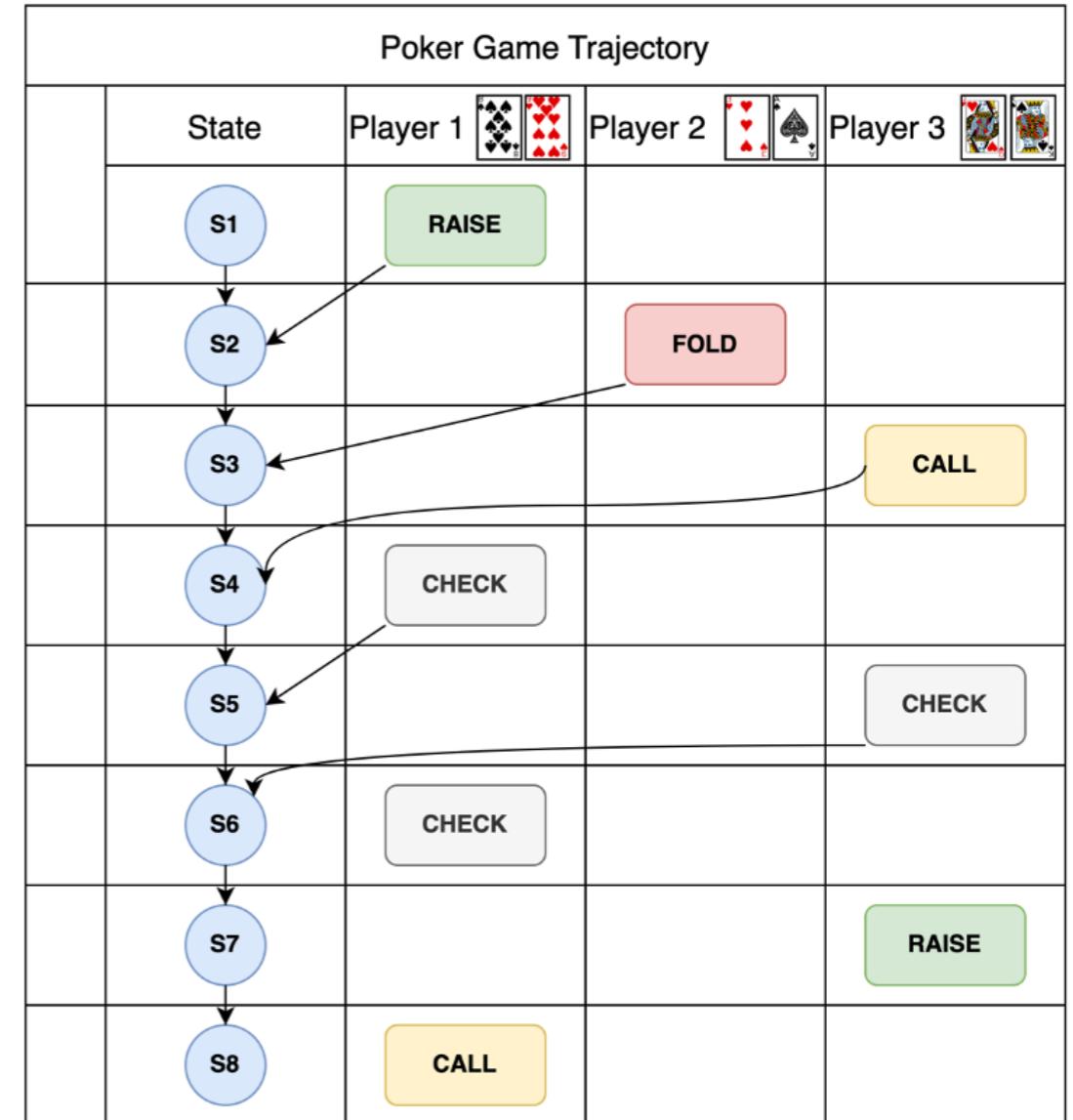
Winner gets all chips

Introduction

Problem Statement



One Game of Poker

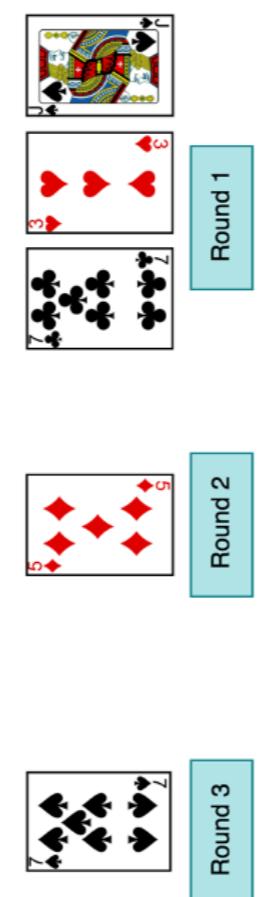


Winner gets all chips

CALL RAISE CHECK FOLD

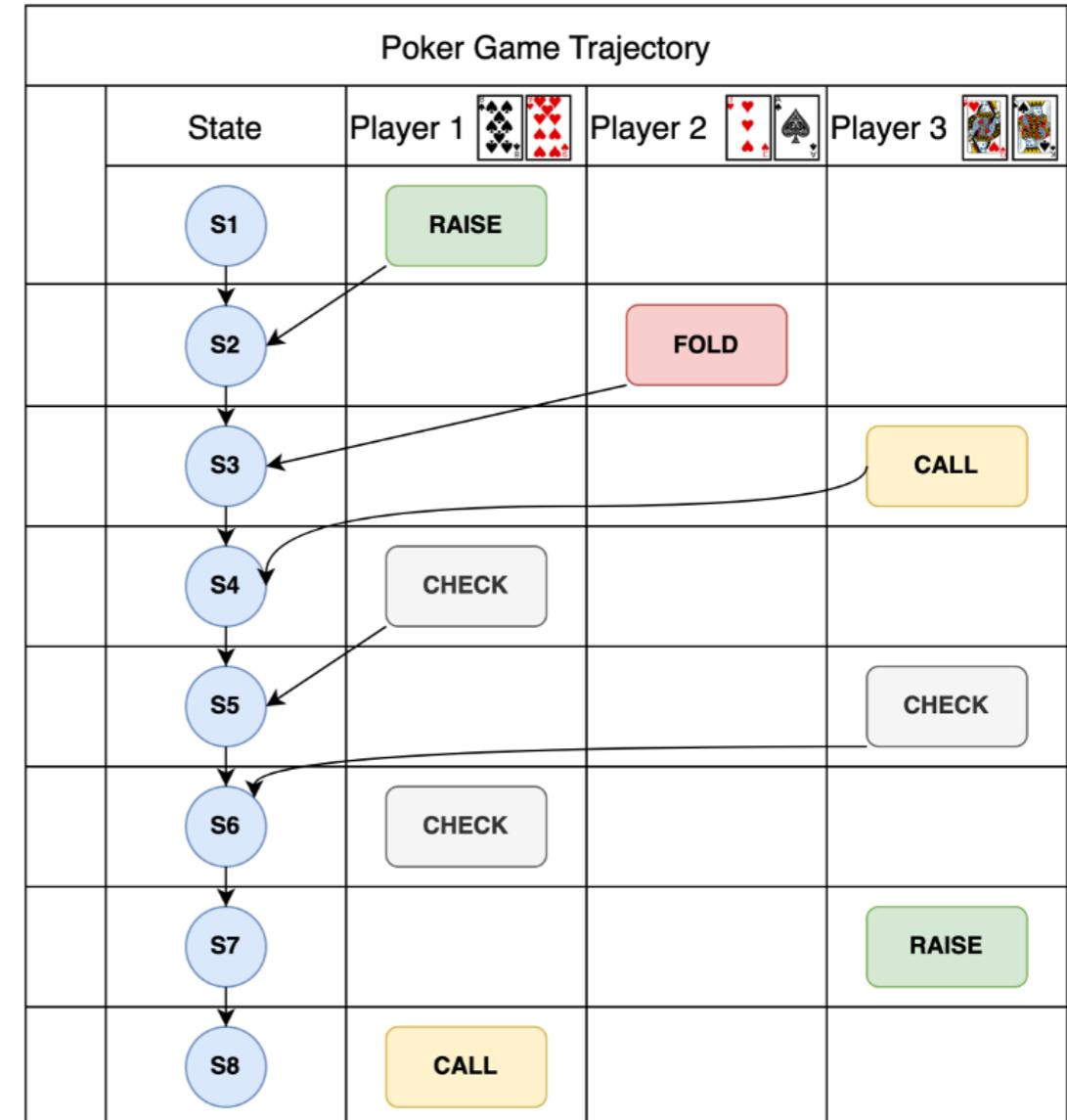
Introduction

Problem Statement



CALL RAISE CHECK FOLD

One Game of Poker



Winner gets all chips

Introduction

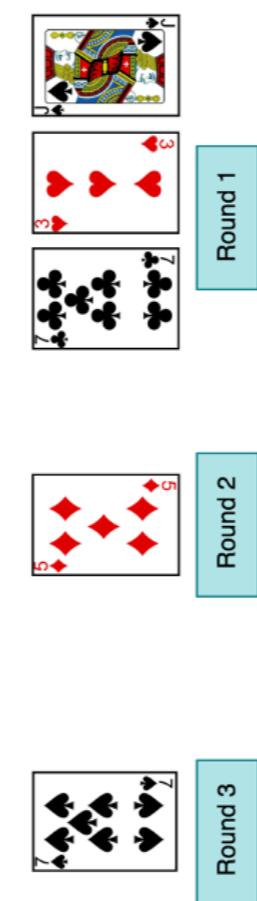
Problem Statement

Texas Hold'em Poker



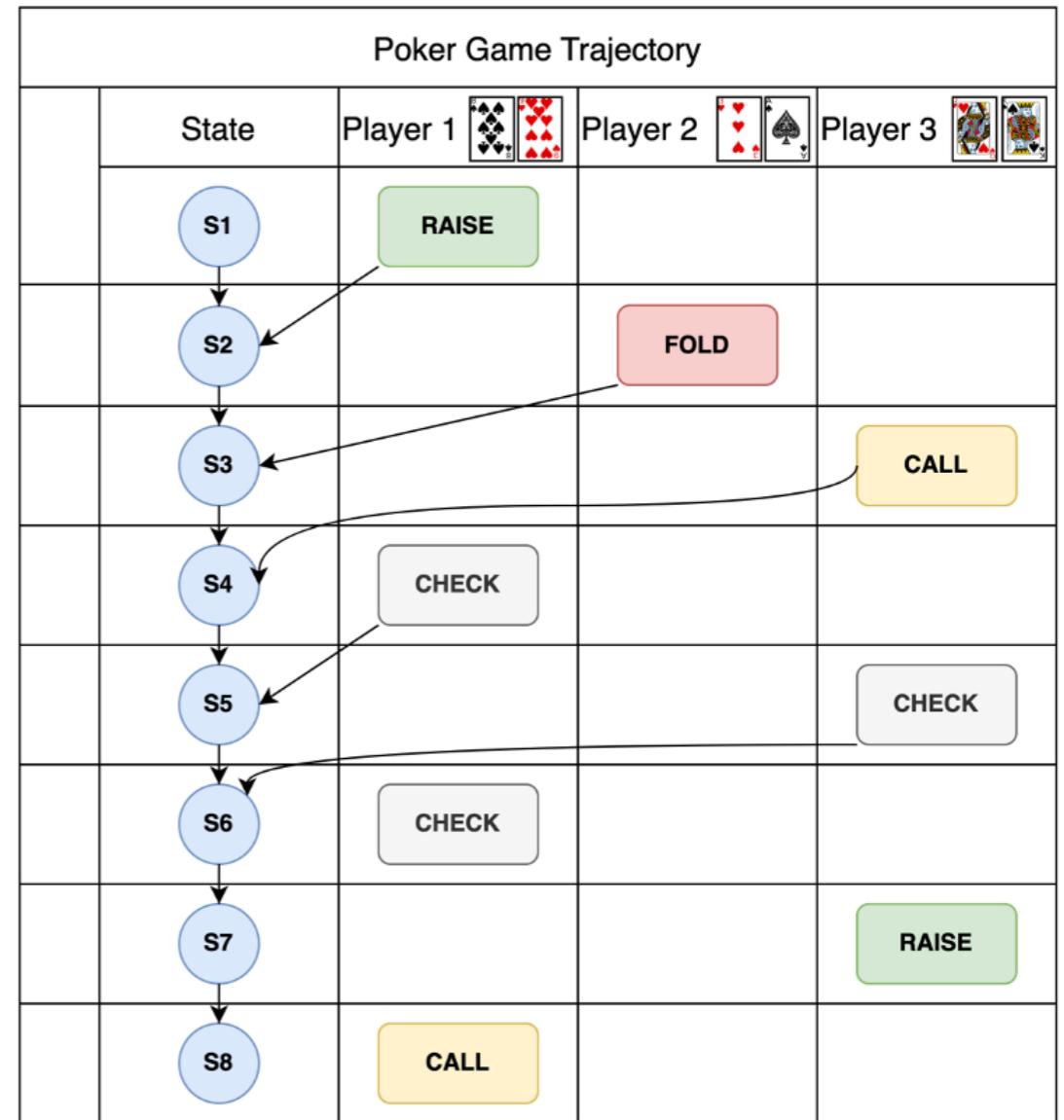
A popular skill and luck based gambling game

Modelling Poker in RL Setting



CALL RAISE CHECK FOLD

One Game of Poker



Winner gets all chips

Introduction

Problem Statement

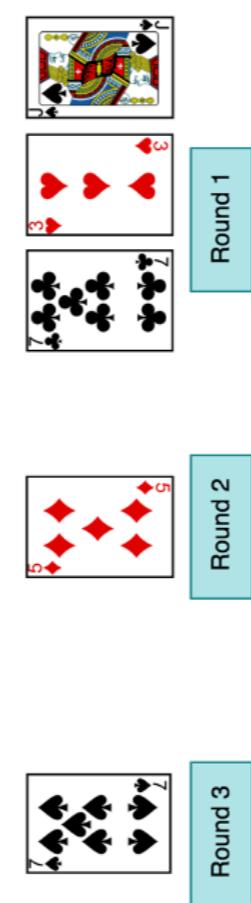
Texas Hold'em Poker



A popular skill and luck based gambling game

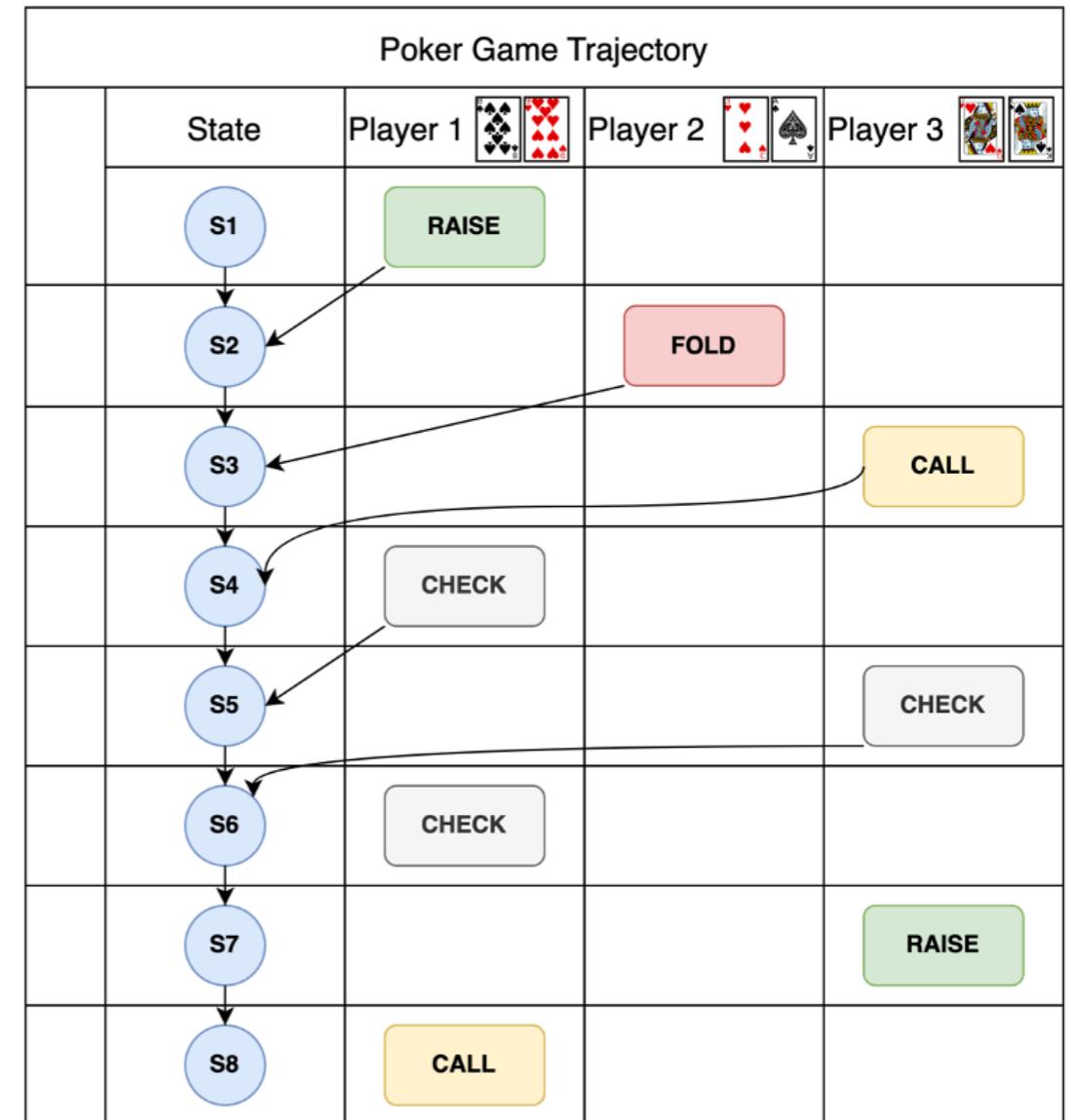
Modelling Poker in RL Setting

- Partially Observable



CALL RAISE CHECK FOLD

One Game of Poker



Introduction

Problem Statement

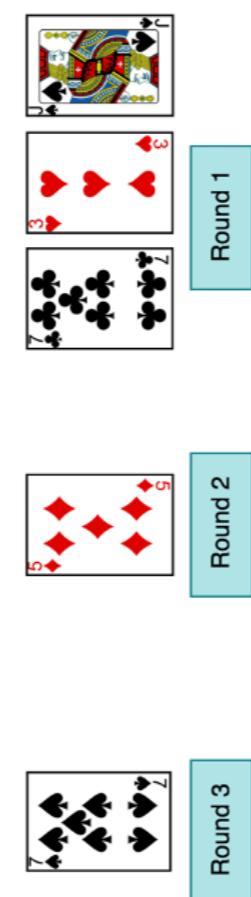
Texas Hold'em Poker



A popular skill and luck based gambling game

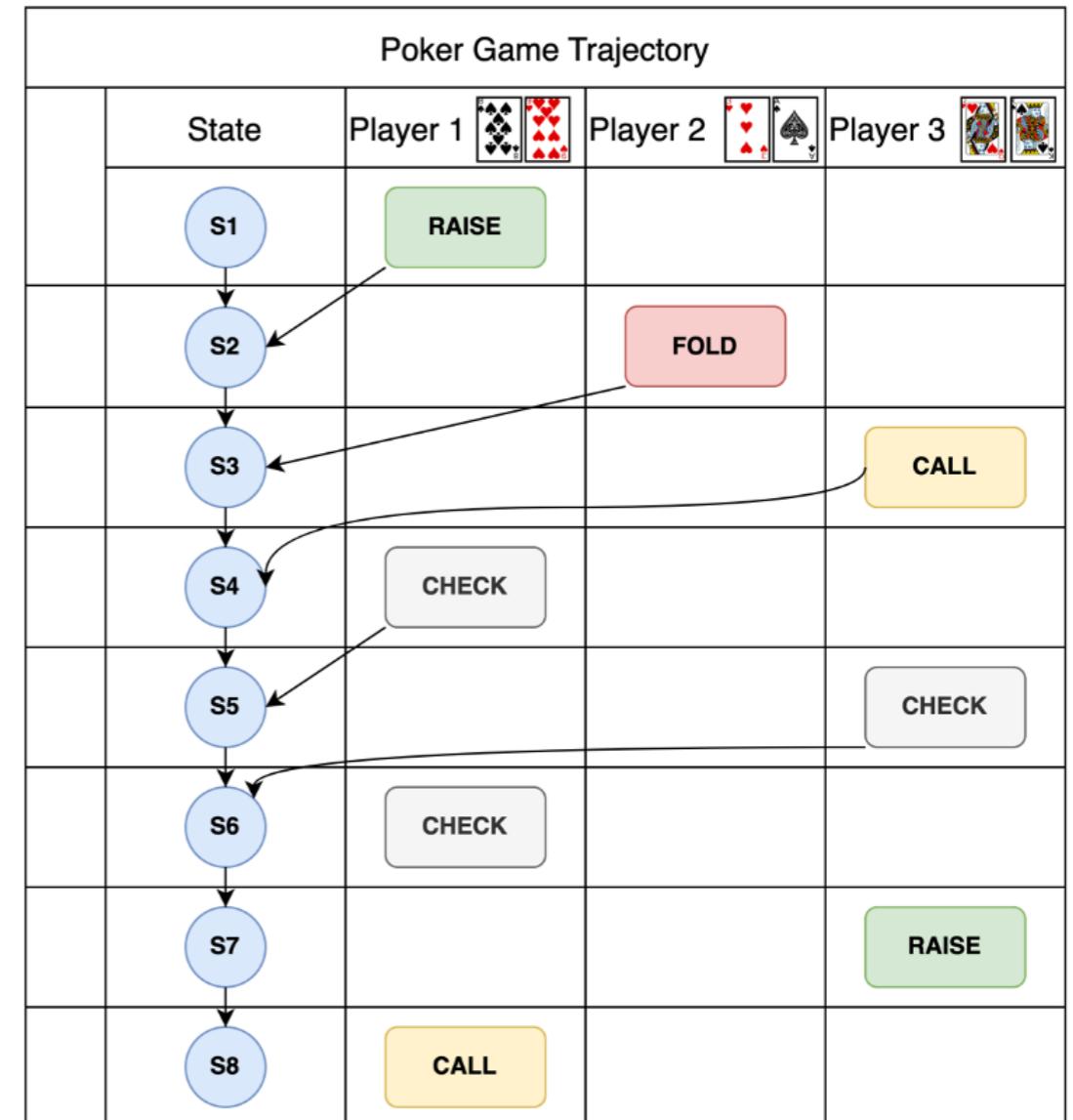
Modelling Poker in RL Setting

- Partially Observable
- Multi-Agent Zero-Sum



CALL RAISE CHECK FOLD

One Game of Poker



Winner gets all chips

Introduction

Problem Statement

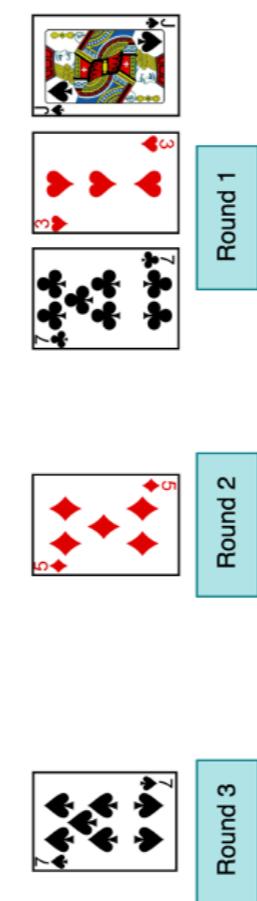
Texas Hold'em Poker



A popular skill and luck based gambling game

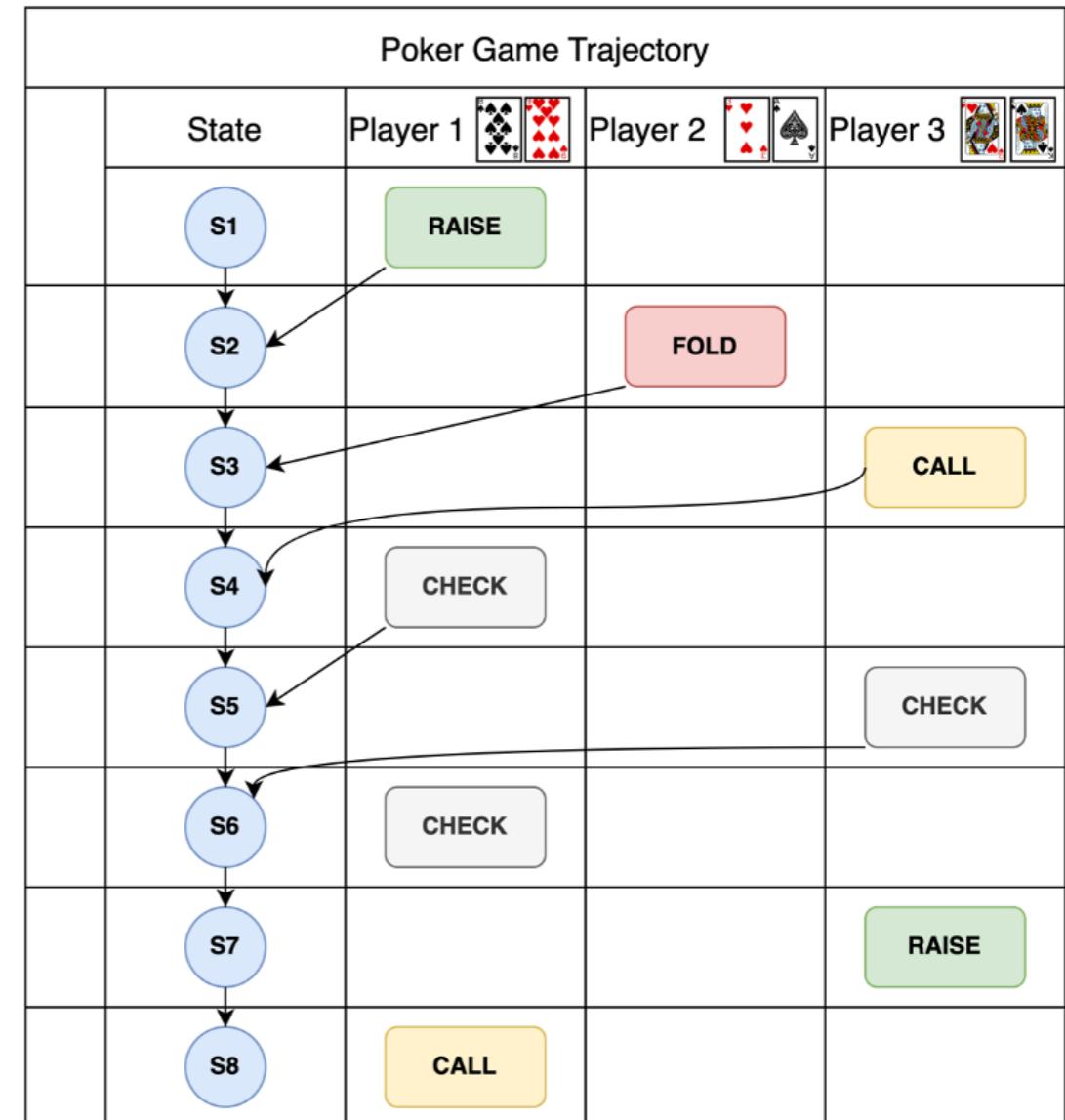
Modelling Poker in RL Setting

- Partially Observable
- Multi-Agent Zero-Sum
- Extensive Form game



CALL RAISE CHECK FOLD

One Game of Poker



Winner gets all chips

Introduction

Problem Statement

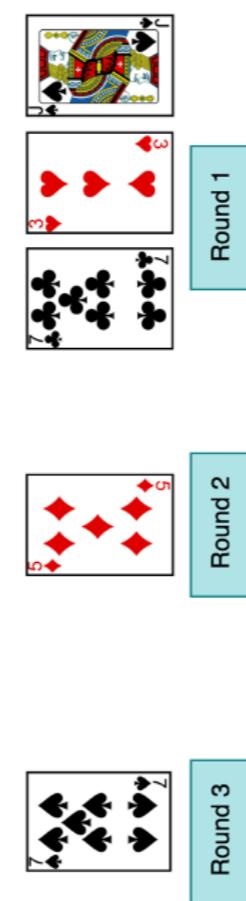
Texas Hold'em Poker



A popular skill and luck based gambling game

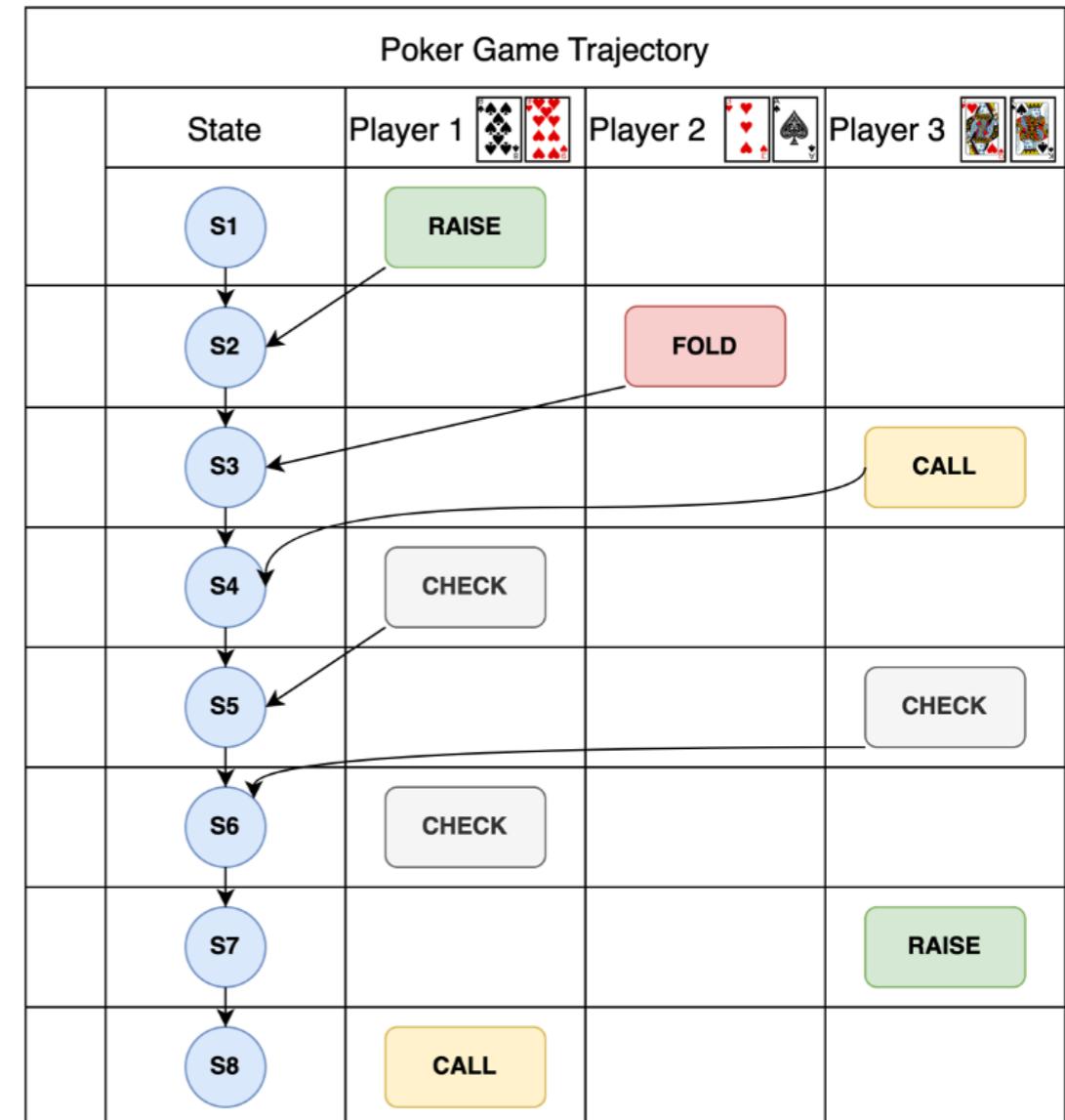
Modelling Poker in RL Setting

- Partially Observable
- Multi-Agent Zero-Sum
- Extensive Form game
- Intractable State-Space: 10^{14}



CALL RAISE CHECK FOLD

One Game of Poker



Winner gets all chips

Introduction

Objectives

Introduction

Objectives

- Train fully competitive agent to play poker.

Training Brain 1



Introduction

Objectives

- Train fully competitive agent to play poker.
- Train agents that learn to cheat via collaboration.

Training Brain 1



Training Brain 2

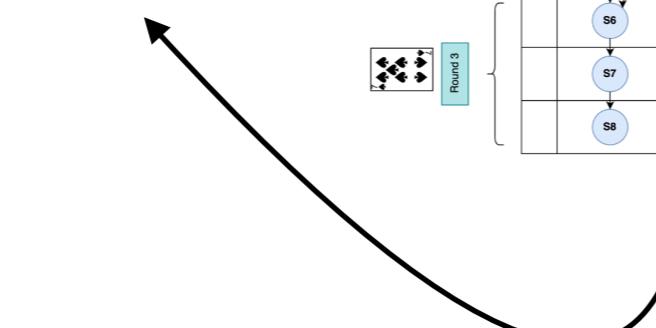


Introduction

Objectives

- Train fully competitive agent to play poker.
- Train agents that learn to cheat via collaboration.
- Create a classifier which can detect agents that are cheating.

$$p(\{B_1, B_2\} \mid \tau)$$



Poker Game Trajectory			
State	Player 1	Player 2	Player 3
S1	RAISE		
S2		FOLD	
S3			CALL
S4		CHECK	
S5			CHECK
S6		CHECK	
S7			RAISE
S8	CALL		

Training Brain 1



Training Brain 2



A group of blue, metallic, steampunk-style robots are gathered around a table, playing a board game. The robots have intricate mechanical details, glowing blue eyes, and various mechanical components visible on their bodies. They are positioned around a circular table covered with a light-colored cloth, which has several small, colorful pieces and a small bowl on it. The background is dark and smoky, suggesting a dimly lit room.

Brain 1: Training Fully Competitive Agents

Brain 1: Competitive Agents

Algorithm

Brain 1: Competitive Agents

Algorithm

Training Brain 1

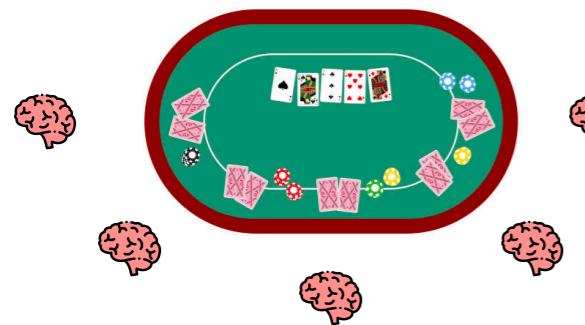


State Space

- Hand Cards
- Community Cards
- History of Raises

Brain 1: Competitive Agents Algorithm

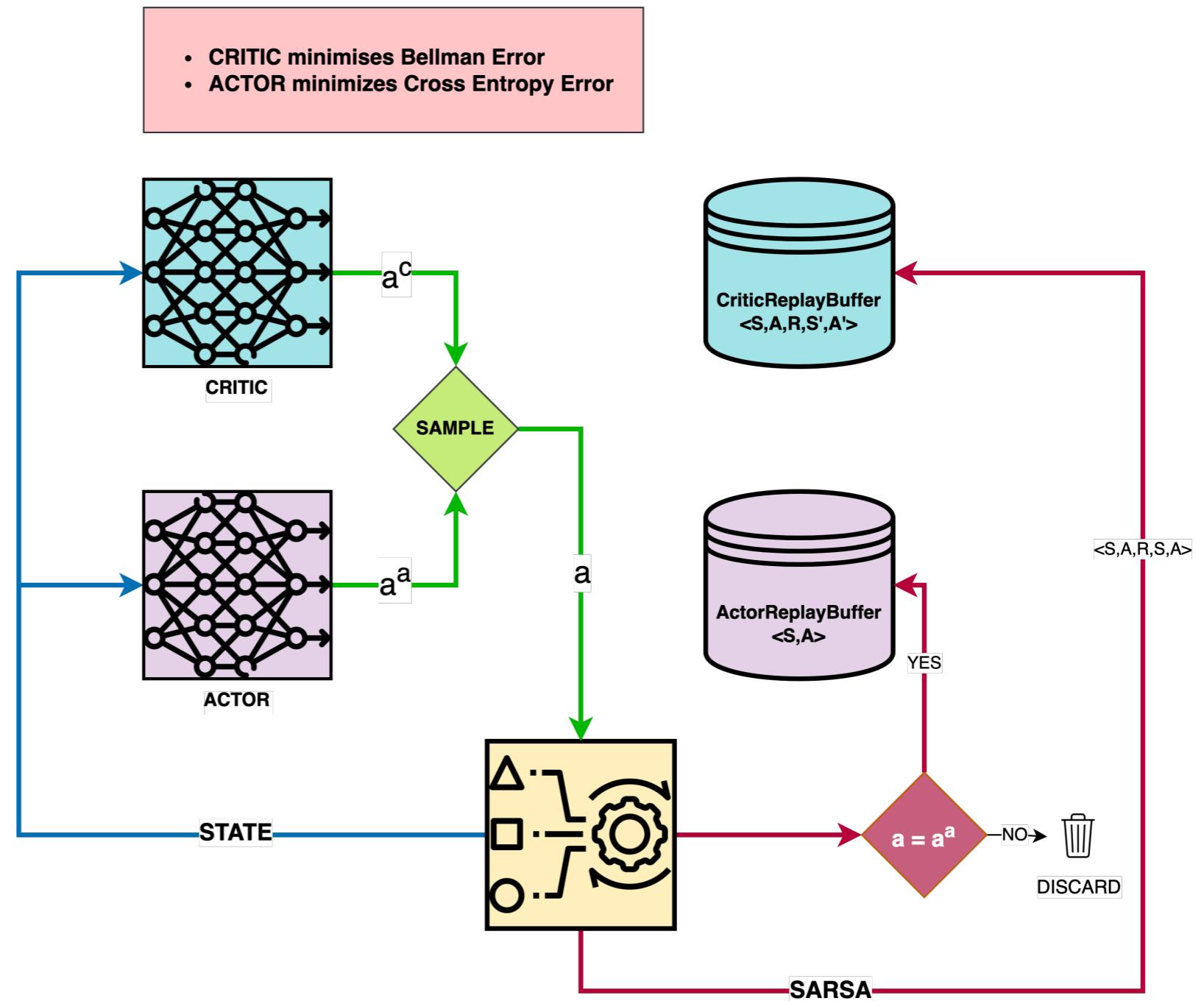
Training Brain 1



State Space

- Hand Cards
- Community Cards
- History of Raises

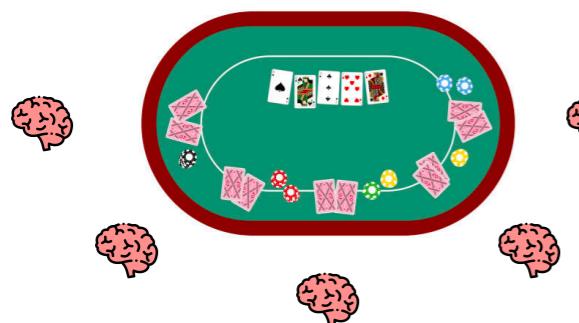
Neural Fictitious Self Play



Brain 1: Competitive Agents

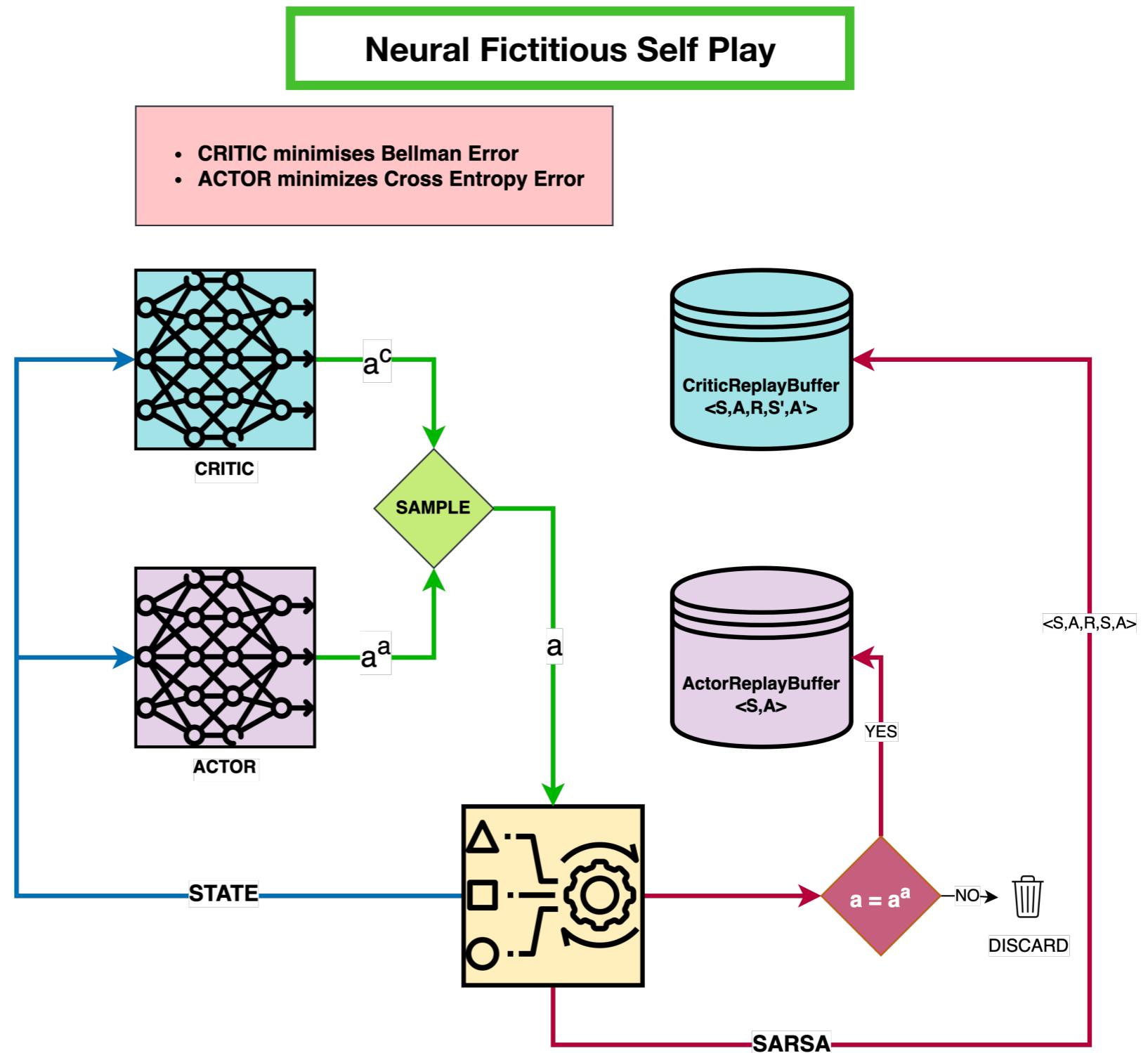
Algorithm

Training Brain 1



State Space

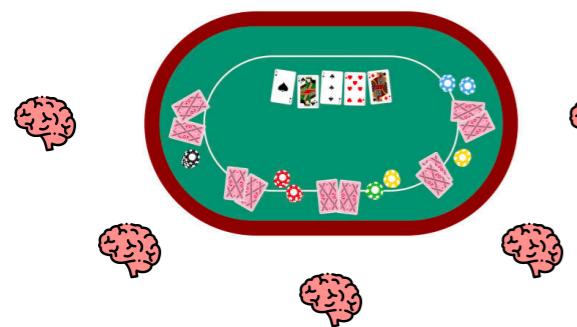
- Hand Cards
- Community Cards
- History of Raises



Brain 1: Competitive Agents

Algorithm

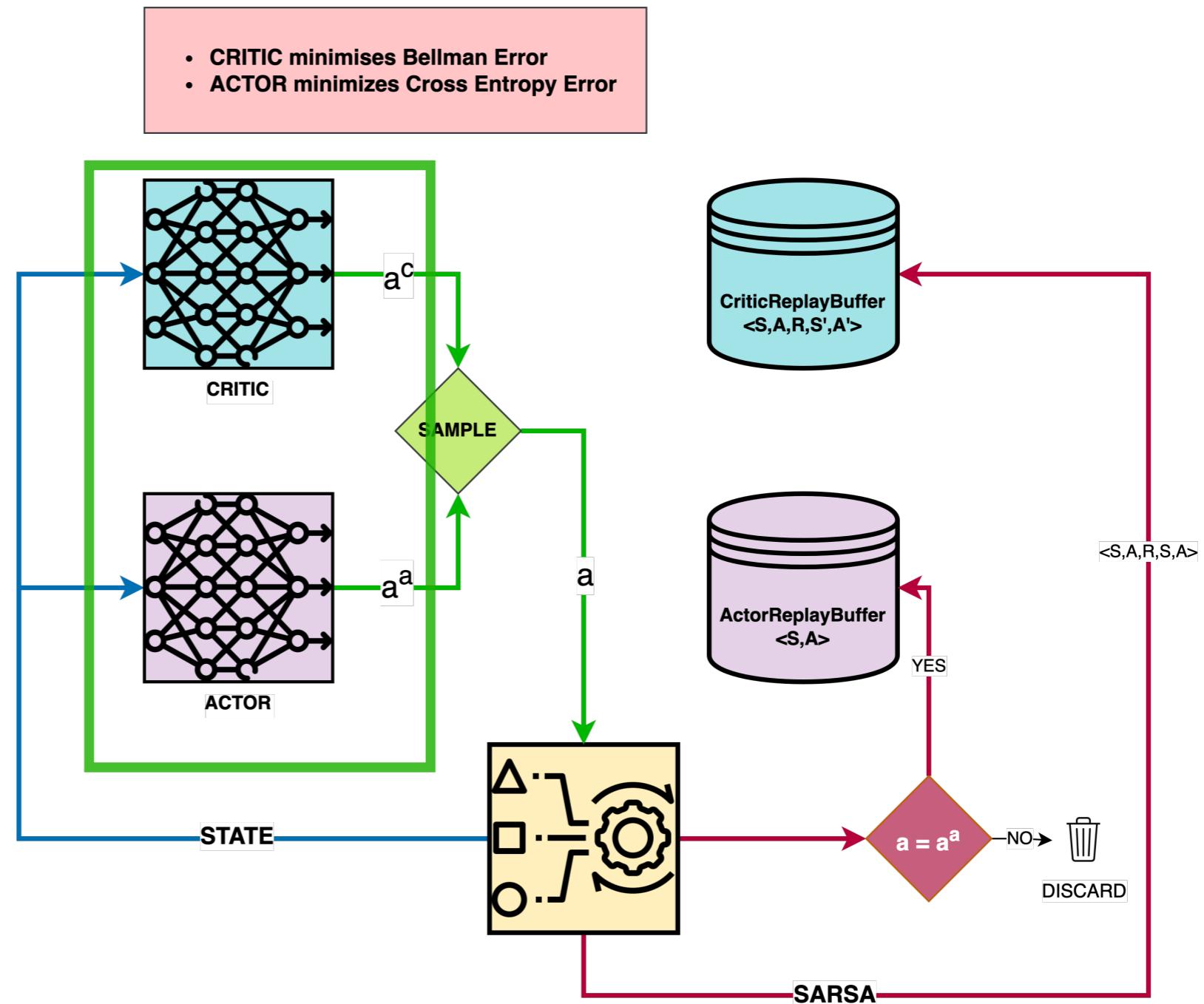
Training Brain 1



State Space

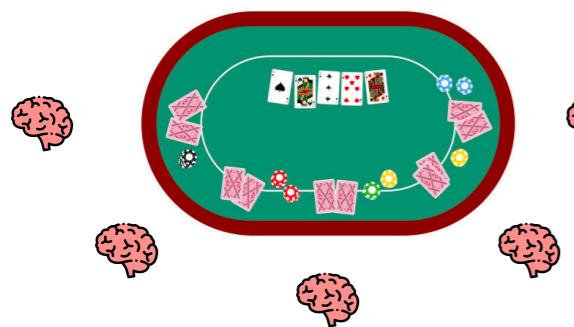
- Hand Cards
- Community Cards
- History of Raises

Neural Fictitious Self Play



Brain 1: Competitive Agents Algorithm

Training Brain 1



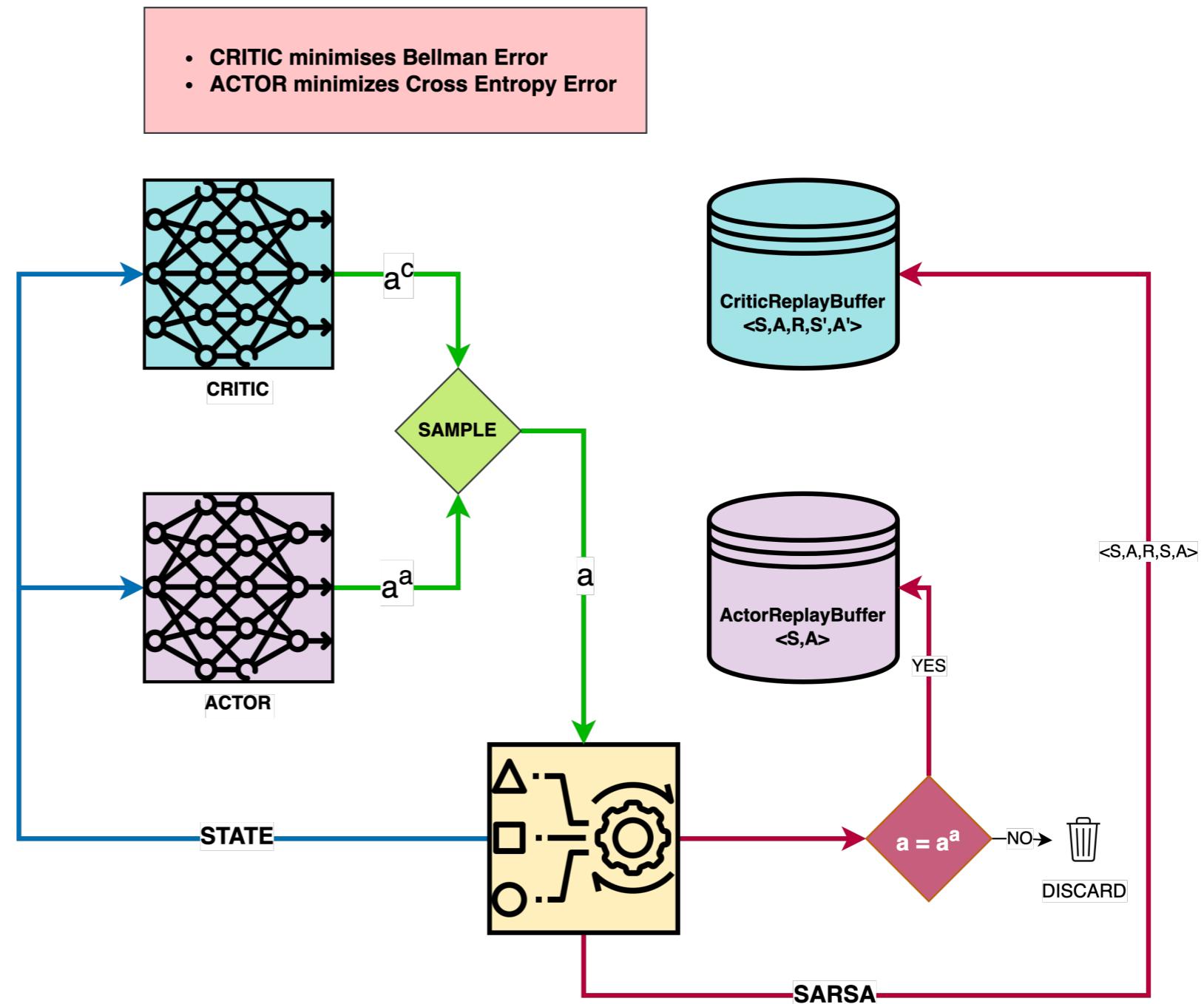
State Space

- Hand Cards
- Community Cards
- History of Raises

Evaluation

- NSFP consistently beats a Rule Based agent with win-ratio **3.65**

Neural Fictitious Self Play



A dark, atmospheric illustration featuring three humanoid figures with pale blue skin and red glowing eyes. They are wearing ornate, flowing robes in shades of red and blue. The figures are seated around a table, focused on a game of poker. The table is covered with a blue cloth and holds several playing cards and poker chips. The lighting is dramatic, casting deep shadows and highlighting the intricate details of their skin and robes.

Brain 2: Training Collaborative Agents

Brain 2: Collaborative Agents

Setup

Brain 2: Collaborative Agents

Setup

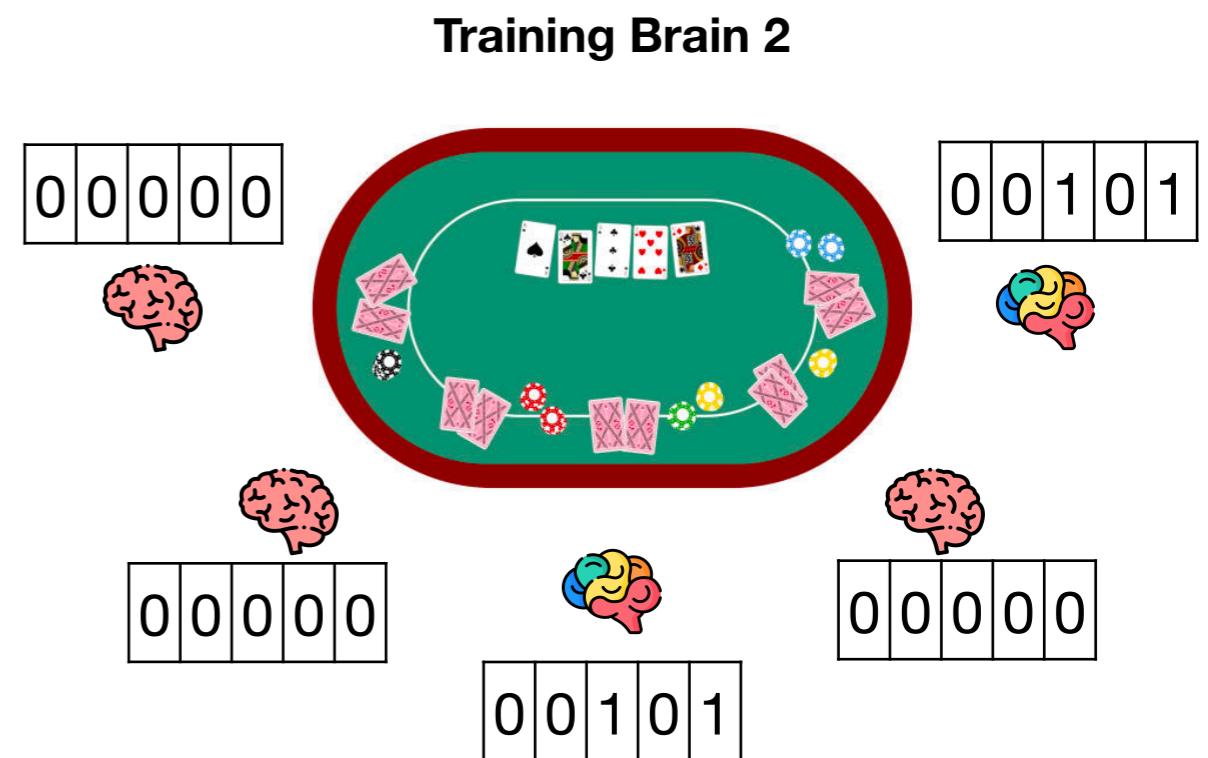
Training Brain 2



Brain 2: Collaborative Agents

Setup

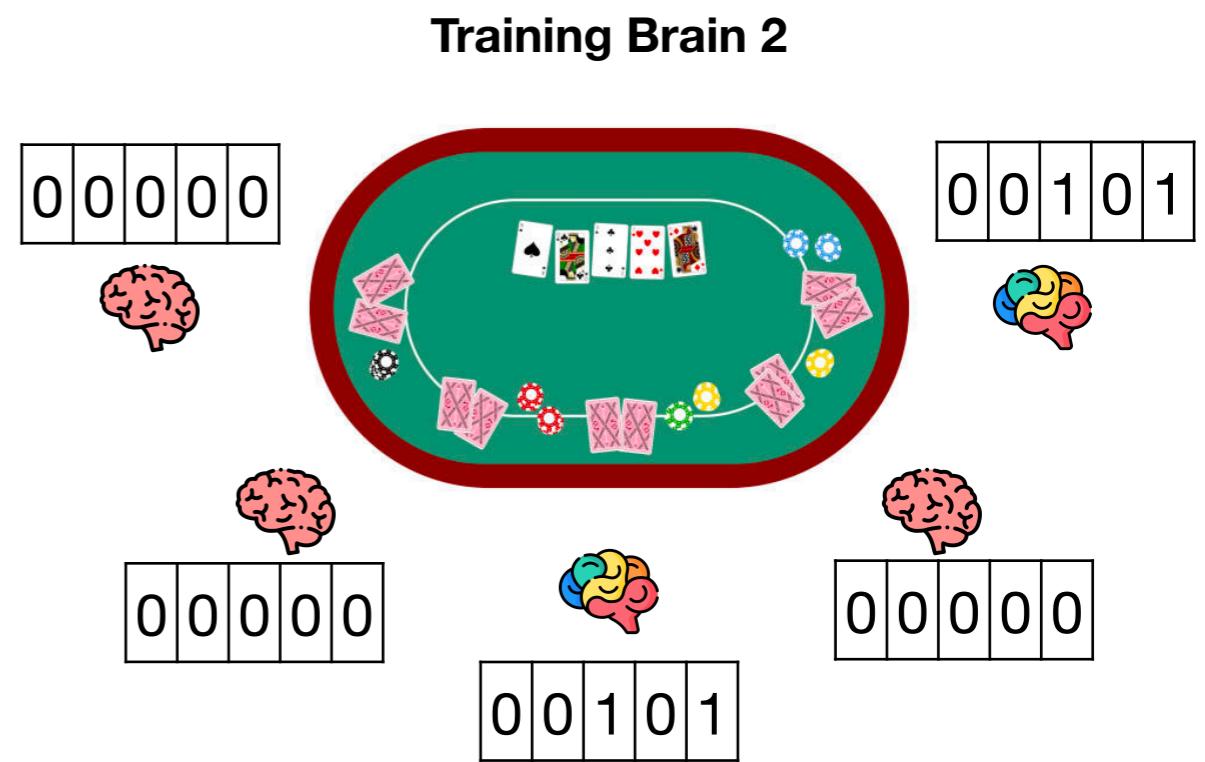
- Add context to the agents as to who their partner is by indicating which historical raises were made by the agent's partner



Brain 2: Collaborative Agents

Setup

- Add context to the agents as to who their partner is by indicating which historical raises were made by the agent's partner
- Update the reward function:
 - Either agent wins: **Max Winning**
 - Both agents lose: **Avg Loss**

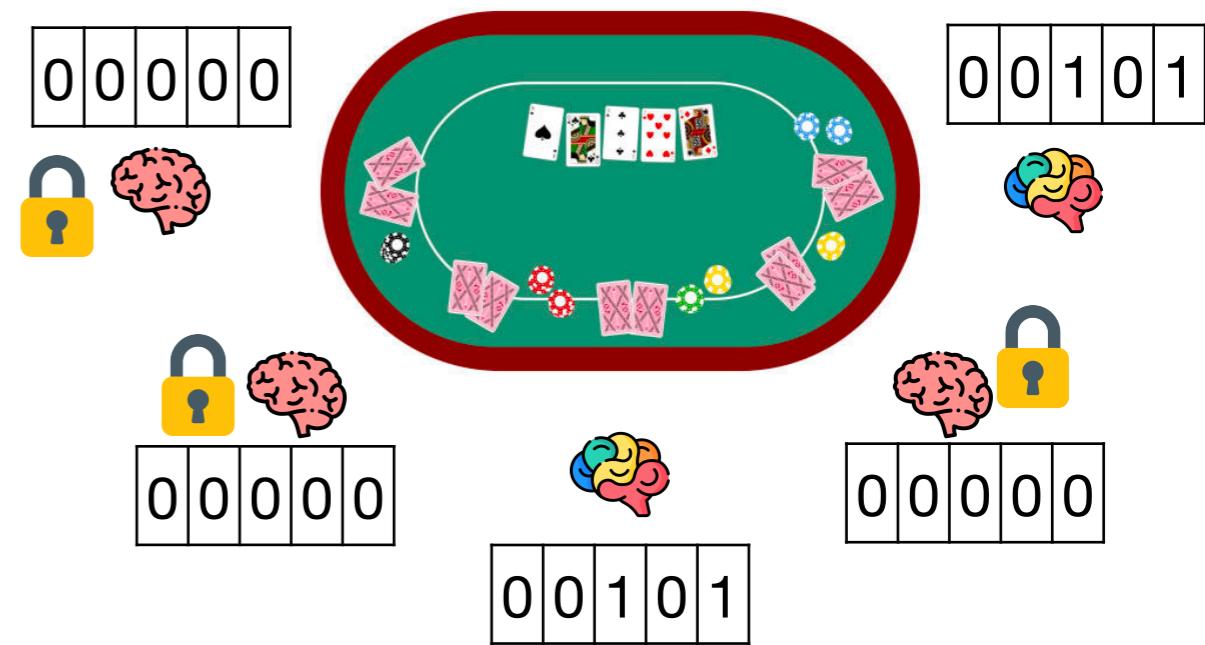


Brain 2: Collaborative Agents

Setup

- Add context to the agents as to who their partner is by indicating which historical raises were made by the agent's partner
- Update the reward function:
 - Either agent wins: **Max Winning**
 - Both agents lose: **Avg Loss**
- Freeze the competitive agents during training

Training Brain 2

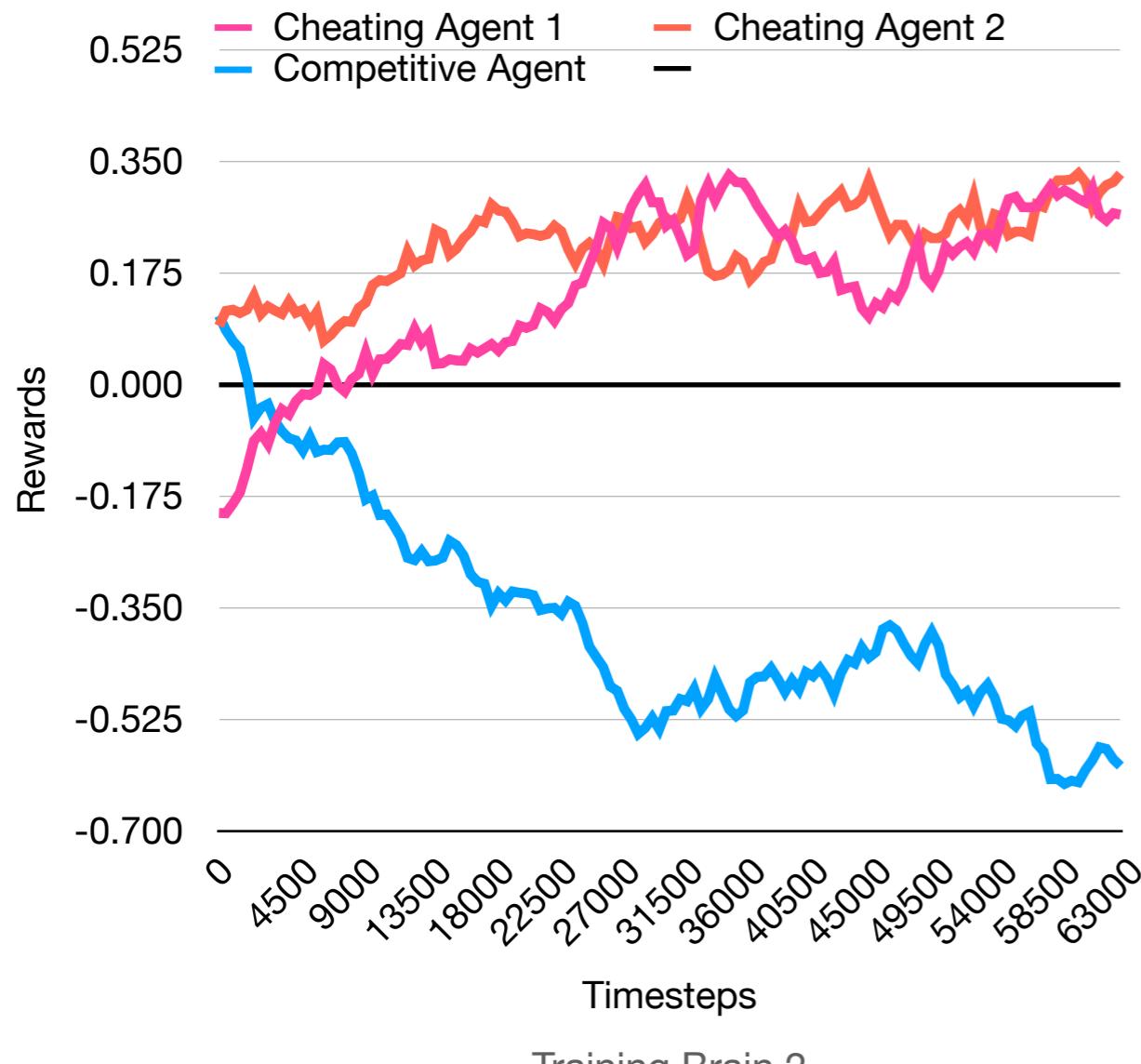


Brain 2: Collaborative Agents

Training and Results

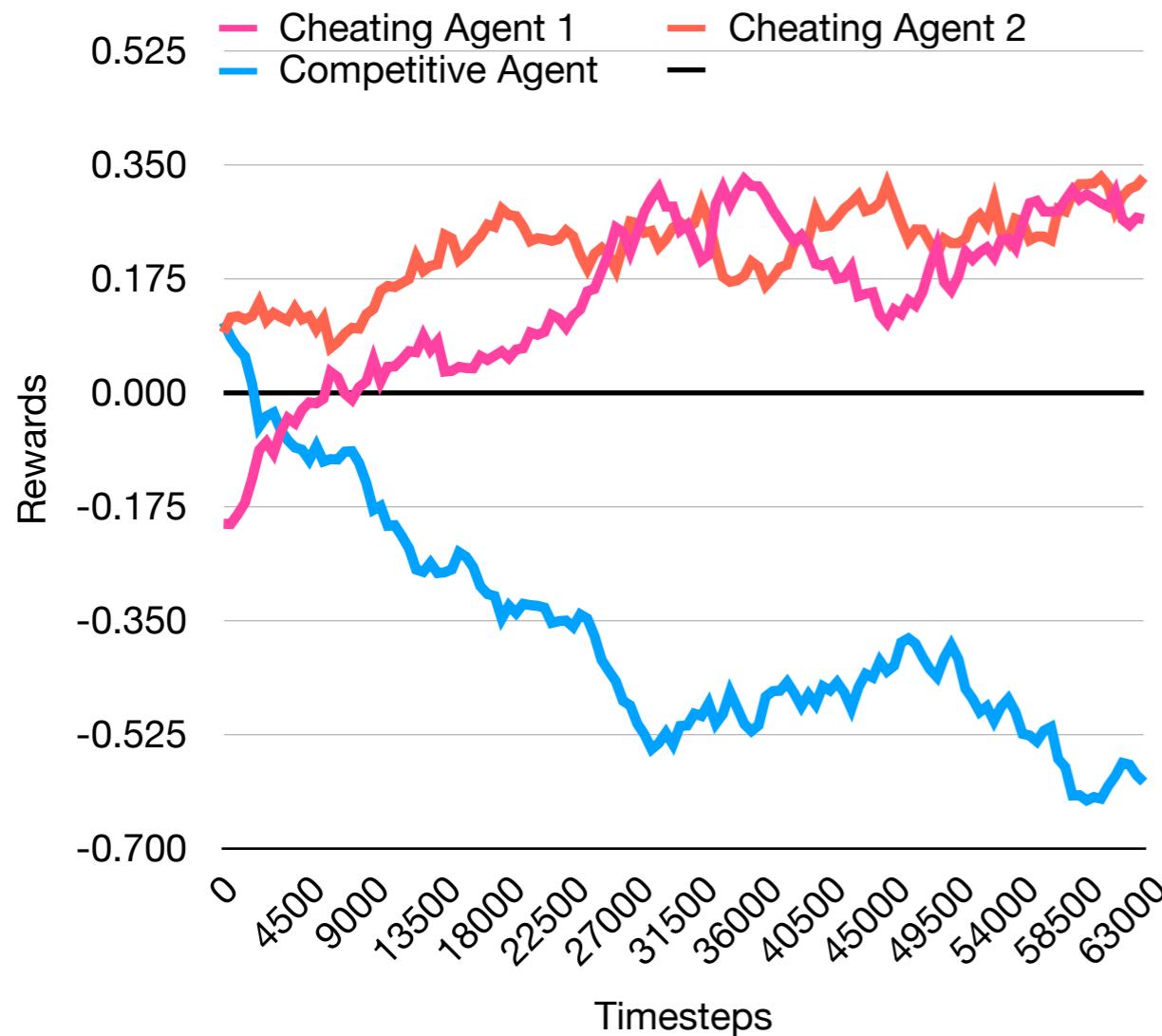
Brain 2: Collaborative Agents

Training and Results



Brain 2: Collaborative Agents

Training and Results

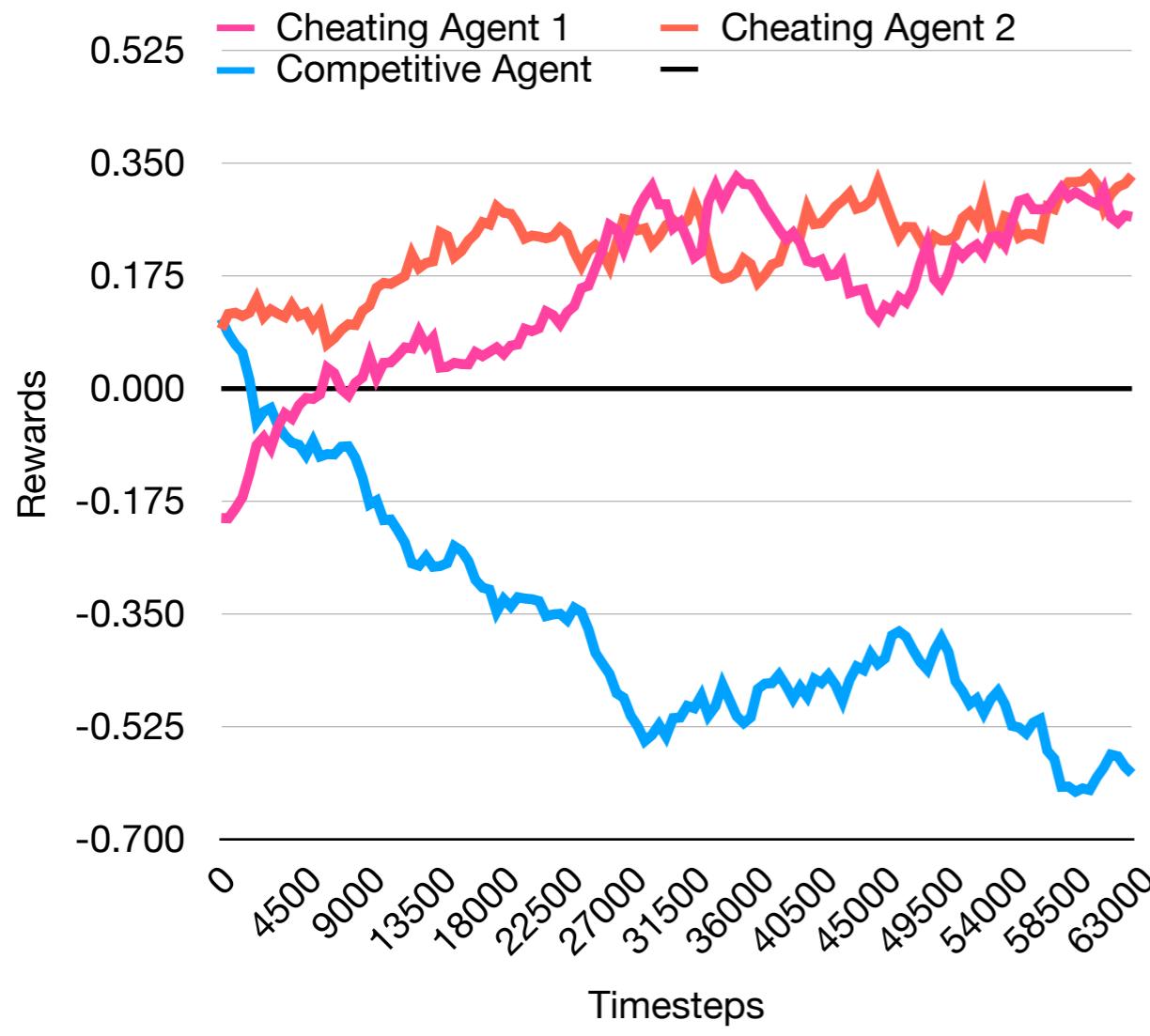


Training Brain 2

WinRatio(Cheater vs Fair) = 1.14

Brain 2: Collaborative Agents

Training and Results



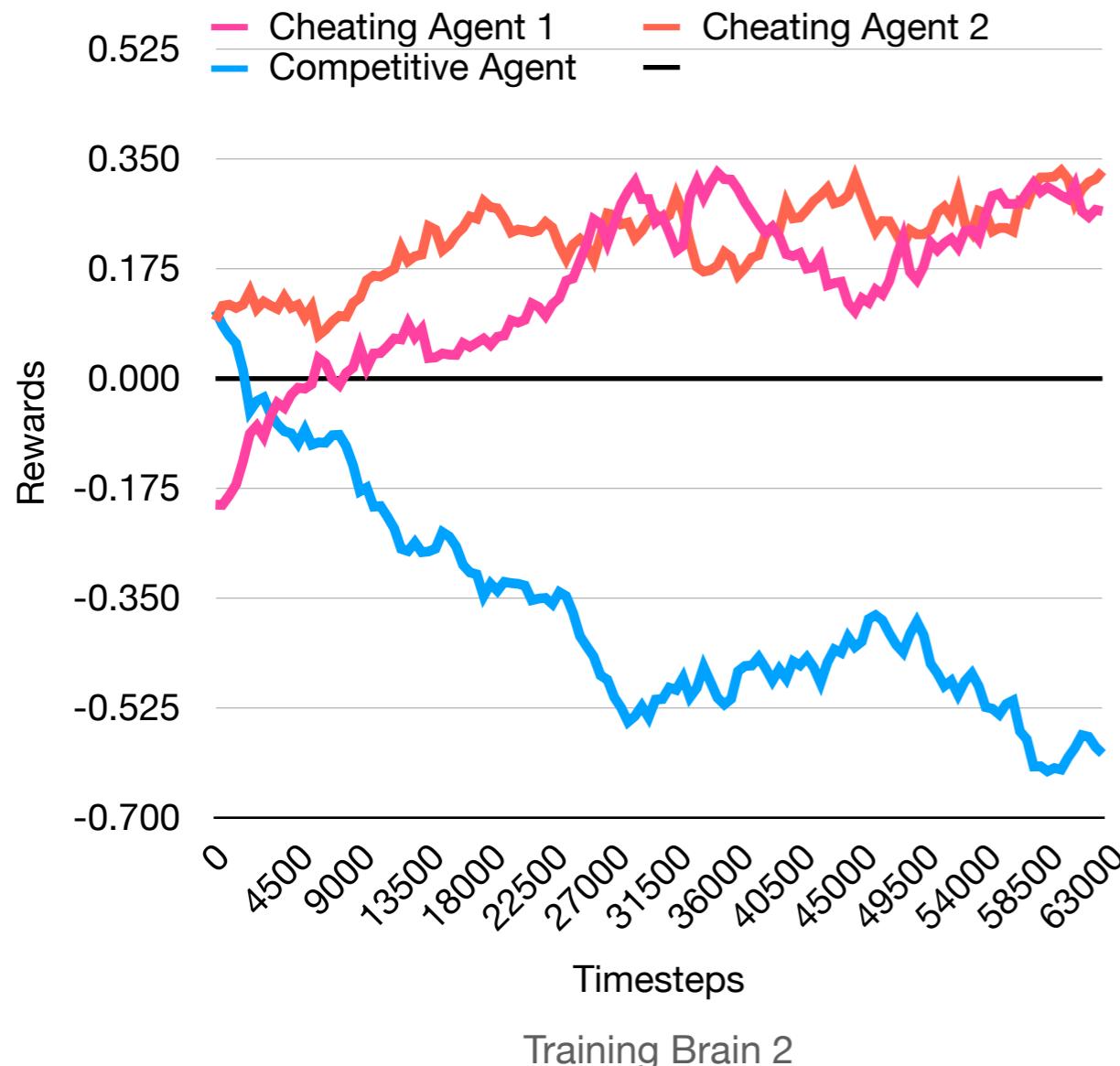
Training Brain 2

WinRatio(Cheater vs Fair) = 1.14

WinRatio(Cheater vs Fair(Same Epochs)) = 1.09

Brain 2: Collaborative Agents

Training and Results



WinRatio(Cheater vs Fair) = 1.14

WinRatio(Cheater vs Fair(Same Epochs)) = 1.09

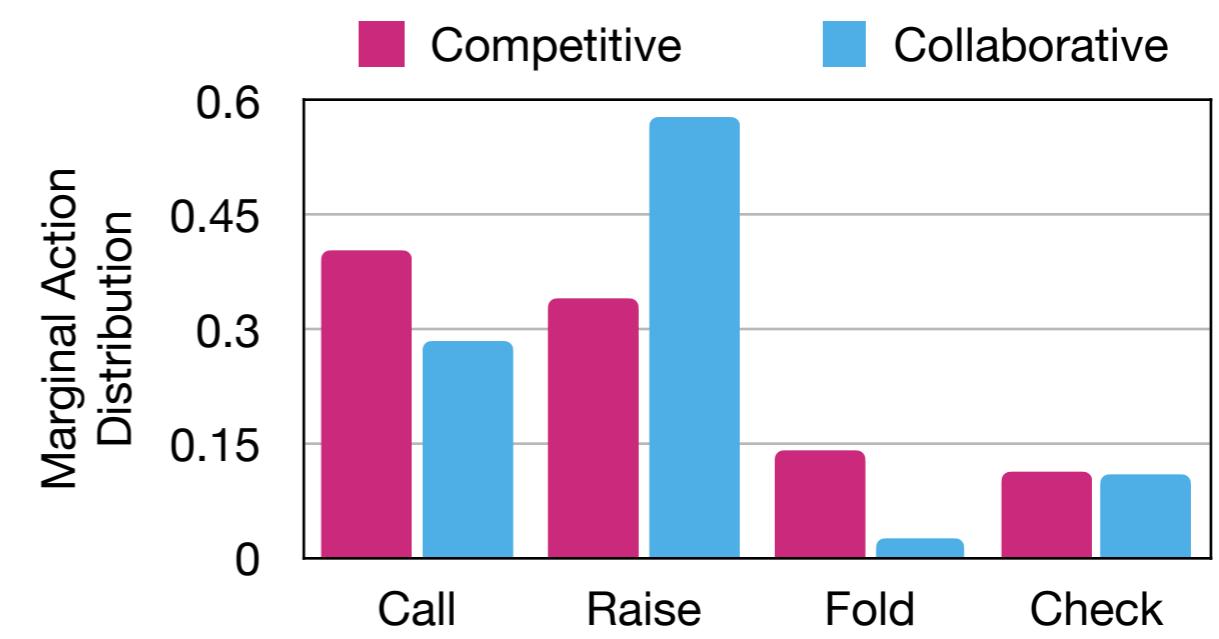
Action Distribution: $p(B_1, B_2)$

		call	raise	fold	check
Brain 1: Competitive	call	0.194	0.202	0.008	0.000
	raise	0.023	0.273	0.003	0.043
	fold	0.067	0.066	0.005	0.004
	check	0.000	0.039	0.010	0.063

Brain 2: Collaborative

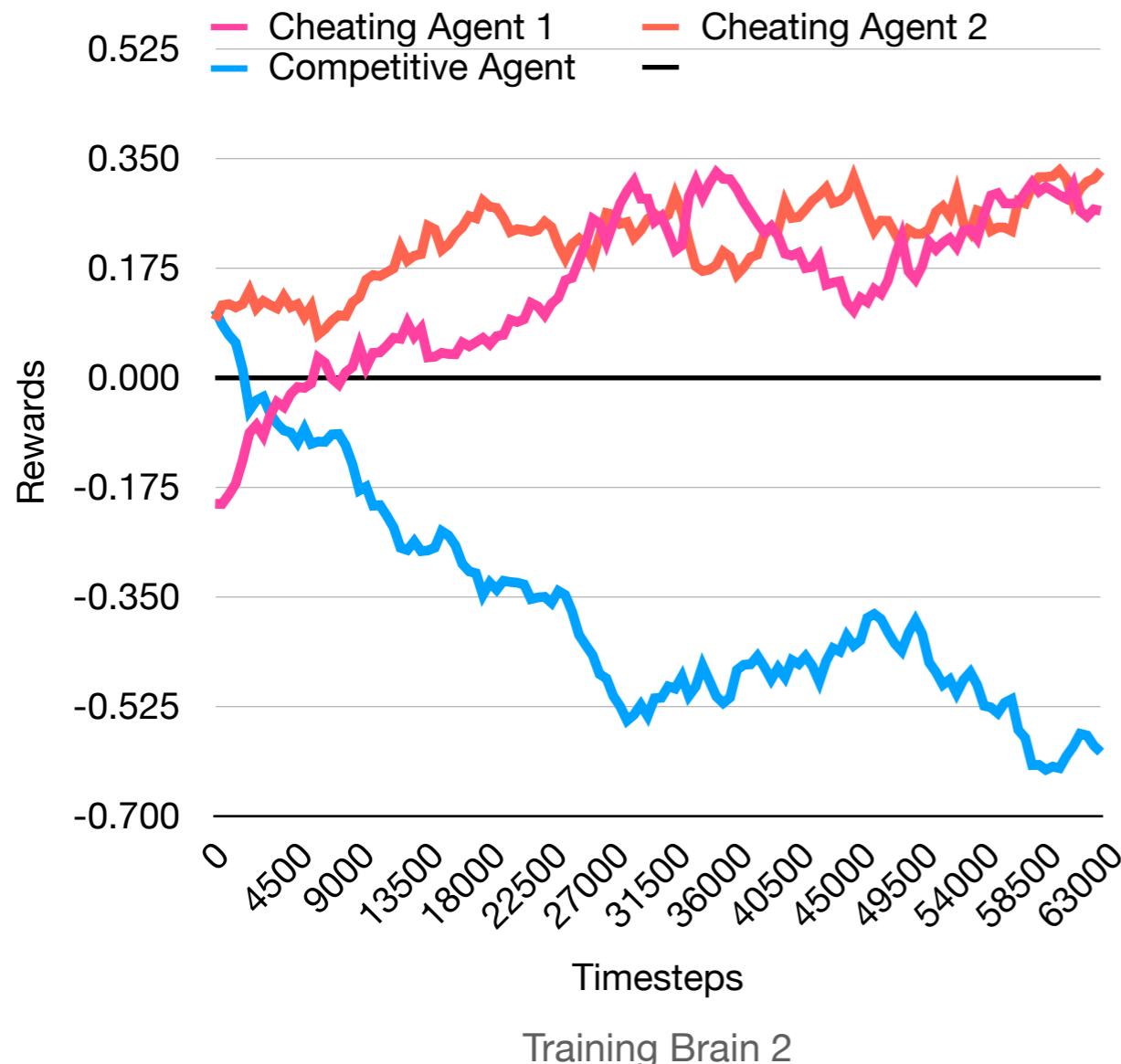
Probability of Action Similarity among the Brains

$$p(B_1 == B_2) = \frac{\sum A_{i,j}}{\sum A_{i,j}} = 0.535$$



Brain 2: Collaborative Agents

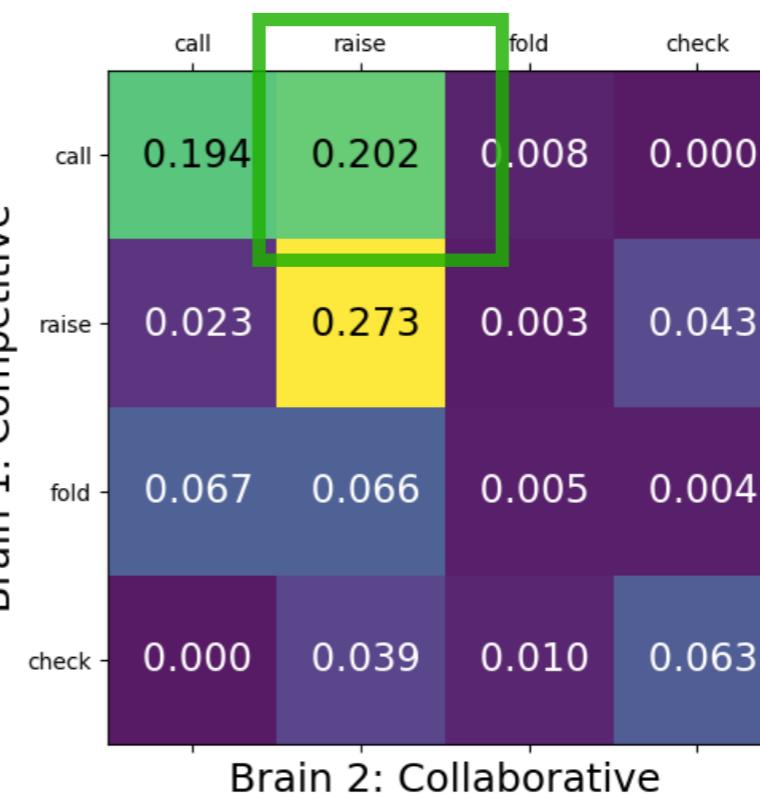
Training and Results



WinRatio(Cheater vs Fair) = 1.14

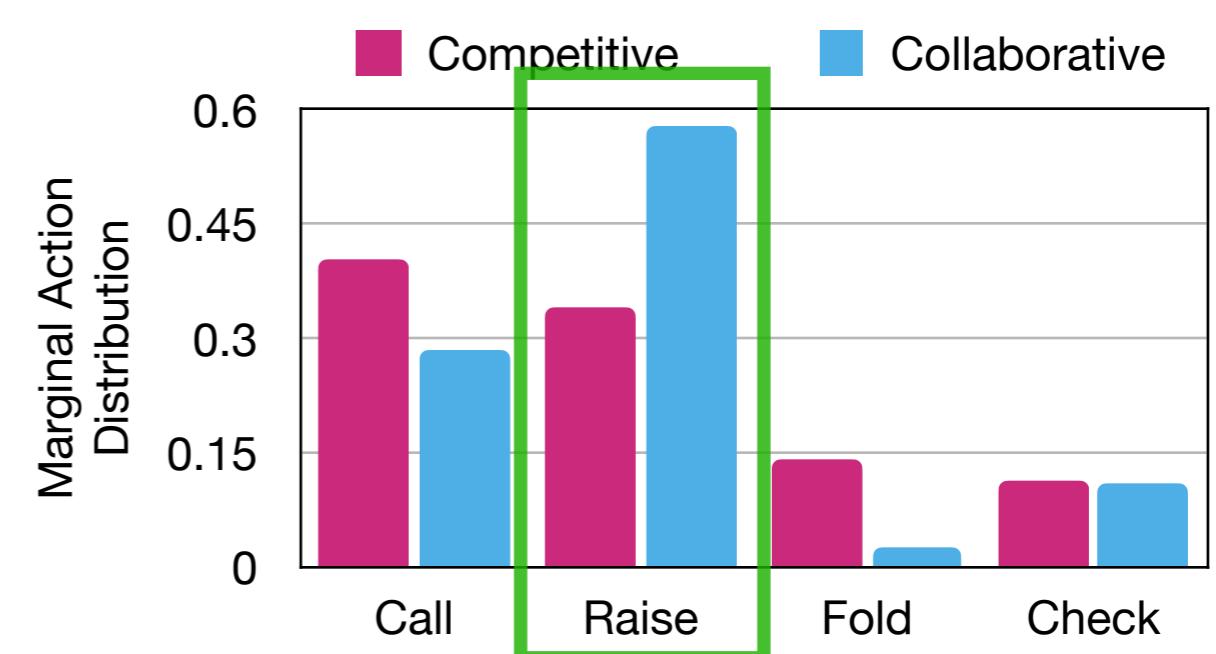
WinRatio(Cheater vs Fair(Same Epochs)) = 1.09

Action Distribution: $p(B_1, B_2)$



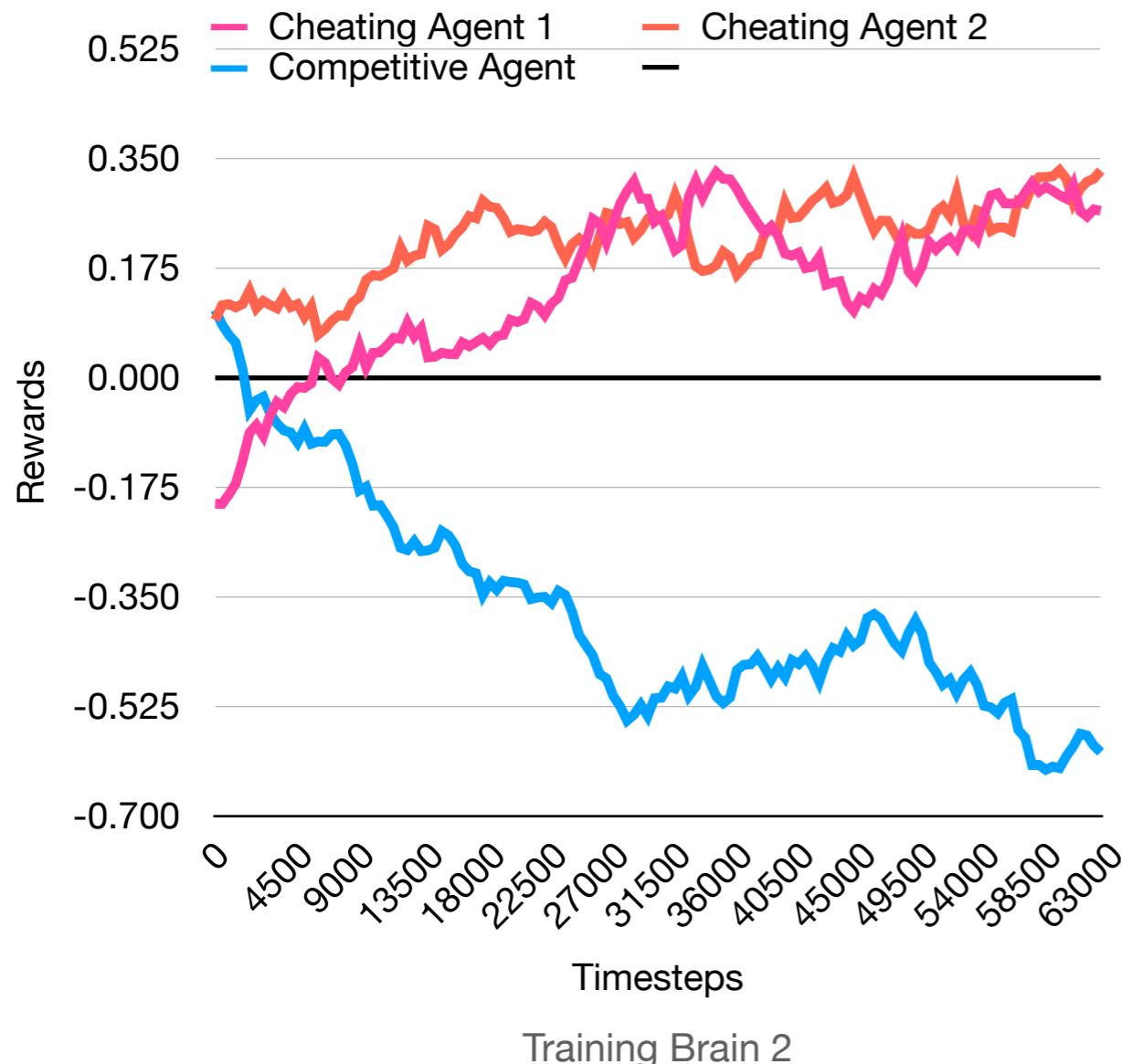
Probability of Action Similarity among the Brains

$$p(B_1 == B_2) = \frac{\sum_{i=j} A_{i,j}}{\sum A_{i,j}} = 0.535$$



Brain 2: Collaborative Agents

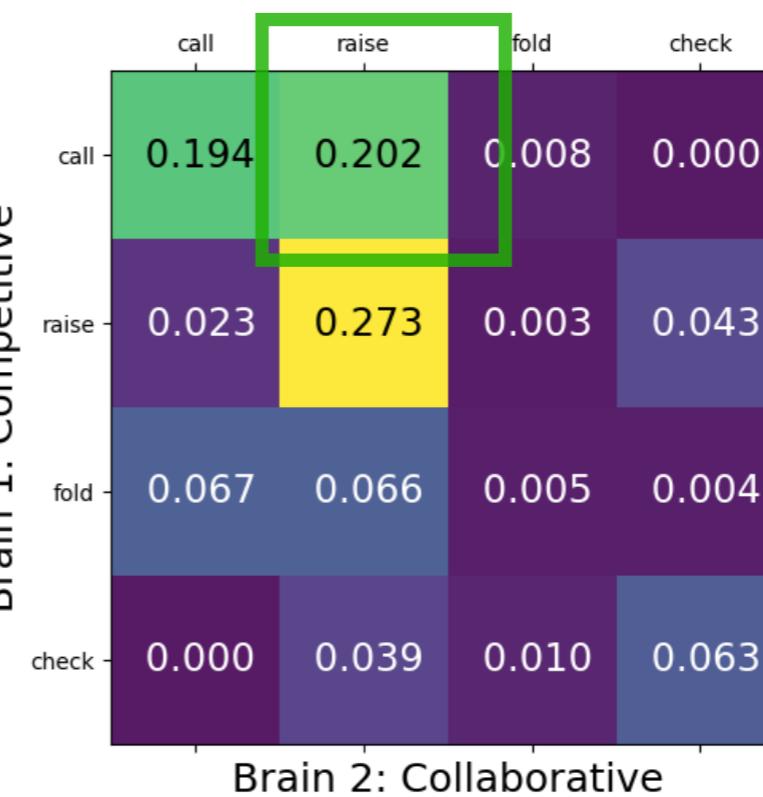
Training and Results



WinRatio(Cheater vs Fair) = 1.14

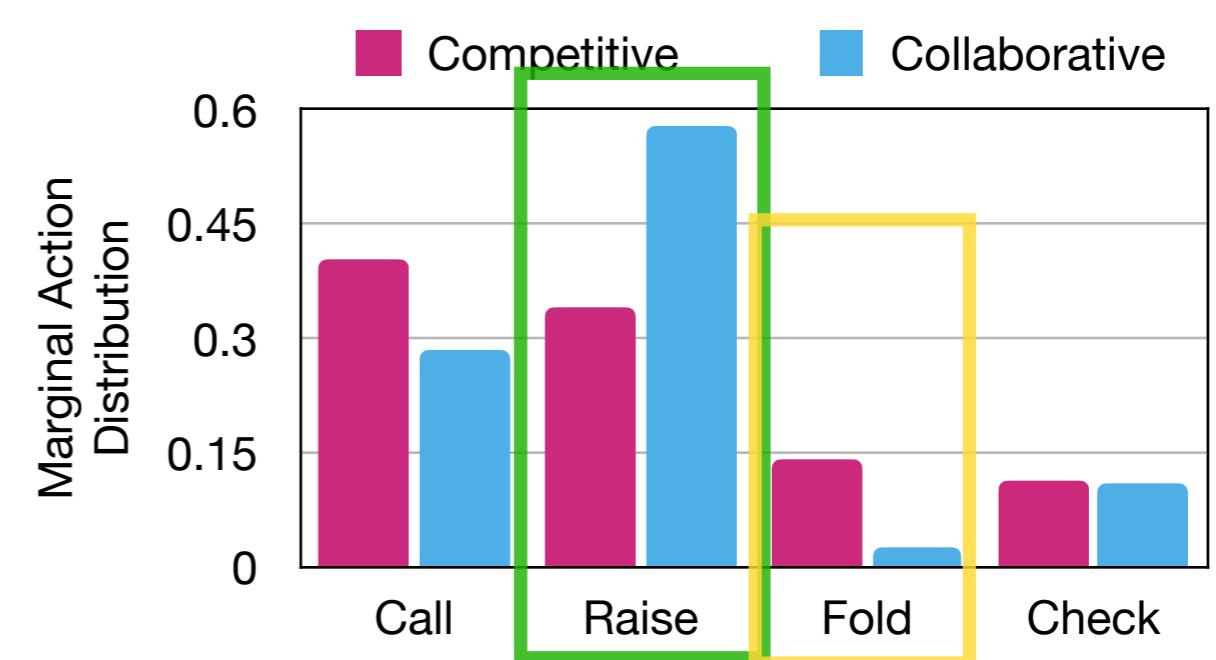
WinRatio(Cheater vs Fair(Same Epochs)) = 1.09

Action Distribution: $p(B_1, B_2)$



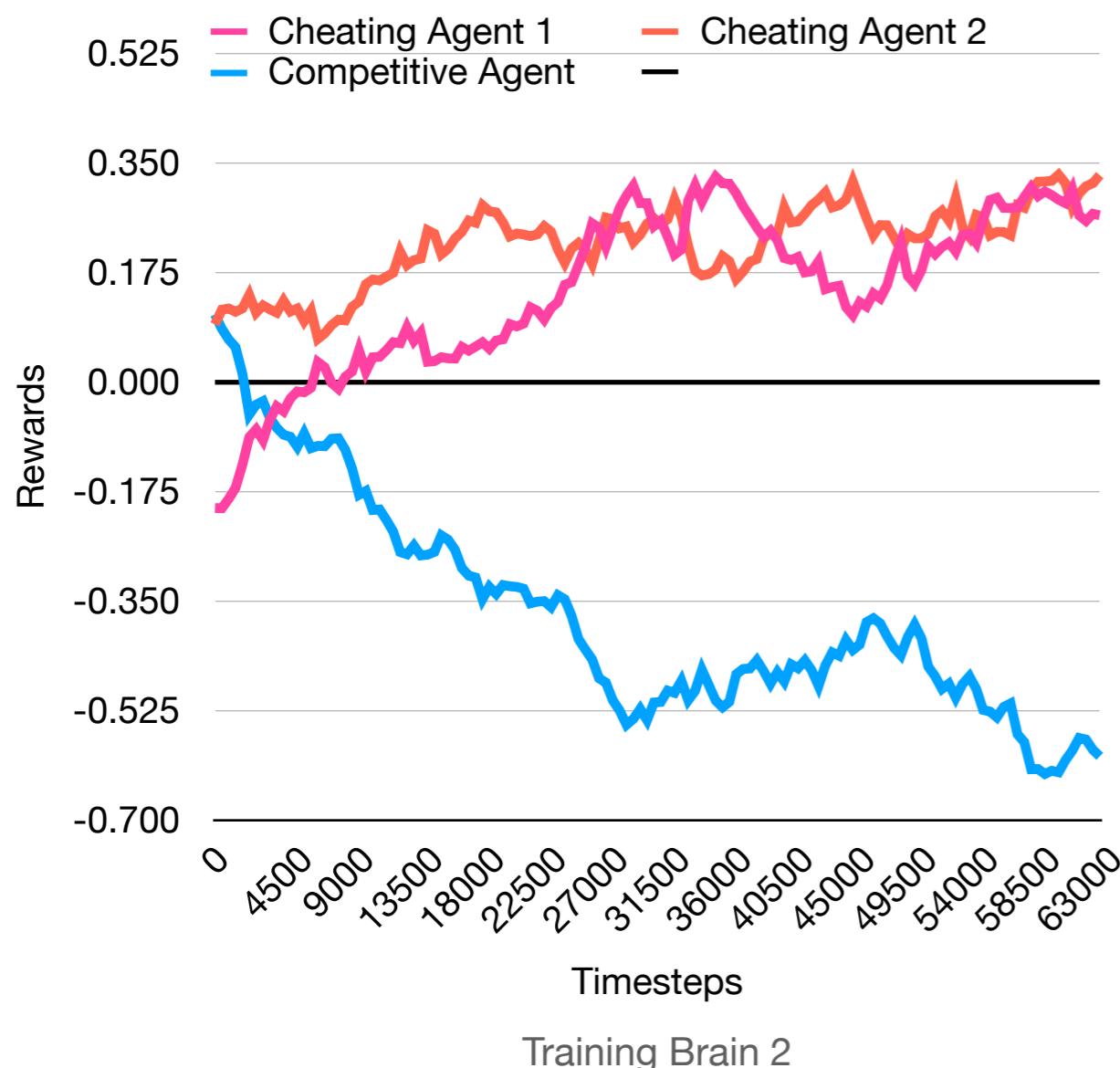
Probability of Action Similarity among the Brains

$$p(B_1 == B_2) = \frac{\sum A_{i,j}}{\sum A_{i,j}} = 0.535$$



Brain 2: Collaborative Agents

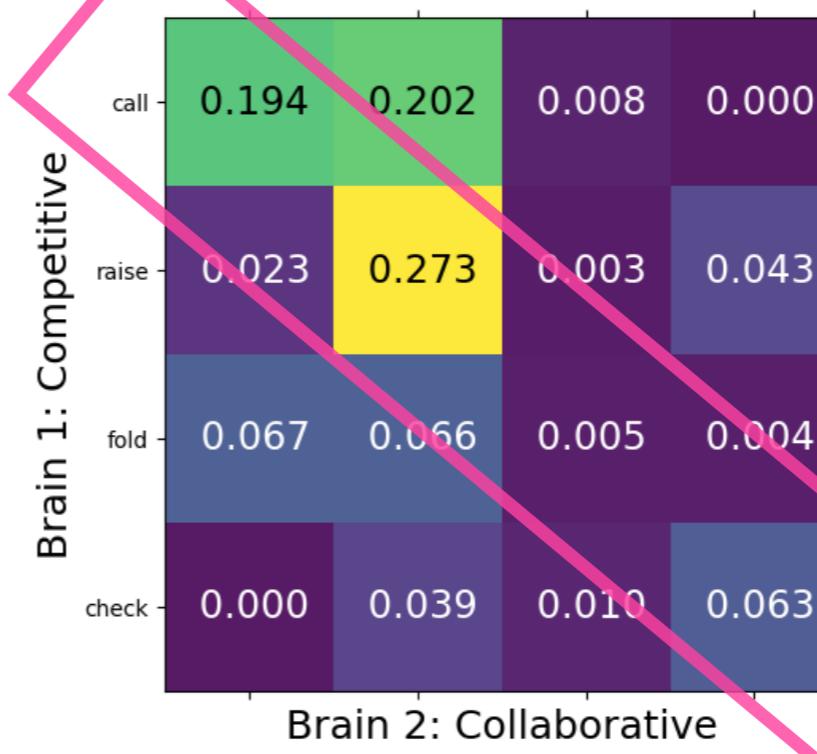
Training and Results



WinRatio(Cheater vs Fair) = 1.14

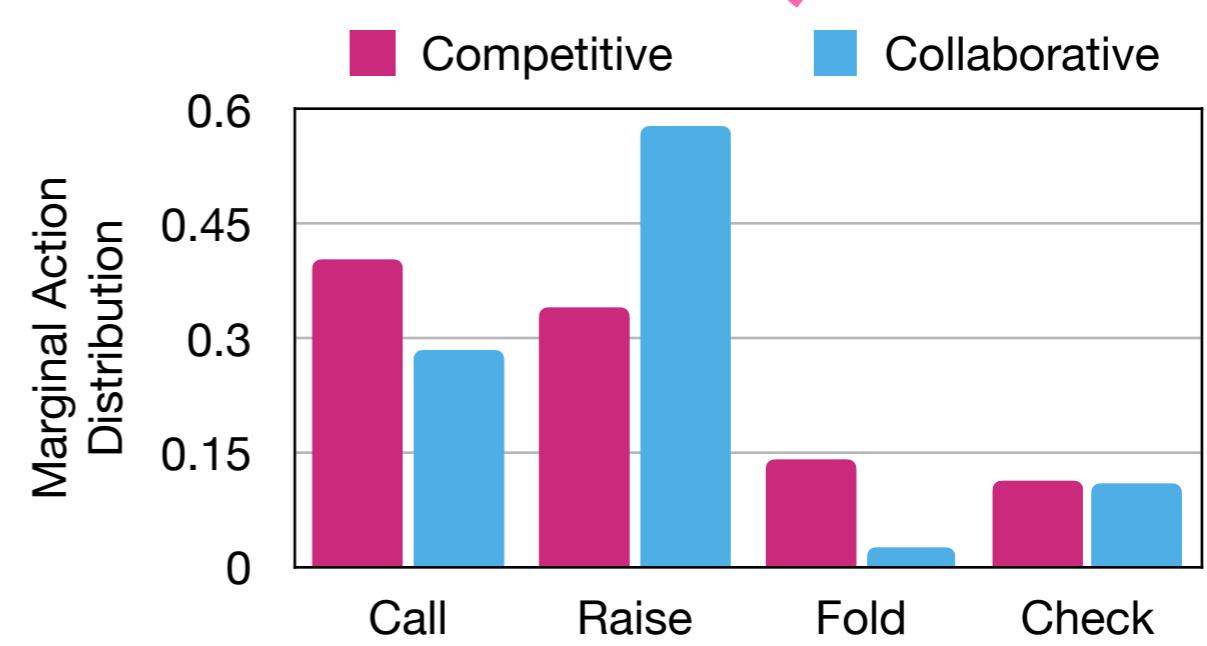
WinRatio(Cheater vs Fair(Same Epochs)) = 1.09

Action Distribution: $p(B_1, B_2)$



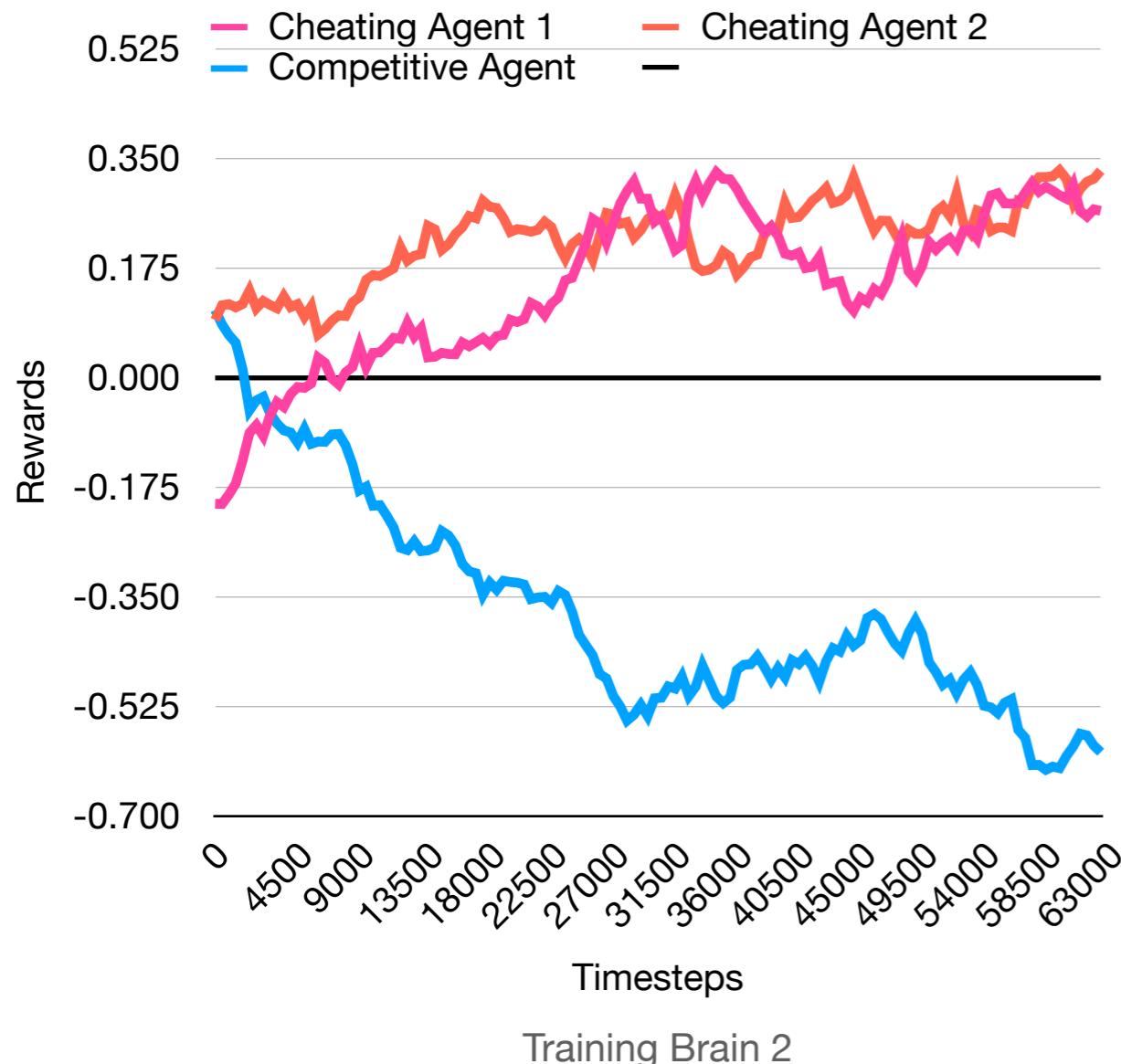
Probability of Action Similarity among the Brains

$$p(B_1 == B_2) = \frac{\sum A_{i,j}}{\sum A_{i,j}} = 0.535$$



Brain 2: Collaborative Agents

Training and Results



WinRatio(Cheater vs Fair) = 1.14

WinRatio(Cheater vs Fair(Same Epochs)) = 1.09

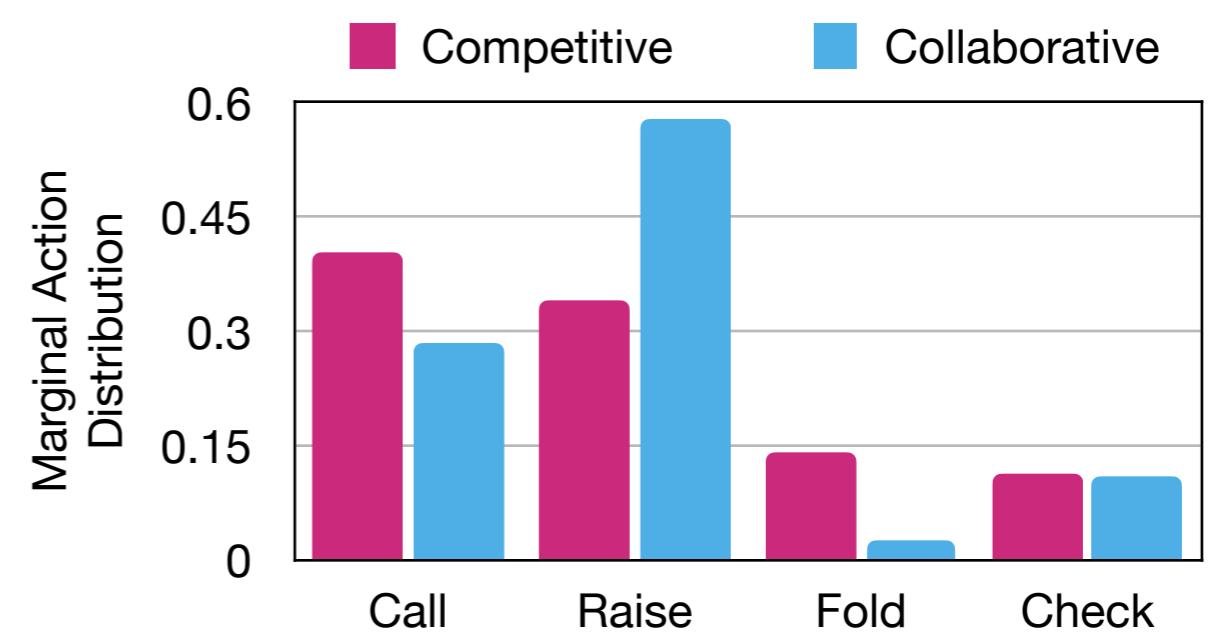
Action Distribution: $p(B_1, B_2)$

		call	raise	fold	check
Brain 1: Competitive	call	0.194	0.202	0.008	0.000
	raise	0.023	0.273	0.003	0.043
	fold	0.067	0.066	0.005	0.004
	check	0.000	0.039	0.010	0.063

Brain 2: Collaborative

Probability of Action Similarity among the Brains

$$p(B_1 == B_2) = \frac{\sum A_{i,j}}{\sum A_{i,j}} = 0.535$$



A detailed illustration of two robots playing cards at a table. On the left, a red and silver robot with a large circular head and a small antenna on its forehead is looking down at the cards it's holding. On the right, a blue and silver robot with a more complex, segmented head and several small glowing blue lights on its forehead is also looking at its cards. They are seated around a round wooden table covered with a green cloth, which has several playing cards and some coins on it. In the background, there's a dark room with a painting of red flowers on the wall and a lamp hanging from the ceiling. The overall atmosphere is mysterious and focused.

Discriminator:
Detecting who is
cheating

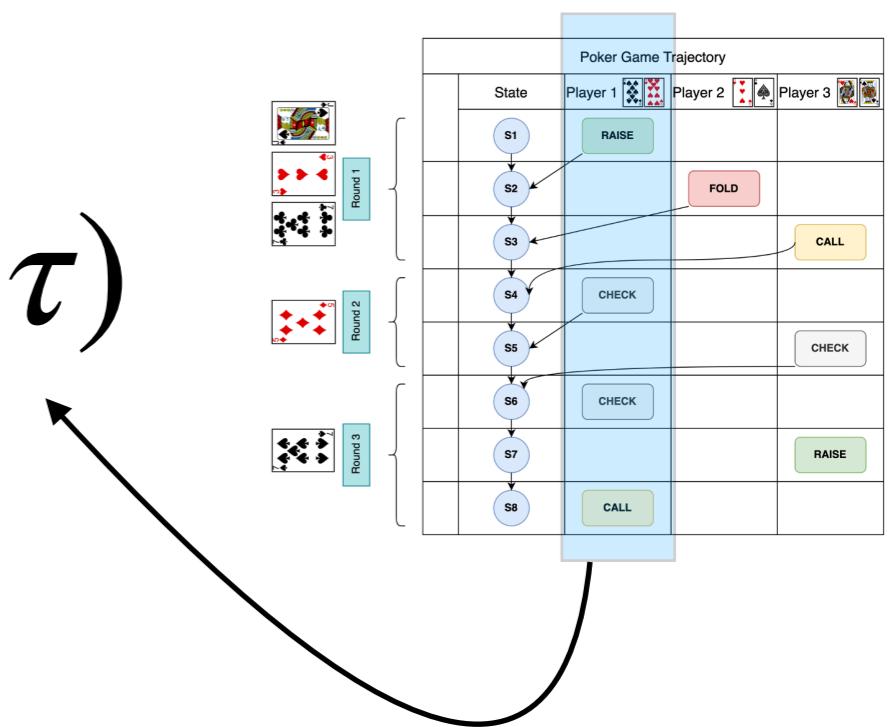
Discriminator

Setup and Challenges

Discriminator

Setup and Challenges

$$p(\{B_1, B_2\} \mid \tau)$$

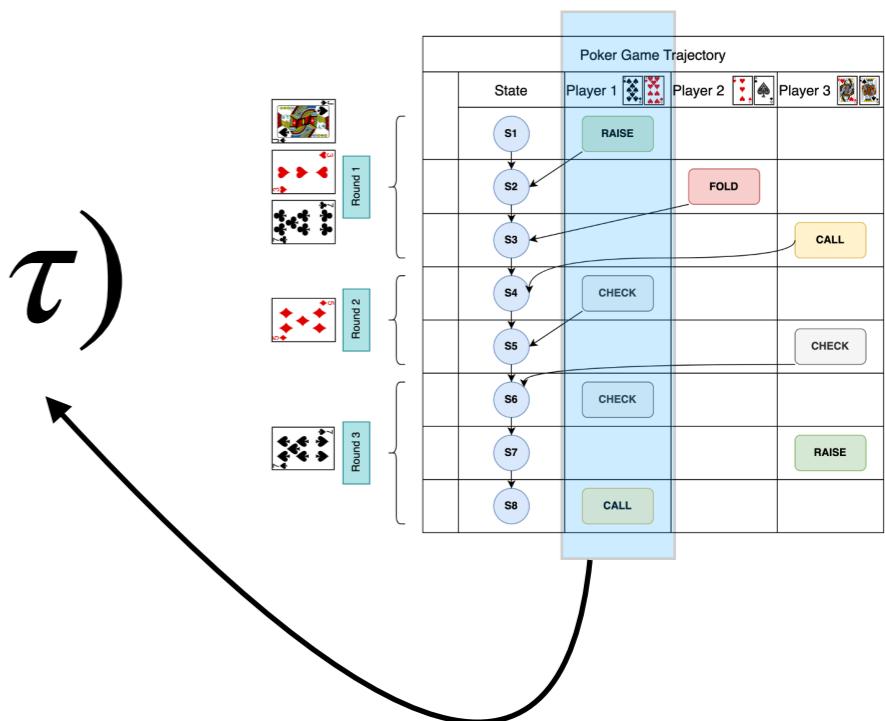


Discriminator

Setup and Challenges

- $f_{B_1}(s) \rightarrow a$
- $f_{B_2}(s) \rightarrow a$
- Action Space is small
- State Space is intractable

$$p(\{B_1, B_2\} | \tau)$$

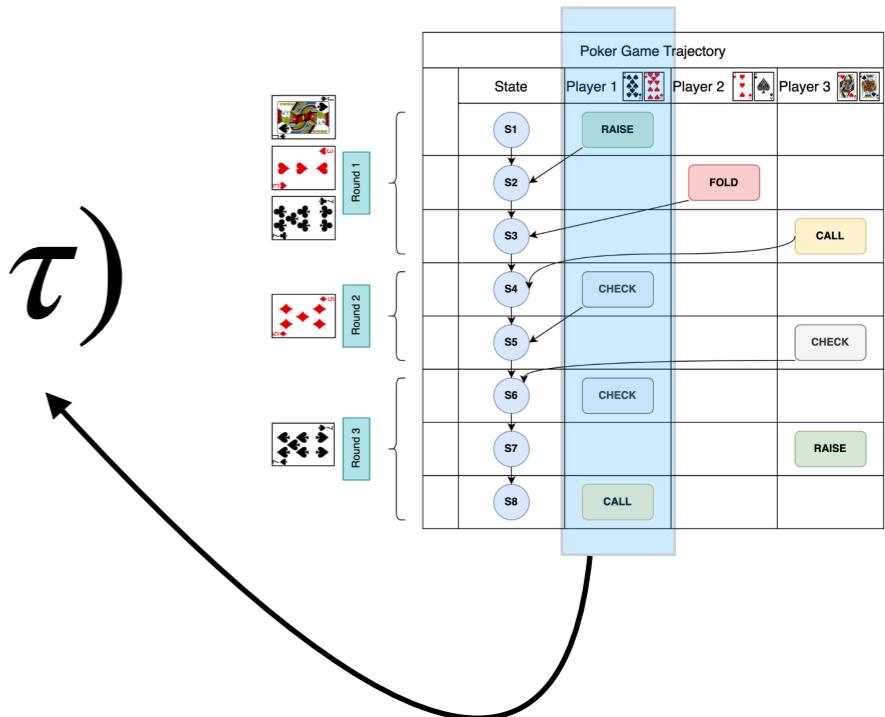


Discriminator

Setup and Challenges

- $f_{B_1}(s) \rightarrow a$
- $f_{B_2}(s) \rightarrow a$
- Action Space is small
- State Space is intractable

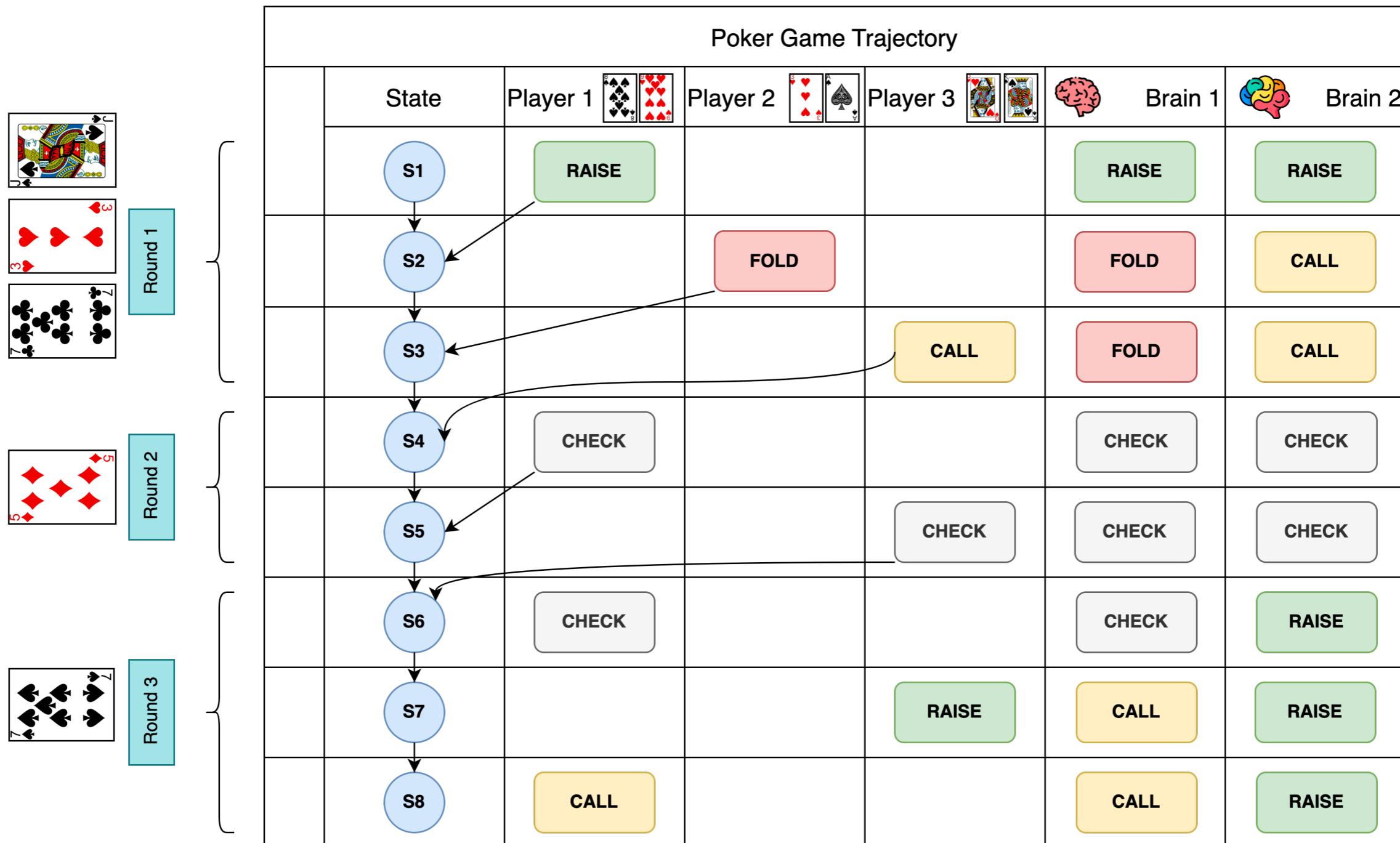
$$p(\{B_1, B_2\} | \tau)$$



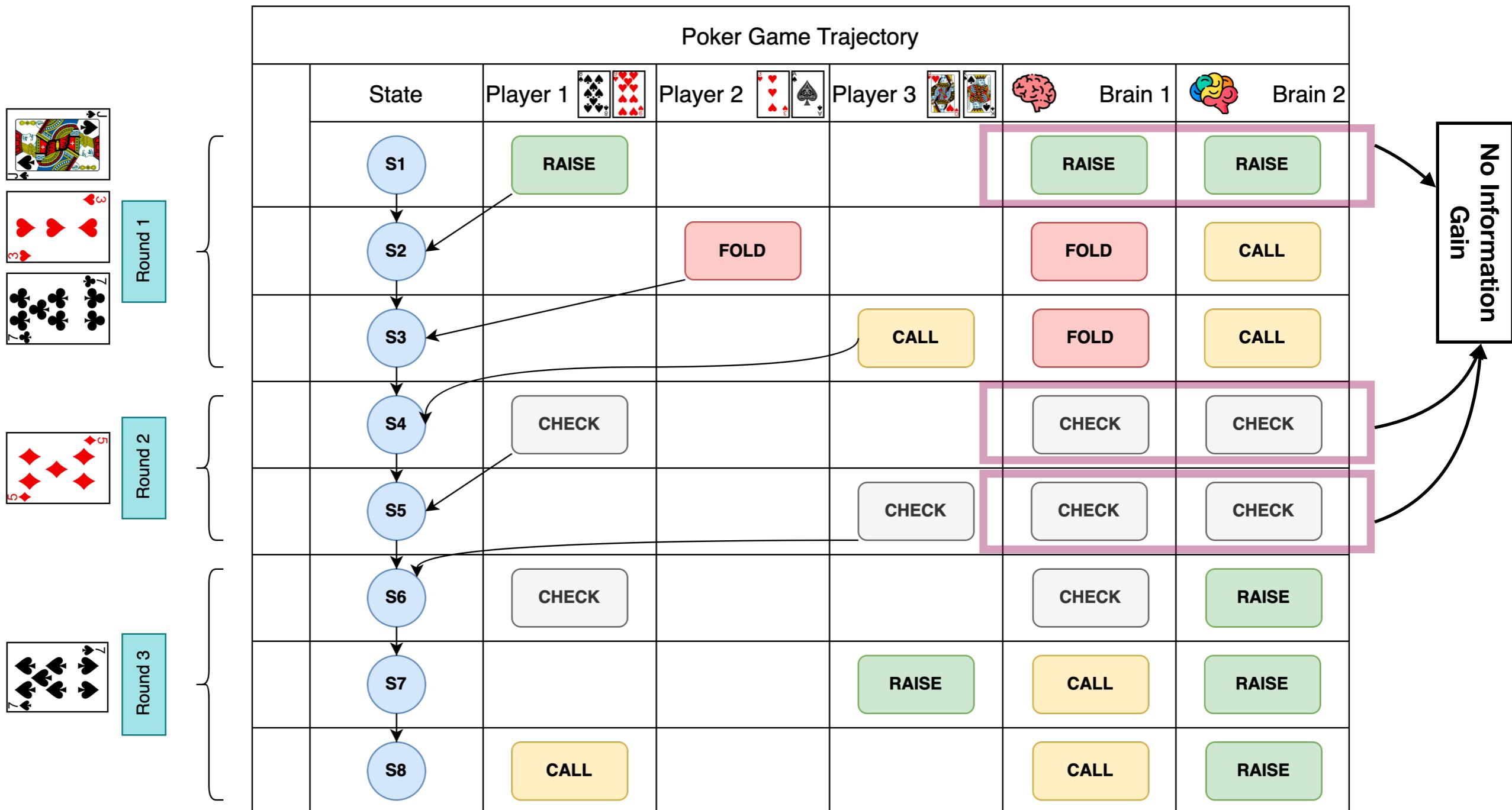
Can we take advantage of small action space to create a discriminator which doesn't enumerate the state space?

Discriminator Logic

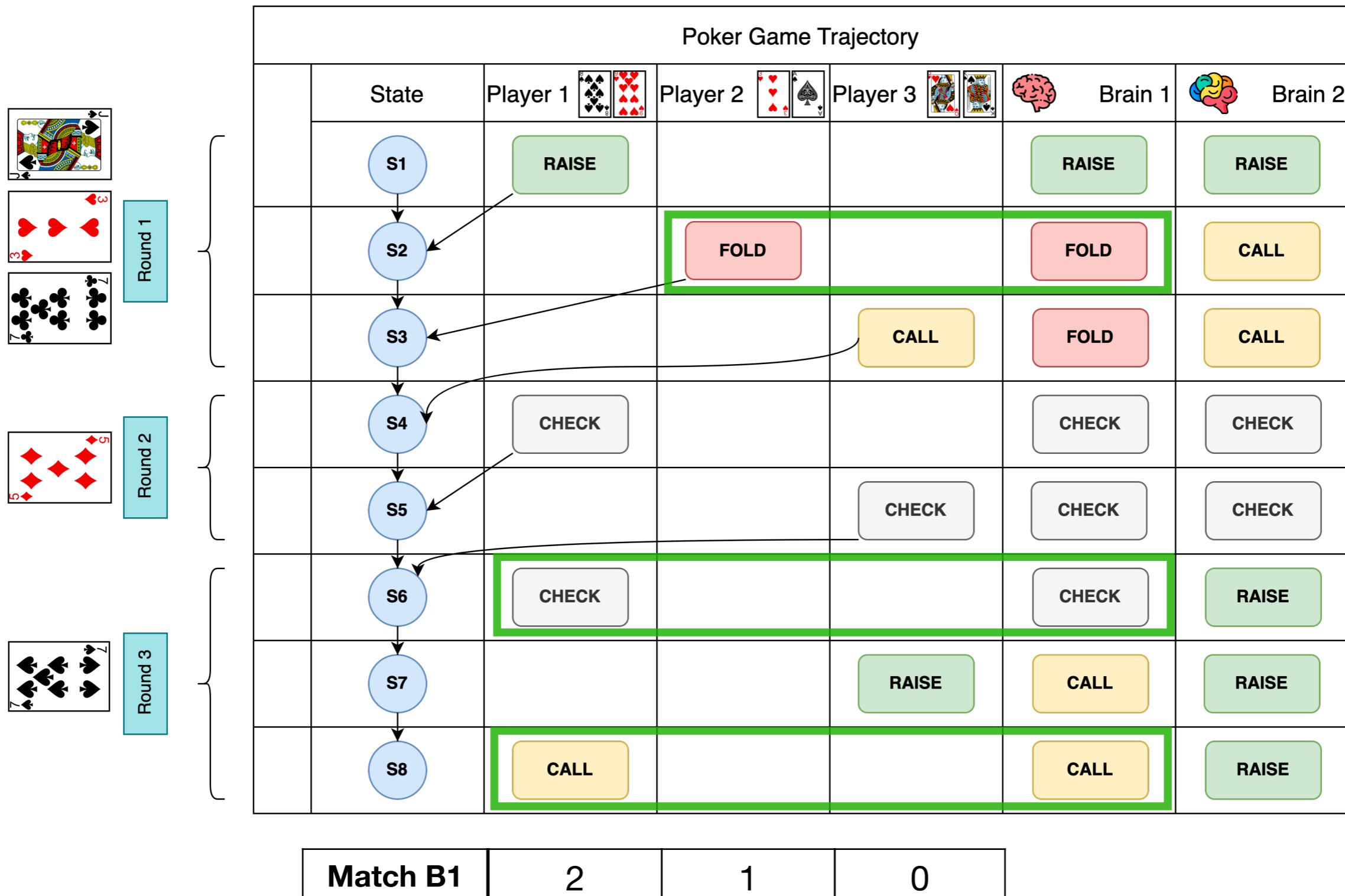
Discriminator Logic



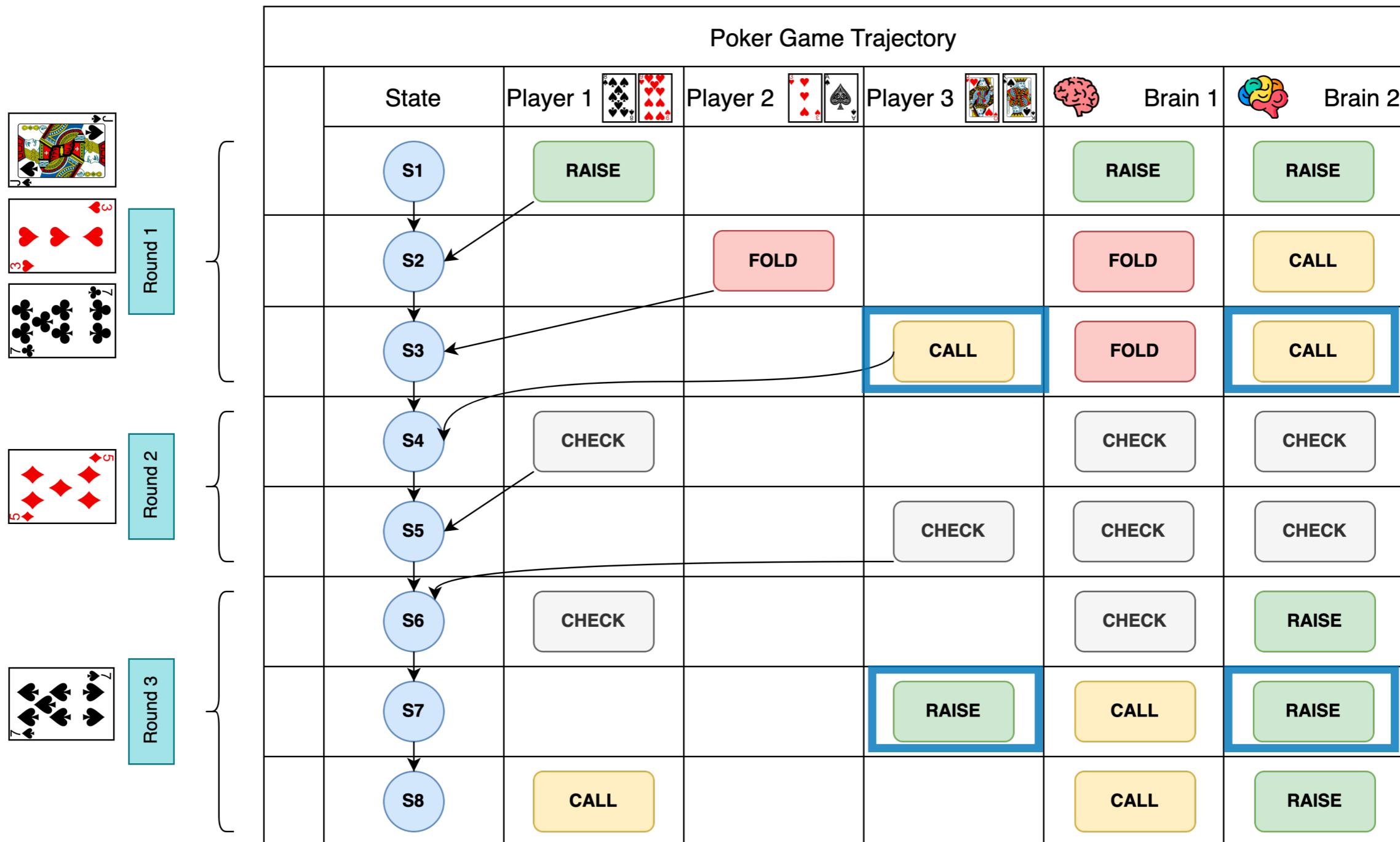
Discriminator Logic



Discriminator Logic



Discriminator Logic



Match B1	2	1	0
Match B2	0	0	2

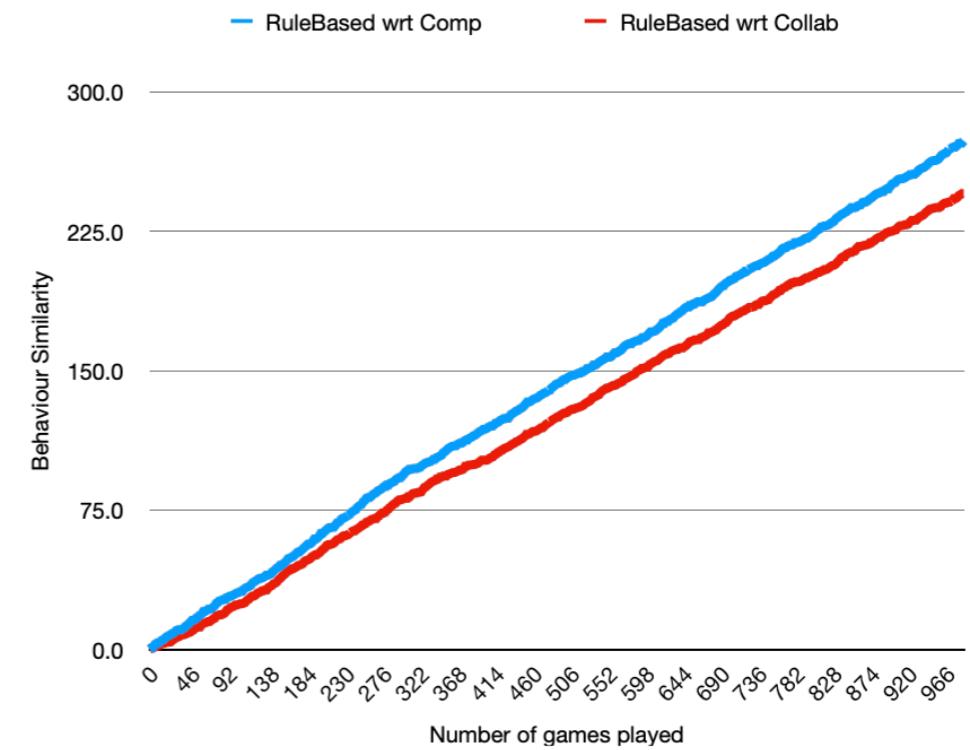
Discriminator Evaluation

Discriminator Evaluation

- GT Fair: RuleBased Agent
- GT Cheater1: Surrogate Cheater
- GT Cheater2: Surrogate Cheater

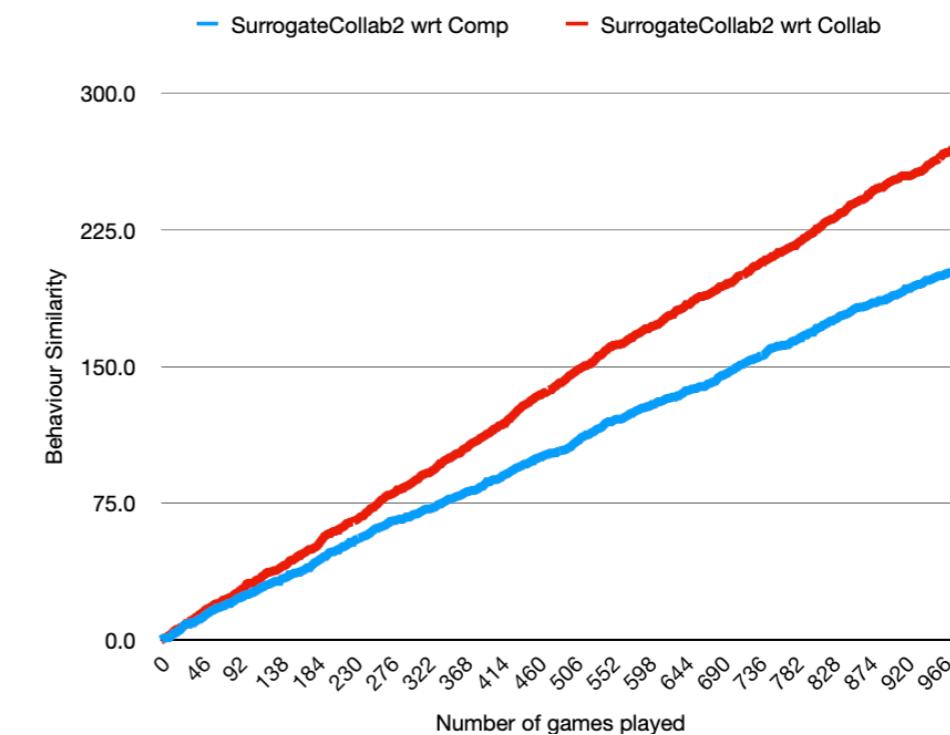
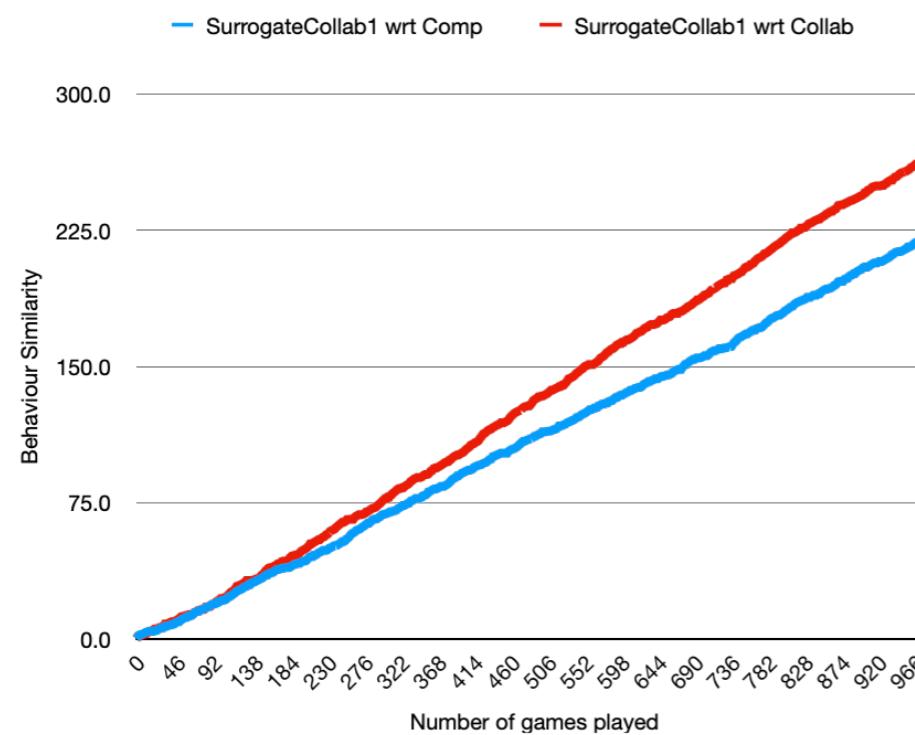
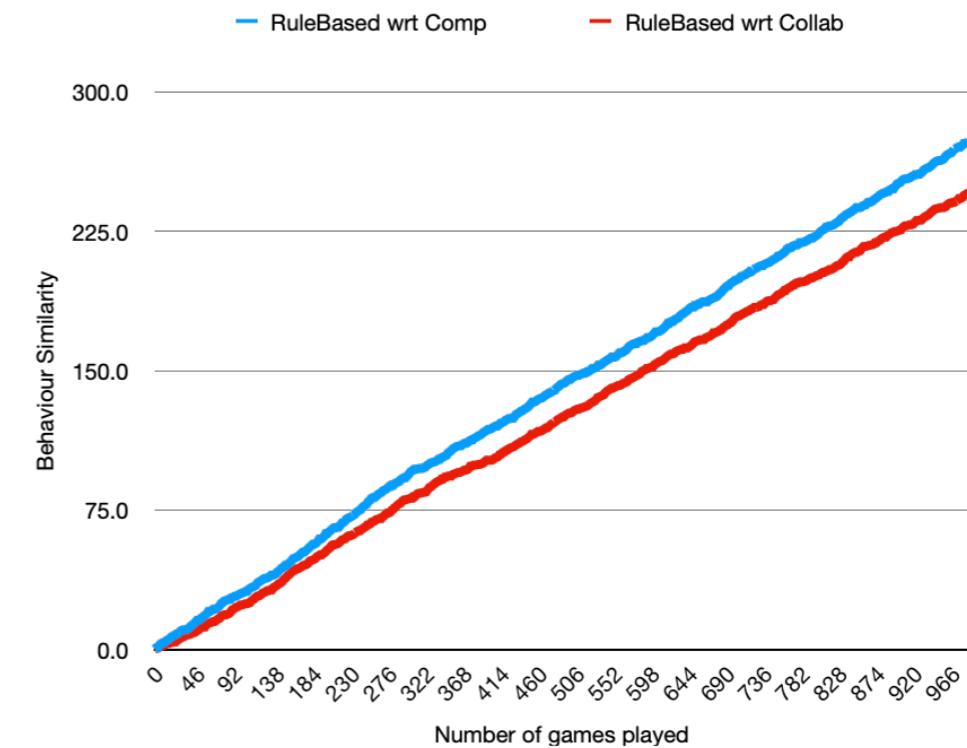
Discriminator Evaluation

- GT Fair: RuleBased Agent
- GT Cheater1: Surrogate Cheater
- GT Cheater2: Surrogate Cheater



Discriminator Evaluation

- GT Fair: RuleBased Agent
- GT Cheater1: Surrogate Cheater
- GT Cheater2: Surrogate Cheater



References

1. Bowling, M., Burch, N., Johanson, M., & Tammelin, O. (2015). Heads-up limit hold'em poker is solved. *Science*, 347(6218), 145–149. <https://doi.org/10.1126/science.1259433>
2. Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., & Whiteson, S. (2017). *Counterfactual Multi-Agent Policy Gradients* (arXiv:1705.08926). arXiv. <https://doi.org/10.48550/arXiv.1705.08926>
3. Heinrich, J., Lanctot, M., & Silver, D. (2015). Fictitious self-play in extensive-form games. Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, 805–813. <http://proceedings.mlr.press/v37/heinrich15.pdf>
4. Zhang, K., Yang, Z., & Başar, T. (2021). Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms (arXiv:1911.10635). arXiv. <http://arxiv.org/abs/1911.10635>
5. Icons images from FlatIcon.

Takeaways

- **Pro:**
 - The agents are model-agnostic for the discriminator framework to work.
 - This can scale up to $N > 3$ agents since we only make use of the tractable action-space.
- **Con:**
 - Our discriminator assumes that in real life scenarios, agents who are cheating deploys strategies similar to strategies learned by our policy. However, this assumption can be broken at times.