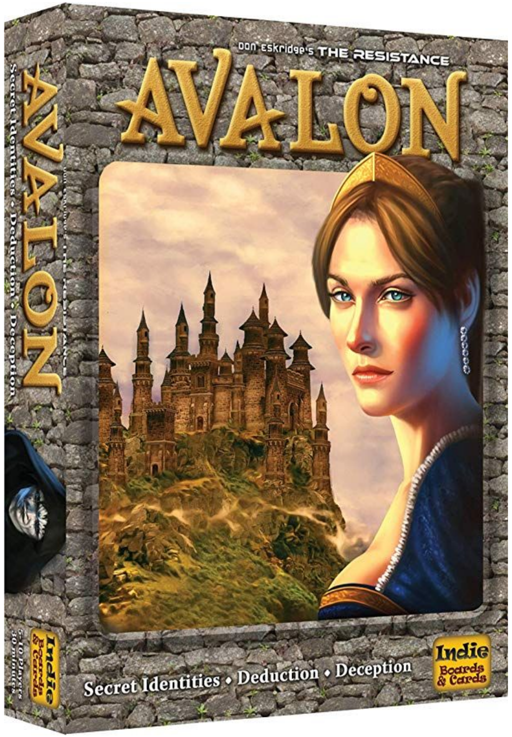
Finding Friend and Foe in Multi-agent Games

Jack Serrino*, Max Kleiman-Weiner*,
David Parkes, Josh Tenenbaum

Harvard, MIT, Diffeo

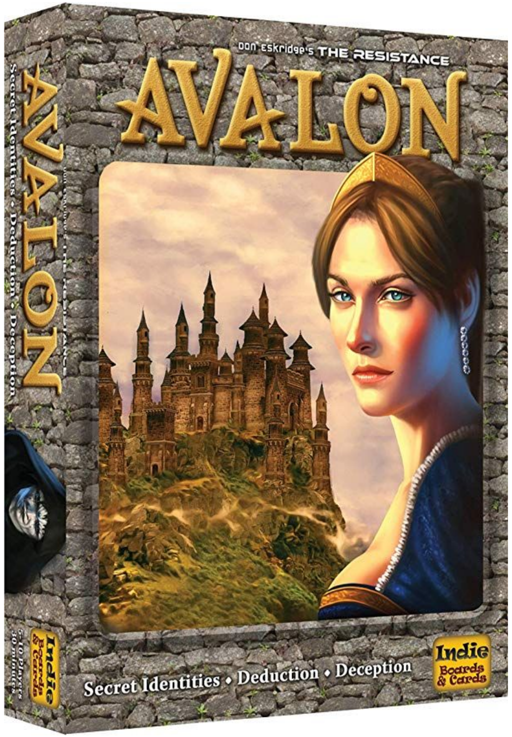
Poster #197

The Resistance: Avalon as a testbed for multi-agent learning and thinking



(Eskridge, 2012)

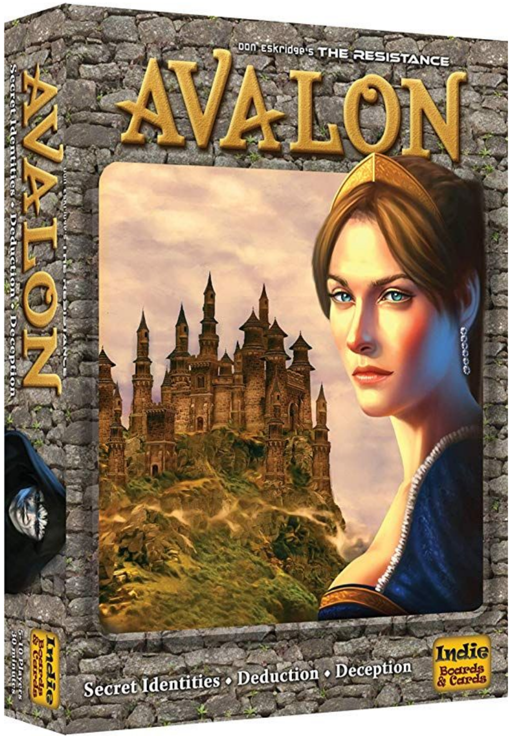
The Resistance: Avalon as a testbed for multi-agent learning and thinking



(Eskridge, 2012)

Recent progress limited to games where teams are known or play is fully adversarial (Dota, Go, Poker).

The Resistance: Avalon as a testbed for multi-agent learning and thinking



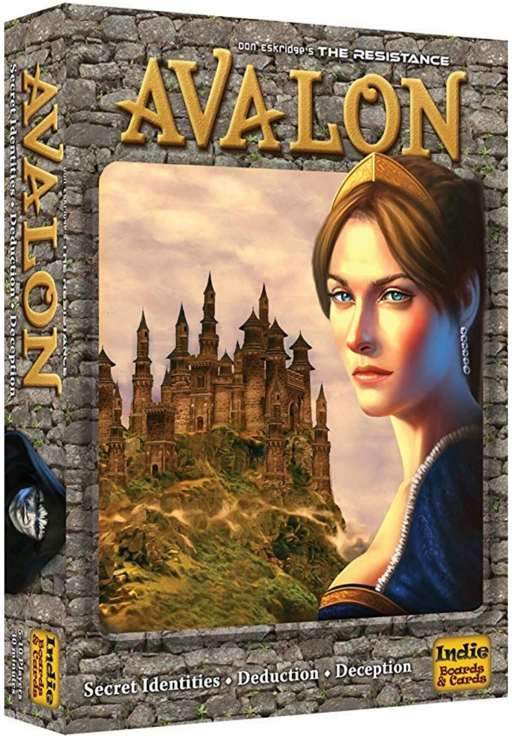
(Eskridge, 2012)

Recent progress limited to games where teams are known or play is fully adversarial (Dota, Go, Poker).

Avalon (5 Players)

- Two teams: "*Spy*" and "*Resistance*"
    - **Spies** know who is Spy and who is Resistance
        - Goal: *plan* to sabotage Resistance while hiding their own identity.
    - **Resistance** only know they are Resistance
        - Goal: *learn* who is a Spy & who is Resistance.

The Resistance: Avalon as a testbed for multi-agent learning and thinking



(Eskridge, 2012)

Recent progress limited to games where teams are known or play is fully adversarial (Dota, Go, Poker).

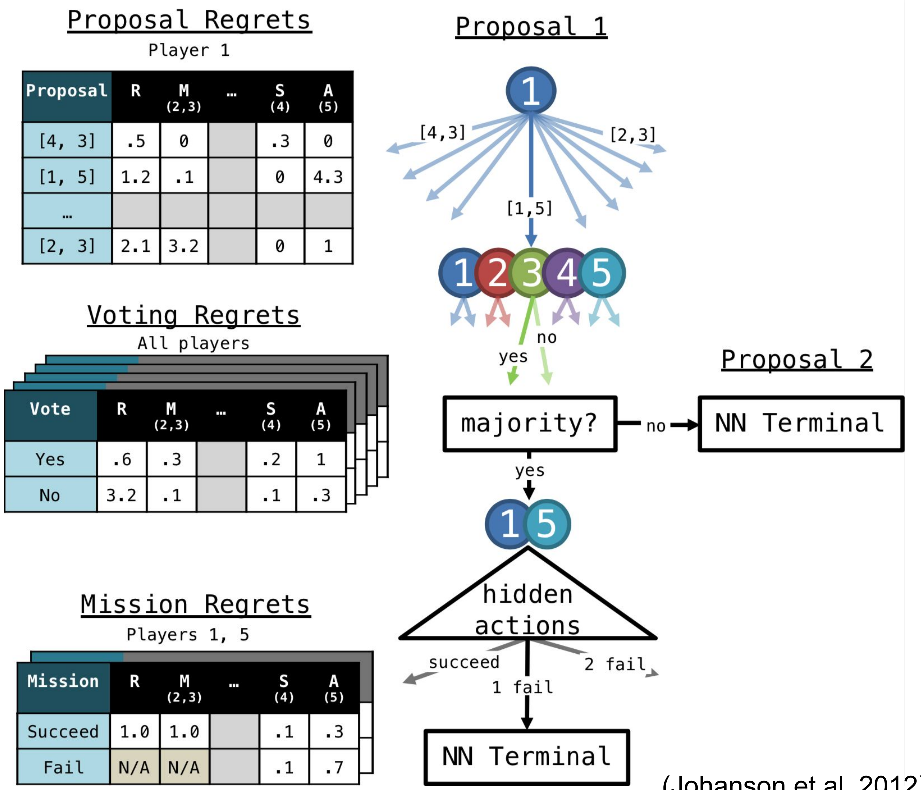
Avalon (5 Players)

- Two teams: "*Spy*" and "*Resistance*"
    - **Spies** know who is Spy and who is Resistance
        - Goal: *plan* to sabotage Resistance while hiding their own identity.
    - **Resistance** only know they are Resistance
        - Goal: *learn* who is a Spy & who is Resistance.

Information about intent is often noisy and ambiguous and adversaries may be intentionally acting to deceive.

Combining counterfactual regret minimization with deep value networks

- Approach follows DeepStack system developed for NL poker (Moravcik et al, 2017).



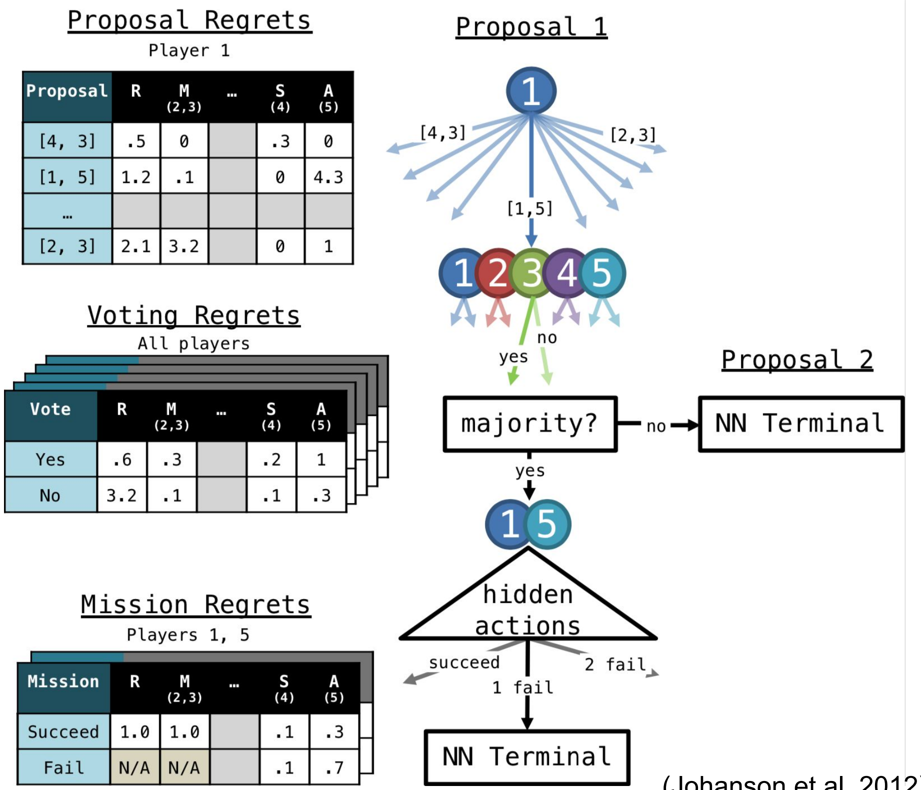
(Johanson et al, 2012)

Combining counterfactual regret minimization with deep value networks

- Approach follows DeepStack system developed for NL poker (Moravcik et al, 2017).

Main contributions:

- Actions themselves are only partially observed:
    - Deduction required in the loop of learning



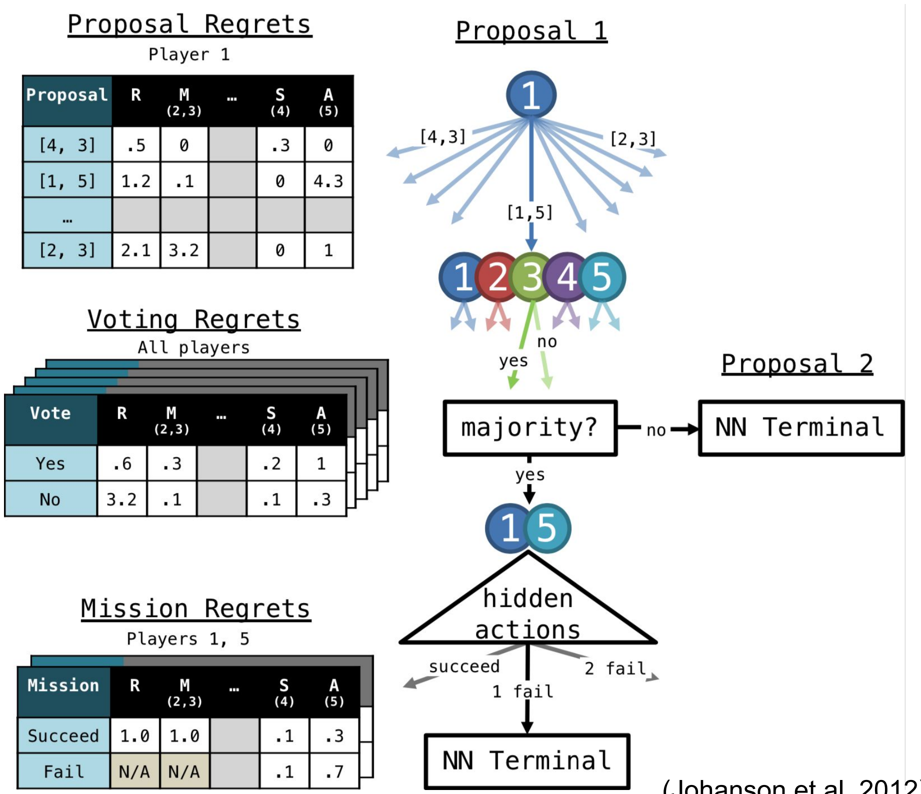
(Johanson et al, 2012)

Combining counterfactual regret minimization with deep value networks

- Approach follows DeepStack system developed for NL poker (Moravcik et al, 2017).

Main contributions:

- Actions themselves are only partially observed:
  - Deduction required in the loop of learning
- Unconstrained value networks are slower and less interpretable:
  - Develop an interpretable win-probability layer with better sample efficiency.



(Johanson et al, 2012)

Deductive reasoning enhances learning when actions are not fully public

procedure CALCTERMINALBELIEF($h, \mathbf{b}, \vec{\pi}_{1\ldots p}$)
  for $\rho \in \mathbf{b}$ **do**
1. $\mathbf{b}_{\text{term}}[\rho] \leftarrow \mathbf{b}[\rho] \prod_i \vec{\pi}_i(I_i(h, \rho))$
2. $\mathbf{b}_{\text{term}}[\rho] \leftarrow \mathbf{b}_{\text{term}}[\rho](1 - \mathbb{1}\{h \vdash \neg\rho\})$ $\triangleright$ Zero beliefs that are logically inconsistent
  end for
  return $\mathbf{b}_{\text{term}}$
end procedure

1. Calculate joint probability of assignment given the public game history

2. Zero out assignments that are impossible given the history.

2) is not necessary in games like Poker, with fully observable actions!

The Win Layer

$$V(I, \pi^\sigma) \in \mathbb{R}^{n \times |P|}$$

$$|P| := \text{number of assignments to roles, } \rho$$
$$n := \text{number of players}$$

Previous approaches:

$$NN(I, \pi^\sigma) \approx V(I, \pi^\sigma)$$

- In 5-player Avalon, 300 values to estimate!
- Correlations are learned imperfectly.

Our approach:

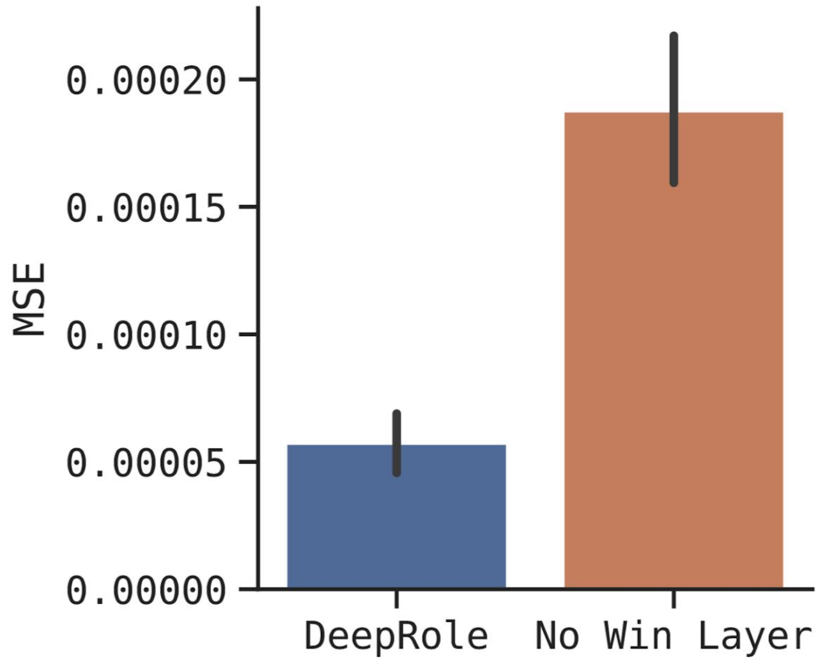
$$\vec{\mathbf{w}}(I, \pi^\sigma) = \begin{bmatrix} P(\text{good win}|I, \pi^\sigma, \rho_1) \\ \vdots \\ P(\text{good win}|I, \pi^\sigma, \rho_{|P|}) \end{bmatrix} \in [0, 1]^{|P|}$$

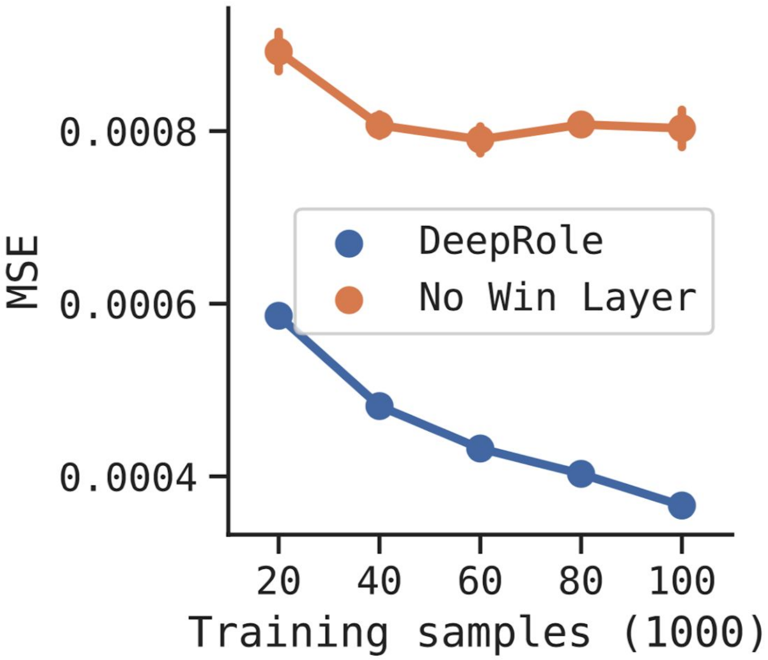
$$V(I, \pi^\sigma) = f(\vec{\mathbf{w}}(I, \pi^\sigma))$$

$$NN(I, \pi^\sigma) \approx \vec{\mathbf{w}}(I, \pi^\sigma)$$

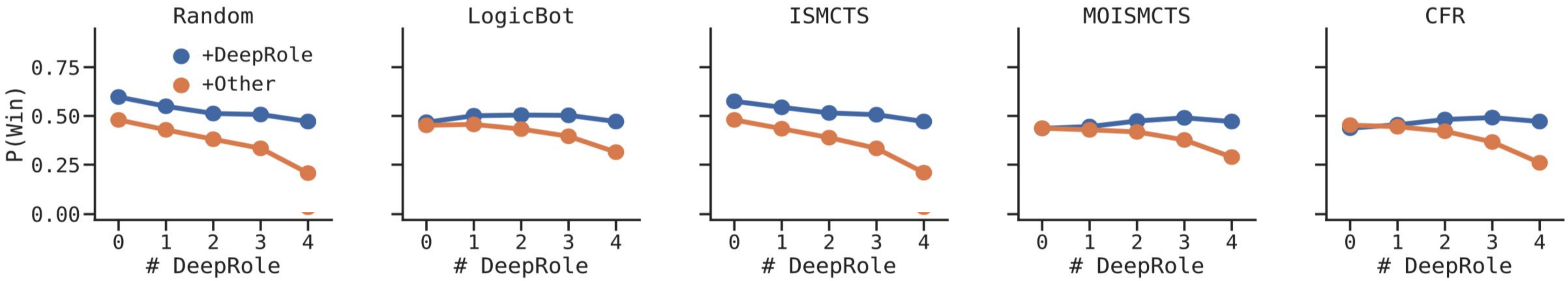
- 60 values to estimate (via sigmoid)
- Correlations are exact.

The Win Layer enables faster + better NN training



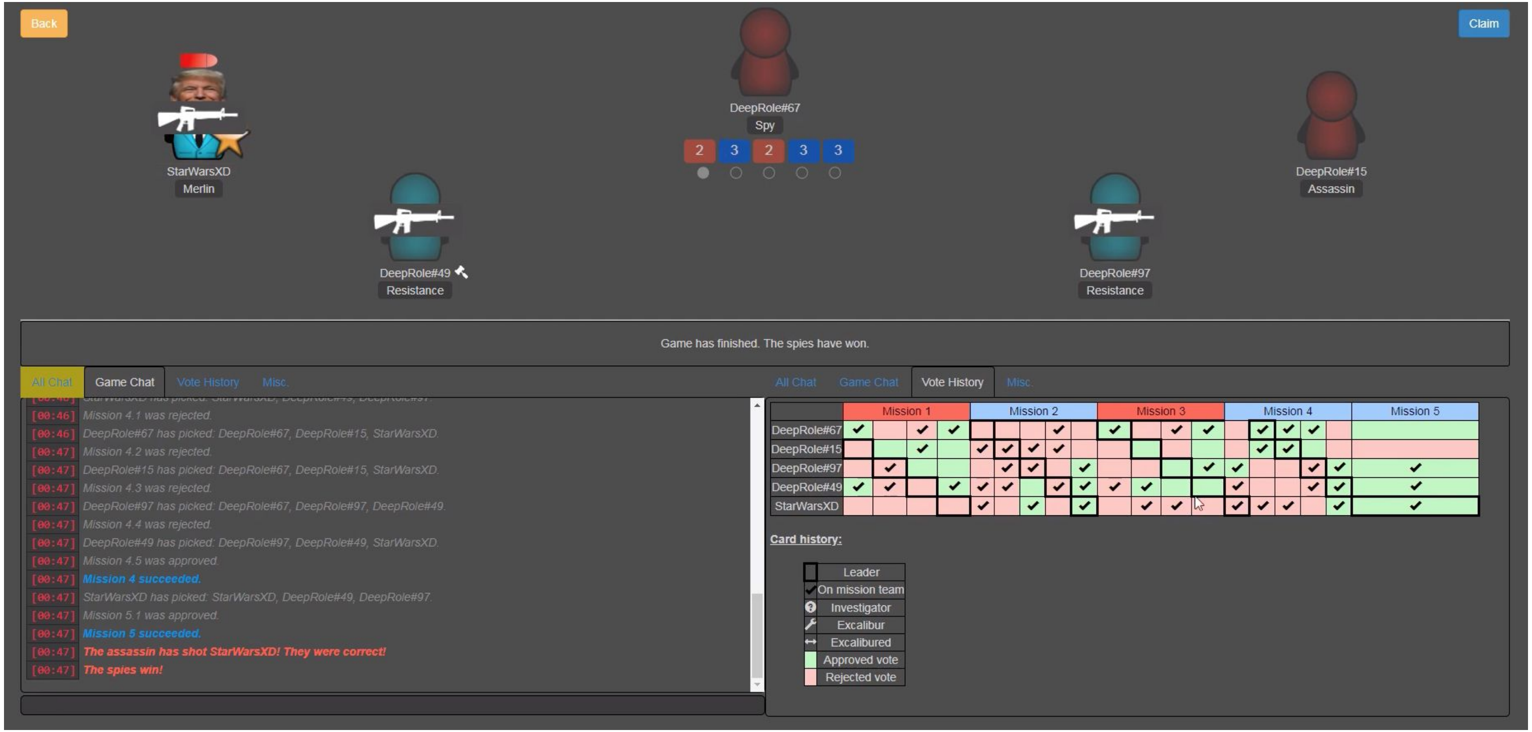


DeepRole wins at higher rates than: vanilla-CFR, MCTS, heuristic algorithms



(Wellman, 2006; Tuyls et al 2018)

DeepRole played online in mixed teams of human and bot players w/o communication (1,500+ games)



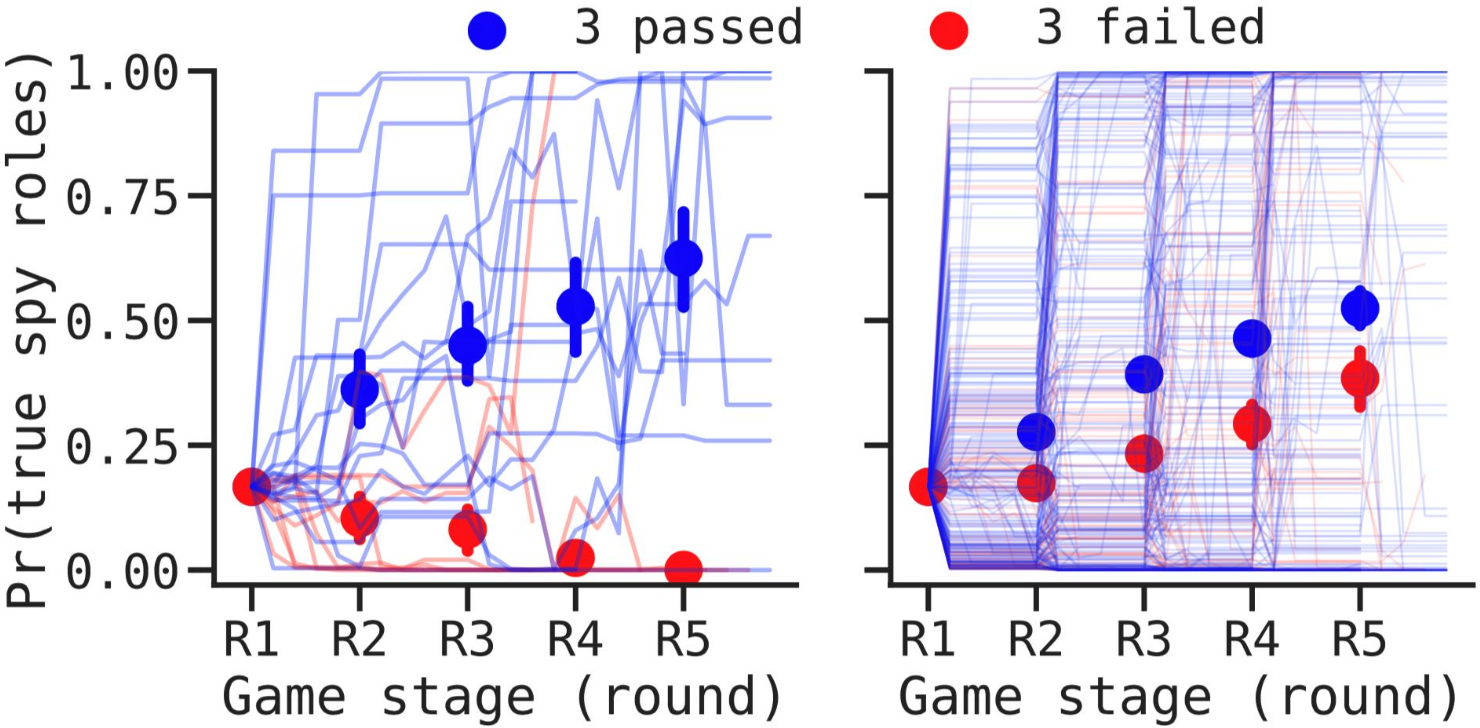
DeepRole outperformed humans playing online as both a collaborator and competitor

	Adding DeepRole or a Human to 4 DeepRole			
	+DeepRole		+Human	
	Win Rate (%)	(N)	Win Rate (%)	(N)
Overall	**46.9 ± 0.6**	(7500)	38.8 ± 1.3	(1451)
Resistance	**34.4 ± 0.7**	(4500)	25.6 ± 1.5	(856)
Spy	**65.6 ± 0.9**	(3000)	57.8 ± 2.0	(595)

DeepRole outperformed humans playing online as both a collaborator and competitor

	Adding DeepRole or a Human to 4 DeepRole				Adding DeepRole or a Human to 4 Human			
	+DeepRole		+Human		+DeepRole		+Human	
	Win Rate (%)	(N)	Win Rate (%)	(N)	Win Rate (%)	(N)	Win Rate (%)	(N)
Overall	**46.9 ± 0.6**	(7500)	38.8 ± 1.3	(1451)	**60.0 ± 5.5**	(80)	48.1 ± 1.2	(1675)
Resistance	**34.4 ± 0.7**	(4500)	25.6 ± 1.5	(856)	**51.4 ± 8.2**	(37)	40.3 ± 1.5	(1005)
Spy	**65.6 ± 0.9**	(3000)	57.8 ± 2.0	(595)	**67.4 ± 7.1**	(43)	59.7 ± 1.9	(670)

DeepRole make rapid accurate inferences about human roles during play and observation



Finding Friend and Foe in Multi-agent Games

Jack Serrino*, Max Kleiman-Weiner*,
David Parkes, Josh Tenenbaum

Harvard, MIT, Diffeo

Poster #197

Play online: **ProAvalon.com**