Self-Play and Zero-Shot (Human-AI) Coordination (in Hanabi)

Jakob Foerster
Research Scientist, Facebook AI Research
Assistant Professor, University of Toronto / Vector Institute (incoming)

Roadmap

- Intro:
  - Why study Multi-Agent Learning?
  - Theory of Mind and the Hanabi Challenge
- Part I: Self-Play and the Bayesian Action Decoder
  - Self-Play Setting
  - Public Beliefs
  - The Bayesian Action Decoder
  - Results
- Part II: Zero-Shot Coordination and “Other-Play”
  - The Zero-Shot Coordination Setting
  - “Other-Play”
  - Results in Hanabi
  - Human-AI Results

Why study Multi-Agent Learning?

Answer 1: The obvious Real World applications

https://www.intelematics.com/news/how-internet-of-things-is-connecting-the-world-one-device-at-a-time/, https://www.cdrinfo.com/Sections/news/Print.aspx?NewsId=42290, https://www.qriotek.com/

Answer 2: Interacting with Humans and other agents

Agency in Traffic

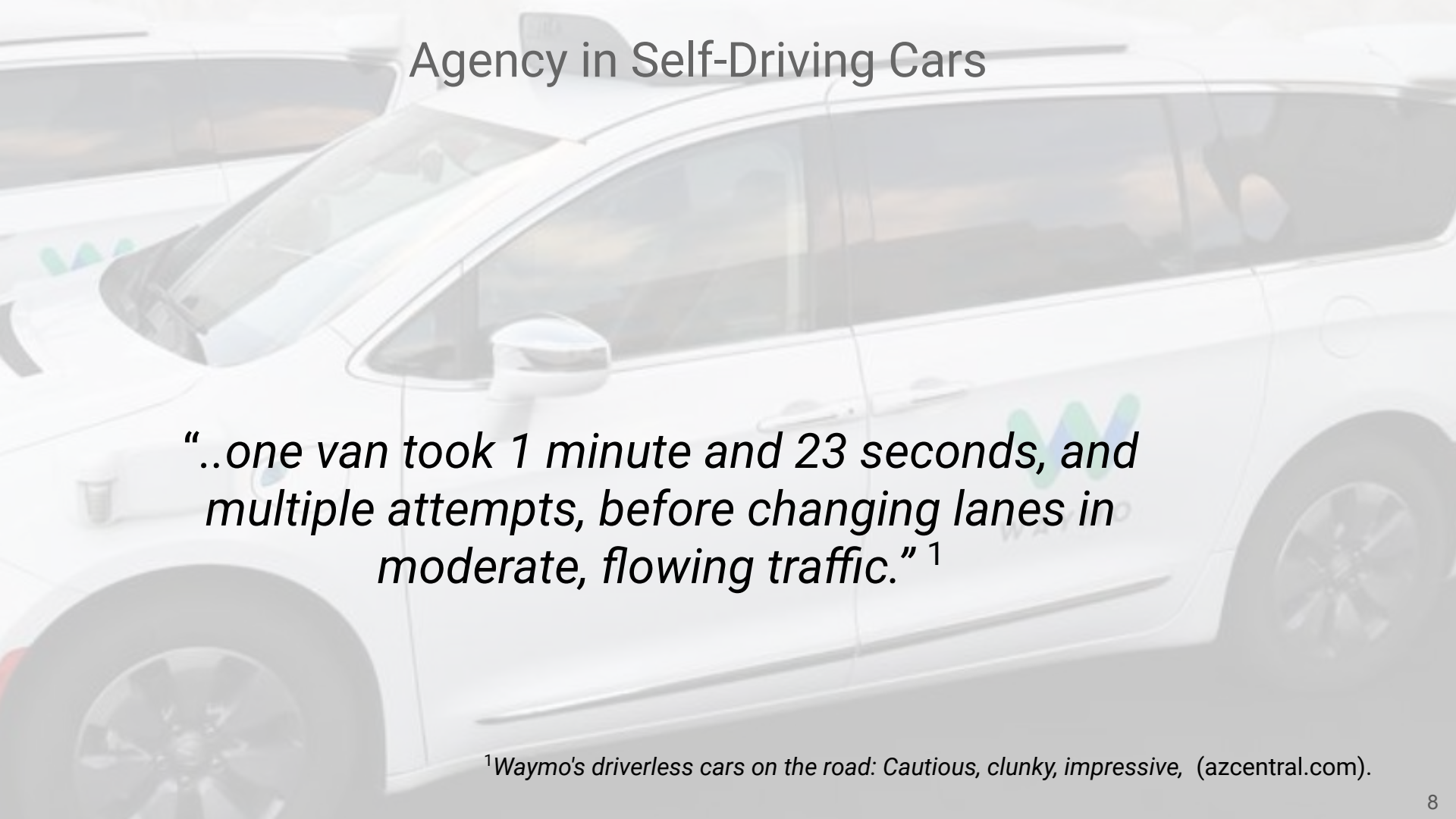
azcentral.com

Agency in Self-Driving Cars

"..highly interactive situations such as merging [..] will demand simulating other (human) traffic participants, a rich area of ongoing research" [1]

[1] *ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst,* Mayank Bansal, Alex Krizhevsky, Abhijit Ogale, 2018 (Waymo)

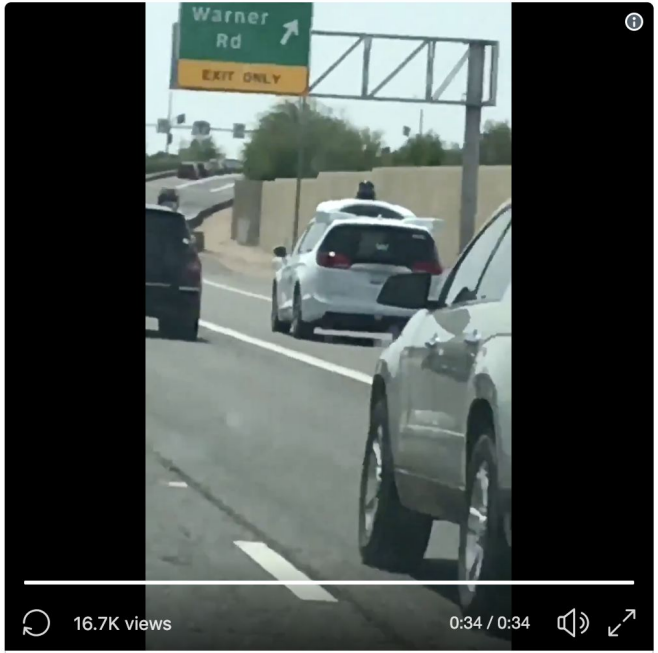
Agency in Self-Driving Cars



"..one van took 1 minute and 23 seconds, and multiple attempts, before changing lanes in moderate, flowing traffic." [1]

[1] *Waymo's driverless cars on the road: Cautious, clunky, impressive,* (azcentral.com).

Agency in Self-Driving Cars



Nitin Gupta
@nitguptaa

Self driving #Waymo car tried merging onto the highway, missed multiple opportunities (programmed defensive driving?), then rerouted to exit after failing 😂😂

♡ 140 12:06 PM - Apr 29, 2018 · Tempe, AZ

https://twitter.com/nitguptaa/status/990683818825736192?lang=en

Answer 3: (almost) All other ML Applications deployed in the real world

Agency in Supervised problems

Machine Translation

Training Data

Users

Agency in Supervised problems

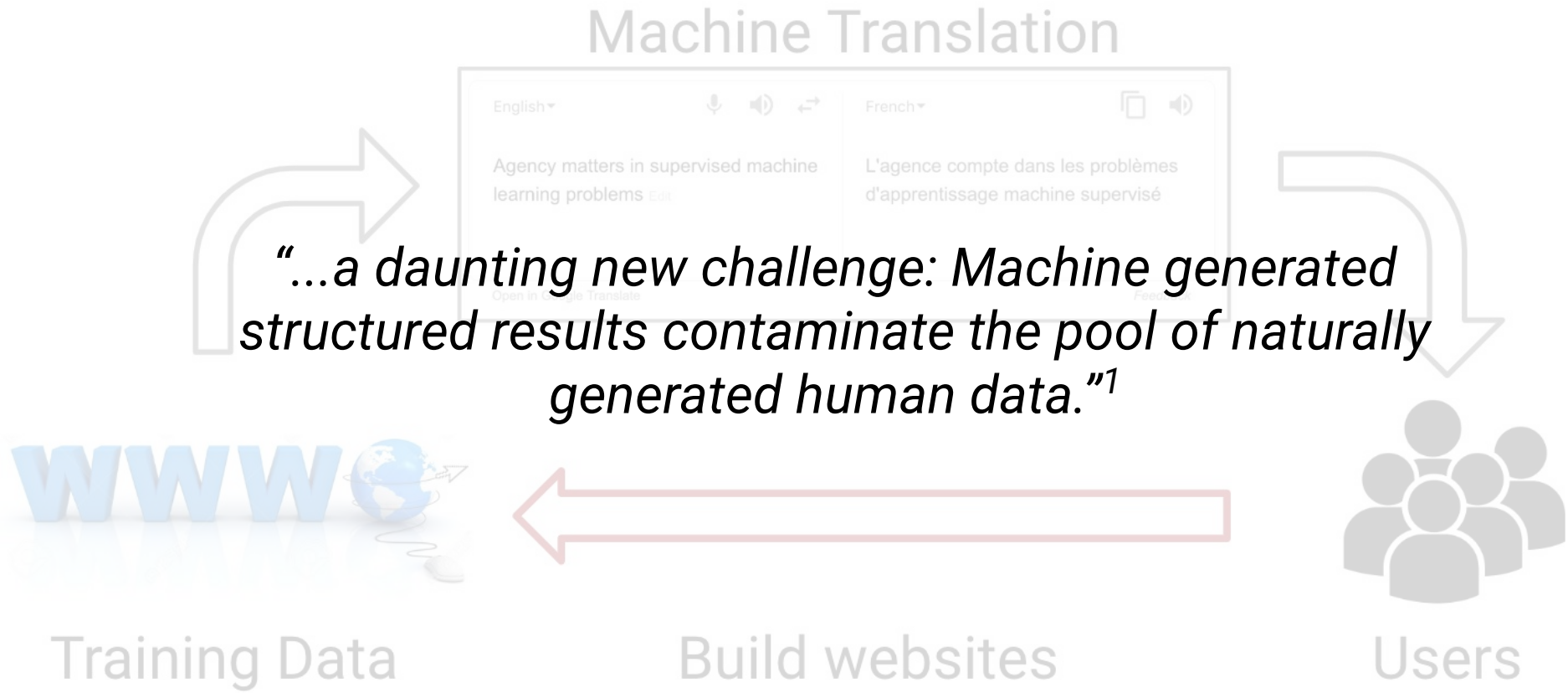
Machine Translation

Training Data

Build websites

Users

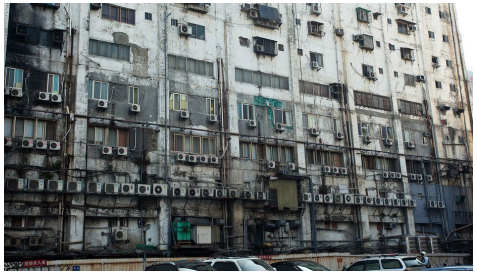
Agency in Supervised problems



"...a daunting new challenge: Machine generated structured results contaminate the pool of naturally generated human data."[1]

"Watermarking the Outputs of Structured Prediction with an application in Statistical Machine Translation", Ashish Venugopal, Jakob Uszkoreit, David Talbot, Franz J. Och, Juri Ganitkevitch, EMNLP 2011

Agency in Markets



Seller 1 (Profnath):

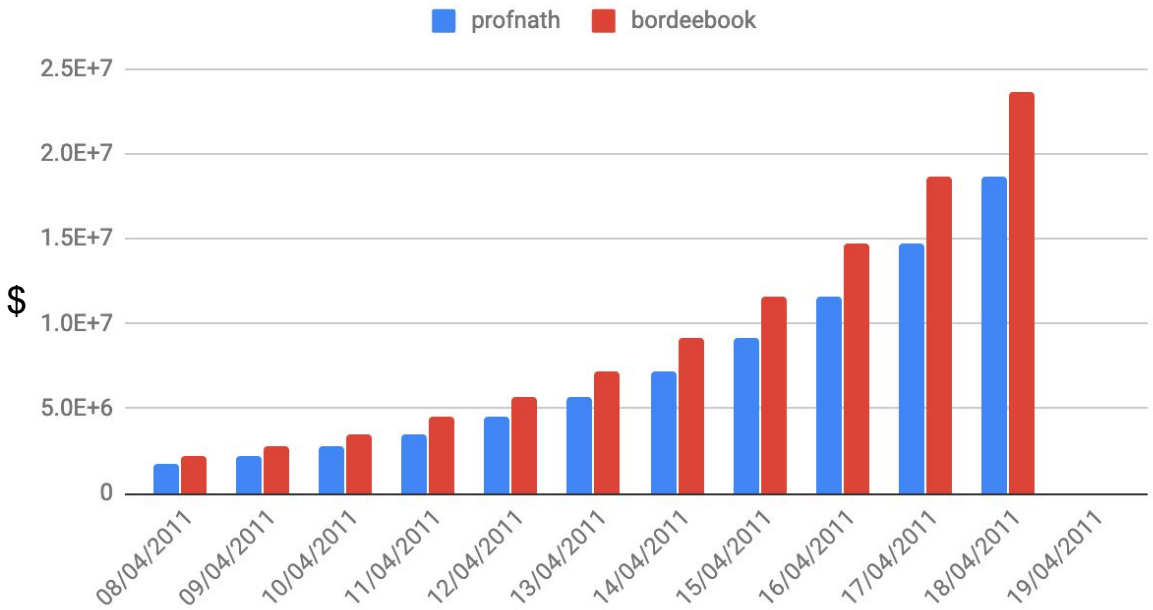
- Compete on price
- Set price just below (-0.127%) cheapest competitor from previous day

Seller 2 (Bordeebook):

- Great customer service
- Loads of positive reviews
- Set price above (+27%) of most expensive competitor

Book Pricing

Agency in Markets



Book Pricing

Answer 4: Agent-Based Modelling

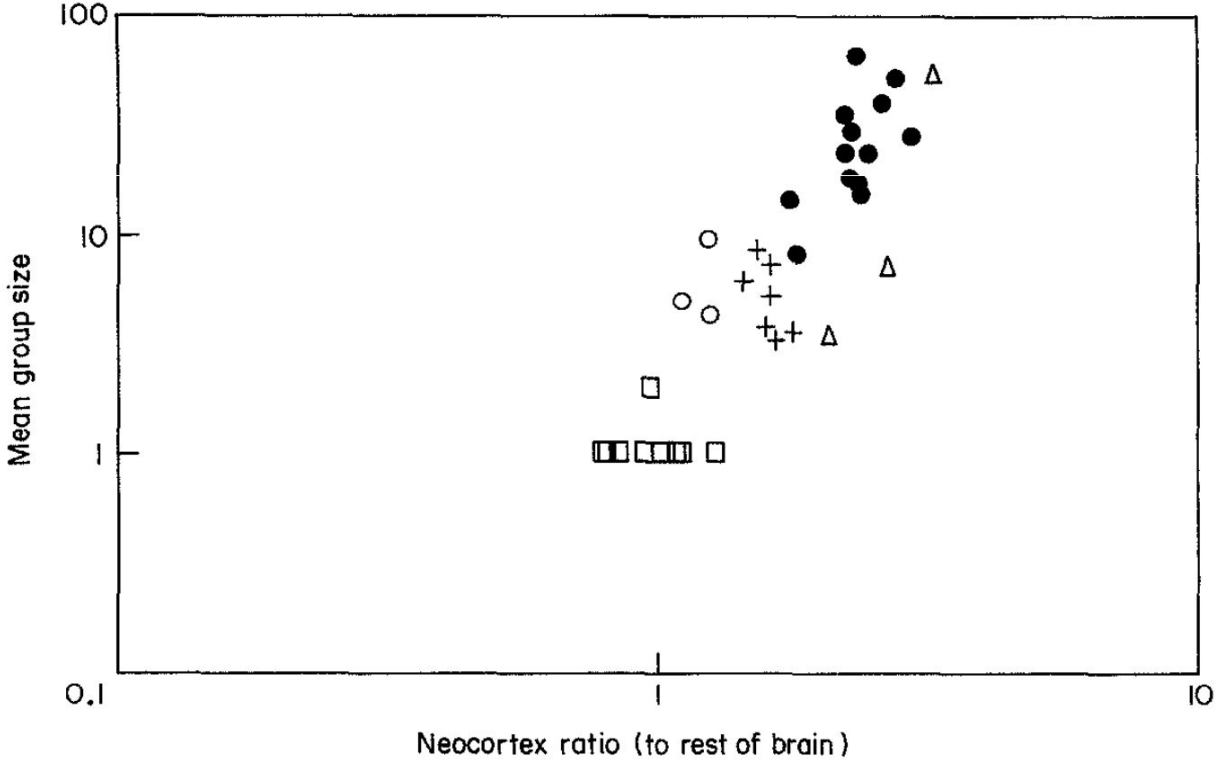
https://climate.nasa.gov/system/content_pages/main_images/1320_effects-image.jpg

https://www.theatlantic.com/technology/archive/2018/03/largest-study-ever-fake-news-mit-twitter/555104/

https://www.aljazeera.com/news/2020/04/06/world-opinion-shifts-in-favour-of-masks-as-virus-fight-deepens/

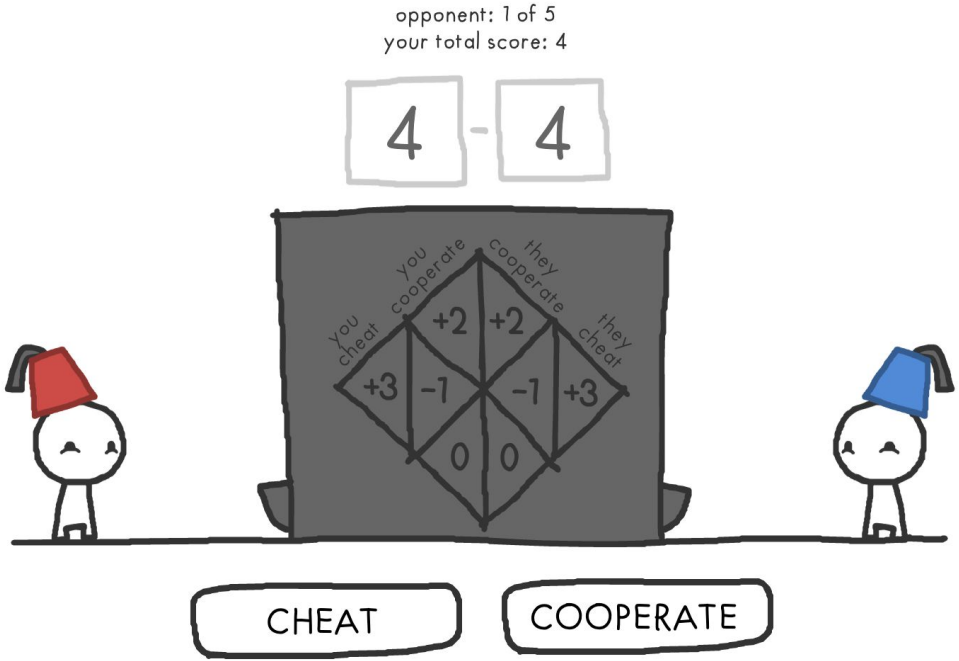
Answer 5: Because Human Level Intelligence is a multi-agent phenomena

“Neocortex size as a constraint on group size in primates” (R. I. M. Dunbar, 1989)



Answer 6:
An Easy Way to create Hard Problems

Iterated Matrix Games. So simple and yet *so hard.*

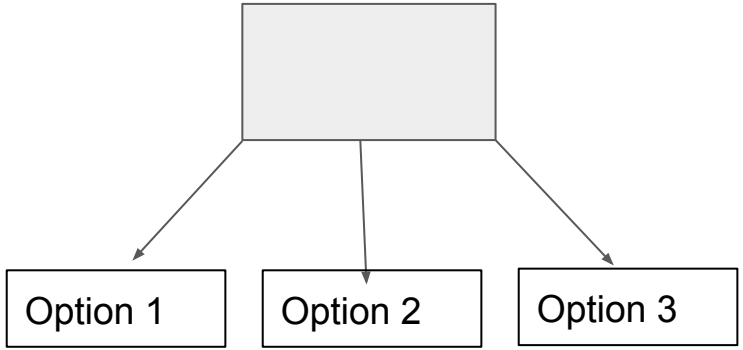


Try it yourself! https://ncase.me/trust/

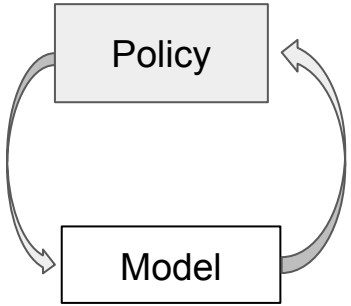
Answer 7:
Many Machine Learning Methods are Naturally Framed as Multi-Agent

GANs

Hierarchical Reinforcement Learning

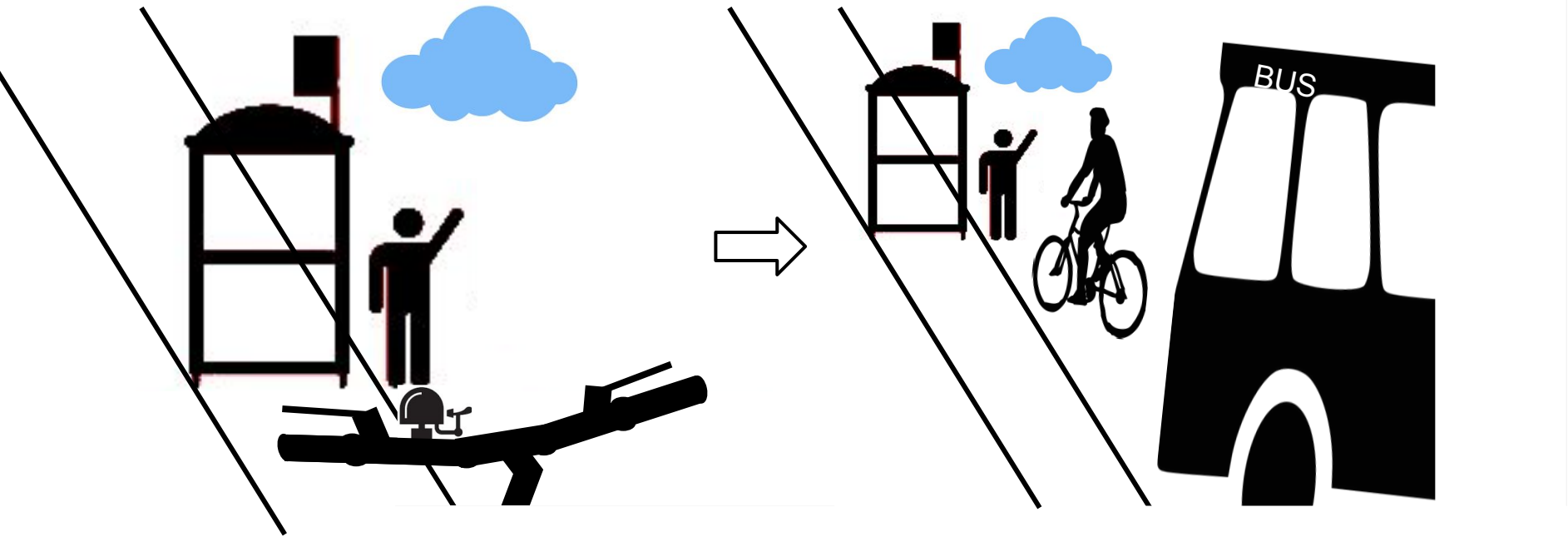


Model Based RL



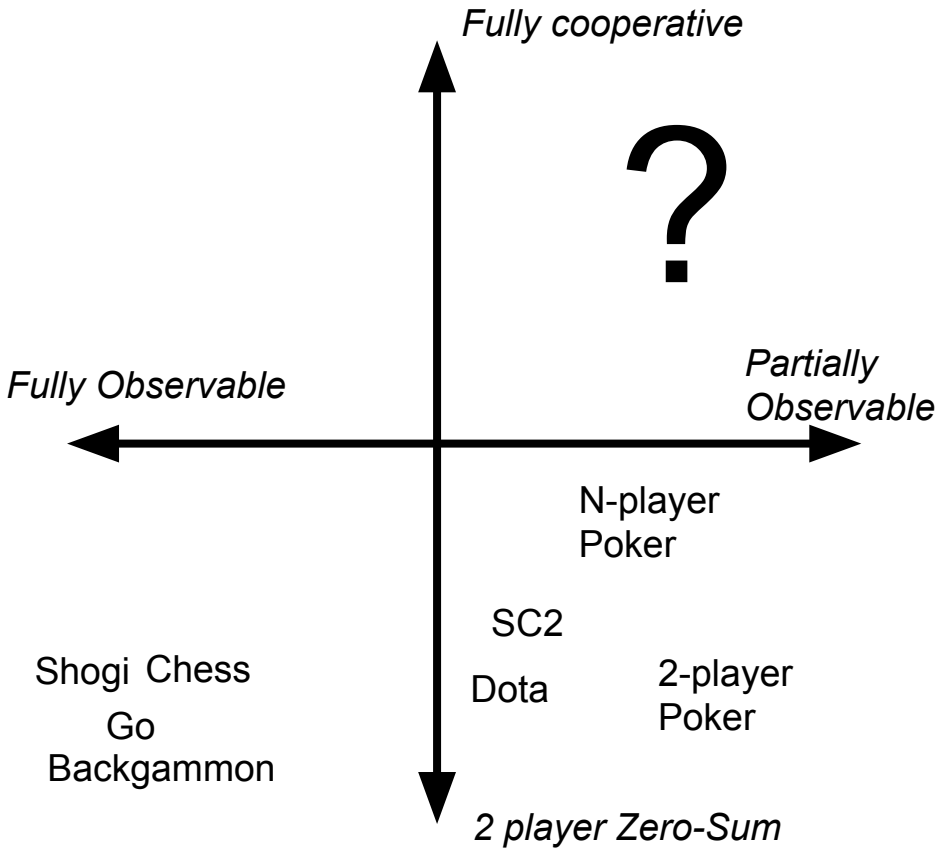
Answer 8:
It’s fun and fulfilling!!

Theory of Mind



- Interpret the actions of others when observing them
- Take actions that are informative when observed by others

Progress in AI applied to Games



So what? Maybe Two-Player Zero-Sum is sufficient?

Sadly, no!

Even Self-Play in Fully Cooperative partially observable settings has unique challenges.

Part I

In 2-player Zero-Sum all Nash are created equal (interchangeable)!

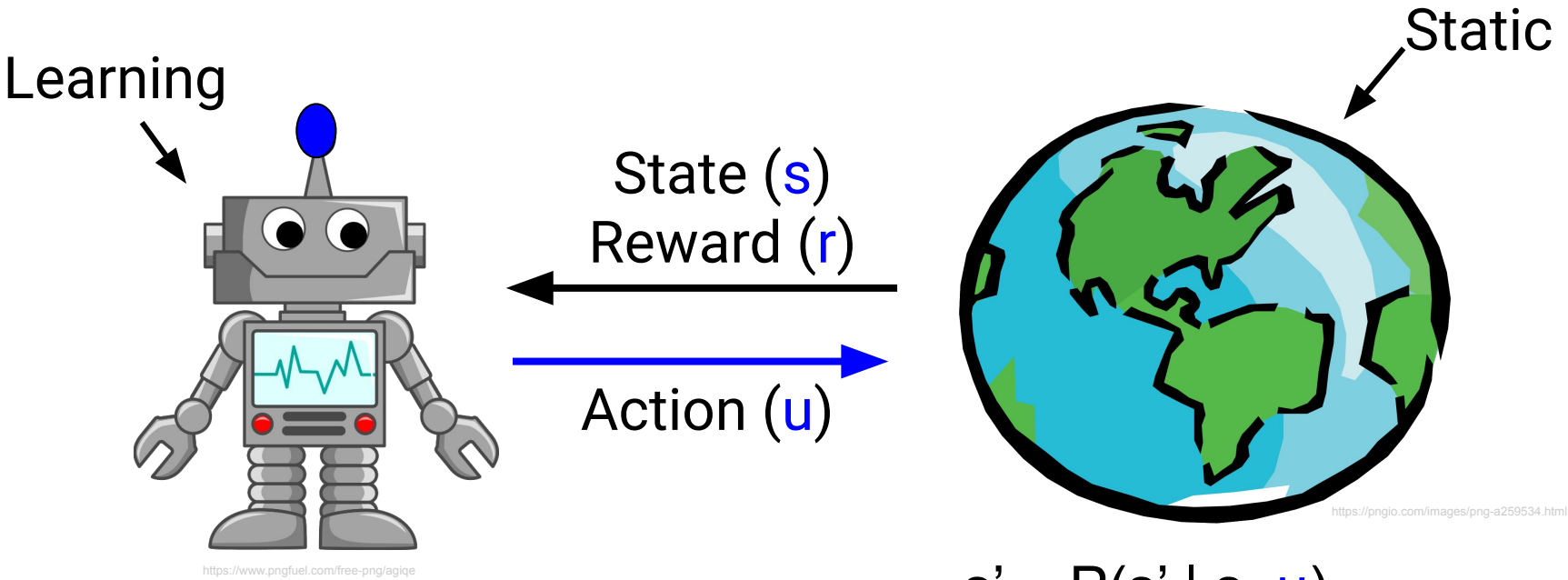
Variations of **Self-Play** guarantee you can't lose at test-time when matched with a stranger.

In fully cooperative settings, even *perfect* **Self-Play** policies can be arbitrarily bad when paired with a stranger

We need fundamentally new algorithms, formalisms and solution schemes for the fully-cooperative case.
In particular for Zero-Shot coordination and Human-AI coordination

Part II

Reinforcement Learning



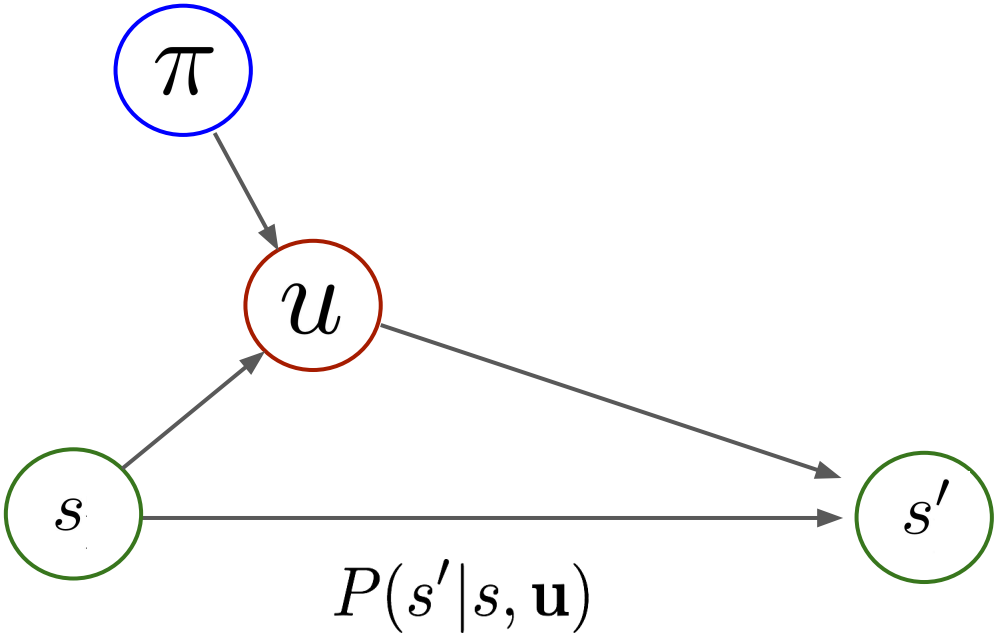
$s' \sim P(s' \mid s, u)$

Goal is to maximise *total return* per episode: $J = \sum_t \gamma^t r_t$

$$s' \sim P(s' \mid s, u)$$

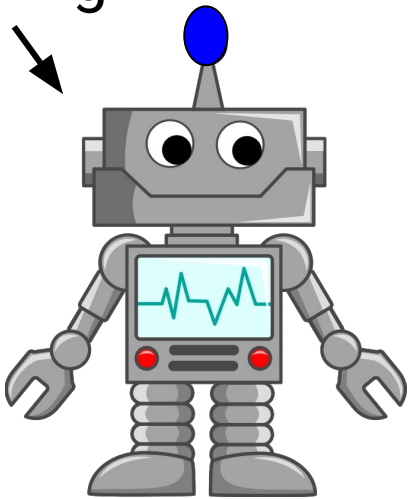
Only *what* happens (u) matters for the future.
Why, i.e. the policy, does not.

Markov Decision Process



Reinforcement Learning

Learning



https://www.pngfuel.com/free-png/agiqe

Observation (o)

~~State (s)~~

Reward (r)

Action (u)

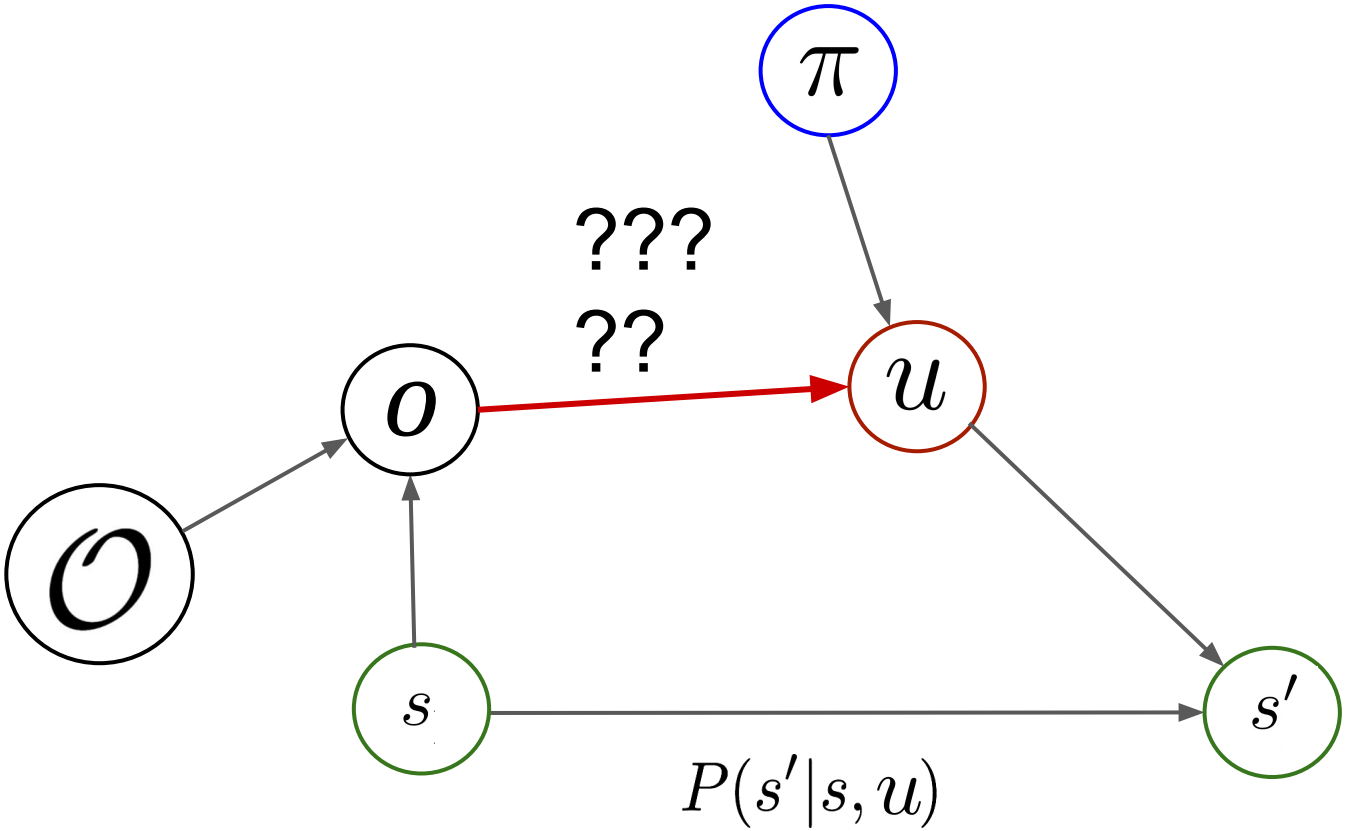
Static

https://pngio.com/images/png-a259534.html

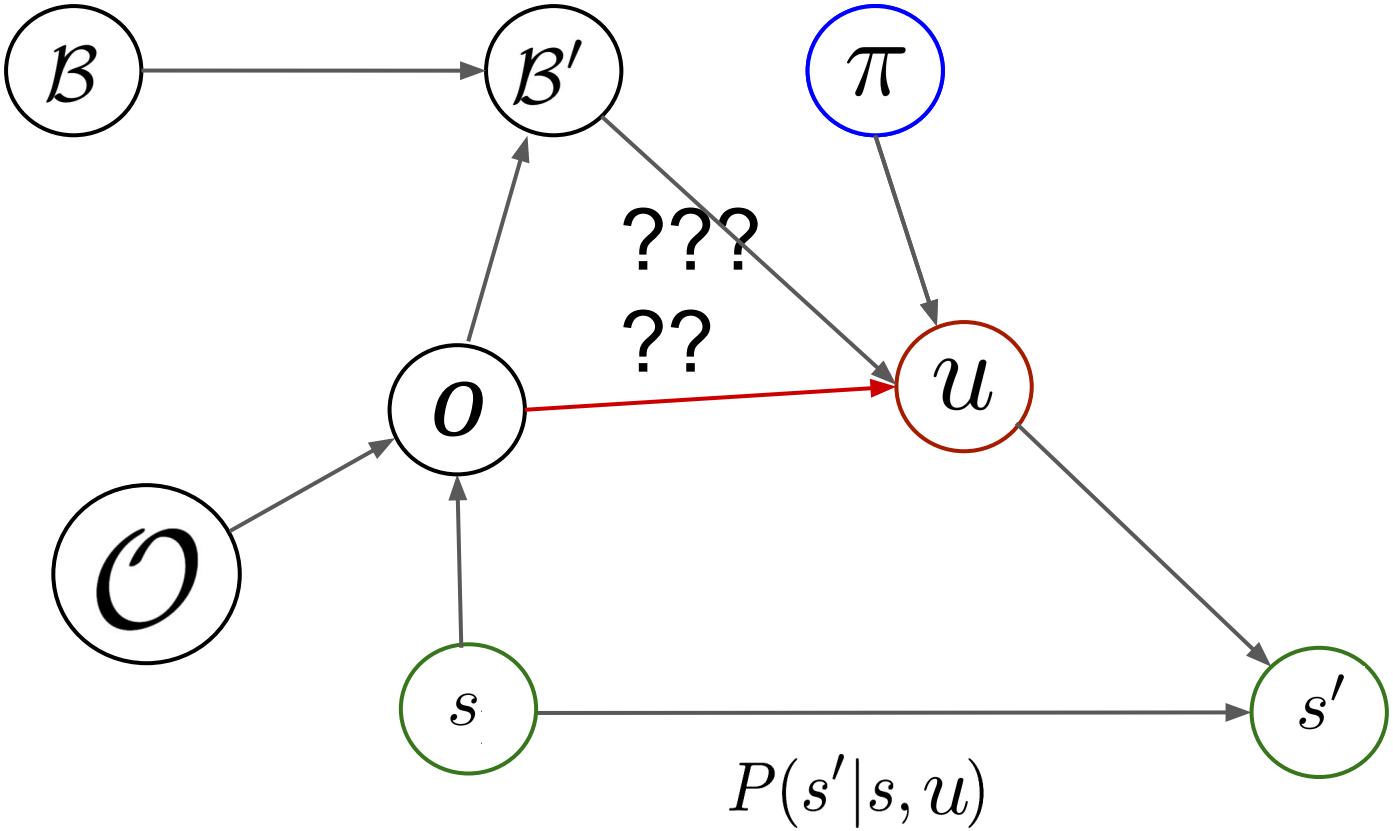
$s' \sim P(s' \mid s, u)$

Goal is to maximise *total return* per episode: $J = \sum_t \gamma^t r_t$

Partially Observable Markov Decision Process (POMDP)



Partially Observable Markov Decision Process (POMDP)



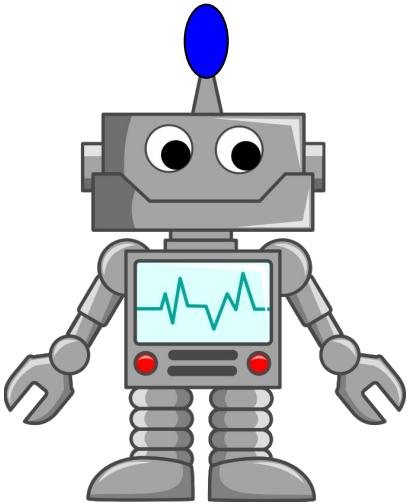
Partially Observable Markov Decision Process (POMDP)

$$\mathcal{B}(s) = P(s|\tau)$$

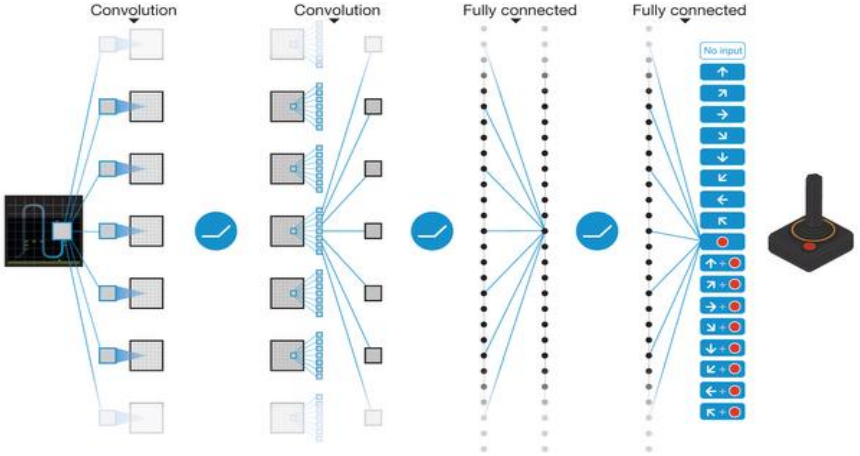
Bayes rule:

$$P(X \mid Y) = \frac{P(Y \mid X)P(X)}{P(Y)}$$

Deep Reinforcement Learning

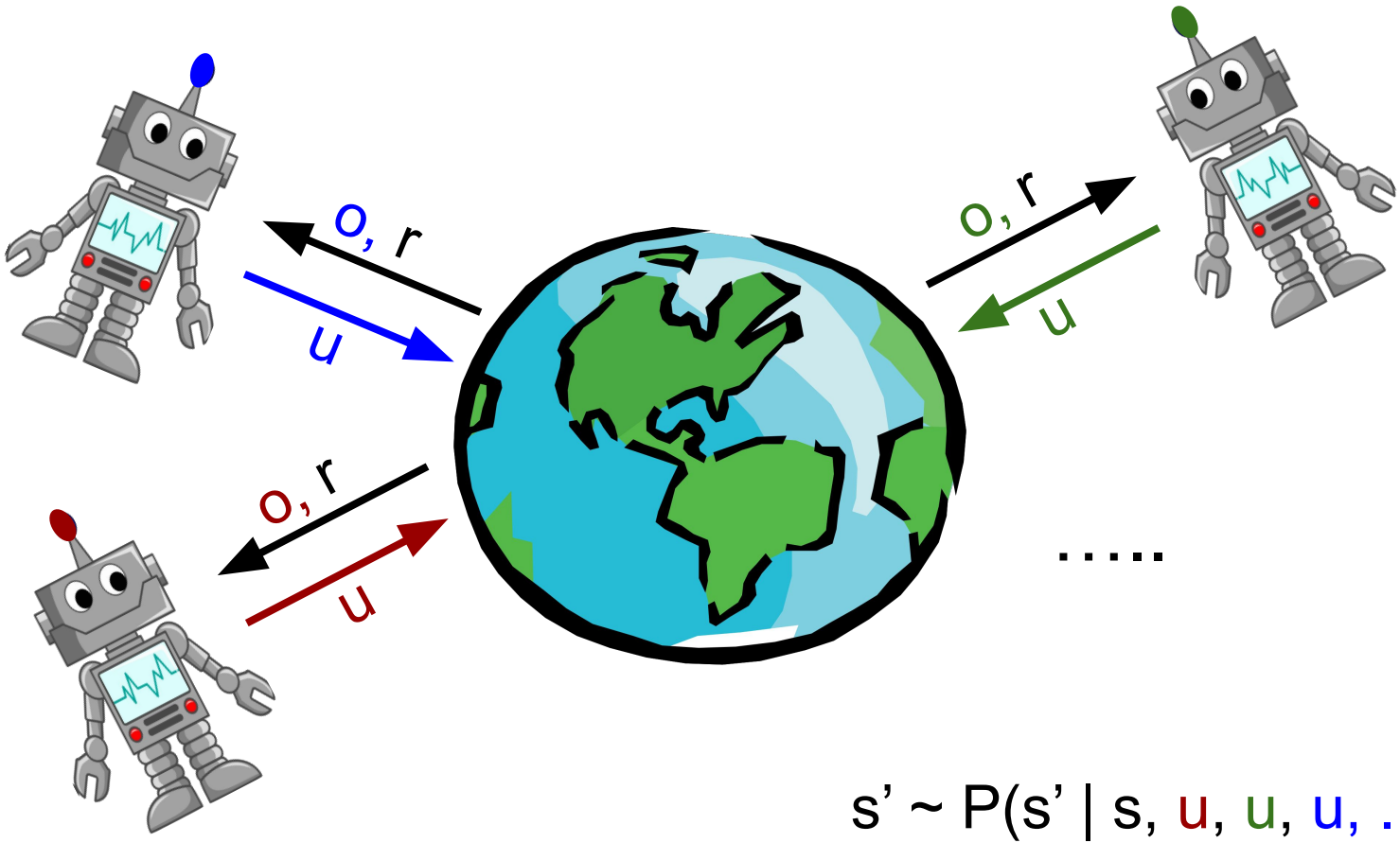


=

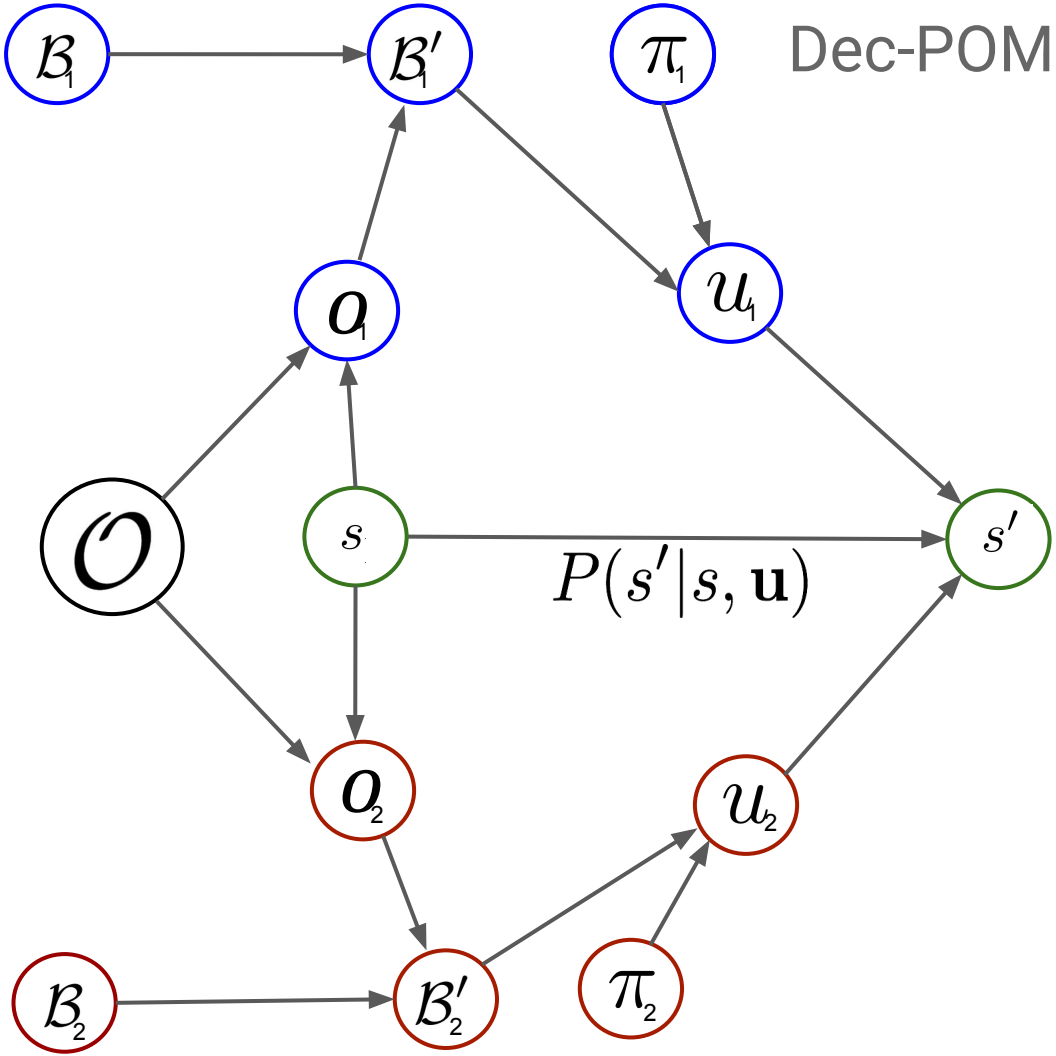


θ

Decentralized-POMDP (Dec-POMDP)

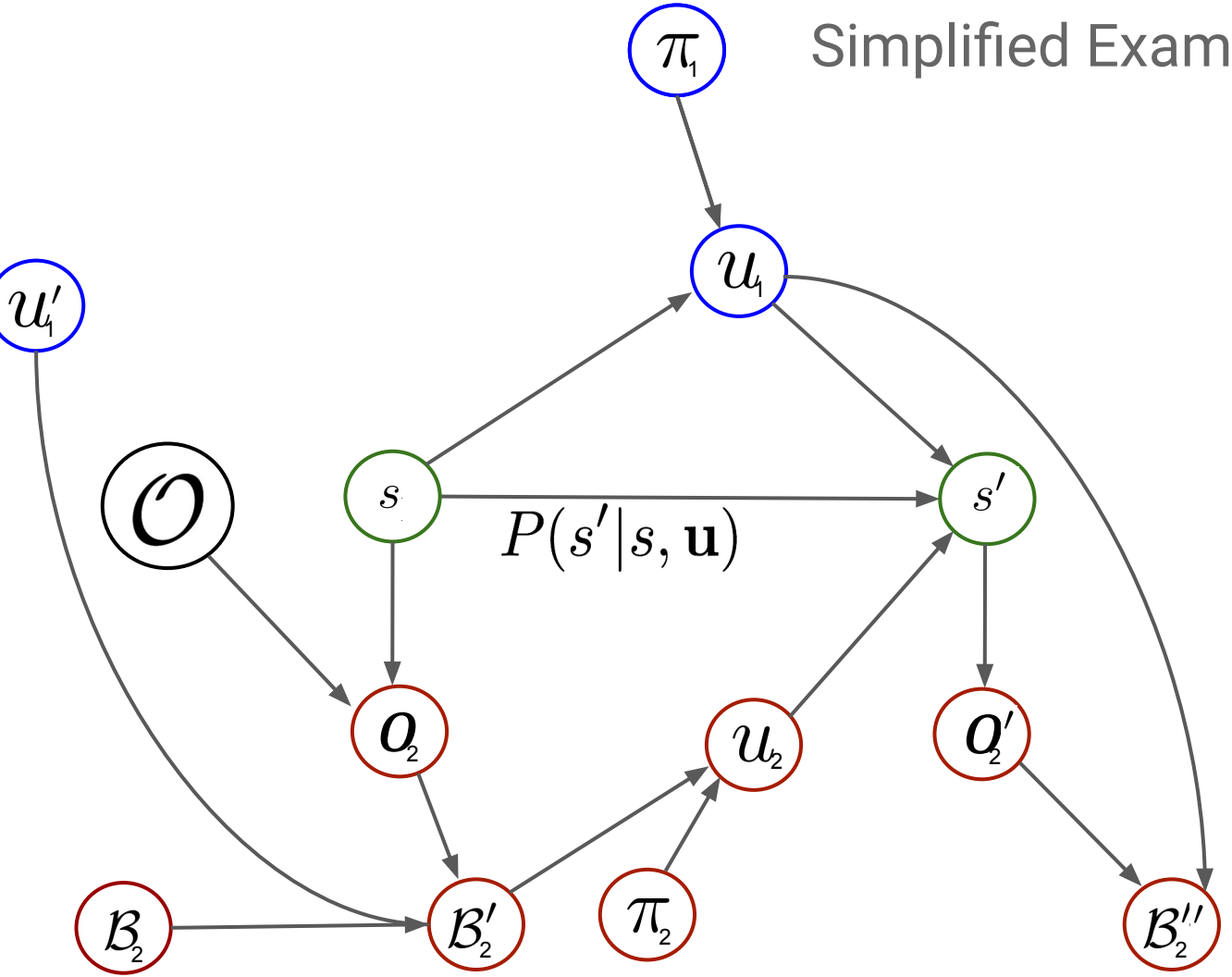


$s' \sim P(s' \mid s, u, u, u, ..)$


$\mathcal{B}_1$
$\mathcal{B}'_1$
π_1
o_1
u_1
$\mathcal{O}$
s
s'
$P(s'|s,\mathbf{u})$
o_2
u_2
$\mathcal{B}_2$
$\mathcal{B}'_2$
π_2

Dec-POMDP

Simplified Example



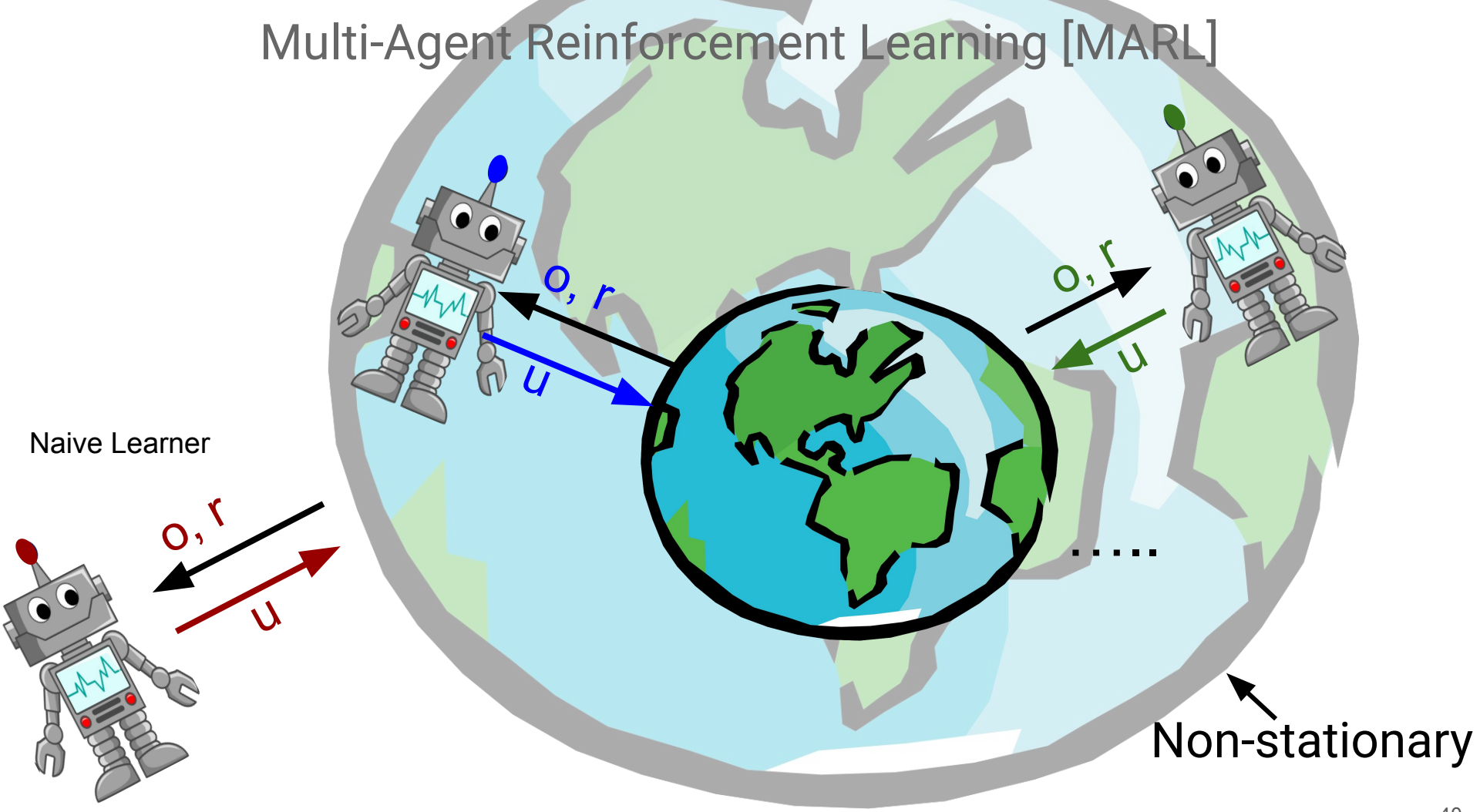
Simplified Dec-POMDP

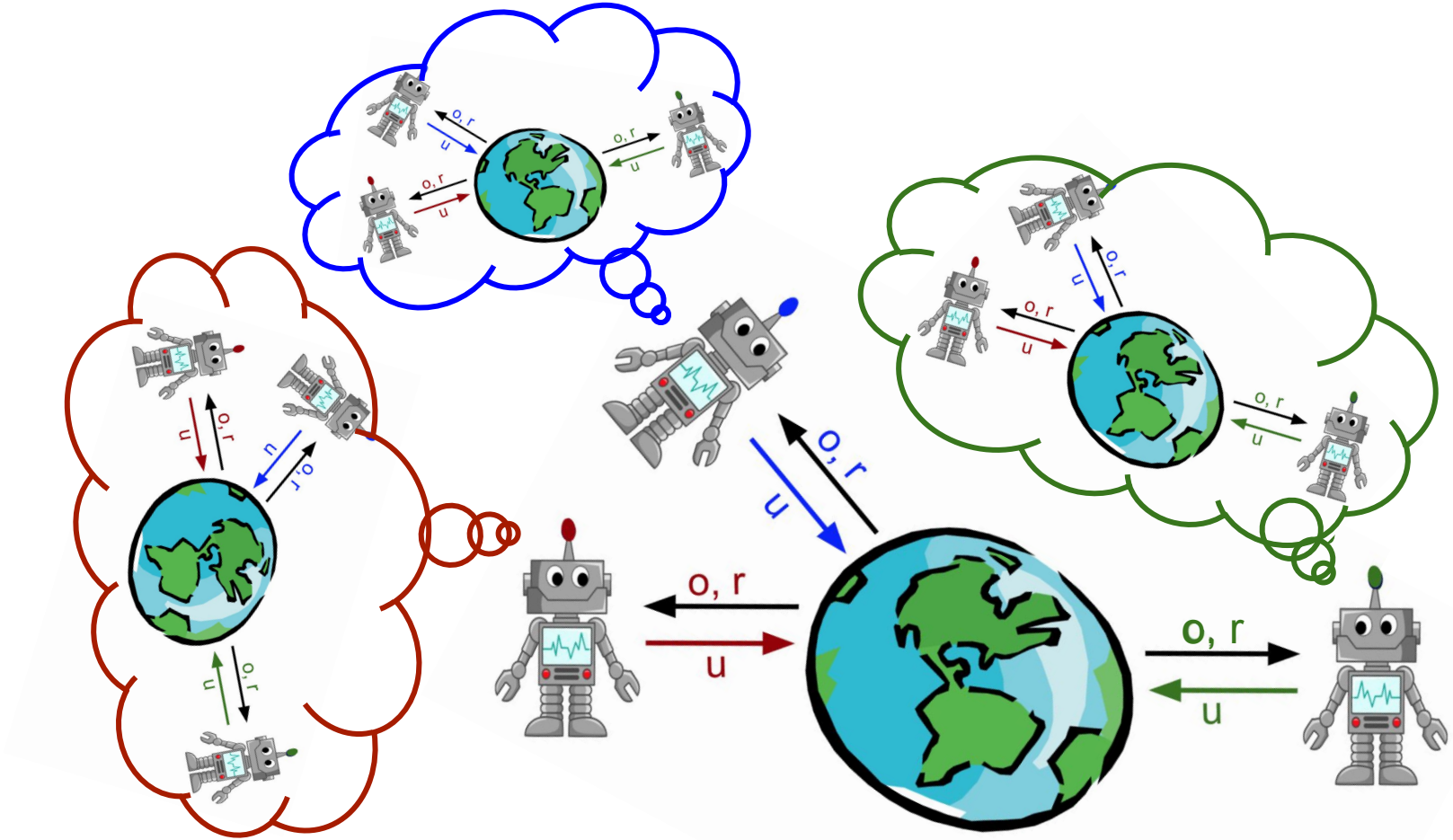
$$\mathcal{B}_2(s) = P(s_2|\tau_2)$$

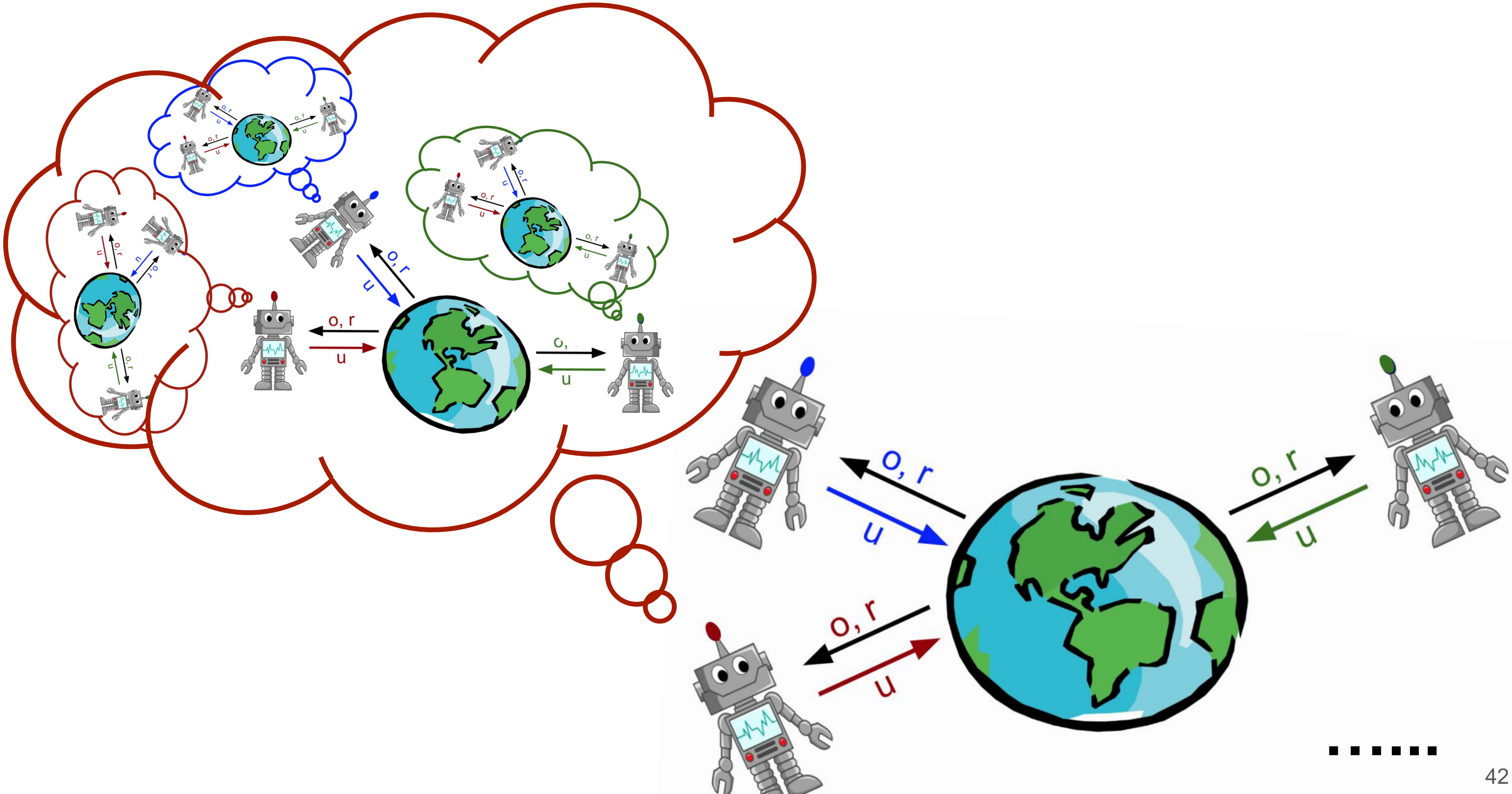
π_1 acts like an *observation function* for agent 2!

$$P(X \mid Y) = \frac{P(Y \mid X)P(X)}{P(Y)}$$

Multi-Agent Reinforcement Learning [MARL]




o, r
u
o, r
u
o, r
u
o, r
u
o, r
u
o, r
u
o, r
u
o, r
u
o, r
u
o, r
u
o, r
u
o, r
u


o, r
u
o, r
u
o, r
u
o, r
u
o, r
u
o,
u

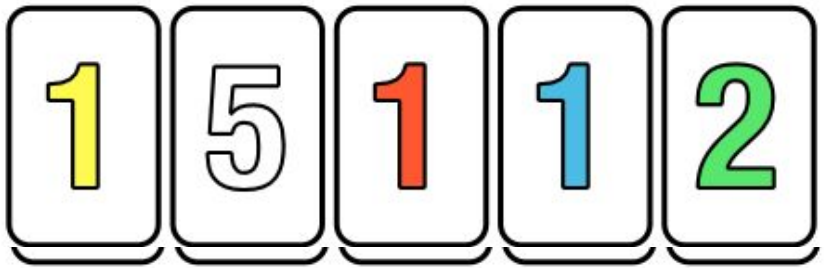
Hanabi!

Fully cooperative

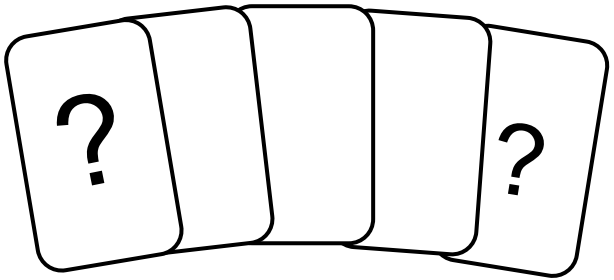
Partially Observable

Hanabi

"Fireworks"



Your cards (hand)



Your friend's cards (hand)

Some possible hints:
"Your 1st and 5th cards are twos"
"Your second card is red"
"Your 4th card is the only four"
"Your 3rd and 5th card are while"
"..."

Hanabi

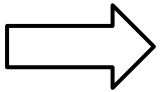
The Hanabi Challenge: A New Frontier for AI Research
Nolan Bard*, Jakob N. Foerster*, Sarath Chandar, Neil Burch, Marc Lanctot, H. Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, Iain Dunning, Shibl Mourad, Hugo Larochelle, Marc G. Bellemare, Michael Bowling
Artificial Intelligence, 2020

Part I: Self-Play

Self-Play:
Train a team of agents to maximise expected return when test *as the same team*

$$\pi^* = \arg\max_{\pi} J(\pi^1, \pi^2)$$

Training



Testing

Bayesian **A**ction **D**ecoder (BAD) for Deep Multi-Agent RL

Jakob Foerster*, Francis Song*
Ed Hughes, Neil Burch, Iain Dunning, Shimon Whiteson, Matt Botvinick, Mike Bowling

ICML 2019 (https://arxiv.org/abs/1811.01458)

UNIVERSITY OF OXFORD

DeepMind

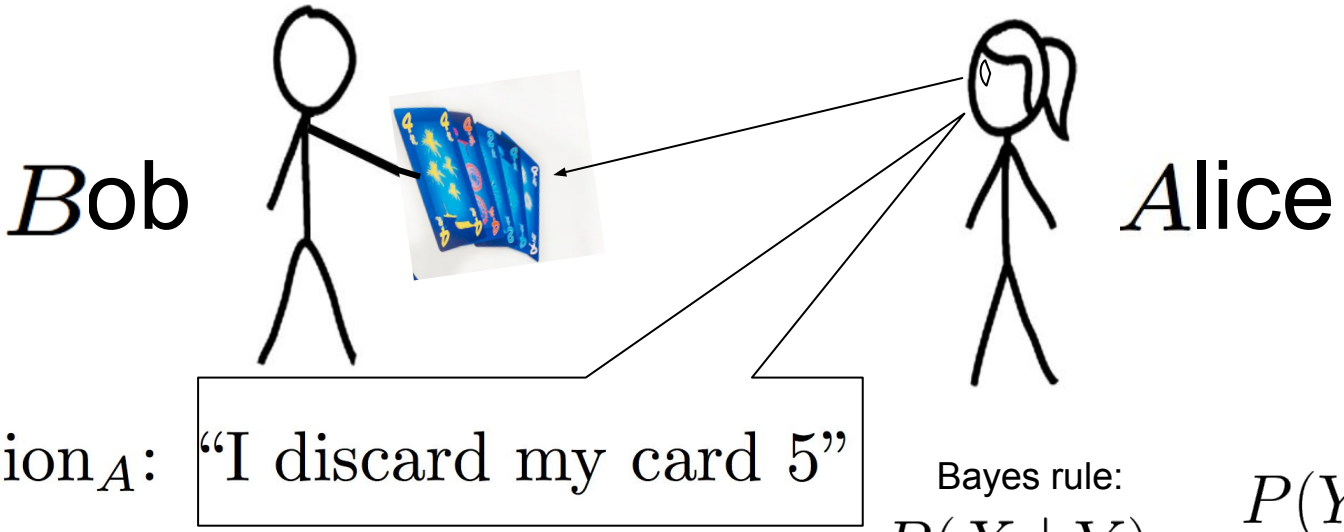
(PhD) 

(intern 2017) 

Bayesian Reasoning and Communication

simplified!!

$$P(h_B|u_A) = \frac{P(u_A|h_B)P(h_B)}{\sum_{h'_B} P(u_A|h'_B)P(h'_B)} = \frac{\pi_A(u_A|h_B)P(h_B)}{\sum_{h'_B} \pi_A(u_A|h'_B)P(h'_B)}$$



Bayes rule:

$$P(X \mid Y) = \frac{P(Y \mid X)P(X)}{P(Y)}$$

$$s' \sim P(s' \mid s, u)$$

$$B_B = P(h_B)$$

$$B_B' = P(h_B' \mid B_B, \pi, u_A)$$

what happened

Future depends not just *what* happened, but *why* it happened.

$$B_B' = P(h_B' \mid B_B, \pi, u_A)$$

Future depends not just *what* happened, but *why* it happened.

⇒ Need to explore over *policies,* rather than actions

⇒ Need to compute correct belief for each of the policies being explored

Problem: Beliefs over beliefs

$$P(h_B|u_A) = \frac{P(u_A|h_B)P(h_B)}{\sum_{h'_B} P(u_A|h'_B)P(h'_B)} = \frac{\pi_A(u_A|h_B)P(h_B)}{\sum_{h'_B} \pi_A(u_A|h'_B)P(h'_B)}$$

simplified!!

$$\mathcal{B}_{\mathcal{A}} = P(h_A)$$

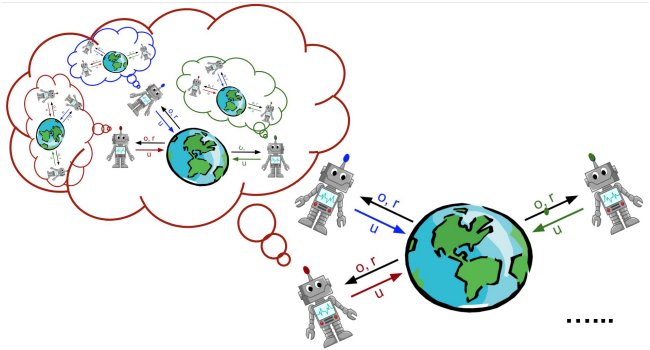
$$P(h_B|u_A) = \frac{\pi_A(u_A|h_B, \mathcal{B}_{\mathcal{A}}, \text{Pub})P(h_B)}{\sum_{h'_B} \pi_A(u_A|h'_B, \mathcal{B}_{\mathcal{A}}, \text{Pub})P(h'_B)}$$

Full input

⇨ Bob needs to reason over beliefs of Alice and vice versa.

One “solution”: Recursive beliefs

- Interactive POMDP: Model beliefs, beliefs over beliefs, beliefs over beliefs over beliefs, etc. (Gmytrasiewicz & Doshi, 2005).
- Intractable even with *finite nesting* of beliefs for all but the tiniest problems.



Alternatively, we can use **common knowledge**

(Lewis, 1969; Thomas et al., 2014).

Public and private features in Hanabi

f^{pri}

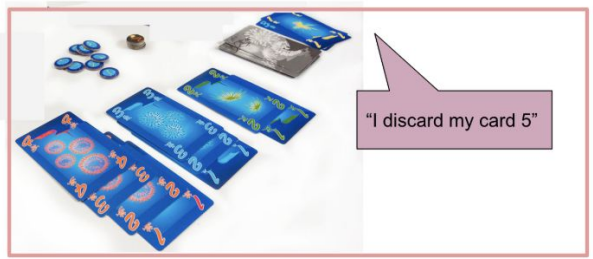
f^{pub}

Solution: Joint *Public* belief over *private* features

$$\mathcal{B}_A = P(\text{hand}_A | \text{hand}_B, f^{\text{pub}})$$
$$\mathcal{B}_B = P(\text{hand}_B | \text{hand}_A, f^{\text{pub}})$$

$$\Longrightarrow \quad \mathcal{B}_t = P(\text{hand}_A, \text{hand}_B \,|\, f^{\text{pub}})$$

$$\mathcal{B}_t = P(f^{\text{pri}} \;\; \text{P1:} \;\; \text{P2:} \;\; | \;\; f^{\text{pub}} \;\;)$$



Conditions only on **publicly available information**, so can be computed independently by every agent to get the same result.

Example Public Belief

$$\mathcal{B}_t = \left\{ \begin{array}{ll} \text{hand}_A, \text{hand}_B & P(\text{hand}_A, \text{hand}_B) \\ \textbf{23344, 11122}: & \textbf{\textit{0.0000}} \\ \textbf{23344, 11212}: & \textbf{\textit{0.0233}} \\ \vdots & \vdots \\ \textbf{44444, 23344}: & \textbf{\textit{0.0013}} \\ \textbf{44444, 55555}: & \textbf{\textit{0.0005}} \end{array} \right\}$$

Exploration in the space of deterministic policies

- **Informative actions** require **low** entropy in the policies, ideally ***deterministic*** policies (in hand_B)

$$P(\text{hand}_B|\text{action}_A) \propto P(\text{action}_A|\text{hand}_B)P(\text{hand}_B)$$

- **Exploration and differentiation** require **high** entropy (randomness) in the policies.

Public agent uses only public belief + public observation to sample a **deterministic partial policy** that tells the acting agent what to do for *any private observation*:

I.e. (for Alice)

$$\hat{\pi} : \{f^a\} \to \mathcal{U}$$

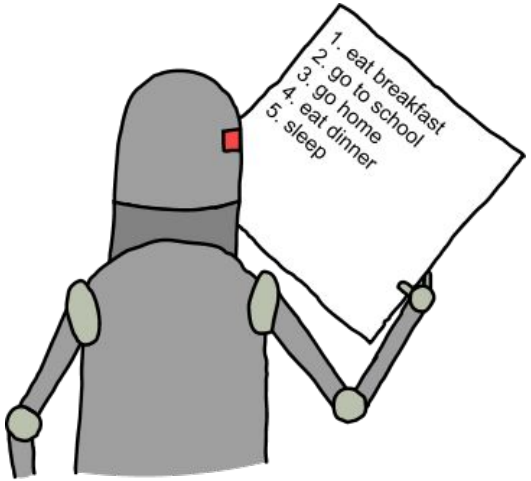
$$\hat{\pi} : \{\text{hand}_B\} \to \{\text{action}_A\}$$

Deterministic

$$\hat{\pi} \sim \pi^{\theta}_{\text{BAD}}(\hat{\pi}|\mathcal{B}_t, f^{\text{pub}})$$

Stochastic

Deterministic policies = Lists of instructions



$$\overset{\Delta}{\pi} : \{f^a\} \rightarrow \mathcal{U}$$

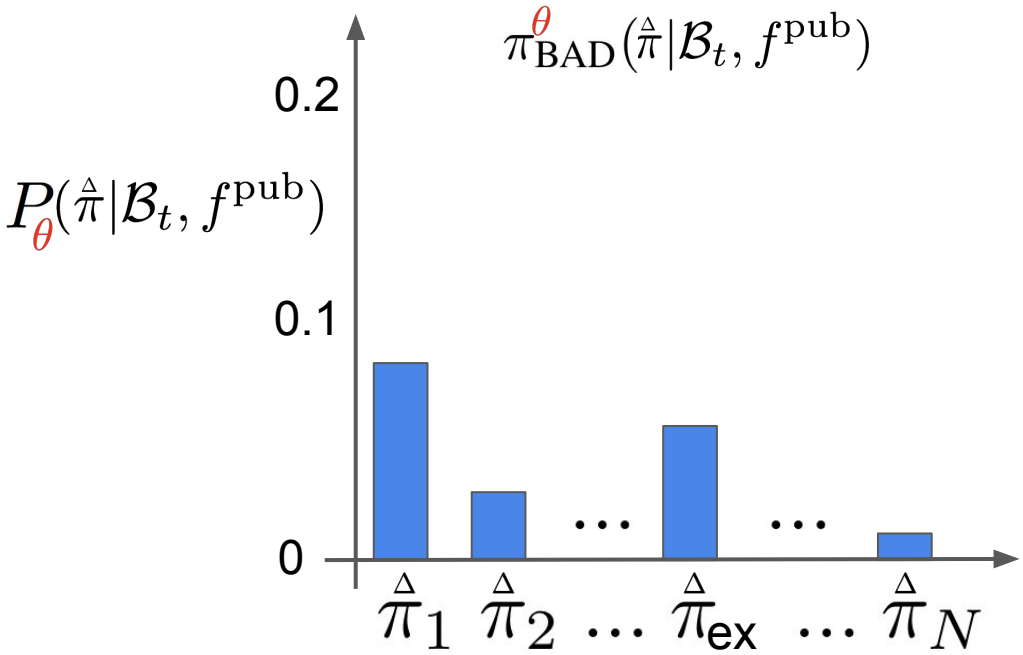
Example:

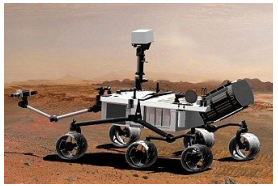
$$\overset{\Delta}{\pi} = \left\{ \begin{array}{ccc} 11122 & \rightarrow & \textit{Play 5th card} \\ 11212 & \rightarrow & \textit{Hint blue} \\ \vdots & & \vdots \\ 55555 & \rightarrow & \textit{Discard 3rd card} \end{array} \right\}$$

$$\overset{\Delta}{\pi}(11122) = \textit{Play 5th card}$$

$$\vdots$$

Probability over Deterministic Maps





Bayesian Update

$$\mathcal{B}_{t+1} = P(f^{\text{pri}} | u_t^a, \hat{\pi}, f'^{\text{pub}}, \mathcal{B}_t)$$

$$\mathcal{B}_t = \left\{ \begin{array}{ll} \text{hand}_A, \text{hand}_B & P(\text{hand}_A, \text{hand}_B) \\ \text{23344, 11122}: & 0.0000 \\ \text{23344, 11212}: & 0.0233 \\ \vdots & \vdots \\ \text{44444, 23344}: & 0.0013 \\ \text{44444, 55555}: & 0.0005 \end{array} \right\} + \hat{\pi} = \left\{ \begin{array}{ll} \text{11122} & \rightarrow \text{Play 5th card} \\ \text{11212} & \rightarrow \text{Hint blue} \\ \vdots & \vdots \\ \text{55555} & \rightarrow \text{Discard 3rd card} \end{array} \right\} + u_A = \text{Discard 3rd card}$$

$$\Rightarrow \mathcal{B}_{t+1} = \left\{ \begin{array}{ll} \text{hand}_A, \text{hand}_B & P(\text{hand}_A, \text{hand}_B) \\ \text{23344, 11122}: & \sout{0.0000} \\ \text{23344, 11212}: & \sout{0.0233} \\ \vdots & \vdots \\ \text{44444, 23344}: & \sout{0.0013} \\ \text{44444, 55555}: & 0.0025 \end{array} \right\}$$

Bayesian Action Decoder (BAD) full picture

1) Sample ‘list of instructions’ given public info and beliefs:

$$\overset{\triangle}{\pi} \sim \pi^{\theta}_{\text{BAD}}(\overset{\triangle}{\pi} | \underbrace{\mathcal{B}_t, f^{\text{pub}}}_{s_{\text{BAD}}})$$

2) Acting agent, a , evaluates $\overset{\triangle}{\pi}$ for their private observation:

$$u_t^a = \overset{\triangle}{\pi}(f_t^a)$$

3) Action, u_t^a, is observed by all players and sent to environment:

$$s' \sim P(s' | s, u_t^a)$$

4) Action and ‘list of instructions’ induce a Bayesian update:

$$\mathcal{B}_{t+1} = P(f^{\text{pri}} | u_t^a, \overset{\triangle}{\pi}, f'^{\text{pub}}, \mathcal{B}_t)$$

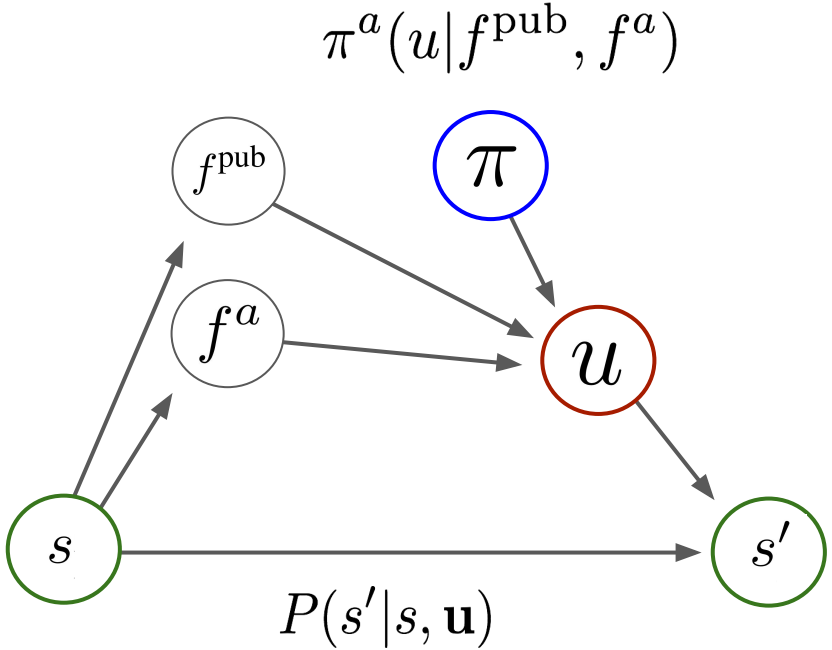
Markov Decision Process with public and private features

f^{pub} Public features: common (**vs. shared**) knowledge to all agents (e.g., actions, cards on the table).

f^{pri} Private features: observable to at least one, but not all, agents (e.g., cards held by each player).

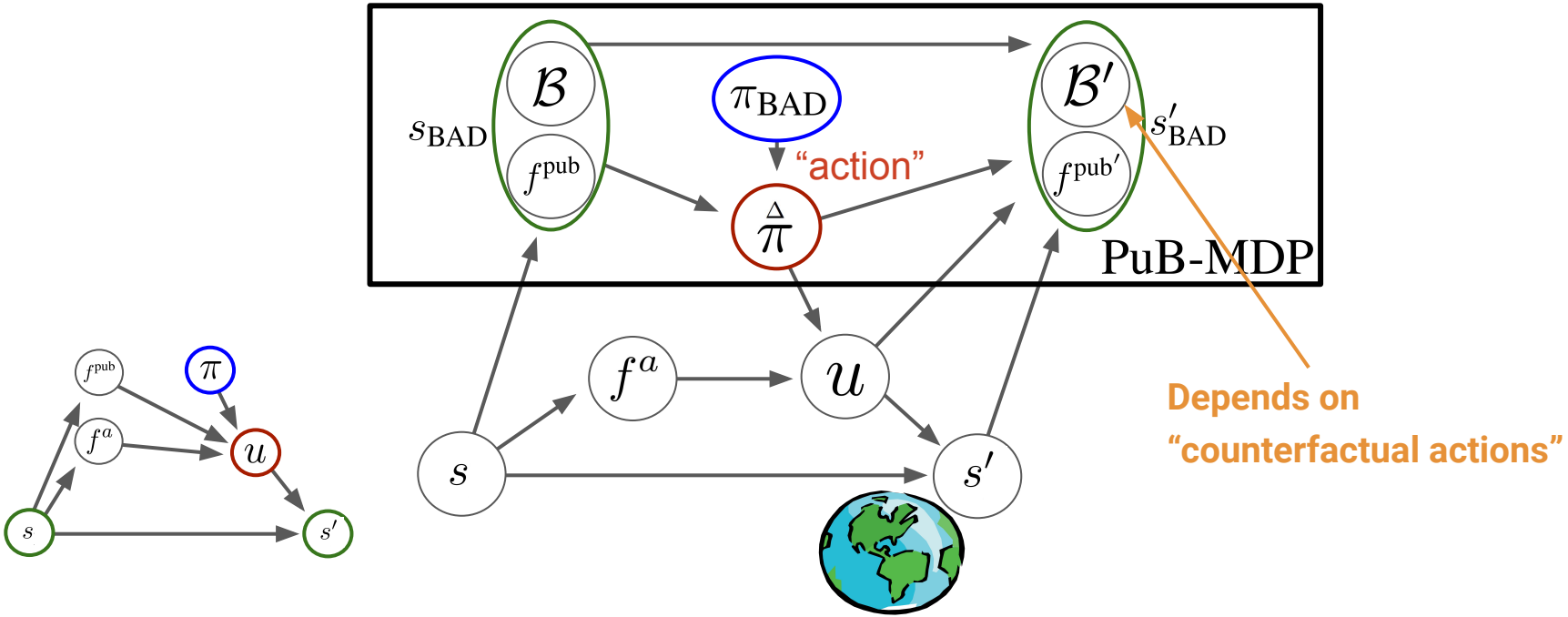
$\downarrow$

f^{a} Cards visible to agent *a*.



Public belief MDP (PuB-MDP)

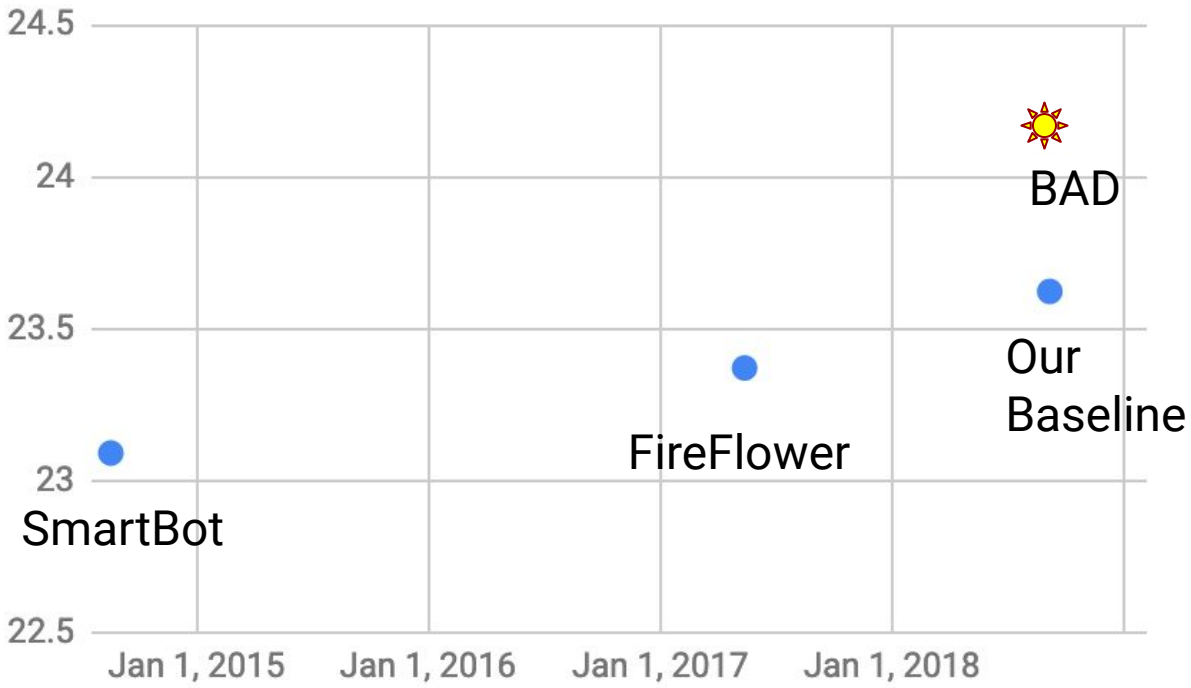
Transition function $P(s'_{\text{BAD}} | s_{\text{BAD}}, \hat{\pi})$



Depends on "counterfactual actions"

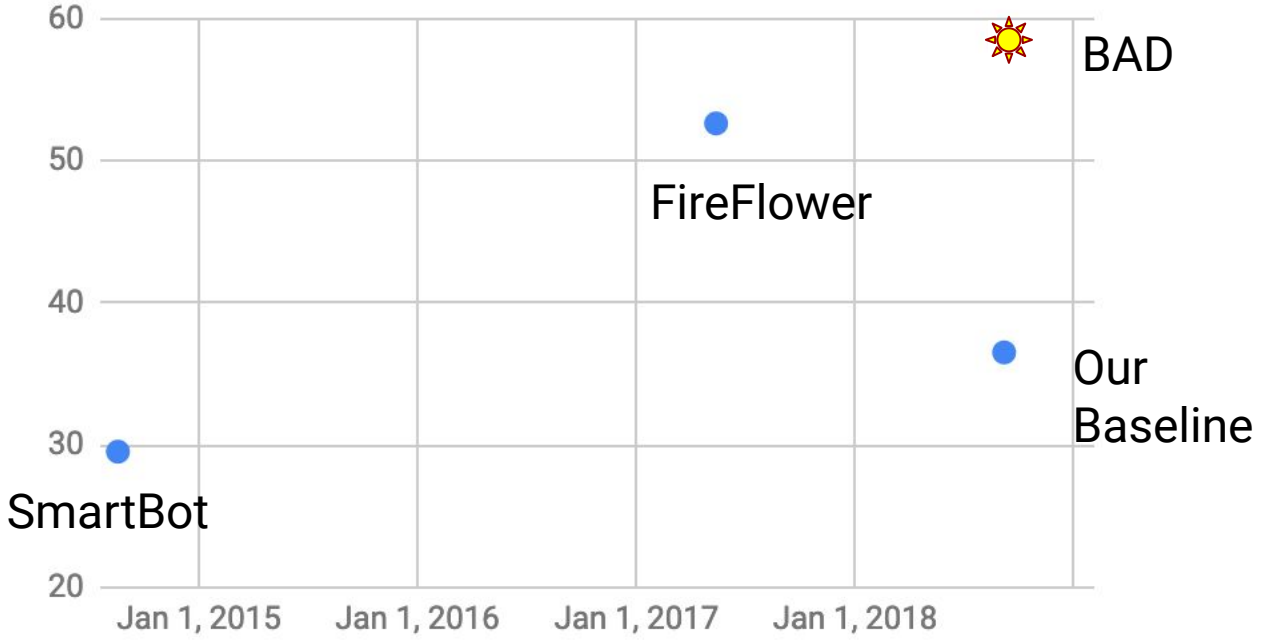

RESULTS

Hanabi



github.com/lightvector/fireflower
github.com/Quuxplusone/Hanabi

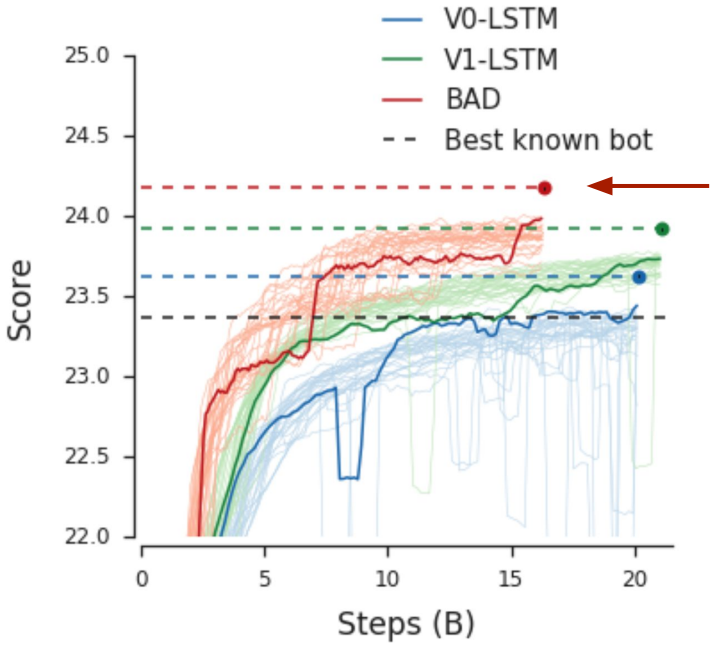
Hanabi



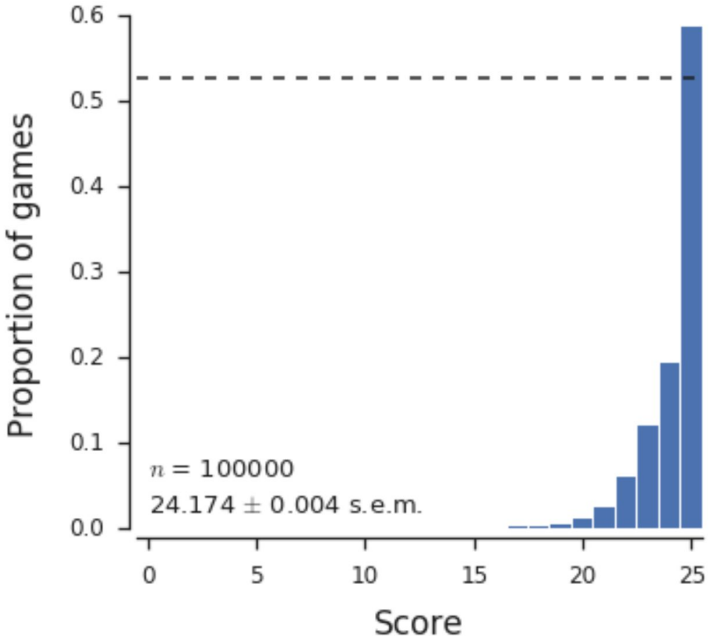
github.com/lightvector/fireflower
github.com/Quuxplusone/Hanabi

Hanabi

Training curves



Histogram of scores



Gameplay

Notable conventions:

- Hint Red or Yellow: ”Play newest card.”
- Hint White and Blue: “Probably discard last card”

*“The bot is *very* strong in the early game, and there its convention set is overall far more efficient than ‘natural’ human convention sets … It’s really quite beautiful”*

David Wu (creator of FireFlower)

Progress on Self-Play Since

"Simplified Action Decoder for Deep Multi-Agent Reinforcement Learning "

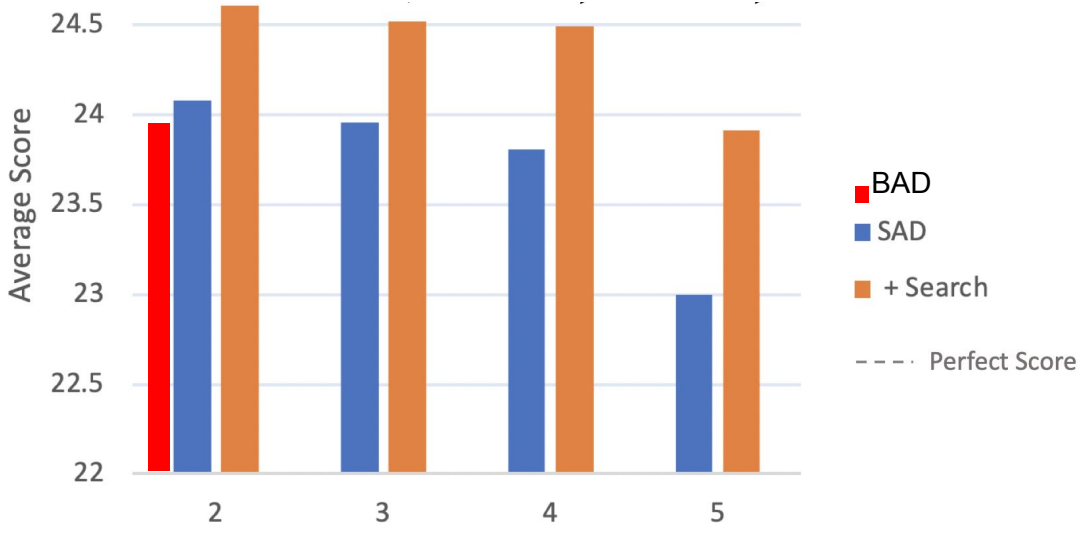
H Hu, **JN Foerster**

International Conference on Learning Representations, 2020

"Improving Policies via Search in Cooperative Partially Observable Games "

A Lerer, H Hu, **JN Foerster**, N Brown

AAAI Conference on Artificial Intelligence, 2020



	2	3	4	5
Prior SOTA: ACHA (Bard et al., 2019)	$20.24 \pm 1.1\%$ 1.1%	21.57 ± 0.12 2.4%	16.80 ± 0.13 0%	

Part II: Zero-Shot Coordination

Zero-Shot: Train team of agents to maximise expected return when tested *with an independently trained version*

Training

Testing in *Cross-Play*

Coaches can agree on a training strategy *before* training starts.

What should the strategy be?

A Surprisingly hard Problem

Our Objective is Cross-Play:

So, why not optimize it directly?

We are not allowed to evaluate this objective during training!

It's easy to tell *after* training if you failed.
No known (scalable) algorithm exists to solve this problem in general.

“Other-Play” for Zero-Shot Coordination

Hengyuan Hu*, Adam Lerer, Alex Peysakhovich, Jakob Foerster*

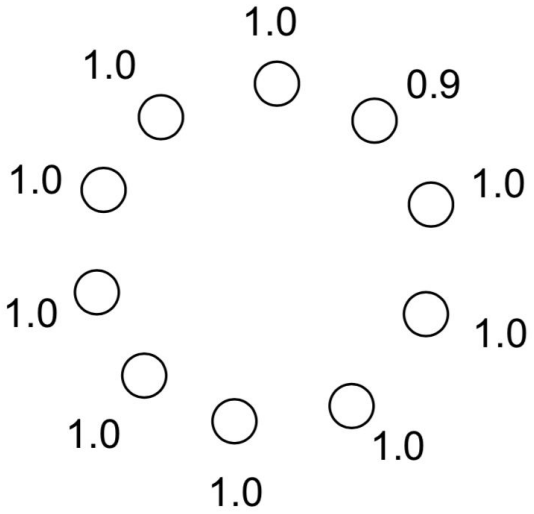
ICML 2020
https://arxiv.org/abs/2003.02979

facebook Artificial Intelligence

Formal Problem Definition

“Suppose that multiple independent AI designers will construct agents that have to interact in various but ex-ante unknown Dec-POMDPs without being able to coordinate beforehand, what learning rule should these designers agree on?”

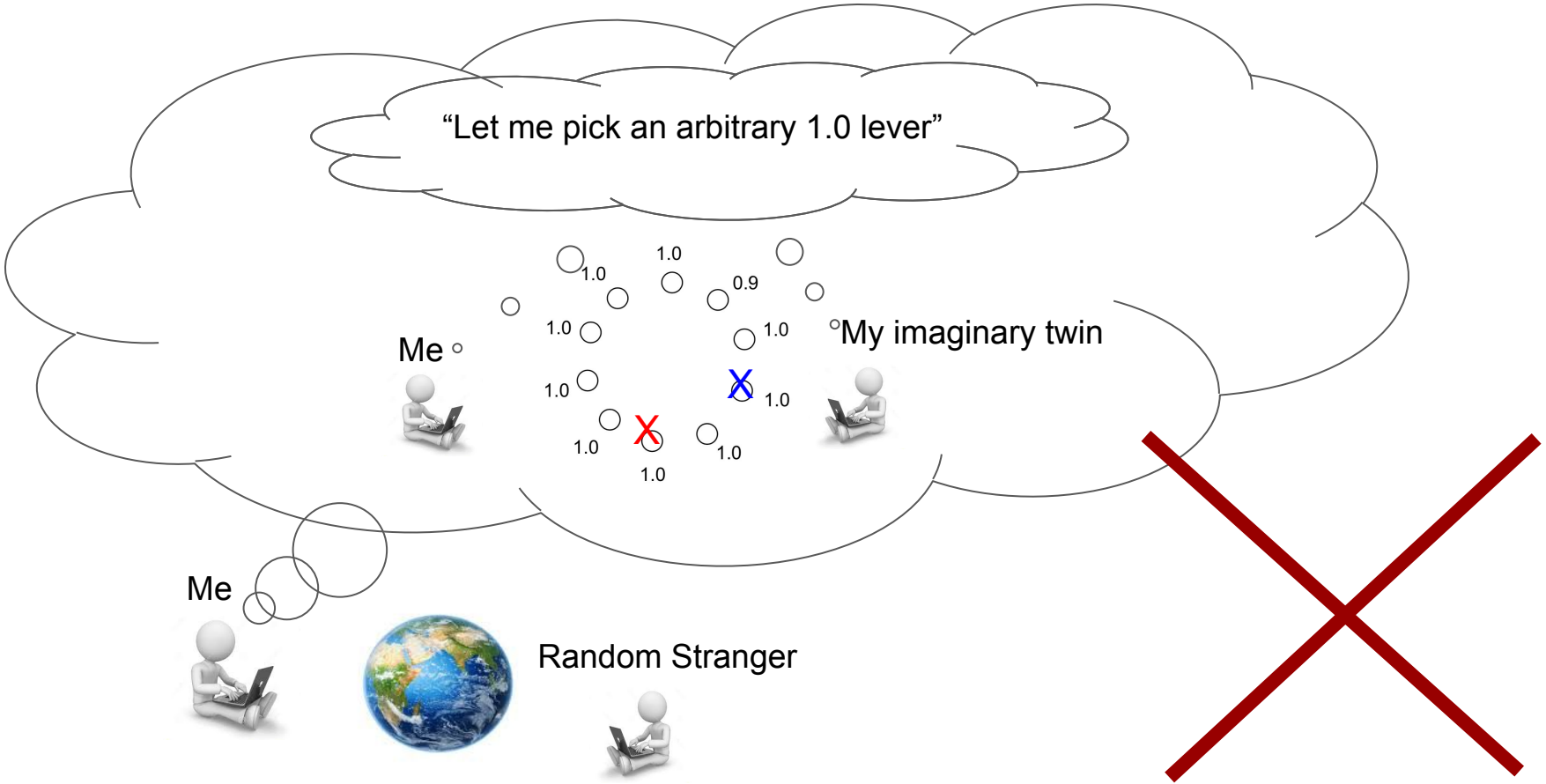
This is hard even in the simplest cases..

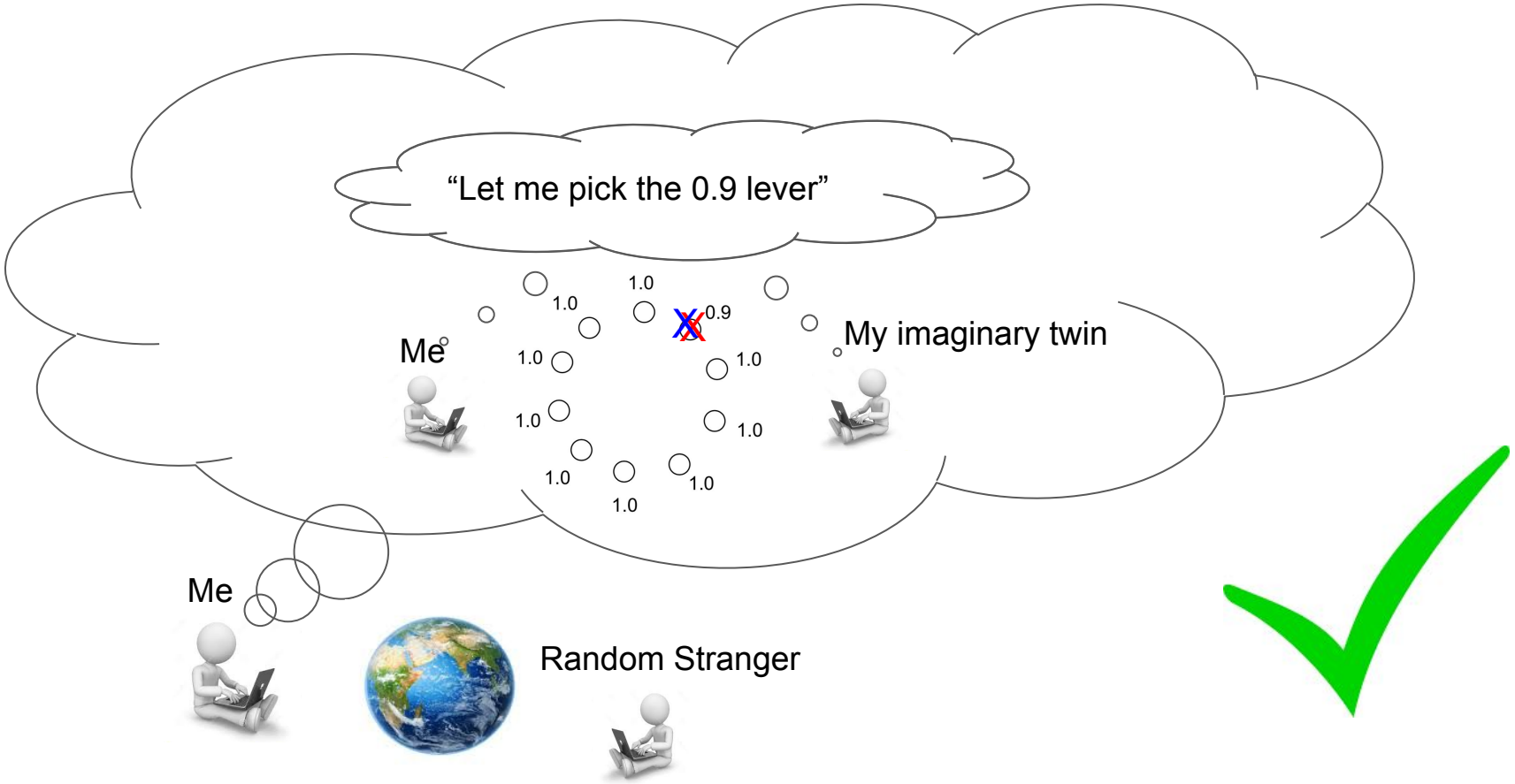


Task description:

- You need to cooperate with a random stranger by choosing one of the 10 levers
- If the two of you pick the same lever, both of you get the reward shown next to the lever in the image.
- Otherwise the reward is zero.
- You have a single attempt.
- The task description is common knowledge.

What do you do?


“Let me pick an arbitrary 1.0 lever”
1.0
1.0
0.9
1.0
1.0
1.0
1.0
1.0
1.0
1.0
Me
My imaginary twin
Me
Random Stranger


“Let me pick the 0.9 lever”
1.0
1.0
0.9
1.0
1.0
1.0
1.0
1.0
1.0
1.0
Me
My imaginary twin
Me
Random Stranger

Key Idea for our solution: Symmetries

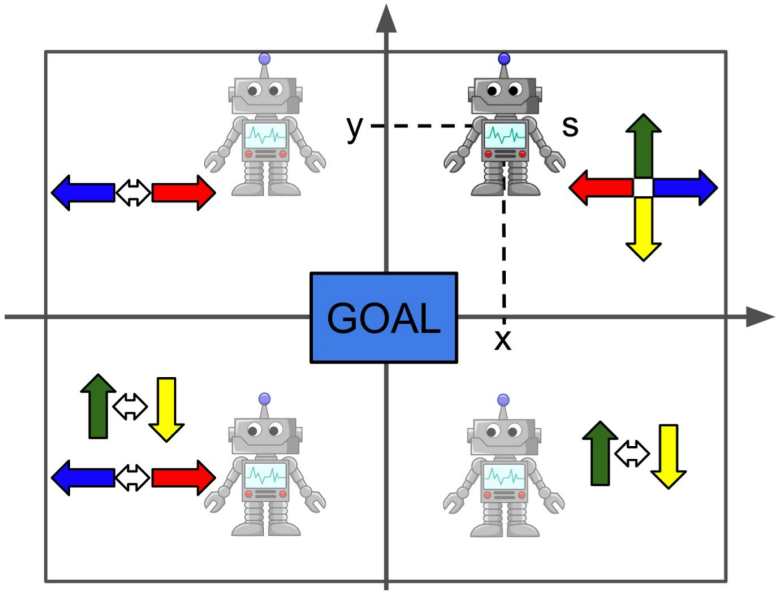
Symmetries are mappings that leave the Dec- POMDP unchanged:

$$\begin{aligned}\phi \in \Phi \iff & P(\phi(s')|\phi(s),\phi(a)) = P(s'|s,a) \\ & \wedge R(\phi(s'),\phi(a),\phi(s)) = R(s',a,s) \\ & \wedge O(\phi(o)|\phi(s),\phi(a),i) = O(o|s,a,i) \\ & \text{where equalities apply } \forall s', s, a'\end{aligned}$$

Can also apply this to policies:

$$\pi' = \phi(\pi) \iff \pi'(\phi(a)|\phi(\tau)) = \pi(a|\tau), \forall \tau, a$$

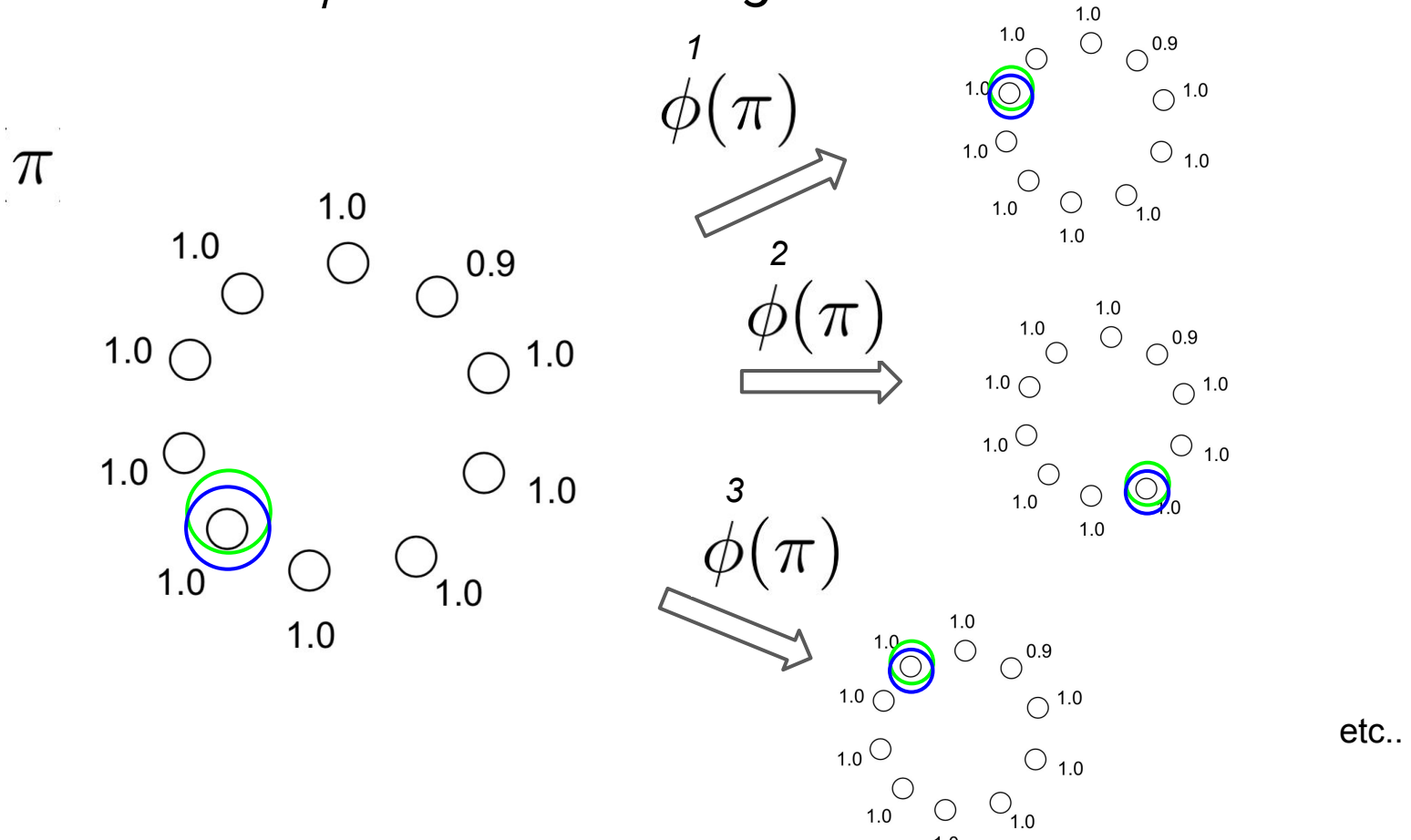
Toy Example



Here symmetries invert x-/y-axis *and* flip the corresponding action.

Example: Being in the top-right and moving down is *equivalent* to being the bottom right and moving up.

What does ϕ do in the lever game?



etc..

From Self-Play to Other-Play

Optimize both sides

Can specify partner (imaginary twin) only up to equivalence class

Self-Play:

$$\pi^* = \arg\max_{\pi} J(\pi^1, \pi^2)$$

Other-Play:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\phi\sim\Phi}\ J(\pi^1, \phi(\pi^2))$$

Proof:

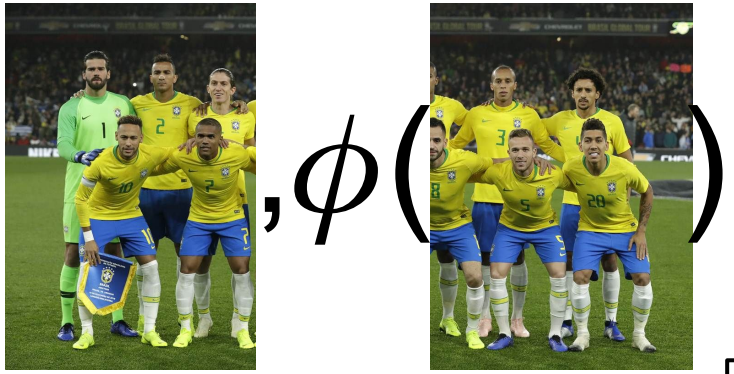
$$\begin{aligned} J_{OP}(\pi) &= \mathbb{E}_{\phi\sim\Phi}\ J(\pi^1, \phi(\pi^2)) \\ &= \mathbb{E}_{\phi_1\sim\Phi,\phi_2\sim\Phi}\ J(\phi_1(\pi^1), \phi_1(\phi_2(\pi^2))) \\ &= \mathbb{E}_{\phi_1\sim\Phi,\phi_2\sim\Phi}\ J(\phi_1(\pi^1), \phi_2(\pi^2)) \\ &= J(\pi_\Phi) \end{aligned}$$

Note 1*: OP is the best possible meta-equilibrium.*

Note 2: If Symmetries are known, OP can be implemented on top of any Deep RL algorithm.

Other Play as a picture:

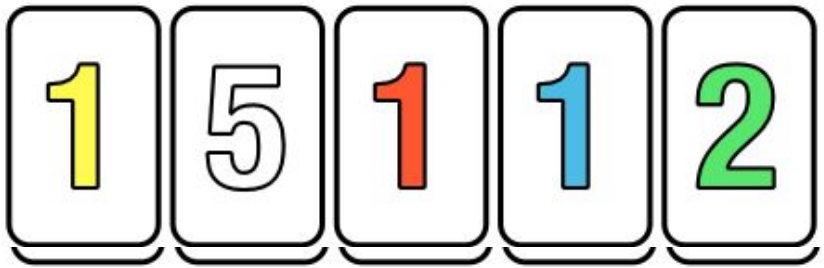
Training in Other-Play



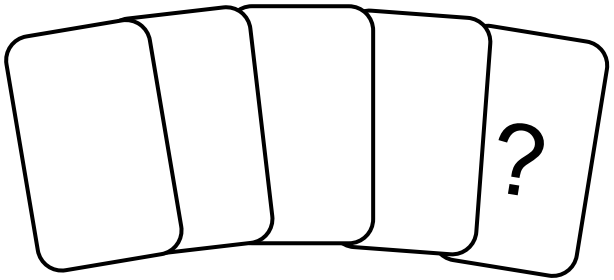
Testing in *Cross-Play*

Symmetries in Hanabi

“Fireworks”



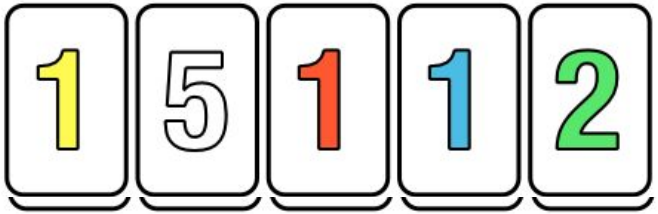
Your friend’s cards (hand)



All colors are equivalent.
Each ϕ is a permutation of the colors,
including Fireworks, cards, action space

Example: ϕ = {Y <-> W}

"Fireworks"



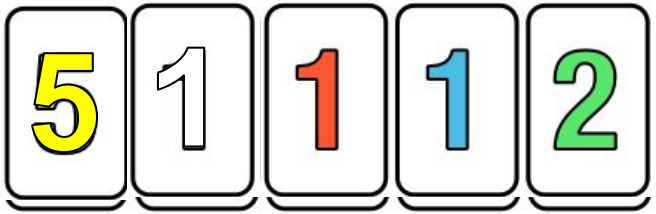
Hands

Hint:

"Your first card is *yellow*"

All players observe and act in the world according to different ϕ!

ϕ(Fireworks)



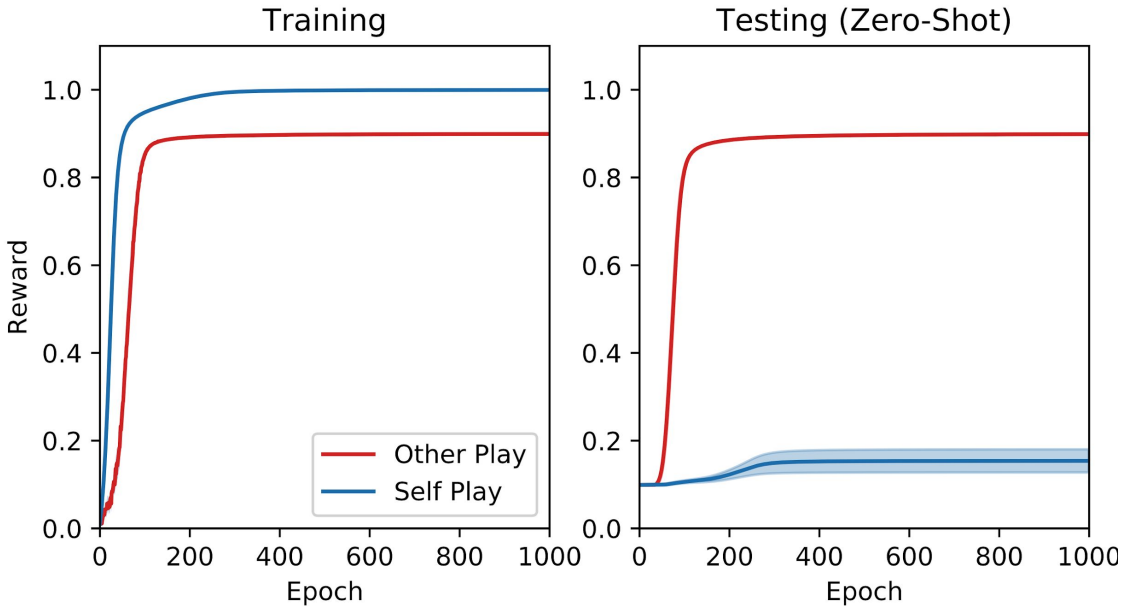
ϕ(Hand)

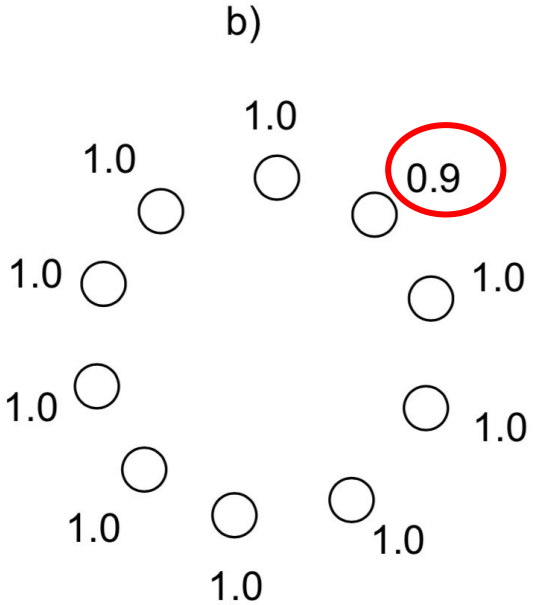
ϕ(Hint):

"Your first card is *white*"


RESULTS

Results: Matrix Game B)

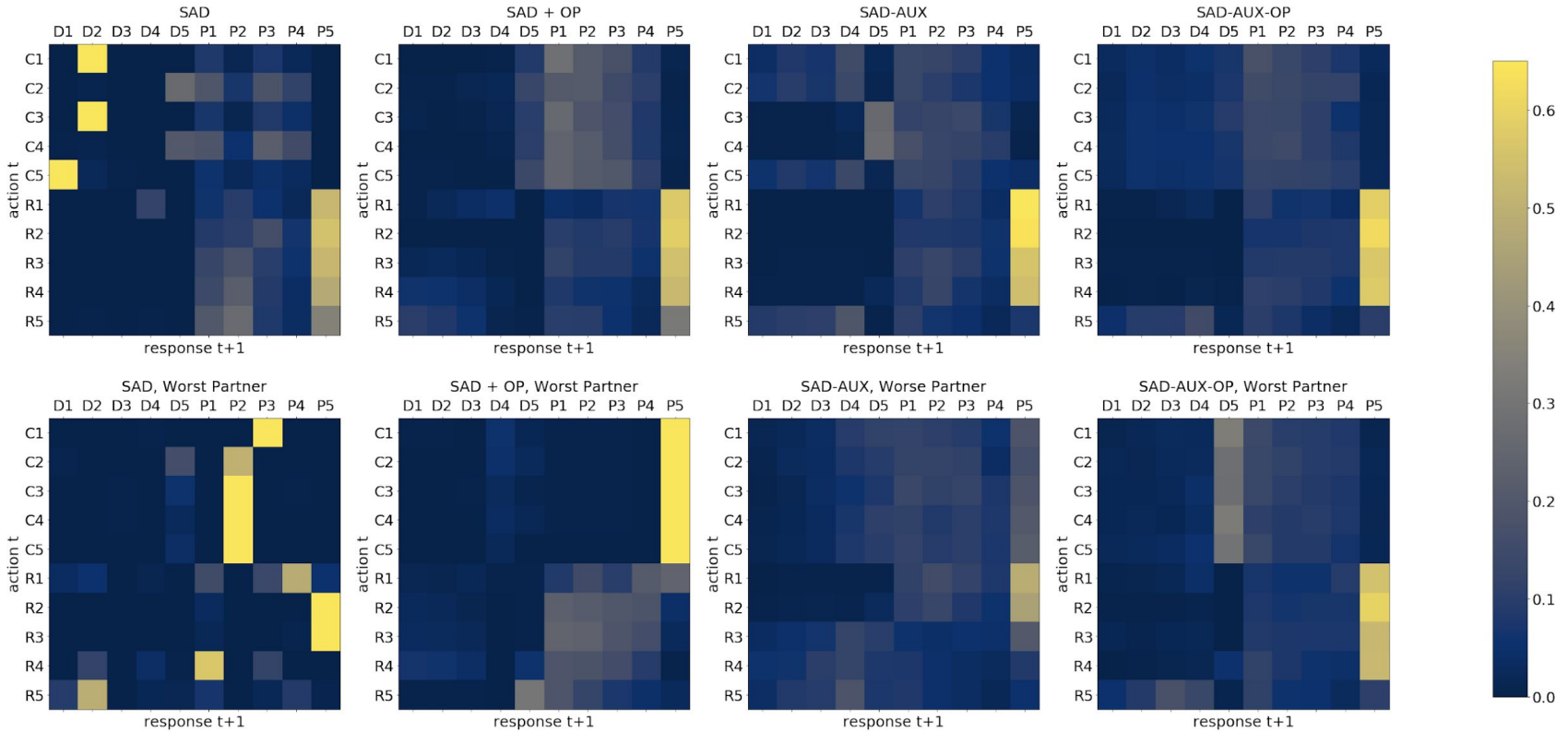




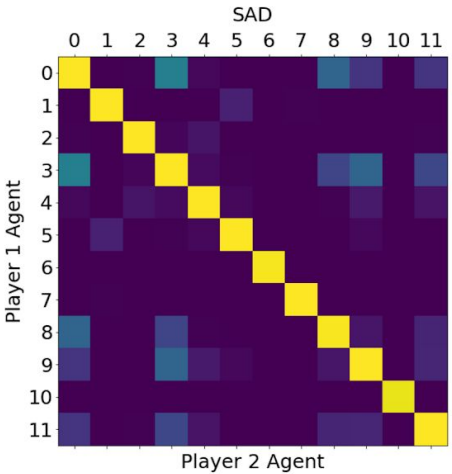
Results: Hanabi

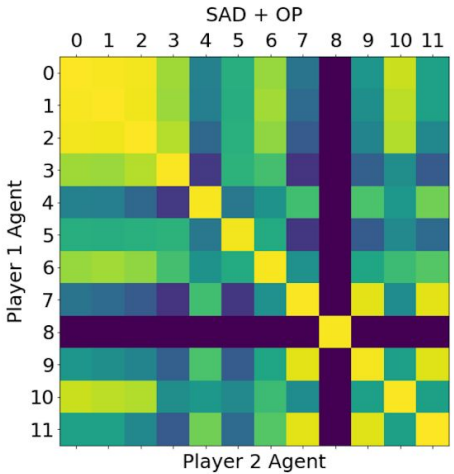
Method	Self-Play	Cross-Play
SAD	23.97 ± 0.04	2.52 ± 0.34
SAD + OP	23.93 ± 0.02	15.32 ± 0.65
SAD + AUX	$\mathbf{24.09 \pm 0.03}$	17.65 ± 0.69
SAD + AUX + OP	24.06 ± 0.02	$\mathbf{22.07 \pm 0.11}$

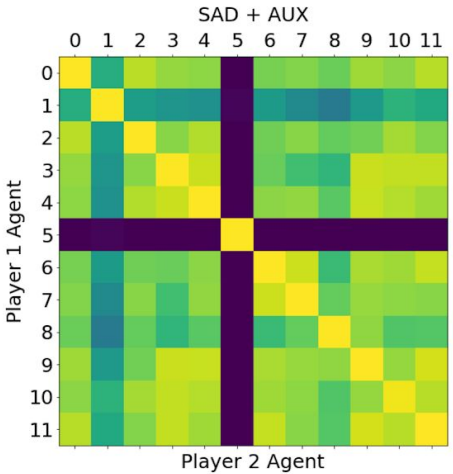
Conditional Action Plots - $P(u_{t+1} | u_t)$

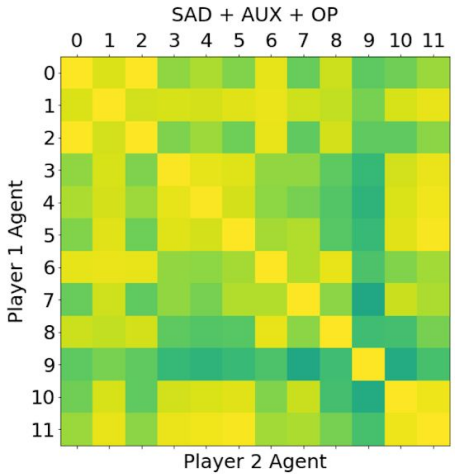


Cross-Play Matrices

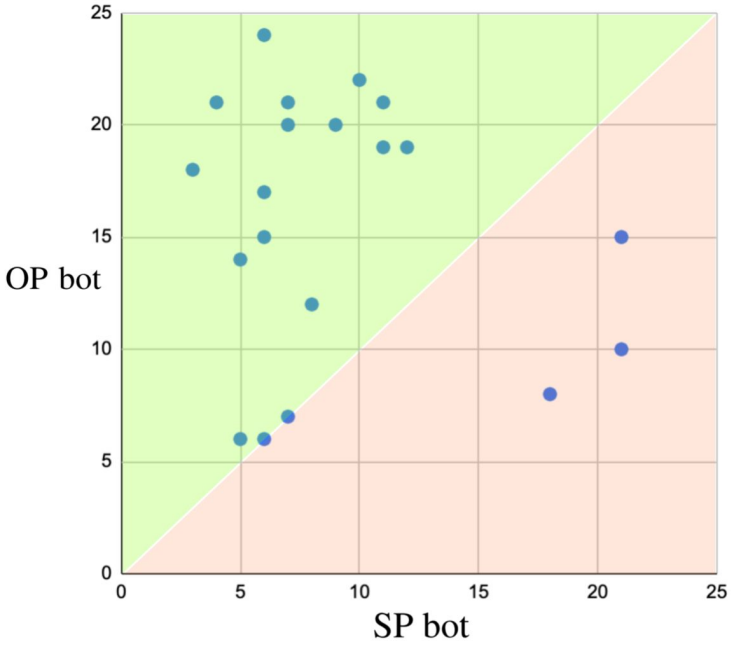








Results: Hanabi - Human AI Interaction!




Open Questions

What if we don’t know the Symmetries?

- Currently OP takes symmetries as input
- Humans manage to discover ‘reasonable’ policies without being provided the symmetries

How do we adapt on the fly?

- Humans excel at interpreting ‘intent’
- As a consequence, good human players can adopt to a broad range of policies within a single move.
- What’s the magic?

Can we break apart ‘understanding the problem’ and solving it?

- Clearly, RL ‘learns’ about the problem while settling on a solution.
- This is highly problematic in zero-shot coordination.

Can we learn Other-Play and similar Algorithms?

Code:

- Hanabi-Learning-Env:
  - github.com/deepmind/hanabi-learning-environmer
- Hanabi-Search:
  - github.com/facebookresearch/Hanabi_SPARTA
- SAD:
  - github.com/facebookresearch/Hanabi_SAD

Note1: SAD repo contains pre-trained RL agents!

Note 2: “Other-Play” code will be open-sourced in the future

Thank you for listening!

Questions?

The End