Tutorial: Multi-Agent Learning

D Balduzzi, T Graepel, E Hughes, M Jaderberg, S Omidshafiei, J Perolat, K Tuyls

DeepMind

Joint work with many great collaborators, including:


Daniel Hennes


Mark Rowland

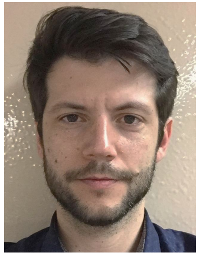

Wojciech Czarnecki


Remi Munos


Joel Z. Leibo


Sébastien Racanière

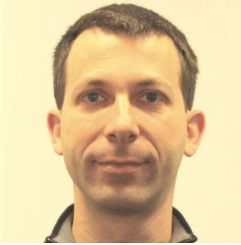

Christos Papadimitriou


Georgios Piliouras


Marc Lanctot


Dustin Morrill


Audrunas Gruslys


David Silver


Georg Ostrovski


Vinicius Zambaldi

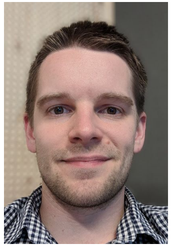

Jean-Baptiste Lespiau


Jakob Foerster

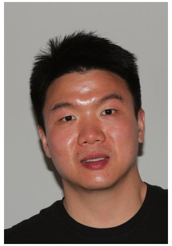

Guy Lever


James Martens


Edgar Duéñez-Guzmán


Luke Marris


Nicolas Heess


Zhe Wang


Edward Lockhart


Siqi Liu

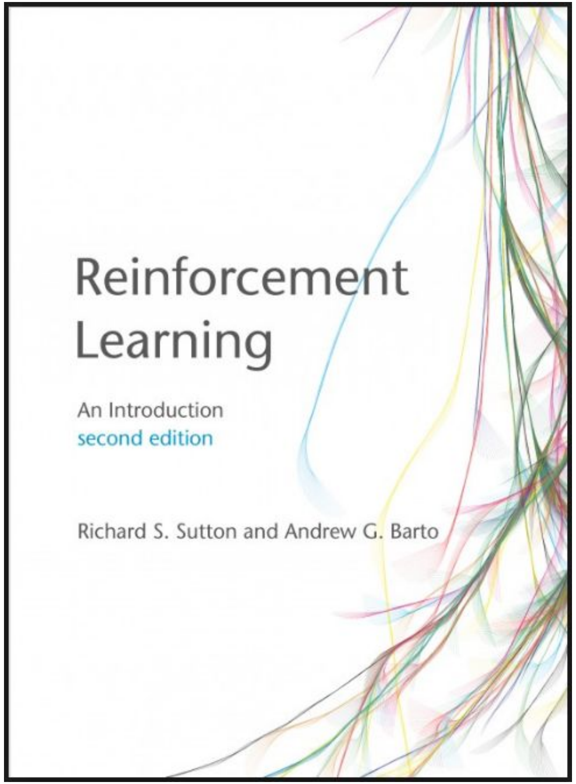

Michael Bowling


Finbarr Timbers

We won't cover ...

- Single Agent Reinforcement Learning
  - Markov Decision Processes
  - Algorithms

- A good resource though



Reinforcement Learning

An Introduction
second edition

Richard S. Sutton and Andrew G. Barto

Part I. Background & Theory

1. Introduction
2. NFGs and Markov Games
3. Social Learning

DeepMind

Part I: Background & Theory

- Motivation
- What is Multi-Agent Learning?
  - General Setup
  - Different Realizations: RL-based, Swarms, Evo-based
  - Role of (Evolutionary) Game Theory
- Game Theoretic Intuitions: NFG and Replicator Dynamics
- Opportunities & Challenges

Motivation

- Re-thinking fundamentals of whole area
  - Special issue Shoham 2007
  - AI Magazine article (Weiss & Tuyls)
  - The rise of Deep Learning and building AGI

- A unified formal framework
- Better understanding/theoretical underpinnings
- Application to complex systems

Based on a recent paper:

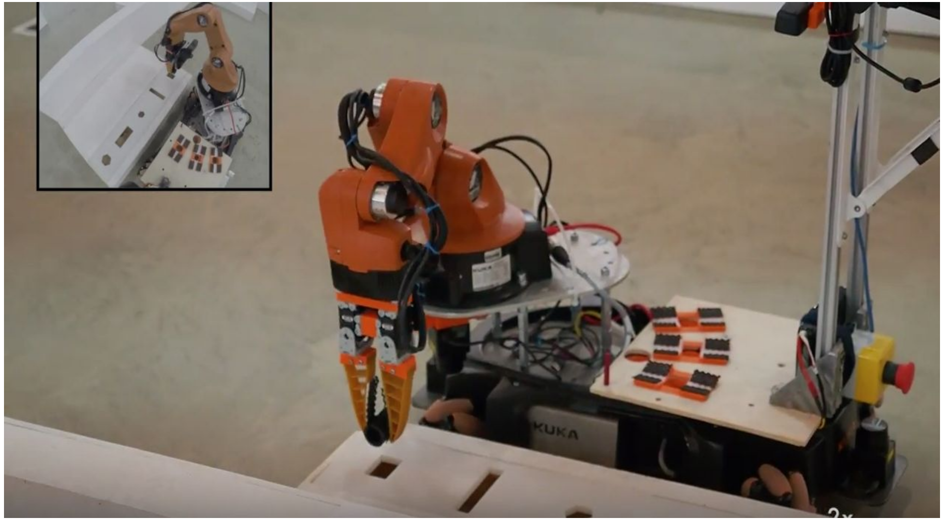
K. Tuyls and P. Stone: *Multiagent Learning Paradigms*. To Appear

Motivation

Surge in **Autonomous Systems** and **Artificial Intelligence Research**

Deep reinforcement learning

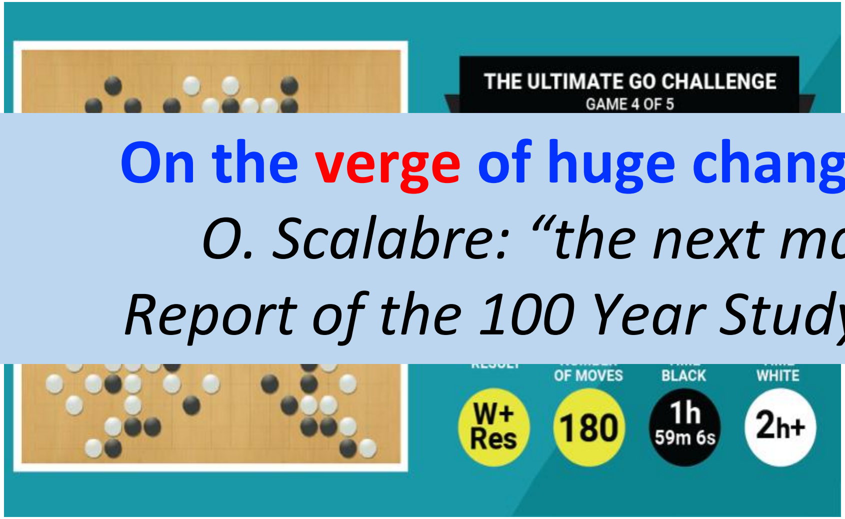
RoboCup@work (smARTLab)



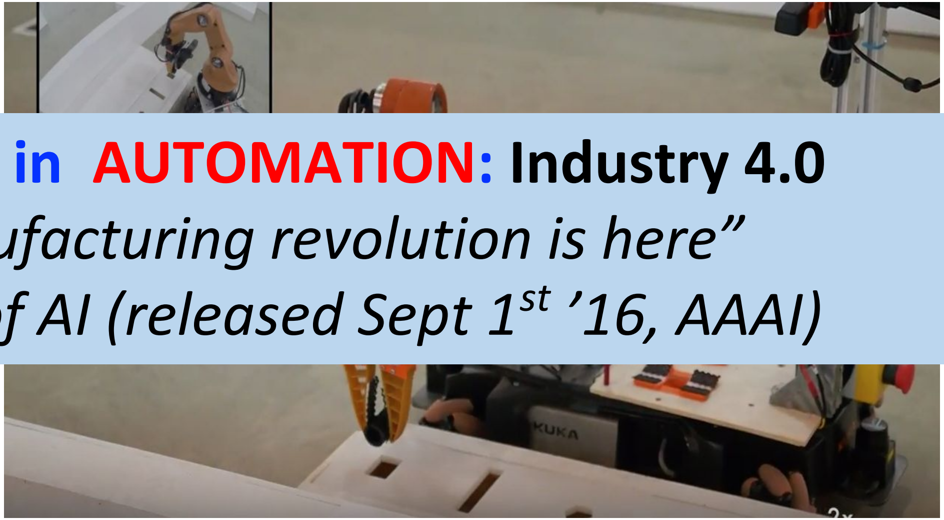
Motivation

Surge in **Autonomous Systems** and **Artificial Intelligence Research**

Deep reinforcement learning



RoboCup@work (smARTLab)



On the verge of huge changes in AUTOMATION: Industry 4.0

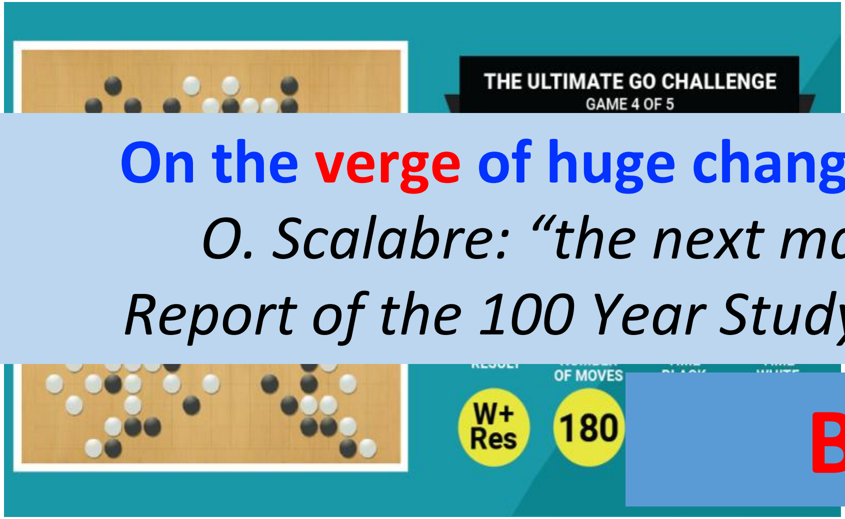
O. Scalabre: “the next manufacturing revolution is here”

Report of the 100 Year Study of AI (released Sept 1st ’16, AAAI)

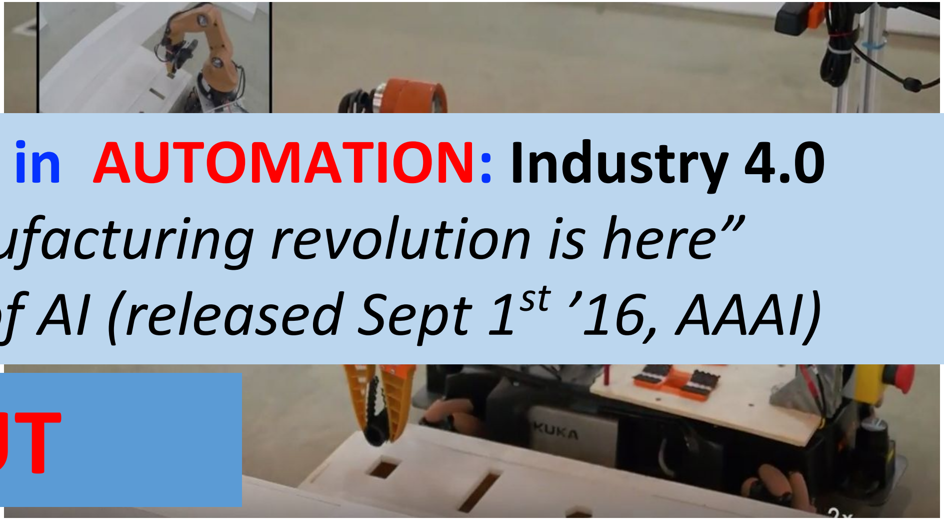
Motivation

Surge in **Autonomous Systems** and **Artificial Intelligence Research**

Deep reinforcement learning



RoboCup@work (smARTLab)



On the verge of huge changes in AUTOMATION: Industry 4.0

O. Scalabre: "the next manufacturing revolution is here"

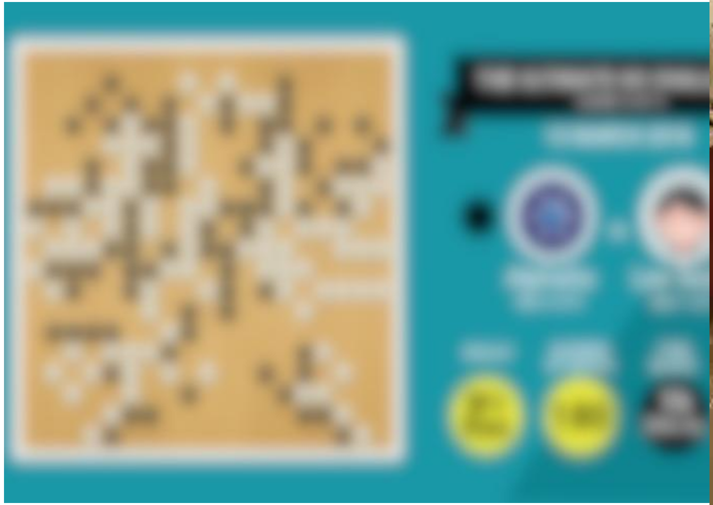
Report of the 100 Year Study of AI (released Sept 1st '16, AAAI)

BUT

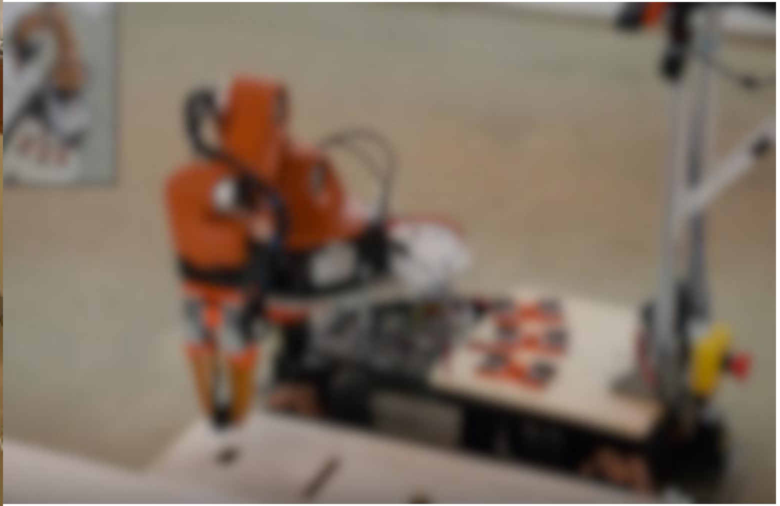
Motivation

Surge in **Autonomous Systems** and **Artificial Intelligence Research**

Deep reinforcement learning



RoboCup@work (smARTLab)

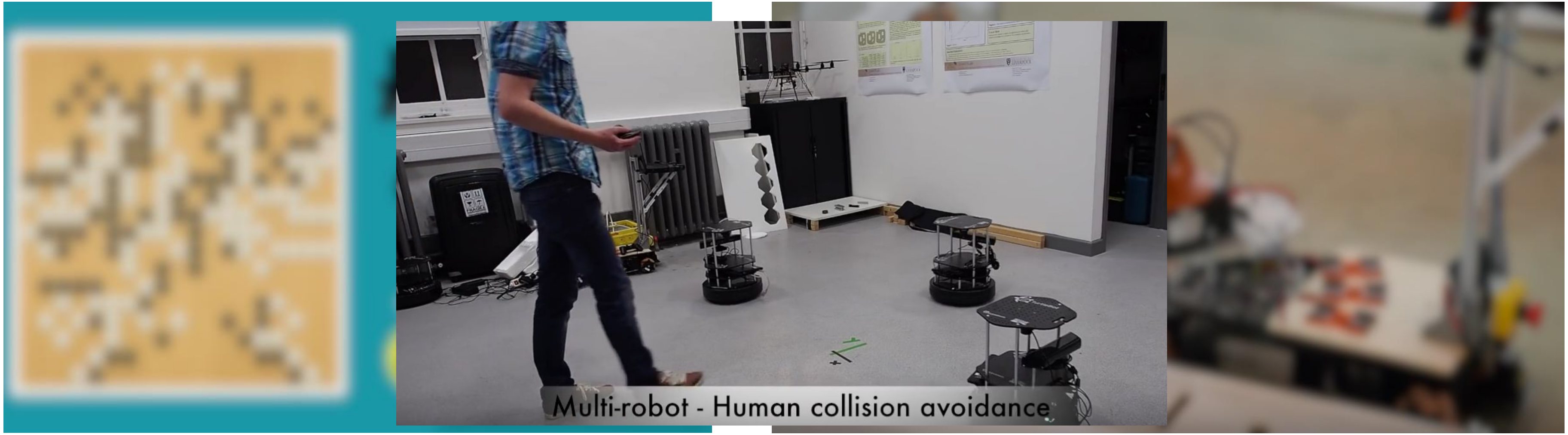


Motivation

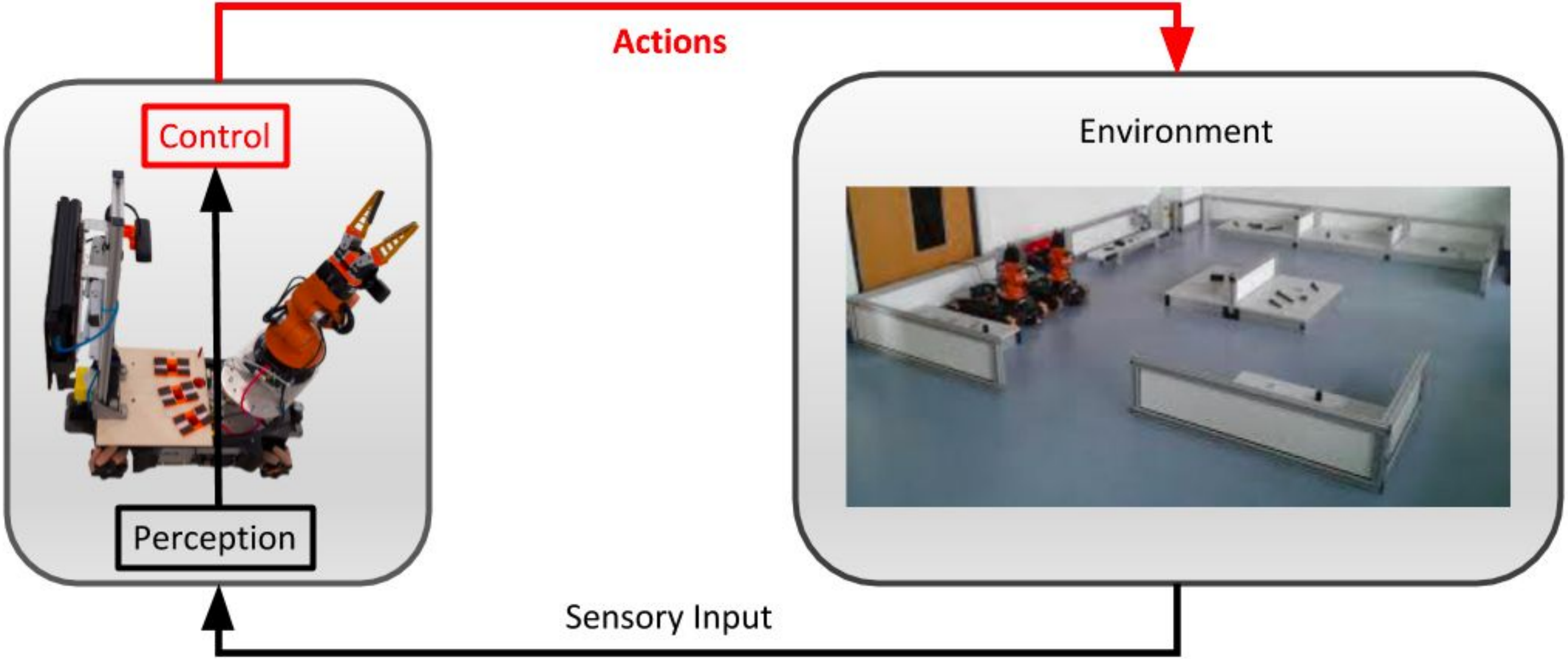
Surge in **Autonomous Systems** and **Artificial Intelligence Research**

Deep reinforcement learning

RoboCup@work (smARTLab)

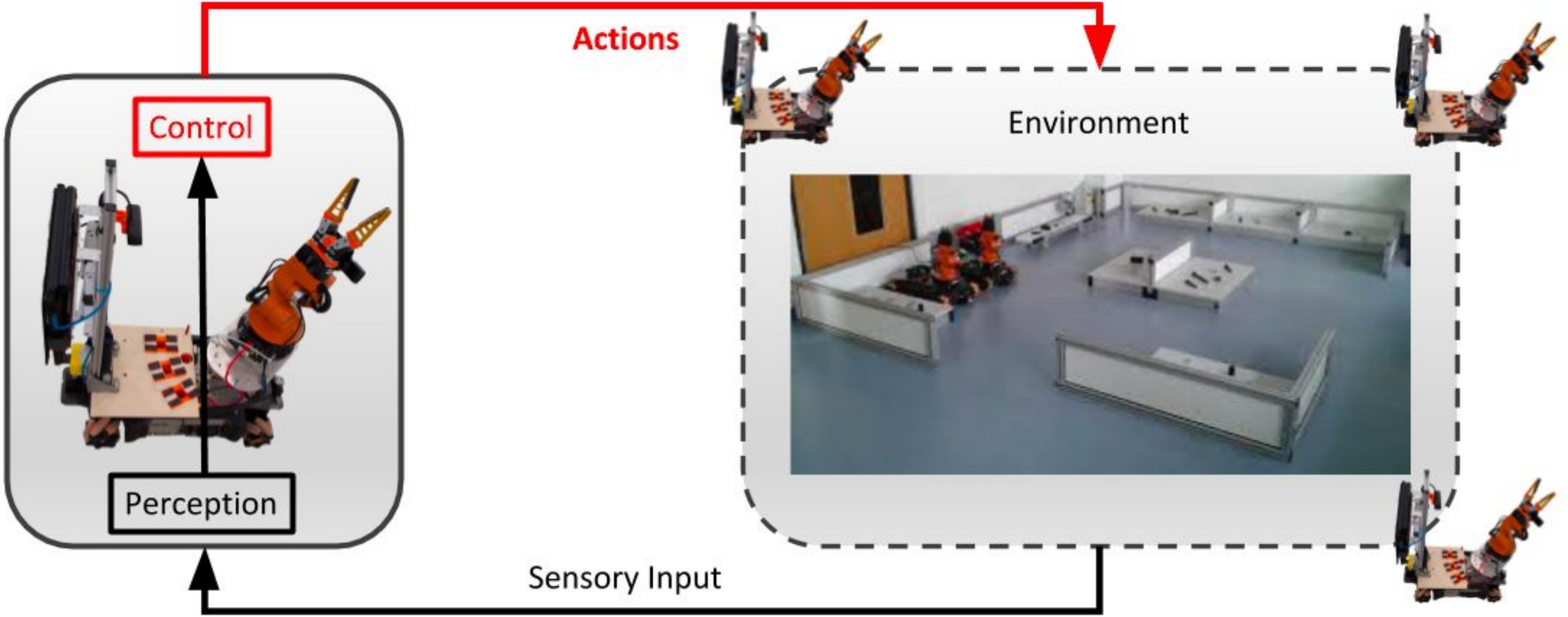


Motivation



DeepMind

Motivation



Actions

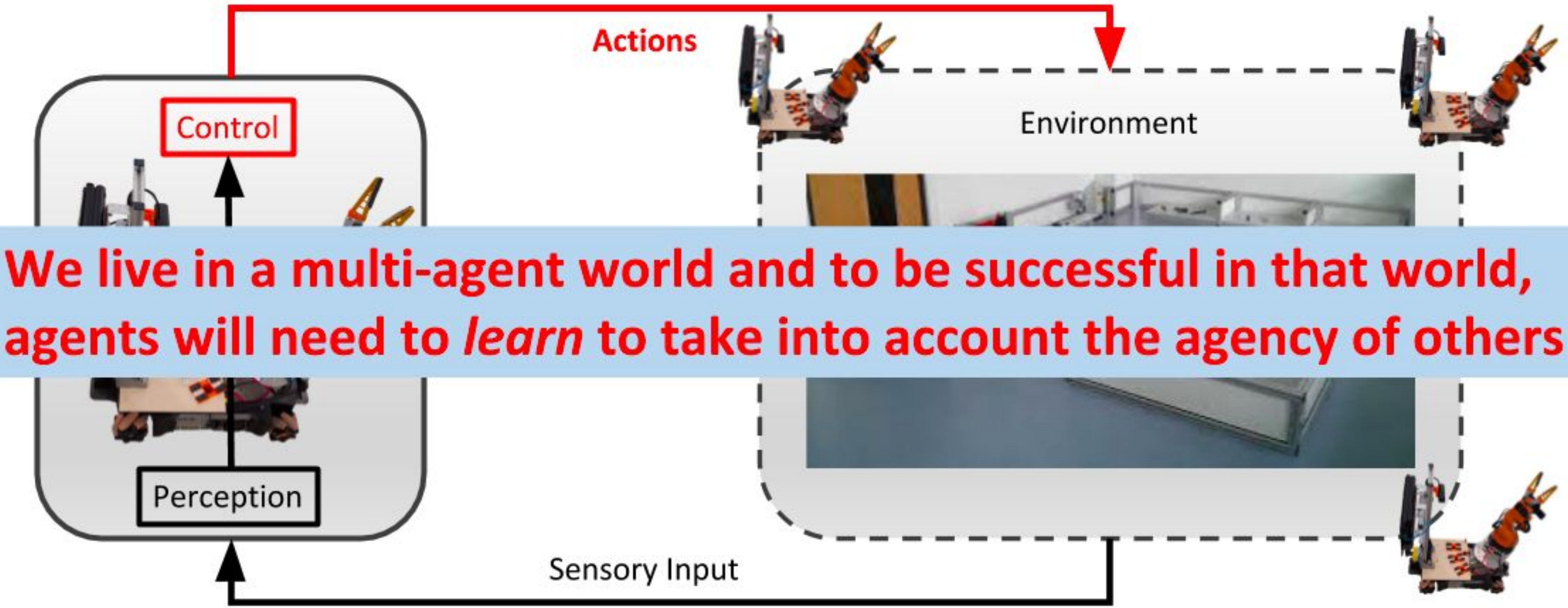
Control

Perception

Environment

Sensory Input

Motivation



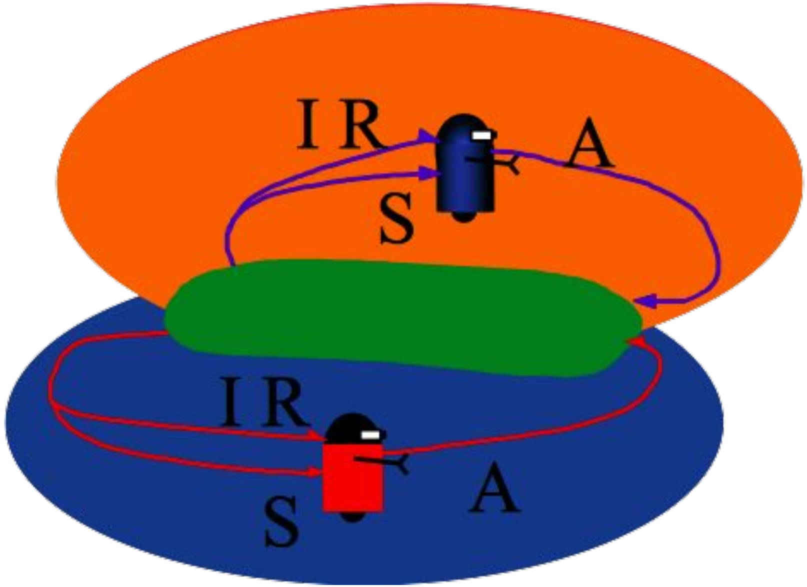
We live in a multi-agent world and to be successful in that world, agents will need to *learn* to take into account the agency of others

Example (RoboCup)

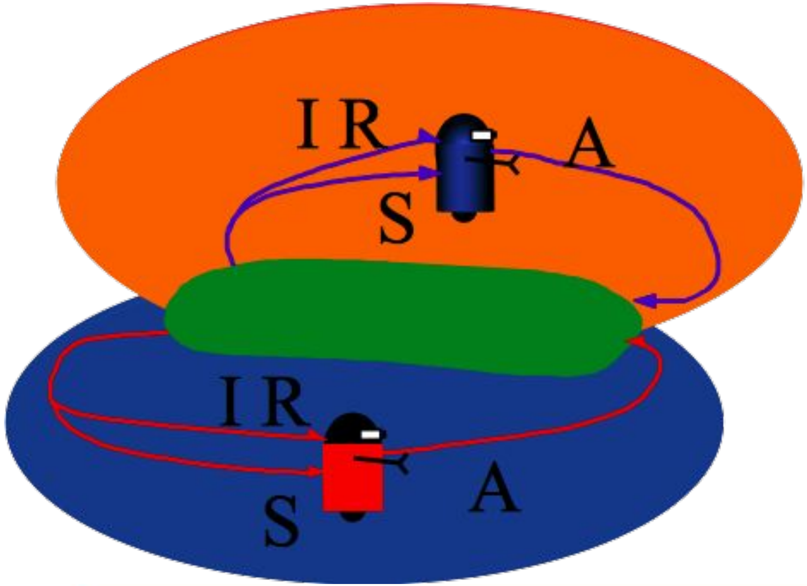
Example warehouse commissioning

The robots decide autonomously which actions to take.
They receive the global state from the warehouse management software.
The global state consists of the currently active orders and approximate positions of the other robots.

What is Multi-Agent Learning?

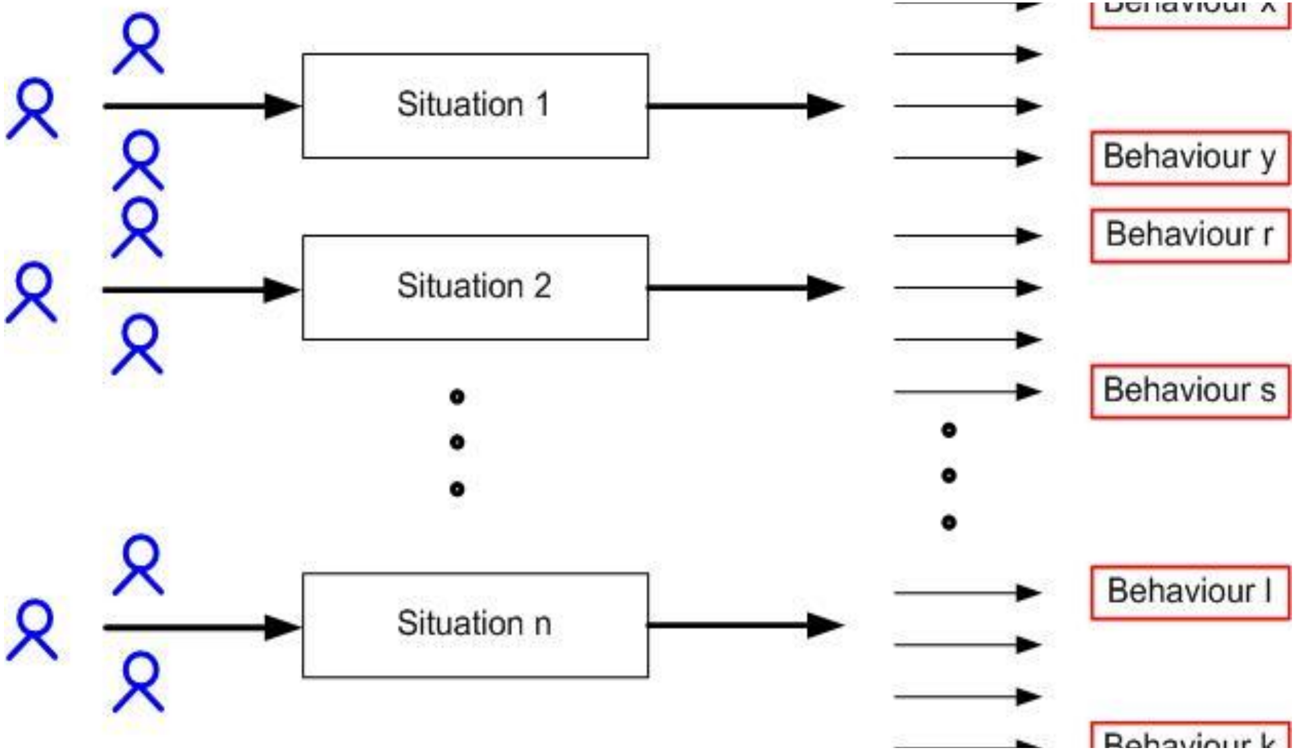


What is Multi-Agent Learning?



Multi-Agent Learning lacks a Foundation, or Theory, of its own

What is Multi-Agent Learning?



Situation 1

Situation 2

Situation n

Behaviour x

Behaviour y

Behaviour r

Behaviour s

Behaviour l

Behaviour k

What is Multi-Agent Learning?

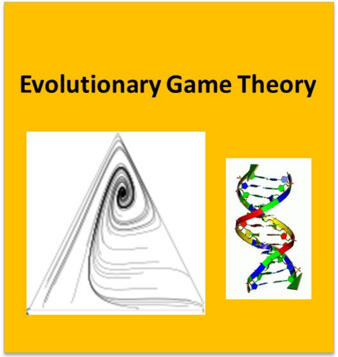
The study of multi-agent systems in which one or more of the autonomous entities improves automatically through experience

K. Tuyls and P. Stone: *Multiagent Learning Paradigms*.

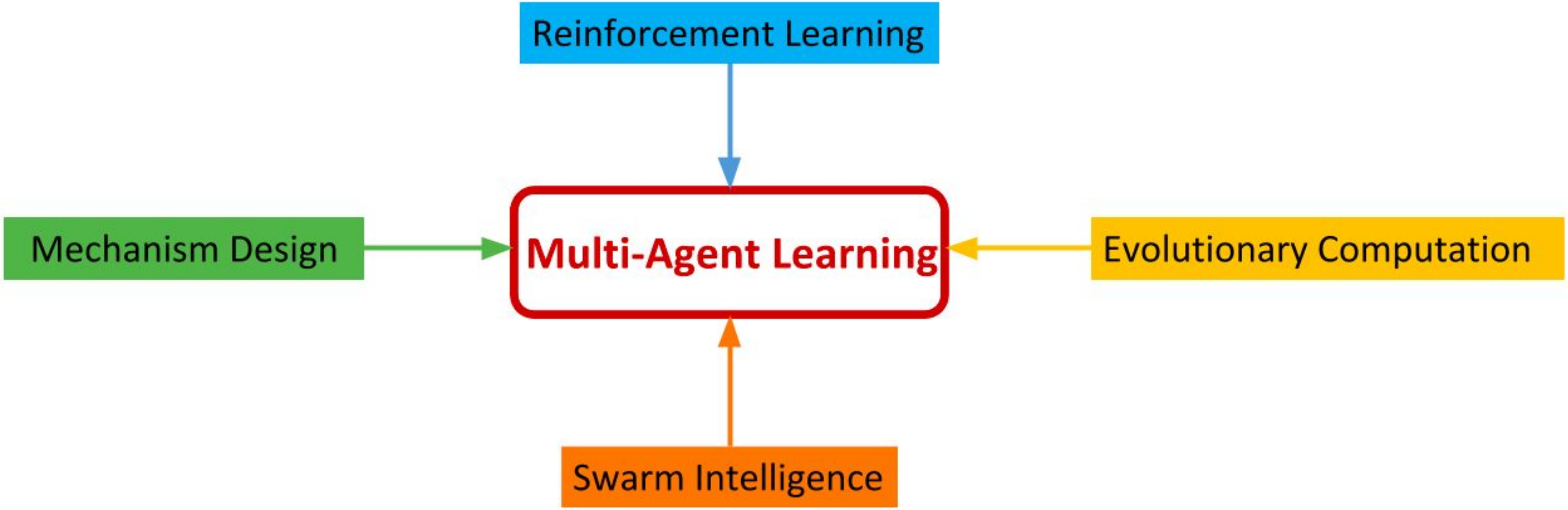
What is Multi-Agent Learning?

- RL towards individual utility
- RL towards social welfare
- Co-evolutionary learning
- Swarm Intelligence
- Adaptive mechanism design

- Tools
    - EGT
    - (Opponent Modelling)

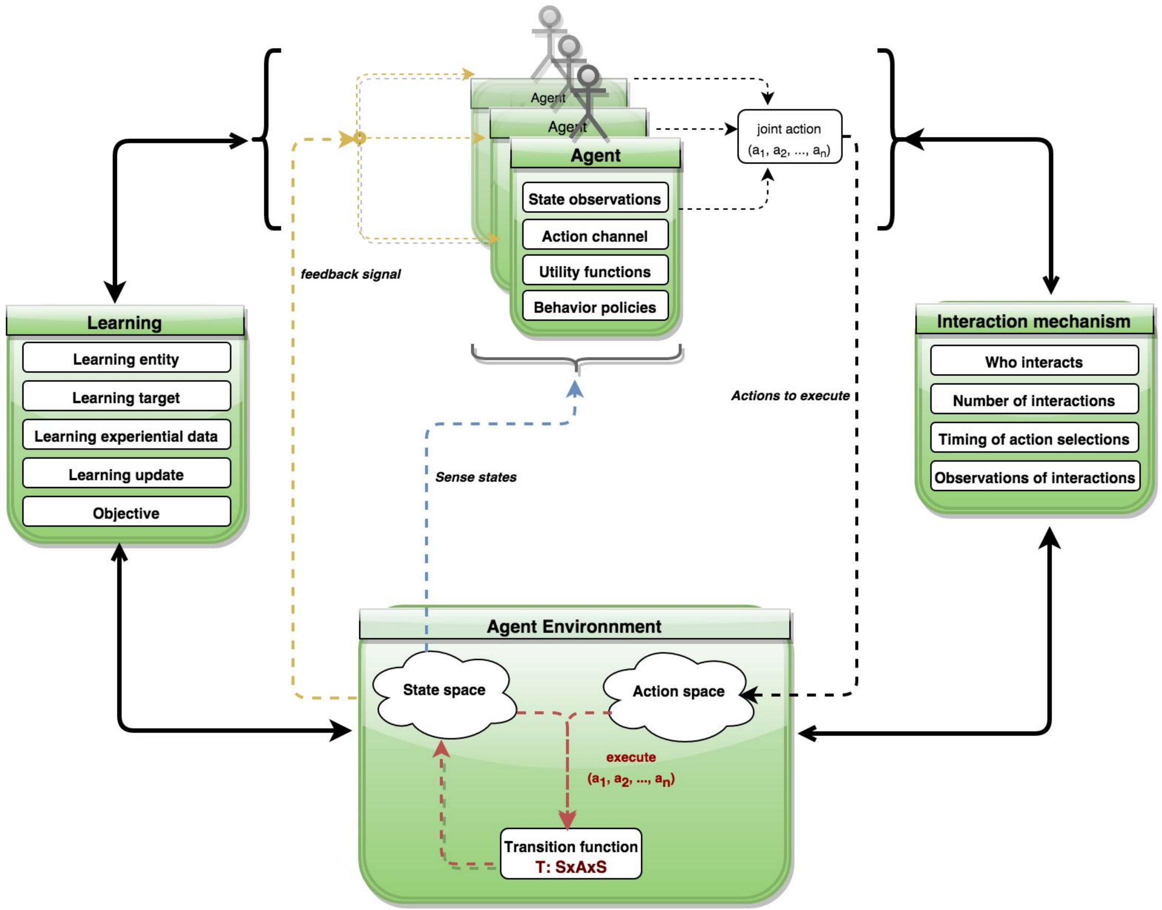


What is Multi-Agent Learning?



"Perhaps a thing is simple if you can describe it fully in several different ways, without immediately knowing that you are describing the same thing" R. Feynman

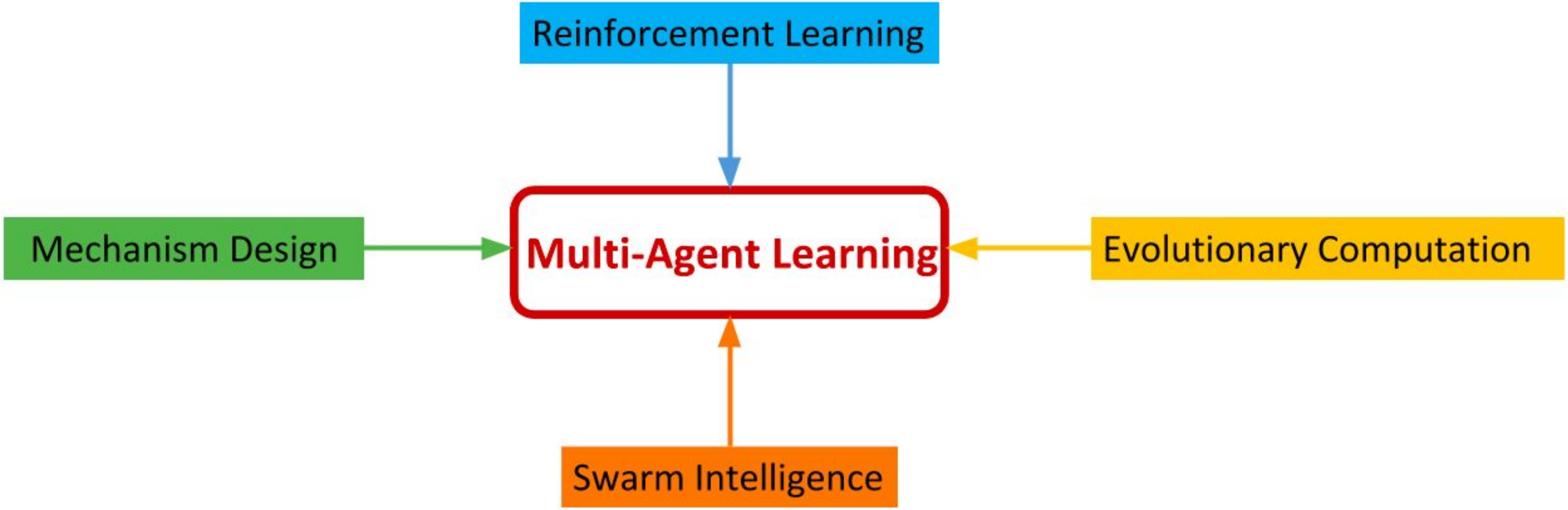
General Setup



Several Realizations

1. Online RL towards individual utility
2. Online RL towards social welfare
3. Co-Evolutionary approaches
4. Swarm Intelligence
5. Adaptive Mechanism Design

What is Multi-Agent Learning?



Reinforcement Learning

Mechanism Design

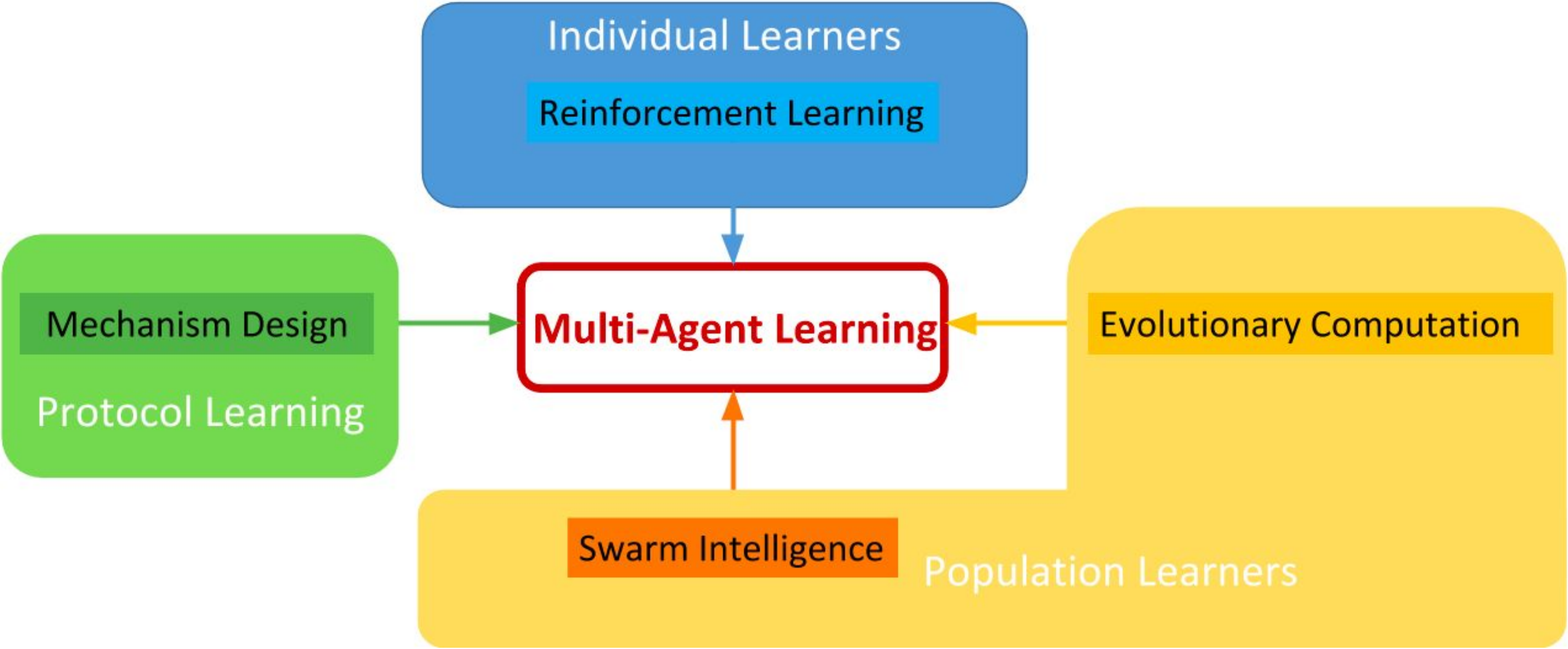
Multi-Agent Learning

Evolutionary Computation

Swarm Intelligence

“Perhaps a thing is simple if you can describe it fully in several different ways, without immediately knowing that you are describing the same thing” R. Feynman

EGT: unified theory (Role of EGT)



Individual Learners

Reinforcement Learning

Mechanism Design

Protocol Learning

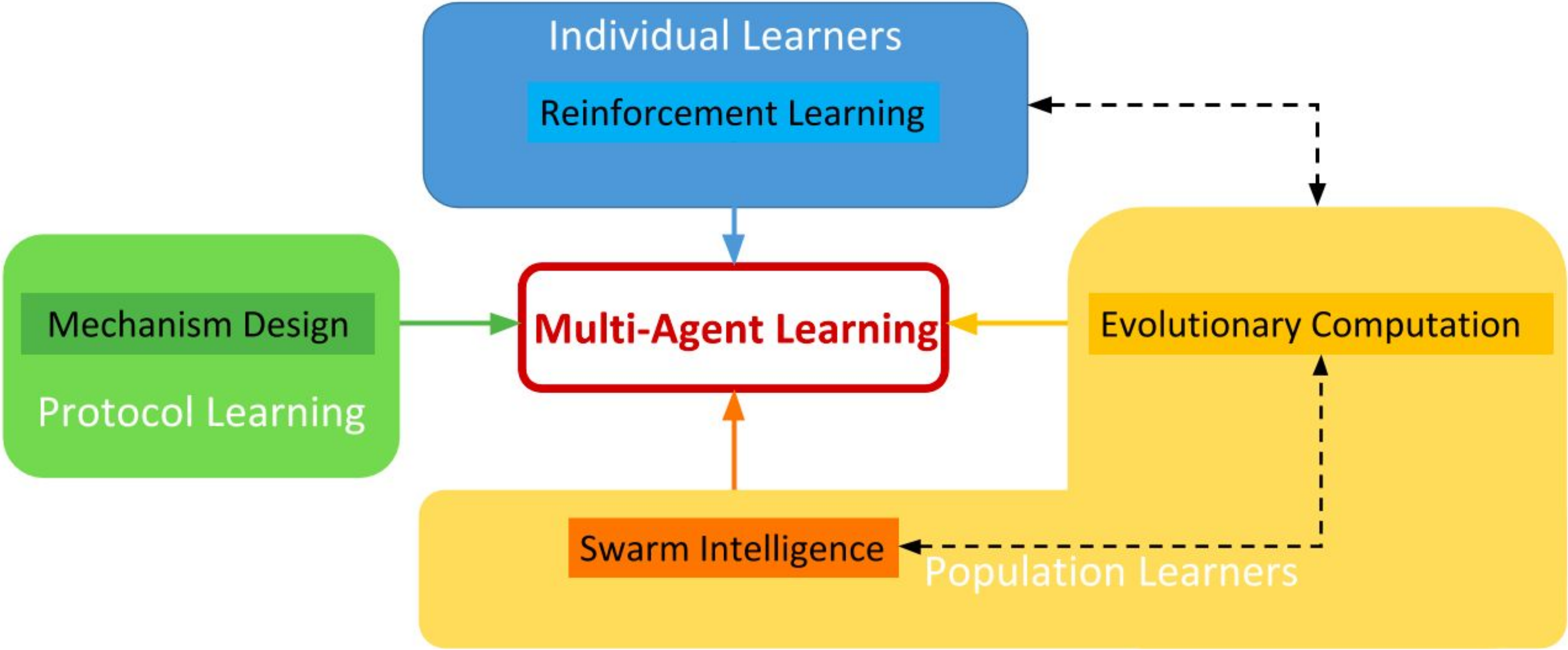
Multi-Agent Learning

Evolutionary Computation

Swarm Intelligence

Population Learners

EGT: unified theory (Role of EGT)



Individual Learners

Reinforcement Learning

Mechanism Design

Protocol Learning

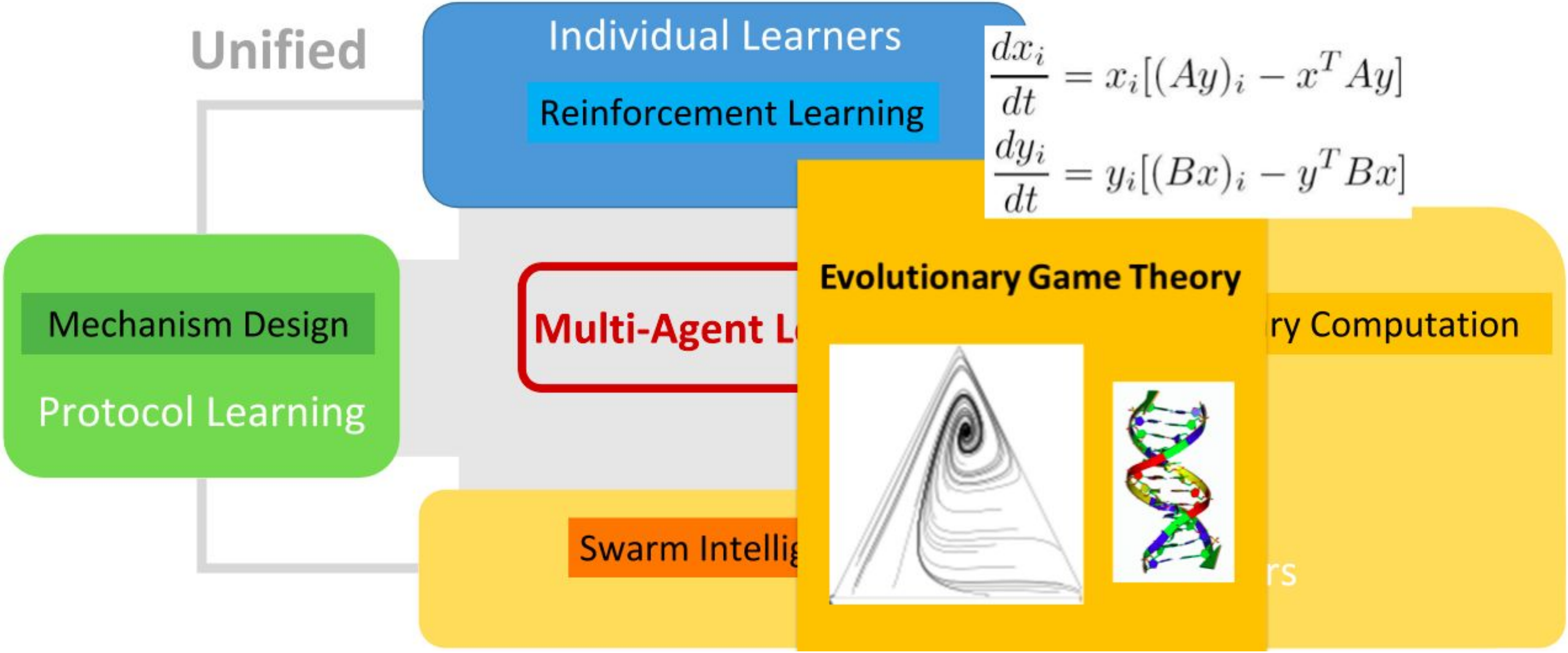
Multi-Agent Learning

Evolutionary Computation

Swarm Intelligence

Population Learners

EGT: unified theory (Role of EGT)



Unified

Individual Learners

Reinforcement Learning

$$\frac{dx_i}{dt} = x_i[(Ay)_i - x^T Ay]$$

$$\frac{dy_i}{dt} = y_i[(Bx)_i - y^T Bx]$$

Mechanism Design

Protocol Learning

Multi-Agent L

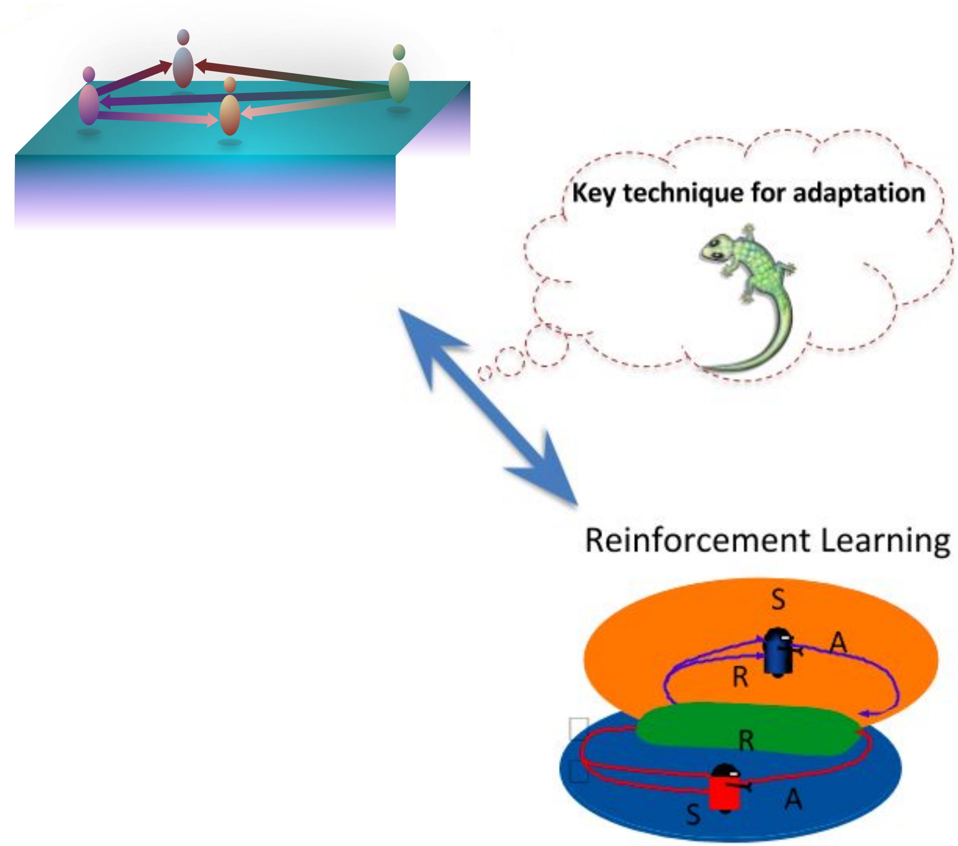
Evolutionary Game Theory

ry Computation

Swarm Intellig

rs

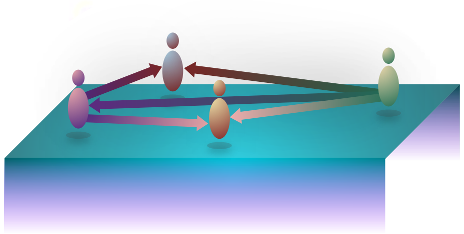
EGT: Towards a Unified Theory (Role of EGT)

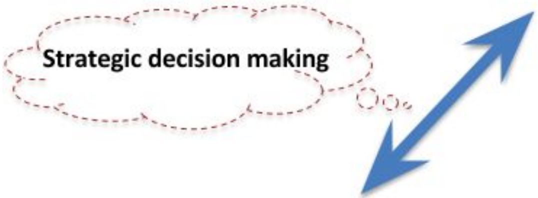


Key technique for adaptation

Reinforcement Learning

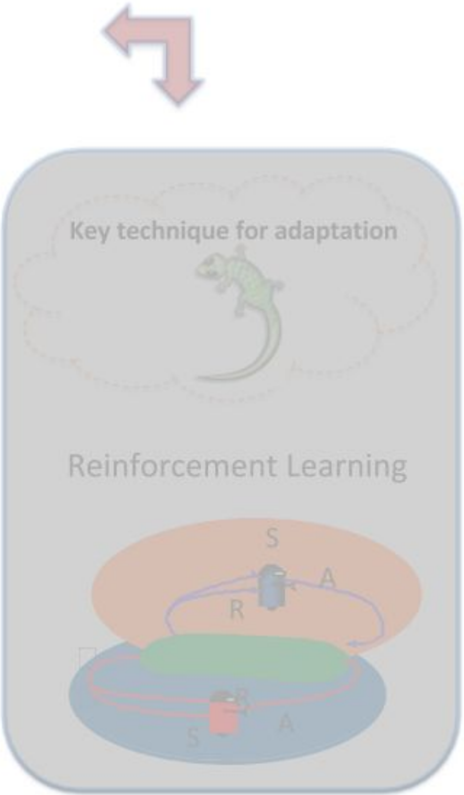
EGT: Towards a Unified Theory (Role of EGT)





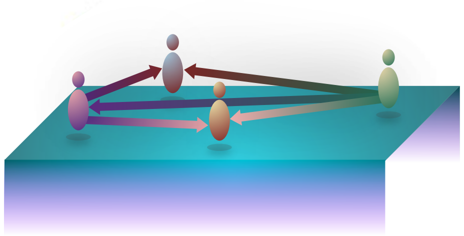
Classical Game Theory

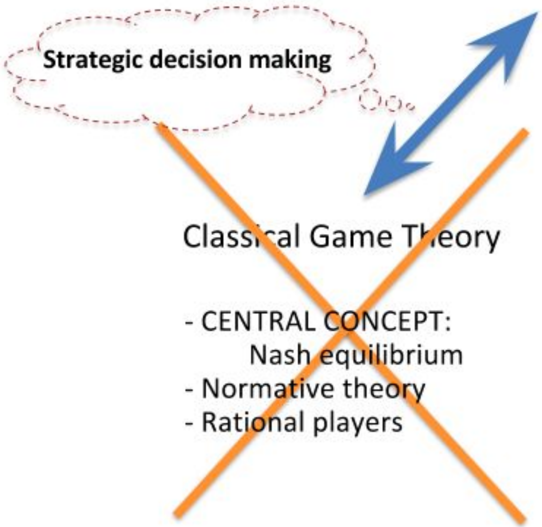
- CENTRAL CONCEPT:
  Nash equilibrium
- Normative theory
- Rational players


Key technique for adaptation

Reinforcement Learning

EGT: Towards a Unified Theory (Role of EGT)

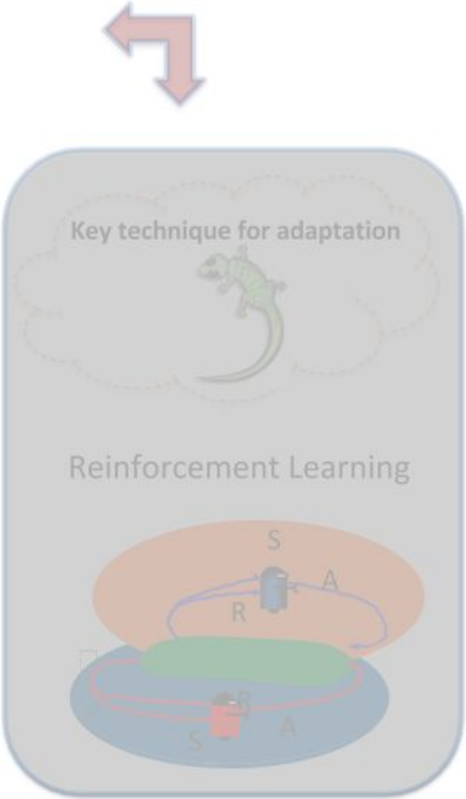




Strategic decision making

~~Classical Game Theory~~

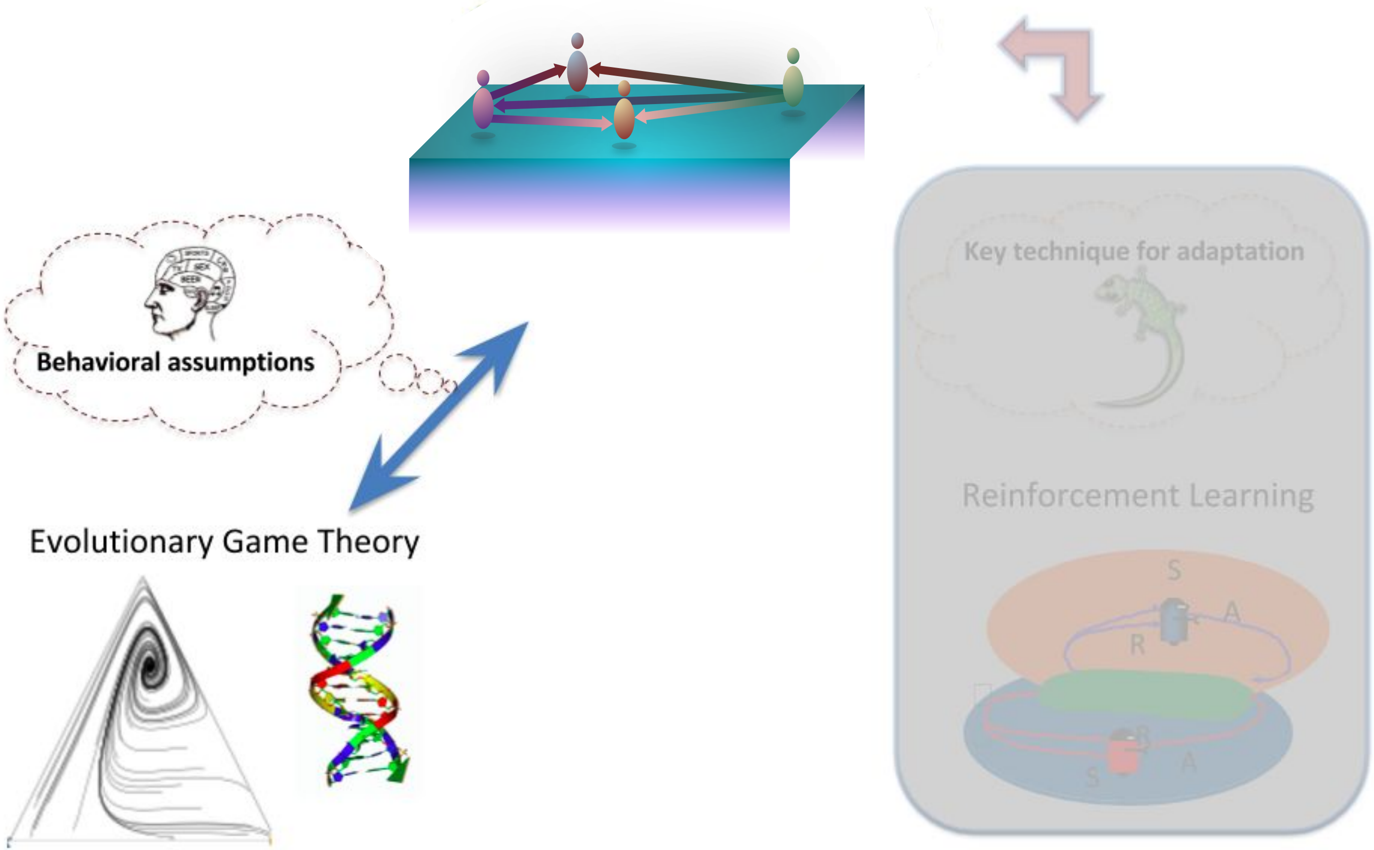
- ~~CENTRAL CONCEPT: Nash equilibrium~~
- ~~Normative theory~~
- ~~Rational players~~



Key technique for adaptation

Reinforcement Learning

EGT: Towards a Unified Theory (Role of EGT)



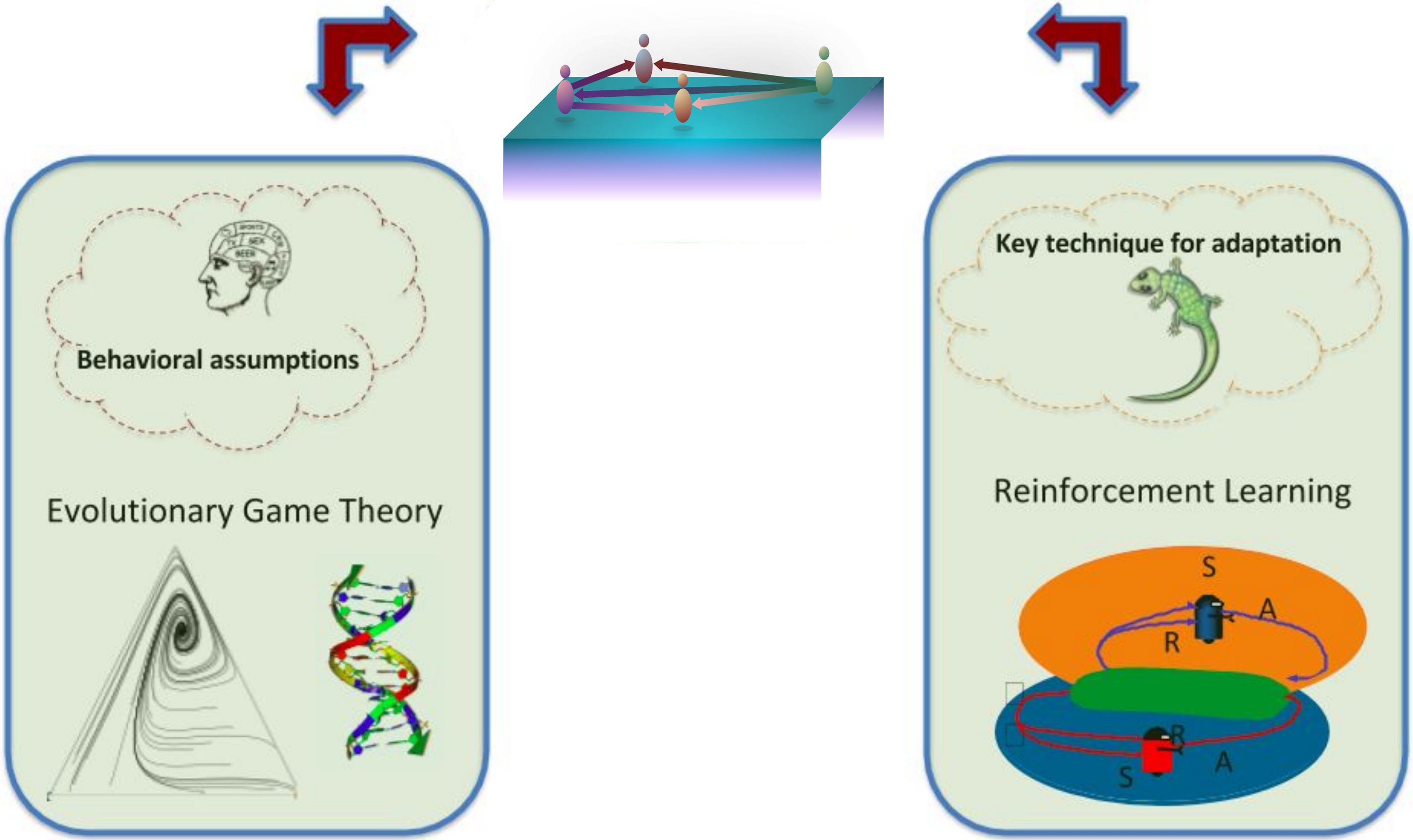
Behavioral assumptions

Evolutionary Game Theory

Key technique for adaptation

Reinforcement Learning

EGT: Towards a Unified Theory (Role of EGT)



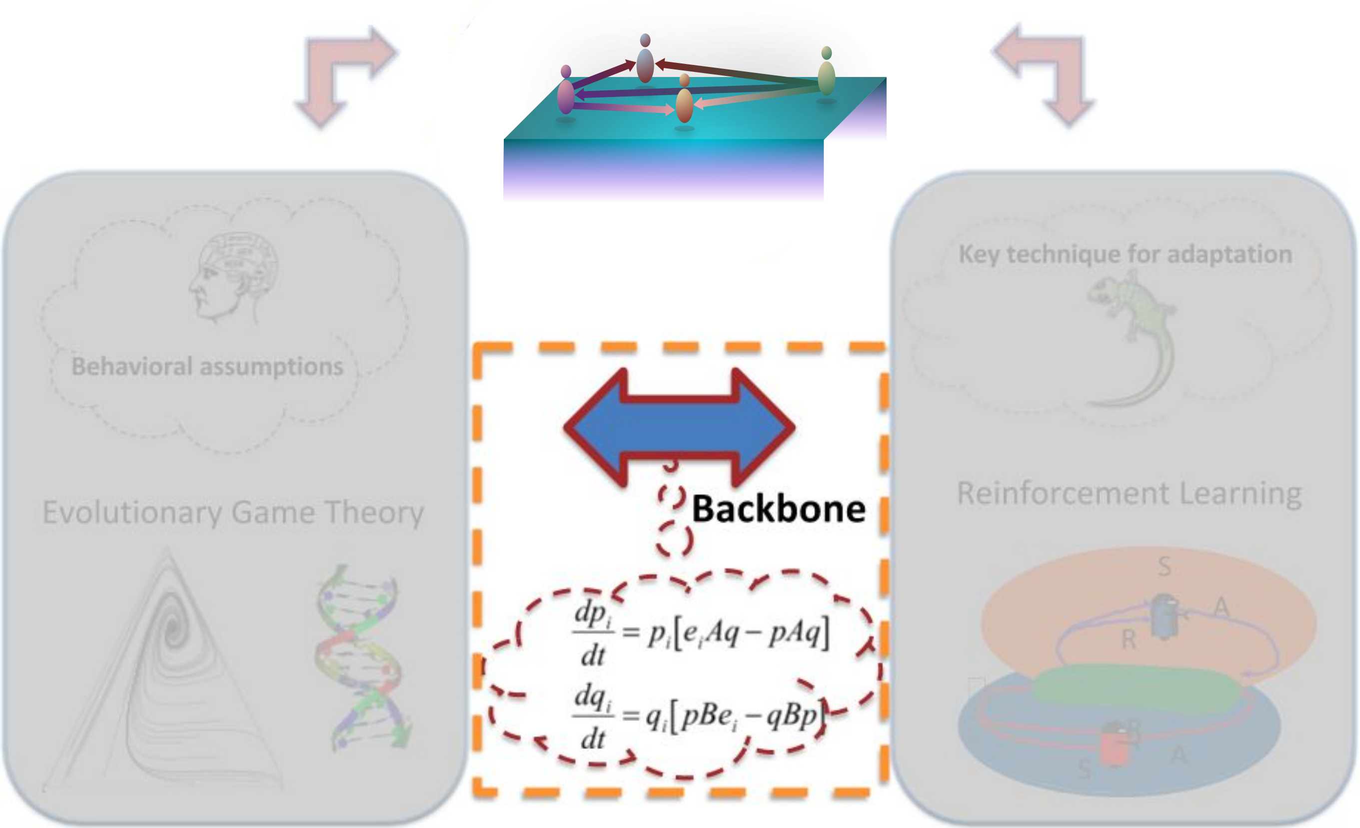
Behavioral assumptions

Evolutionary Game Theory

Key technique for adaptation

Reinforcement Learning

EGT: Towards a Unified Theory (Role of EGT)



Behavioral assumptions

Evolutionary Game Theory

Backbone

$$\frac{dp_i}{dt} = p_i[e_i Aq - pAq]$$

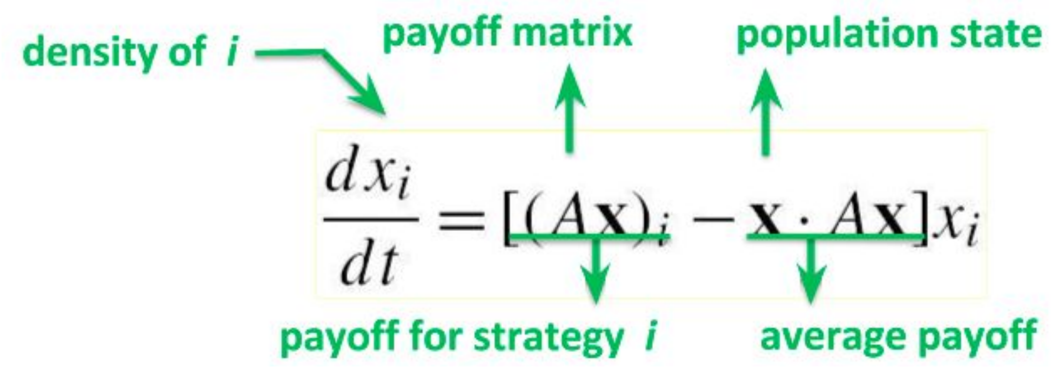
$$\frac{dq_i}{dt} = q_i[pBe_i - qBp]$$

Key technique for adaptation

Reinforcement Learning

Game Theoretic Intuitions

- Evolutionary Game Theory (EGT), 1
    - Application of game theory to evolving populations of lifeforms in biology (1973, Smith & Price)
    - EGT differs from classical GT by focusing more on the dynamics of strategy change (quality, frequency)
    - Common approach: **replicator equations**, describing growth rate of the proportion of organisms using a certain strategy



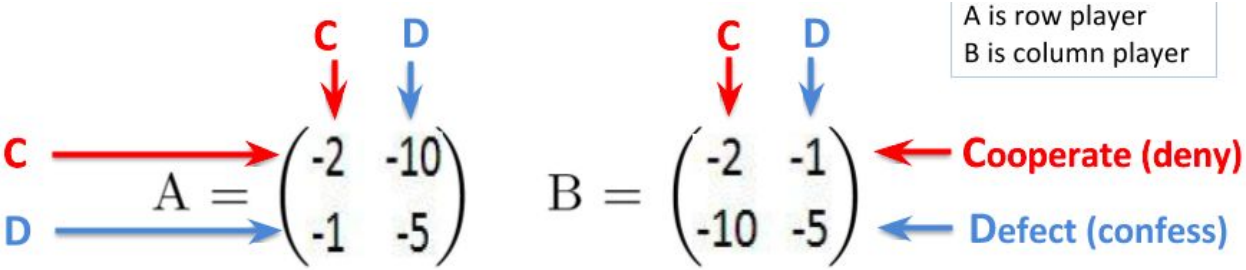
$$\frac{dx_i}{dt} = [(A\mathbf{x})_i - \mathbf{x}\cdot A\mathbf{x}]x_i$$

Game Theoretic Intuitions

- Evolutionary Game Theory (EGT), 2
  - **Extension** to two-player game situations, coupled replicator equations:

$$\frac{dx_i}{dt} = x_i[(Ay)_i - x^T Ay]$$
$$\frac{dy_i}{dt} = y_i[(Bx)_i - y^T Bx]$$

  - **Example**: Prisoner's dilemma



$$\begin{array}{cc} & \begin{array}{cc} C & D \end{array} \\ \begin{array}{c} C \\ D \end{array} & A = \begin{pmatrix} -2 & -10 \\ -1 & -5 \end{pmatrix} \end{array} \qquad \begin{array}{cc} \begin{array}{cc} C & D \end{array} & \\ B = \begin{pmatrix} -2 & -1 \\ -10 & -5 \end{pmatrix} & \begin{array}{c} \text{Cooperate (deny)} \\ \text{Defect (confess)} \end{array} \end{array}$$

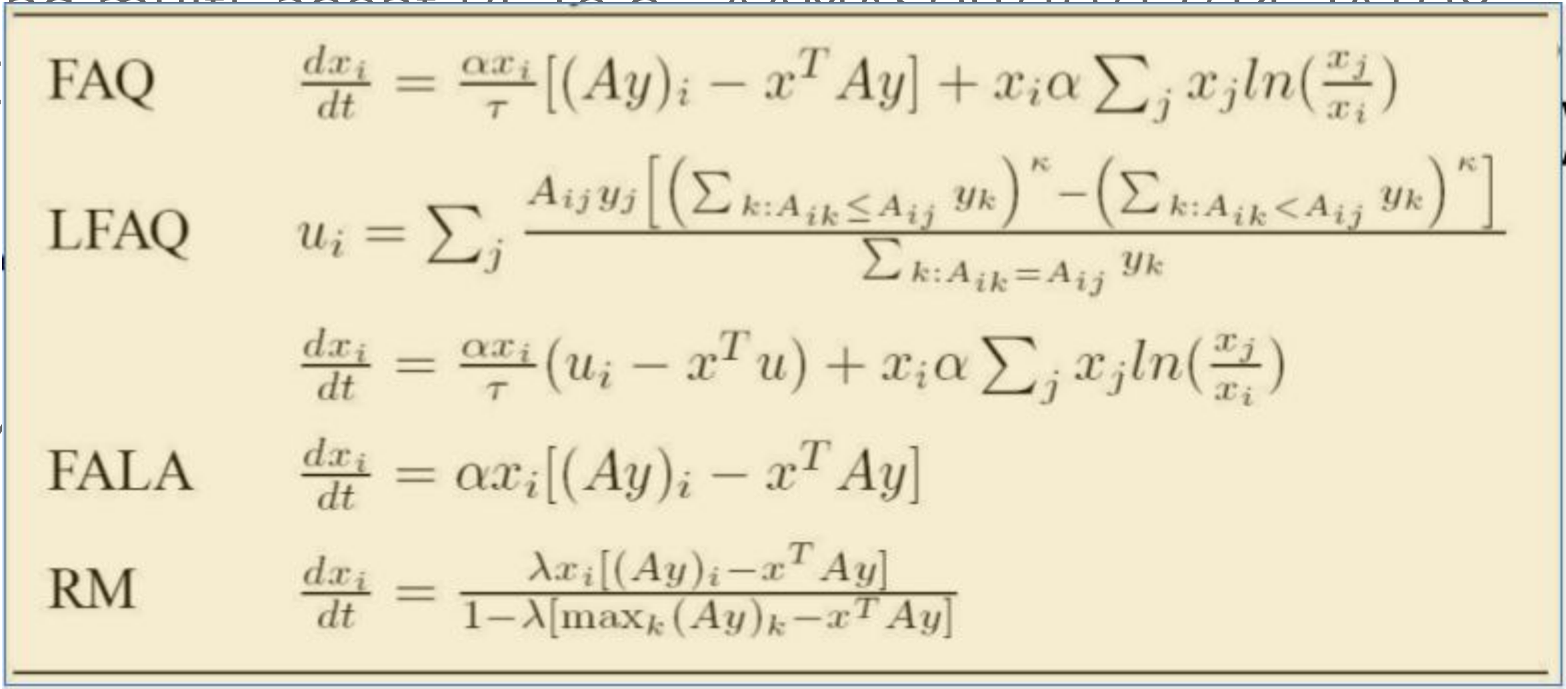
A is row player
B is column player

Game Theoretic Intuitions

- There are strong formal links between EGT and multi-agent RL [e.g., AAMAS09/10/12/14, IAT08, ECML, AAAI’14, JAIR’15 etc.]
  - Learning dynamics corresponds to replicator dynamics
  - The concept of evolutionary stable strategies (ESS) can be transferred to multi-agent RL (⇨ Nash equilibria)
- Multi-agent RL methods and evolutionary models
- Recently connection between PG and RD (Neural Replicator Dynamics)

Game Theoretic Intuitions

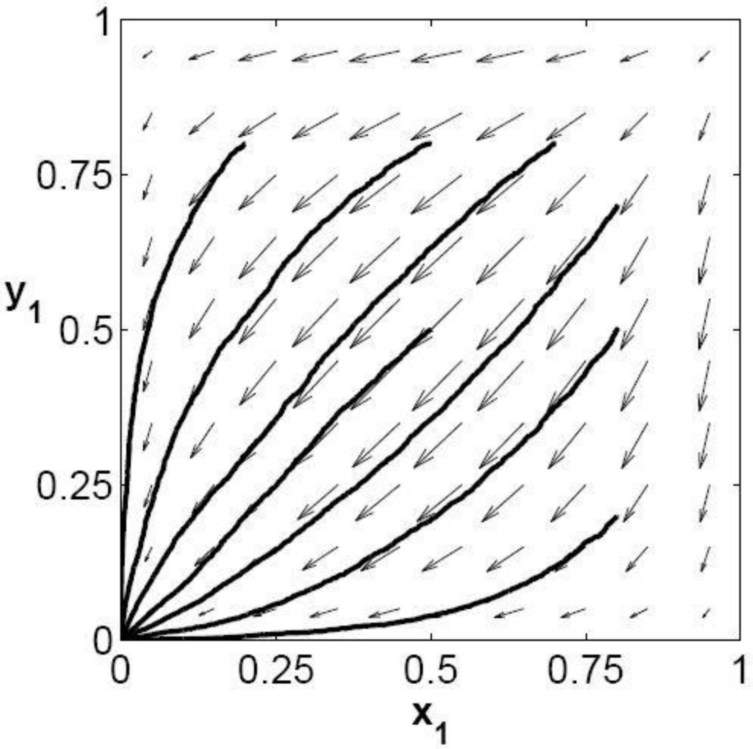
- We showed that there are strong formal links between EGT and multi-agent RL [e.g., AAMAS09/10/12/14,



FAQ	$\frac{dx_i}{dt} = \frac{\alpha x_i}{\tau}[(Ay)_i - x^T Ay] + x_i\alpha\sum_j x_j ln(\frac{x_j}{x_i})$
LFAQ	$u_i = \sum_j \frac{A_{ij}y_j\left[\left(\sum_{k:A_{ik}\leq A_{ij}} y_k\right)^{\kappa} - \left(\sum_{k:A_{ik}<A_{ij}} y_k\right)^{\kappa}\right]}{\sum_{k:A_{ik}=A_{ij}} y_k}$
	$\frac{dx_i}{dt} = \frac{\alpha x_i}{\tau}(u_i - x^T u) + x_i\alpha\sum_j x_j ln(\frac{x_j}{x_i})$
FALA	$\frac{dx_i}{dt} = \alpha x_i[(Ay)_i - x^T Ay]$
RM	$\frac{dx_i}{dt} = \frac{\lambda x_i[(Ay)_i - x^T Ay]}{1-\lambda[\max_k(Ay)_k - x^T Ay]}$

Game Theoretic Intuitions

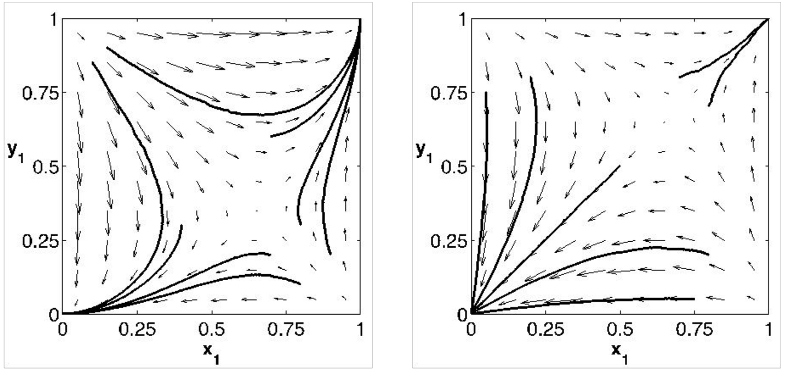
FAQ and Prisoner`s Dilemma



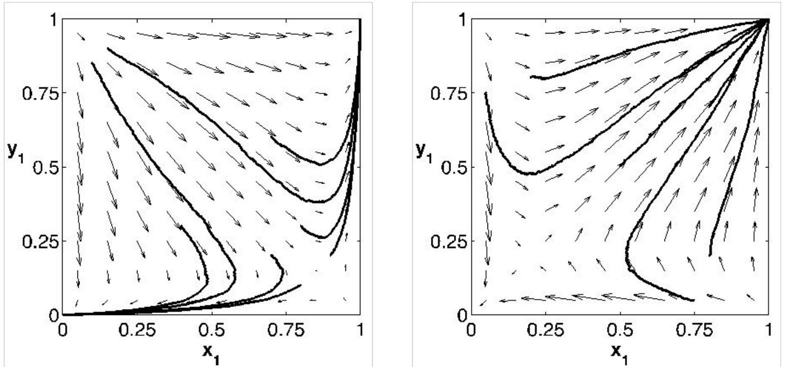
Game Theoretic Intuitions

	Football	Ballet
Football	2 1	0 0
Ballet	0 0	1 2

FAQ self play

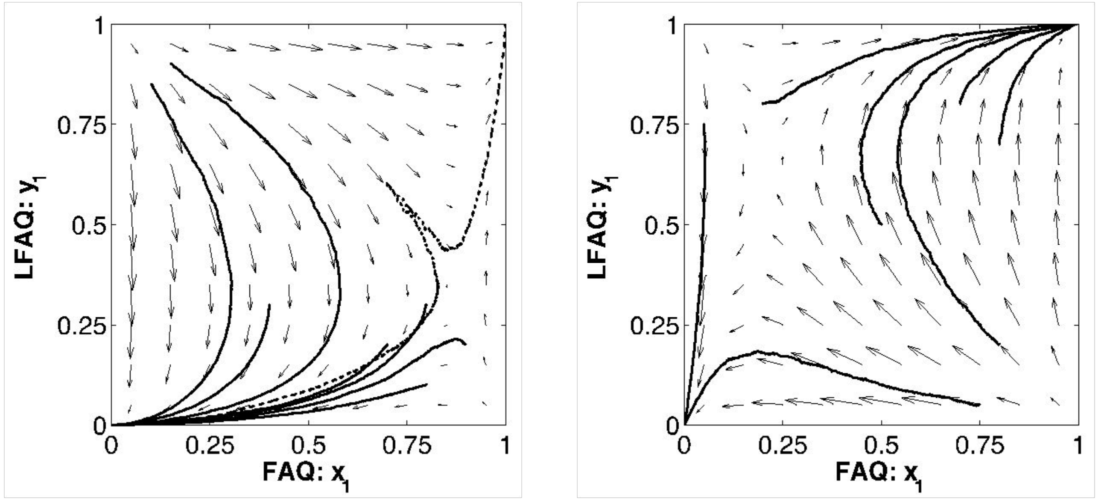


LFAQ self play



Game Theoretic Intuitions

FAQ vs. LFAQ mixed play



Battle of the Sexes

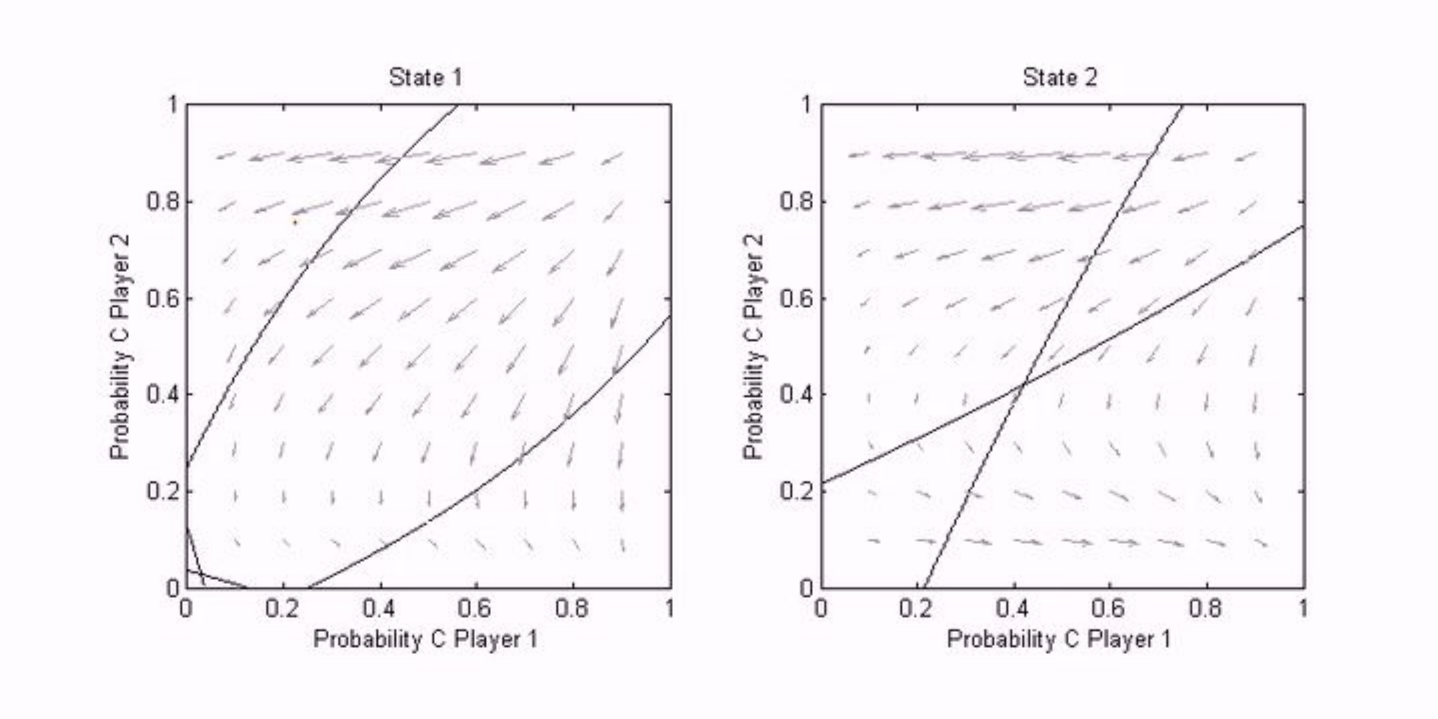
Stag Hunt

Game Theoretic Intuitions

Switching dynamics

	2 State PD					
	State 1			State 2		
		C	D		C	D
Rewards	C	0.3, 0.3	0, 1	C	0.4, 0.4	0, 1
	D	1, 0	0.2, 0.2	D	1, 0	0.1, 0.1
Transitions	(C,C)→ (0.9,0.1)			(C,C)→ (0.1,0.9)		
	(C,D)→ (0.1,0.9)			(C,D)→ (0.9,0.1)		
	(D,C)→ (0.1,0.9)			(D,C)→ (0.9,0.1)		
	(D,D)→ (0.9,0.1)			(D,D)→ (0.1,0.9)		

Game Theoretic Intuitions



Other paradigms

- Swarm Intelligence: Haitham Bou-Ammar, Karl Tuyls, Michael Kaisers: Evolutionary Dynamics of Ant Colony Optimization. MATES 2012: 40-52

- Co-evolution: Liviu Panait, Karl Tuyls, Sean Luke: Theoretical Advantages of Lenient Learners: An Evolutionary Game Theoretic Perspective. Journal of Machine Learning Research 9: 423-457 (2008)

(Some) References

- M. L. Littman. Markov Games as a Framework for Multi-Agent Reinforcement Learning. ICML 1994: 157-163
- C. Claus, C. Boutilier. The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems. AAAI/IAAI 1998: 746-752
- G. Weiss. MultiAgent Systems (2nd edition), 2013. ISBN 978-0-262-01889-0
- Yoav Shoham, Rob Powers, Trond Grenager. If multi-agent learning is the answer, what is the question? Artif. Intell. 171(7): 365-377 (2007)
- D, Bloembergen, K. Tuyls, D. Hennes, and M. Kaisers. Evolutionary Dynamics of Multi-Agent Learning: A Survey. Journal of Artificial Intelligence Research (JAIR), Volume 53, pages 659-697, 2015
- K. Tuyls and P. Stone. Multiagent learning paradigms. To appear.
- P. Stone. Multiagent learning is not the answer. It is the question. Artif. Intell. 171(7): 402-405 (2007)
- P. Stone, M. Veloso. Multiagent Systems: A Survey from a Machine Learning Perspective. Auton. Robots 8(3): 345-383 (2000)

2. From Normal Form to Markov Games

DeepMind

Game Theory 101

- Game theory's role in multi-agent learning:
  - Model of agent interactions
  - Analytic toolkit for evaluating agents
  - Consistent driver of innovations in learning algorithms

- **Objective:**

  Provide foundational & intuitive understanding of key game theory concepts

From Normal Form to Markov Games

Normal Form Games	Definitions: ● Model ● Solution concepts	Algorithms Based on Best Response
Markov Games	Definitions: ● Model ● Optimal policy	Learning in Markov Games (Part II)

From Normal Form to Markov Games

Normal Form Games

Definitions:

- Model
- Solution concepts

Algorithms Based on Best Response

Markov Games

Definitions:
- Model
- Optimal policy

Learning in Markov Games (Part II)

Normal Form Games: Formal Description

Let’s start with a two-player Normal Form Game (NFG):

Player 2

Player 1		a^2_1	…	a^2_n
	a^1_1			
	⋮		$r^1(a^1_i,a^2_j)$	
	a^1_n			

Player 1 payoff table R¹

Player 2

Player 1		a^2_1	…	a^2_n
	a^1_1			
	⋮		$r^2(a^1_i,a^2_j)$	
	a^1_n			

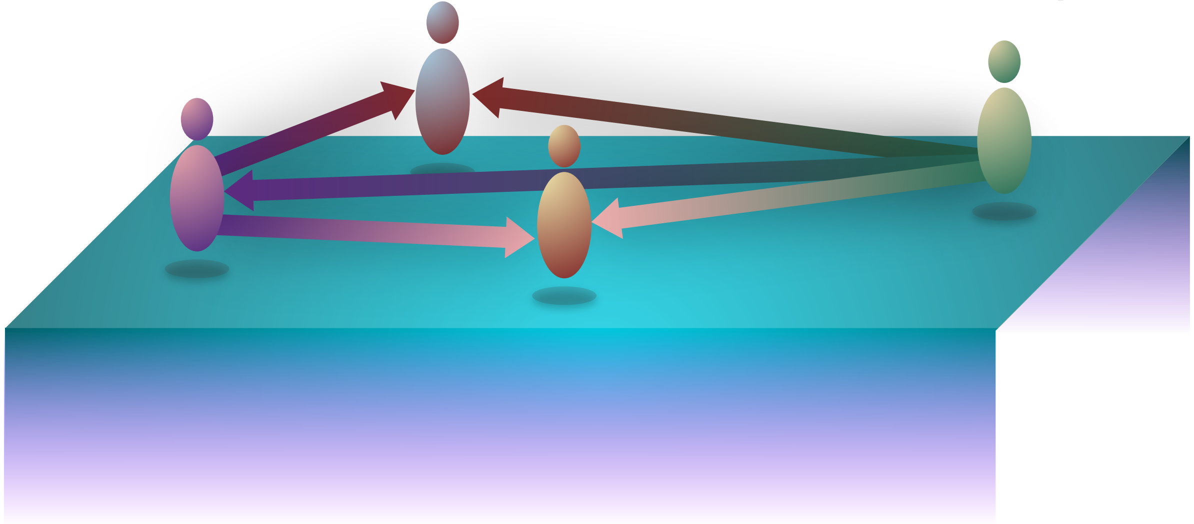
Player 2 payoff table R²

If **pure** strategies are selected according to **mixed strategies** π^1 and π^2 (*i.e.*, $a^1 \sim \pi^1$ and $a^2 \sim \pi^2$):

Player 1 will receive $E_{\pi 1,\pi 2}\ [r^1(a^1,a^2)] = \pi^{1T}\ R^1\ \pi^2$

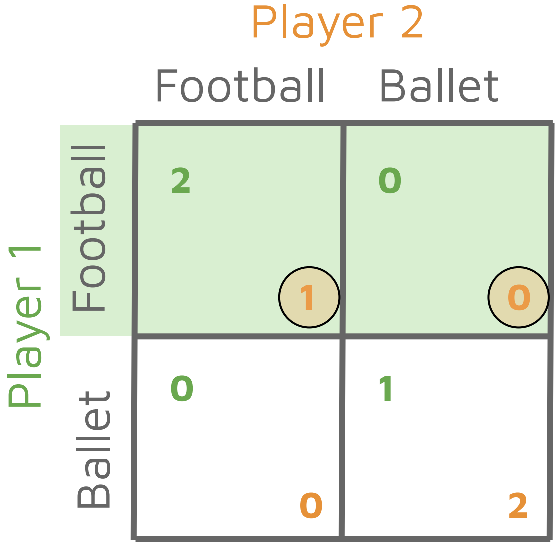
Player 2 will receive $E_{\pi 1,\pi 2}\ [r^2(a^1,a^2)] = \pi^{1T}\ R^2\ \pi^2$

Normal Form Games: Solution Concept



- Next step: analyze agent behaviors given this model of interactions
- A **solution concept** is a formal set of principles that can be:
  - Descriptive: forecasts how agents **will** behave
  - Prescriptive: suggests how agents **should** behave

Normal Form Games: Solution Concept



		Player 2	
		Football	Ballet
Player 1	Football	2 1	0 0
	Ballet	0 0	1 2

Best response (BR): the strategy with highest payoff for a player, given knowledge of the other players' strategies

$$\pi^{2,BR} = BR(\pi^1 = (1,0)) = (1,0)$$

$$\pi^{2,BR} = BR(\pi^1 = (0,1)) = (0,1)$$

Normal Form Games: Solution Concept

- **Nash Equilibrium:**

  A strategy profile where all players in simultaneous best responses to each other

  $$\max_{\boldsymbol{\pi}} \ \boldsymbol{\pi}^{T} R^{1} \pi^{2} = \pi^{1T} R^{1} \pi^{2} \qquad \text{and} \qquad \max_{\boldsymbol{\pi}} \ \pi^{1T} R^{2} \boldsymbol{\pi} = \pi^{1T} R^{2} \pi^{2}$$

  i.e., no player can do better by unilaterally deviating

- **Nash's theorem [1950]:**

  Every finite game has a mixed strategy Nash equilibrium

- Not unique in general → equilibrium selection problem

Normal Form Games: Solution Concept

		Player 2	
		Football	Ballet
Player 1	Football	2 1	0 0
	Ballet	0 0	1 2

Nash equilibria and their **expected payoffs**:

1. $\pi^1,\pi^2 = (1,0), (1,0) \rightarrow$ **$(2,1)$**
2. $\pi^1,\pi^2 = (0,1), (0,1) \rightarrow$ **$(1,2)$**
3. $\pi^1,\pi^2 = (\frac{2}{3},\frac{1}{3}), (\frac{1}{3},\frac{2}{3}) \rightarrow$ **$(\frac{2}{3},\frac{2}{3})$**

- Very different outcomes!
- Intractable in general [Daskalakis et al., 2009]
  - Though polynomial-time computable for two-player zero-sum games

Normal Form Games: Solution Concept

		Player 2	
		Stop	Go
Player 1	Stop	0 0	0 1
	Go	1 0	-100 -100

Nash equilibria and their **expected payoffs**:

1. $\pi^1,\pi^2 = (0,1), (1,0) \rightarrow$ **$(1,0)$**
2. $\pi^1,\pi^2 = (1,0), (0,1) \rightarrow$ **$(0,1)$**
3. $\pi^1,\pi^2 = ({}^{100}/{}_{101},{}^{1}/{}_{101}), ({}^{100}/{}_{101},{}^{1}/{}_{101}) \rightarrow$ **$(0,0)$**

3rd equilibrium may seem reasonable, but >0 probability of (-100,-100) reward for both players!

Normal Form Games: Solution Concept

A better alternative might be to play the distribution on the right:

Player 2

Player 1	Stop	Go
Stop	0 0	0 1
Go	1 0	-100 -100

Player 2

Player 1	Stop	Go
Stop	0%	50%
Go	50%	0%

Unfortunately, no set of **independent** mixed strategies can result in this joint distribution!

Normal Form Games: Solution Concept

- **Idea:** address the issue of independent randomness by using a joint distribution
  - Correlated equilibria

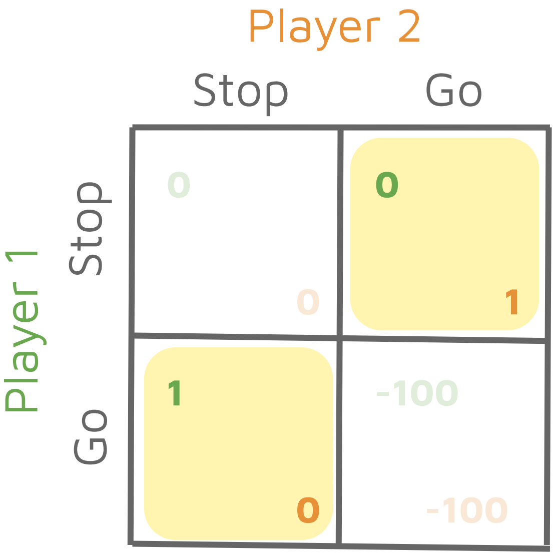
		Player 2	
		Stop	Go
Player 1	Stop	0 0	0 1
	Go	1 0	-100 -100

A correlated equilibrium is a distribution, D, over strategy profiles such that for every player *i*:

$$E_{\underbrace{a\sim D}_{\text{Joint action samples}}} [r^i(a^i, a^{-i}) \mid \underbrace{a^i}_{\text{Sampled action for player i}}] \geqslant \max_{\mathbf{a}} E_{a\sim D} [r^i(\mathbf{a}, a^{-i}) \mid a^i]$$

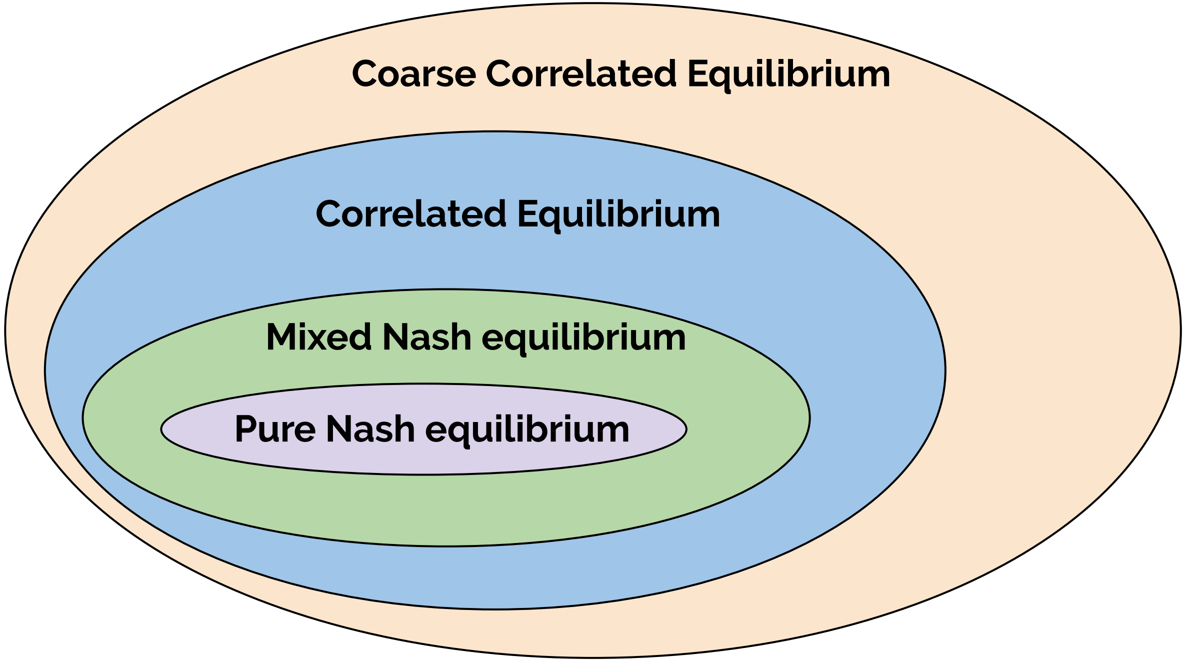
Normal Form Games: Solution Concept

- **Idea:** address the issue of independent randomness by using a joint distribution
  - Correlated equilibria



		Player 2	
		Stop	Go
Player 1	Stop	0 0	0 1
	Go	1 0	-100 -100

Topology of Solution Concepts



Coarse Correlated Equilibrium

Correlated Equilibrium

Mixed Nash equilibrium

Pure Nash equilibrium

From Normal Form to Markov Games

Normal Form Games	Definitions: ● Model ● Solution concepts	Algorithms Based on Best Response
Markov Games	Definitions: ● Model ● Optimal policy	Learning in Markov Games (Part II)

Normal Form Games: Algorithms

So far: solution concepts (e.g., Nash Equilibria) given full knowledge of game

Learning dynamics: do the dynamical interactions of players *with limited knowledge* lead to these solution concepts?

Normal Form Games: Algorithms

Let’s weaken our assumptions:

- Players interact in rounds
- Each player knows their own strategy, but not the full payoff table
- After each round, each player observes their pure strategies’ expected payoffs:

Player 1 observes vector $R^1\ \pi^2$

Player 2 observes vector $\pi^{1^T}\ R^2$

Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:
  - Play a best response w.r.t. history of play in the *T* previous rounds

$$\pi^1 \in \operatorname{argmax}_\pi \pi^T({}^1/_T \underbrace{\Sigma_t R^1\pi_t^2)}_{\textbf{Observed payoff vector in round } \boldsymbol{t}}$$

$$\pi^1 \in \operatorname{argmax}_\pi \pi^T R^1 (\underbrace{\Sigma_t {}^1/_T \pi_t^2}_{\textbf{Time-average opponent play}})$$

  - “Fictitious” in the sense that each player maintains a belief over opponent strategies according the play history

Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:

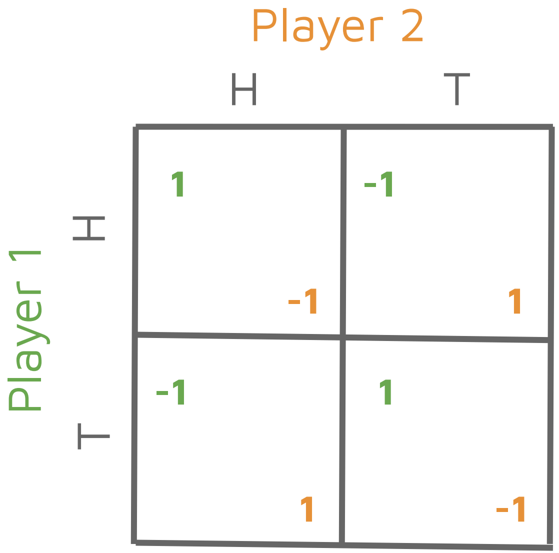
		Player 2	
		H	T
Player 1	H	1 -1	-1 1
	T	-1 1	1 -1

Unique mixed Nash:

$$\pi_t^1 = (\tfrac{1}{2}, \tfrac{1}{2}), \ \pi_t^2 = (\tfrac{1}{2}, \tfrac{1}{2})$$

Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:



Unique mixed Nash:

$\pi_t^1 = (\frac{1}{2}, \frac{1}{2})$, $\pi_t^2 = (\frac{1}{2}, \frac{1}{2})$

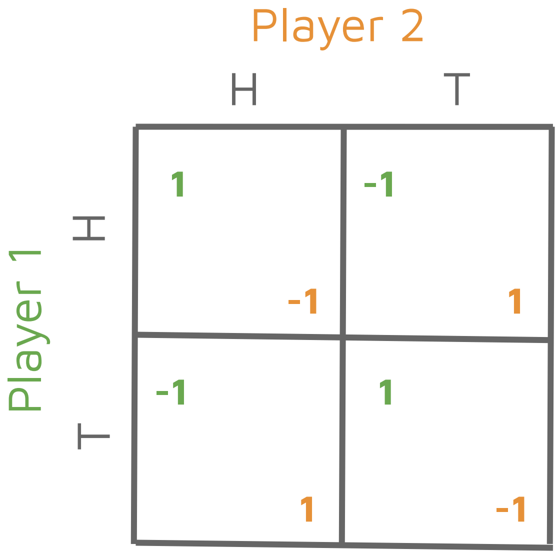
t	π_t^1	π_t^2	n_t^1 (H,T)	n_t^2 (H,T)
0			(0, 2)	(0, 0)
1				
2				
3				
4				
5				
6				
7				
8				

Counts of Player 1's taken actions

Counts of Player 2's taken actions

Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:



	H	T
H	1, -1	-1, 1
T	-1, 1	1, -1

Unique mixed Nash:

$\pi_t^1 = (½, ½), \ \pi_t^2 = (½, ½)$

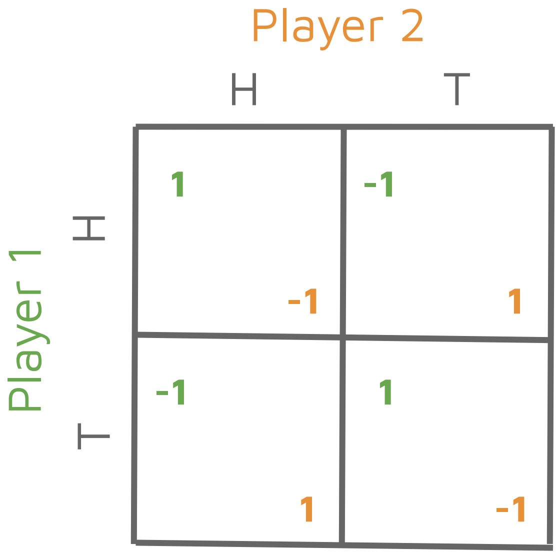
t	π_t^1	π_t^2	n^1_t (H,T)	n^2_t (H,T)
0			(0, 2)	(0, 0)
1	H	H		
2				
3				
4				
5				
6				
7				
8				

Player 2 chooses argmin of n^1_{t-1}

Player 1 chooses argmax of n^2_{t-1}

Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:



Unique mixed Nash:

$\pi_t^1 = (½, ½), \pi_t^2 = (½, ½)$

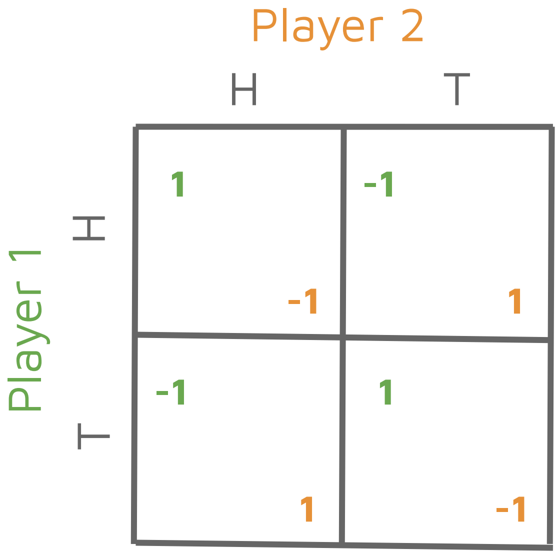
t	π_t^1	π_t^2	n^1_t (H,T)	n^2_t (H,T)
0			(0, 2)	(0, 0)
1	H	H	(1, 2)	(1, 0)
2				
3				
4				
5				
6				
7				
8				

Player 2 chooses argmin of n^1_{t-1}

Player 1 chooses argmax of n^2_{t-1}

Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:



	H	T
H	1, -1	-1, 1
T	-1, 1	1, -1

Unique mixed Nash:

$\pi_t^1 = (\frac{1}{2}, \frac{1}{2})$, $\pi_t^2 = (\frac{1}{2}, \frac{1}{2})$

t	π_t^1	π_t^2	n^1_t (H,T)	n^2_t (H,T)
0			(0, 2)	(0, 0)
1	H	H	(1, 2)	(1, 0)
2	H	H		
3				
4				
5				
6				
7				
8				

Player 2 chooses argmin of n^1_{t-1}

Player 1 chooses argmax of n^2_{t-1}

Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:

Player 2

Player 1	H	T
H	1, -1	-1, 1
T	-1, 1	1, -1

Unique mixed Nash:

$\pi_t^1 = (½, ½), \ \pi_t^2 = (½, ½)$

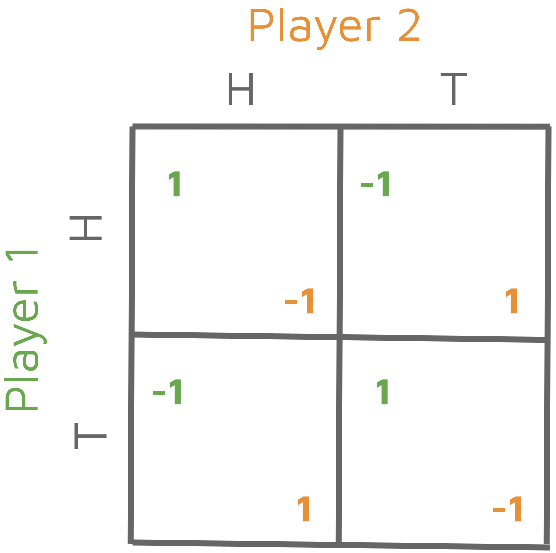
t	π_t^1	π_t^2	n^1_t (H,T)	n^2_t (H,T)
0			(0, 2)	(0, 0)
1	H	H	(1, 2)	(1, 0)
2	H	H	(2, 2)	(2, 0)
3				
4				
5				
6				
7				
8				

Player 2 chooses argmin of n^1_{t-1}

Player 1 chooses argmax of n^2_{t-1}

Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:



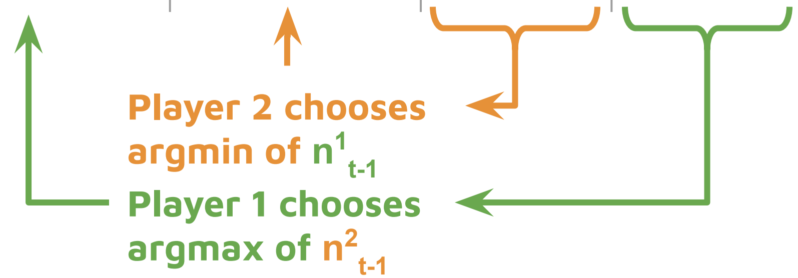
Player 2

Player 1	H	T
H	1, -1	-1, 1
T	-1, 1	1, -1

Unique mixed Nash:

$\pi_t^1 = (½, ½)$, $\pi_t^2 = (½, ½)$

t	π_t^1	π_t^2	n_t^1 (H,T)	n_t^2 (H,T)
0			(0, 2)	(0, 0)
1	H	H	(1, 2)	(1, 0)
2	H	H	(2, 2)	(2, 0)
3	H	T		
4				
5				
6				
7				
8				



Player 2 chooses argmin of n_{t-1}^1

Player 1 chooses argmax of n_{t-1}^2

Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:

Player 2

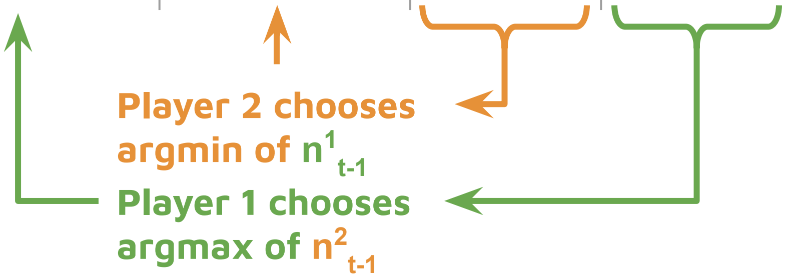
Player 1

	H	T
H	1, -1	-1, 1
T	-1, 1	1, -1

Unique mixed Nash:

$\pi_t^1 = (½, ½)$, $\pi_t^2 = (½, ½)$

t	π_t^1	π_t^2	n^1_t (H,T)	n^2_t (H,T)
0			(0, 2)	(0, 0)
1	H	H	(1, 2)	(1, 0)
2	H	H	(2, 2)	(2, 0)
3	H	T	(3, 2)	(2, 1)
4				
5				
6				
7				
8				

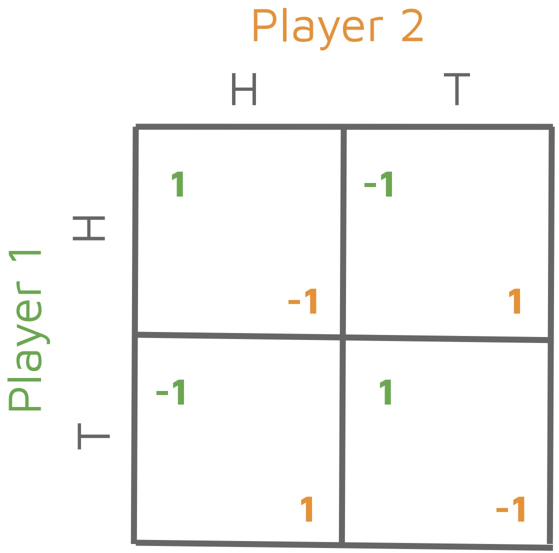


Player 2 chooses argmin of n^1_{t-1}

Player 1 chooses argmax of n^2_{t-1}

Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:

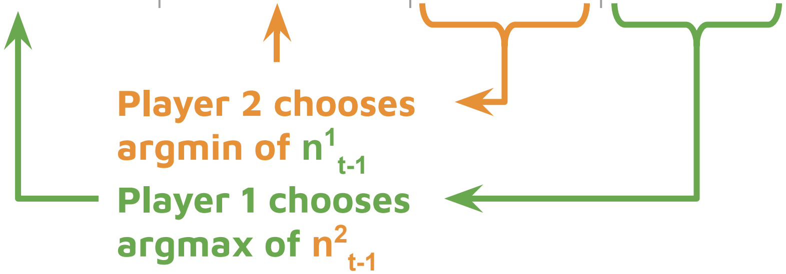


	H	T
H	1, -1	-1, 1
T	-1, 1	1, -1

Unique mixed Nash:

$\pi_t^1 = (½, ½)$, $\pi_t^2 = (½, ½)$

t	π_t^1	π_t^2	n^1_t (H,T)	n^2_t (H,T)
0			(0, 2)	(0, 0)
1	H	H	(1, 2)	(1, 0)
2	H	H	(2, 2)	(2, 0)
3	H	T	(3, 2)	(2, 1)
4	H	T		
5				
6				
7				
8				

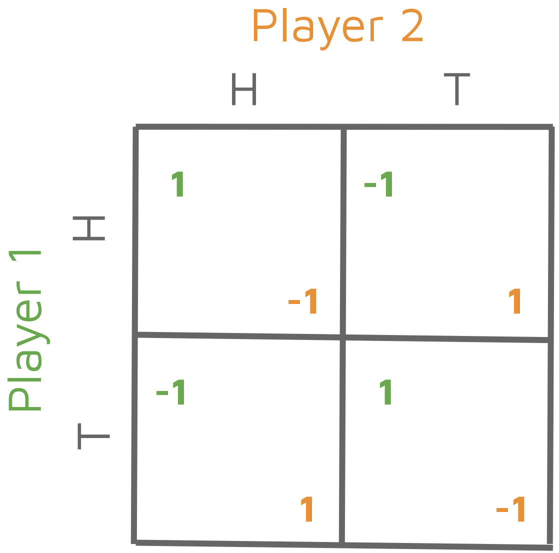


Player 2 chooses argmin of n^1_{t-1}

Player 1 chooses argmax of n^2_{t-1}

Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:



Player 2

Player 1

	H	T
H	1 -1	-1 1
T	-1 1	1 -1

Unique mixed Nash:

$\pi_t^1 = (½, ½)$, $\pi_t^2 = (½, ½)$

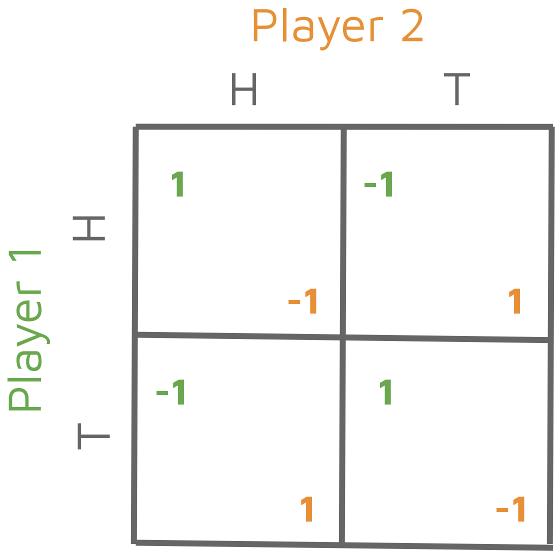
t	π_t^1	π_t^2	n^1_t (H,T)	n^2_t (H,T)
0			(0, 2)	(0, 0)
1	H	H	(1, 2)	(1, 0)
2	H	H	(2, 2)	(2, 0)
3	H	T	(3, 2)	(2, 1)
4	H	T	(4, 2)	(2, 2)
5				
6				
7				
8				

Player 2 chooses argmin of n^1_{t-1}

Player 1 chooses argmax of n^2_{t-1}

Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:



Unique mixed Nash:

$\pi_t^1 = (½, ½)$, $\pi_t^2 = (½, ½)$

t	π_t^1	π_t^2	n^1_t (H,T)	n^2_t (H,T)
0			(0, 2)	(0, 0)
1	H	H	(1, 2)	(1, 0)
2	H	H	(2, 2)	(2, 0)
3	H	T	(3, 2)	(2, 1)
4	H	T	(4, 2)	(2, 2)
5	T	T	(4, 3)	(2, 3)
6	T	T	(4, 4)	(2, 4)
7	T	H	(4, 5)	(3, 4)
8	T	H	(4, 6)	(4, 4)

Player 2 chooses argmin of n^1_{t-1}

Player 1 chooses argmax of n^2_{t-1}

Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:

Player 2

Player 1	H	T
H	1 -1	-1 1
T	-1 1	1 -1

Unique mixed Nash:

$\pi_t^1 = (\frac{1}{2}, \frac{1}{2})$, $\pi_t^2 = (\frac{1}{2}, \frac{1}{2})$

t	π_t^1	π_t^2	n^1_t (H,T)	n^2_t (H,T)
0			(0, 2)	(0, 0)
1	H	H	(1, 2)	(1, 0)
2	H	H	(2, 2)	(2, 0)
3	H	T	(3, 2)	(2, 1)
4	H	T	(4, 2)	(2, 2)
5	T	T	(4, 3)	(2, 3)
6	T	T	(4, 4)	(2, 4)
7	T	H	(4, 5)	(3, 4)
8	T	H	(4, 6)	(4, 4)

Play will continue to cycle deterministically, with time-average strategies converging to Nash

Normal Form Games: Fictitious Play

- When does Fictitious Play converge, and to what?

- Average-time strategies of fictitious players converge to a Nash in:
    - Two-player zero-sum games
    - 2x2 games
    - Potential games
    - …

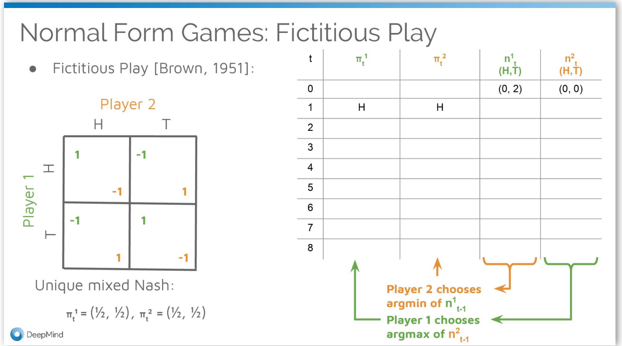
- Not guaranteed in general! Try it on modified RPS:

Player 2

		Rock	Paper	Scissors
Player 1	Rock	0,0	0,1	1,0
	Paper	1,0	0,0	0,1
	Scissors	0,1	1,0	0,0

Normal Form Games: Oracle Algorithms

- **Goal:** compute a Nash equilibrium of the game (AKA “solve” the game)
- **Insight:** computing a best response is generally cheaper than solving the game



Normal Form Games: Fictitious Play

- Fictitious Play [Brown, 1951]:

Player 2

Player 1	H	T
H	1 -1	-1 1
T	-1 1	1 -1

Unique mixed Nash:

$\pi_t^1 = (½, ½)$, $\pi_t^2 = (½, ½)$

t	π_t^1	π_t^2	n_t^1 (H,T)	n_t^2 (H,T)
0			(0, 2)	(0, 0)
1	H	H		
2				
3				
4				
5				
6				
7				
8				

Player 2 chooses argmin of n^1_{t-1}

Player 1 chooses argmax of n^2_{t-1}

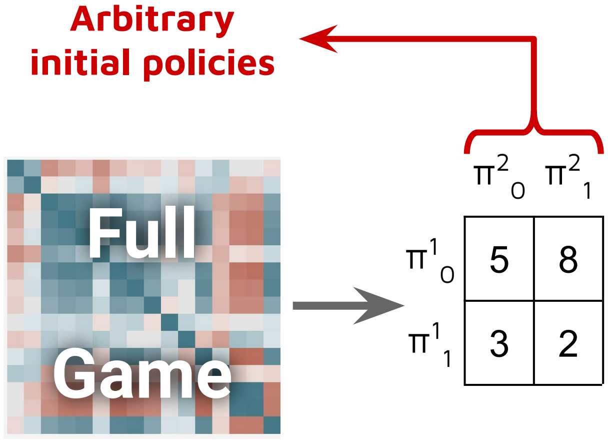
DeepMind

  - Reduction to a single-player optimization problem
  - Due to their efficiency, BR algorithms sometimes called “oracles”
- Oracle algorithms use BR to solve the game:
  - Single/double oracle: one/both player(s) use the oracle algorithm

DeepMind

Normal Form Games: Oracle Algorithms

- Double oracle [McMahan et al., 2003]:



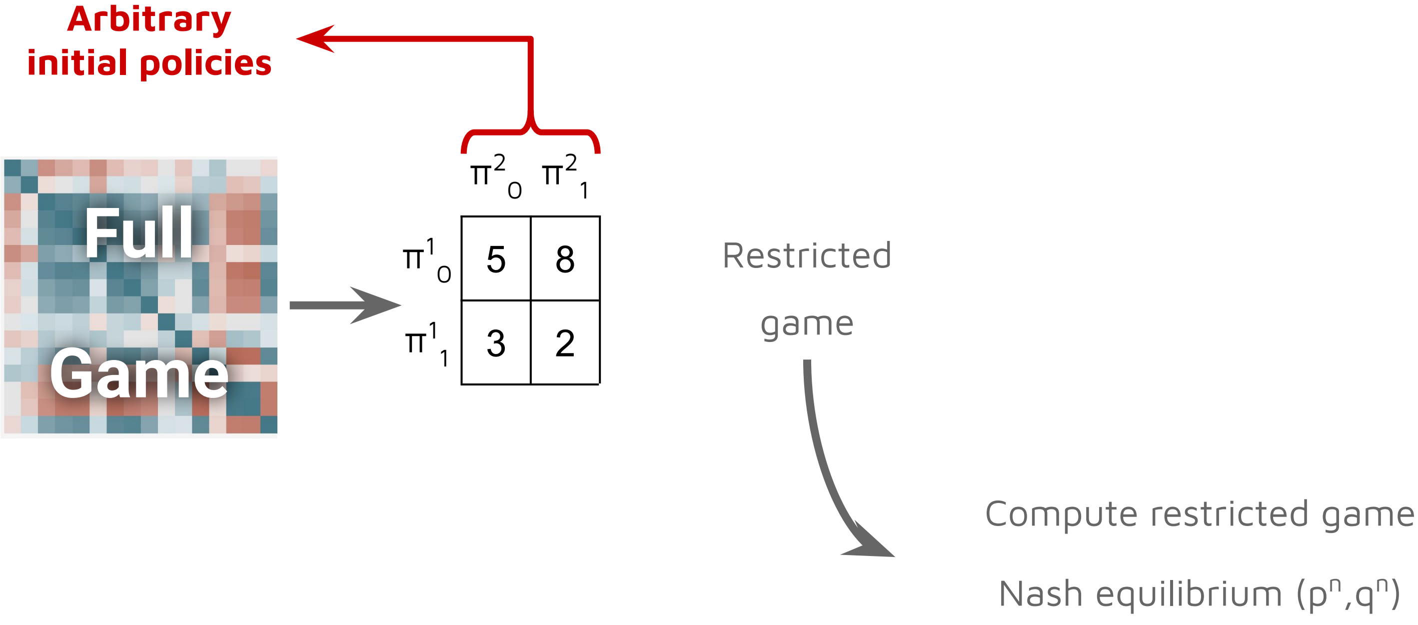
Arbitrary initial policies

$$\begin{array}{c|c|c|}
 & \pi^2_0 & \pi^2_1 \\
\hline
\pi^1_0 & 5 & 8 \\
\hline
\pi^1_1 & 3 & 2 \\
\hline
\end{array}$$

Restricted game

Normal Form Games: Oracle Algorithms

- Double oracle [McMahan et al., 2003]:



Arbitrary initial policies

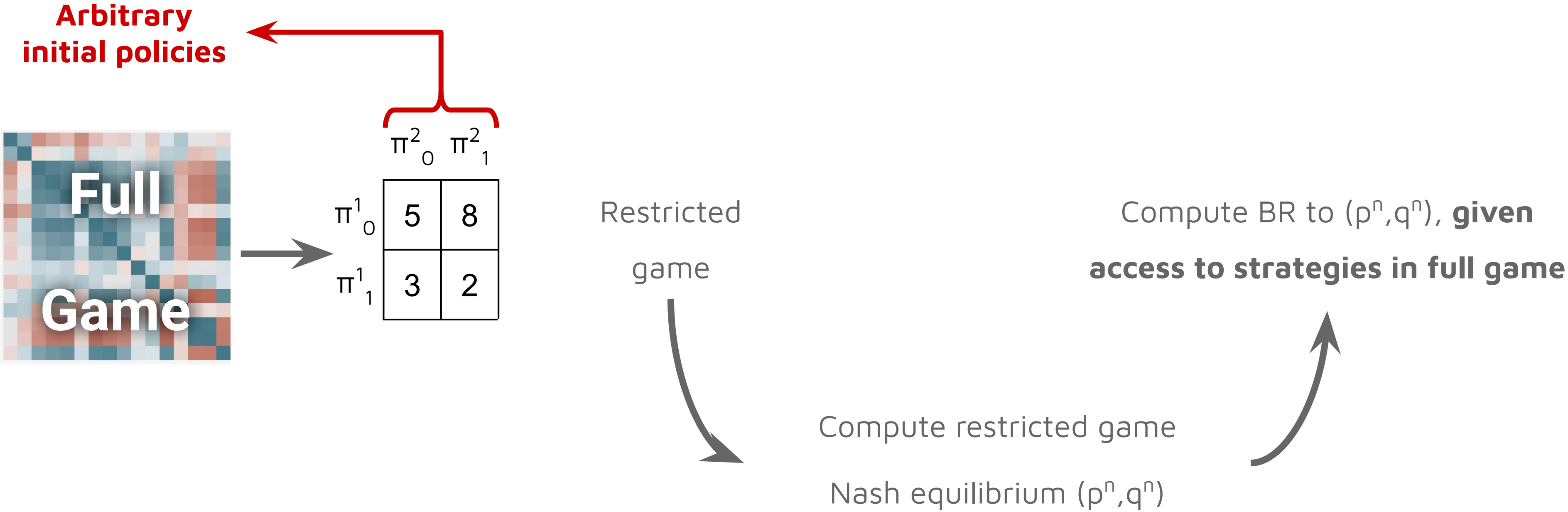
$$\begin{array}{c|cc} & \pi^2_0 & \pi^2_1 \\ \hline \pi^1_0 & 5 & 8 \\ \pi^1_1 & 3 & 2 \end{array}$$

Restricted game

Compute restricted game Nash equilibrium (p^n, q^n)

Normal Form Games: Oracle Algorithms

- Double oracle [McMahan et al., 2003]:



Arbitrary initial policies

$$\begin{array}{c|c|c|}
 & \pi^2_0 & \pi^2_1 \\
\hline
\pi^1_0 & 5 & 8 \\
\hline
\pi^1_1 & 3 & 2 \\
\hline
\end{array}$$

Restricted

game

Compute restricted game

Nash equilibrium (p^n,q^n)

Compute BR to (p^n,q^n), **given**

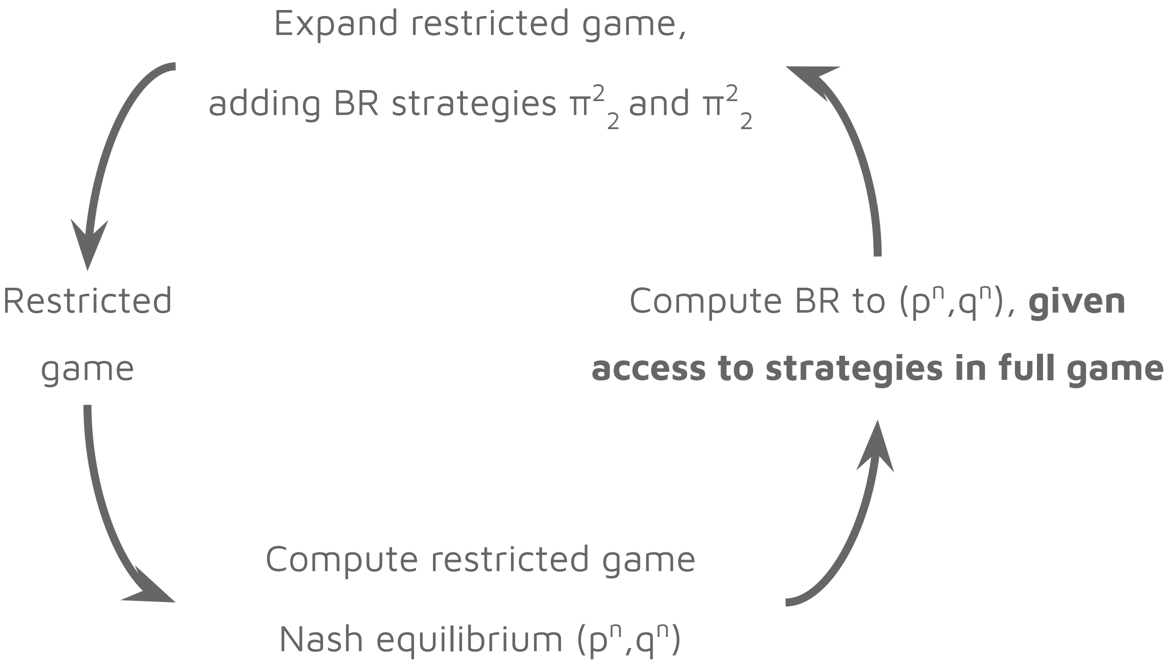
access to strategies in full game

Normal Form Games: Oracle Algorithms

- Double oracle [McMahan et al., 2003]:

Arbitrary initial policies

$$\begin{array}{c|c|c|c|}
 & \pi^2_0 & \pi^2_1 & \pi^2_2 \\ \hline
\pi^1_0 & 5 & 8 & 10 \\ \hline
\pi^1_1 & 3 & 2 & 2 \\ \hline
\pi^1_2 & 7 & 8 & 0 \\ \hline
\end{array}$$



Expand restricted game, adding BR strategies π^2_2 and π^2_2

Restricted game

Compute restricted game Nash equilibrium (p^n,q^n)

Compute BR to (p^n,q^n), **given access to strategies in full game**

Normal Form Games: Oracle Algorithms

- Double oracle [McMahan et al., 2003]:

	R
R	0

- Iteration 0: restricted game of R vs. R
- Iteration 1:
  - Solve restricted game:

    (1, 0, 0), (1, 0, 0)

  - Unrestricted BR^1_1,BR^2_1 = P, P

Normal Form Games: Oracle Algorithms

- Double oracle [McMahan et al., 2003]:

	R	P
R	0	-1
P	1	0

- Iteration 0: restricted game of R vs. R
- Iteration 1:
  - Solve restricted game:

    (1, 0, 0), (1, 0, 0)

  - Unrestricted BR^1_1, BR^2_1 = P, P
- Iteration 2:
  - Solve restricted game:

    (0, 1, 0), (0, 1, 0)

  - Unrestricted BR^1_2, BR^2_2 = S, S

Normal Form Games: Oracle Algorithms

- Double oracle [McMahan et al., 2003]:

	R	P	S
R	0	-1	1
P	1	0	-1
S	-1	1	0

- Iteration 0: restricted game of R vs. R
- Iteration 1:
  - Solve restricted game:

    (1, 0, 0), (1, 0, 0)

  - Unrestricted BR^1_1,BR^2_1 = P, P
- Iteration 2:
  - Solve restricted game:

    (0, 1, 0), (0, 1, 0)

  - Unrestricted BR^1_2,BR^2_2 = S, S
- Iteration 2:
  - Solve restricted game:

    (⅓, ⅓, ⅓), (⅓, ⅓, ⅓)

Normal Form Games: Oracle Algorithms

- Computation time improvements vs. solving full game [McMahan et al., 2003]:

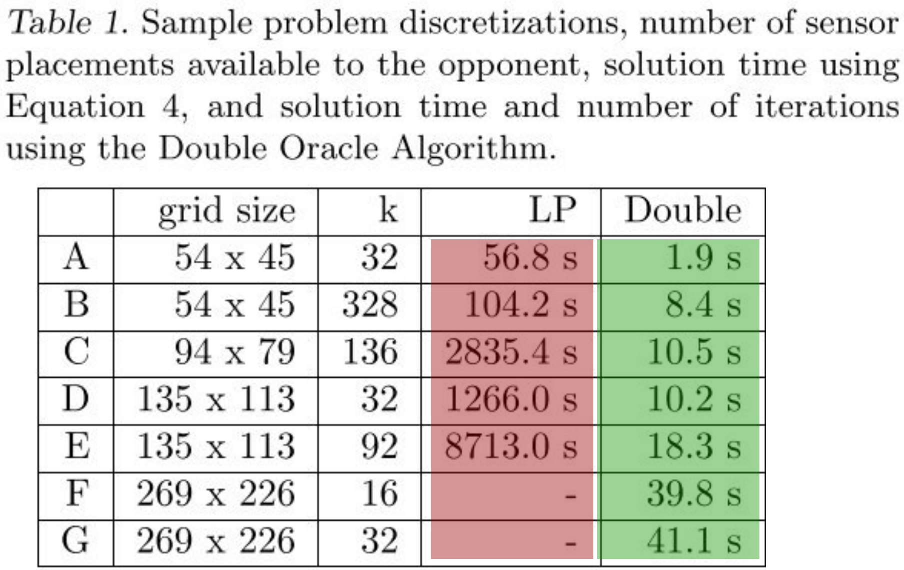


Table 1. Sample problem discretizations, number of sensor placements available to the opponent, solution time using Equation 4, and solution time and number of iterations using the Double Oracle Algorithm.

	grid size	k	LP	Double
A	54 x 45	32	56.8 s	1.9 s
B	54 x 45	328	104.2 s	8.4 s
C	94 x 79	136	2835.4 s	10.5 s
D	135 x 113	32	1266.0 s	10.2 s
E	135 x 113	92	8713.0 s	18.3 s
F	269 x 226	16	-	39.8 s
G	269 x 226	32	-	41.1 s

Normal Form Games: Algorithms

- When does Double Oracle converge, and to what?

- Convergence guaranteed for two-player finite games
    - Proof: worst case, the restricted game just expands to the full game

- Convergence to minimax equilibrium in finite games [McMahan et al. 2003]

From Normal Form to Markov Games

Normal Form Games

Definitions:

- Model
- Solution concepts

Algorithms Based on Best Response

Markov Games

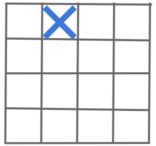
Definitions:

- Model
- Optimal policy

Learning in Markov Games (Part II)

Markov Games: Description

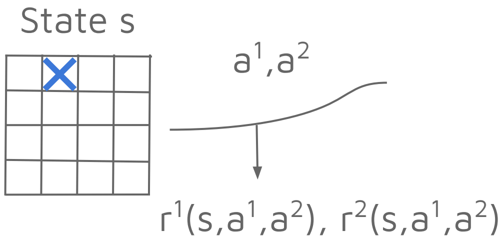
State s



Setting (e.g., in a 2-player game):

- Agents in environment with state s

Markov Games: Description



State s

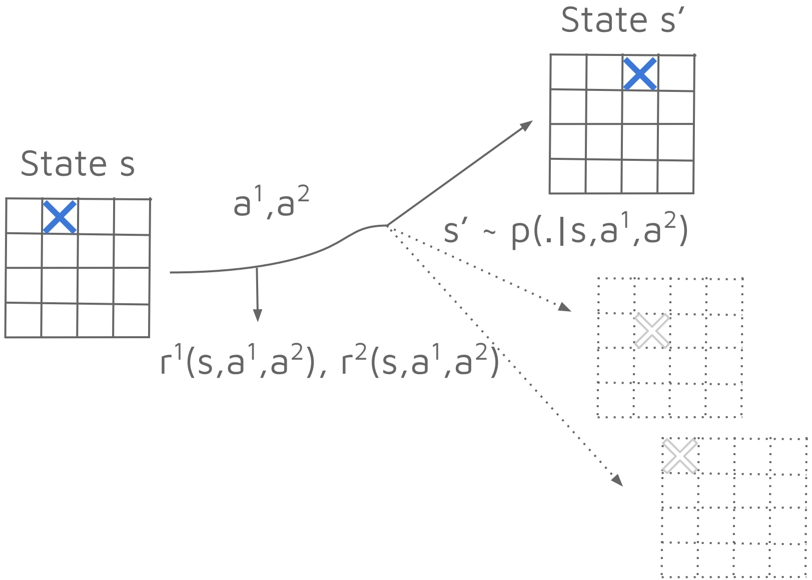
a^1, a^2

$r^1(s, a^1, a^2)$, $r^2(s, a^1, a^2)$

Setting (e.g., in a 2-player game):

- Agents in environment with state s
- Simultaneously select actions a^1 & a^2
- Receive rewards $r^1(s, a^1, a^2)$ & $r^2(s, a^1, a^2)$

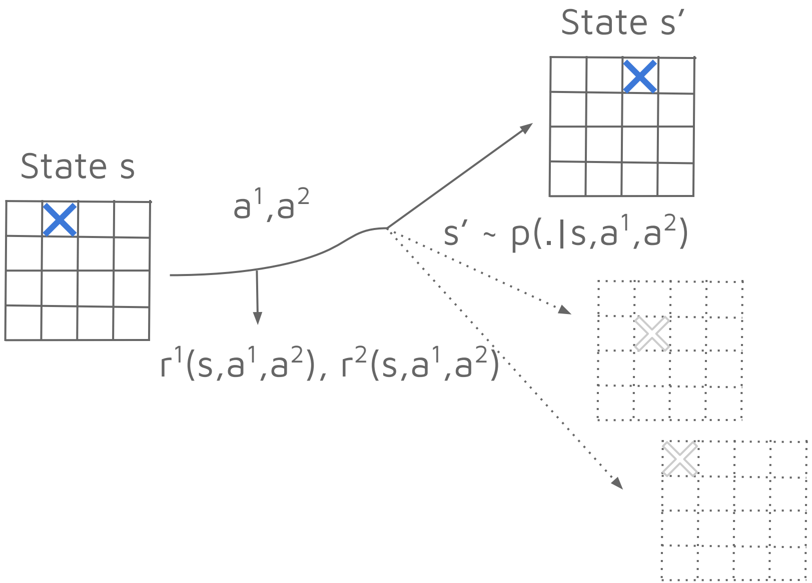
Markov Games: Description



Setting (e.g., in a 2-player game):

- Agents in environment with state s
- Simultaneously select actions a^1 & a^2
- Receive rewards $r^1(s,a^1,a^2)$ & $r^2(s,a^1,a^2)$
- Move to state $s' \sim p(.|s,a^1,a^2)$

Markov Games: Description



State s

a^1,a^2

$r^1(s,a^1,a^2)$, $r^2(s,a^1,a^2)$

State s’

$s' \sim p(.|s,a^1,a^2)$

Setting (e.g., in a 2-player game):

- Agents in environment with state s
- Simultaneously select actions a^1 & a^2
- Receive rewards $r^1(s,a^1,a^2)$ & $r^2(s,a^1,a^2)$
- Move to state $s' \sim p(.|s,a^1,a^2)$

Goal: find the “optimal” policy

If actions are selected according to policies $\pi^1(.|s)$ & $\pi^2(.|s)$, *i.e.*, $a^1 \sim \pi^1(.|s)$ and $a^2 \sim \pi^2(.|s)$:

Player 1 receives $v^1_{\pi 1,\pi 2}(s_0) = E_{\pi 1,\pi 2}\,[r^1(s_0,a^1_0,a^2_0) + \gamma\, r^1(s_1,a^1_1,a^2_1) + \ldots]$

Player 2 receives $v^2_{\pi 1,\pi 2}(s_0) = E_{\pi 1,\pi 2}\,[r^2(s_0,a^1_0,a^2_0) + \underbrace{\gamma}\, r^2(s_1,a^1_1,a^2_1) + \ldots]$

Discount factor ∈ [0,1)

DeepMind

From Normal Form to Markov Games

Normal Form Games

Definitions:

- Model
- Solution concepts

Algorithms Based on Best Response

Markov Games

Definitions:
- Model
- Optimal policy

Learning in Markov Games (Part II)

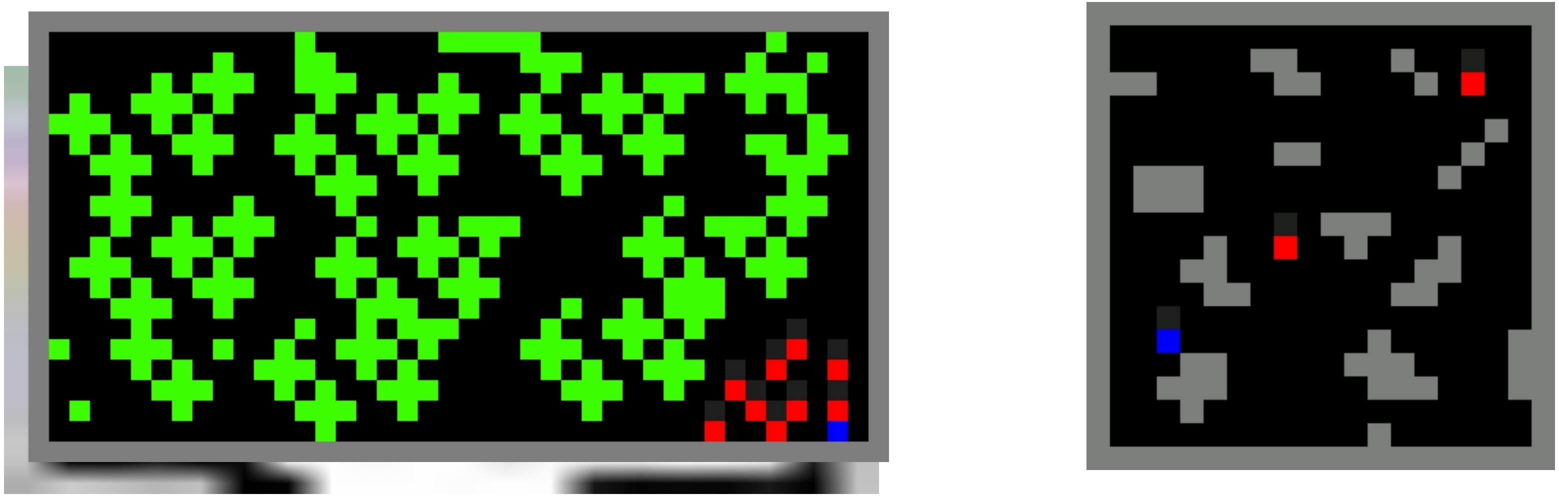
References

- L. S. Shapley. Stochastic Games. In Proc. of the National Academy of Sciences of the United States of America, 1953
- A. J. Hoffman, R. M. Karp. On nonterminating stochastic games. Management Science, 12(5):359–370, 1966.
- M. Pollatschek, B. Avi-Itzhak. Algorithms for Stochastic Games with Geometrical Interpretation. Management Science, 1969
- J. A. Filar, B. Tolwinski. On the Algorithm of Pollatschek and Avi-Itzhak. Springer, 1991.
- M. Lanctot, V. Zambaldi, A. Gruslys, A. Lazaridou, K. Tuyls, J. Pérolat, D. Silver, T. Graepel. A unified game-theoretic approach to multiagent reinforcement learning. NIPS 2017.
- J. Heinrich, D. Silver. Deep Reinforcement Learning from Self-Play in Imperfect-Information Games. arXiv 2016.
- J. Perolat. Reinforcement Learning: The Multi-Player Case. PhD thesis.

3. Social Learning

DeepMind

Social dilemmas



Situations where any individual may profit from selfishness unless too many individuals choose the selfish option, in which case the whole group loses.

"Social dilemmas expose tensions between collective and individual rationality"

-Anatol Rapoport (1974)

Social dilemmas (Liebrand 1983, Macy & Flache 2002)

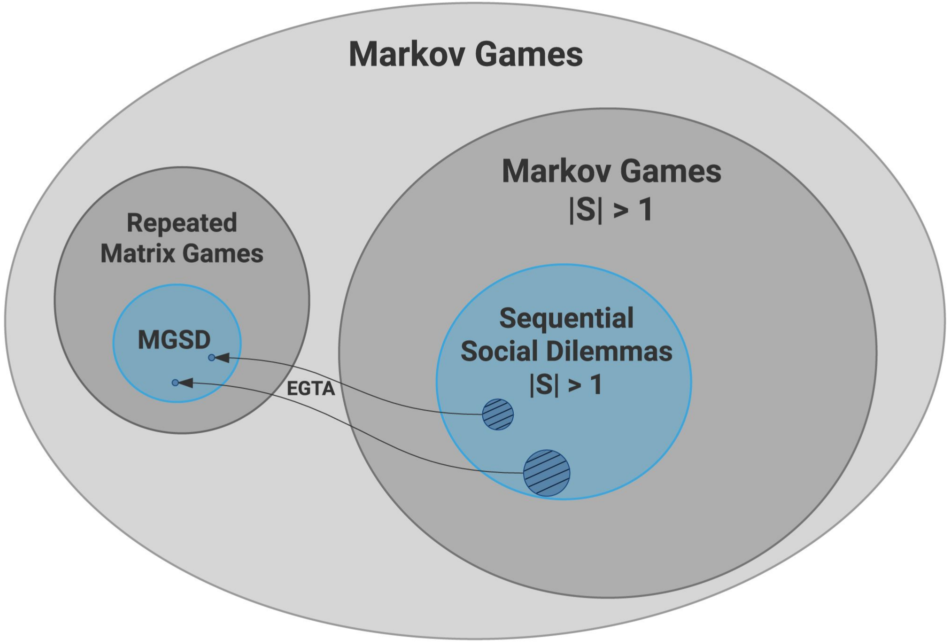
	C	D
C	R, R	S, T
D	T, S	P, P

- **R**eward for mutual cooperation
- **S**ucker for cooperating with defector
- **P**unishment for mutual defection
- **T**emptation to defect on a cooperator

1. **R** > **P** (mutual cooperation better than mutual defection)
2. **R** > **S** (mutual cooperation better than being exploited)
3. **T > P** (being greedy better than being punished)
4. either (fear) **S < P** (being sucker worse than mutual defection)

   ... or (greed) **T > R** (being greedy better than mutual cooperation)

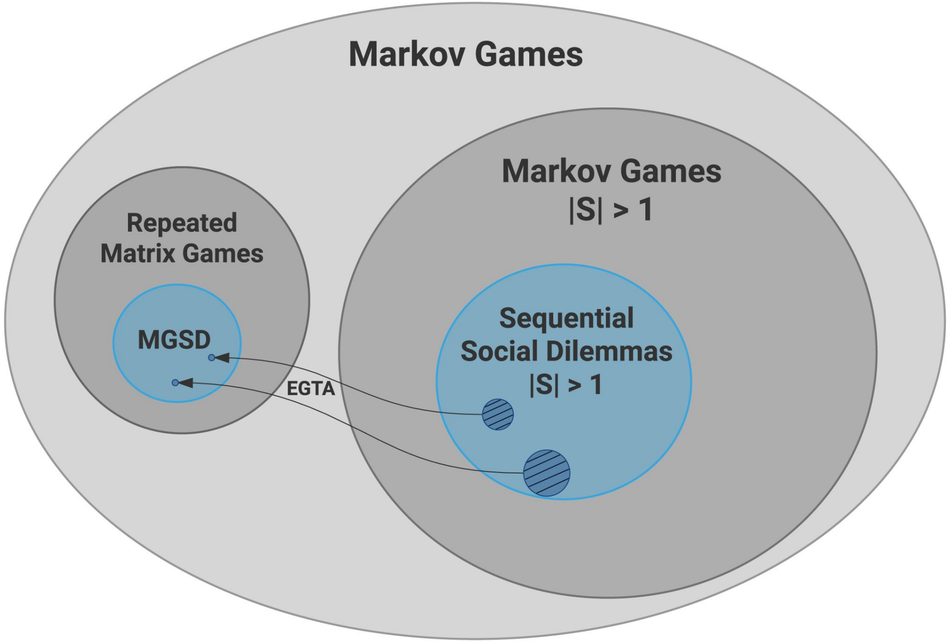
Sequential Social Dilemmas



- MGSD = Matrix Game Social Dilemma
- SSD = Sequential Social Dilemma
- EGTA = Empirical Game Theory Analysis

- MGSDs are defined as repeated matrix games for which the social dilemma inequalities hold.
- The social dilemma inequalities enforce the mixed motivation structure of the game: both competition and cooperation are motivated.
- SSDs are defined by an EGTA mapping to an associated MGSD.

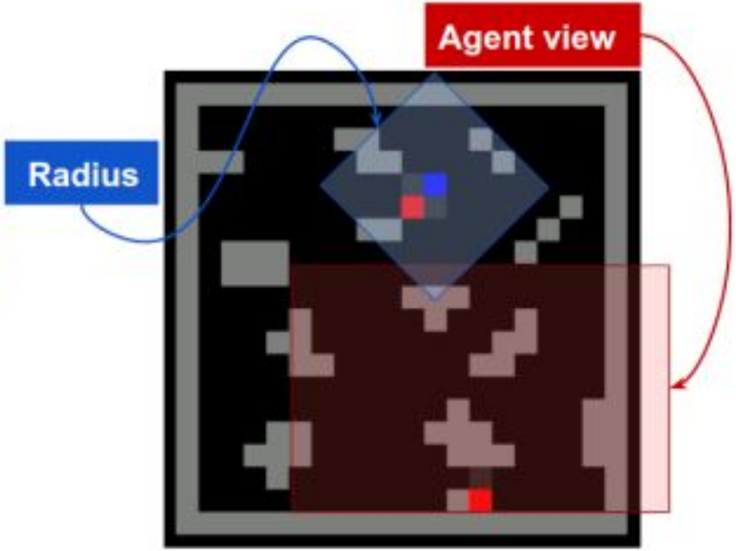
Sequential Social Dilemmas



- MGSD = Matrix Game Social Dilemma
- SSD = Sequential Social Dilemma
- EGTA = Empirical Game Theory Analysis

- Can we ***design*** an agent that can promote cooperation and take fairness into account in SSDs?
- Can we do this based on the Fehr and Schmidt model of inequity aversion?

Examples (Leibo et al. 2017)



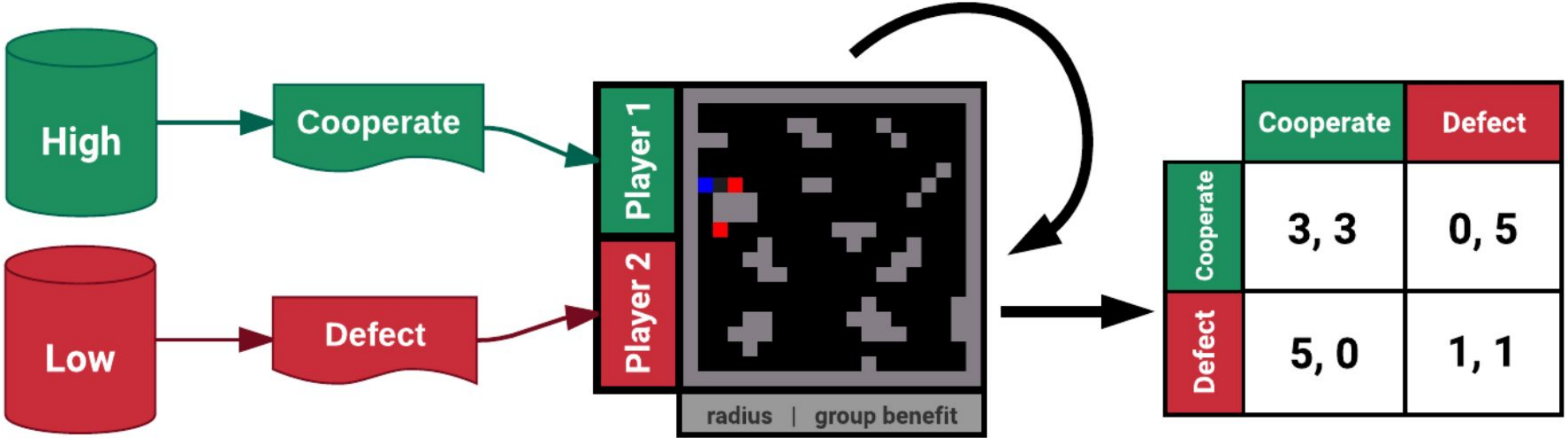
Gathering

- Cooperation = not tagging
- Defection = tagging

Wolfpack

- Cooperation = team capture
- Defection = individual capture

Proving that these are SSDs (by Schelling diagrams)



High

Low

Cooperate

Defect

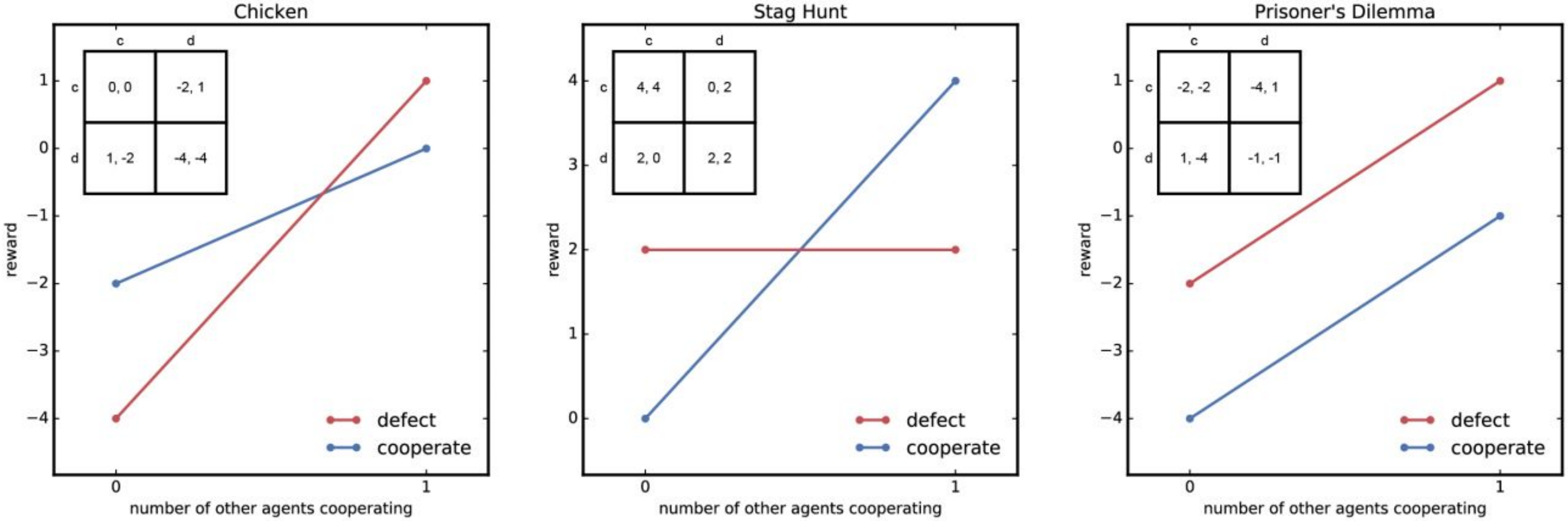
Player 1

Player 2

	Cooperate	Defect
Cooperate	3, 3	0, 5
Defect	5, 0	1, 1

Examples

- Each line shows the payoff to an individual agent (y) for choosing C or D as a function of number of others that chose C (x).



Chicken

	c	d
c	0, 0	-2, 1
d	1, -2	-4, -4

Stag Hunt

	c	d
c	4, 4	0, 2
d	2, 0	2, 2

Prisoner's Dilemma

	c	d
c	-2, -2	-4, 1
d	1, -4	-1, -1

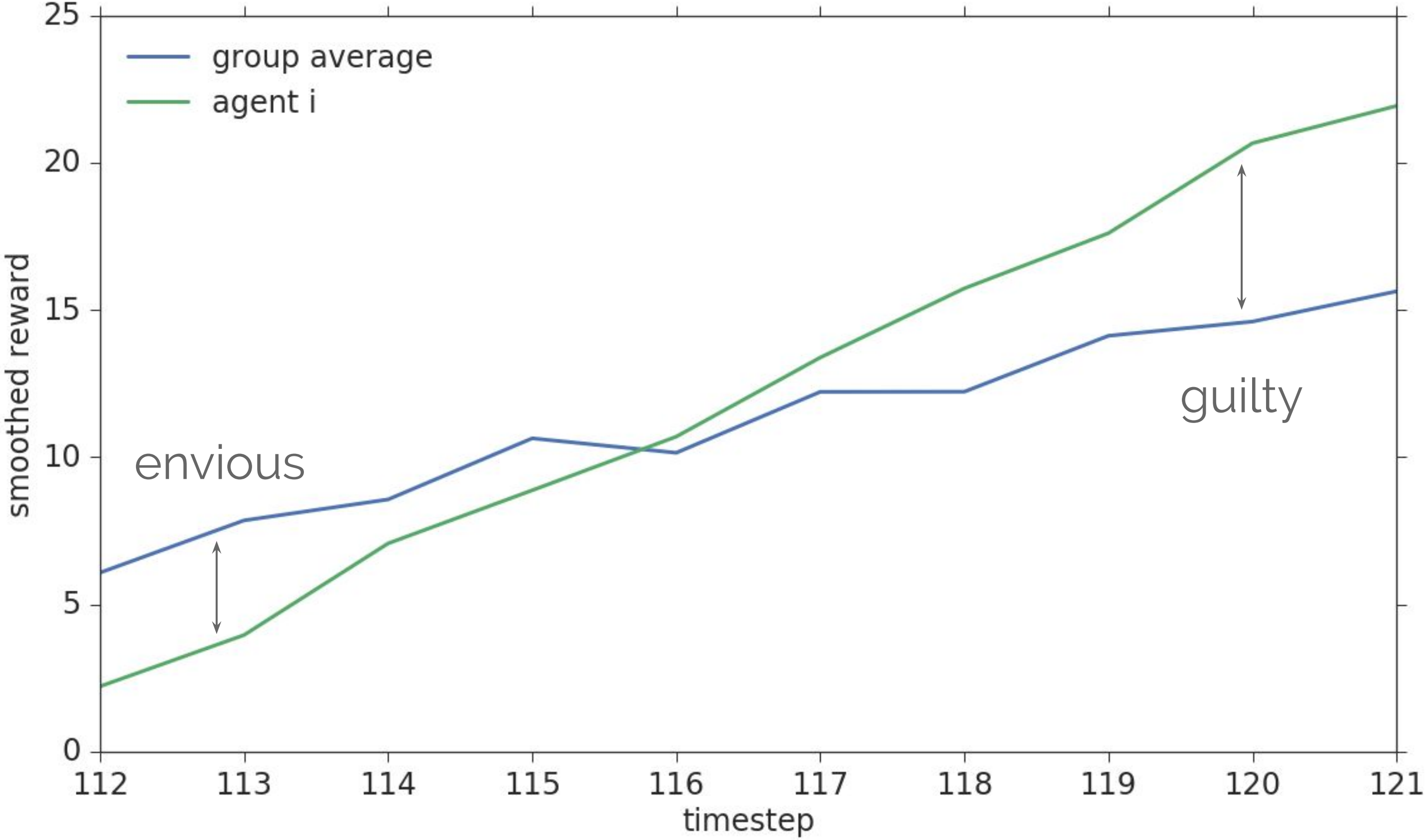
The Fehr and Schmidt model (Fehr and Schmidt, 1999)

$$
\begin{aligned}
U_i(r_i, \ldots r_N) = \;& r_i \\
& - \frac{\alpha_i}{N-1} \sum_{j \neq i} \max\left(r_j - r_i, 0\right) \quad \longleftarrow \quad \text{envy} \\
& - \frac{\beta_i}{N-1} \sum_{j \neq i} \max\left(r_i - r_j, 0\right) \quad \longleftarrow \quad \text{guilt}
\end{aligned}
$$

The inequity-averse agent model (Hughes, Leibo, Tuyls et al. 2018)

$$
\begin{aligned}
u_i(s_i^t, a_i^t) = \quad & r_i(s_i^t, a_i^t, \theta_{ii}) \\
& - \frac{\alpha_i}{N-1} \sum_{j \neq i} \max(e_j^t r_j(s_j^t, a_j^t, \theta_{ij}) && \longleftarrow \text{envy} \\
& \quad - e_i^t r_i(s_i^t, a_i^t, \theta_{ii}), 0) \\
& - \frac{\beta_i}{N-1} \sum_{j \neq i} \max(e_j^t r_i(s_i^t, a_i^t, \theta_{ii}) && \longleftarrow \text{guilt} \\
& \quad - e_j^t r_j(s_j^t, a_j^t, \theta_{ij}), 0) \, ,
\end{aligned}
$$

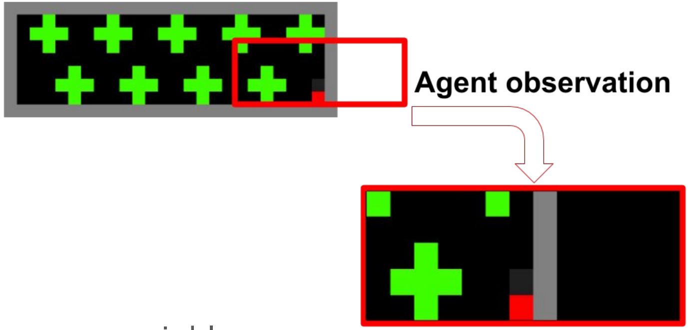
Envy and guilt



The Tragedy of the Commons (Hardin 1968)

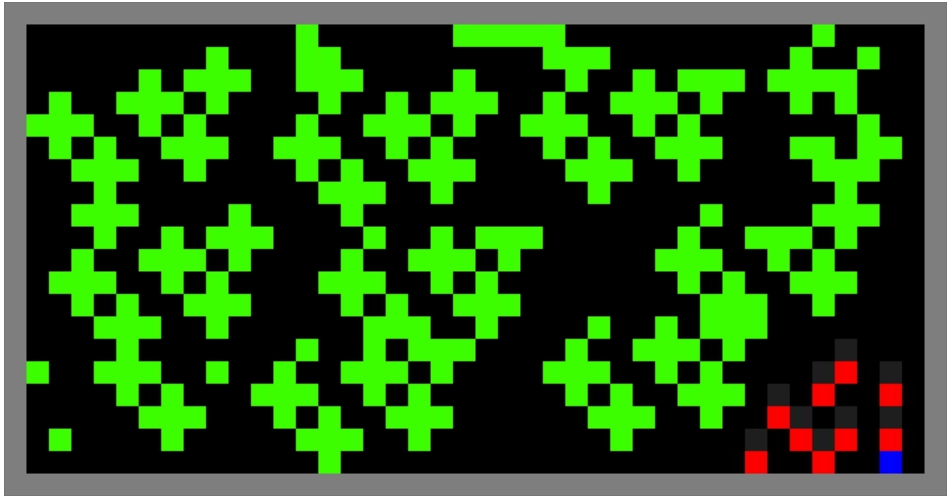
Tension between collective and individual rationality.

The Commons Game (Leibo, Perolat et al. 2017)



1. Agents move around on a grid world.
2. Agents are only rewarded when they collect an apple.
3. The apple growth rule is density dependent. So apples grow more quickly adjacent to nearby apples.
4. If all the apples in a local patch are removed then none grow back.
5. Episodes last 1000 steps, after which the game resets to its initial condition.
6. Agents have a “time-out beam” with which they can zap one another. A zapped agent gets removed from the game for 25 steps.

The Commons Game



- N = 10 players
- Each agent can individually profit from selfishness, but the group is doomed if all elect that option.
- There can be a “tragedy of the commons”   (G. Hardin 1968)

DeepMind

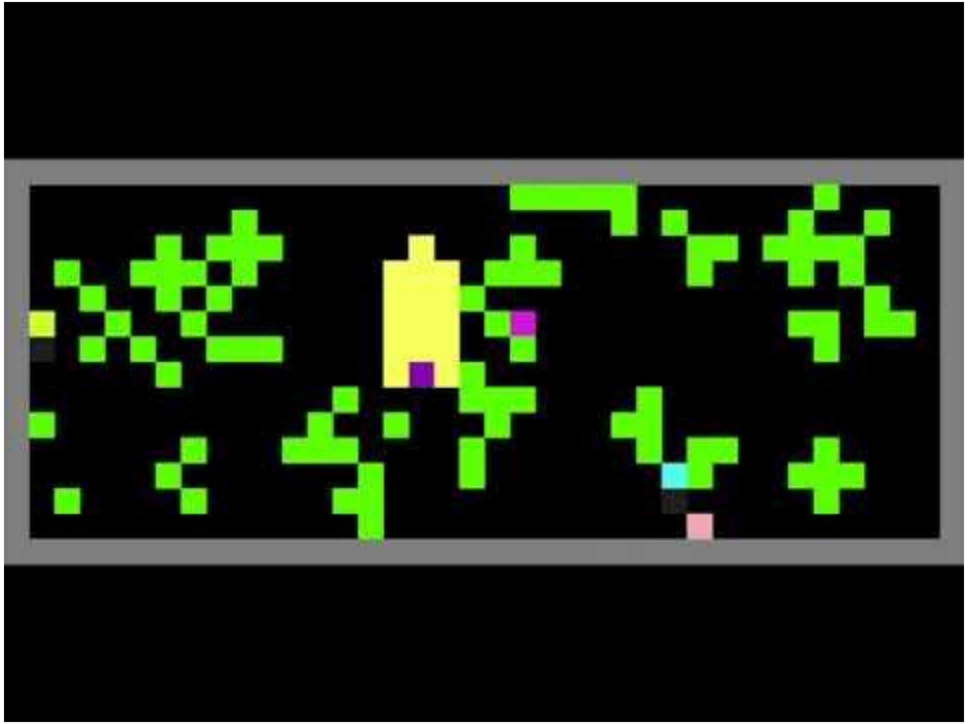
Multiple social outcome metrics

Societal-level measurement is complicated!

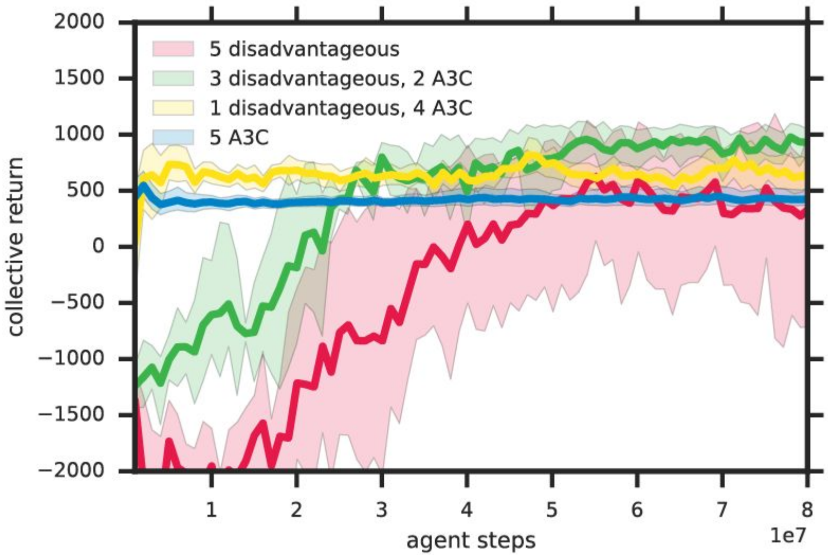
1. **Utilitarian efficiency (U)** = total reward (sum over all players)
2. **Sustainability (S)** = average time of reward collection in episode
3. **Peacefulness (P)** = average number of unzapped agent steps

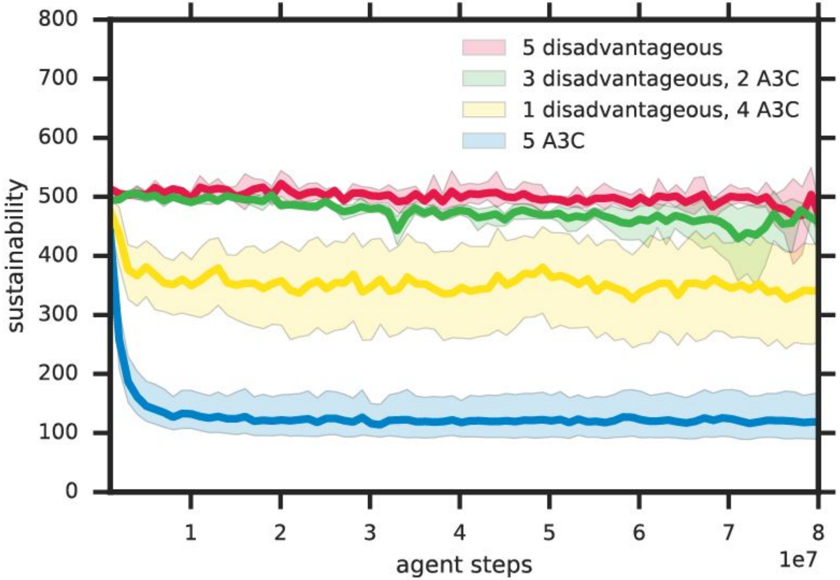
Only illustrate a couple of experiments

Envious agents become police



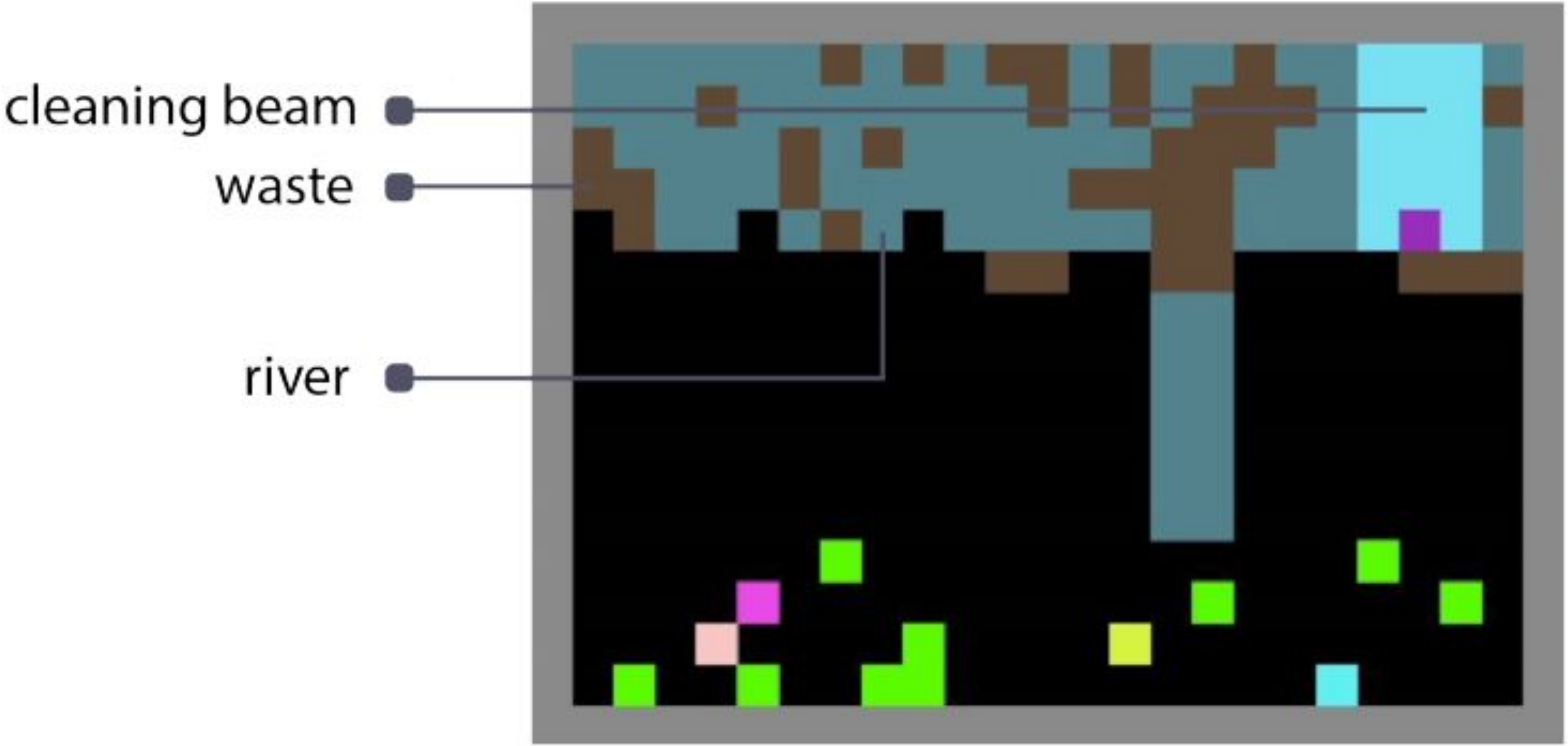
Envious agents become police



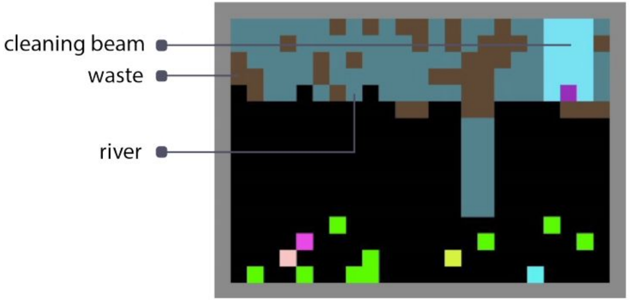


DeepMind

The Public Goods Game (Hughes, Leibo, Tuyls et al. 2018)

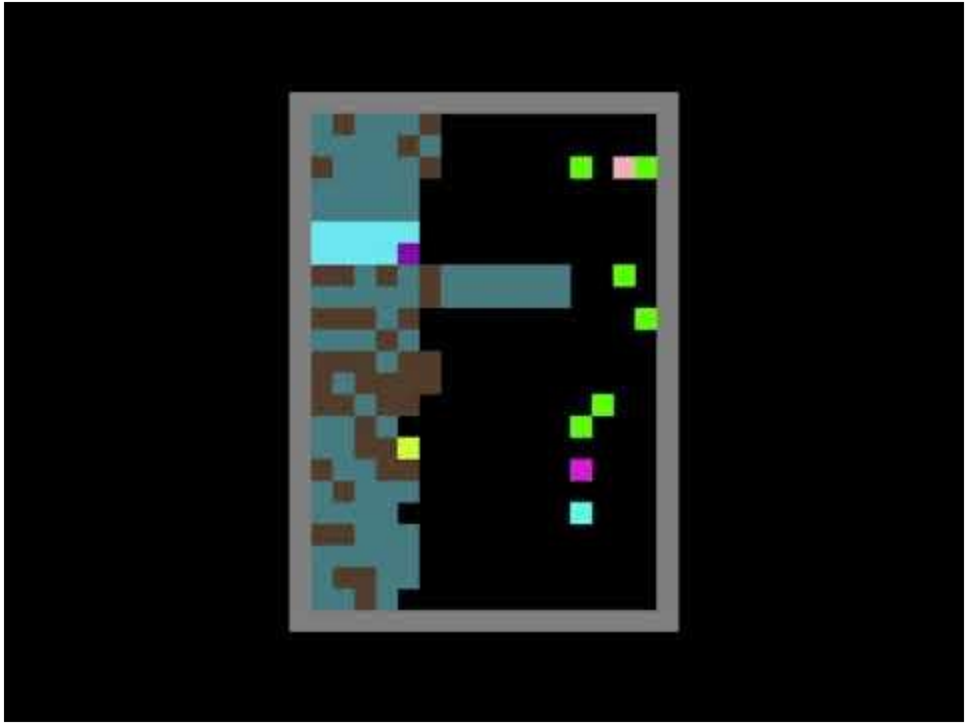


The Public Goods Game

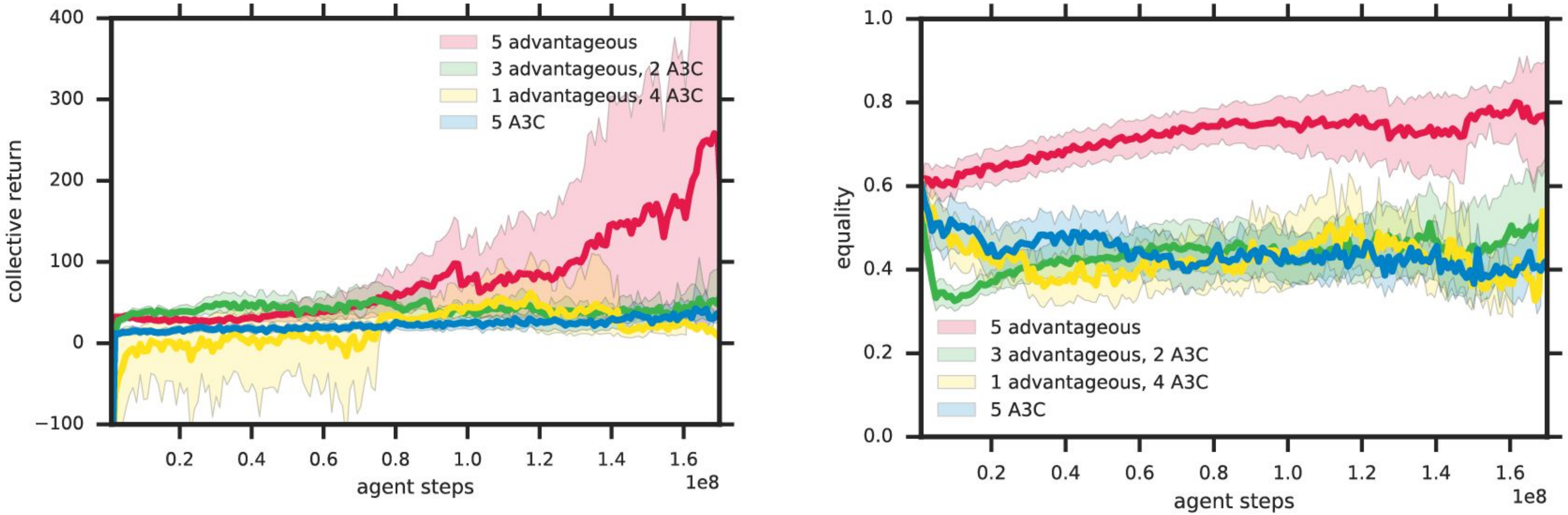


1. Agents move around on a grid world.
2. Agents are only rewarded when they collect an apple.
3. **The apple growth rule is dependent on the waste density. The lower the waste, the higher the apple growth.**
4. **Initially the waste density is so high that no apples can spawn.**
5. Episodes last 1000 steps, after which the game resets to its initial condition.
6. Agents have a “fining beam” with which they can zap one another. Fining costs -1 reward, and causes the fined agent -50 reward.

Guilty agents provide public goods



Guilty agents provide public goods



Take home

- Understanding several MAL paradigms within 1 framework
- EGT as a tool to capture MAL dynamics
- Deep Reinforcement Learning opens new possibilities in many respects, revisiting some of the old results
- Evaluation, Dynamics, and new Algorithmics

Part II. Evaluation & Learning

4. Evaluation
5. Gradients in Games
6. Multi-agent Learning at Scale
7. The Importance of Games

DeepMind

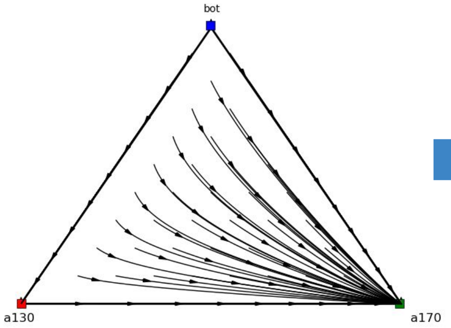
4. Evaluation

DeepMind

How to evaluate agents in a multi-agent context?

Overview

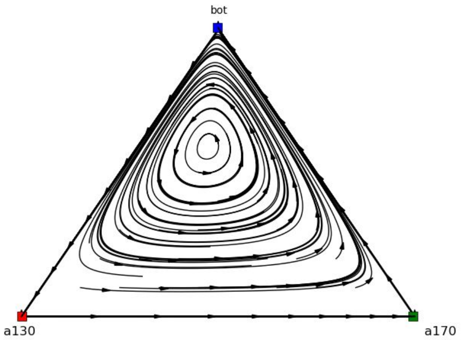
Elo Rating



- Static score
- Cannot capture dynamics
- Cannot deal with intransitivities

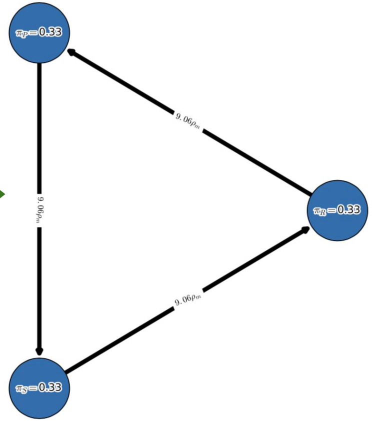
Empirical Game Theory

Continuous-time Evolutionary Dynamics



- Limited to evaluating 3/4 agents
- Stable/unstable Nash equilibria
- Generally intractable to compute & select

Discrete-time Evolutionary Dynamics



- Many-agent interactions
- Stable agents & Markov-Conley Chains
- Unique, tractable to compute & select

Little hope for a **general predictive theory** in terms of **Nash equilibrium**

For a detailed description of an evaluation method based on Nash (Nash Averaging), see:
David Balduzzi, Karl Tuyls, Julien Pérolat, Thore Graepel: Re-evaluating evaluation. NeurIPS 2018: 3272-3283

Elo Evaluation

"The logic of the equation is evident without algebraic demonstration: a player performing above his expectancy gains points, and a player performing below his expectancy loses points." – Arpad E. Elo

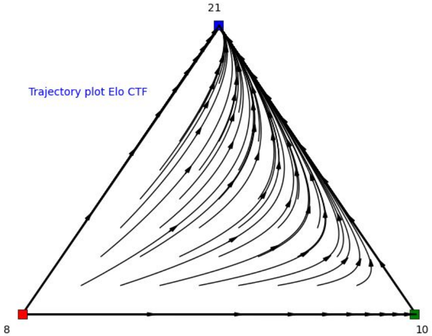
- Update rule: $R^i_{t+1} = R^i_t + K[S^i_{t+1} - E^i_t]$
- Win probability: $p_{ij} = \dfrac{1}{1 + e^{-\alpha(R_i - R_j)}}$
- Chess: $p_{ij} = \frac{1}{1+10^{(R_i - R_j)/400}}$

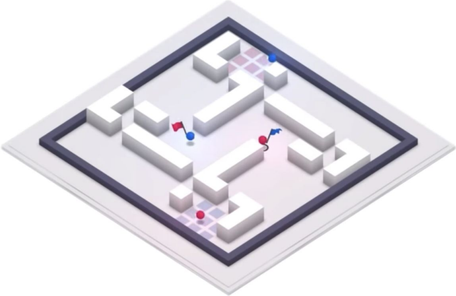
Elo picked 10 as basis and 400 as the denominator because then a difference of 400 points corresponds to a 90% winning probability.

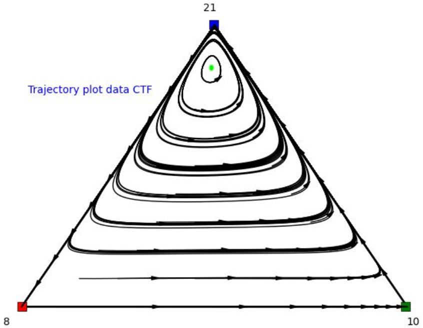
Elo Evaluation

$$\left(\begin{array}{ccc|ccc} 8 & 10 & 21 & U_{i1} & U_{i2} & U_{i3} \\ \hline 2 & 0 & 0 & 0.5 & 0 & 0 \\ 1 & 0 & 1 & 0.014 & 0 & 0.986 \\ 0 & 2 & 0 & 0 & 0.5 & 0 \\ 1 & 1 & 0 & 0.03 & 0.97 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0.5 \\ 0 & 1 & 1 & 0 & 0.3 & 0.7 \end{array}\right)$$

$$\left(\begin{array}{ccc|ccc} 8 & 10 & 21 & U_{i1} & U_{i2} & U_{i3} \\ \hline 2 & 0 & 0 & 0.5 & 0 & 0 \\ 1 & 0 & 1 & 0.54 & 0 & 0.46 \\ 0 & 2 & 0 & 0 & 0.5 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0.5 \\ 0 & 1 & 1 & 0 & 0.45 & 0.55 \end{array}\right)$$







8: 1330
10: 1927
21: 2069

In reality: 8>21, 21>10 and 10>8

Empirical Game Theory Analysis

- A symmetric multi-agent *Meta-Game*:

  (S, A, M, p-type)

- Policies are atomic actions, |A|=n
- *n* does not need to equal *p*
- *S* and *A* can coincide
- E.g. Go dataset: (S, A, M, 2-type)
  - |A|=30 and S=A

Payoff table from data

$$P = \left(\begin{array}{ccc|ccc} N_{i1} & N_{i2} & N_{i3} & U_{i1} & U_{i2} & U_{i3} \\ \hline 6 & 0 & 0 & 0 & 0 & 0 \\ & \dots & & & \dots & \\ 4 & 0 & 2 & -0.5 & 0 & 1 \\ & \dots & & & \dots & \\ 0 & 0 & 6 & 0 & 0 & 0 \end{array}\right)$$

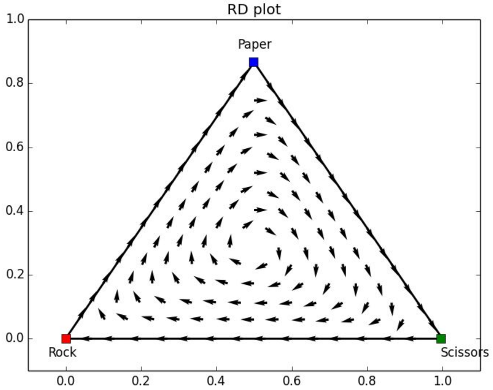
$$\left(\begin{array}{ccc|ccc} N_{i1,j1} & N_{i2,j2} & N_{i3,j3} & U_{i1,j1} & U_{i2,j2} & U_{i3,j3} \\ \hline (1,1) & 0 & 0 & (2,3) & 0 & 0 \\ & \dots & & & \dots & \\ (1,0) & (0,1) & 0 & (0.5,0) & (0,0.5) & 0 \\ (0,1) & (1,0) & 0 & (0,0.4) & (0.3,0) & 0 \\ & \dots & & & \dots & \\ 0 & 0 & (1,1) & 0 & 0 & (3,2) \end{array}\right)$$

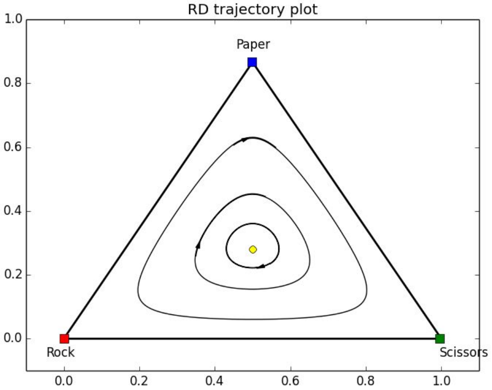
DeepMind

Meta-Game analysis

- Example Rock-Paper-Scissors

	Rock	Paper	Scissors
Rock	(0,0)	(-1,1)	(1,-1)
Paper	(1,-1)	(0,0)	(-1,1)
Scissors	(-1,1)	(1,-1)	(0.0)





- Strategy Space Consumption:
  - Use *sizes of basins* of attraction to rate strategies
  - Combine with *curl* and *sizes* of differential

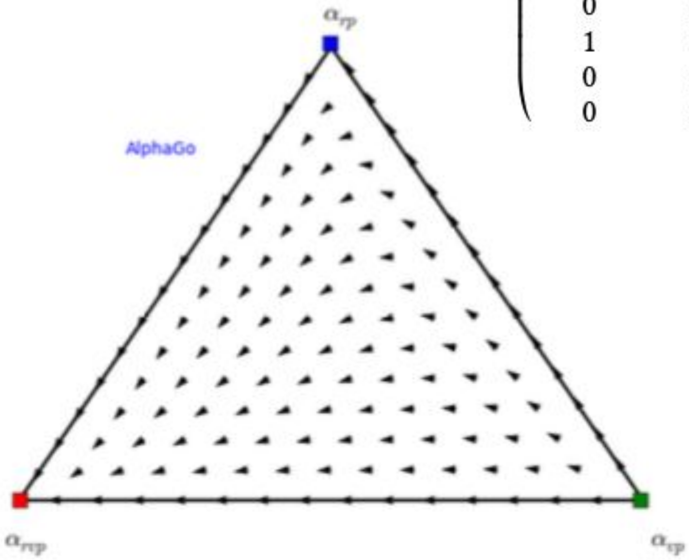
Experiments

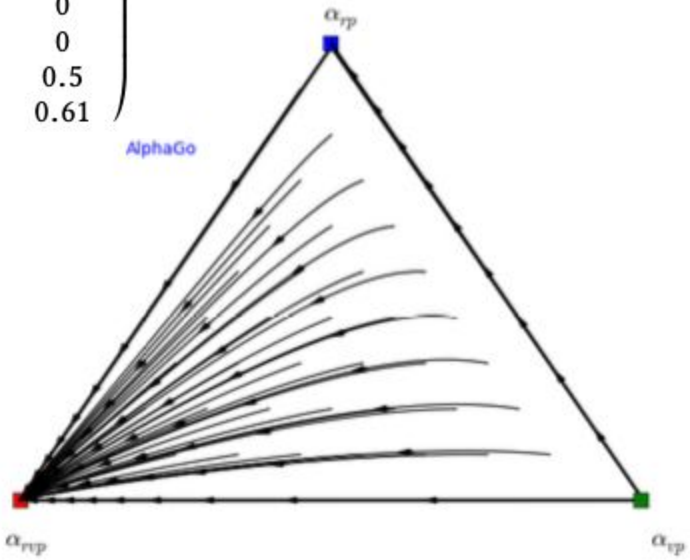
AlphaGo, Colonel Blotto, Leduc Poker

AlphaGo data set

Set of 30 strategies.

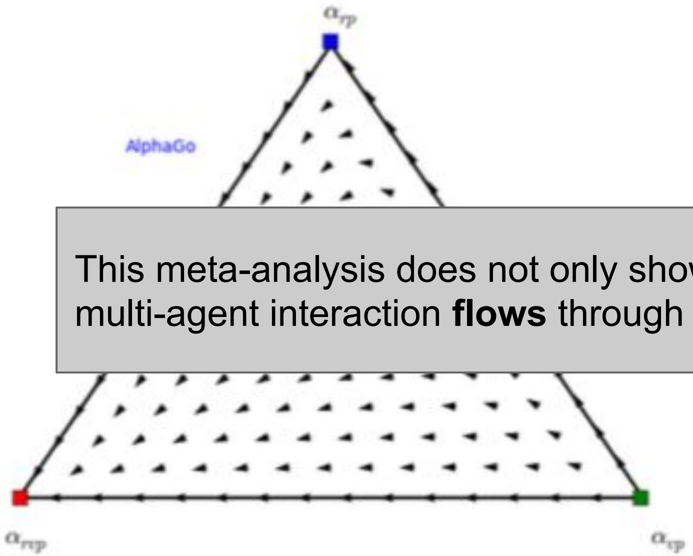
$$\left(\begin{array}{ccc|ccc} \alpha_{rvp} & \alpha_{vp} & \alpha_{rp} & U_{i1} & U_{i2} & U_{i3} \\ \hline 2 & 0 & 0 & 0.5 & 0 & 0 \\ 1 & 0 & 1 & 0.95 & 0 & 0.05 \\ 0 & 2 & 0 & 0 & 0.5 & 0 \\ 1 & 1 & 0 & 0.99 & 0.01 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0.5 \\ 0 & 1 & 1 & 0 & 0.39 & 0.61 \end{array}\right)$$

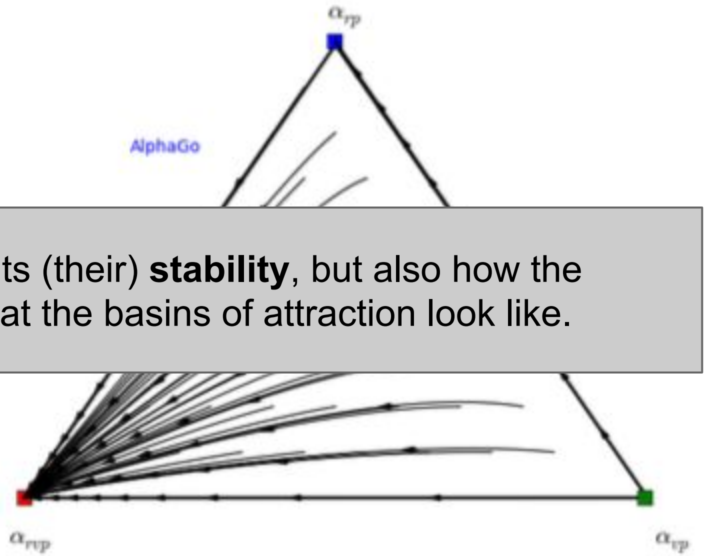




AlphaGo data set

Set of 30 strategies.

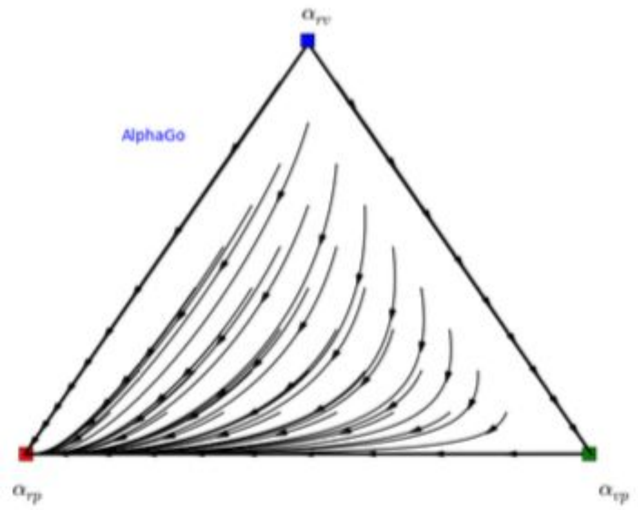


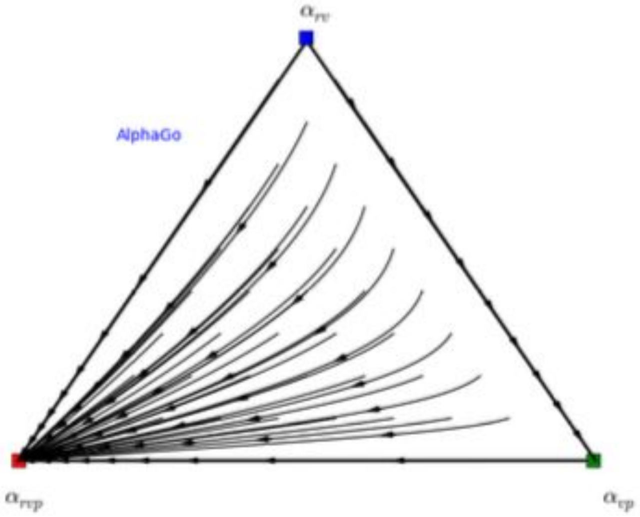


This meta-analysis does not only show **the attractor(s)** and its (their) **stability**, but also how the multi-agent interaction **flows** through strategy space, and what the basins of attraction look like.

AlphaGo data set

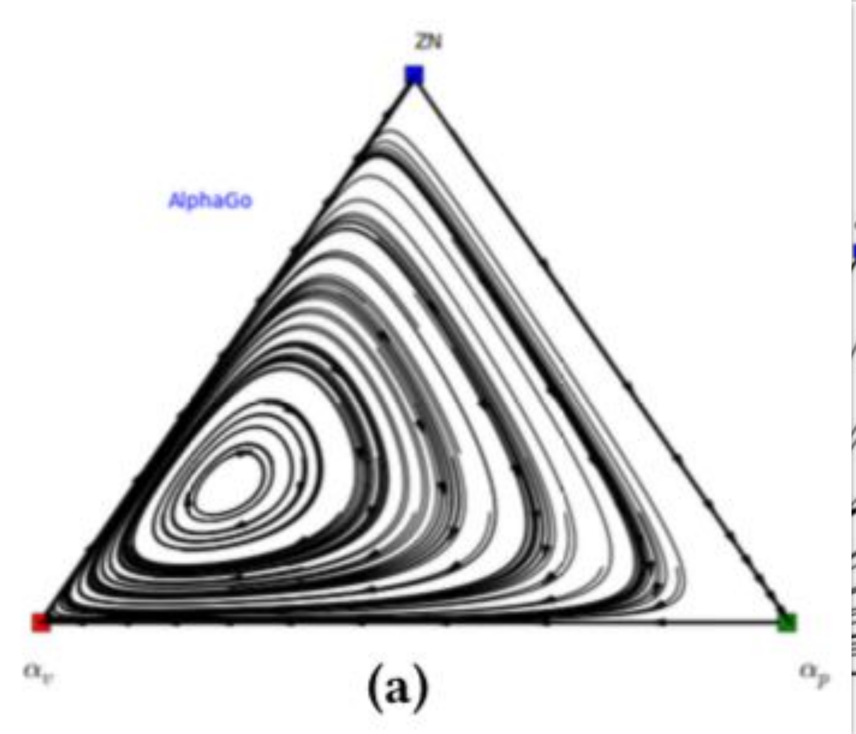
The **curl**, **size** and **direction** of the differential play a role in the determination of the **strength** and **weakness** of a strategy in strategy space, and will be useful for the strategy *space consumption concept*.





AlphaGo data set

Go Leaderboard



Colonel Blotto Game

See https://github.com/deepmind/open_spiel for description / implementation

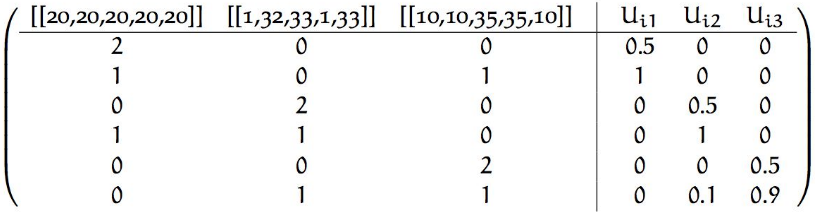
- 2 players, 100 troops each
- Divide over 5 lands

[[20, 20, 20, 20, 20]]

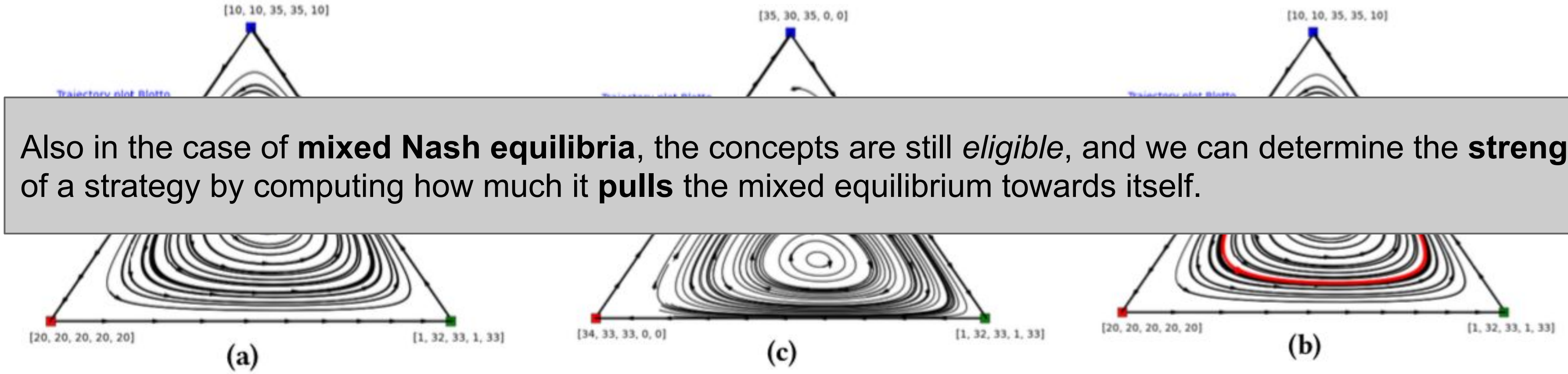
[[33, 1, 32, 1, 33]]

Colonel Blotto

Examined 10 most played strategies



[[20,20,20,20,20]]	[[1,32,33,1,33]]	[[10,10,35,35,10]]	u_{i1}	u_{i2}	u_{i3}
2	0	0	0.5	0	0
1	0	1	1	0	0
0	2	0	0	0.5	0
1	1	0	0	1	0
0	0	2	0	0	0.5
0	1	1	0	0.1	0.9



(a) (c) (b)

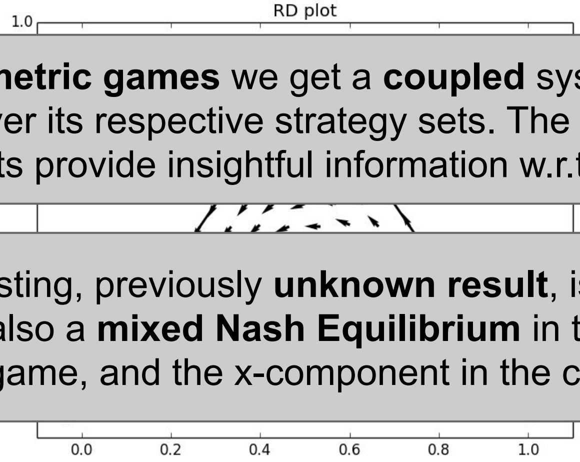
human play

Also in the case of **mixed Nash equilibria**, the concepts are still *eligible*, and we can determine the **strength** of a strategy by computing how much it **pulls** the mixed equilibrium towards itself.

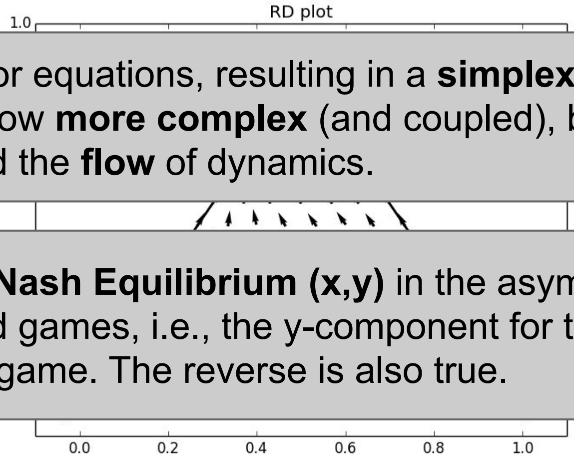
Leduc Poker (PSRO)

PSRO -- asymmetric games - symmetrised replicator dynamics - Leduc

Player 1



Player 2

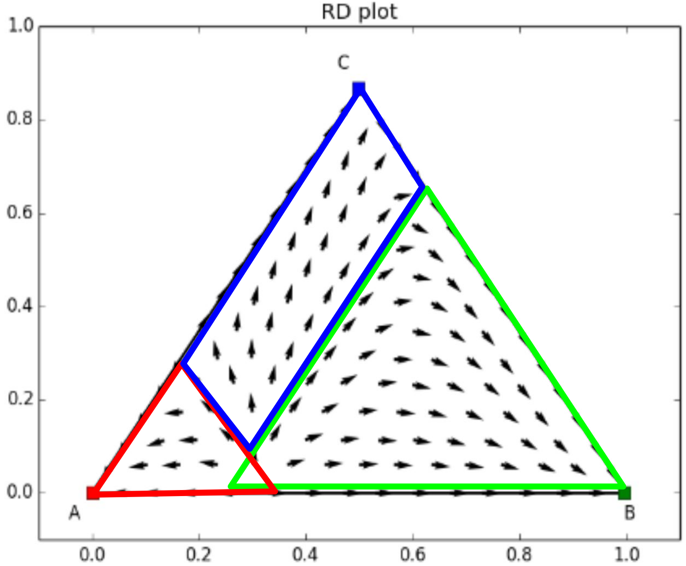


In **asymmetric games** we get a **coupled** system of replicator equations, resulting in a **simplex** for **each player** over its respective strategy sets. The dynamics are now **more complex** (and coupled), but still these plots provide insightful information w.r.t. **equilibria** and the **flow** of dynamics.

An interesting, previously **unknown result**, is that a **mixed Nash Equilibrium (x,y)** in the asymmetric game is also a **mixed Nash Equilibrium** in the symmetrised games, i.e., the y-component for the row player's game, and the x-component in the column player's game. The reverse is also true.

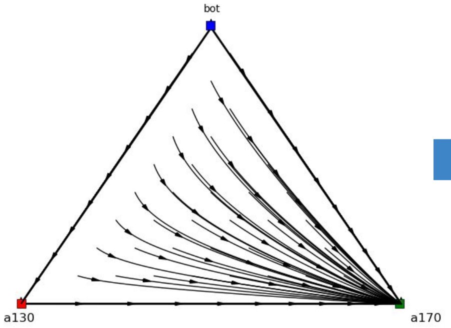
In Conclusion

- EGT/meta-games well suited for both ***symmetric*** and ***asymmetric games***
    - Poker, Go, Auctions, Robotics
- Provide bounds that tell you how reliable the estimated game is
- Limited to 3/4 strategies



Multi-Agent Evaluation

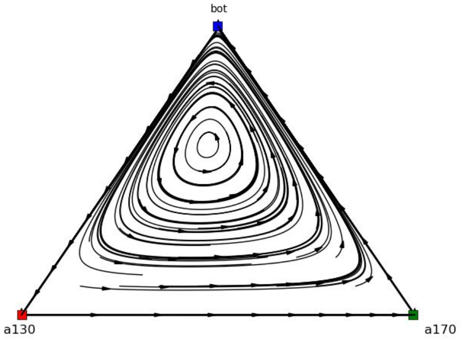
Elo Rating



- Static score
- Cannot capture dynamics
- Cannot deal with intransitivities

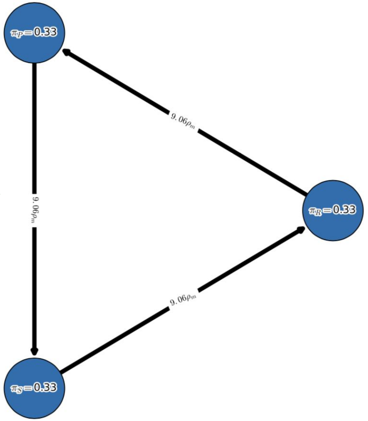
Empirical Game Theory

Continuous-time Evolutionary Dynamics



- Limited to evaluating 3/4 agents
- Stable/unstable Nash equilibria
- Generally intractable to compute & select

Discrete-time Evolutionary Dynamics



- Many-agent interactions
- Stable agents & Markov-Conley Chains
- Unique, tractable to compute & select

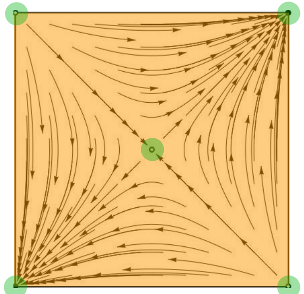
Little hope for a **general predictive theory** in terms of **Nash equilibrium**

DeepMind

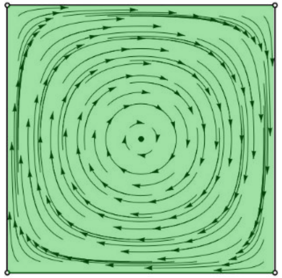
Dynamical Systems Foundations

- Analogous to Nash using Kakutani's fixed point theorem as a basis for his solution concept, we use Conley's Fundamental Theorem of Dynamical Systems (Conley, 1978):

> *"Any flow on a compact metric space decomposes into a* ***gradient-like part*** *that leads to a* ***recurrent part****."*



Coordination game



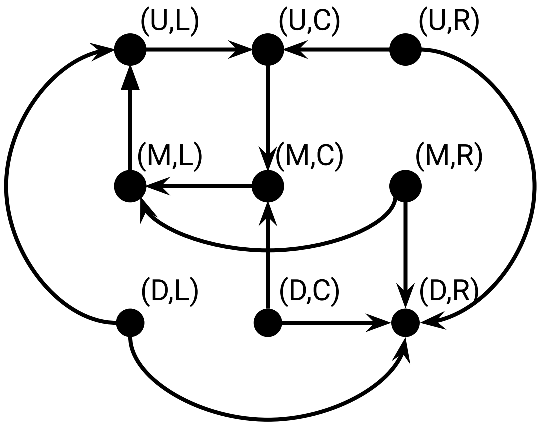
Matching pennies game

- **Markov-Conley Chains (MCCs)** are the discrete analogs of the recurrent set above
  - Capture irreducible long-term dynamical interactions between agents
  - Correspond to the **unique stationary distribution** of an underlying discrete-time evolutionary process
  - Pinpoint diverse set of agents that are **evolutionarily stable** (cannot be mutated or invaded)

A Dynamical Solution Concept

- Caveat: difficult to study these recurrent sets theoretically
    - We need a **meaningful approximation** that can be tractably analyzed
- **Response graph:** directed graph where nodes correspond to pure strategy profiles, and directed edges if the deviating player's new strategy is a better-response

		Player 2		
		L	C	R
Player 1	U	(2,0)	(0,2)	(0,0)
	M	(0,2)	(2,0)	(0,0)
	D	(0,0)	(0,0)	(1,1)

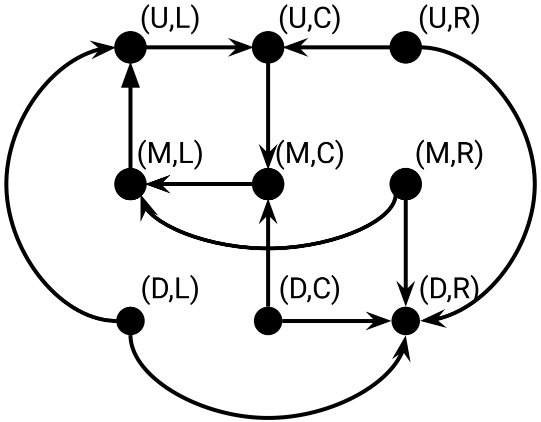


A Dynamical Solution Concept

- Caveat: difficult to study these recurrent sets theoretically
    - We need a **meaningful approximation** that can be tractably analyzed
- **Response graph:** directed graph where nodes correspond to pure strategy profiles, and directed edges if the deviating player's new strategy is a better-response
- **Markov-Conley chains (MCCs):**
    - Markov chains over the sink strongly connected components of response graph
    - Our dynamical solution concept!

Player 2

		L	C	R
	U	(2,0)	(0,2)	(0,0)
Player 1	M	(0,2)	(2,0)	(0,0)
	D	(0,0)	(0,0)	(1,1)



Quiz Question

- **Markov-Conley chains (MCCs):**
  - Markov chains over the **sink** strongly connected components of response graph
  - ***Hint:*** *a directed graph is strongly connected if there is a path between all pairs of its vertices.*

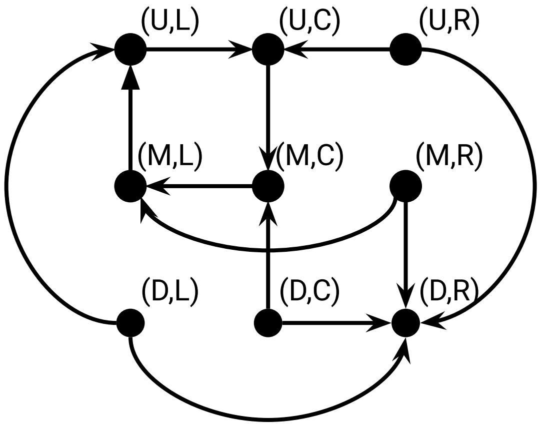
How many MCCs exist in the below response graph?

A. 0
B. 1
C. 2
D. 9

Player 2

Player 1

	L	C	R
U	(2,0)	(0,2)	(0,0)
M	(0,2)	(2,0)	(0,0)
D	(0,0)	(0,0)	(1,1)



Quiz Question

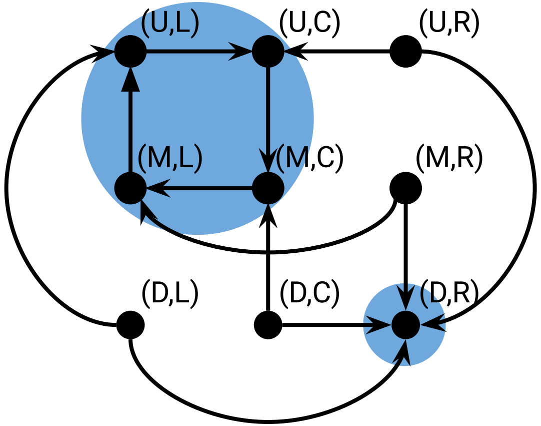
- **Markov-Conley chains (MCCs):**
    - ○ Markov chains over the sink strongly connected components of response graph

How many MCCs exist in the below response graph?

A. 0
B. 1
C. 2
D. 9

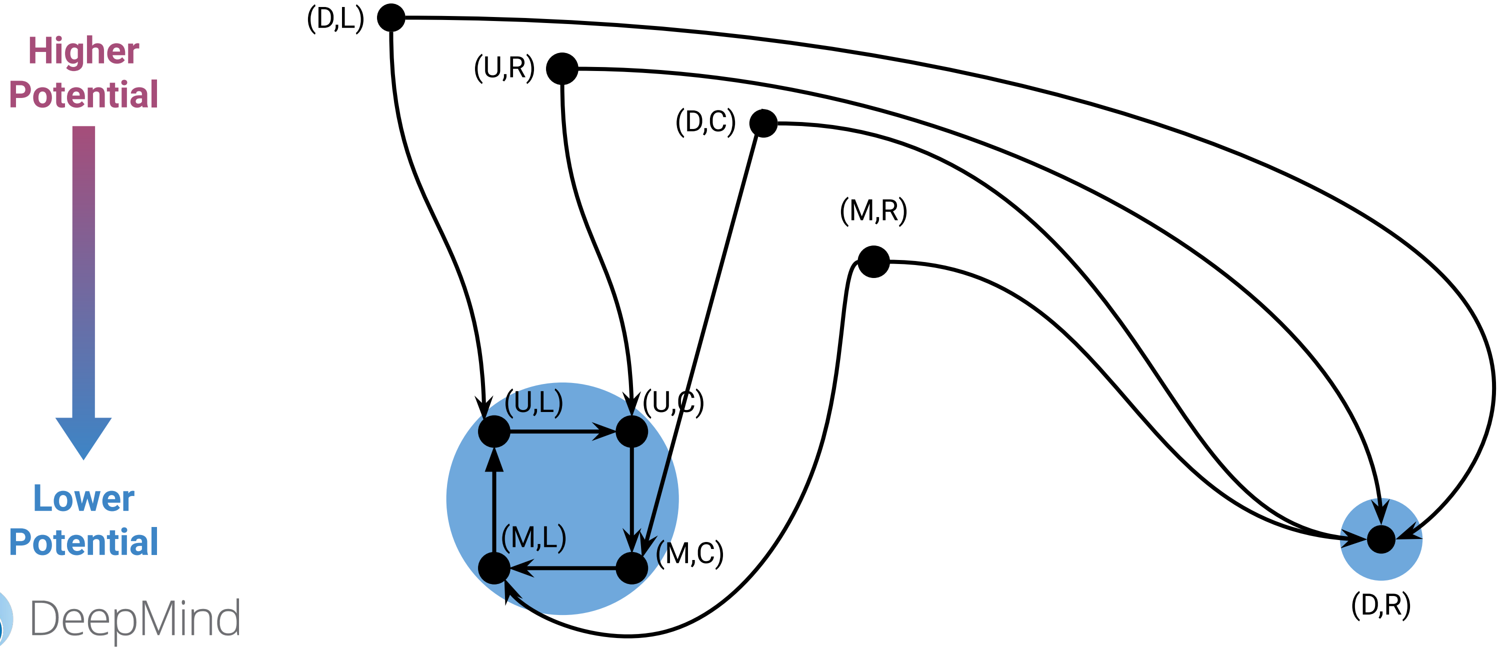
Player 2

		L	C	R
	U	(2,0)	(0,2)	(0,0)
Player 1	M	(0,2)	(2,0)	(0,0)
	D	(0,0)	(0,0)	(1,1)



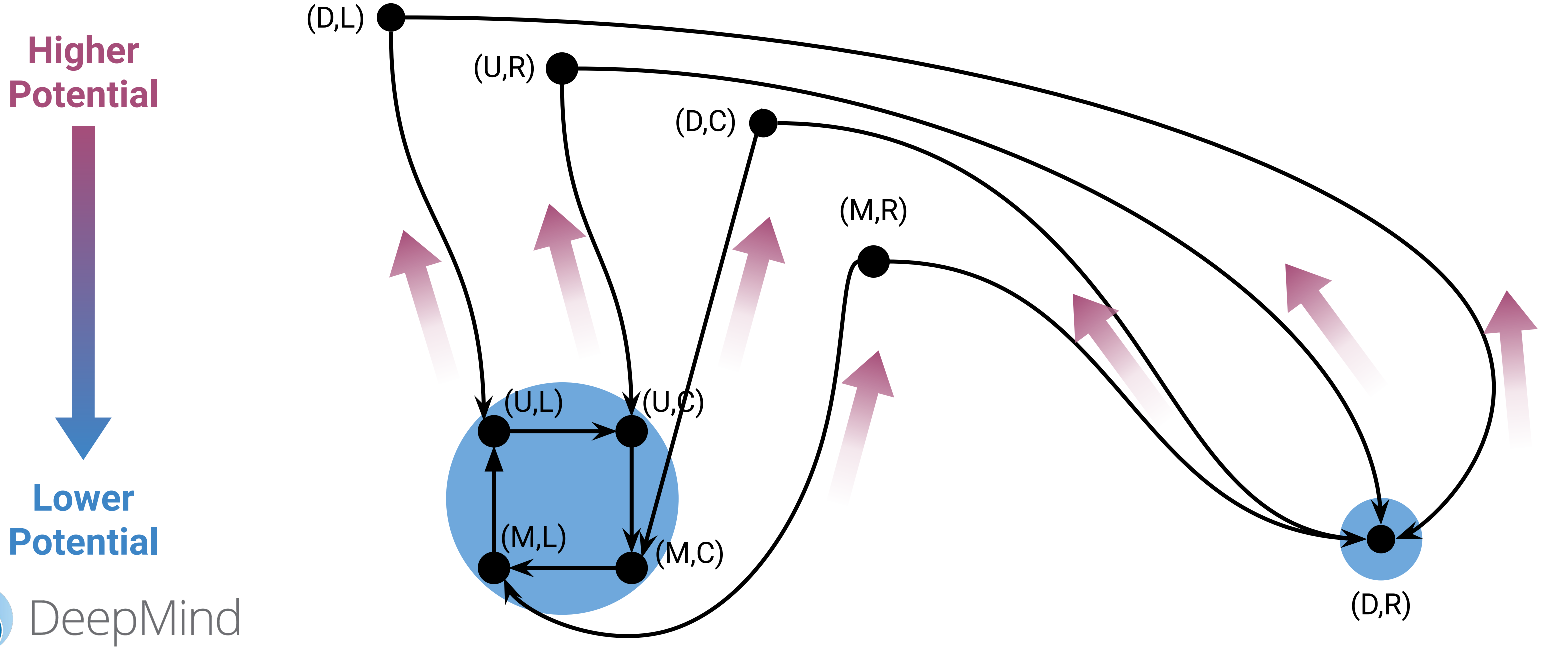
Resolving Equilibrium Selection

- MCCs are computationally attractive, but face equilibrium selection issues akin to Nash



Resolving Equilibrium Selection

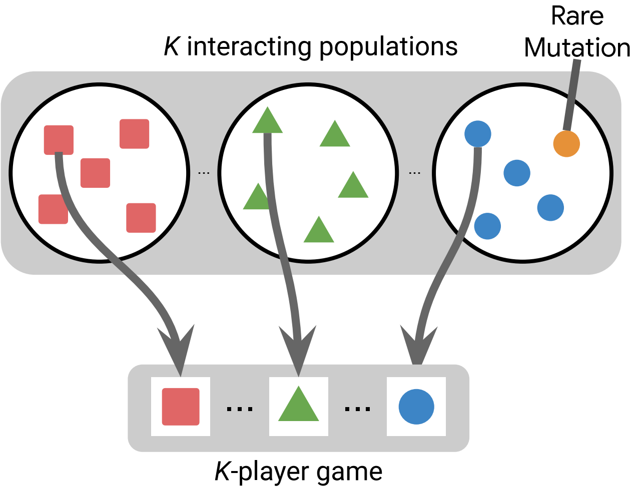
- MCCs are computationally attractive, but face equilibrium selection issues akin to Nash
- **Solution:** perturb the response graph such that a random walk can **climb upward** on the potential hills and hop between **MCCs** (sinks) with a very small probability
    - Irreducible Markov chain → unique stationary distribution → unique MCC rankings



DeepMind

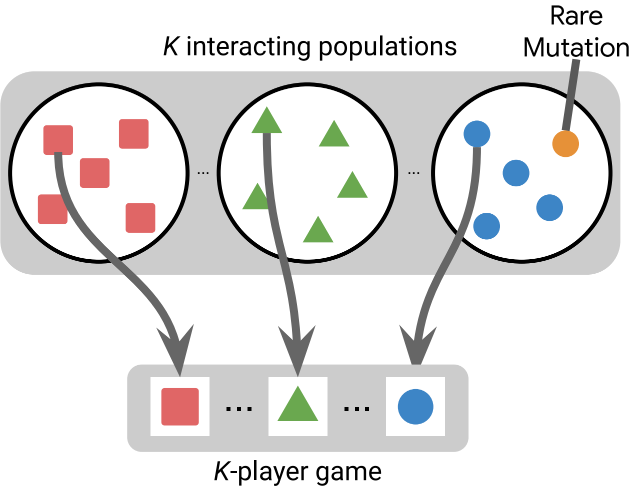
Linking MCCs and Evolution

- Remarkably, our perturbed model is equivalent to **a discrete-time evolutionary process**
  - Well-studied in the literature for pairwise/symmetric games
  - Generalized in our work to *K*-player asymmetric games
- **Basic idea:** model a selection-mutation process over a set of interacting populations



Linking MCCs and Evolution

- Remarkably, our perturbed model is equivalent to **a discrete-time evolutionary process**
  - Well-studied in the literature for pairwise/symmetric games
  - Generalized in our work to *K*-player asymmetric games
- **Basic idea:** model a selection-mutation process over a set of interacting populations
  - Strong agents (i.e., those resistant to mutants) propagate via a selection function:



$$\underbrace{\mathbb{P}(\tau \to \sigma, s^{-k})}_{\text{Probability of competing agent } \boldsymbol{\sigma} \text{ taking over}} = \left(1 + e^{\alpha \underbrace{(f^k(\tau, s^{-k}) - f^k(\sigma, s^{-k}))}_{\text{Fitness of resident agent } \boldsymbol{\tau} \text{ vs. competing agent } \boldsymbol{\sigma}}}\right)^{-1}$$

- Small α
- Weak selection

Ranking-intensity value α

- **Large α**
- **Strong selection**
- **MCC solution concept**
- **α-Rank**

α-Rank Algorithm

Construct the meta-game payoff tables from multi-agent simulations

Define a Markov chain where states are the agents being evaluated

Compute transition matrix $\boldsymbol{C}$ according to an evolutionary process with selection-intensity parameter α

Compute the **unique** stationary distribution π of $\boldsymbol{C}$

Agent rankings/scores correspond to the ordered masses of π

Ranking guaranteed to exist and is unique

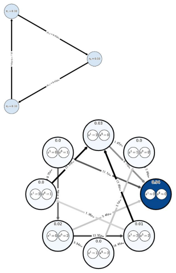
Handles cycles/intransitivities

Scalable and applies to general-sum, symmetric/asymmetric, many-player games

DeepMind

Unified View of Multi-agent Evaluation by Evolution

Macro-model:
Discrete-time Dynamics



Analytical toolkit:

- Markov chain
- Stationary distribution
- Fixation probabilities

Applicability:

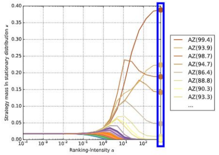
- K-wise interactions
- Symmetric and asymmetric games

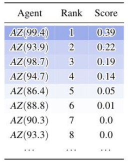
Unifying ranking model:
Markov Conley Chains & α-Rank

α-Rank

Selection-intensity parameter α

Agent Ranking





Agent	Rank	Score
$AZ(99.4)$	1	0.39
$AZ(93.9)$	2	0.22
$AZ(98.7)$	3	0.19
$AZ(94.7)$	4	0.14
$AZ(86.4)$	5	0.05
$AZ(88.8)$	6	0.01
$AZ(90.3)$	7	0.0
$AZ(93.3)$	8	0.0
…	…	…

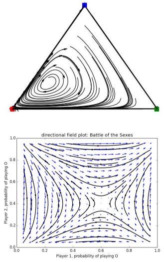
Foundations:

- Conley's Fundamental Theorem
- Chain recurrent sets and components

Advantages:

- Captures dynamic behavior
- More tractable to compute than Nash
- Filters out transient agents
- Involves only a single hyperparameter, α

Micro-model:
Continuous-time Dynamics



Analytical toolkit:

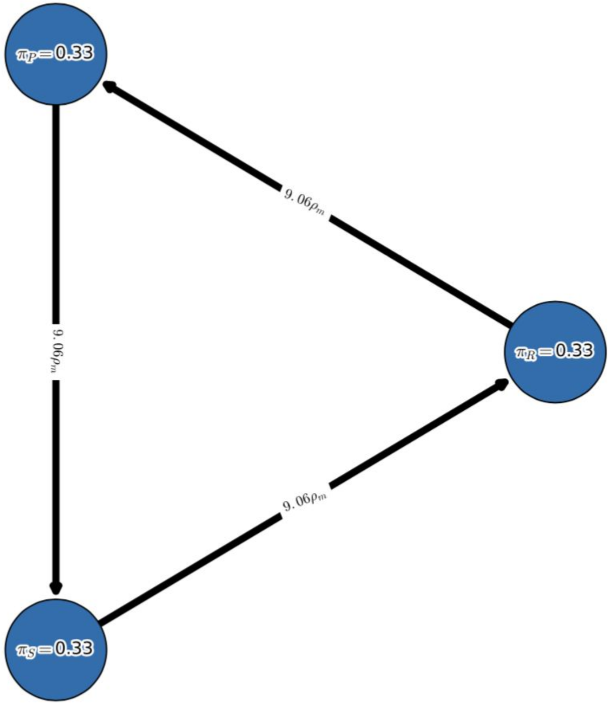
- Flow diagrams sub-graph
- Attractors, equilibria

Applicability:

- 3 to 4 agents max
- Symmetric games and 2-population asymmetric games

Demonstrations

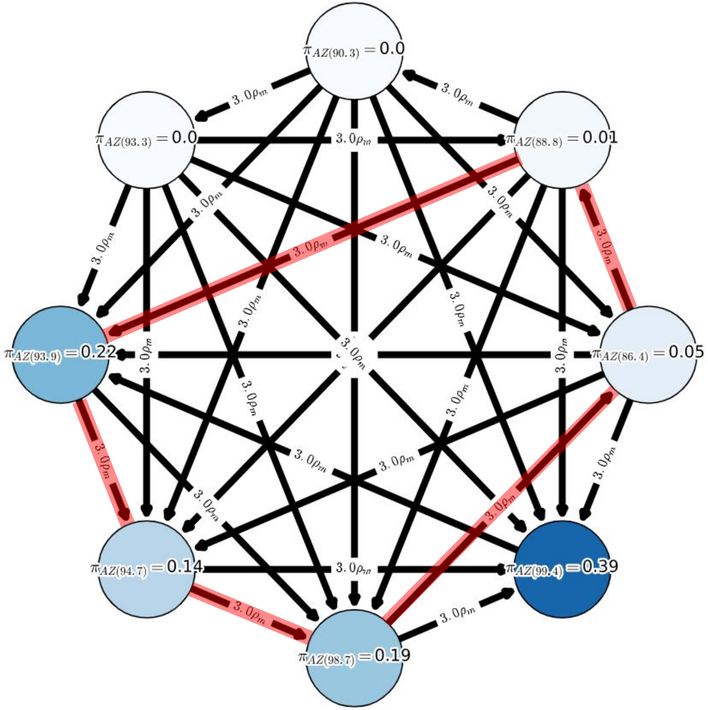
- Rock-Paper-Scissors (2-player, symmetric, 3 agents)



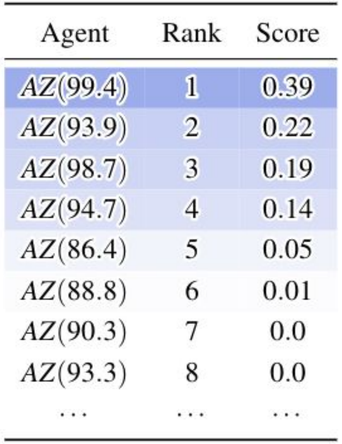
Agent	Rank	Score
R	1	0.33
P	1	0.33
S	1	0.33

Demonstrations

- AlphaZero Chess (2-player game, 56 agent snapshots taken during training)



Top-8 agents shown

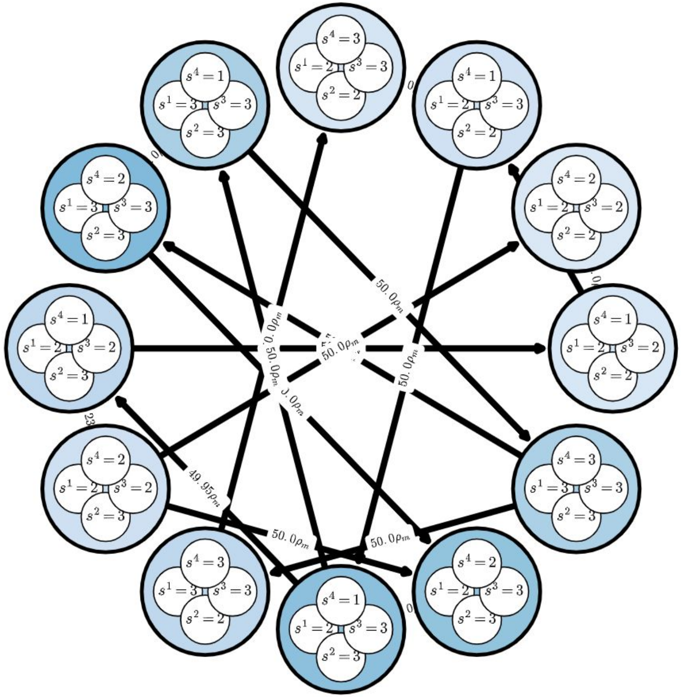


Agent	Rank	Score
$AZ(99.4)$	1	0.39
$AZ(93.9)$	2	0.22
$AZ(98.7)$	3	0.19
$AZ(94.7)$	4	0.14
$AZ(86.4)$	5	0.05
$AZ(88.8)$	6	0.01
$AZ(90.3)$	7	0.0
$AZ(93.3)$	8	0.0
…	…	…

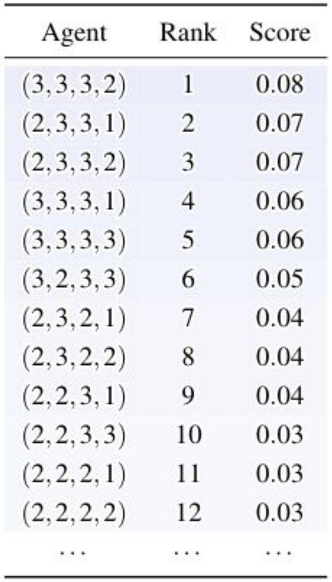
Top-8 agents
(training percent complete in parentheses)

Demonstrations

- Kuhn Poker (4-player, asymmetric, 256 agent profiles)



Top-12 profiles shown



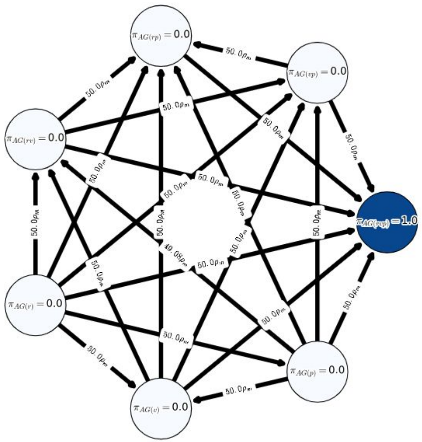
Agent	Rank	Score
$(3,3,3,2)$	1	0.08
$(2,3,3,1)$	2	0.07
$(2,3,3,2)$	3	0.07
$(3,3,3,1)$	4	0.06
$(3,3,3,3)$	5	0.06
$(3,2,3,3)$	6	0.05
$(2,3,2,1)$	7	0.04
$(2,3,2,2)$	8	0.04
$(2,2,3,1)$	9	0.04
$(2,2,3,3)$	10	0.03
$(2,2,2,1)$	11	0.03
$(2,2,2,2)$	12	0.03
...	...	...

Top-12 profiles shown

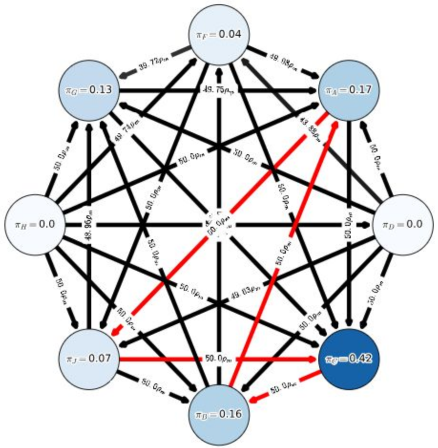
DeepMind

Summary

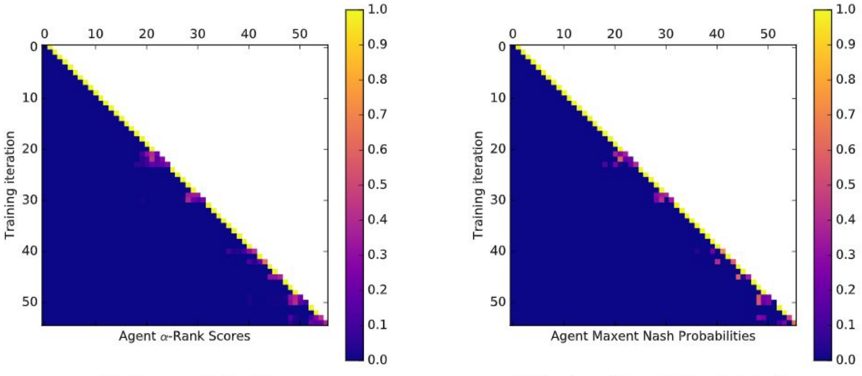
- α-Rank: principled multi-agent evaluation method
  - To appear in Nature's Scientific Reports journal, check out arXiv draft for more:



AlphaGo results



MuJoCo Soccer results



(a) α-Score vs. Training Time.

(b) Maximum Entropy Nash vs. Training Time.

α-Rank vs. Nash in two-player games

5. Gradients in Games

DeepMind

Where are we?

“If you have a large big dataset, and you train a very big neural network, then success is guaranteed!”
-- **Ilya Sutskever (NIPS 2014)**

Where are we?

The central dogma of deep (supervised) learning:

- compose **differentiable modules** into a neural net;
- convert data into a differentiable **objective function**;
- add **backprop**; and
- **press go**.

"If you have a large big dataset, and you train a very big neural network, then success is guaranteed!"
-- **Ilya Sutskever (NIPS 2014)**

How’d we get here?

Lots of “small” things:

- **differentiable modules:**
  - CNNs, LSTMs, ResNets, ReLUs, clever initializations, BatchNorm, …
- **objective functions:**
  - datasets ⟶ losses
- **backprop:**
  - momentum, Adam, RMSProp, learning rates, hyper-parameters
- **press go:**
  - libraries (TensorFlow, PyTorch, …) and GPUs take care of almost everything

Why here?

One big thing: **the loss landscape**

Everything depends on gradient descent finding (good) local minima in the loss landscape

Image Credit - Vectors Market

Is this it?

Trouble in paradise

- Modules aren't actually modules:
  - Trained NNs are nowhere near plug-and-play
  - NNs are invariably (re)trained from scratch
  - Not data-efficient
- Rampant overfitting
  - transfer learning is extremely difficult
  - adversarial examples

End-to-end learning doesn't scale

What’s next?

William Gibson: *“The future is already here — it's just not very evenly distributed.”*

What's next?

William Gibson: *"The future is already here — it's just not very evenly distributed."*

- **Generative Adversarial Networks (Goodfellow *et al*, NIPS 2014)**
- **Cycle-consistent adversarial nets (Zhu *et al*, ICCV 2017)**
- Synthetic gradients (Jaderberberg *et al*, ICML 2017)
- Deep learning and neurosci (Marblestone *et al*, 2016)
- Intrinsic curiosity (Pathak *et al*, ICML 2017)

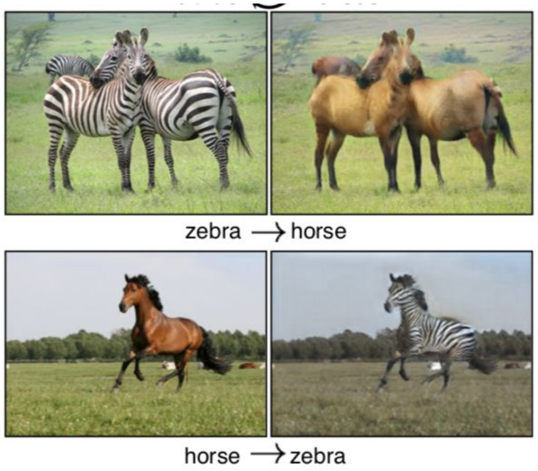


Image Credit - Zhu *et al*

Generative adversarial networks

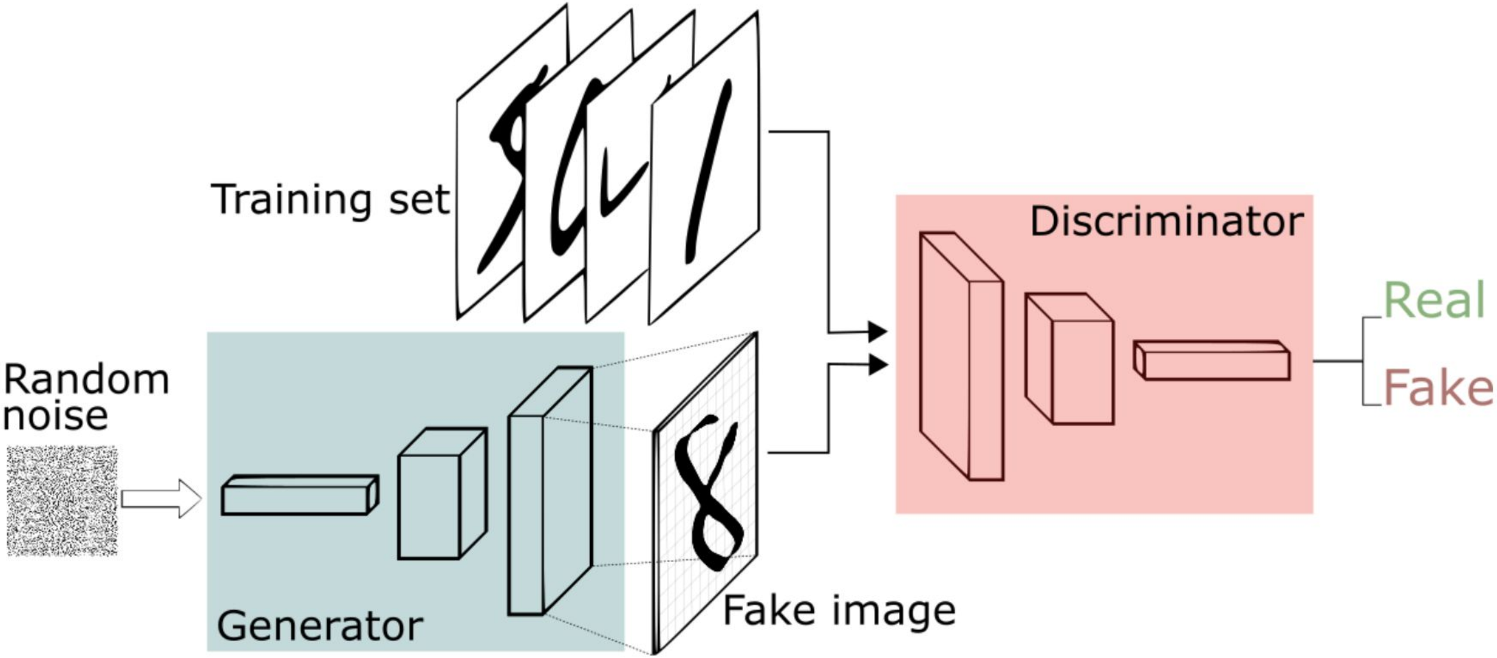


Image Credit - deeplearning4j.org

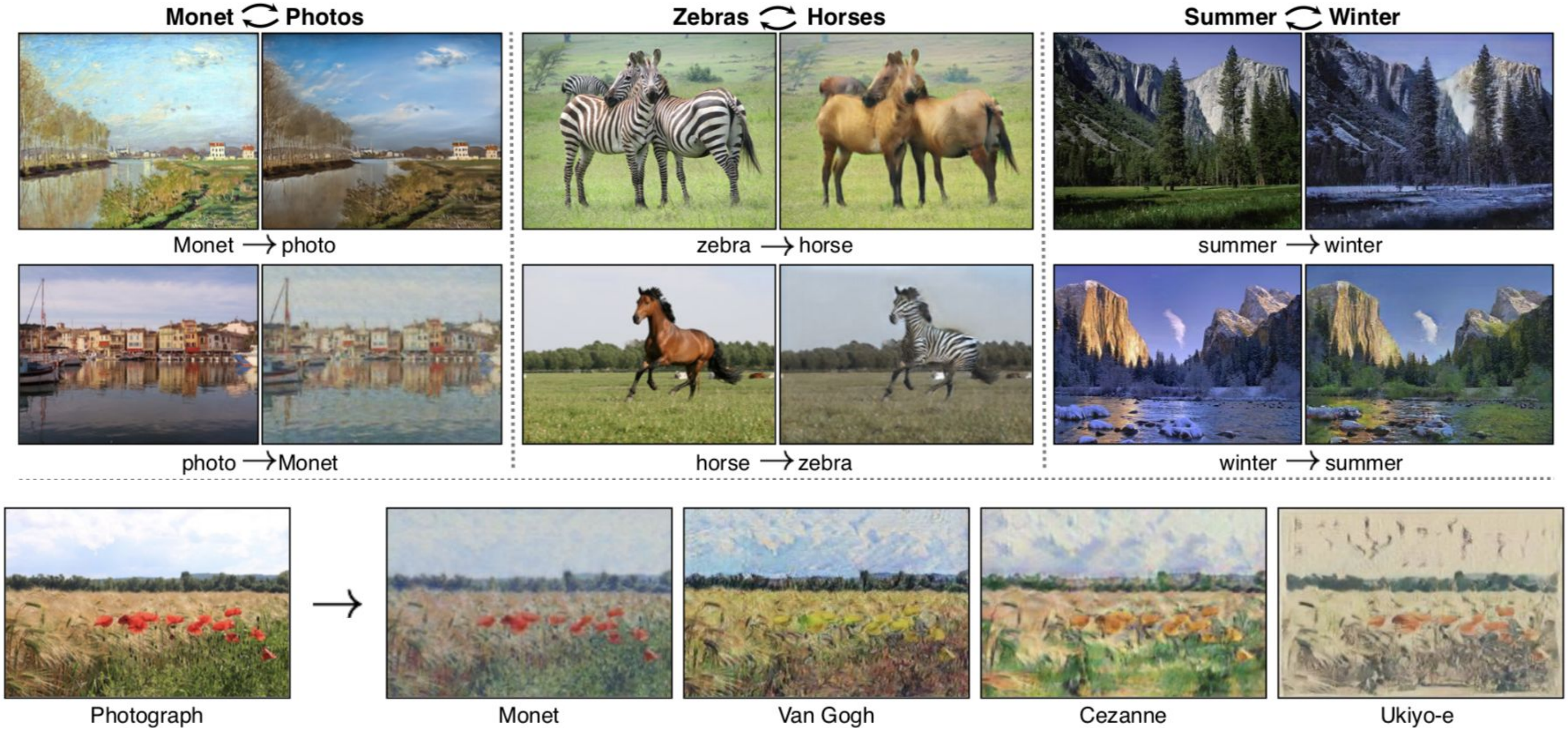
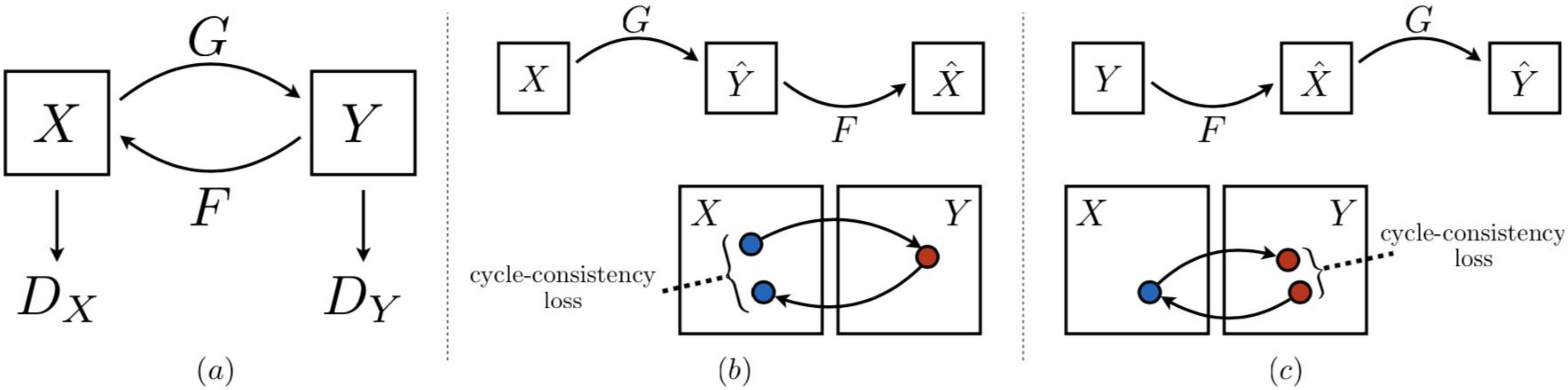


Image Credit - Zhu *et al*

Cycle-GANs



cycle-consistency = { learning a commutative diagram }

Image Credit - Zhu *et al*

What’s next?

William Gibson: *“The future is already here — it's just not very evenly distributed.”*

- Generative Adversarial Networks (Goodfellow *et al*, NIPS 2014)
- Cycle-consistent adversarial nets (Zhu *et al*, ICCV 2017)
- Synthetic gradients (Jaderberberg *et al*, ICML 2017)
- Deep learning and neurosci (Marblestone *et al*, 2016)
- Intrinsic curiosity (Pathak *et al*, ICML 2017)

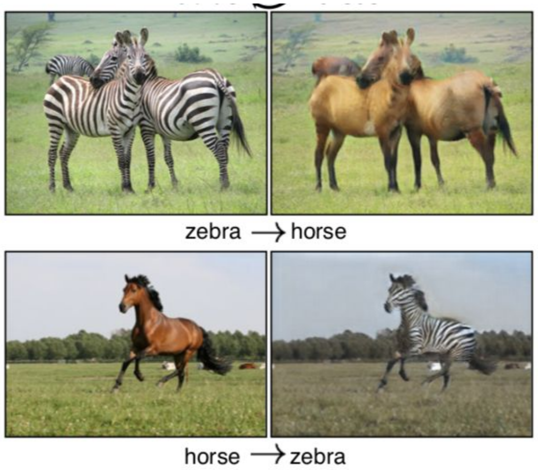


Image Credit - Zhu *et al*

Themes:

- Interacting losses and datasets
- It’s hard work and *ad hoc*

A brief history of ML

- Learning:
  - **Why?** Don’t want to hand-code behaviors
  - **Catch:** Weaker guarantees

A brief history of ML

- Learning:
  - **Why?** Don't want to hand-code behaviors
  - **Catch:** Weaker guarantees
- Learning representations:
  - **Why?** Don't want to hand-design features
  - **Catch:** Non-convex optimization

A brief history of ML

- Learning:
    - **Why?** Don't want to hand-code behaviors
    - **Catch:** Weaker guarantees
- Learning representations:
    - **Why?** Don't want to hand-design features
    - **Catch:** Non-convex optimization
- Learning losses:
    - **Why?** Don't want to hand-label data
    - **Catch:** ...

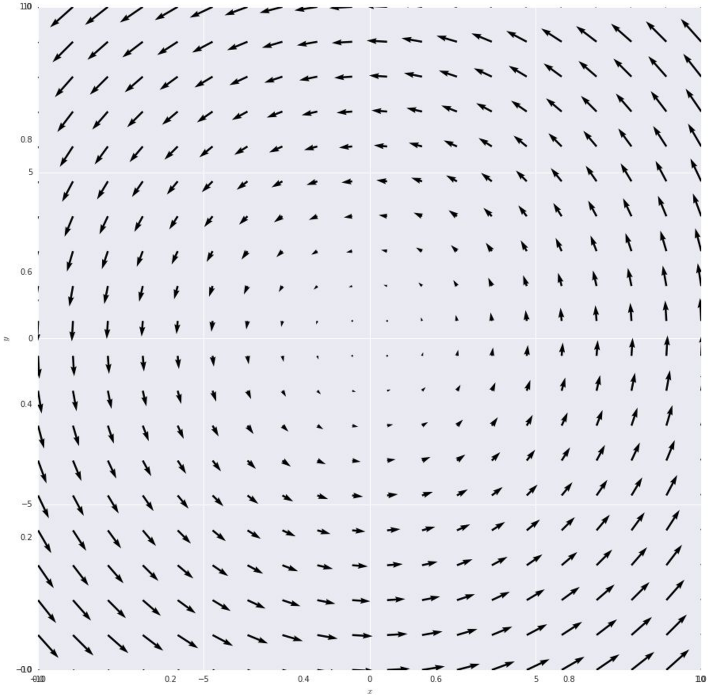
What’s the problem?

Minimal example

$$\ell_1(x, y) = xy \qquad \ell_2(x, y) = -xy$$

$$\xi = \left(\frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y} \right) = (y, -x)$$

- Dynamics cycle around origin



But there's no landscape

$$\ell_1(x, y) = xy \qquad \ell_2(x, y) = -xy$$

$$\xi = \left(\frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y} \right) = (y, -x)$$

- Dynamics cycle around origin
- There's **no** consistent "down direction"

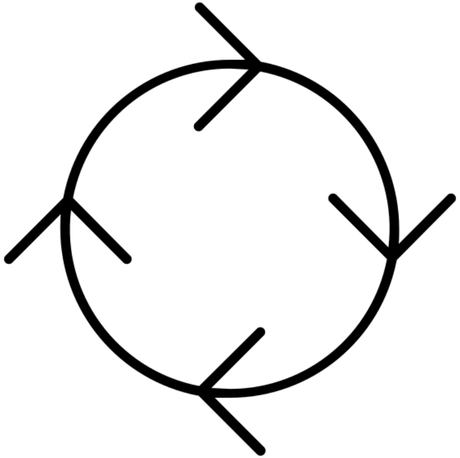
Image Credit - Heritage Auctions, MC Escher

DeepMind

But there's no landscape

$$\ell_1(x,y) = xy \qquad \ell_2(x,y) = -xy$$

$$\xi = \left(\frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y}\right) = (y, -x)$$



- Dynamics cycle around origin
- There's **no** consistent "down direction"

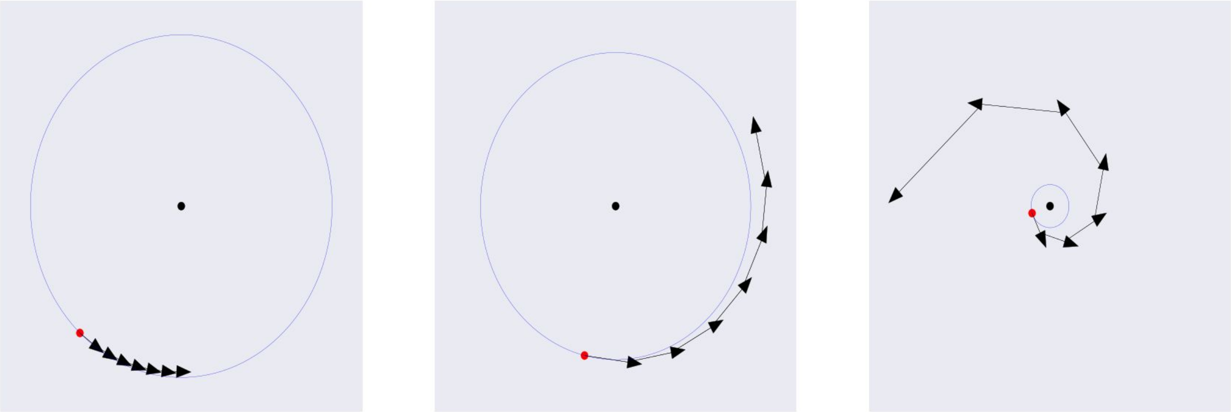
Technical problem:

- Vector field isn't a gradient vector field

Image Credit - prakruti

Three problems

1. Gradient descent isn’t guaranteed to converge (to anything, at all)
2. Even if it does, it can be very unstable and slow
3. Actually, can’t even measure progress



Learning rate 0.01 0.032 0.1

Which geometry?

Mathematicians and physicists have been studying geometry for centuries.

There must be something on-the-shelf that we can use.

Div, grad, and curl

Helmholtz decomposition:

Any vector field in R^3 decomposes as a sum of a **gradient vector field** (a curl-free or **irrotational** component) and a **divergence-free** component:

$$\xi = \nabla\phi + \mathrm{curl}(\rho)$$

landscape-ish

Escher-ish
(measures infinitesimal
tendency to rotate)

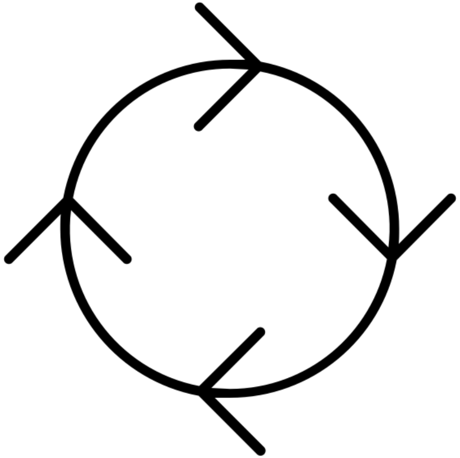
Minimal example

$$\xi = \left(\frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y} \right) = (y, -x, 0)$$

Minimal example

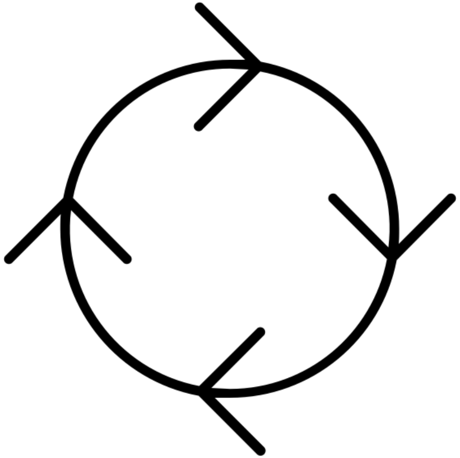
$$\xi = \left(\frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y} \right) = (y, -x, 0)$$

- Vector field is divergence-free
  - There’s no function that is being optimized



Minimal example

$$\xi = \left(\frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y}\right) = (y, -x, 0)$$



- Vector field is divergence-free
  - There's no function that is being optimized

$$\xi = \mathrm{curl}(-xz, -yz, 0)$$

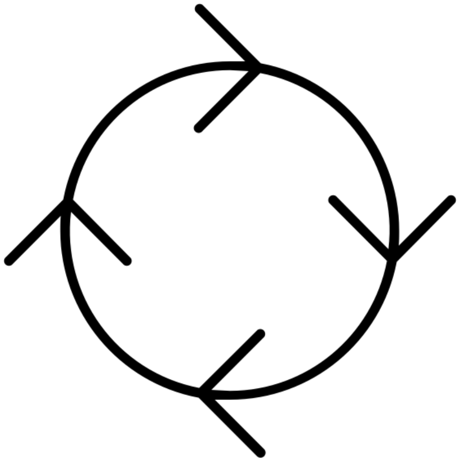
- ???

Minimal example

Dead end

$$\xi = \left(\frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y}\right) = (y, -x, 0)$$

- Vector field is divergence free
  - There's no function that is being optimized



$$\xi = \mathrm{curl}(-xz, -yz, 0)$$

- ???

Which geometry?

Mathematicians and physicists have been studying geometry for centuries.
There must be something on-the-shelf that we can use.

Actually, those **cycles** look like **planetary orbits** …

Classical mechanics (in one slide)

Canonical coordinates: position **q** and momentum **p = mv**

Hamiltonian: total (potential + kinetic) energy $\mathcal{H}(\mathbf{q}, \mathbf{p})$

Dynamics: $\frac{dq_i}{dt} = \frac{\partial \mathcal{H}}{\partial p_i}$ $\frac{dp_i}{dt} = -\frac{\partial \mathcal{H}}{\partial q_i}$ $\xi = (\nabla_{\mathbf{p}} \mathcal{H}, -\nabla_{\mathbf{q}} \mathcal{H})$

Conservation of energy: $\langle \xi, \nabla \mathcal{H} \rangle = 0$

The dynamics lives on the level sets of the Hamiltonian.

Game mechanics?

Position, momentum, and conservation of energy don’t feature in good old fashioned game theory.

Eg: zero-sum bimatrix games

$$\ell_1(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\intercal \mathbf{A} \mathbf{y} \qquad \ell_2(\mathbf{x}, \mathbf{y}) = -\mathbf{x}^\intercal \mathbf{A} \mathbf{y}$$

Singular value decomposition:

$$\mathbf{A} = \mathbf{U}^\intercal \mathbf{D} \mathbf{V}$$

Change of coordinates:

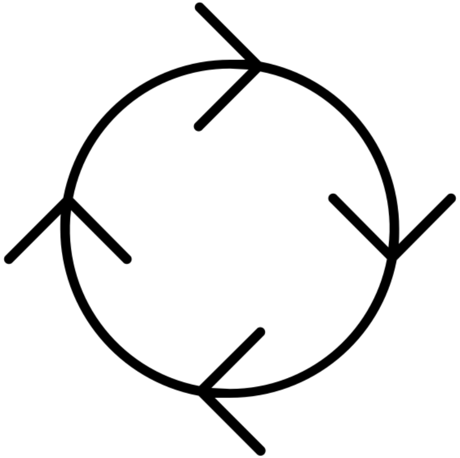
$$\mathbf{u} = \mathbf{D}^{\frac{1}{2}} \mathbf{U} \mathbf{x} \qquad \mathbf{v} = \mathbf{D}^{\frac{1}{2}} \mathbf{V} \mathbf{y}$$

New losses:

$$\ell_1(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\intercal \mathbf{v} \qquad \ell_2(\mathbf{u}, \mathbf{v}) = -\mathbf{u}^\intercal \mathbf{v}$$

Eg: zero-sum bimatrix games

$$\ell_1(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\intercal \mathbf{v} \qquad \ell_2(\mathbf{u}, \mathbf{v}) = -\mathbf{u}^\intercal \mathbf{v} \qquad \xi = (\mathbf{v}, -\mathbf{u})$$



Hamiltonian: $$\mathcal{H}(\mathbf{u}, \mathbf{v}) = \frac{1}{2}(\mathbf{u}^\intercal \mathbf{u} + \mathbf{v}^\intercal \mathbf{v})$$

Level sets are ellipses (in original coordinates)

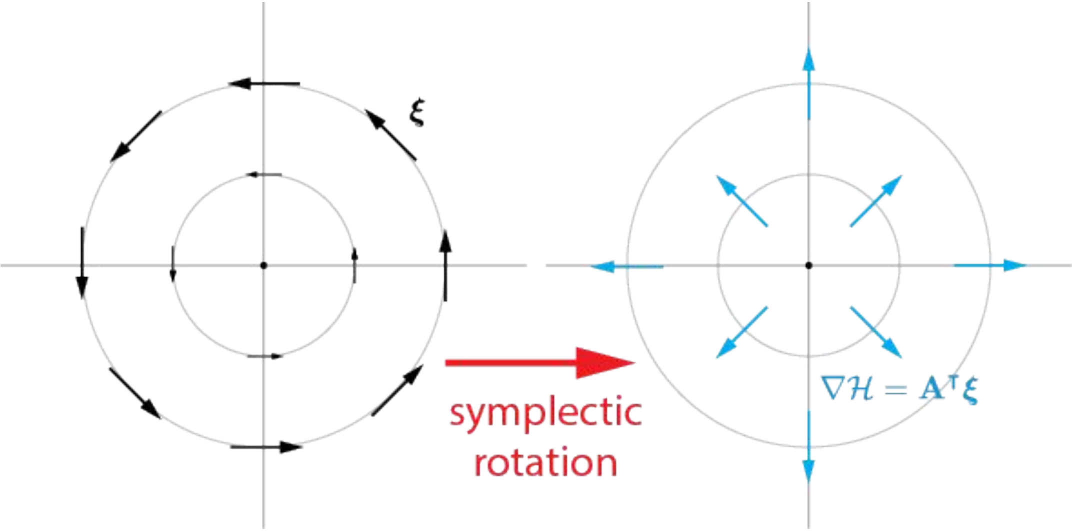
Hamiltonian dynamics:

$$\xi = (\nabla_\mathbf{v}\mathcal{H}, -\nabla_\mathbf{u}\mathcal{H})$$

How to solve Hamiltonian games

- Level sets of the Hamiltonian (ellipses) are **conserved** by simultaneous gradient descent on the losses

- Gradient descent on the **Hamiltonian** (**not** the losses) finds **Nash equilibrium**

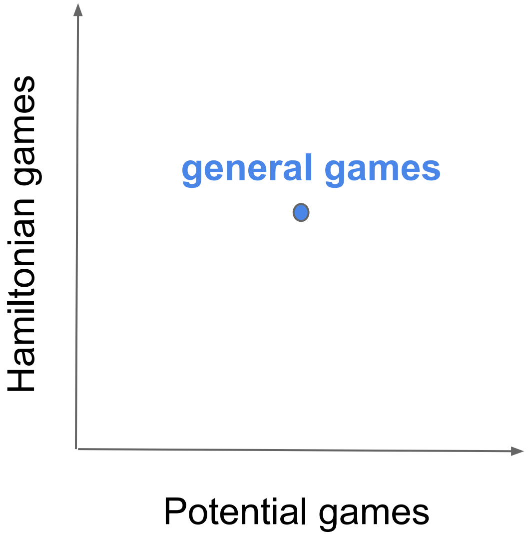


Game over?

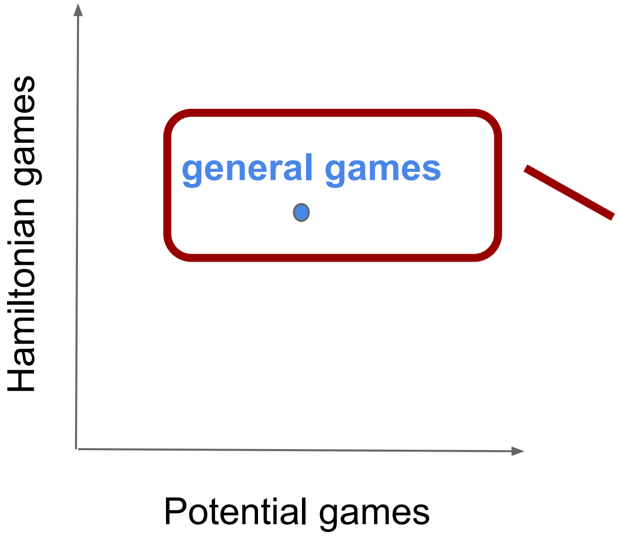
- Constructing the Hamiltonian relied on simultaneously SVD-ability of losses.

- Can something like this be done in general? **No.**

The big picture

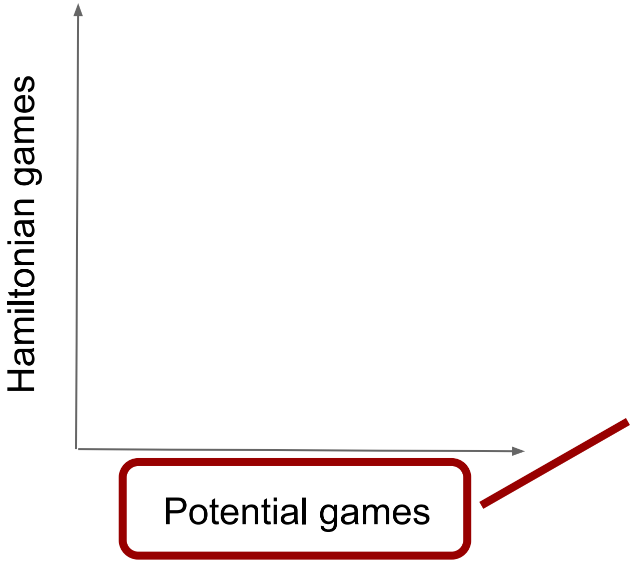


The big picture



- PPAD hard (even for good old fashioned games)
- No tractable, general purpose method

The big picture

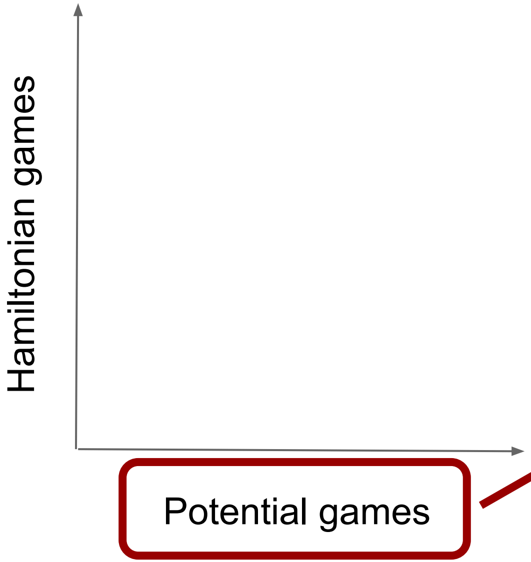


Hamiltonian games

Potential games

- Studied by game theorists for 30+ years
- “cooperative”
- Simultaneous gradient descent on **losses** finds local Nash

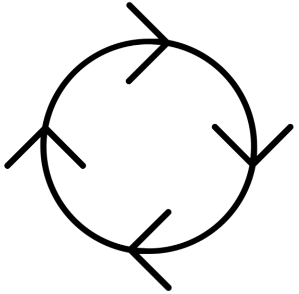
The big picture

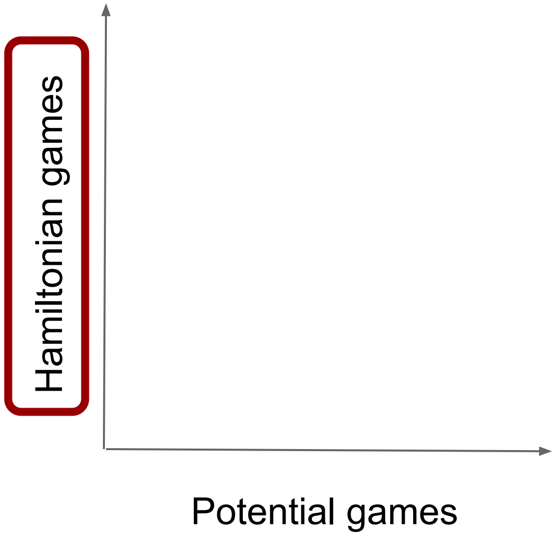

Hamiltonian games

Potential games

- (supervised) deep learning lives here

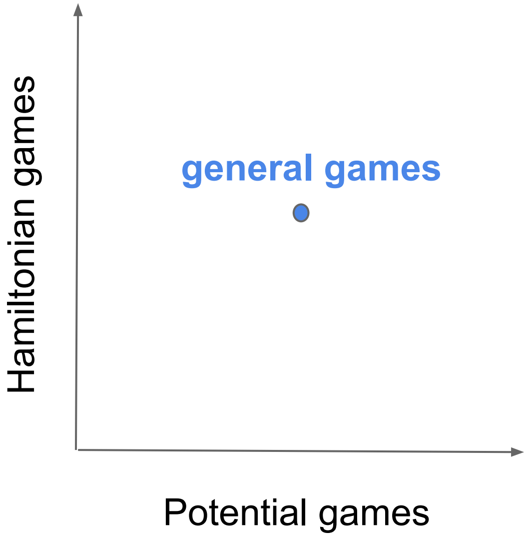
The big picture





- New class of games
- “hyper-adversarial”
- Gradient descent on **Hamiltonian** finds local Nash

The big picture



- Gradient descent on **Hamiltonian** finds local Nash

- Gradient descent on **losses** finds local Nash

Infinitesimal structure of gradients

$$\ell_1(x, y) = xy \qquad \ell_2(x, y) = -xy$$

$$\xi = \left(\frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y}\right) = (y, -x)$$

Game Hessian:

$$\mathbf{H}_\xi = \begin{pmatrix} \frac{\partial \xi_1}{\partial x} & \frac{\partial \xi_1}{\partial y} \\ \frac{\partial \xi_2}{\partial x} & \frac{\partial \xi_2}{\partial y} \end{pmatrix}$$

Infinitesimal structure of gradients

$$\ell_1(x,y) = xy \qquad \ell_2(x,y) = -xy$$

$$\xi = \left(\frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y}\right) = (y, -x)$$

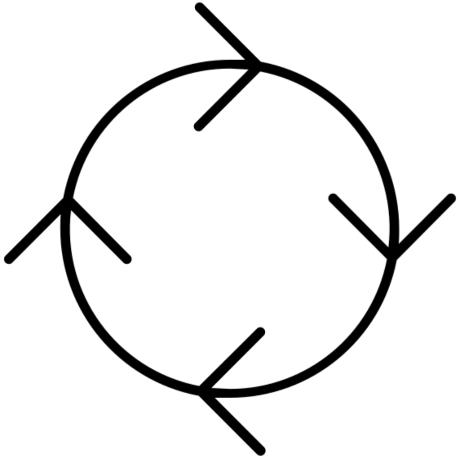
Generalized Helmholtz decomposition:

$$\mathbf{H}_\xi = \begin{pmatrix} \frac{\partial \xi_1}{\partial x} & \frac{\partial \xi_1}{\partial y} \\ \frac{\partial \xi_2}{\partial x} & \frac{\partial \xi_2}{\partial y} \end{pmatrix} = \underbrace{\mathbf{S}}_{\frac{\mathbf{H}+\mathbf{H}^\intercal}{2}} + \underbrace{\mathbf{A}}_{\frac{\mathbf{H}-\mathbf{H}^\intercal}{2}}$$

Infinitesimal structure of gradients

$$\ell_1(x, y) = xy \qquad \ell_2(x, y) = -xy$$

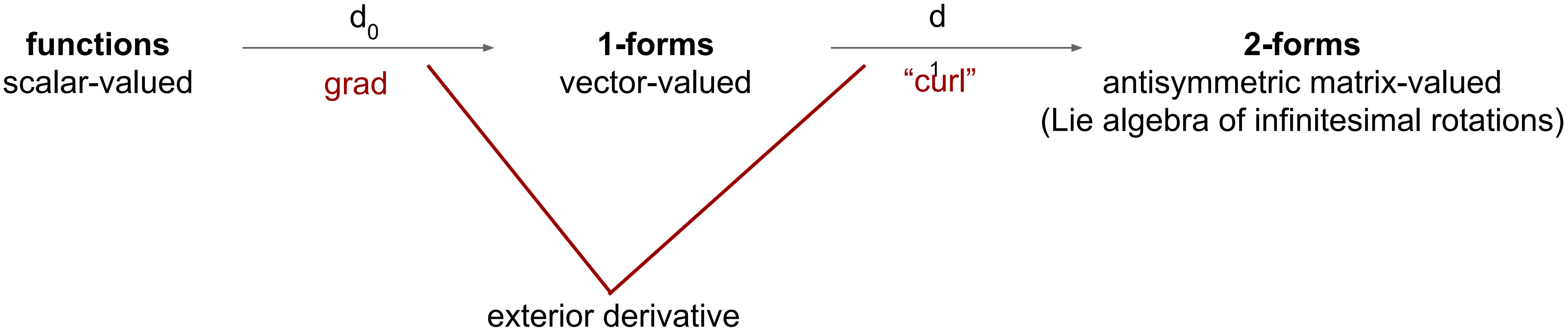
$$\xi = \left(\frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y}\right) = (y, -x)$$



Generalized Helmholtz decomposition:

$$H_\xi = \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}}_{S} + \underbrace{\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}}_{A}$$

Div, grad, and curl (again)



$$\text{functions} \xrightarrow[\text{grad}]{d_0} \text{1-forms} \xrightarrow[\text{“curl”}]{d_1} \text{2-forms}$$

functions
scalar-valued

grad

1-forms
vector-valued

“curl”

2-forms
antisymmetric matrix-valued
(Lie algebra of infinitesimal rotations)

exterior derivative

Div, grad, and curl (again)

$$\begin{tikzcd}\textbf{functions} \arrow[r, "d_0", "\text{grad}"'] & \textbf{1-forms} \arrow[r, "d_1", "\text{“curl”}"'] & \textbf{2-forms}\end{tikzcd}$$

functions
scalar-valued

grad

1-forms
vector-valued

“curl”

2-forms
antisymmetric matrix-valued
(Lie algebra of infinitesimal rotations)

$$\xi = \left(\frac{\partial \ell_1}{\partial x}, \frac{\partial \ell_2}{\partial y}\right) = (y, -x)$$

$$d_1 \xi = \begin{pmatrix} \frac{\partial \xi_1}{\partial x} & \frac{\partial \xi_1}{\partial y} \\ \frac{\partial \xi_2}{\partial x} & \frac{\partial \xi_2}{\partial y} \end{pmatrix} = A$$

2-form measures failure to
be a gradient vector field

Div, grad, and curl (again)

$$\begin{tikzcd} \textbf{functions} \arrow[r, "d_0", "\text{grad}"'] & \textbf{1-forms} \arrow[r, "d_1", "\text{“curl”}"'] & \textbf{2-forms} \end{tikzcd}$$

scalar-valued

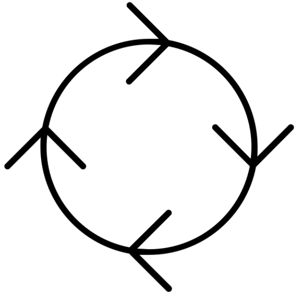
vector-valued

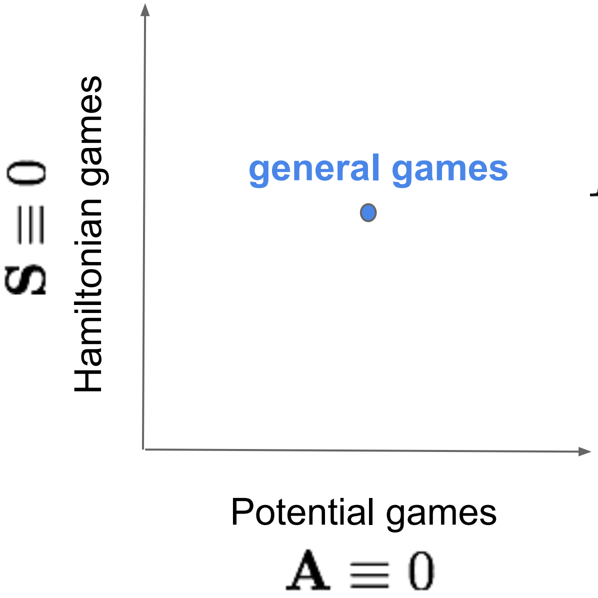
antisymmetric matrix-valued
(Lie algebra of infinitesimal rotations)

The generalized Helmholtz decomposition:

The game Hessian decomposes as $H = \underbrace{S}_{\text{“gradient component”}} + \underbrace{A}_{\text{“curl component”}}$

The big picture





Hamiltonian games

$\mathbf{S} \equiv 0$

general games

Potential games

$\mathbf{A} \equiv 0$

$$H = S + A$$

Symplectic Gradient Adjustment (SGA)

$$\xi + \lambda \cdot A^{\intercal}\xi$$

- $\lambda = \pm 1$
- computational cost is 2x backprop

Symplectic Gradient Adjustment (SGA)

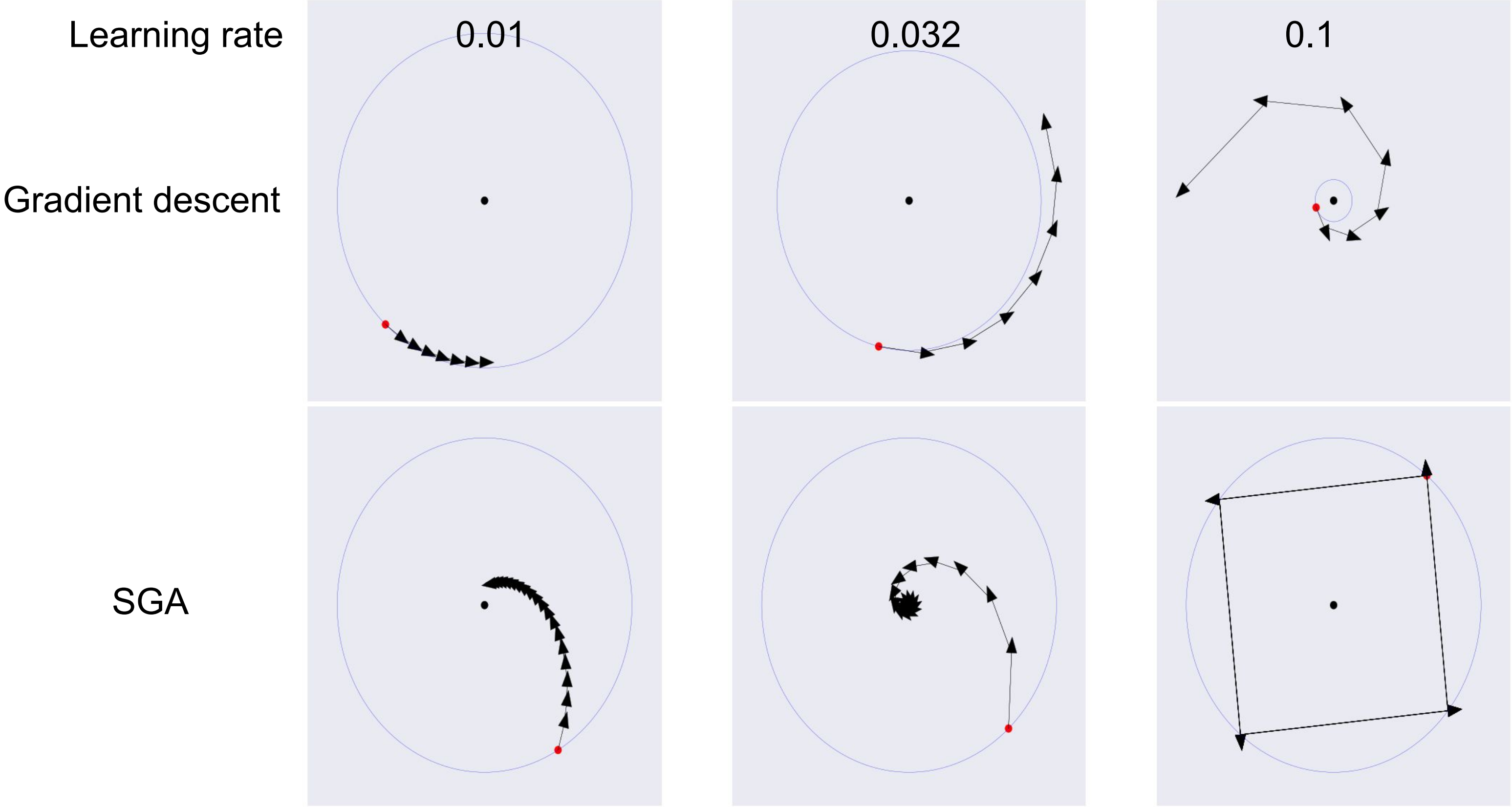
$$\xi + \lambda \cdot A^{\intercal}\xi$$

Properties:

- $\xi \perp A^{\intercal}\xi$: adjustment is compatible with original dynamics
  - Related: **consensus optimization** (Mescheder *et al*, NIPS 2017), $\xi + \lambda \cdot H^{\intercal}\xi$ which is attracted to local maxima

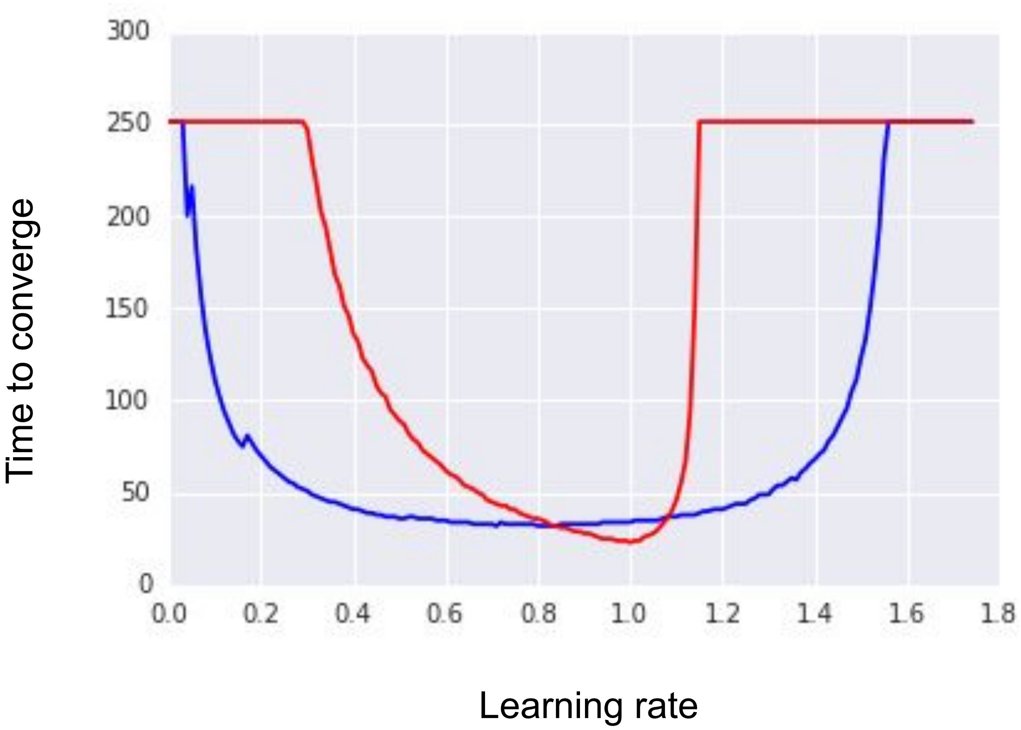
- if potential game then SGA is gradient descent → finds local min
- if Hamiltonian game then SGA finds **local Nash equilibrium**
- behaves correctly near stable and unstable equilibria

SGA allows higher learning rates

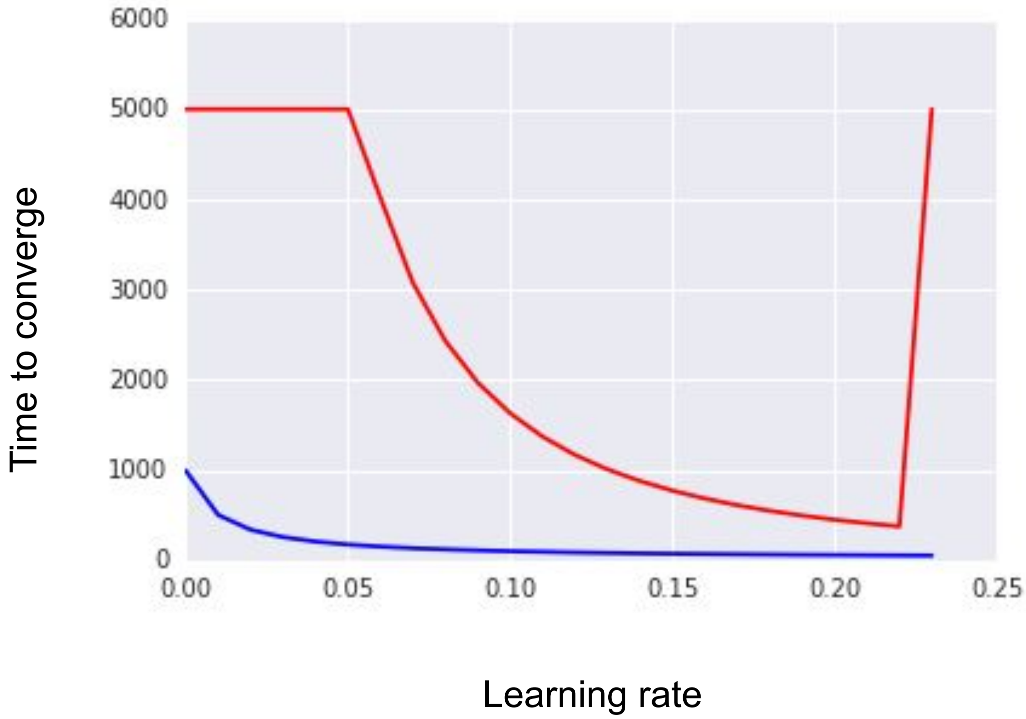


DeepMind

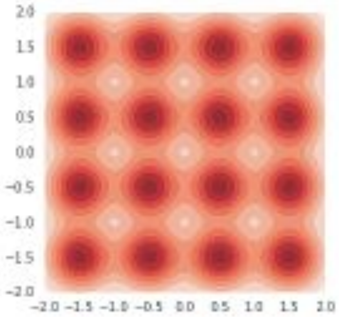
Comparison with Optimistic Mirror Descent: 2-players



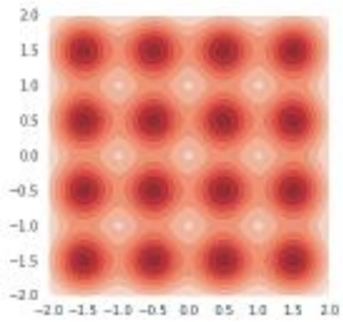
Comparison with Optimistic Mirror Descent: 4-players

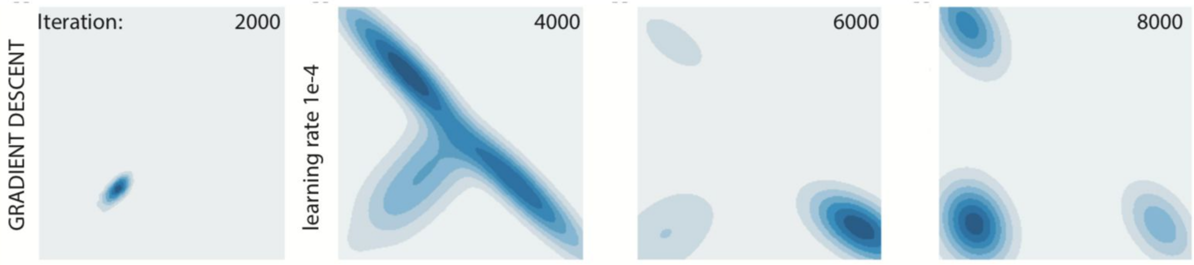


Performance on synthetic GAN

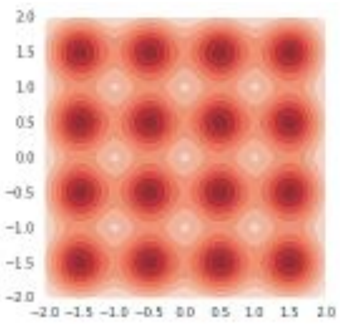


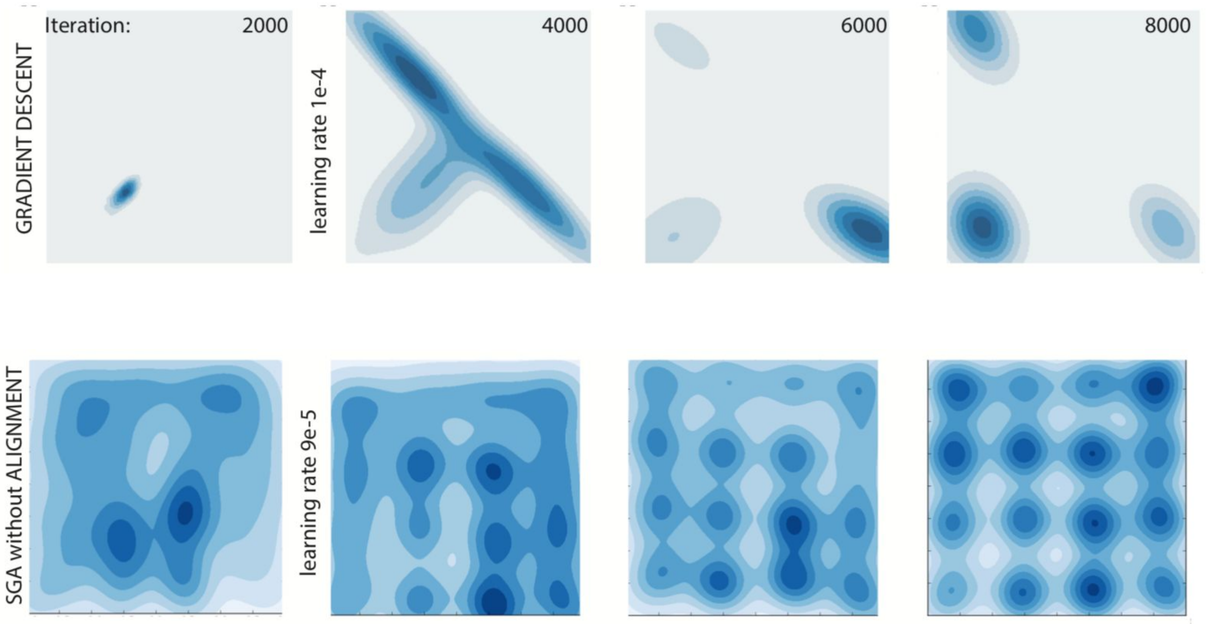
Performance on synthetic GAN





Performance on synthetic GAN





Summary

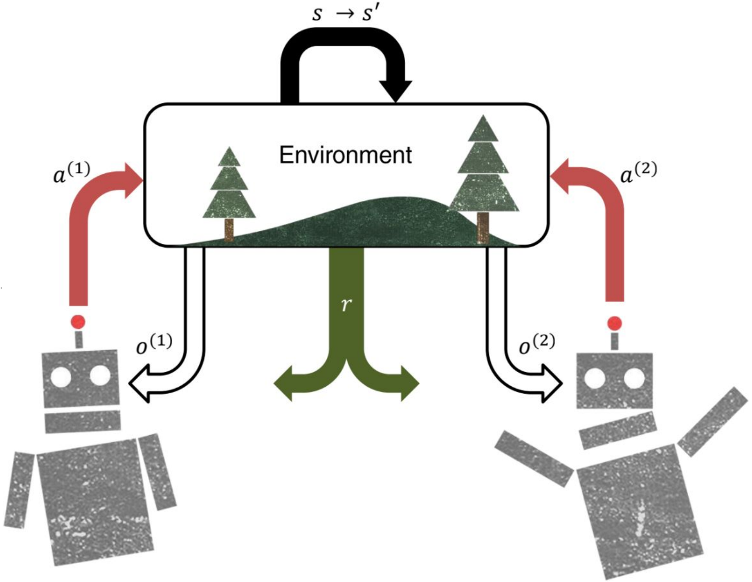
- Deep (supervised) learning is gradient descent on a loss
    - Simple, effective, one-concept-fits-all
    - Compositionality comes for free

- We're starting to work with interacting losses
    - We don't really know when or how to compose losses
    - There's real thinking to be done

6. Multi-agent Learning at Scale

DeepMind

Multi-agent Reinforcement Learning (MARL)



Objective: find policy that maximizes local or joint value:

Competitive Cooperative

$$V^* = \max_{\boldsymbol{\pi}} \mathbb{E}\left[\sum_t \gamma^t \mathcal{R}(s_t, \boldsymbol{a}_t) | P(s_0), \boldsymbol{\pi}\right]$$

MARL: Training and Execution

Decentralized Learning

Centralized Learning

Centralized Execution

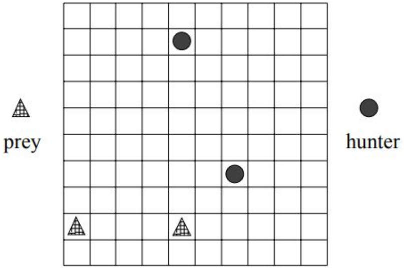
Decentralized Execution

Independent Q-Learning Approaches

Independent Q-learning [Tan, 1993]

$$Q(x,a) \leftarrow Q(x,a) + \beta(r + \gamma V(y) - Q(x,a))$$

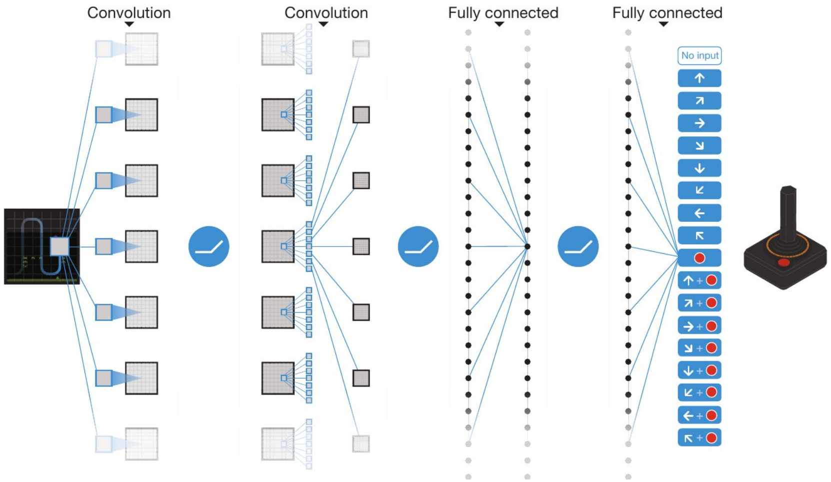
$$V(x) = \max_{b \in actions} Q(x,b)$$

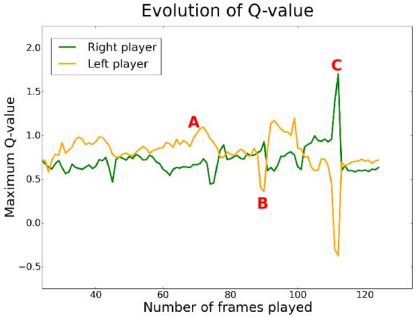


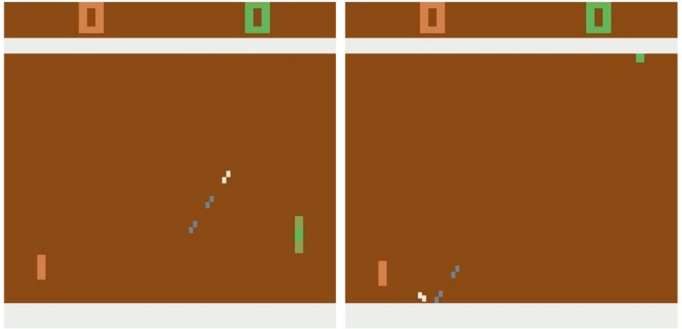
N-of-prey/N-of-hunters	1/1	1/2
Random hunters	123.08	56.47
Learning hunters	25.32	12.21

Table 1: Average Number of Steps to Capture a Prey

Independent Deep Q-Networks [Tampuu et al., 2015]







Lenient Learning Approaches

- **Issue**: Non-stationarities → policy/Q-value degradation and destabilization
- **Idea**: learners should be lenient against/ignore Q-value degradation
  - See Lenient Deep Q-Networks (Palmer et al., 2018) and Hysteretic Q-Networks (Omidshafiei et al., 2017)

Hysteretic Q-Networks:

$$L(\theta_j^i) = \underbrace{(r_t^i + \gamma \max_{a'} Q(o_{t+1}^i, h_t^i, a'; \hat{\theta}_j^i) - Q(o_t^i, h_{t-1}^i, a_t^i; \theta_j^i))}_{\text{Local TD error } \delta_t^i}{}^2$$

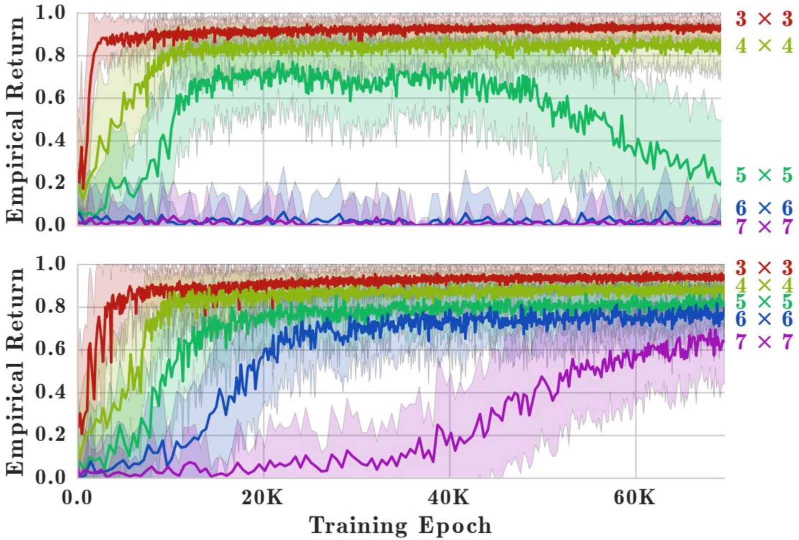
$$\theta_j \leftarrow \begin{cases} \theta_j - \alpha \nabla_{\theta_j} L(\theta_j^i) & \delta_t^i > 0 \text{ (underestimate)} \\ \theta_j - \beta \nabla_{\theta_j} L(\theta_j^i) & \delta_t^i \leq 0 \text{ (overestimate/degradation)} \end{cases} \quad \text{where } 0 < \beta < \alpha$$

Lenient Learning Approaches

- **Issue**: Non-stationarities → policy/Q-value degradation and destabilization
- **Idea**: learners should be lenient against/ignore Q-value degradation

Non-lenient learning

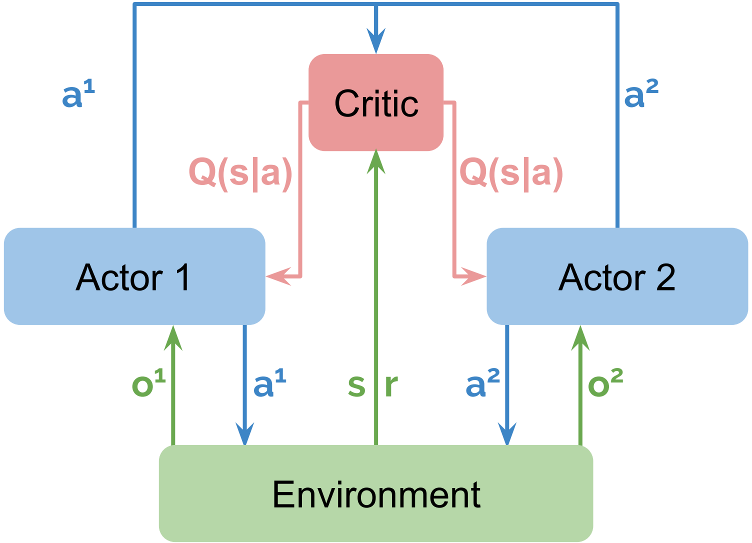
Lenient learning



- Converges to optimal in deterministic cooperative MDPs [Lauer et al., 2000]

Centralized Critic Decentralized Actor Approaches

- **Idea**: reduce nonstationarity & credit assignment issues using a central critic
- **Examples**: MADDPG [Lowe et al., 2017] & COMA [Foerster et al., 2017]
- Apply to both cooperative and competitive games



Centralized critic trained to minimize loss:

$$\mathcal{L}(\theta_i) = \mathbb{E}_{\mathbf{x},a,r,\mathbf{x}'}[(Q_i^{\boldsymbol{\pi}}(\mathbf{x}, a_1, \ldots, a_N) - y)^2],$$

$$y = r_i + \gamma Q_i^{\boldsymbol{\pi}'}(\mathbf{x}', a_1', \ldots, a_N')\big|_{a_j'=\boldsymbol{\pi}_j'(o_j)}$$

Decentralized actors trained via policy gradient:

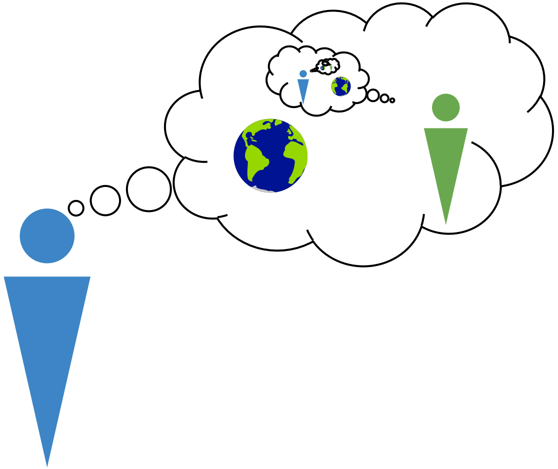
$$\nabla_{\theta_i} J(\theta_i) = \mathbb{E}_{s\sim p^{\boldsymbol{\mu}}, a_i\sim\boldsymbol{\pi}_i}[\nabla_{\theta_i}\log \underbrace{\boldsymbol{\pi}_i(a_i|o_i)}_{\text{Actor}} \underbrace{Q_i^{\boldsymbol{\pi}}(\mathbf{x}, a_1, ..., a_N)}_{\text{Critic}}]$$

Opponent-aware Models

- **Idea:** account for beliefs, models, and/or learning algorithms of other agents

Interactive POMDPs [Gmytrasiewicz & Doshi, 2005]

Maintain a belief over environment state *and* the other agents' models (e.g., learning algorithms, observation functions, their beliefs over other agents, etc.)



Extended Replicator Dynamics [Tuyls et al., 2003]

In standard replicator dynamics (RD), player strategies evolve greedily w.r.t. current payoff:

$$\frac{dx_i}{dt} = \underbrace{[(A\mathbf{x})_i - \mathbf{x}\cdot A\mathbf{x}]x_i}_{\textbf{RD(x)}}$$

In the extended RD, players take into account payoff growth in the future:

$$f(x) = RD(x) + \underbrace{(dRD(x)/dt) * \eta}_{\textbf{2nd order term}}$$

Learning with Opponent-Learning Awareness (LOLA) [Foerster et al., 2018]

"Naive" learner policy gradient update for agent 1:

$$\boldsymbol{\theta}^1_{i+1} = \boldsymbol{\theta}^1_i + f^1_{\text{nl}}(\boldsymbol{\theta}^1_i, \boldsymbol{\theta}^2_i),$$
$$f^1_{\text{nl}} = \nabla_{\boldsymbol{\theta}^1_i} V^1(\boldsymbol{\theta}^1_i, \boldsymbol{\theta}^2_i)\cdot\delta$$

Taylor-expand agent 1's value given agent 2's update:

$$V^1(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2 + \Delta\boldsymbol{\theta}^2)$$
$$\approx V^1(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2) + (\Delta\boldsymbol{\theta}^2)^T \nabla_{\boldsymbol{\theta}^2} V^1(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2)$$

Assuming agent 2 is a naive learner with update

$$\Delta\boldsymbol{\theta}^2 = \nabla_{\boldsymbol{\theta}^2} V^2(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2)\cdot\eta$$

then we arrive at the LOLA update rule:

$$f^1_{\text{lola}}(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2) = \nabla_{\boldsymbol{\theta}^1} V^1(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2)\cdot\delta$$
$$+ \left(\nabla_{\boldsymbol{\theta}^2} V^1(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2)\right)^T \nabla_{\boldsymbol{\theta}^1}\nabla_{\boldsymbol{\theta}^2} V^2(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2)\cdot\delta\eta$$

Games and Reinforcement Learning

Game theory

- Solutions are **strategy profiles** specifying joint actions at all possible **information sets**

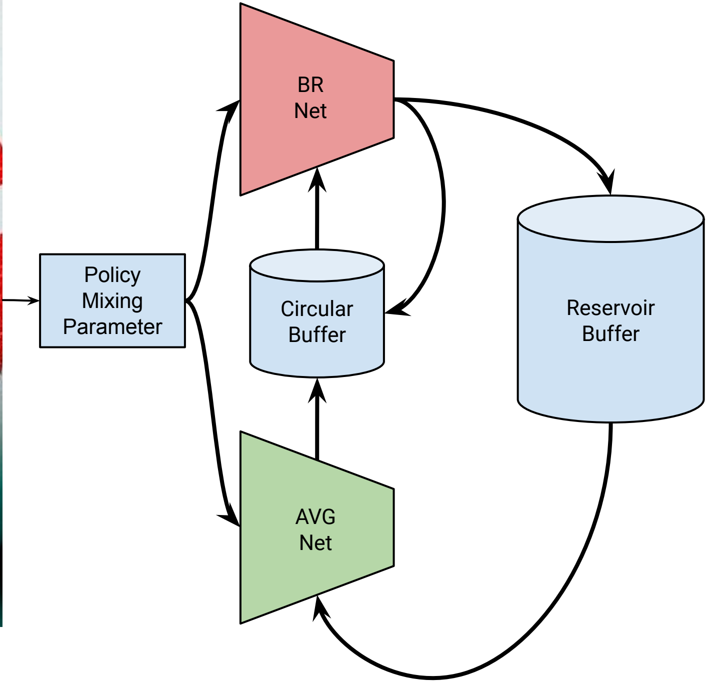
Reinforcement learning

- Solutions are **joint policies** specifying joint actions at all possible **partially observed states**

Neural Fictitious Self-Play [Heinrich & Silver 2016]

- **Idea:** Fictitious self-play (FSP) + deep reinforcement learning
- Approximate NE via two neural networks:

1. **Best response net (BR):**
    - Estimate a best response
    - Trained via RL
2. **Average policy net (AVG):**
    - Estimate the time-average policy
    - Trained via supervised learning



Policy Mixing Parameter

BR Net

Circular Buffer

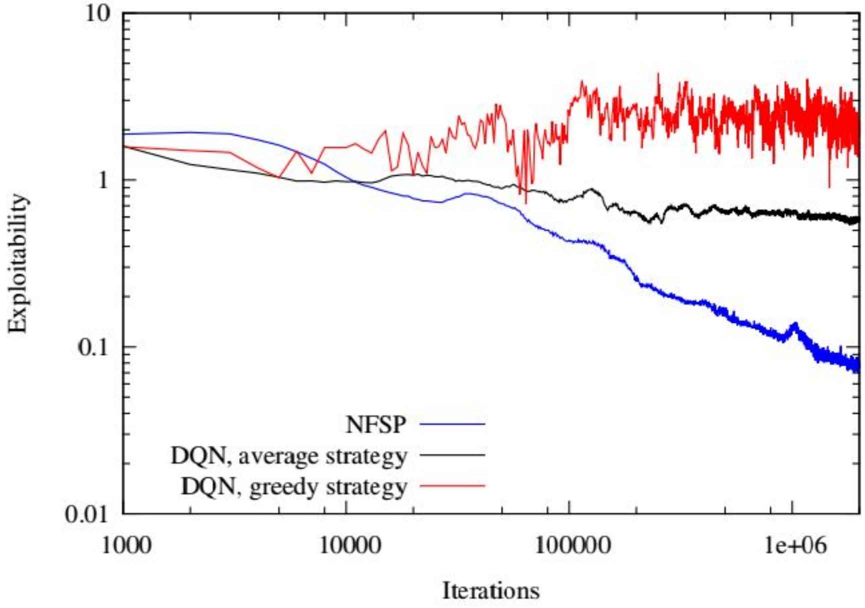
AVG Net

Reservoir Buffer

Neural Fictitious Self-Play [Heinrich & Silver 2016]

- Leduc Hold’em poker experiments:

“Closeness” to Nash



- 1st scalable end-to-end approach to learn **approximate Nash equilibria w/o prior domain knowledge**
    - Competitive with superhuman computer poker programs when it was released

Learning under Nonstationarity

Policy Gradient (Advantage Actor-Critic)

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\pi}}\left[\nabla_{\boldsymbol{\theta}} \log \pi(a_t \mid s_t; \boldsymbol{\theta}) A(s_t, a_t; \boldsymbol{w}, \boldsymbol{\theta})\right]$$

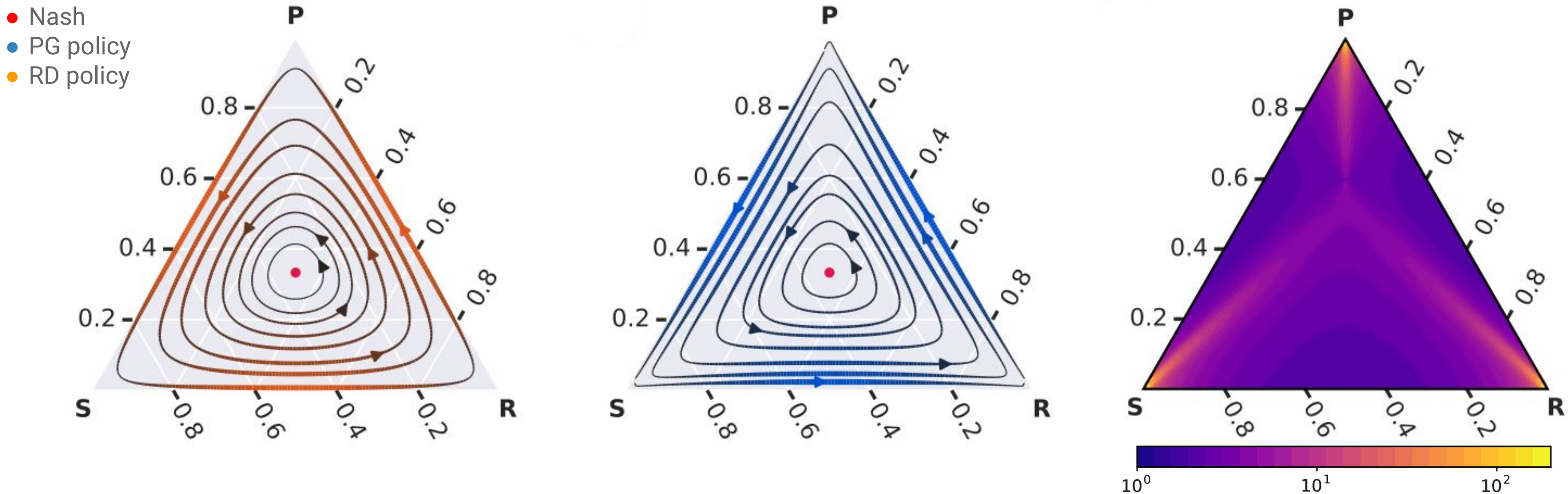
Replicator Dynamics

$$\dot{\pi}(a) = \pi(a) A(a)$$

logit space $\boldsymbol{\pi} = softmax(\boldsymbol{y})$ stateless tabular case

$$y_t(a) = y_{t-1}(a) + \eta \pi(a) A(a)$$

$$y_t(a) = y_{t-1}(a) + \eta A(a)$$



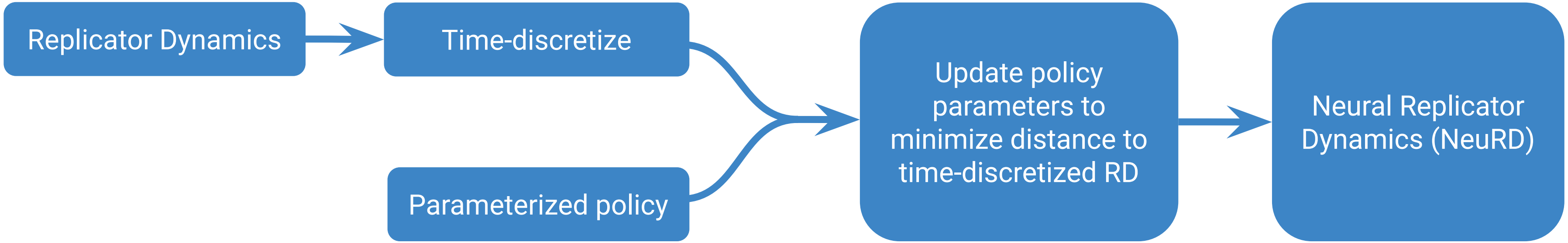
$$\frac{||\dot{\boldsymbol{\pi}}_{\mathrm{RD}}||}{||\dot{\boldsymbol{\pi}}_{\mathrm{PG}}||}$$

Neural Replicator Dynamics (NeuRD)

- Policy Gradient handles high-dimensional state- and action-spaces seamlessly
    - Replicator Dynamics are limited to tabular settings
- Replicator Dynamics are no-regret (time-average convergence to Nash)
    - Policy Gradient has no such guarantees

Neural Replicator Dynamics: *best of both worlds!*

Neural Replicator Dynamics (NeuRD)



Replicator Dynamics

Time-discretize

Parameterized policy

Update policy parameters to minimize distance to time-discretized RD

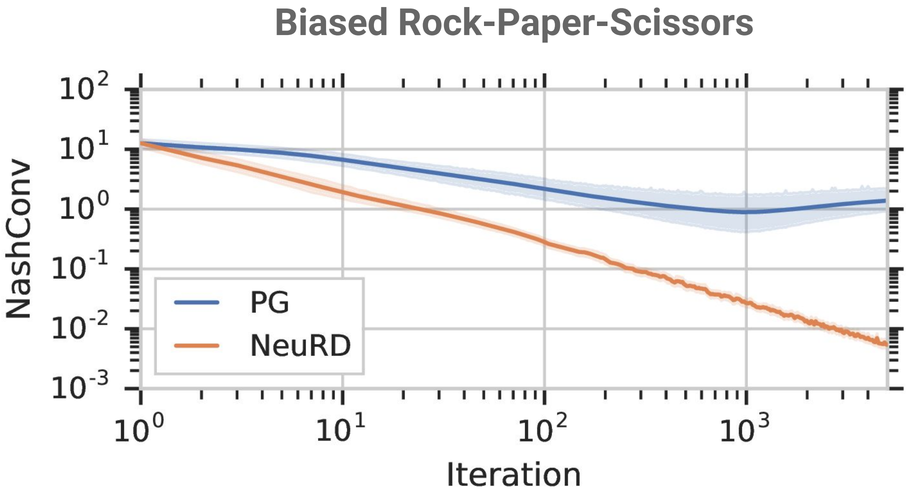
Neural Replicator Dynamics (NeuRD)

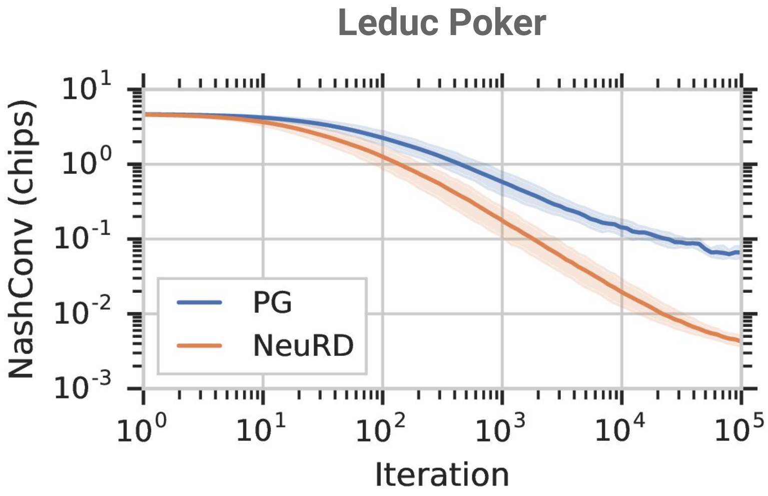
$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t+1} + \eta \sum_{s,a} \nabla_{\boldsymbol{\theta}} \underbrace{y_{t-1}(s_t, a_t; \boldsymbol{\theta})}_{\text{Logits, where policy is } \boldsymbol{\pi} = softmax(\boldsymbol{y})} \underbrace{A(s_t, a_t; \boldsymbol{\theta}, \boldsymbol{w})}_{\text{Advantage } q(s,a)-v(s)}$$

Logits, where policy is
$\boldsymbol{\pi} = softmax(\boldsymbol{y})$

Advantage q(s,a)-v(s)

Results


Biased Rock-Paper-Scissors


Leduc Poker

A MARL Retrospective

Foundational Algorithm	Modern and/or Deep RL Counterpart
Fictitious Play [Brown, 1951]	Extensive-form Fictitious Play [Heinrich et al., 2015] Neural Fictitious Self-Play [Heinrich & Silver, 2016]
Independent Q-learning [Tan, 1993]	Multi-agent Deep Q-Networks [Tampuu et al., 2015]
Double Oracle [McMahan et al., 2003]	Policy-Space Response Oracles [Lanctot et al., 2017]
Hysteretic Q-learning [Matignon et al., 2007]	Recurrent Hysteretic Q-Networks [Omidshafiei et al., 2017]
Extended Replicator Dynamics [Tuyls et al., 2003]	Learning with Opponent-Learning Awareness [Foerster et al., 2017]
Lenient Learning [Panait et al., 2006; Panait, Tuyls, Luke, 2008]	Lenient Deep Q-Networks [Palmer, Tuyls et al., 2018]
Replicator Dynamics [Taylor & Jonker, 1978; Smith, 1982; Schuster & Sigmund, 1983]	Neural Replicator Dynamics [Omidshafiei et al., 2019]

Non-exhaustive list! For more, check out:

“Deep Reinforcement Learning for Multi-Agent Systems: A Review of Challenges, Solutions and Applications” (Nguyen et al., 2019)

“Is multiagent deep reinforcement learning the answer or the question? A brief survey” (Hernandez-Leal et al., 2018)

“Multiagent learning: Basics, challenges, and prospects." (Tuyls & Weiss, 2012)

“Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems.” (Matignon et al., 2008)

References

Tan, Ming. "Multi-agent reinforcement learning: Independent vs. cooperative agents." Proceedings of the tenth international conference on machine learning. 1993.

Tampuu, Ardi, et al. "Multiagent Cooperation and Competition with Deep Reinforcement Learning." arXiv preprint arXiv:1511.08779 (2015).

Matignon, Laëtitia, Guillaume J. Laurent, and Nadine Le Fort-Piat. "Hysteretic q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams." 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2007.

Omidshafiei, Shayegan, et al. "Deep decentralized multi-task multi-agent reinforcement learning under partial observability." Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017.

Liviu Panait, Keith Sullivan, and Sean Luke. 2006. Lenient learners in cooperative multiagent systems. In Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems. ACM, 801–803.

Palmer, Gregory, et al. "Lenient multi-agent deep reinforcement learning." Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Brown, George W. "Iterative solution of games by fictitious play." Activity analysis of production and allocation 13.1 (1951): 374-376.

Heinrich, Johannes, and David Silver. "Deep reinforcement learning from self-play in imperfect-information games." arXiv preprint arXiv:1603.01121 (2016).

H.B. McMahan, G. Gordon, and A. Blum. Planning in the presence of cost functions controlled by an adversary. In Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), 2003

Taylor, P. and L. Jonker (1978). Evolutionarily stable strategies and game dynamics. Math. Biosciences 40: 145-156.

Maynard Smith, J. (1982). Evolutionq Game theory. Cambridge University Press, Cambridge

Schuster. P. and K. Sigmund (1983). Replicator dynamics. J. %or. Biology 100: 535-538.

Tuyls, Karl, et al. "Extended replicator dynamics as a key to reinforcement learning in multi-agent systems." European Conference on Machine Learning. Springer, Berlin, Heidelberg, 2003.

Foerster, Jakob, et al. "Learning with opponent-learning awareness." Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Martin Lauer and Martin Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In Proc. of the Seventeenth International Conf. on Machine Learning. Citeseer, 2000.

Gmytrasiewicz, Piotr J., and Prashant Doshi. "A framework for sequential planning in multi-agent settings." Journal of Artificial Intelligence Research 24 (2005): 49-79.

7. Why are Games Important?

Wrap-up

DeepMind

Games as a Multi-Agent Platform

Image credit: S.M.S.I., Inc. – Owen Williams, The Kasparov Agency

How Life Imitates Chess G. Kasparov

“Unfortunately, the number of ways to do something wrong always exceeds the number of ways to do it right”

“A CEO must combine analysis and research with creative thinking to lead his company effectively”

Games for AI

Good controlled model for Multi-Agent Learning

- Simple rules, deep concepts
- Studied for hundreds or thousands of years
- Co-evolution artifact -> Learning
- ‘Drosophila’ of artificial intelligence
- Microcosmos encapsulating real world issues
- Games are fun!

Games for AI - A theory of Games

- Concept from traditional Game Theory
- Hyper-rational players
- Static concept

Intuitively: A **Nash Equilibrium** is a strategy profile for a game, such that no player can increase its payoff by unilaterally changing its strategy.

- Players are not hyper rational, but

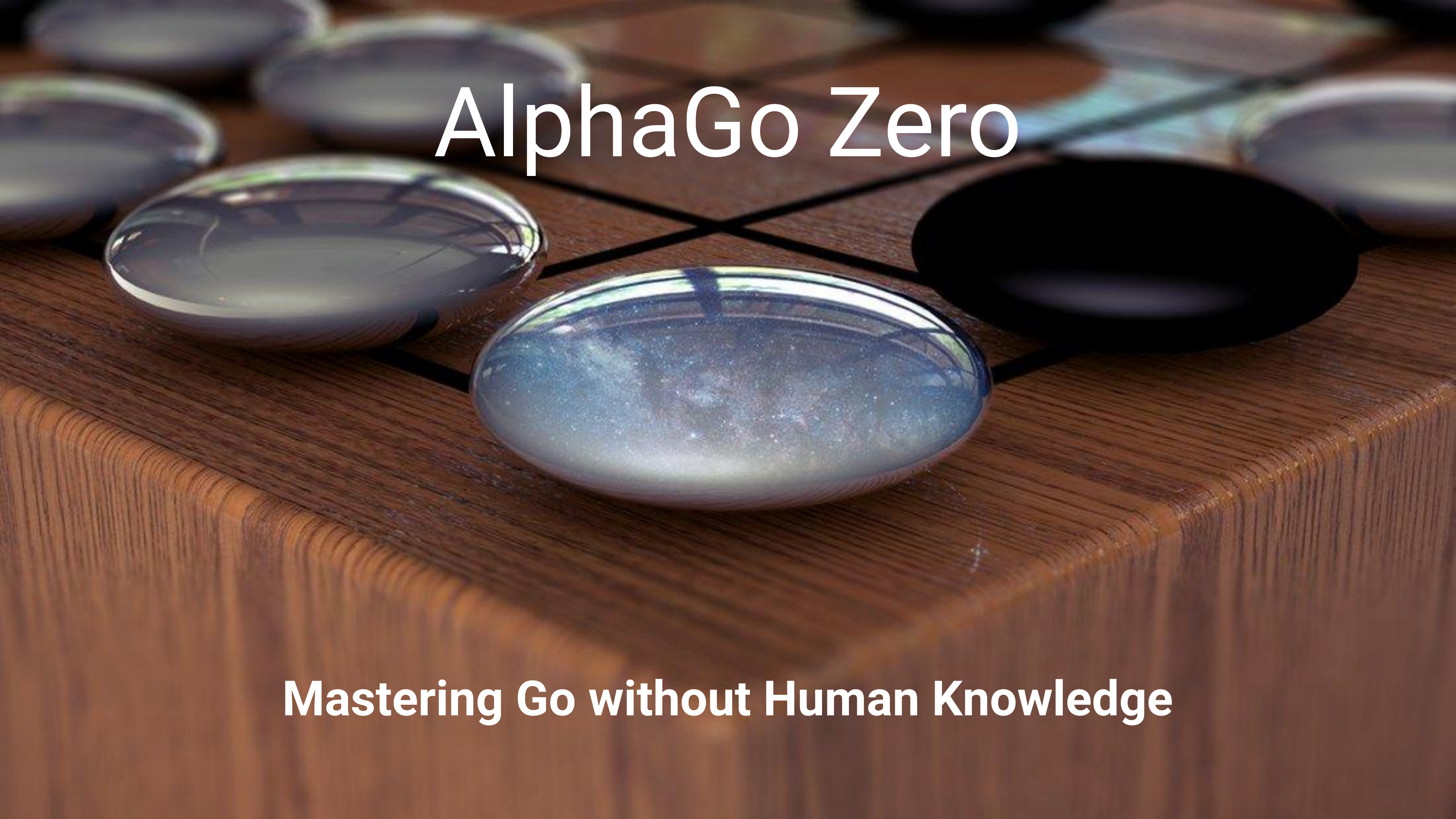
  also ***biologically*** and ***socially*** conditioned

Zero-Sum Games for AI

- Why are zero-sum games of interest?
  - Many standard AI benchmark domains are inherently zero-sum

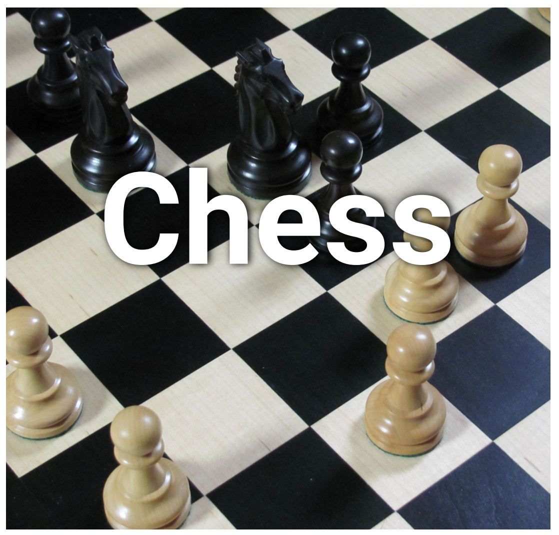
  - Strong theoretical guarantees for zero-sum games
  - Strict relations over outcomes → strategize by maximizing wins/rewards
  - Existence of standard algorithm evaluation methods

AlphaGo Zero



Mastering Go without Human Knowledge

AlphaZero: One Algorithm, Three Games



Video Games

Started with **toy MDPs**.

Grid worlds starting to feel like games.

Atari - very engaging for humans.

Mnih et al, 2018.

Video Games

Started with **toy MDPs**.

Grid worlds starting to feel like games.

Atari - very engaging for humans.

3D single-player - even richer potential task space. (**DeepMind Lab**, VizDoom, Minecraft)

A3C Vmnih et al 2016,

UNREAL Jaderberg et al, 2016.

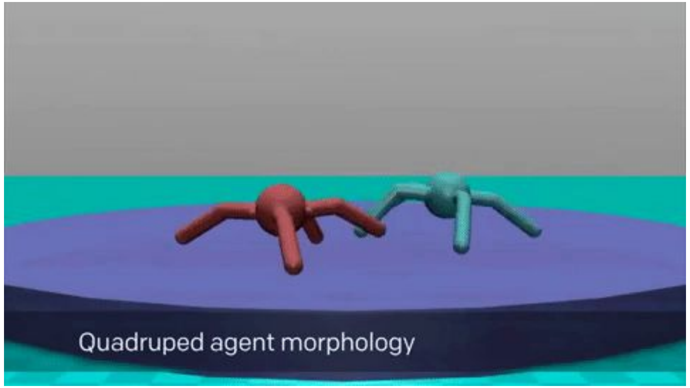
Video Games: **Multi-agent**

Much richer task space with simple rules: competitive and cooperative

Diversity of solution: robustness

Auto-curricula

Non-stationary: continual learning



Bansal et al, 2017.

Dorer vs Stone, 2017.

The Importance of Games

- Development of **general applicable** techniques in
  - Controlled **environments**
  - Fast **simulations**
  - Principled **evaluation** and **understanding**
  - Drives the **AI Frontiers**
- Can be deployed in various **other domains**
  - Fraud detection systems
  - Auction agents
  - Energy systems (smart grid)
  - Industry 4.0 systems