6.S890:
Topics in Multiagent Learning

Lecture 1

Fall 2023

Recent AI Breakthroughs

images

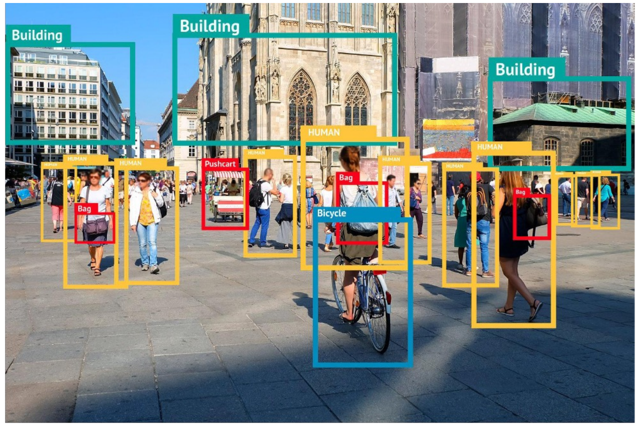
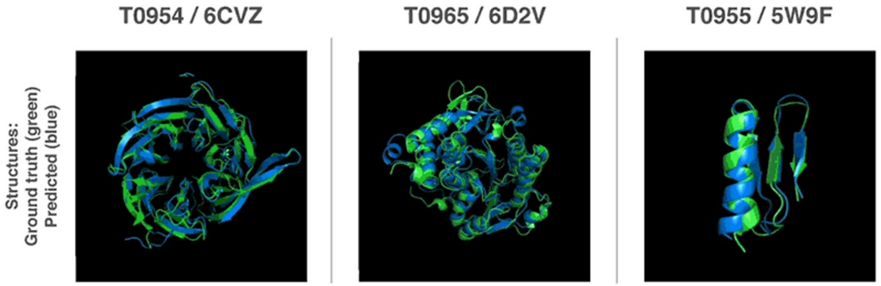


image recognition, reconstruction, generation, super-resolution,…

molecules



protein folding, molecule design,…

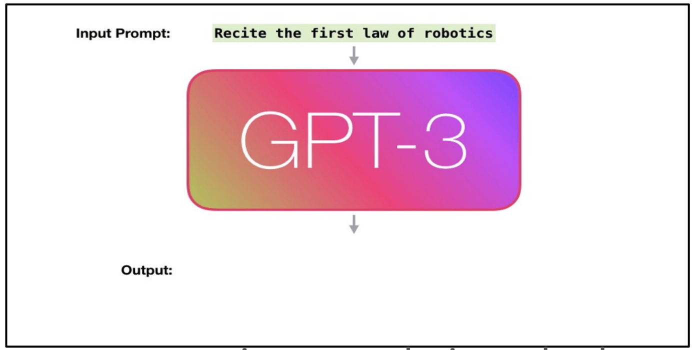
games

super-human play

time-series data

speech recognition, forecasting

natural language

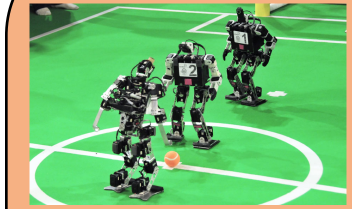


text generation, translation, chatbots, text embeddings,…

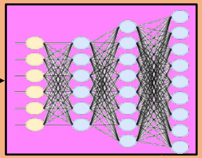
A Dawn of *Multi-Agent* Applications



Multi-player Game-Playing:
- Superhuman GO, Poker, Gran Turismo
- Human-level Starcraft, Diplomacy

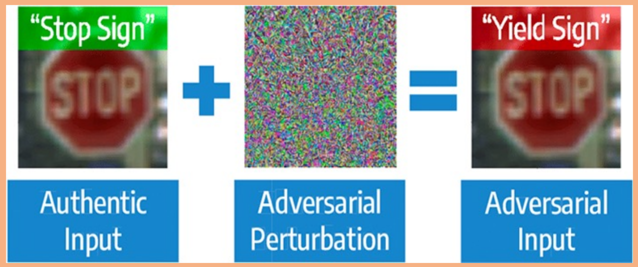


- Multi-robot interactions
- Autonomous driving
- Automated Economic policy design

$Z \sim \mathcal{N}(0, I) \longrightarrow$  $\longrightarrow$  $\sim P_{\text{target}}$

boring randomness — neural network — interesting randomness

Generative Adversarial Networks (GANs)
synthetic data generation



Adversarial Training
robustifying models against adversarial attacks

Example: Deep Generative Models

$G_\theta(Z)$

$Z \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ $G_\theta(\cdot)$  $\sim \boldsymbol{P}_{\text{interesting}}$

Deep Neural Network (DNN)
with well-tuned parameters
θ

Example: Deep Generative Models

How to train a Deep Generative Model?

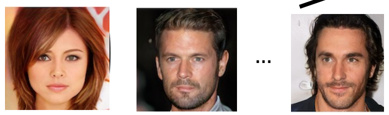
$Z \sim \mathcal{N}(0, I) \rightarrow G_\theta(\cdot) \rightarrow$  $\sim P_{\text{interesting}}$

[Goodfellow et al'14]: Set up a ***two-player zero-game*** between a player tuning the parameters θ of a Deep Neural Network (called the "generator") and a player tuning the parameters ω of a Deep Neural Network (called the "discriminator")

Real or Hallucinated?

Discriminator: DNN w/ parameters ω

Hallucinated Images (from generator)



Real Images (from training set)

Generator: DNN w/ parameters θ

Simple Randomness

$Z_1, Z_2, \ldots, Z_N \sim N(0, I)$

- Reward discriminator for *distinguishing* real from fake images
- Reward generator for *fooling* the discriminator

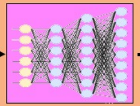
A Dawn of *Multi-Agent* Applications



Multi-player Game-Playing:
- Superhuman GO, Poker, Gran Turismo
- Human-level Starcraft, Diplomacy

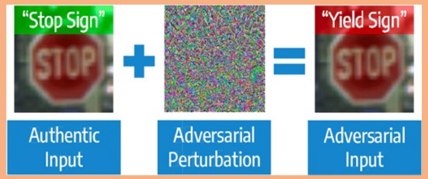


- Multi-robot interactions
- Autonomous driving
- Automated Economic policy design

$Z \sim \mathcal{N}(0, I) \longrightarrow$  $\longrightarrow$  $\sim P_{\text{target}}$

boring randomness | neural network | interesting randomness

Generative Adversarial Networks (GANs)
synthetic data generation



Adversarial Training
robustifying models against adversarial attacks

Important notes and caveats…

(I) Strategic Behavior does not emerge from standard training

ChatGPT

(I) Strategic Behavior does not emerge from standard training (cont'd)

CO

I am the x player in a game of tic-tac-toe, the other player is o, I am supposed to play next, and the current board configuration looks as follows. Where should I put x?

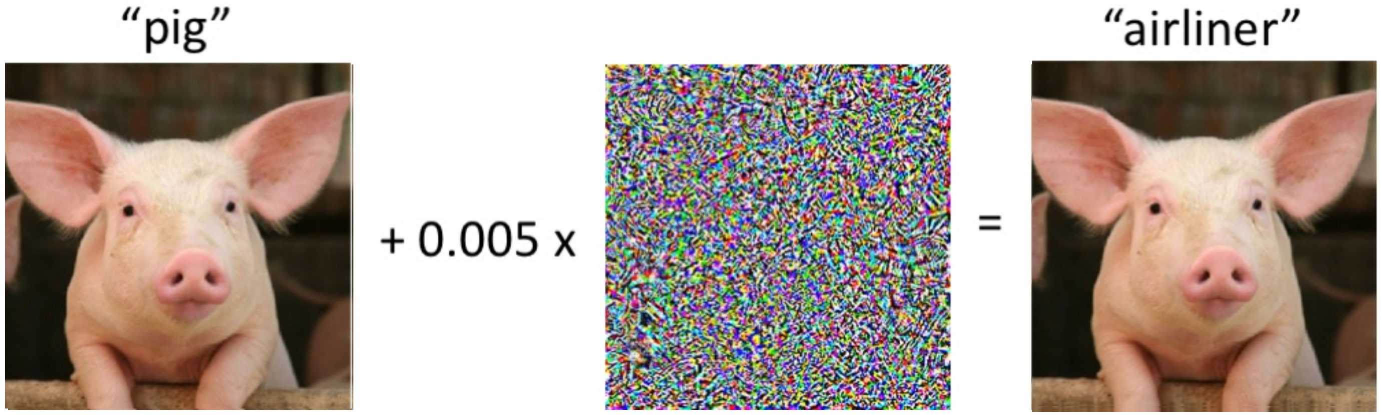
```
x| |x
o|o|
 | |
```

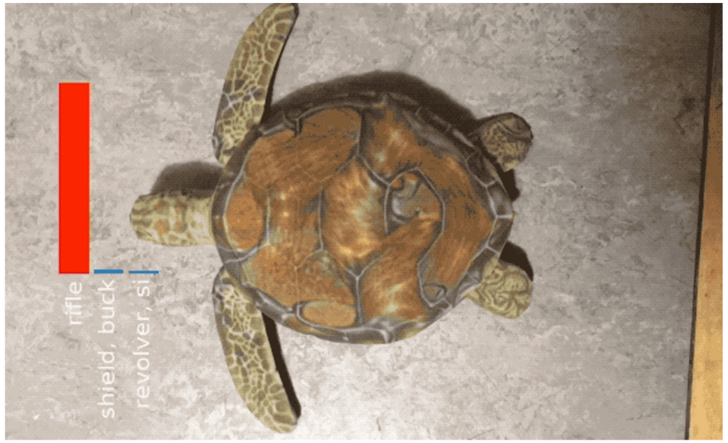
Based on the current board configuration, you should put an "x" in the middle-right cell to block your opponent from winning the game.

The updated board would look like this:

```
x| |x
o|o|x
| |
```

(II) Naively trained models can be manipulated





[Athalye, Engstrom, Ilyas, Kwok ICML’18]

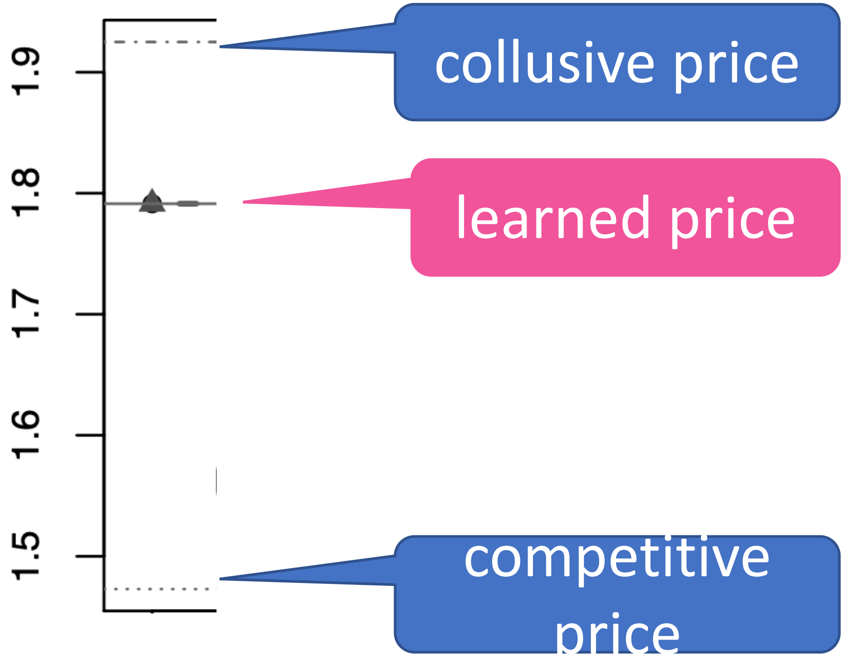
[Engstrom et al. 2019]

(III) Combining agents that were trained in isolation can lead to undesirable behavior

Example: AI for dynamic pricing

Setting: Duopoly w/ two symmetric firms

Independent Learning: firms cannot communicate other than setting prices, observing their profit and adjusting their price using some standard AI algorithm



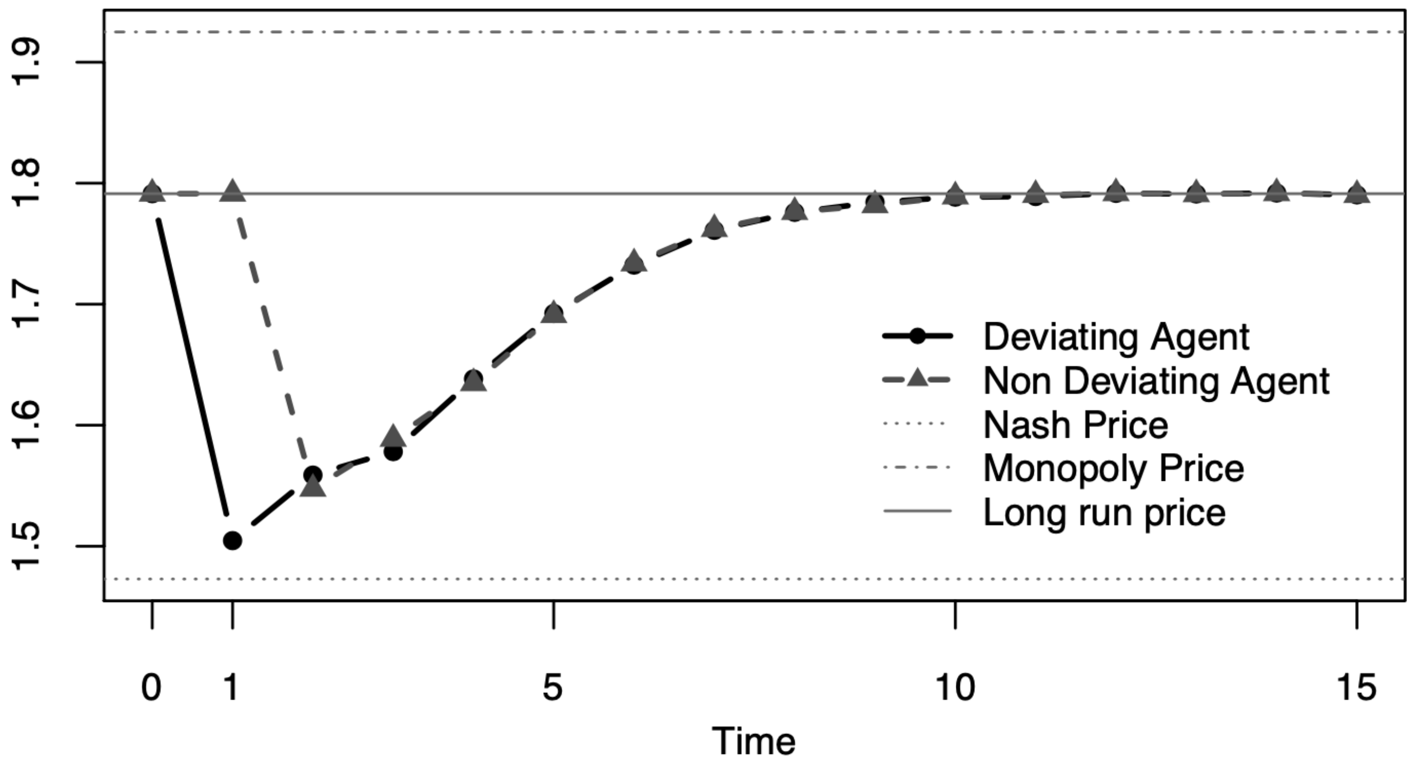
[Calvano, Calzolari, Denicolo, Pastorello: “Artificial Intelligence, Algorithmic Pricing, and Collusion,” American Economic Review, 2020]

(III) Combining agents that were trained in isolation can lead to undesirable behavior

Example: AI for dynamic pricing

Setting: Duopoly w/ two symmetric firms

Independent Learning: firms cannot communicate other than setting prices, observing their profit and adjusting their price using some standard AI algorithm



How deviations are punished by the learned price policies

[Calvano, Calzolari, Denicolo, Pastorello: "Artificial Intelligence, Algorithmic Pricing, and Collusion," American Economic Review, 2020]

(IV) The optimization workhorse of Deep Learning struggles in multi-agent settings

(IV) The optimization workhorse of Deep Learning struggles in multi-agent settings

STANDARD DEEP LEARNING ESTIMATION PROBLEM

$$\min_\theta \ell(\theta)$$

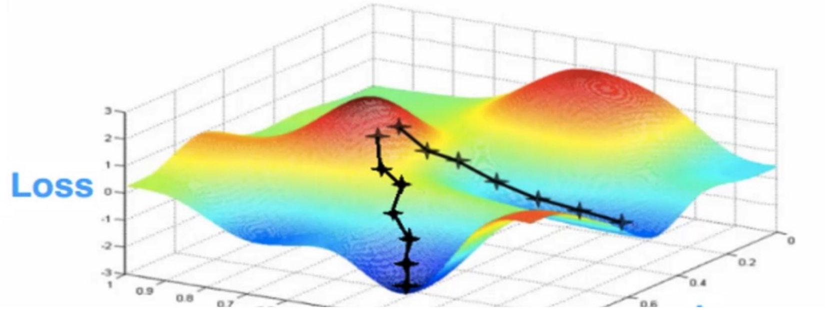
θ: high-dimensional

ℓ: nonconvex

essentially only accessible through $\ell(\theta)$ and $\nabla\ell(\theta)$ queries

$$\boxed{\theta_{t+1} = \theta_t - \eta \cdot \nabla\ell(\theta_t)}$$

Gradient Descent

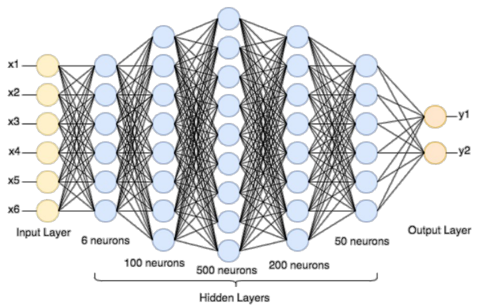


Theoretical Guarantee: Even if ℓ nonconvex, Gradient Descent efficiently computes *local minima* *[Lee et al 2017, Ge et al '15]*

Empirical Finding: *Local minima* are good enough

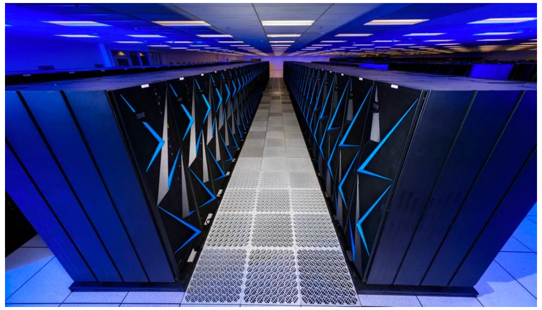
(IV) The optimization workhorse of Deep Learning struggles in multi-agent settings

Prominent Paradigm:



$$+ \ \theta_{t+1} \leftarrow \theta_t - \nabla_\theta \ell(\theta_t) \quad +$$

$+$



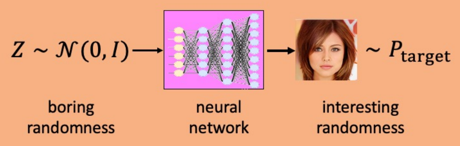
Caffe Caffe2 Chainer Microsoft Cognitive Toolkit MATLAB mxnet

PaddlePaddle PyTorch TensorFlow torch Wolfram Language

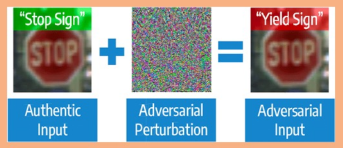
(IV) The optimization workhorse of Deep Learning struggles in multi-agent settings

Multi-player Game-Playing:
- Superhuman GO, Poker, Gran Turismo
- Human-level Starcraft, Diplomacy

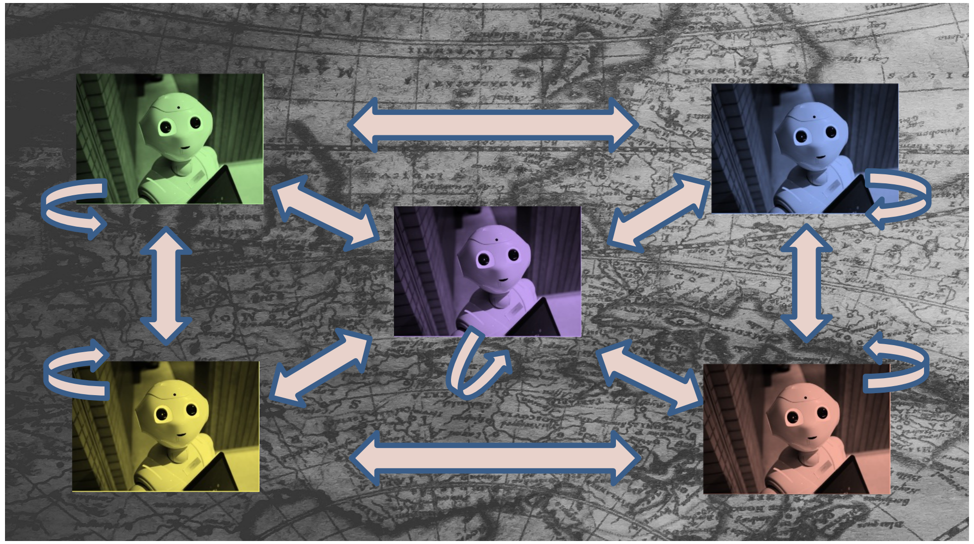
- Multi-robot interactions
- Autonomous driving
- Automated Economic policy design



Generative Adversarial Networks (GANs)
synthetic data generation



Adversarial Training
robustifying models against adversarial attacks



Practical Experience: GD vs GD (vs GD…) is cyclic or chaotic, and it is a hard engineering challenge to make it identify a good solution

(IV) The optimization workhorse of Deep Learning struggles in multi-agent settings

GANs: $\ell_D(\theta,\omega) = -\ell_G(\theta,\omega)$
ℓ_G, ℓ_D: nonconvex in θ & ω resp.;
θ, ω: high-dimensional

Simultaneous Gradient Descent (GD) Dynamics:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_\theta \ell_G(\theta_t, \omega_t)$$
$$\omega_{t+1} = \omega_t - \eta \cdot \nabla_\omega \ell_D(\theta_t, \omega_t)$$

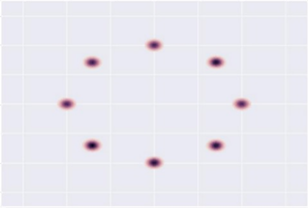
GAN training on MNIST Data:

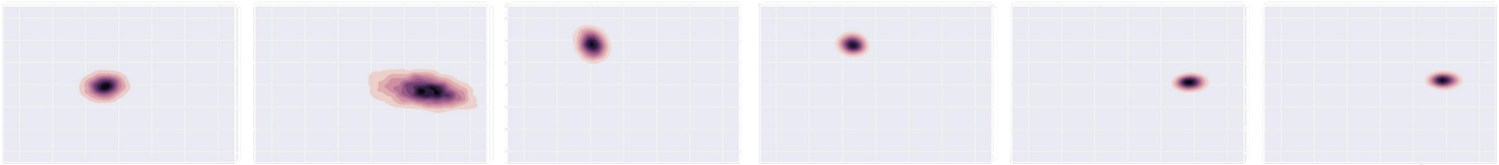
Target:

10k steps 20k steps 50K steps 100k steps

GAN training on Gaussian Mixture Data:

Target:





Step 0 Step 5k Step 10k Step 15k Step 20k Step 25k

pictures from [Metz et al ICLR’17]

(V) Finally Game Theory May Break

Real or Hallucinated?

Discriminator: DNN w/ parameters ω

Hallucinated Images (from generator)

Real Images (from training set)

Generator: DNN w/ parameters θ

Simple Randomness $Z_1, Z_2, \ldots, Z_N \sim N(0, I)$

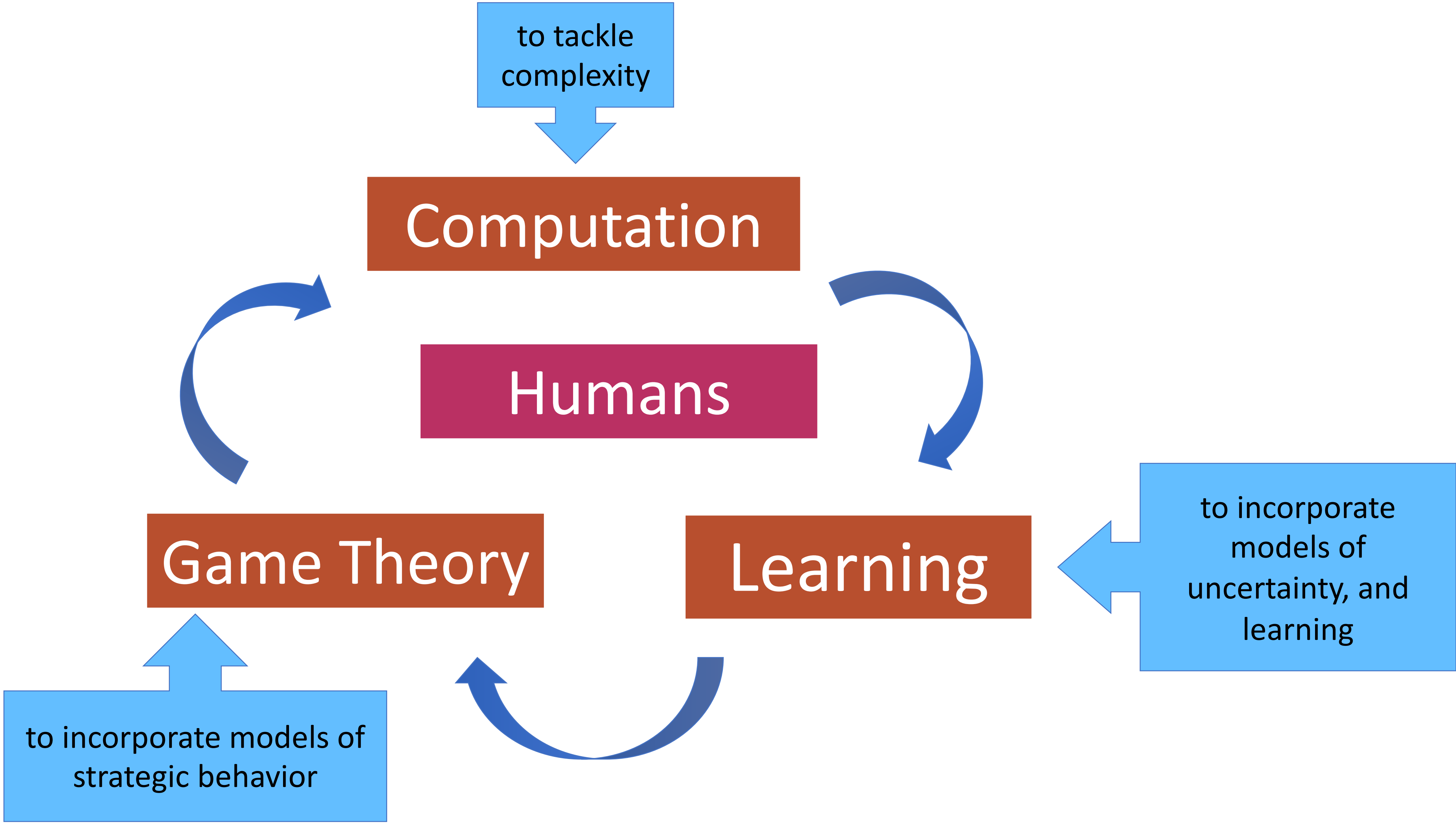
In applications involving DNNs, agents' losses are non-convex (a.k.a. their utilities are non-concave)

Without further structure, this is trouble for Game Theory, in that standard ways to solve the game are not applicable.

Summary so far…

- (I) Strategic Behavior does not emerge from standard training
- (II) Naively trained models can be manipulated
- (III) Combining agents that were trained in isolation can lead to undesirable (e.g. collusive) behavior
- (IV) The optimization workhorse of Deep Learning (namely gradient descent) struggles in multi-agent settings
- (V) Finally Game Theory (namely standard models and solutions) are inadequate to address non-convexity

This class: lay modern foundations of multi-agent learning



Today’s Menu

- Motivation
- **Administrivia**
- Course Overview

Administrivia

- Course website: https://web.mit.edu/~gfarina/www/6S890/
  - We will use the website as the public face of the course, and to post lecture notes and slides
  - Private discussions, questions, grading, etc. will be arranged on Canvas
- Lecturers: Costis Daskalakis, Gabriele Farina
  - TA? tbd
- Attendance: Everyone is welcome! If just auditing please register as a listener
- Office hours are flexible: email us to schedule

If Registered for Credit:

- Solve problem sets: 2-3 problems sets, two weeks to solve each (weight: 40%)
- Project: proposal due end of September (weight: 50%)
  - Project brainstorming class on 9/26
  - Project break on the week of 11/13-17
  - Project presentations starting 11/30
  - Project can be theoretical, practical, or a mix
  - We encourage creativity!
  - Feel free to run your ideas by us
  - Feel free to apply ideas from this class to your own area of interest
  - Project can be done in teams
- No exams

Any questions regarding logistics and projects?

Today’s Menu

- Motivation
- Administrivia
- **Course Overview**

Our goal in this class, revisited

How can we, and machines, systematically reason about the behavior, incentives, and outcomes of multiagent systems?

And as computer scientists, how can we teach machines to compute, predict, or learn such behavior?

The Concept of **Game**

Games are thought experiments to help us learn how to ***predict rational behavior*** in ***situations of conflict.***

Situation of conflict: Everybody's actions affect others.

Rational Behavior: The players want to maximize their own expected utility. No altruism, envy, masochism, or externalities (if my neighbor gets the money, he will buy louder stereo, so I will hurt a little myself...).

Predict: We want to know what happens in a game. Such predictions are called *solution concepts* (e.g., Nash equilibrium).

Situations Modeled as Games

- Recreational games
  - Rock paper scissors
  - Diplomacy
  - Poker
  - Go
  - …
- Non-recreational settings
  - Auctions
  - Markets
  - Logistics
  - Budget allocation (e.g., political campaigns)
  - Generative networks
  - Multi-robot interactions
  - Fraud detection systems
  - …

Scenic Tour

Part I: Normal-Form Games

These are "matrix" games

- Simultaneous actions
- Single move per player

Simple model but already captures several important aspects

	Rock	Paper	Scissors
Rock	0,0	-1,1	1,-1
Paper	1,-1	0,0	-1,1
Scissors	-1,1	1,-1	0,0

Rock-paper-scissors

	Deny (cooperate)	Confess (betray)
Deny (cooperate)	-1, -1	-3, 0
Confess (betray)	0, -3	-2, -2

Prisoner's dilemma

(-1 = 1 year in jail)

Despite their simplicity, normal-form games will provide the ground to start looking into the following key concepts in multiagent settings:

- Solution concepts and equilibria (Nash, maxmin, correlated, …)
- Learning from repeated play
  - Learning enables iteratively refining strategies to become stronger and stronger, and it has been a key component in all recent game AI breakthroughs
  - Local learning of each agent can often be connected to global notion of equilibrium
  - ≈ Mental model: “reinforcement learning but also works in nonstationary settings”
- Deep connection between equilibria and other important concepts in computer science

- After that, we will move on to notions of games that capture more interesting / real-world phenomena, especially:
  - Sequential moves, Imperfect information, nonconvexity

What is “rational play” for the agents?

Example: What should happen in prisoner’s dilemma?

- From blue player’s point of view, Confess ***dominates*** Deny, i.e. no matter what orange plays blue is better off by playing Confess.
- Likewise, From orange’s point of view, Confess ***dominates*** Deny.
- So the rational strategy for both is to play Confess.
- It is a ***dominant strategy equilibrium***.
- It is worse for both compared to (Deny, Deny)…

	Deny (cooperate)	Confess (betray)
Deny (cooperate)	-1, -1	-3, 0
Confess (betray)	0, -3	-2, -2

Benefit of dominant strategy equilibrium: requires no ”counterspeculation”

In general, counterspeculation cannot be avoided

- Dominant strategy equilibrium is the exception, not the norm: no strategies in general might be dominated

Furthermore, this example shows that **it is generally a bad idea for any player to stick to a single action**

Instead, players should play from a **distribution** over actions

General principle: randomization is generally required to play "optimally"

	Rock	Paper	Scissors
Rock	0,0	-1,1	1,-1
Paper	1,-1	0,0	-1,1
Scissors	-1,1	1,-1	0,0

Rock-paper-scissors

- Idea: Ideally, we would want a strategy that we are comfortable playing over and over
  - Even if the opponent learned about our distribution, and computed an optimal counterstrategy, we would not want to change our strategy

- In zero-sum games are lucky: if we settle on a strategy x, we know that an "expert" will maximize their own utility, which is the opposite of ours. So, if we play against an expert we can predict that our utility will be

$$v(x) = \min_y E_{a\sim x, b\sim y}[u(a,b)]$$

Our strategy

Strategy of opponent

Payoff associated with actions

Action of opponent

- We then want to select a strategy x with maximum return:

$$x^* \in \arg\max v(x)$$

Maxmin strategies

- The notion we just introduced is called a maxmin strategy
- Natural when playing against a strong player in a **two-player zero-sum** game
  - E.g. was used to beat top poker pros in Head's Up No Limit Hold'em
  - Does not require human data
  - Might not be the most natural choice against a weak opponent though

All in: CMU's poker computer busts humans over 20-day competition

January 30, 2017 10:49 PM

Darrell Sapp/Post-Gazette

Daniel McAulay of Scotland rubs his eyes Monday as he competes in the poker tournament against a Carnegie Mellon computer at the Rivers Casino on the North Shore.

By Sean D. Hamill / Pittsburgh Post-Gazette

The machines are taking over — at least in poker.

Though it had been a point conceded by the humans for the past week, Carnegie Mellon University's poker-playing computer, Libratus, on Monday finally, definitely and soundly defeated four of the world's best Heads-Up, No-Limit, Texas Hold 'em poker players by a resounding $1,766,250 in

Maxmin strategies

- Computation of maximin strategies is tractable
  - Convex optimization problem
  - In fact, linear -> We can use the simplex algorithm or interior point methods
- Even better: there exist very attractive learning algorithm
  - Start by playing the game
  - After every round, refine the strategy according to the learning algorithm
  - Repeat
  - Many algorithms known: hedge, optimistic multiplicative weights, regret matching, online gradient ascent, …
  - Extremely scalable and practical algorithm

Nash equilibria

- An assignment of maxmin strategies for each player in two-player zero-sum games forms a **Nash equilibrium**
  - That is, an assignment of independent strategies for each player, so that no player has any incentive to unilaterally deviate
  - Each player is best responding to each of the other players
  - Concept generalizes to any number of players and beyond zero-sum games

	Rock	Paper	Scissors
Rock	0,0	-1,1	1,-1
Paper	1,-1	0,0	-1,1
Scissors	-1,1	1,-1	0,0

Rock-paper-scissors

Since maxmin strategies can be computed in polynomial time, this means that a Nash equilibrium in two-player zero-sum games always exists and can be computed in polynomial time

Beyond Zero-Sum?

Nash's Theorem

[John Nash '50]: A Nash equilibrium exists in every finite game.

Deep influence in Economics, enabling other existence results.

Proof highly non-constructive (uses Brouwer's fixed point thm)

No simpler proof has been discovered

[Daskalakis-Goldberg-Papadimitriou'06]: no simpler proof exists

i.e. **Nash Equilibrium** $\Longleftrightarrow$ **Brouwer's Fixed Point Theorem**

- In practice, people have been successful applying learning methods beyond two-player zero-sum settings and achieving human or even superhuman performance
  - Example: six-player poker was solved this way

- In general-sum games, learning dynamics also provably converge to relaxations of the Nash equilibrium (e.g., correlated and coarse-correlated equilibria), which are interesting on their own

Part I: Normal-form games

2	9/12	**Setting and equilibria: Nash equilibrium** Definition of normal-form games. Properties of Nash equilibria. Nash equilibria in two-player games as linear and linear complementarity problems
3	9/14	**Setting and equilibria: Correlated equilibrium** Definition of Correlated and coarse correlated equilibria. Their relationships with Nash equilibria in two-player zero-sum games. Linear programming formulations
4	9/19	**Learning in games: Foundations** Regret and hindsight rationality. Phi-regret minimization and special cases. Connections with equilibrium computation and saddle-point optimization
5	9/21	**Learning in games: Algorithms** Regret matching, regret matching plus, hedge, FTRL and OMD

Part Ib: Complexity of equilibrium

7	9/28	**Nash equilibrium and PPAD complexity** Sperner's lemma, Brouwer's fixed point, and the PPAD complexity class. Nash's proof
8	10/3	**PPAD-completeness of Nash equilibria, and open problems.**

Parts II and III

Most interactions do not look like a normal-form games

Players can often make more than one move, and they often have imperfect information

Part II: Markov (aka Stochastic) Games

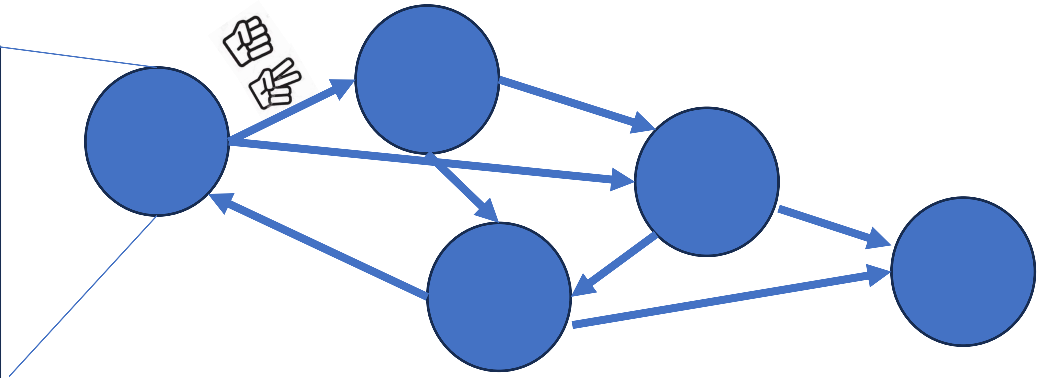
Markov games: many "normal-form" stage games played on a graph.
(≈ think: "MDP of normal-form games")

Can be generally solved via **backward induction** (bottom up induction)

Surprising complexity of stationary equilibria

- Example: diplomacy

	0	-1	+1
	+1	0	-1
	-1	+1	0



Part II: Stochastic games

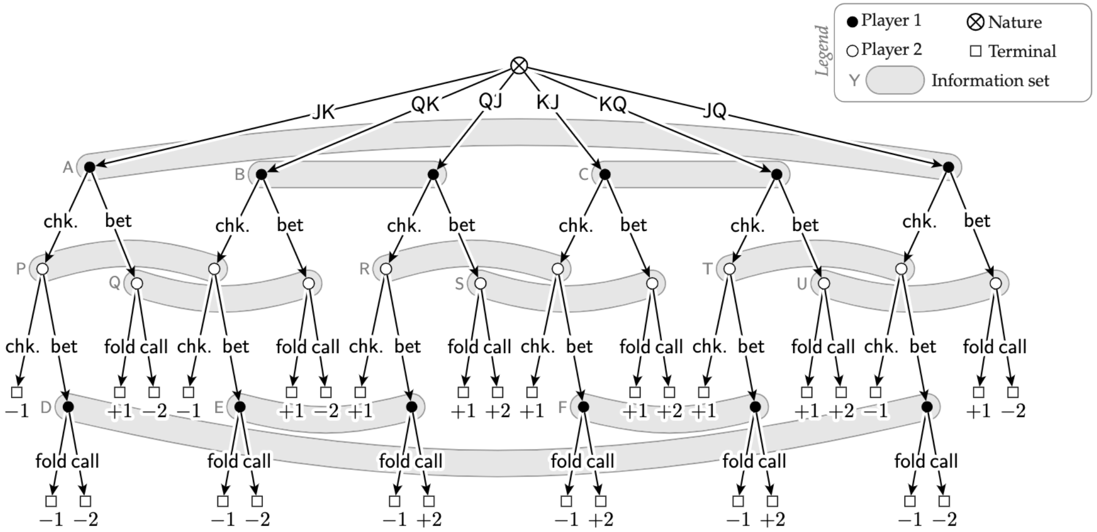
9	10/5	Stochastic games Minimax theorem, and existence of equilibrium. Stationary Markov Nash equilibria.
10	10/12	Computation of equilibria in stochastic games Finite horizon vs infinite horizon (discounted) setting, the role of nonstationarity, and backward induction.
11	10/17	Minimax stationary Markov learning
12	10/19	Complexity of correlated equilibria in general-sum stochastic games

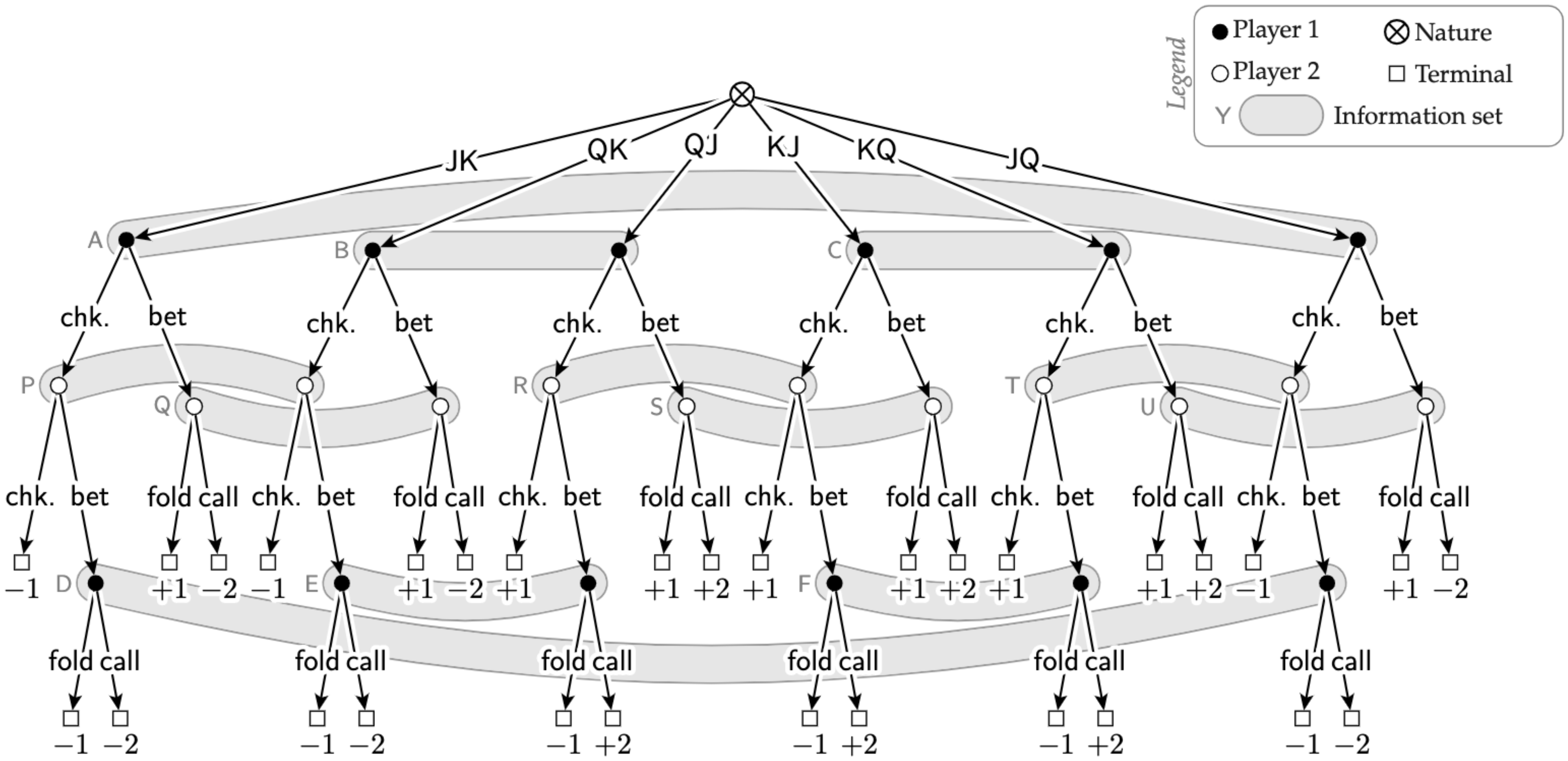
Part III: Imperfect-Information Extensive-Form Games

While stochastic games capture *sequential* moves, they do not address **imperfect information**

Imperfect information extensive-form games: ≈ “stochastic games + imperfect information + tree structure (instead of general graph)”

- Example: poker




Legend
Player 1
Nature
Player 2
Terminal
Y
Information set
JK
QK
QJ
KJ
KQ
JQ
A
B
C
D
E
F
P
Q
R
S
T
U
chk.
bet
fold
call
−1
+1
−2
+2

Difficulties with Imperfect Information

- Compared to normal-form games, imperfect-information extensive-form games bring many conceptual challenges

1. The number of (deterministic) strategies grows exponentially in the game tree

2. Imperfect information makes backward induction and local reasoning not viable

   Think about poker: need to reason about **misdirection.** *General principle: you need to think about what the opponents don't know about you and leverage that to your advantage*

3. Other players have control over what part of the game tree is visited/explored

- Nonetheless: many positive results
  - In fact, we live in a world where machines bluff at poker better than humans

- As an example of a positive result, we will show that learning can be carried out efficiently in imperfect-information games

PART III: IMPERFECT-INFORMATION GAMES

13	10/24	Foundations of imperfect-information extensive-form games Complete versus imperfect information. Kuhn's theorem. Normal-form and sequence-form strategies. Similarities and differences with normal-form games.
14	10/26	Counterfactual regret minimization Construction and proof of learning algorithms for extensive-form games.
15	10/31	Extensive-form correlated equilibria Uncoupled computation of extensive-form correlated equilibria in multi-player general-sum games via learning
16	11/2	Optimal equilibria and team coordination Correlation plans and von Stengel-Forges's theorem. Connections between correlation and teams.
17	11/7	Nash equilibrium refinements and sequential rationality Trembling-hand equilibrium refinements: quasi-perfect equilibrium (QPE) and extensive-form perfect equilibrium (EFPE). Relationships among refinements. Computational complexity and techniques
18	11/9	Practice of solving large games

By the end of this part…

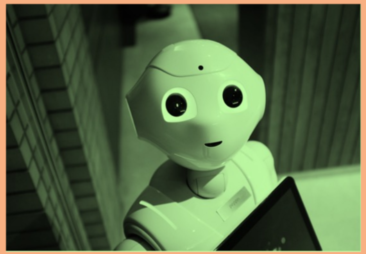
- By the end of this part, you should have acquired:
  - A **language** to think about and describe different equilibrium points of multiagent interactions (Nash equilibrium, maxmin strategies, correlated equilibria, …)
  - An appreciation for what is **computationally tractable** in every case, and what only in special cases
  - The ability to **implement learning dynamics** to progressively refine strategies, including in imperfect-information domains
  - A general understanding of what techniques are used to push scalability, and what major areas of investigation remain **underexplored**

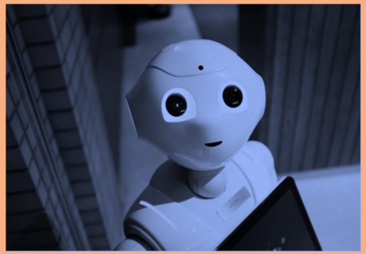
Part IV: Nonconvexity (preview)

21	11/21	Aspects of nonconvex-nonconcave games (TBD)
22	11/28	Aspects of nonconvex-nonconcave games (TBD)

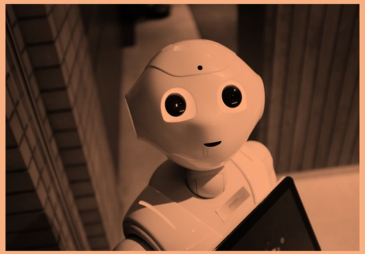
Part IV: Nonconcave games

Setting:


action: $x_1 \in \mathcal{X}_1 \subset \mathbb{R}^{d_1}$
goal: max $u_1(x_1, \dots, x_n)$


action: $x_2 \in \mathcal{X}_2 \subset \mathbb{R}^{d_2}$
goal: max $u_2(x_1, \dots, x_n)$

…


action: $x_n \in \mathcal{X}_n \subset \mathbb{R}^{d_n}$
goal: max $u_n(x_1, \dots, x_n)$

[often: global constraints $(x_1, x_2, \dots, x_n) \in \mathcal{S} \subseteq \times_i \mathcal{X}_i$]
[often: u_i is Lipschitz and smooth (i.e. Lipschitz gradient) a.e.]

Emerging applications in ***Machine Learning*** involve multiple agents who:

- choose high-dimensional strategies $x_i \in \mathcal{X}_i \subset \mathbb{R}^{d_i}$
- maximize utility functions $u_i(x_i\,; x_{-i})$ that are typically ***nonconcave*** in their own strategy (a.k.a. minimize loss functions that are **nonconvex** in their own strategy)

Issue: Game Theory is fragile when utilities are nonconcave

- in particular, Nash equilibrium (and other types of equilibrium) may not exist
- so what is even our recommendation about reasonable optimization targets in the multi-agent setting?

And finally…

PROJECT PRESENTATIONS

23	11/30	Projects
24	12/5	Projects
25	12/7	Projects