Game Theory and Machine Learning

Constantinos (a.k.a. “Costis”) Daskalakis
CSAIL and EECS, MIT

Future of AI: learning + strategic reasoning

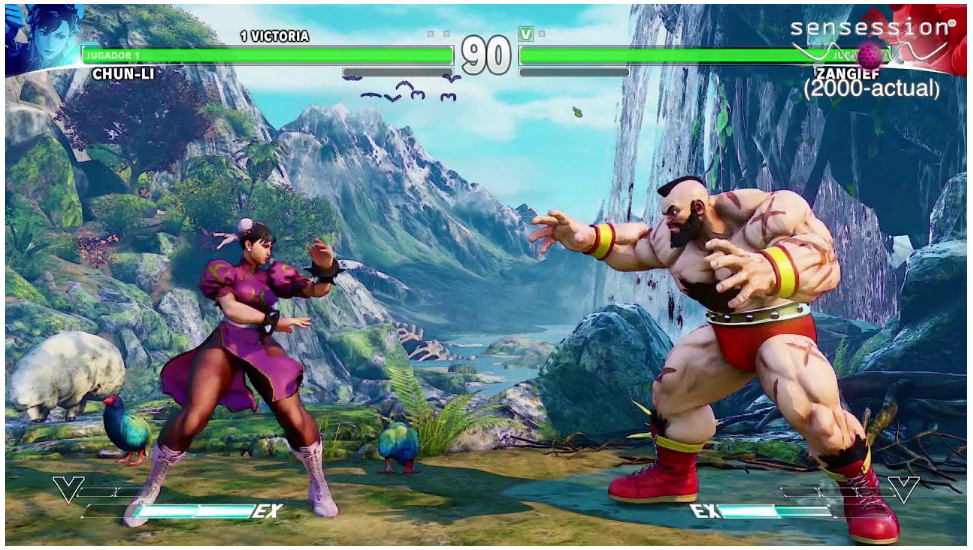
Example 1: Platform Design

- *How to design a better Uber platform?*
- Collect data about behavior of riders and drivers
- Tune knobs to get better welfare/revenue
- **Need behavioral model of rider and driver behavior!**
- **No model $\Leftrightarrow$ No counterfactuals**

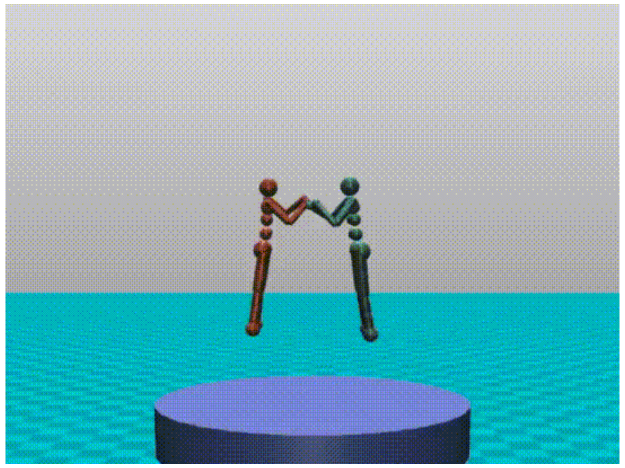
Example 2: Recommender Systems

- *How to design restaurant recommender system?*
- Restaurant seats $\neq$ Digital goods
- Not the Netflix challenge problem
- **Need to respect supply $\Rightarrow$ Combined Learning + Market Design**

Example 3: Humanoid vs Spider



vs



- **Insufficient to learn a *symmetric* optimal policy**
- **A bone fide multi-agent RL problem**

A dictum

Minimization is to current AI

what min-max optimization is to future AI

(or, more broadly, multi-agent equilibrium learning)

- **Minimization:** AI agent is learning/decision-making in a stationary environment
- **Min-max optimization/equilibrium learning:** AI agent is learning and decision-making in a *changing* environment due to…
    - the presence of another/multiple other learning agents with conflicting/aligned interests
    - noise/adversaries poisoning or corrupting the data
    - the need to enforce constraints on the learning outcome, e.g. GANs, private release of data etc.

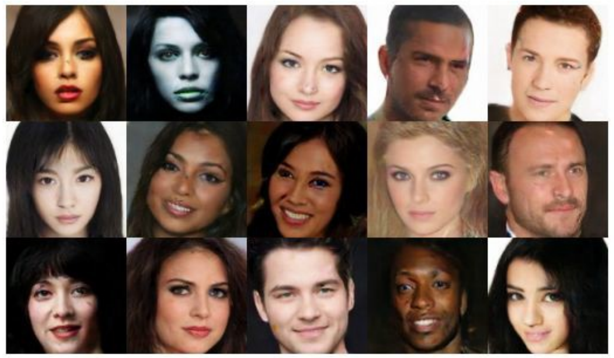
Our focus: min-max optimization

Solve: $\inf_{\theta} \sup_{w} f(\theta, w)$
where θ, w high-dimensional
& potentially constrained

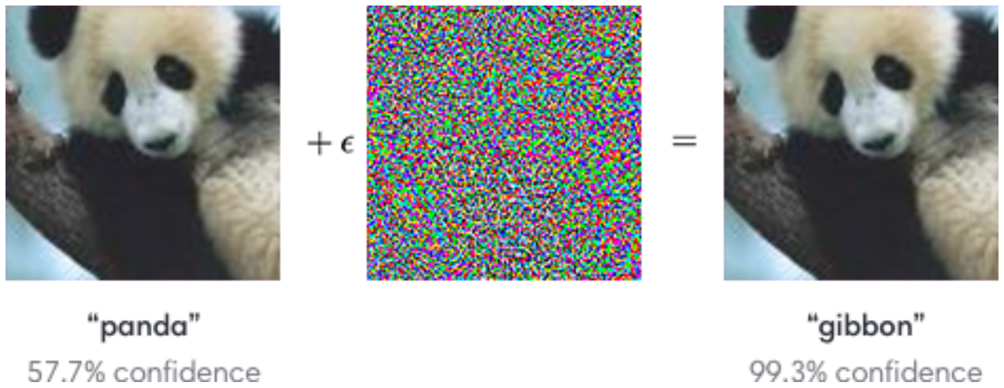
- **Applications:** Mathematics, Optimization, Game Theory,…
  [von Neumann 1928, Dantzig '47, Brown'50, Robinson'51, Blackwell'56,…]

- **Best-Case Scenario:** f is convex in θ, concave in w

- **Many Deep Learning Applications:** GANs, training against adversarial attacks, …
  – *exacerbate the importance of first-order methods, non convex-concave objectives*



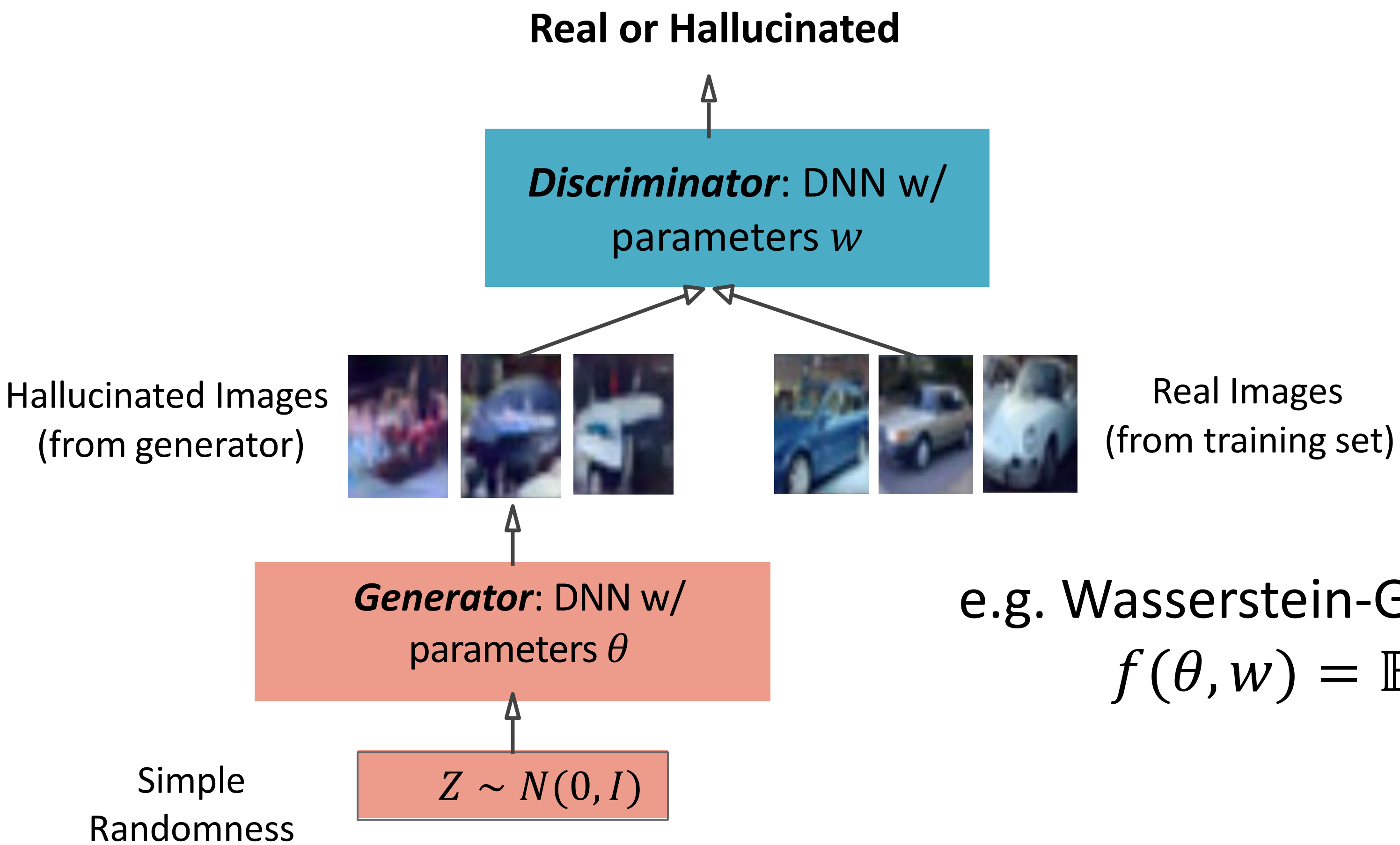
[BEGAN. Bertholet et al. 2017]



[Goodefllow-Shlens-Szegedy 2014]

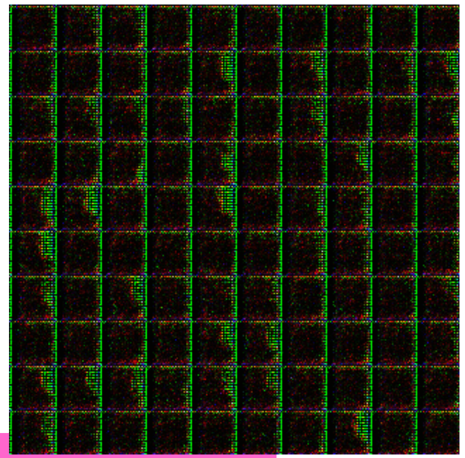
Generative Adversarial Networks (GANs)

[Goodfellow et al. NeurIPS'14]



$$\inf_{\theta} \sup_{w} \underbrace{\boldsymbol{f(\theta, w)}}$$

expresses how well Discriminator distinguishes true vs generated images



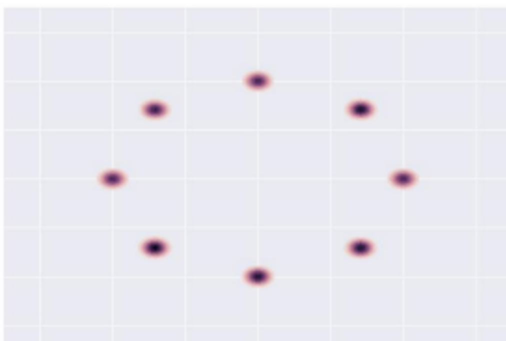
e.g. Wasserstein-GANs **[Arjovsky et al. 2017]**:

$$f(\theta, w) = \mathbb{E}_{X \sim p_{real}}[D_w(\underbrace{X}_{\text{real sample}})] - \mathbb{E}_{Z \sim N(0,I)}[D_w(\underbrace{G_\theta(Z)}_{\text{hallucinated sample}})]$$

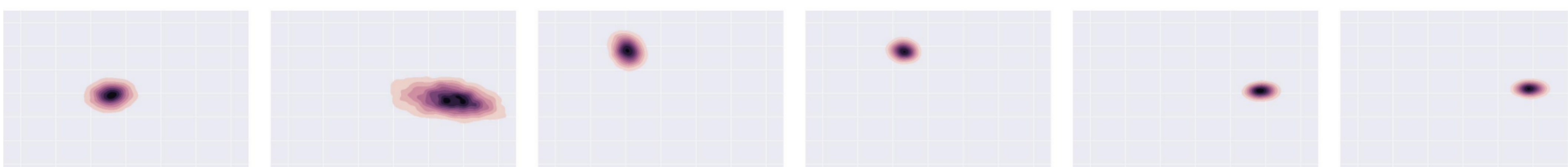
- θ, w: high-dimensional
  ⇝ solve game by having min (resp. max) player run variant of online gradient descent (resp. ascent)

- **major challenges:**
  – training oscillations
  – generated & real distributions high-dimensional ⇝ statistical challenges
    - evidence that GANs don't learn, e.g. **[Arora et al'2017]**

Training Oscillations:
Gaussian Mixture Data



Target Distribution: Mixture of 8 Gaussians on a circle



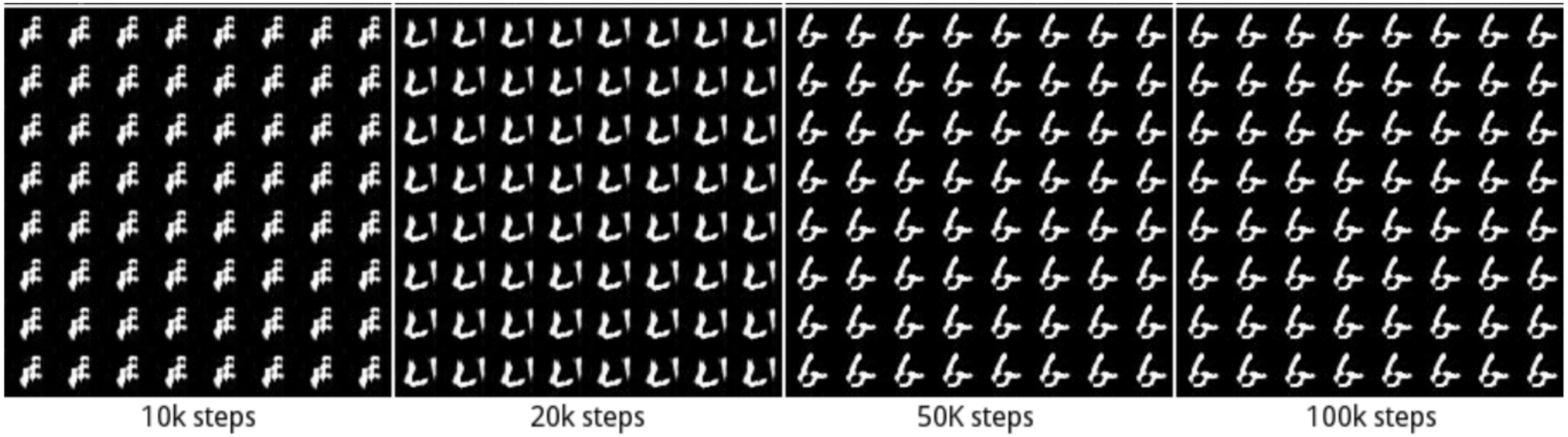
Step 0 Step 5k Step 10k Step 15k Step 20k Step 25k

Output Distribution of standard GAN, trained via gradient descent/ascent dynamics: *cycling through modes at different steps of training*

from **[Metz et al ICLR'17]**

Training Oscillations:
Handwritten Digits

Target Distribution: MNIST



Output Distribution of standard GAN, trained via gradient descent/ascent dynamics *cycling through "proto-digits" at different steps of training*

from **[Metz et al ICLR'17]**

Training Oscillations:
even for Gaussian data/bilinear objectives!

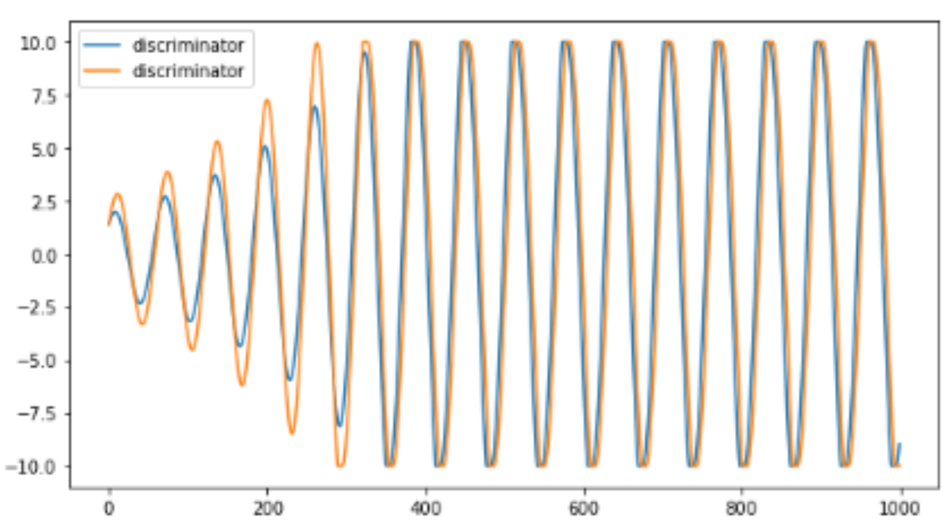
- **True distribution:** isotropic Normal distribution, namely $X \sim \mathcal{N}\left(\begin{bmatrix}3\\4\end{bmatrix}, I_{2\times2}\right)$
- **Generator architecture**: $G_{\boldsymbol{\theta}}(Z) = Z + \boldsymbol{\theta}$ (adds input Z to internal params)

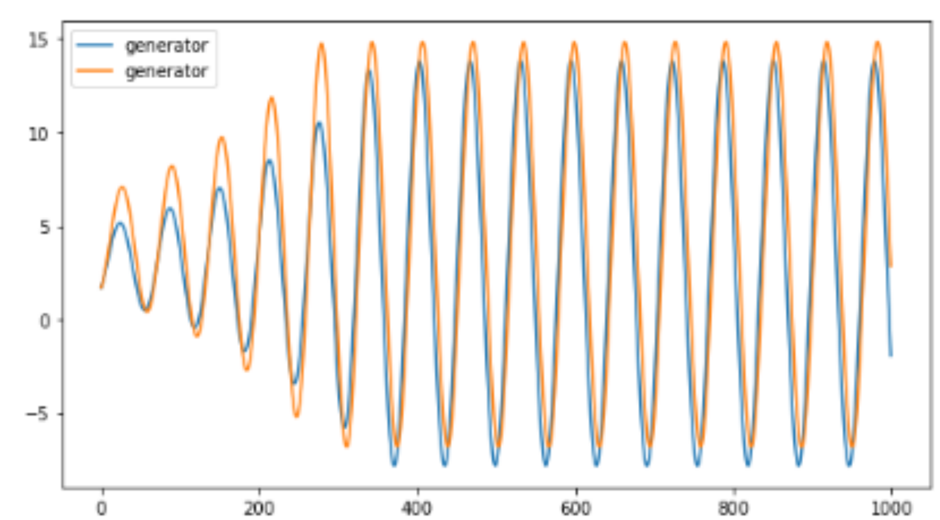
Z, θ, w: 2-dimensional

- **Discriminator architecture**: $D_{\boldsymbol{w}}(\cdot) = \langle \boldsymbol{w}, \cdot \rangle$ (linear projection)
- **W-GAN objective:** $\min_{\boldsymbol{\theta}} \max_{\boldsymbol{w}} \mathbb{E}_X[D_{\boldsymbol{w}}(X)] - \mathbb{E}_Z[D_{\boldsymbol{w}}(G_{\boldsymbol{\theta}}(Z))]$

$$= \min_{\boldsymbol{\theta}} \max_{\boldsymbol{w}} \boldsymbol{w}^{\mathrm{T}} \cdot \left(\begin{bmatrix}3\\4\end{bmatrix} - \boldsymbol{\theta}\right)$$

convex-concave function



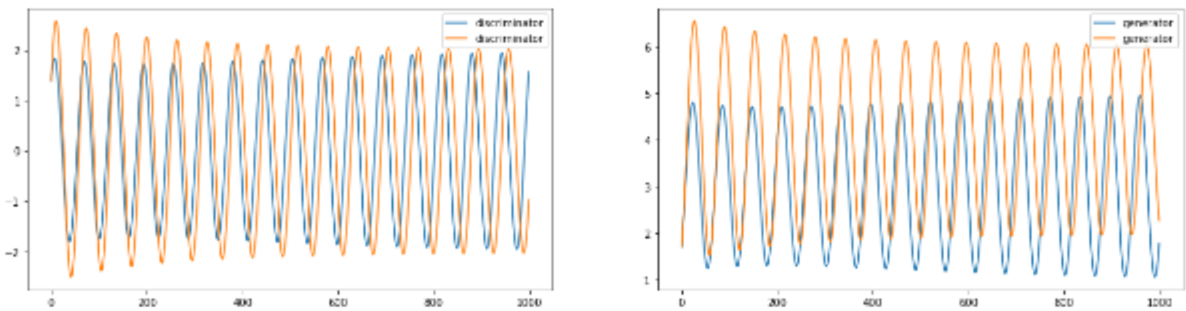


Gradient Descent Dynamics

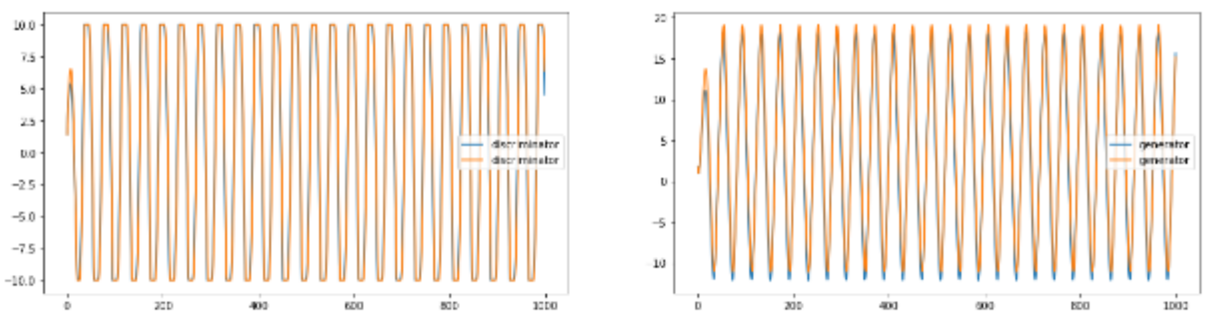
from **[Daskalakis, Ilyas, Syrgkanis, Zeng ICLR'18]**

Training Oscillations:

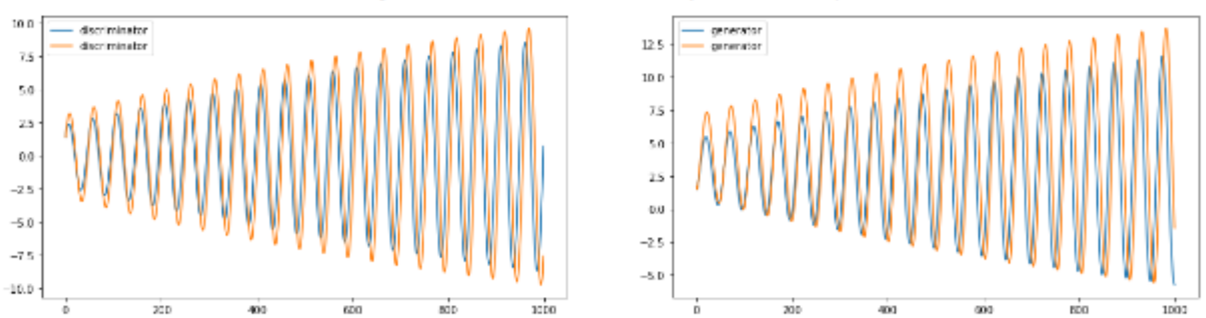
persistence for variants of Online Gradient Descent



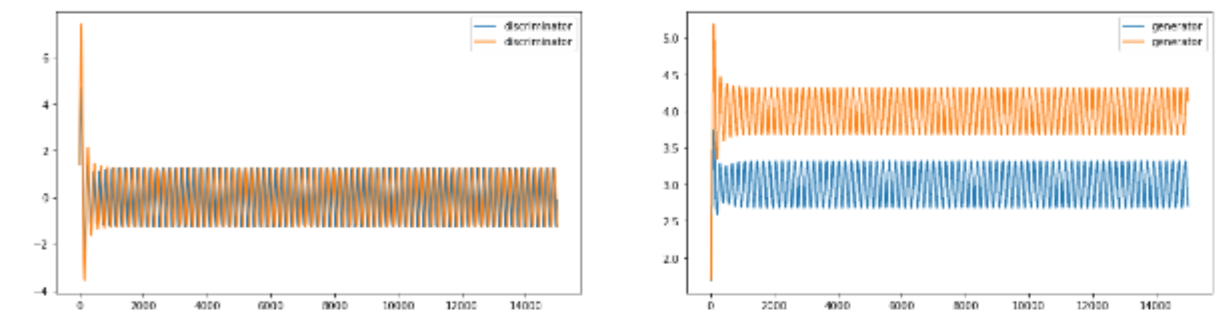
(a) GD dynamics with a gradient penalty added to the loss. $\eta = 0.1$ and $\lambda = 0.1$.



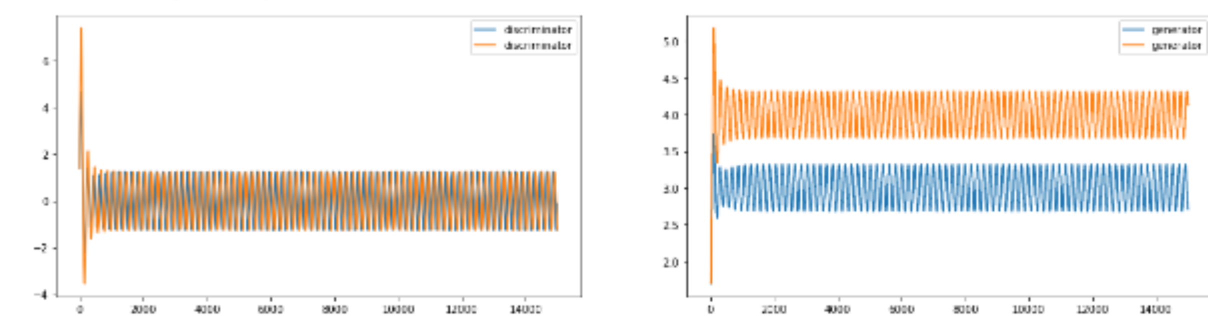
(b) GD dynamics with momentum. $\eta = 0.1$ and $\gamma = 0.5$.



(c) GD dynamics with momentum and gradient penalty. $\eta = .1$, $\gamma = 0.2$ and $\lambda = 0.1$.



(d) GD dynamics with momentum and gradient penalty, training generator every 15 training iterations of the discriminator. $\eta = .1$, $\gamma = 0.2$ and $\lambda = 0.1$.



(e) GD dynamics with Nesterov momentum and gradient penalty, training generator every 15 training iterations of the discriminator. $\eta = .1$, $\gamma = 0.2$ and $\lambda = 0.1$.

from **[Daskalakis, Ilyas, Syrgkanis, Zeng ICLR'18]**

What gives?

- Training oscillations:
  - even for convex-concave objectives
  - even when the function is perfectly known
  - even when there is no state
- So good luck when:
  - objective is nonconvex-nonconcave
  - the function needs to be learned
  - agents’ actions change the state

Menu

- **Motivation**
- Part 1: convex-concave objectives
- Part 2: nonconvex-nonconcave objectives

Menu

- **Motivation**
- Part 1: convex-concave objectives
    - Background: min-max in Game Theory and Learning
    - Removing oscillations via negative momentum
- Part 2: nonconvex-nonconcave objectives
    - Some initial musings
    - Computational complexity
    - Convergence under structure: multi-agent RL
- Some other topics that will show up:
    - PPAD-completeness, multi-agent learning, GANs

Menu

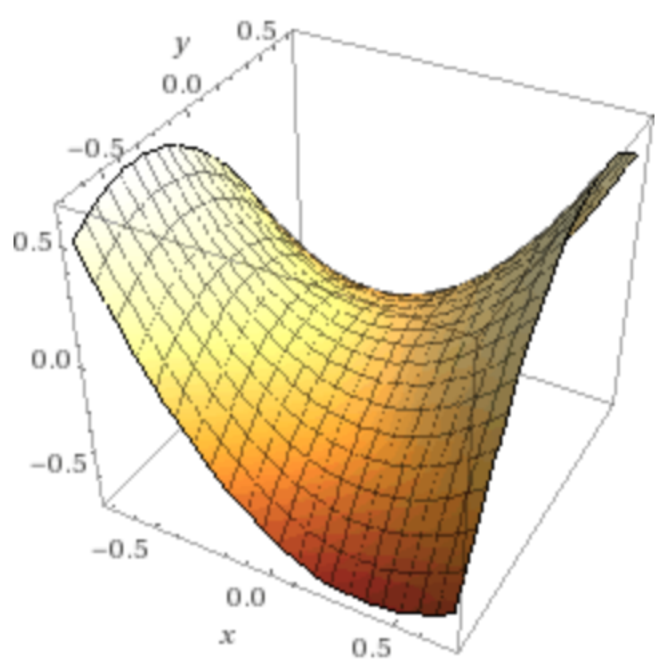
- **Motivation**
- **Part 1: convex-concave objectives**
  - **Background: min-max in Game Theory and Learning**
  - Removing oscillations via negative momentum
- Part 2: nonconvex-nonconcave objectives
  - Some initial musings
  - Computational complexity
  - Convergence under structure: multi-agent RL

The Min-Max Theorem

- **[von Neumann 1928]:** If $X \subset \mathbb{R}^n, Y \subset \mathbb{R}^m$ are compact and convex, and $f: X \times Y \to \mathbb{R}$ is continuous and convex-concave (i.e. $f(x,y)$ is convex in x for all y and is concave in y for all x), then

$$\min_{x \in X} \max_{y \in Y} f(x,y) = \max_{y \in Y} \min_{x \in X} f(x,y)$$

- Min-max optimal point (x,y) is essentially unique (unique if f is strictly convex-concave, o.w. a convex set of solutions); value always unique
- E.g. $f(x,y) = x^2 - y^2 + x \cdot y$



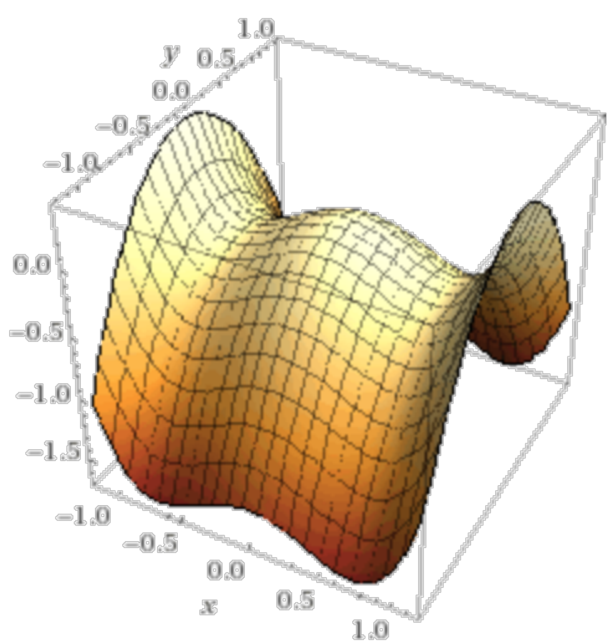
The Min-Max Theorem

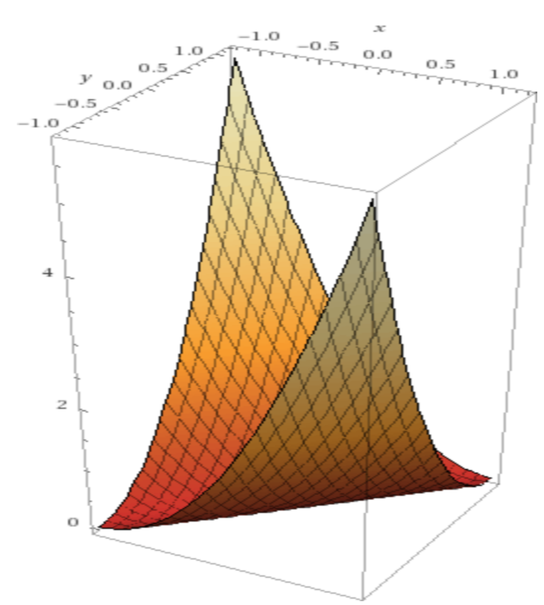
- **[von Neumann 1928]:** If $X \subset \mathbb{R}^n, Y \subset \mathbb{R}^m$ are compact and convex, and $f: X \times Y \to \mathbb{R}$ is continuous and convex-concave (i.e. $f(x,y)$ is convex in x for all y and is concave in y for all x), then

$$\min_{x \in X} \max_{y \in Y} f(x,y) = \max_{y \in Y} \min_{x \in X} f(x,y)$$

- Min-max optimal point (x,y) is essentially unique (unique if f is strictly convex-concave, o.w. a convex set of solutions); value always unique

- If f is not convex-concave all bets are off
  - $f(x,y) = x^4 - x^2 - y^2$ $\qquad$ $f(x,y) = (x-y)^2$





The Min-Max Theorem

- **[von Neumann 1928]:** If $X \subset \mathbb{R}^n, Y \subset \mathbb{R}^m$ are compact and convex, and $f: X \times Y \rightarrow \mathbb{R}$ is continuous and convex-concave (i.e. $f(x, y)$ is convex in x for all y and is concave in y for all x), then
$$\min_{x \in X} \max_{y \in Y} f(x, y) = \max_{y \in Y} \min_{x \in X} f(x, y)$$
- Min-max optimal point (x, y) is essentially unique (unique if f is strictly convex-concave, o.w. a convex set of solutions); value always unique
- Min-max points = equilibria of zero-sum game where min player pays max player $f(x, y)$
- von Neumann: "*As far as I can see, there could be no theory of games … without that theorem … I thought there was nothing worth publishing until the Minimax Theorem was proved*“
- When f is bilinear, i.e. $f(x, y) = x^{\mathrm{T}} A y + b^T x + c^T y$ and *X*, *Y* polytopes
  - **[von Neumann-Dantzig 1947, Adler IJGT’13]:** Min-max thm ⇔ strong LP duality
  - min-max solutions can be found w/ Linear Programming and vice versa
  - mathematical structure arguably crucial in recent success of computers beating humans in two-player zero-sum games (chess, poker, go)

The Min-Max Theorem

- **[Brown RAND'49]:** proposes *fictitious play* as a method to solve bilinear case on product of simplices:
$$\min_{x \in \Delta_n} \max_{y \in \Delta_m} x^T A y = \max_{y \in \Delta_m} \min_{x \in \Delta_n} x^T A y$$
- **Fictitious play:** $(x_t, y_t)_{t=1,\ldots}$ where for all t:
  - $x_t \in \operatorname{argmin} \sum_{\tau<t} f(\cdot, y_\tau)$
  - $y_t \in \operatorname{argmax} \sum_{\tau<t} f(x_\tau, \cdot)$
- **[Robinson Annals of Math'51]:** shows fictitious play converges in bilinear case in an average sense: $\frac{1}{t}\sum_\tau f(x_\tau, y_\tau) \to \min\max f(x, y)$
- **[Karlin'59]:** conjectures convergence rate is $\sim 1/\sqrt{t}$
- **[Daskalakis-Pan FOCS'14]:** actually exponentially slow $\left(\sim 1/t^{1/m+n}\right)$
- Faster methods?

Min-Max and No-Regret Learning

Online Learning: Every day $t = 1, \dots, T$:
- learner chooses $z_t \in Z$
- world chooses L-Lipschitz convex f'n $\ell_t(\cdot)$
- learner loses $\ell_t(z_t)$; "observes" $\ell_t(\cdot)$

Goal:
$\frac{1}{T}\sum_t \ell_t(z_t) - \frac{1}{T}\min \sum_t \ell_t(\cdot) \to 0$

- **Def:** $R: Z \to \mathbb{R}$ is α-strongly convex w.r.t. norm $\|\cdot\|$ iff for all $z, z_0 \in Z$:

$$R(z) \geq R(z_0) + \nabla R(z_0)^T \cdot (z - z_0) + \frac{\alpha}{2}\|z - z_0\|^2$$

e.g.1: $R(z) = ||z||^2/2$
e.g.2: $R(z) = -H(z), z \in \Delta$

- **Follow-The-Regularized-Leader (FTRL):** On day t choose:

$$z_t \in \arg\min \left[\sum_{\tau<t} \ell_\tau(\cdot) + \frac{1}{\eta} \cdot R(\cdot)\right]$$

for some parameter η, and some strongly convex regularization function $R(\cdot)$

- **Theorem:** Suppose, $\forall t = 1, \dots, T$, ℓ_t is convex and L-Lipschitz w.r.t. some norm $\|\cdot\|$, and R is 1-strongly convex w.r.t. $\|\cdot\|$. Then FTRL with parameter η satisfies:

$$\frac{1}{T}\sum_t \ell_t(z_t) - \frac{1}{T}\min\sum_t \ell_t(\cdot) \leq \frac{1}{T}\frac{\max_Z R(\cdot) - \min_Z R(\cdot)}{\eta} + \eta \cdot L^2$$

- set $\eta = L^{-1} \cdot \sqrt{(\max R(\cdot) - \min R(\cdot))/T}$ to get average regret $L \cdot \sqrt{(\max R(\cdot) - \min R(\cdot))/T}$.

Min-Max and No-Regret Learning

Online Learning: Every day $t = 1, \dots, T$:
- learner chooses $z_t \in Z$
- world chooses L-Lipschitz convex f'n $\ell_t(\cdot)$
- learner loses $\ell_t(z_t)$; "observes" $\ell_t(\cdot)$

Goal:
$\frac{1}{T}\sum_t \ell_t(z_t) - \frac{1}{T}\min \sum_t \ell_t(\cdot) \to 0$

- **Def:** $R: Z \to \mathbb{R}$ is α-strongly convex w.r.t. norm $\|\cdot\|$ iff for all $z, z_0 \in Z$:

$$R(z) \geq R(z_0) + \nabla R(z_0)^T \cdot (z - z_0) + \frac{\alpha}{2}\|z - z_0\|^2$$

e.g.1: $R(z) = ||z||^2/2$
e.g.2: $R(z) = -H(z), z \in \Delta$

- **Follow-The-Regularized-Leader (FTRL):** On day t choose:

$$z_t \in \arg\min \left[\sum_{\tau<t} \ell_\tau(\cdot) + \frac{1}{\eta} \cdot R(\cdot)\right]$$

for some parameter η, and some strongly convex regularization function $R(\cdot)$

- **FTRL instantiations:**
  - FTRL w/ ℓ_2^2-regularizer $\approx$ online gradient descent
  - FTRL on simplex w/ negative entropy regularizer $=$ multiplicative-weights-update method

Min-Max and No-Regret Learning

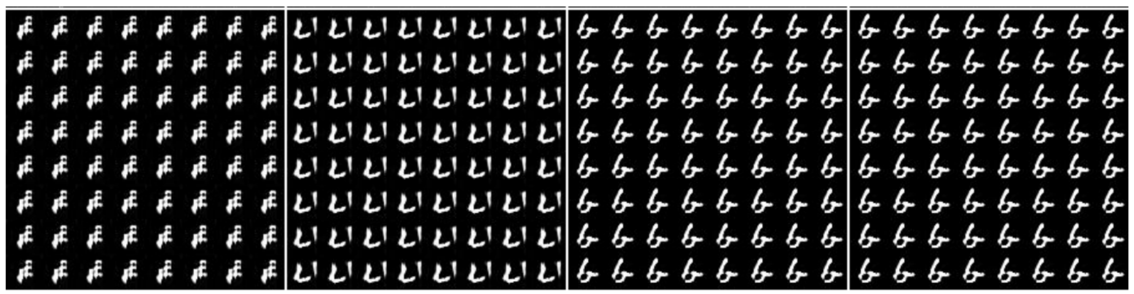
- Suppose $f(x, y)$ continuous&convex-concave, and both x and y players update strategies using no-regret learning algorithm, with average regret rate $\varepsilon(T) \to 0$

I.e.
  - the x-player chooses x_t using no-regret learning algorithm on loss sequence $f(\cdot, y_t)$
  - the y-player chooses y_t using no-regret learning algorithm on loss sequence $-f(x_t, \cdot)$

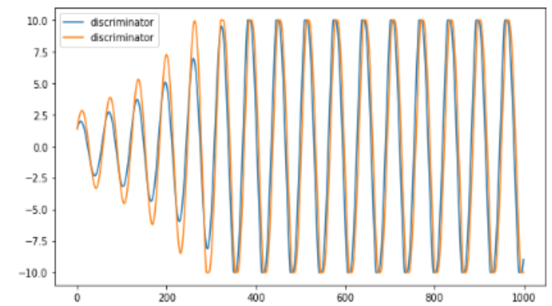
- **Theorem:** If x and y players update strategies using no-regret learning, then:
  - $\frac{1}{T}\sum_{t=1}^{T} f(x_t, y_t) = \min_x \max_y f(x, y) \pm \varepsilon(T)$
  - Moreover, ***the average strategies*** $\bar{x}_T = \frac{1}{T}\sum_t x_t$ and $\bar{y}_T = \frac{1}{T}\sum_t y_t$ comprise an $\varepsilon(T)$-approximate Nash equilibrium, i.e.
    - $f(\bar{x}_T, \bar{y}_T) \leq \min f(\cdot, \bar{y}_T) + \varepsilon(T)$
    - $f(\bar{x}_T, \bar{y}_T) \geq \max f(\bar{x}_T, \cdot) - \varepsilon(T)$

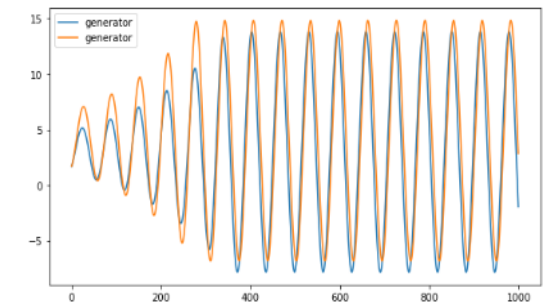
- **Can we show last-iterate convergence of no-regret learning methods, i.e. convergence of (x_t, y_t) to equilibrium?**
  - **[Mertikopoulos-Papadimitriou-Piliouras SODA'18]:** No for (continuous time) FTRL methods

Training Oscillations

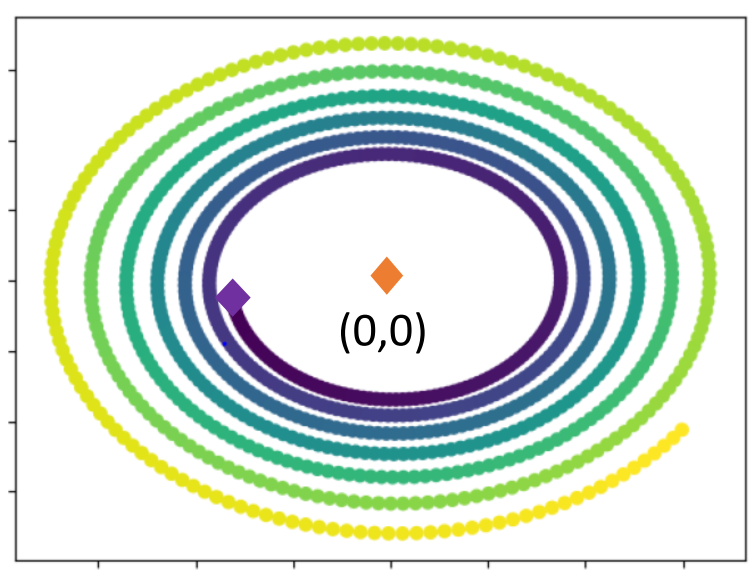


10k steps | 20k steps | 50K steps | 100k steps





the simplest example: $\min_{x} \max_{y} (x \cdot y)$,



Gradient Descent/Ascent (GDA)

$$x_{t+1} = x_t - \eta \cdot \nabla_x f(x_t, y_t) = x_t - \eta \cdot y_t$$
$$y_{t+1} = y_t + \eta \cdot \nabla_y f(x_t, y_t) = y_t + \eta \cdot x_t$$

◆ : start

◆ : min-max equilibrium

Menu

- **Motivation**
- **Part 1: convex-concave objectives**
  - **Background: min-max in Game Theory and Learning**
  - Removing oscillations via negative momentum
- Part 2: nonconvex-nonconcave objectives
  - Some initial musings
  - Computational complexity
  - Convergence under structure: multi-agent RL

Menu

- **Motivation**
- **Part 1: convex-concave objectives**
  - **Background: min-max in Game Theory and Learning**
  - **Removing oscillations via negative momentum**
- Part 2: nonconvex-nonconcave objectives
  - Some initial musings
  - Computational complexity
  - Convergence under structure: multi-agent RL

Gradient Descent w/ Negative Momentum

- Optimistic Gradient Descent **[Popov 1980]**:

$$\forall t:\ x_{t+1} = x_t - \eta \cdot \nabla f(x_t) + \boldsymbol{\eta/2 \cdot \nabla f(x_{t-1})}$$

- **Interpretation:** undo today, some of yesterday's gradient; ie negative momentum

- Optimistic Gradient Descent

  = **Optimistic** FTRL w/ ℓ_2^2-regularization **[Rakhlin-Sridharan COLT'13, Syrgkanis et al. NeurIPS'15]**

  ≈ **extra-gradient** method **[Korpelevich'76]**

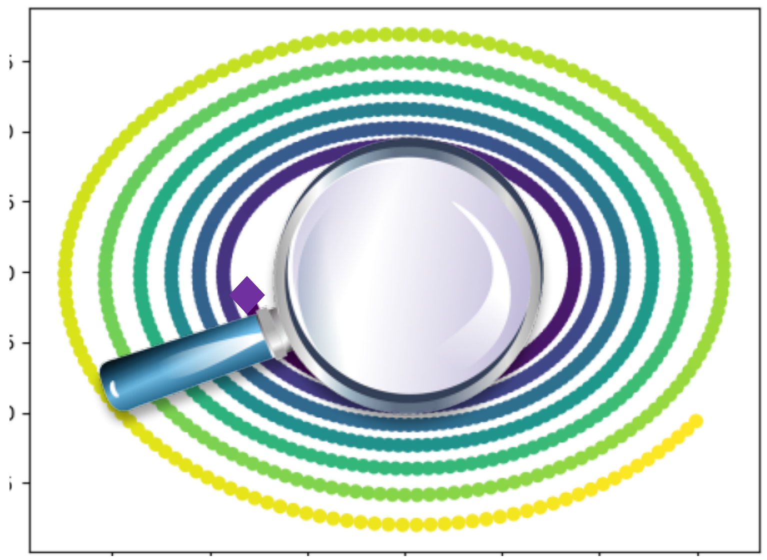
$$\forall t:\ x_{t+1/2} = x_t - \eta \cdot \nabla f(x_t)$$
$$x_{t+1} = x_t - \eta \cdot \nabla f\left(x_{t+1/2}\right)$$

  $=$ **mirror prox** method w/ ℓ_2^2-regularization **[Nemirovski'04]**

- **Does it help in min-max optimization?**

Negative Momentum: why might it help?

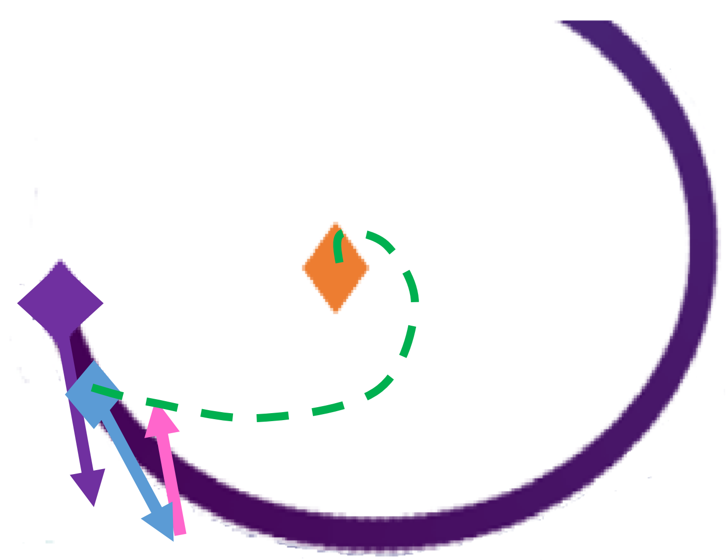
- E.g. $f(x, y) = x \cdot y$



$$\boxed{\begin{aligned} x_{t+1} &= x_t - \eta \cdot \nabla_x f(x_t, y_t) \\ y_{t+1} &= y_t + \eta \cdot \nabla_y f(x_t, y_t) \end{aligned}}$$

◆ : start

◆ : min-max equilibrium



$$\boxed{\begin{aligned} x_{t+1} &= x_t - \eta \cdot \nabla_x f(x_t, y_t) \\ &\quad + \boldsymbol{\eta/2 \cdot \nabla_x f(x_{t-1}, y_{t-1})} \\ y_{t+1} &= y_t + \eta \cdot \nabla_y f(x_t, y_t) \\ &\quad - \boldsymbol{\eta/2 \cdot \nabla_y f(x_{t-1}, y_{t-1})} \end{aligned}}$$

Negative Momentum: convergence

OGDA

$$x_{t+1} = x_t - \eta \cdot \nabla_x f(x_t, y_t) \\ \qquad + \boldsymbol{\eta/2 \cdot \nabla_x f(x_{t-1}, y_{t-1})}$$

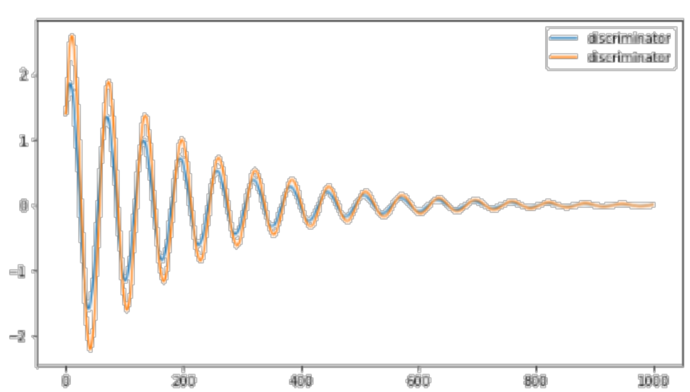
$$y_{t+1} = y_t + \eta \cdot \nabla_y f(x_t, y_t) \\ \qquad - \boldsymbol{\eta/2 \cdot \nabla_y f(x_{t-1}, y_{t-1})}$$

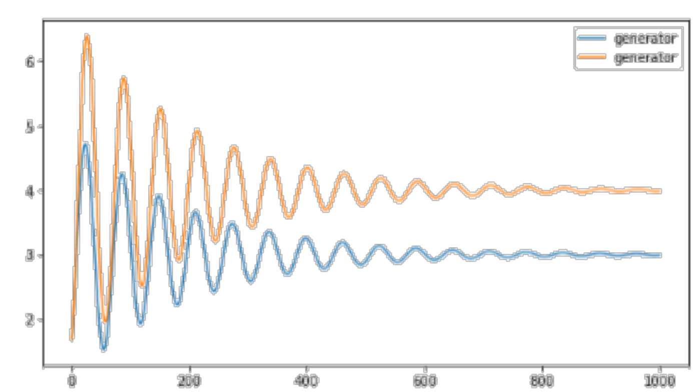
EG

$$x_{t+1/2} = x_t - \eta \cdot \nabla_x f(x_t, y_t)$$
$$x_{t+1} = x_t - \eta \cdot \nabla_x f\left(x_{t+1/2}, y_{t+1/2}\right)$$
$$y_{t+1/2} = y_t + \eta \cdot \nabla_y f(x_t, y_t)$$
$$y_{t+1} = y_t + \eta \cdot \nabla_y f\left(x_{t+1/2}, y_{t+1/2}\right)$$

- **[Daskalakis-Ilyas-Syrgkanis-Zeng ICLR'18, Liang-Stokes AISTATS'19, Gidel et al AISTATS'19, Mohtari et al '19]:** OGDA exhibits last iterate convergence & linear rates for *unconstrained* bilinear games with well-conditioned matrix A: $\min_{x \in \mathbb{R}^n} \max_{y \in \mathbb{R}^m} f(x, y) = x^{\mathrm{T}} A y + b^T x + c^T y$

- In previous isotropic Gaussian example: $X \sim \mathcal{N}\left((3,4), I_{2\times 2}\right)$, $G_\theta(Z) = \theta + Z$, $D_w(\cdot) = \langle w, \cdot \rangle$
- Recall: $\min_{\boldsymbol{\theta}} \max_{\boldsymbol{w}} \boldsymbol{w}^{\mathrm{T}} \cdot \left(\begin{bmatrix} 3 \\ 4 \end{bmatrix} - \boldsymbol{\theta} \right)$





Negative Momentum: convergence

OGDA

$$x_{t+1} = x_t - \eta \cdot \nabla_x f(x_t, y_t) + \boldsymbol{\eta/2 \cdot \nabla_x f(x_{t-1}, y_{t-1})}$$

$$y_{t+1} = y_t + \eta \cdot \nabla_y f(x_t, y_t) - \boldsymbol{\eta/2 \cdot \nabla_y f(x_{t-1}, y_{t-1})}$$

EG

$$x_{t+1/2} = x_t - \eta \cdot \nabla_x f(x_t, y_t)$$

$$x_{t+1} = x_t - \eta \cdot \nabla_x f\left(x_{t+1/2}, y_{t+1/2}\right)$$

$$y_{t+1/2} = y_t + \eta \cdot \nabla_y f(x_t, y_t)$$

$$y_{t+1} = y_t + \eta \cdot \nabla_y f\left(x_{t+1/2}, y_{t+1/2}\right)$$

- **[Daskalakis-Ilyas-Syrgkanis-Zeng ICLR'18, Liang-Stokes AISTATS'19, Gidel et al AISTATS'19, Mohtari et al '19]:** OGDA exhibits last-iterate convergence & linear rates for *unconstrained* bilinear games with well-conditioned matrix A: $\min_{x \in \mathbb{R}^n} \max_{y \in \mathbb{R}^m} f(x, y) = x^{\mathrm{T}} A y + b^T x + c^T y$
- **[Tseng'95, Liang-Stokes AISTATS'19, Mohtari et al '19, Azizian et al AISTATS'20]:** … ditto for *unconstrained strongly* convex-concave $f(x, y)$ and for EG/mirror-prox methods
- **[w/ Golowich, Pattathil, Ozdaglar COLT'20]:** Last-iterate of EG converges at rate $\Theta\left(\frac{1}{\sqrt{T}}\right)$ for *smooth convex-concave* $f(x, y)$ w/ Lipschitz Hessian, which is slower than the convergence rate of the average-iterate $\mathrm{O}\left(\frac{1}{T}\right)$ **[Nemirovski'04]**
- **[w/ Golowich, Pattathil'20]:** Ditto for OGDA

- **[Korpelevich'76, Popov'80, Facchinei and Pang'03, Mertikopoulos et al.'18, Jelena Diakonikolas-Daskalakis'20]:** asymptotic convergence results for OGDA, EG, Mirror-Prox

Negative Momentum: constrained case

(Projected) OGDA

$$x_{t+1/2} = \Pi_{\mathcal{X}}(x_{t-1/2} - \eta \cdot \nabla_x f(x_t, y_t))$$
$$x_{t+1} = \Pi_{\mathcal{X}}\left(x_{t+1/2} - \eta \cdot \nabla_x f(x_t, y_t)\right)$$
$$y_{t+1/2} = \Pi_{\mathcal{Y}}(y_{t-1/2} + \eta \cdot \nabla_y f(x_t, y_t))$$
$$y_{t+1} = \Pi_{\mathcal{Y}}\left(y_{t+1/2} + \eta \cdot \nabla_y f(x_t, y_t)\right)$$

(Projected) EG

$$x_{t+1/2} = \Pi_{\mathcal{X}}(x_t - \eta \cdot \nabla_x f(x_t, y_t))$$
$$x_{t+1} = \Pi_{\mathcal{X}}\left(x_t - \eta \cdot \nabla_x f\left(x_{t+1/2}, y_{t+1/2}\right)\right)$$
$$y_{t+1/2} = \Pi_{\mathcal{Y}}\left(y_t + \eta \cdot \nabla_y f(x_t, y_t)\right)$$
$$y_{t+1} = \Pi_{\mathcal{Y}}\left(y_t + \eta \cdot \nabla_y f\left(x_{t+1/2}, y_{t+1/2}\right)\right)$$

- **[Korpelevich'76, Popov'80, Facchinei and Pang'03, Mertikopoulos et al.'18, Jelena Diakonikolas-Daskalakis'20]:** asymptotic convergence results for OGDA, EG, Mirror-Prox, even in constrained setting
- **[Tseng'95]:** linear rates for EG in the strongly convex-concave case
- **[Korpelevich'76]:** Explicit (not great) rates and last iterate convergence for EG in (constrained) matrix games: $\min_{x \in \Delta_n} \max_{y \in \Delta_m} x^{\mathrm{T}} A y$
- **[Daskalakis-Panageas ITCS'19]:** Ditto for optimistic MWU
- **[Lee-Luo-Wei-Zhang'20]:** Ditto for OGDA

- **Remark:** Results apply to Monotone Variational Inequalities, which covers minimization, min-maximization, and certain multiplayer games as well!
- **Main open question:** Fast, last-iterate convergence rates in constrained case

Menu

- **Motivation**
- **Part 1: convex-concave objectives**
  - **Background: min-max in Game Theory and Learning**
  - **Removing oscillations via negative momentum**
- Part 2: nonconvex-nonconcave objectives
  - Some initial musings
  - Computational complexity
  - Convergence under structure: multi-agent RL

Menu

- **Motivation**
- **Part 1: convex-concave objectives**
  - **Background: min-max in Game Theory and Learning**
  - **Removing oscillations via negative momentum**
  - **Tying some loose ends**
- Part 2: nonconvex-nonconcave objectives
  - Some initial musings
  - Computational complexity
  - Convergence under structure: multi-agent RL

Remark 1: Beyond “Last-Iterate”

- Convex-Concave Setting: Table from **[Lin-Jin-Jordan’20]**

Settings	References	Gradient Complexity
Strongly-Convex-Strongly-Concave	Tseng [1995]	$\tilde{O}(\kappa_{\mathbf{x}} + \kappa_{\mathbf{y}})$
	Nesterov and Scrimali [2006]	
	Gidel et al. [2019]	
	Mokhtari et al. [2019b]	
	Alkousa et al. [2019]	$\tilde{O}(\min\{\kappa_{\mathbf{x}}\sqrt{\kappa_{\mathbf{y}}}, \kappa_{\mathbf{y}}\sqrt{\kappa_{\mathbf{x}}}\})$
	This paper (Theorem 5.1)	$\tilde{O}(\sqrt{\kappa_{\mathbf{x}}\kappa_{\mathbf{y}}})$
	Lower bound [Zhang et al., 2019]	$\tilde{\Omega}(\sqrt{\kappa_{\mathbf{x}}\kappa_{\mathbf{y}}})$
Strongly-Convex-Linear (special case of strongly-convex-concave)	Juditsky and Nemirovski [2011]	$O(\sqrt{\kappa_{\mathbf{x}}/\epsilon})$
	Hamedani and Aybat [2018]	
	Zhao [2019]	
Strongly-Convex-Concave	Thekumparampil et al. [2019]	$\tilde{O}(\kappa_{\mathbf{x}}/\sqrt{\epsilon})$
	This paper (Corollary 5.2)	$\tilde{O}(\sqrt{\kappa_{\mathbf{x}}/\epsilon})$
	Lower bound [Ouyang and Xu, 2019]	$\tilde{\Omega}(\sqrt{\kappa_{\mathbf{x}}/\epsilon})$
Convex-Concave	Nemirovski [2004]	$O(\epsilon^{-1})$
	Nesterov [2007]	
	Tseng [2008]	
	This paper (Corollary 5.3)	$\tilde{O}(\epsilon^{-1})$
	Lower bound [Ouyang and Xu, 2019]	$\Omega(\epsilon^{-1})$

Remark 1: Beyond "Last-Iterate"

- Nonconvex-Concave Setting: Table from **[Lin-Jin-Jordan'20]**

Settings	References	Gradient Complexity
Nonconvex-Strongly-Concave (stationarity of f or stationarity of Φ)	Jin et al. [2019]	$\tilde{O}(\kappa_{\mathbf{y}}^2 \epsilon^{-2})$
	Rafique et al. [2018]	
	Lin et al. [2019]	
	Lu et al. [2019]	
	This paper (Theorem 6.1 & A.7)	$\tilde{O}(\sqrt{\kappa_{\mathbf{y}}} \epsilon^{-2})$
Nonconvex-Concave (stationarity of f)	Lu et al. [2019]	$\tilde{O}(\epsilon^{-4})$
	Nouiehed et al. [2019]	$\tilde{O}(\epsilon^{-3.5})$
	This paper (Corollary 6.2)	$\tilde{O}(\epsilon^{-2.5})$
Nonconvex-Concave (stationarity of Φ)	Jin et al. [2019]	$\tilde{O}(\epsilon^{-6})$
	Rafique et al. [2018]	
	Lin et al. [2019]	
	Kong and Monteiro [2019]	$\tilde{O}(\epsilon^{-3})$
	Thekumparampil et al. [2019]	
	This paper (Corollary A.8)	$\tilde{O}(\epsilon^{-3})$

Remark 2: Connection to Monotone Variational Inequalities

- **Operator:** $F: \mathcal{Z} \longrightarrow \mathcal{Z}$ where $\mathcal{Z} \subseteq \mathbb{R}^d$, F is Lipschitz

- **Def 1:** F is monotone if $\big(F(z) - F(z')\big)(z - z') \geq 0$, for all z, z'

- **Goal:** find z^* s.t. $F(z)(z - z^*) \geq 0$, for all $z \in \mathcal{Z}$. (Minty Variational Inequality (MVI))
- **Goal 2:** find z^* s.t. $F(z^*)(z - z^*) \geq 0$, for all $z \in \mathcal{Z}$. (Stampacchia Variational Inequality (SVI))
- MVI, SVI are equivalent for monotone and continuous F

- Setting $F(x) = \nabla f(x)$, for cont. differentiable convex f : solution to MVI/SVI is global min

- Setting $F(x, y) = (\nabla_x f(x, y), -\nabla_y f(x, y))$, for cont. differentiable convex-concave f: solution to MVI/SVI is min-max eq.

- This formulation covers **multi-player games** as well, e.g. zero-sum polymatrix games **[Daskalakis-Papadimitriou'09, Cai-Daskalakis'11, Cai et al'16]** and socially-concave games **[Even-Dar, Mansour, Nadav'09]**
- Many of the convergence results mentioned earlier apply to solving SVI/MVI, and hence also to these multiplayer games

Remark 3: GAN learning

- **[Daskalakis-Ilyas-Syrgkanis-Zeng ICLR'18]:** WGANs with a linear generator and discriminator, trained using OGDA learn multi-dimensional Gaussians (no sample analysis)
- **[Feizi-Farnia-Ginart-Tse'17]:** WGANs with a linear generator and quadratic discriminator, trained using alternating GDA learn multi-dimensional Gaussians (using poly samples)
- **[w/ Qi Lei, Jason Lee, Alex Dimakis ICML'20]:** WGANS with a single-layer generator and quadratic discriminator, trained using alternating GDA (with different step sizes for descent/ascent) identify true model in realizable setting (using poly samples)
- Generator $\mathrm{G}(\mathrm{z}) = \phi(A \cdot z), z \sim \mathcal{N}(0, \mathrm{I})$, optimizes over *A*
- *Discriminator* $\mathrm{D}(\mathrm{x}) = \langle W, x \cdot x^{\mathrm{T}} \rangle$, optimizes over *W*

Thanks!

- to be continued…

Game Theory and Machine Learning

Constantinos (a.k.a. “Costis”) Daskalakis
CSAIL and EECS, MIT


Future of AI: learning + strategic reasoning

A dictum

Minimization is to current AI

what min-max optimization is to future AI

(or, more broadly, multi-agent equilibrium learning)

- **Minimization:** AI agent is learning/decision-making in a stationary environment
- **Min-max optimization/equilibrium learning:** AI agent is learning and decision-making in a *changing* environment due to…
    - the presence of another/multiple other learning agents with conflicting/aligned interests
    - noise/adversaries poisoning or corrupting the data
    - the need to enforce constraints on the learning outcome, e.g. GANs, private release of data etc.

Our focus: min-max optimization

Solve: $\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x, y)$
where x, y high-dimensional
& potentially constrained

- **Applications:** Mathematics, Optimization, Game Theory,…
  [von Neumann 1928, Dantzig '47, Brown'50, Robinson'51, Blackwell'56,…]

- **Best-Case Scenario:** f is continuous, convex in x for all y, and concave in y for all x
- **[von Neumann 1928]:** If $\mathcal{X}, \mathcal{Y}$ are convex & compact: $\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x, y) = \max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} f(x, y)$ (*)
- Moreover solutions to LHS are convex set, solutions to RHS are convex set
- Interpretation of (*): If x^* opt for LHS, and y^* opt for RHS, then (x^*, y^*) is equilibrium of zero-sum game where min player pays max player $f(x,y)$
- **[von Neumann 1928, Dantzig '47, others]:** equilibrium computation $\Leftrightarrow$ convex programming
- **[Brown'50,Robinson'51,Blackwell'56,Hannan'57,…]:** if min and max players use no-regret learning algorithms to update their strategies, in parallel, the resulting trajectories $(x_t)_t$ and $(y_t)_t$ converge to min-max equilibrium *in the average sense, i.e.*

$$f\left(\frac{1}{t}\sum_{\tau<t} x_\tau, \frac{1}{t}\sum_{\tau<t} y_\tau\right) \longrightarrow f(x^*, y^*)$$

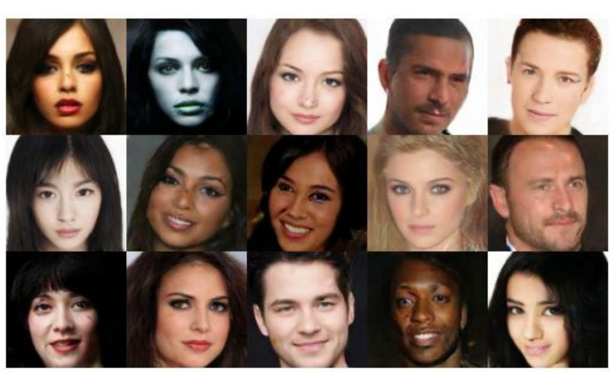
Our focus: min-max optimization

Solve: $\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x, y)$
where x, y high-dimensional
& potentially constrained

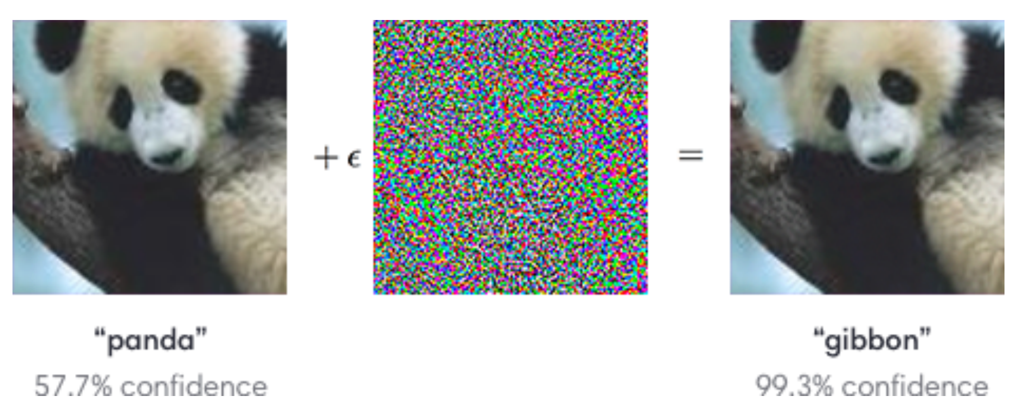
- **Applications:** Mathematics, Optimization, Game Theory,...
  [von Neumann 1928, Dantzig ’47, Brown’50, Robinson’51, Blackwell’56,...]

- **Best-Case Scenario:** f is convex in x for all y, concave in y for all x

- **Many Deep Learning Applications:** GANs **[Goodfellow et al. NeurIPS’14]**, training against adversarial attacks **[Madry et al. ICLR’17]**, …
  – *exacerbate the importance of first-order methods, last-iterate convergence, non convex-concave objectives*

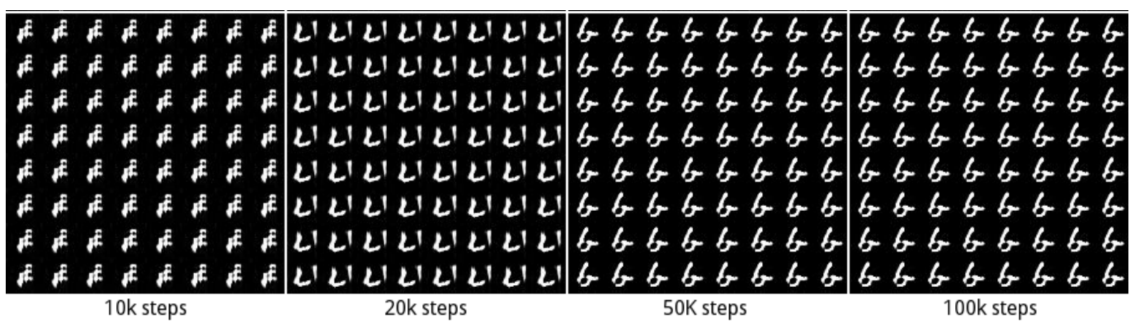


[BEGAN. Bertholet et al. 2017]

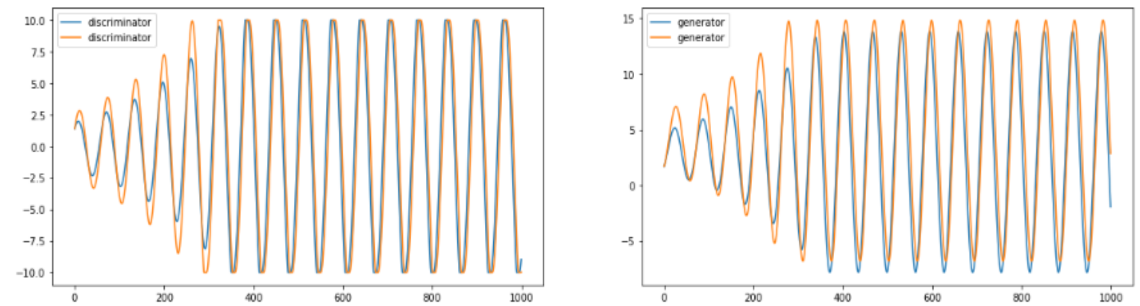


[Goodefllow-Shlens-Szegedy 2014]

Training Oscillations of GDA (and its variants)

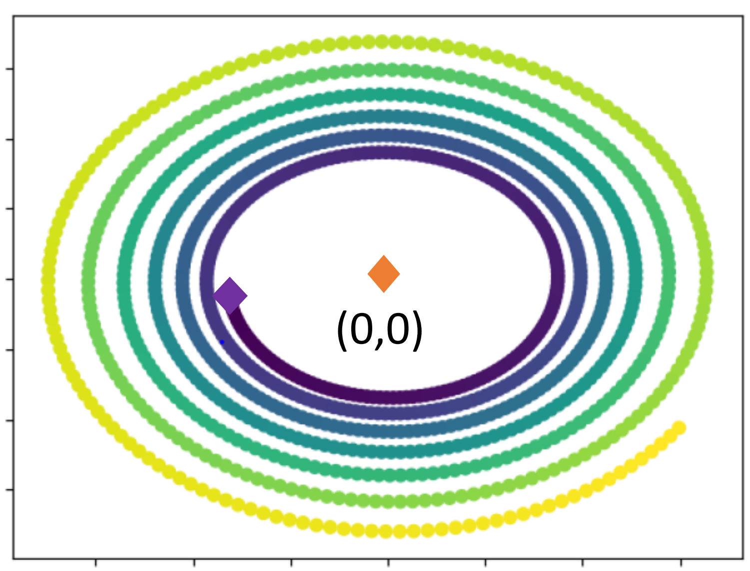


GAN training on MNIST



GAN training on Isotropic Gaussian

the simplest example: $\min_{x} \max_{y} (x \cdot y)$



Gradient Descent/Ascent (GDA)

$$x_{t+1} = x_t - \eta \cdot \nabla_x f(x_t, y_t) = x_t - \eta \cdot y_t$$
$$y_{t+1} = y_t + \eta \cdot \nabla_y f(x_t, y_t) = y_t + \eta \cdot x_t$$

◆ : start

◆ : min-max equilibrium

Negative Momentum: OGDA and EG

OGDA

$$x_{t+1} = x_t - \eta \cdot \nabla_x f(x_t, y_t) \\ \qquad + \boldsymbol{\eta/2 \cdot \nabla_x f(x_{t-1}, y_{t-1})}$$

$$y_{t+1} = y_t + \eta \cdot \nabla_y f(x_t, y_t) \\ \qquad - \boldsymbol{\eta/2 \cdot \nabla_y f(x_{t-1}, y_{t-1})}$$

EG

$$x_{t+1/2} = x_t - \eta \cdot \nabla_x f(x_t, y_t)$$

$$x_{t+1} = x_t - \eta \cdot \nabla_x f\left(x_{t+1/2}, y_{t+1/2}\right)$$

$$y_{t+1/2} = y_t + \eta \cdot \nabla_y f(x_t, y_t)$$

$$y_{t+1} = y_t + \eta \cdot \nabla_y f\left(x_{t+1/2}, y_{t+1/2}\right)$$

- **Last lecture:** OGDA and EG, with constant step size, exhibit last iterate convergence to min-max equilibrium, for smooth convex-concave functions $f(x, y)$
- Convergence rates also known, mostly for the unconstrained case

Menu

- **Motivation**
- **Part 1: convex-concave objectives**
  - **Background: min-max in Game Theory and Learning**
  - **Removing oscillations via negative momentum**
  - **Tying some loose ends**
- Part 2: nonconvex-nonconcave objectives
  - Some initial musings
  - The right solution concept
  - Computational complexity
  - Convergence under structure: multi-agent RL

Menu

- **Motivation**
- **Part 1: convex-concave objectives**
  - **Background: min-max in Game Theory and Learning**
  - **Removing oscillations via negative momentum**
  - **Tying some loose ends**
- **Part 2: nonconvex-nonconcave objectives**
  - **Some initial musings**
  - The right solution concept
  - Computational complexity
  - Convergence under structure: multi-agent RL

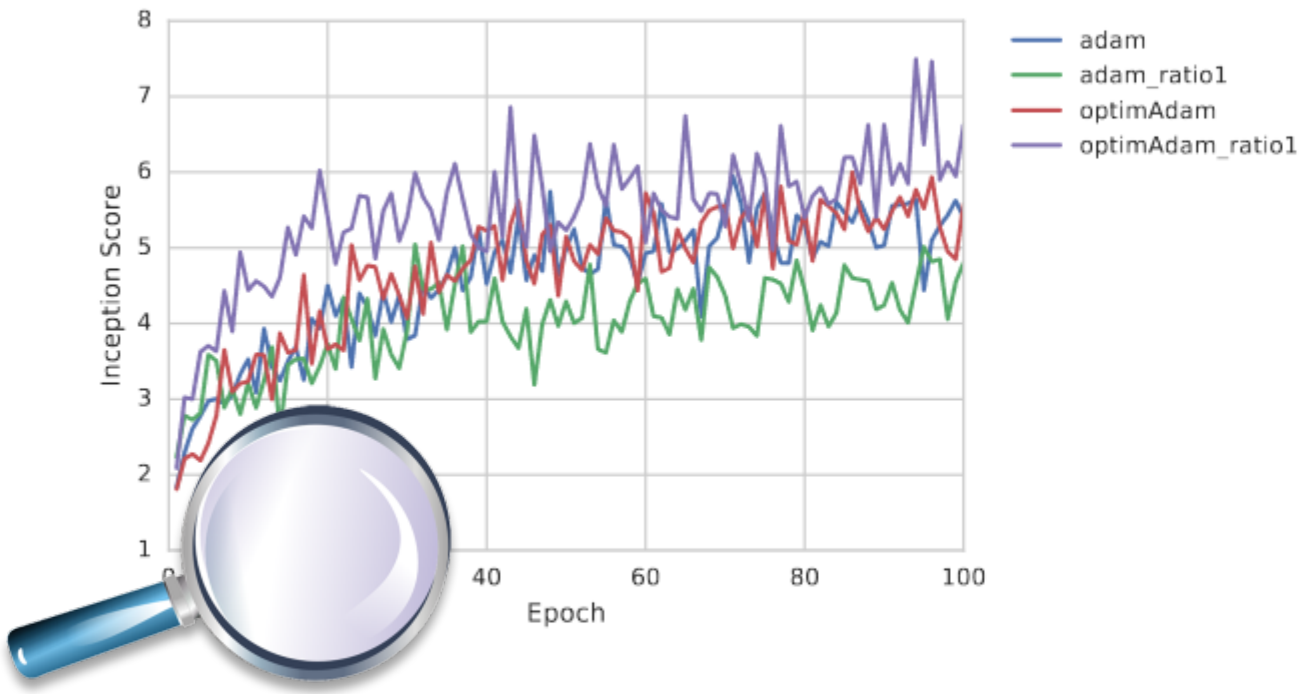
Negative Momentum, in the Wild

- Is negative momentum helpful, outside of the convex-concave setting?

- **[Daskalakis-Ilyas-Syrgkanis-Zeng ICLR'18]:** Optimistic Adam
- ***Adam***, a variant of stochastic gradient descent with momentum and per-parameter adaptive learning rates, proposed by **[Kingma-Ba ICLR'15]**, has found wide adoption in deep learning, although it doesn't always converge, even in simple convex settings **[Reddi-Kale-Kumar ICLR'18]**
- In any event, ***Optimistic Adam*** is the right adaptation of Adam to "undo some of the past gradients," i.e. have negative momentum

Optimistic Adam, on CIFAR10

- Compare **Adam** and **Optimistic Adam**, trained on CIFAR10, in terms of Inception Score
- No fine-tuning for **Optimistic Adam**; used same hyper-parameters for both algorithms as suggested in Gulrajani et al. (2017)



Optimistic Adam, on CIFAR10

- Compare **Adam** and **Optimistic Adam**, trained on CIFAR10, in terms of Inception Score
- No fine-tuning for **Optimistic Adam**; used same hyper-parameters for both algorithms as suggested in Gulrajani et al. (2017)

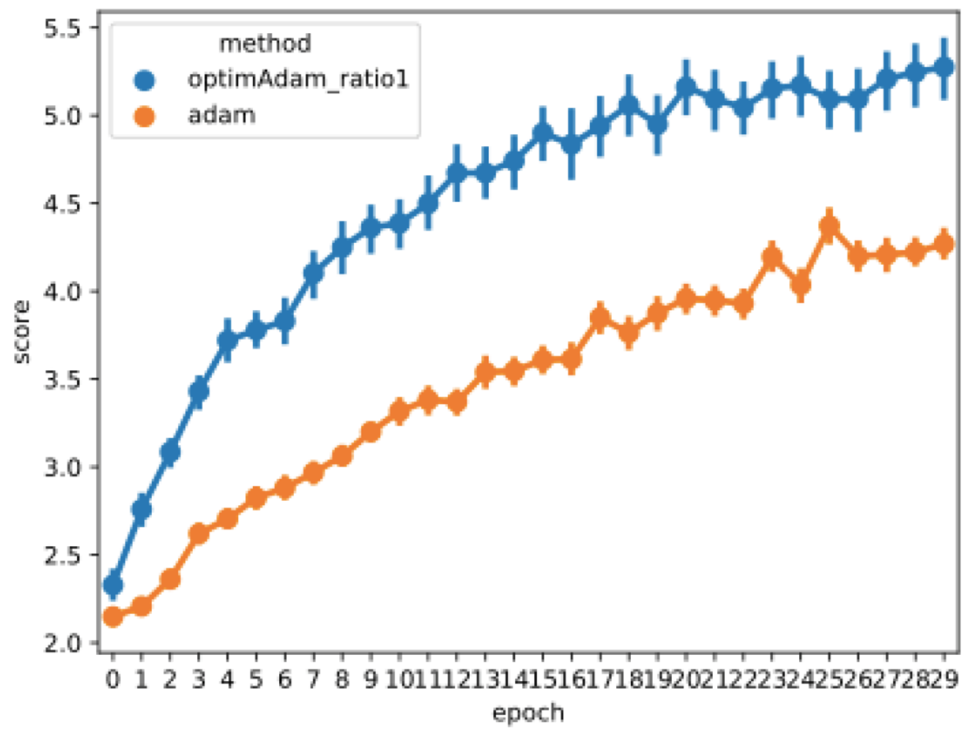
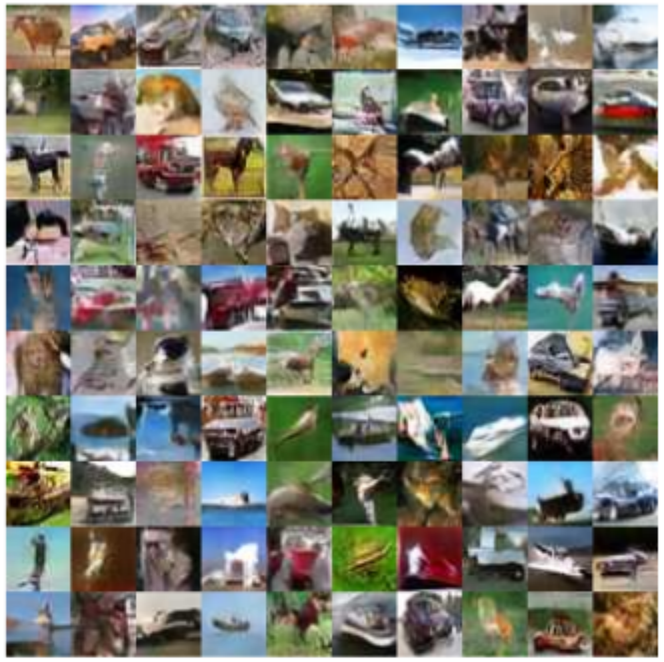


Figure 14: The inception scores across epochs for GANs trained with Optimistic Adam (ratio 1) and Adam (ratio 5) on CIFAR10 (the two top-performing optimizers found in Section 6, with 10%-90% confidence intervals. The GANs were trained for 30 epochs and results gathered across 35 runs.



(b) Sample of images from Generator of Epoch 94, which had the highest inception score.

- Further supporting evidence for negative momentum methods by **[Yadav et al. ICLR'18, Gidel et al. AISTATS'19, Chavdarova et al. NeurIPS'19]**

Decreasing Momentum Trend

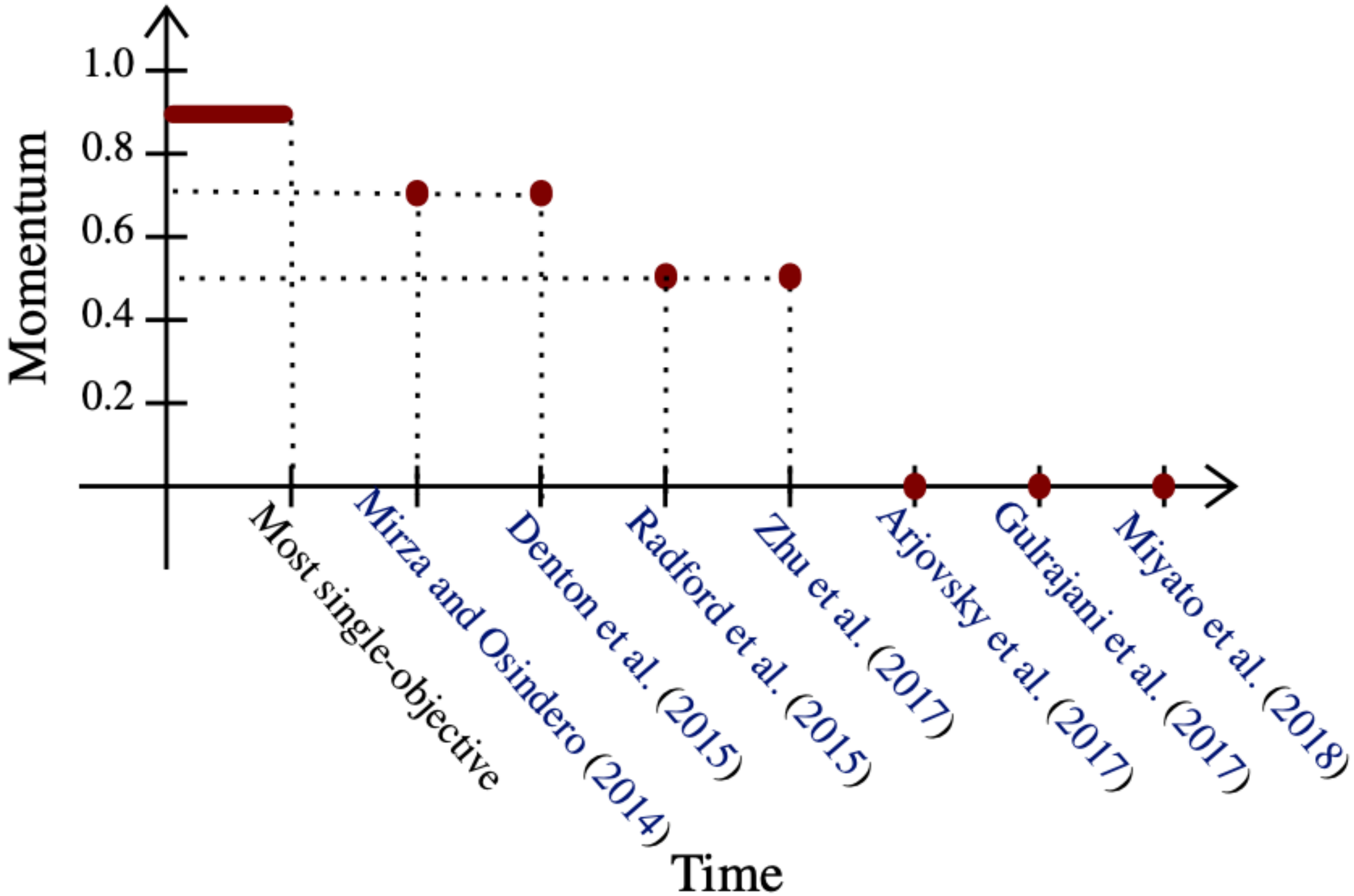


Figure 1: Decreasing trend in the value of momentum used for training GANs across time.

[Gidel et al. AISTATS'19]

Menu

- **Motivation**
- **Part 1: convex-concave objectives**
  - **Background: min-max in Game Theory and Learning**
  - **Removing oscillations via negative momentum**
  - **Tying some loose ends**
- **Part 2: nonconvex-nonconcave objectives**
  - **Some initial musings**
  - The right solution concept
  - Computational complexity
  - Convergence under structure: multi-agent RL

Menu

- **Motivation**
- **Part 1: convex-concave objectives**
  - **Background: min-max in Game Theory and Learning**
  - **Removing oscillations via negative momentum**
  - **Tying some loose ends**
- **Part 2: nonconvex-nonconcave objectives**
  - **Some initial musings**
  - **The right solution concept**
  - Computational complexity
  - Convergence under structure: multi-agent RL

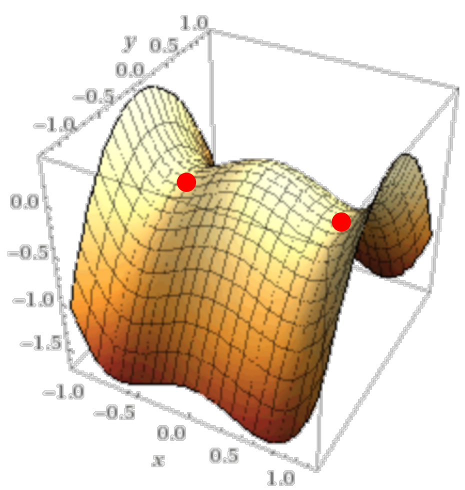
Nonconvex-Nonconcave Objectives

- If $f(x, y)$ is not convex-concave, von Neumann's theorem breaks
- For some f: $\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} f(x, y) \neq \max_{y \in \mathcal{Y}} \min_{x \in \mathcal{X}} f(x, y)$

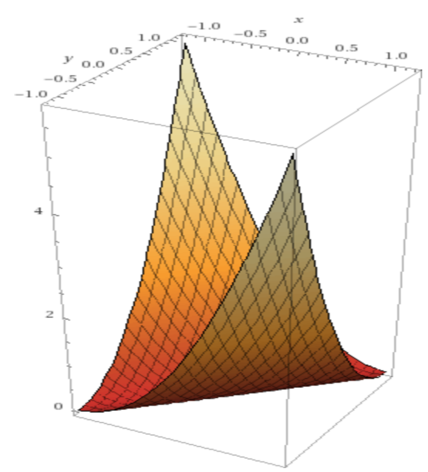
  (even though both are well-defined when f is continuous and $\mathcal{X}$ and $\mathcal{Y}$ are convex and compact)

- For other f: equality holds but there are multiple, disconnected solutions

$f(x, y) = x^4 - x^2 - y^2$



$f(x, y) = (x - y)^2$



$$\min_{x \in [-1,1]} \max_{y \in [-1,1]} f(x, y) \neq \max_{y \in [-1,1]} \min_{x \in [-1,1]} f(x, y)$$

Nonconvex-Nonconcave Objectives

- What should be the goal in min-max optimization when the objective is not convex-concave?

Given continuous f and convex, compact set S:

$$\min_x \max_y f(x, y)$$

$$\text{s.t. } (x, y) \in \text{ S}$$

(More General constrained min-max problem)

Definitions: $S(x,\cdot) = \{y \mid (x, y) \in \text{ S}\}$

$S_X = \{x \mid \exists\, y\ s.t. (x, y) \in \text{ S}\}$

Similarly: $S(\cdot, y)$, S_Y

- Solution Concepts:

Take 1: *local min-global max solution* $\triangleq$ point $(x^*, y^*) \in$ S such that, for some $\delta > 0$,

$$f(x^*, y) \leq f(x^*, y^*) \leq \max_{y' \in S(x,\cdot)} f(x, y'),\ \text{for all } y \in S(x^*,\cdot) \text{ and } x \in \mathcal{N}_\delta(x^*) \cap \text{S}_X$$

Take 2 [Jin-Netrapali-Jordan ICML'20]: *local minimax solution* $\triangleq$ more local version of 1, where y player is also constrained to move locally

Take 3 [Daskalakis-Panageas NeurIPS'18, Mazumdar-Ratliff'18]: *local min-max equilibrium* $\triangleq$ point $(x^*, y^*) \in$ S s.t. for some $\delta > 0$:

$$f(x^*, y) \leq f(x^*, y^*) \leq f(x, y^*),\ \text{for all } y \in \mathcal{N}_\delta(y^*) \cap S(x^*,\cdot) \text{ and } x \in \mathcal{N}_\delta(x^*) \cap S(\cdot, y^*)$$

1, 2: sequential move notions; 3: simultaneous move notion

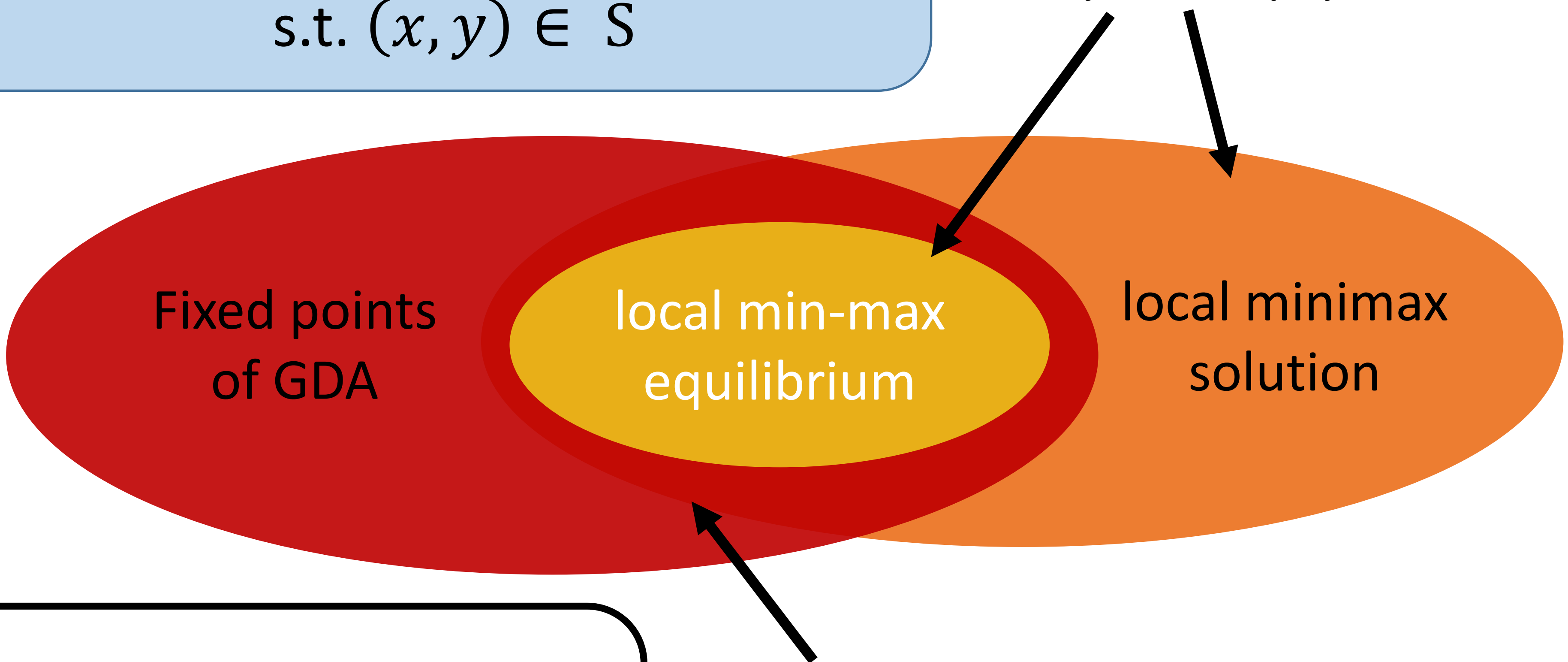
1: guaranteed to exist, but too strong (NP-hard to find); 2, 3: don't always exist

Nonconvex-Nonconcave Objectives

Given continuous f and convex, compact set S:

$$\min_x \max_y f(x, y)$$
$$\text{s.t. } (x, y) \in \ S$$

may be empty



GDA

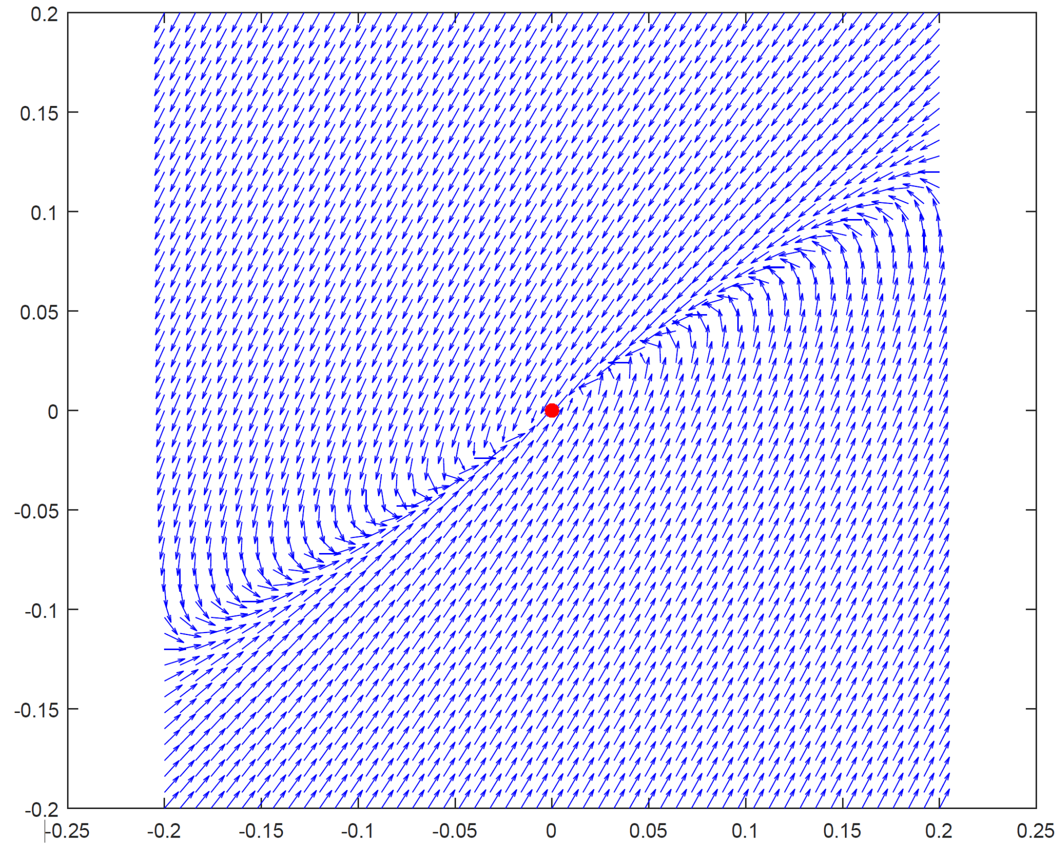
$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix} \leftarrow \Pi_S\left(\begin{bmatrix} x_t - \eta \cdot \nabla_x f(x_t, y_t) \\ y_t + \eta \cdot \nabla_y f(x_t, y_t) \end{bmatrix}\right)$$

guaranteed to be non-empty, if f is continuously differentiable, by Brouwer's fixed point theorem

Discrepancy between fixed points of GDA and local min-max equilibria?
Is it due to unstable fixed points of GDA?

Nonconvex-Nonconcave Objectives

- Well, first of all, local min-max equilibria need not be stable fixed points of GDA
  - recall that (0,0) is not stable for GDA dynamics for $\min_x \max_y x \cdot y$
- **[Daskalakis, Panageas NeurIPS'18]:** Moreover, stable points of GDA can be a strict superset of local min-max equilibria
- e.g. $f(x, y) = -\frac{1}{8} \cdot x^2 - \frac{1}{2} \cdot y^2 + \frac{6}{10} \cdot x \cdot y$



Gradient Descent-Ascent field

- **[Daskalakis, Panageas NeurIPS'18, Adolphs et al. 18]:** Following nested-ness holds in unconstrained case: Local Min-Max $\subseteq$ Stable Points of GDA $\subseteq$ Stable Points of OGDA
- (stability refers to linear stability and left inclusion holds for strongly local min-max equilibria---observe that (0,0) is not strongly local min-max eq for $\min_x \max_y x \cdot y$)

Menu

- **Motivation**
- **Part 1: convex-concave objectives**
  - **Background: min-max in Game Theory and Learning**
  - **Removing oscillations via negative momentum**
  - **Tying some loose ends**
- **Part 2: nonconvex-nonconcave objectives**
  - **Some initial musings**
  - **The right solution concept**
  - Computational complexity
  - Convergence under structure: multi-agent RL

Menu

- **Motivation**
- **Part 1: convex-concave objectives**
  - **Background: min-max in Game Theory and Learning**
  - **Removing oscillations via negative momentum**
  - **Tying some loose ends**
- **Part 2: nonconvex-nonconcave objectives**
  - **Some initial musings**
  - **The right solution concept**
  - **Computational complexity of GDA Fixed Points**
  - Convergence under structure: multi-agent RL

The Complexity of GDA Fixed Points

GDA

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix} \leftarrow \underbrace{\Pi_S\left(\begin{bmatrix} x_t - \eta \cdot \nabla_x f(x_t, y_t) \\ y_t + \eta \cdot \nabla_y f(x_t, y_t) \end{bmatrix}\right)}_{F_{GDA}(x_t, y_t)}$$

Goal: given $\varepsilon, \eta > 0$, description of (or first-order access to) Lipschitz and smooth function f, and projection oracle to S, compute (x^*, y^*) s.t.

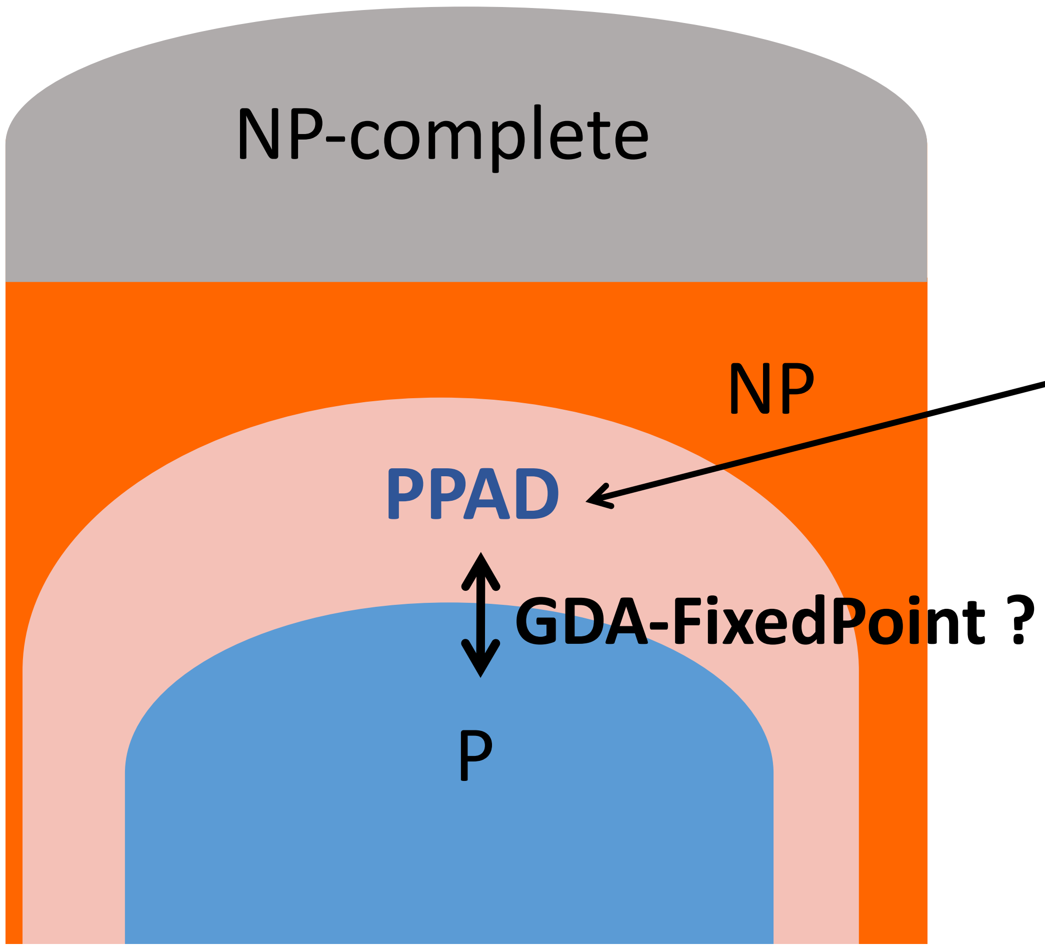
$$\|(x^*, y^*) - F_{GDA}(x^*, y^*)\|_2 \leq \varepsilon$$

If f is arbitrary, this seems computationally difficult.

But is it **NP**-hard to compute?

A: No, because computing approximate Brouwer fixed points of Lipschitz functions resides in complexity class **PPAD**, which is believed to be lower than **NP** (unless **NP**=**coNP** **[Johnson-Papadimitriou-Yannakakis'88, Megiddo-Papadimitriou'91]**)

The Complexity of GDA Fixed Points

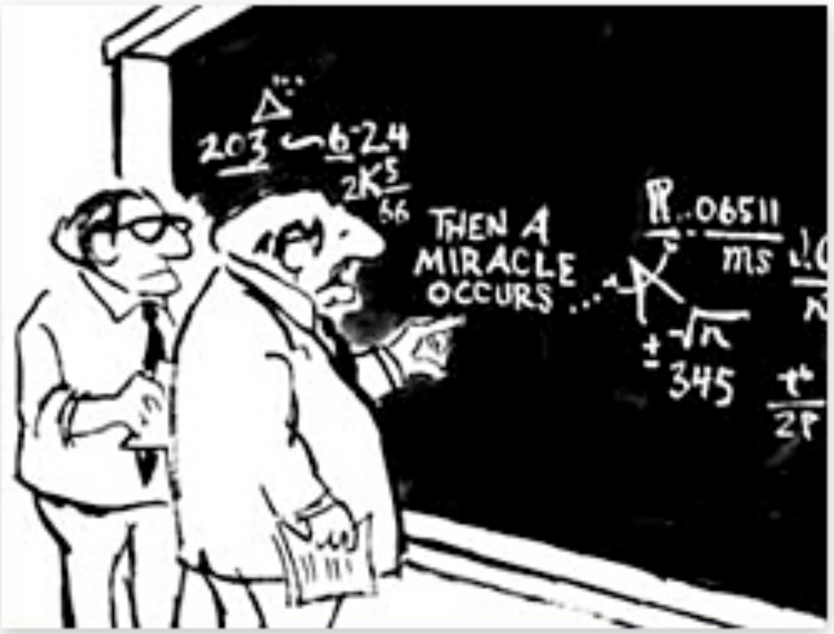


Computing Brouwer Fixed Points, and Nash Equilibria in non-zero sum games are both **PPAD**-complete problems (i.e. as hard as any problem in **PPAD**)

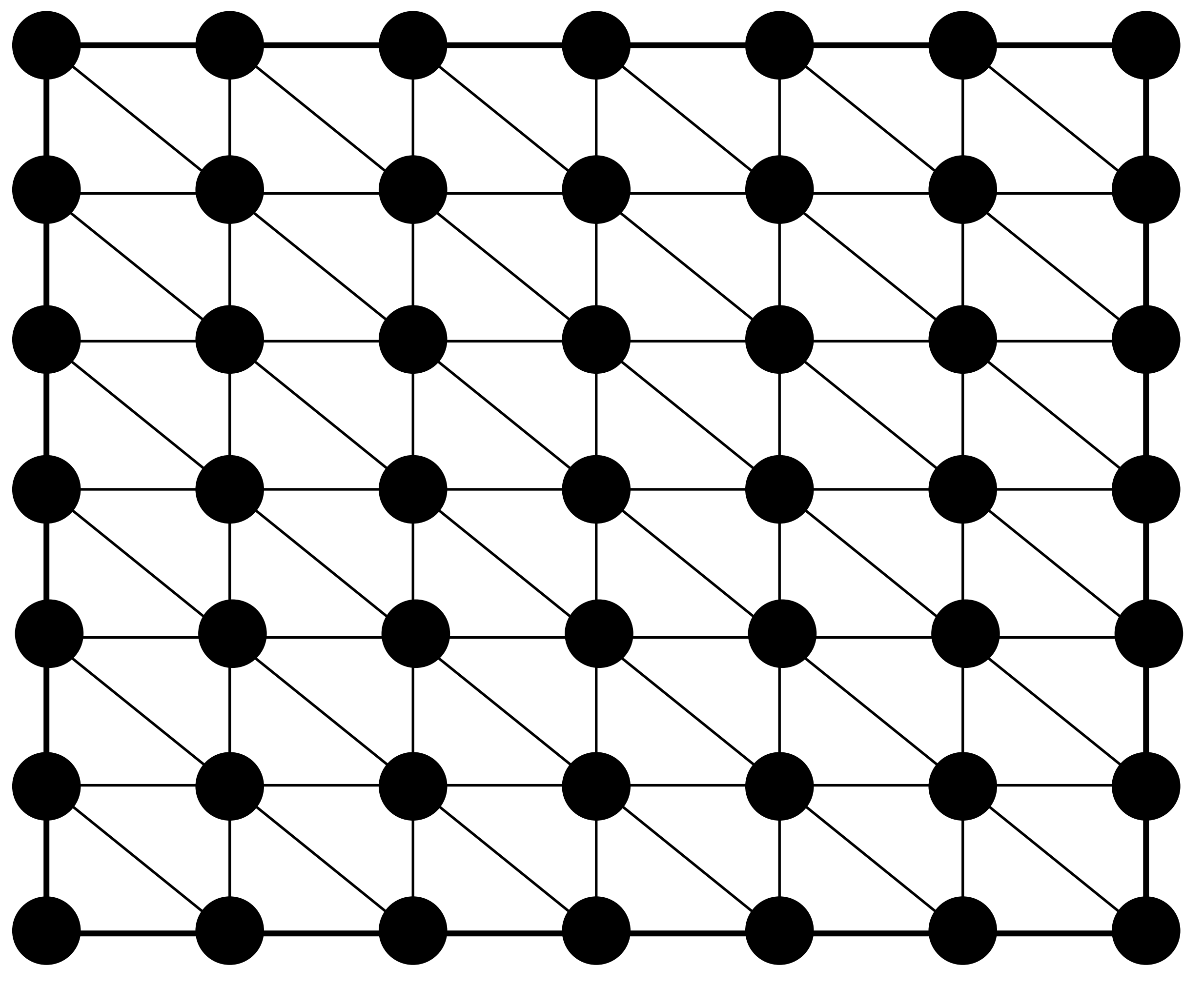
The PPAD Complexity Class [Papadimitriou'94]

PPAD: The class of all problems in **NP** with guaranteed solution by dint of the following graph-theoretic lemma:

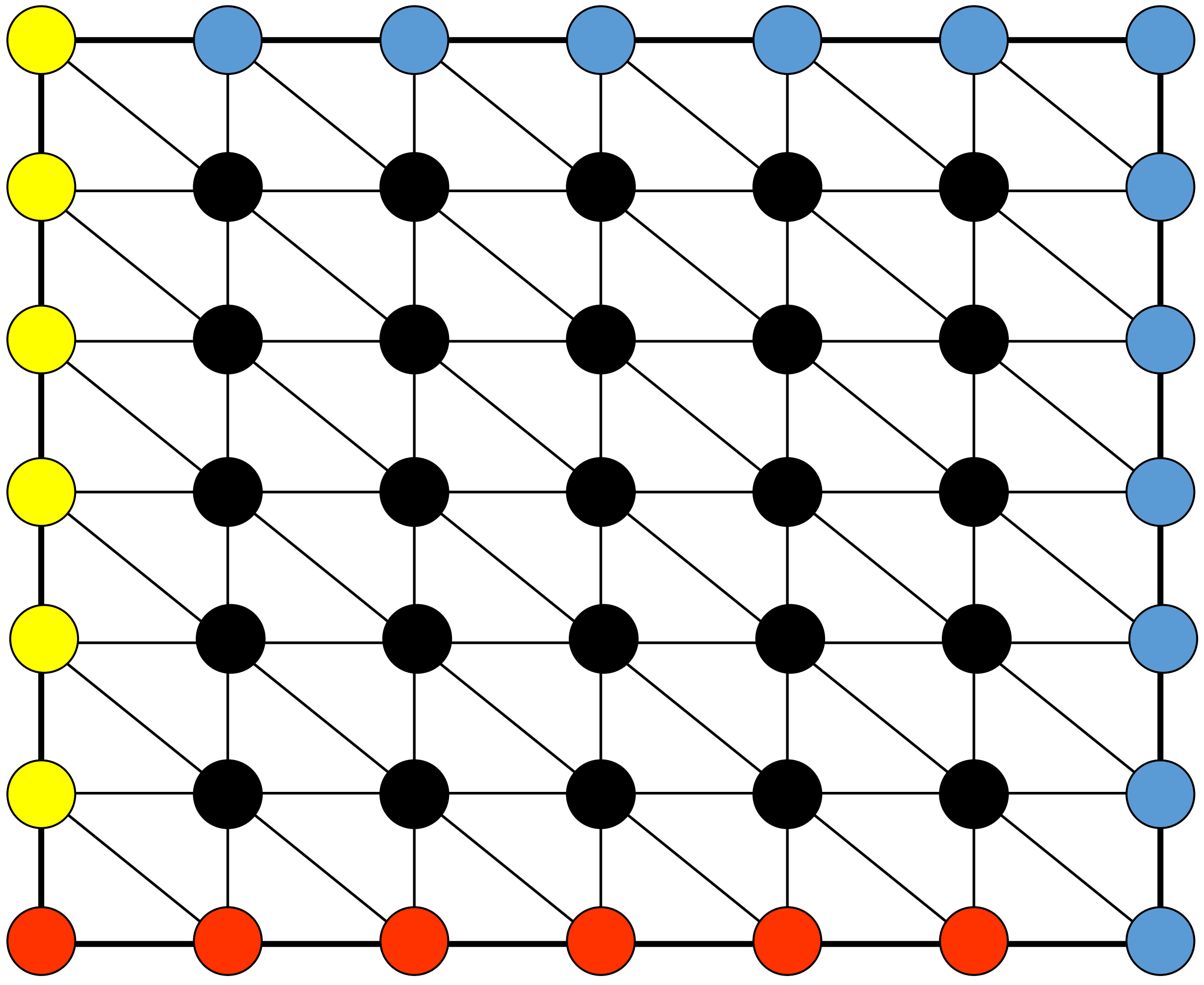
> *If a directed graph has an unbalanced node (a node with indegree ≠ outdegree), it must have another.*



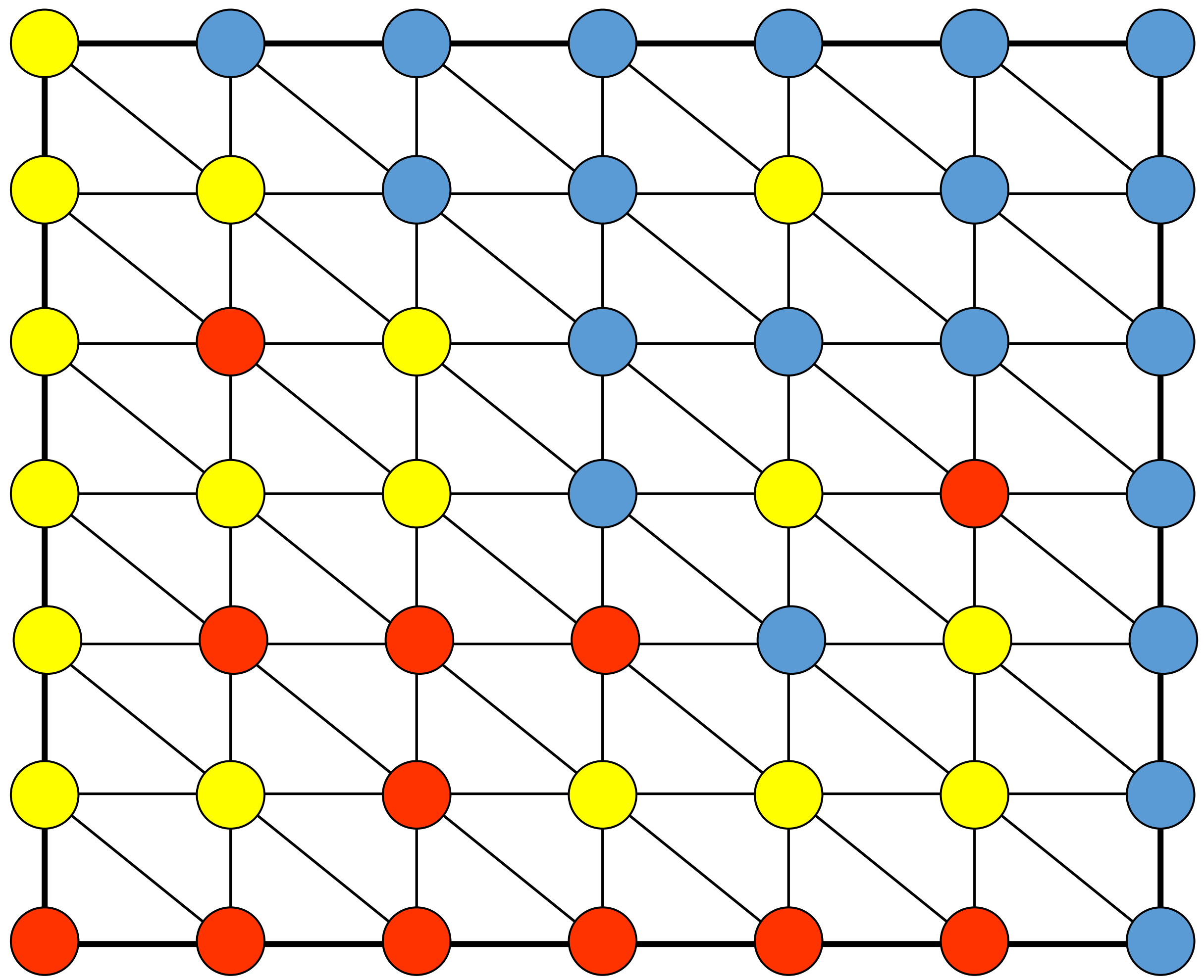
Sperner’s Lemma



Sperner's Lemma

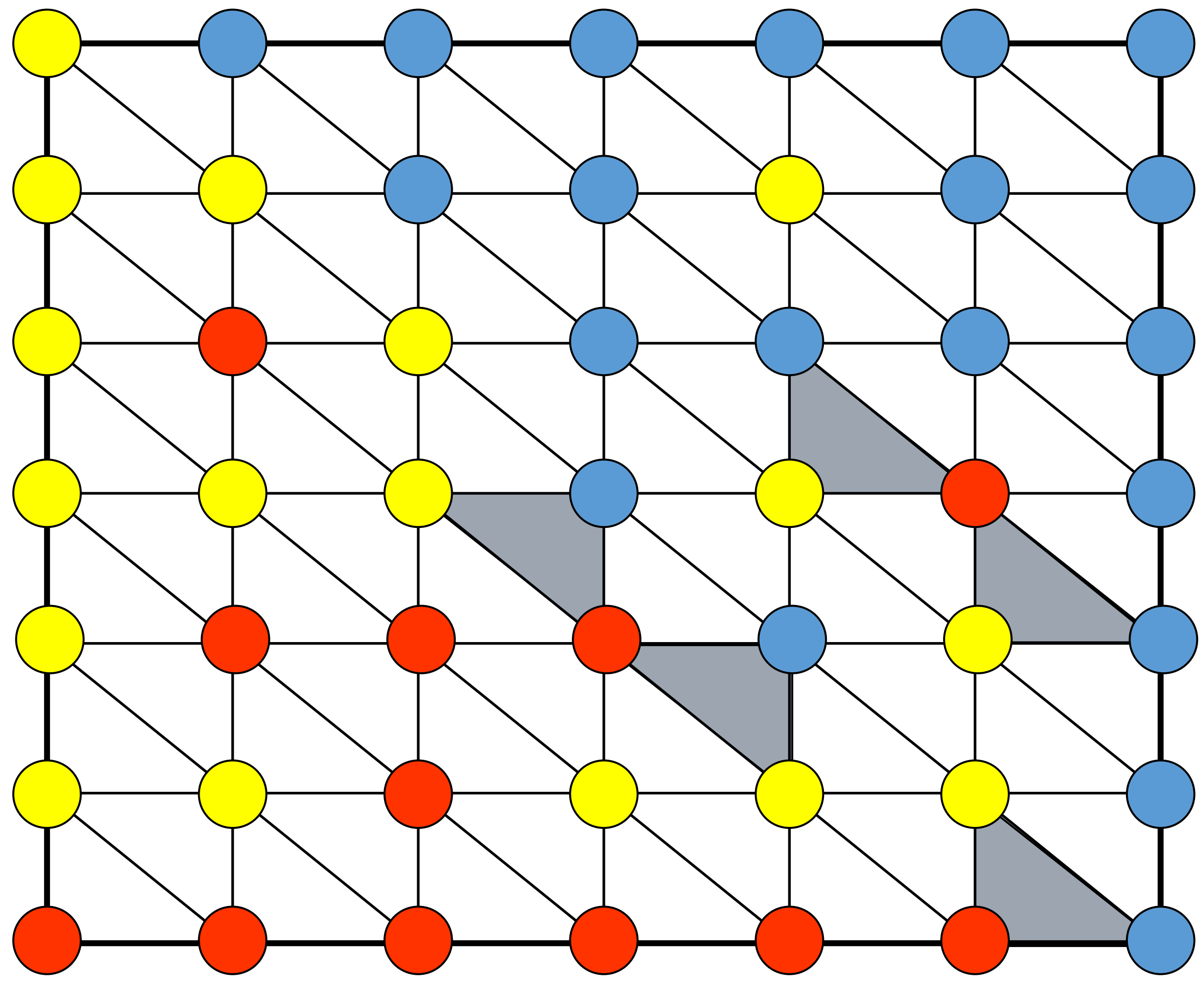


Sperner's Lemma



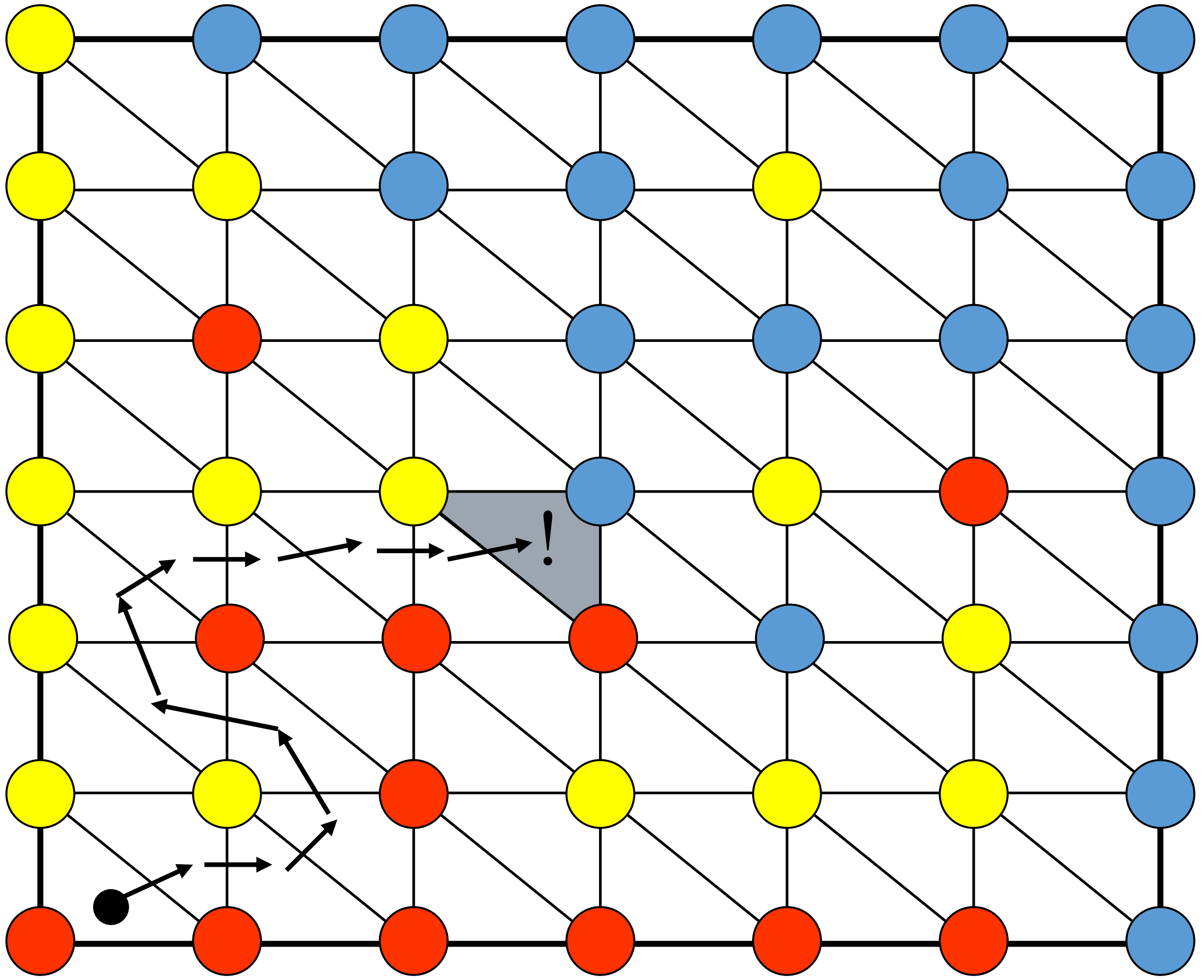
Lemma: No matter how the internal nodes are colored there exists a tri-chromatic triangle. In fact, an odd number of them.

Sperner's Lemma



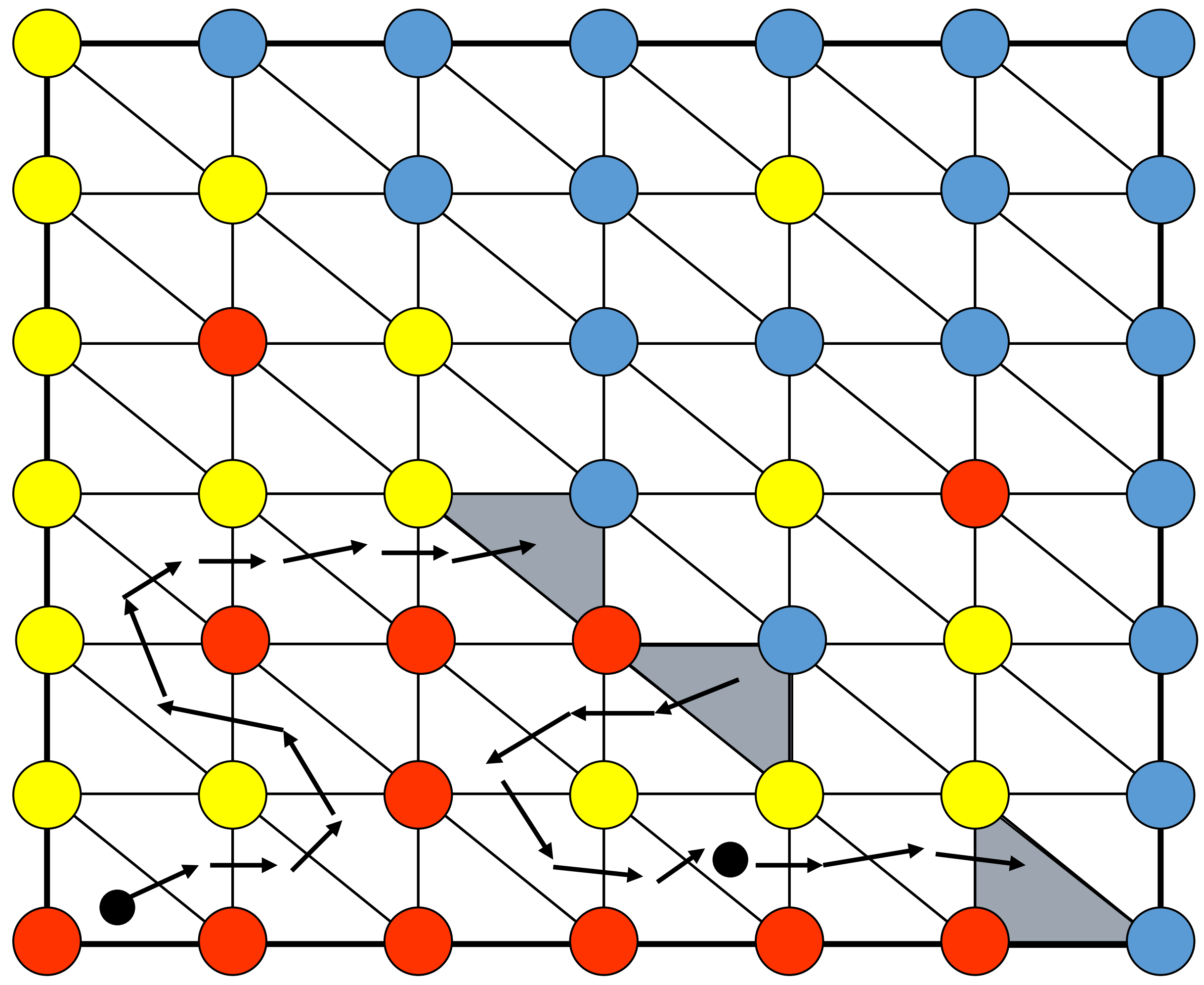
Lemma: No matter how the internal nodes are colored there exists a tri-chromatic triangle. In fact, an odd number of them.

Sperner's Lemma



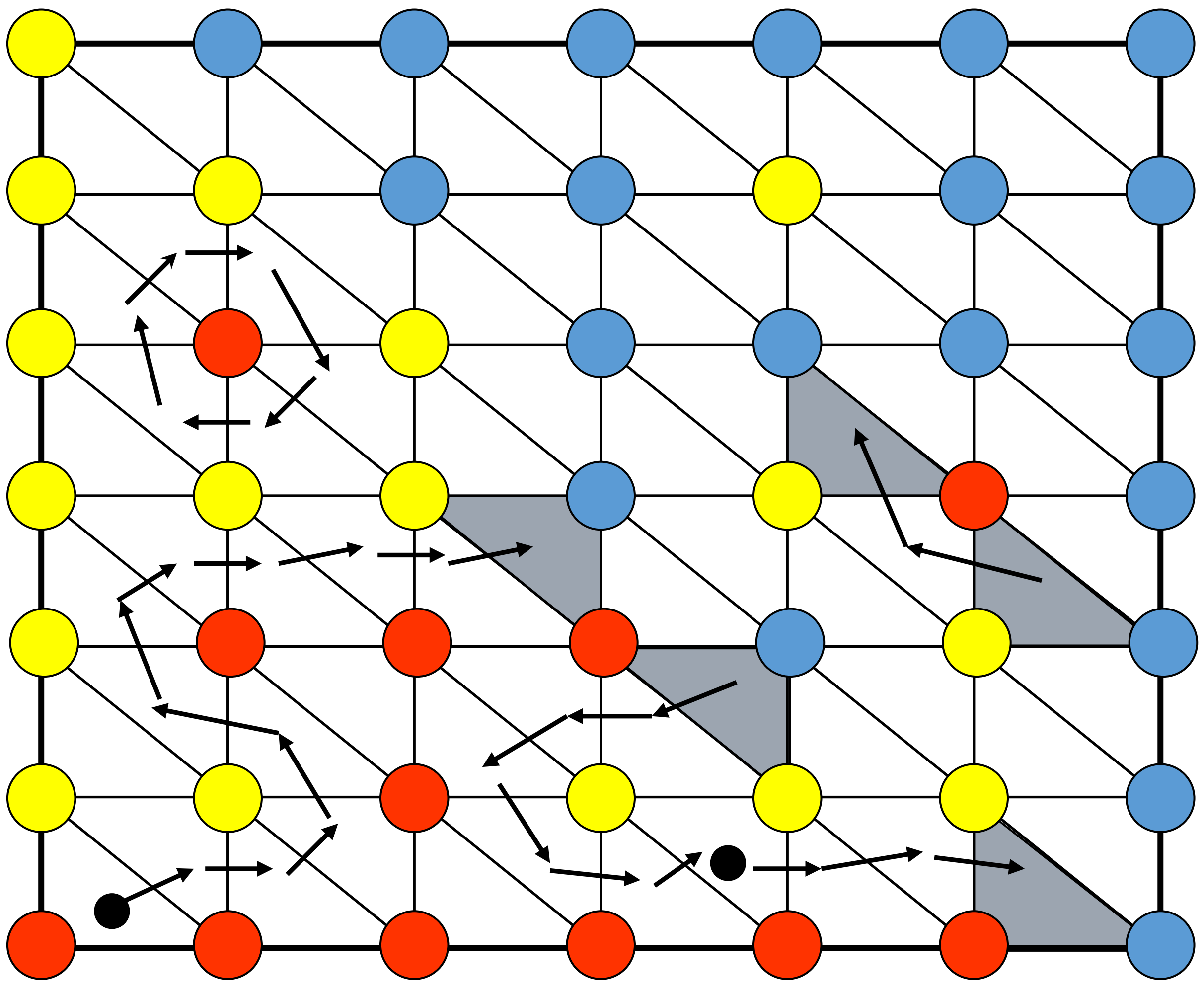
Lemma: No matter how the internal nodes are colored there exists a tri-chromatic triangle. In fact, an odd number of them.

Sperner's Lemma



Lemma: No matter how the internal nodes are colored there exists a tri-chromatic triangle. In fact, an odd number of them.

Sperner's Lemma

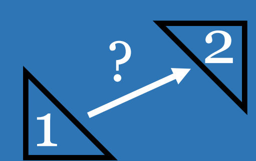


Lemma: No matter how the internal nodes are colored there exists a tri-chromatic triangle. In fact, an odd number of them.

Sperner’s Lemma

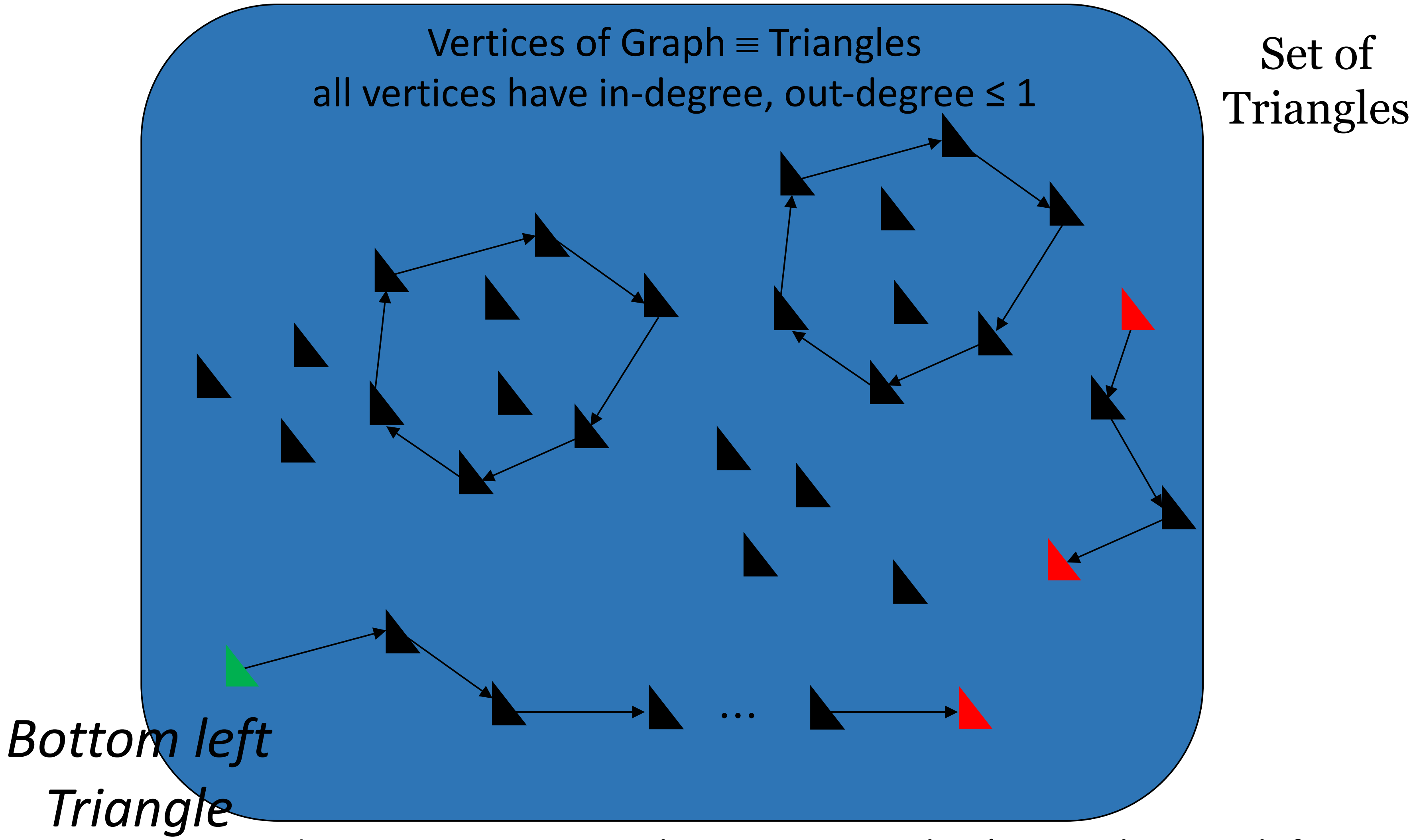
Set of Triangles

Transition Rule: *If ∃ **red** - **yellow** door cross it with **yellow** on your left hand*



Lemma: No matter how the internal nodes are colored there exists a tri-chromatic triangle. In fact, an odd number of them.

Sperner's Lemma



degree 1 vertices: trichromatic triangles (except bottom left triangle)
degree 2 vertices: no blue, non-trichromatic
degree 0 vertices: all other triangles

Proof: ∃ at least one degree 1 node (bottom left triangle) ⇒ ∃ another one (must be trichromatic)
Moreover, b.c. degree 1 nodes come in pairs, ∃ odd number of trichromatic triangles

∴ proof of Sperner's lemma is implied by directed parity argument

The PPAD Complexity Class [Papadimitriou'94]

PPAD: The class of all problems in **NP** with guaranteed solution by dint of the following graph-theoretic lemma:

If a directed graph has an unbalanced node (a node with indegree $\neq$ outdegree), it must have another.

Such problems induce a directed graph G that has an obvious unbalanced node; they require finding another unbalanced node.

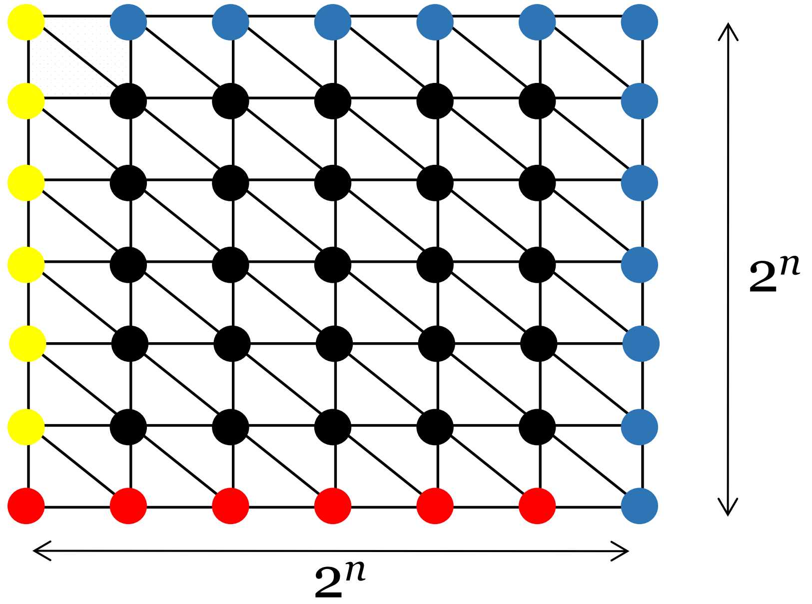
e.g. finding a Sperner triangle is in **PPAD**

But wait a second…given an unbalanced node in a directed graph, why is it not trivial to find another?

By inspecting all nodes?

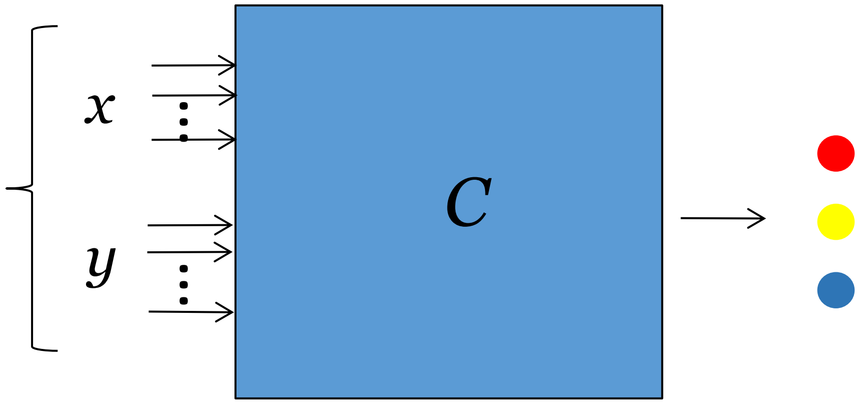
The SPERNER problem (precisely)

Consider square of side 2^n:

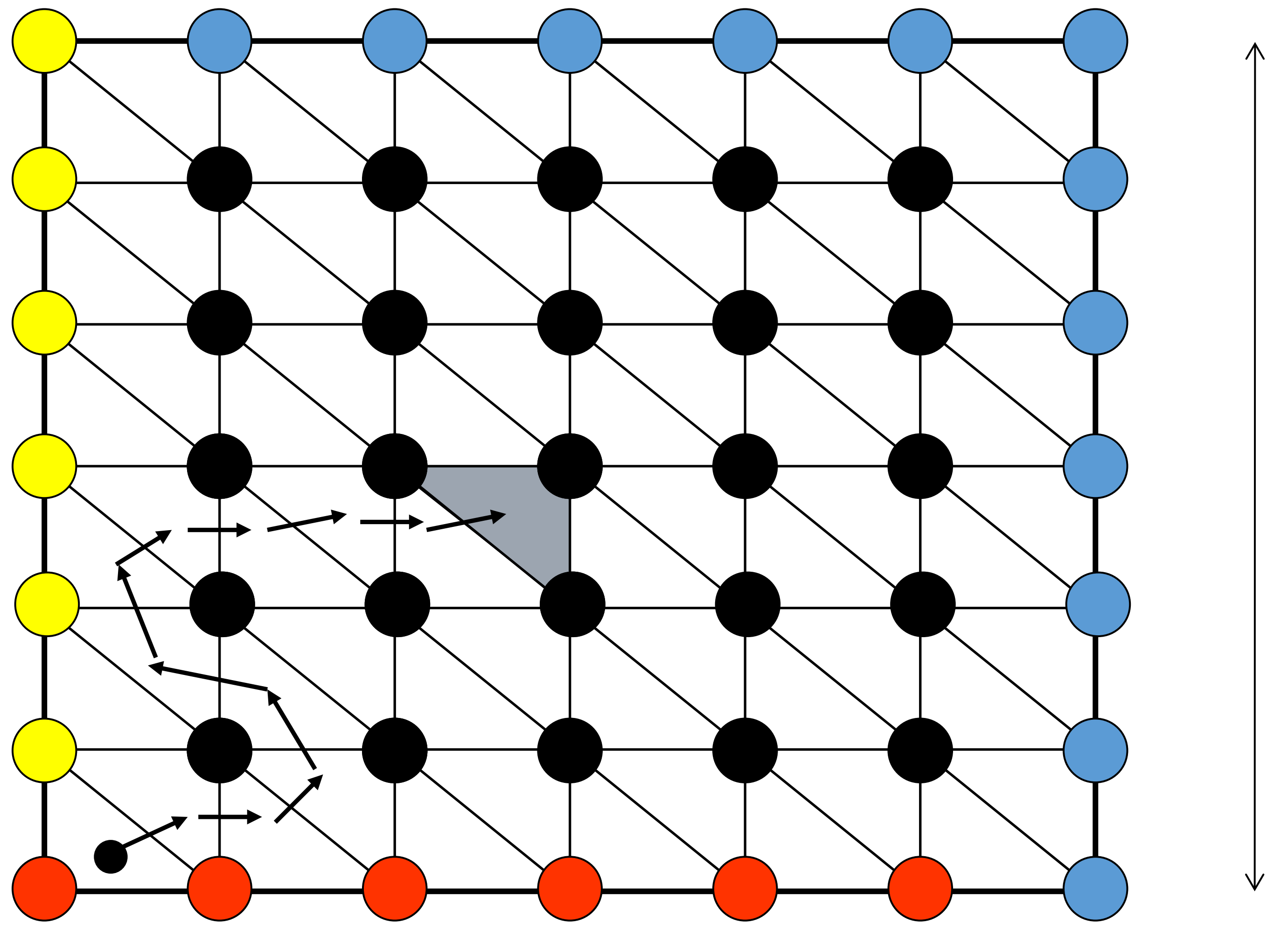


and colors of internal vertices are given by a circuit:

input: the coordinates of a point *(n bits each)*



Solving SPERNER



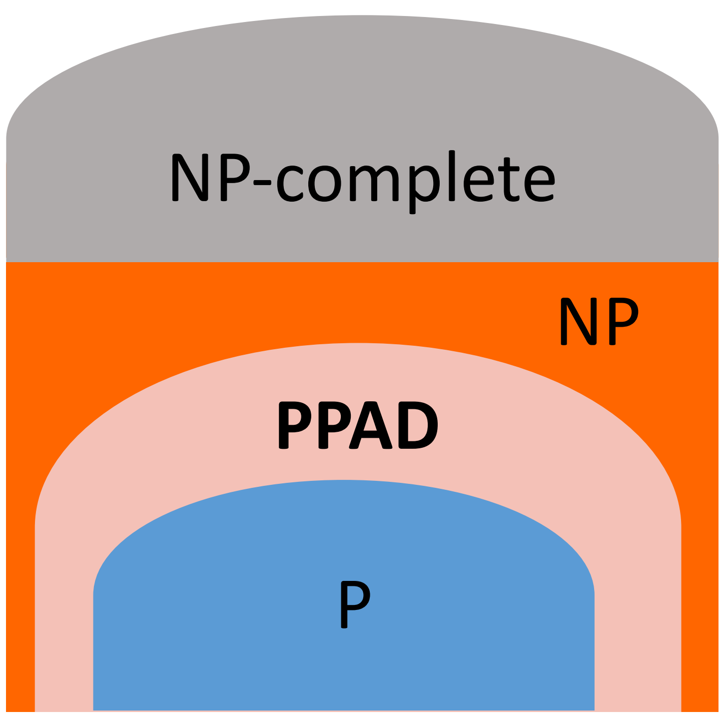
However, the walk may wonder in the box for a long time, before locating the tri-chromatic triangle. Worst-case: 2^{2n}.

The PPAD Complexity Class [Papadimitriou'94]

PPAD: The class of all problems in **NP** with guaranteed solution by dint of the following graph-theoretic lemma:

> *If a directed graph has an unbalanced node (a node with indegree $\neq$ outdegree), it must have another.*

Such problems induce a directed graph *G* (*huge but implicitly defined*), and an obvious unbalanced node *u* of *G; they require finding another unbalanced node.*



The following problems are **PPAD**-complete:

- SPERNER, BROUWER **[Papadimitriou '94; Chen-Deng '05** for 2d**]**
- NASH **[Daskalakis-Goldberg-Papadimitriou'06; Chen-Deng'06** for 2 player games**]**

i.e. these problems are both solvable via the directed parity argument and are at least as hard as any other problem in **PPAD**

The Complexity of GDA Fixed Points

GDA

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \end{bmatrix} \leftarrow \underbrace{\Pi_S\left(\begin{bmatrix} x_t - \eta \cdot \nabla_x f(x_t, y_t) \\ y_t + \eta \cdot \nabla_y f(x_t, y_t) \end{bmatrix}\right)}_{F_{GDA}(x_t, y_t)}$$

Goal: given $\varepsilon, \eta > 0$, description of (or first-order access to) Lipschitz and smooth function f, and projection oracle to S, compute (x^*, y^*) s.t.

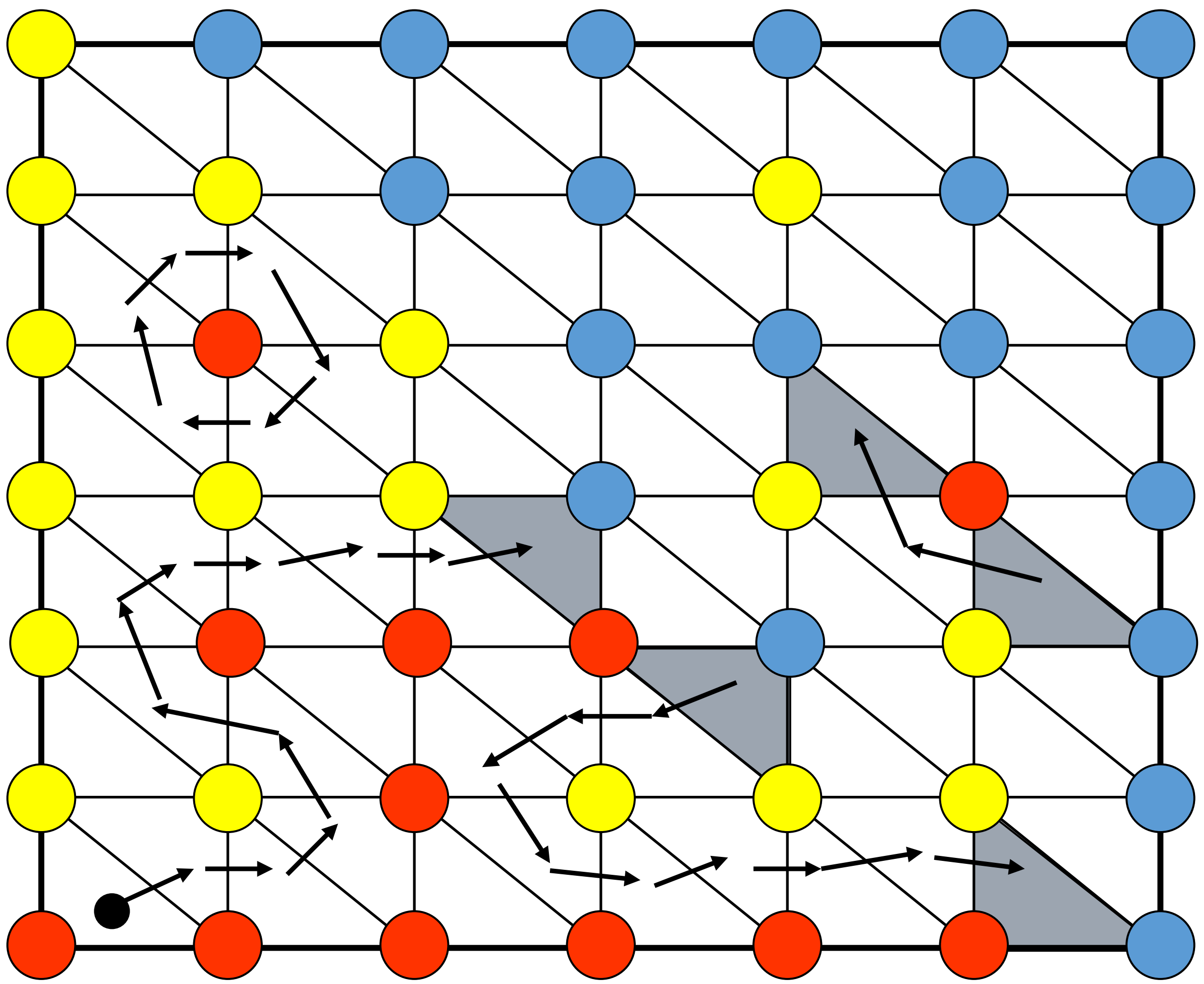
$$\|(x^*, y^*) - F_{GDA}(x^*, y^*)\|_2 \leq \varepsilon$$

[Daskalakis-Skoulakis-Zampetakis'20]: Computing GDA fixed points is PPAD-complete, i.e. exactly as hard as (i) computing general Brouwer fixed points, (ii) computing Nash equilibrium in general games, (iii) any problem in PPAD.

- Interpretation: computing first-order local min-max equilibrium in zero-sum games with nonconvex-nonconcave objectives is exactly as hard as computing Nash equilibrium in general-sum games

Oracle Lower Bound [corollary of our proof]**:** Given oracle access to G-Lipschitz and L-smooth function, f, and its gradient, ∇f, computing ε-approximate fixed point of GDA requires a number of queries to the f and ∇f oracle that is exponential in at least one of $1/\varepsilon$, G, L, dimension.

Rough Proof Idea: Edge-Triangle Game



Reduce an arbitrary instance of SPERNER (which is PPAD-complete) to GDA-FixedPoint, by having the x player choose a triangle, the y player choose an edge of the triangle, and assigning $f(x, y)$ (locally computable given the circuit of SPERNER) depending on whether x chose a triangle that has at least one red-yellow edge, whether y chose a red-yellow edge in that triangle, as well as the orientation of the chosen edge in that triangle.

Best response dynamics traverse paths of the SPERNER graph.

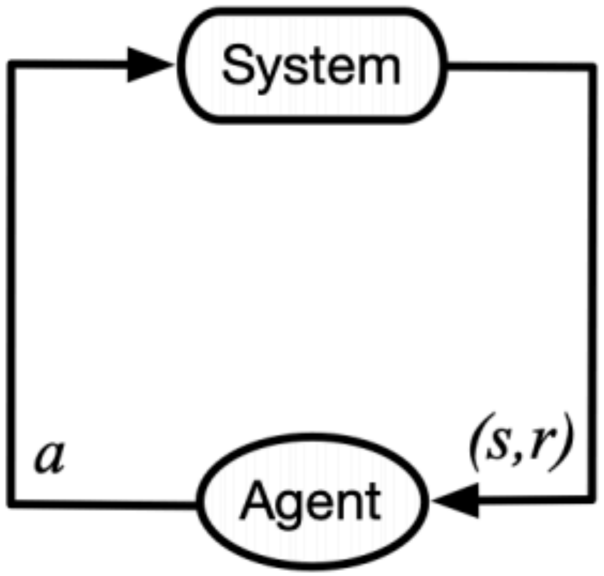
Menu

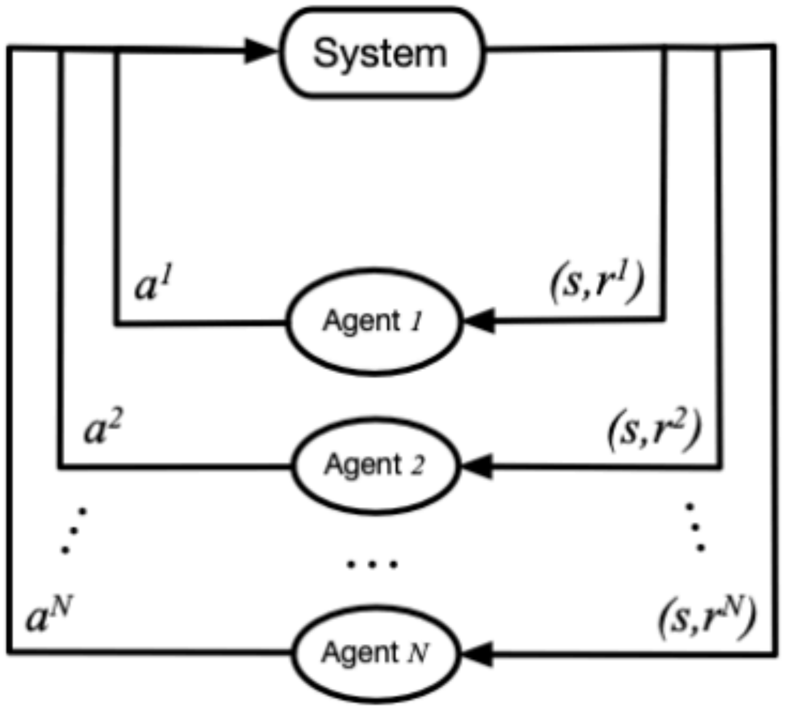
- **Motivation**
- **Part 1: convex-concave objectives**
  - **Background: min-max in Game Theory and Learning**
  - **Removing oscillations via negative momentum**
  - **Tying some loose ends**
- **Part 2: nonconvex-nonconcave objectives**
  - **Some initial musings**
  - **The right solution concept**
  - **Computational complexity of GDA Fixed Points**
  - Convergence under structure: multi-agent RL

Menu

- **Motivation**
- **Part 1: convex-concave objectives**
  - **Background: min-max in Game Theory and Learning**
  - **Removing oscillations via negative momentum**
  - **Tying some loose ends**
- **Part 2: nonconvex-nonconcave objectives**
  - **Some initial musings**
  - **The right solution concept**
  - **Computational complexity of GDA Fixed Points**
  - **Convergence under structure: multi-agent RL**

Reinforcement Learning: Single vs Multi





Markov Decision Process

$$s_{t+1} \sim P(\cdot \mid s_t, a_t)$$

$$r_{t+1} = R(s_t, a_t, s_{t+1})$$

Agent's policy π: specifies $P[a|s]$

Agent goal: $\max_{\pi} V_\pi$ where

$$V_\pi = \mathbb{E}_{\substack{s_0 \sim P_0 \\ a_t \sim \pi(\cdot|s_t) \\ s_{t+1} \sim P(\cdot|s_t, a_t)}} \left[\sum_{t\geq 0} R(s_t, a_t, s_{t+1})\right]$$

Stochastic Game **[Shapley'53, Littman'94]**

$$s_{t+1} \sim P(\cdot \mid s_t, a_t^1, \ldots, a_t^N)$$

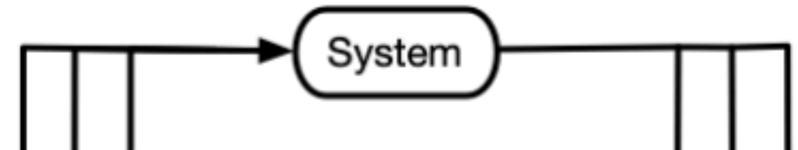
$$r_{t+1}^i = R^i(s_t, a_t^1, \ldots, a_t^N, s_{t+1})$$

Agent i's policy π^i: specifies $P\left[a^i \middle| s\right]$

Agent i's goal: $\max_{\pi^i} V^i_{\pi^i, \pi^{-i}}$ where

$$V^i_{\pi^i, \pi^{-i}} = \mathbb{E}_{\substack{s_0 \sim P_0 \\ a_t^j \sim \pi^j(\cdot|s_t), \forall j \\ s_{t+1} \sim P(\cdot|s_t, \vec{a}_t)}} \left[\sum_{t\geq 0} R^i(s_t, \vec{a}_t, s_{t+1})\right]$$

Reinforcement Learning: Single vs Multi



Nash Equilibrium: Collection of policies $\pi^1, \ldots, \pi^N$ such that for all i: $V^i_{\pi^i,\pi^{-i}} \geq V^i_{\pi',\pi^{-i}}$, for all π'

Exists under mild conditions (roughly that the game stops in finite time with probability 1)

When N=1, this is equivalent to optimal policy of MDP

Markov Decision Process

$$s_{t+1} \sim P(\cdot\,|s_t, a_t)$$

$$r_{t+1} = R(s_t, a_t, s_{t+1})$$

Agent's policy π: specifies $P[a|s]$

Agent goal: $\max_{\pi} V_\pi$ where

$$V_\pi = \mathbb{E}_{\substack{s_0 \sim P_0 \\ a_t \sim \pi(\cdot|s_t) \\ s_{t+1} \sim P(\cdot|s_t, a_t)}} \left[\sum_{t \geq 0} R(s_t, a_t, s_{t+1})\right]$$

Stochastic Game **[Shapley'53, Littman'94]**

$$s_{t+1} \sim P(\cdot\,|s_t, a_t^1, \ldots, a_t^N)$$

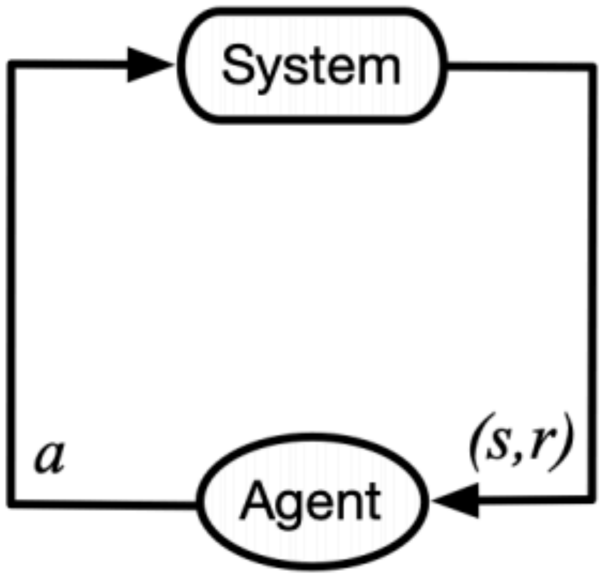
$$r^i_{t+1} = R^i(s_t, a_t^1, \ldots, a_t^N, s_{t+1})$$

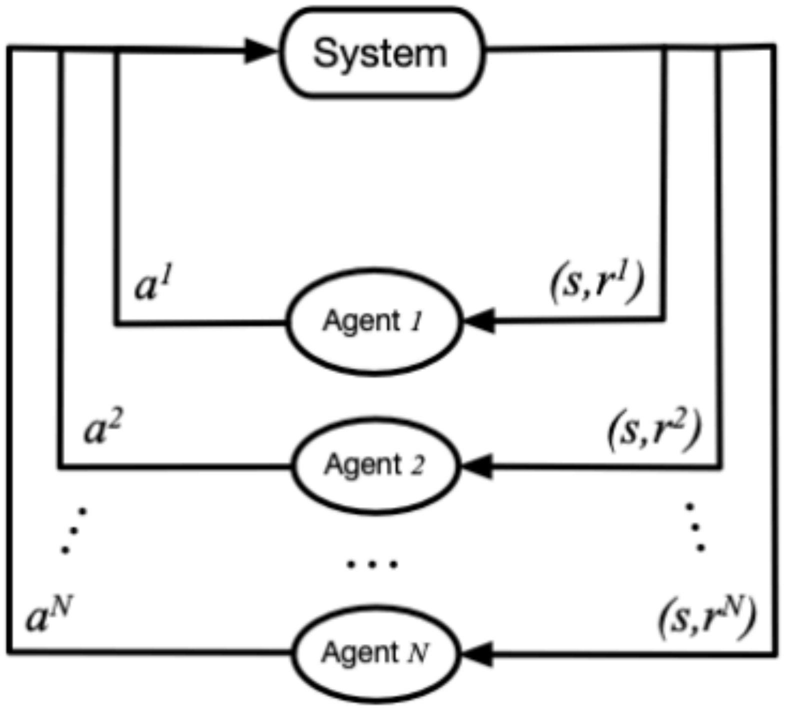
Agent i's policy π^i: specifies $P\left[a^i \middle| s\right]$

Agent i's goal: $\max_{\pi^i} V^i_{\pi^i,\pi^{-i}}$ where

$$V^i_{\pi^i,\pi^{-i}} = \mathbb{E}_{\substack{s_0 \sim P_0 \\ a_t^j \sim \pi^j(\cdot|s_t), \forall j \\ s_{t+1} \sim P(\cdot|s_t, \vec{a}_t)}} \left[\sum_{t \geq 0} R^i(s_t, \vec{a}_t, s_{t+1})\right]$$

Reinforcement Learning: Algorithms





Learning Optimal Policies:
[lots of work]

- Q-Learning
- Policy Iteration
- Policy Gradient Methods
- …

[Jin-AllenZhu-Bubeck-Jordan'18, Agarwal-Kakade-Lee-Mahajan'20]

Learning Nash Equilibrium Policies:
[lots of work] but

- In general, harder than Nash equilibrium in stateless games (already PPAD-hard)
- Typically Algorithms are *centralized,* i.e. a central optimizer controls all agents' policies
- Survey **[Zhang-Yang-Basar'19]**
- How about 2-player zero-sum, independent learning?

2-Player Zero-Sum RL

Upshot [Shapley'53]: min-max theorem holds, i.e.

$$\min_{\pi^1} \max_{\pi^2} V^2_{\pi^1,\pi^2} = \max_{\pi^2} \min_{\pi^1} V^2_{\pi^1,\pi^2}$$

Interesting New Feature: $V^2_{\pi^1,\pi^2}$ is not necessarily convex in π^1, concave in π^2

Yet there is lots of structure

[w/ Noah Golowich, Dylan Foster ongoing]: we show a *gradient domination condition* implying that near stationary points are equilibria

we show that (two-timescale) policy gradient converges to Nash equilibrium, i.e. if each player uses policy gradient method to update their policy independently of the other player (using own action, state, reward trajectories) the resulting trajectory over policies converges to min-max equilibrium (not last iterate, but stronger than average iterate)

Conclusions

- Min-max optimization and equilibrium computation are intimately related to the foundations of Game Theory, Mathematical Programming, and Online Learning Theory
- They have also found profound applications in Statistics, Complexity Theory, and many other fields
- Applications in Machine Learning pose big challenges due to the dimensionality and non-convexity of the problems, as well as the entanglement of decisions with learning
- I expect such applications to explode, going forward, as ML turns more to multi-agent learning applications, and (indirectly) as ML models become more complex and harder to interpret
    - in such settings, learning agents will be introduced to check on the behavior of other learning agents

Thank you!