Two-player Games in the Era of Machine Learning

Gauthier Gidel
Mila and DIRO

Columbia University

ENS
École Normale Supérieure

Inria
Inventeurs du monde numérique

Mila

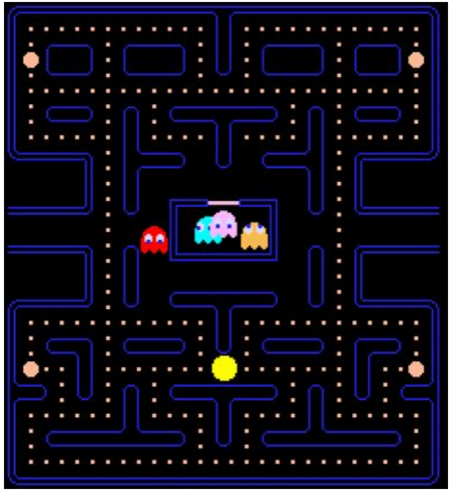
Element AI

DIRO

Université de Montréal

We live in a world full of games

Single player:



Notion of performance fully specified by the environment

Multi-player:

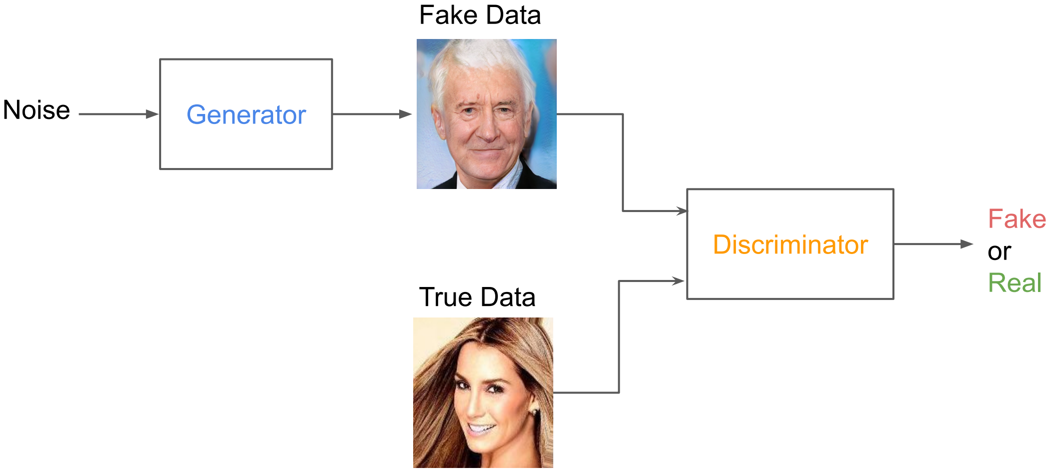
Notion of performance depends on the **opponent(s)**

Games specifically designed for Machine learning purposes

For Generative modeling:

Generative Adversarial Networks
[Goodfellow et al. 2014]

Picture: [Wu et al. 2020]



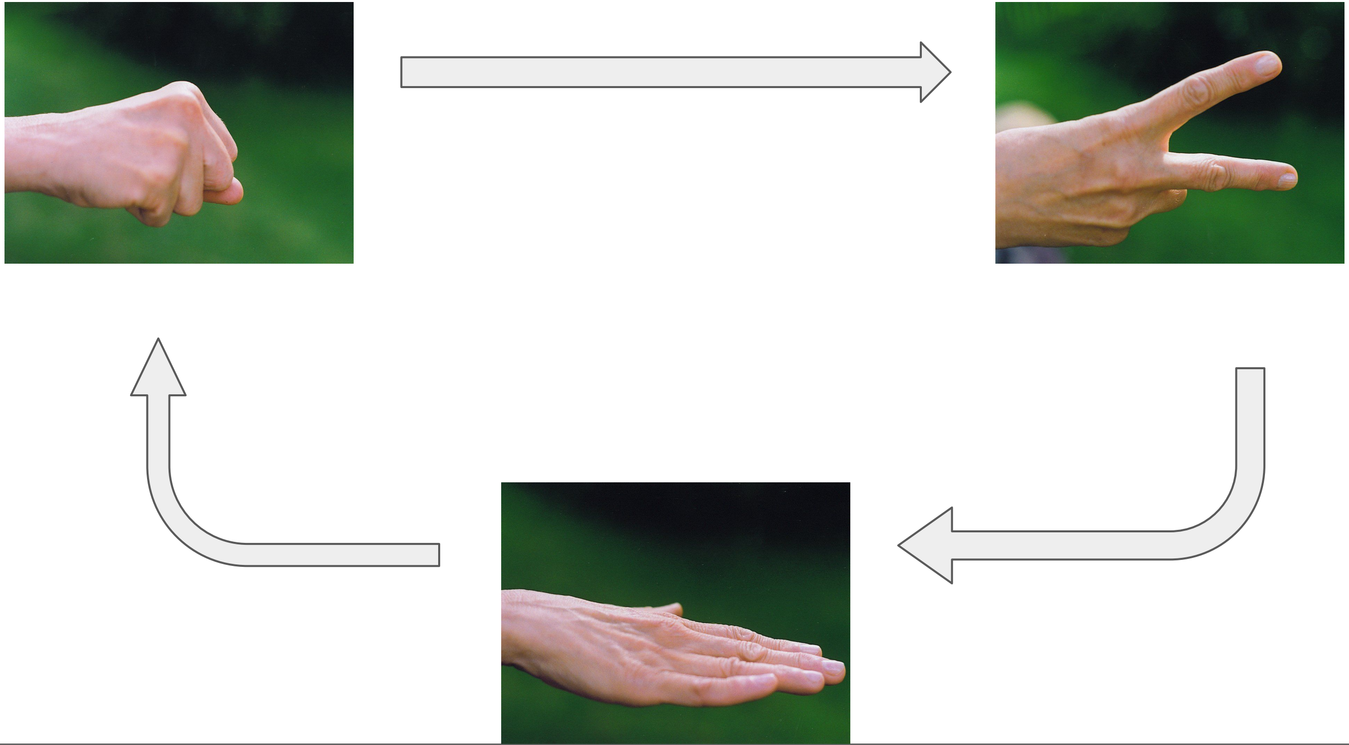
Games are a great tool to learn complex notions

Games are a great tool to learn complex notions

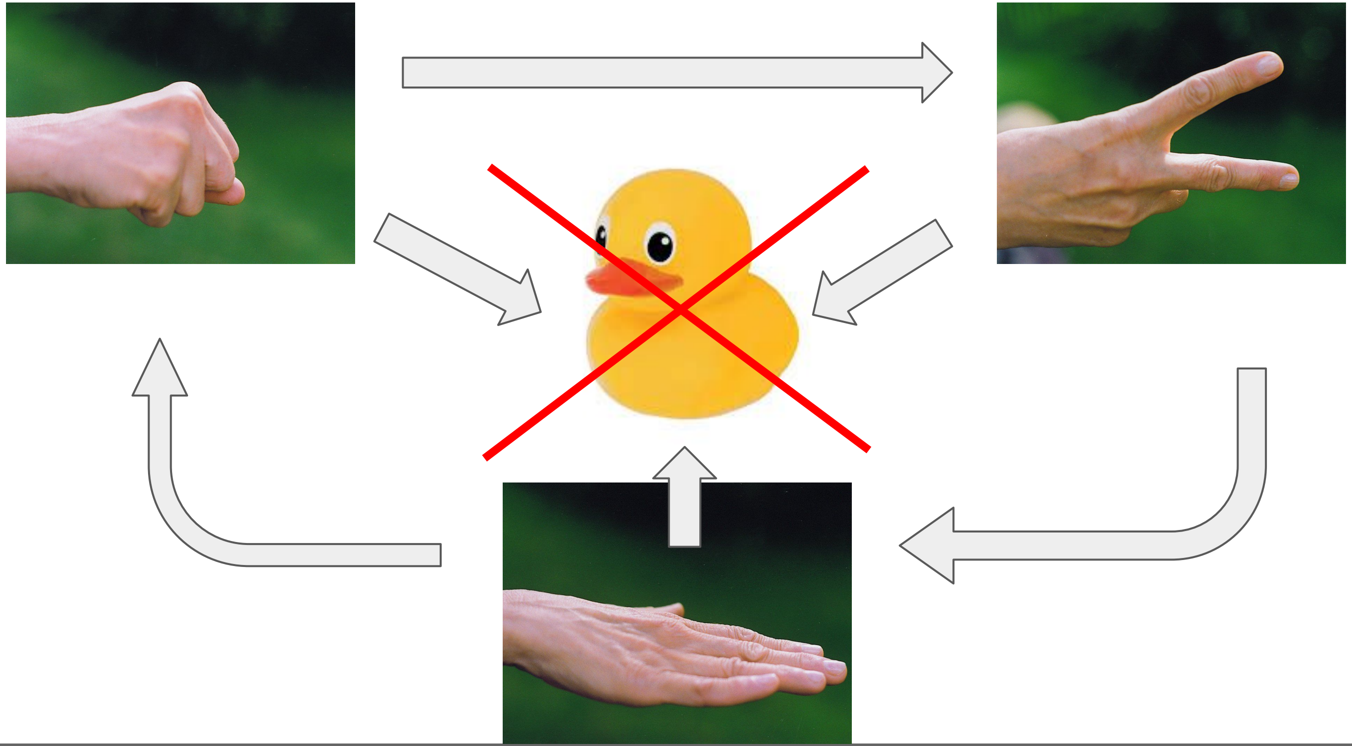
Multi-player games are notoriously challenging to train. [Goodfellow, 2016, Nowozin et al., 2016; Arjovsky et al., 2017].

The learning target is harder to define.

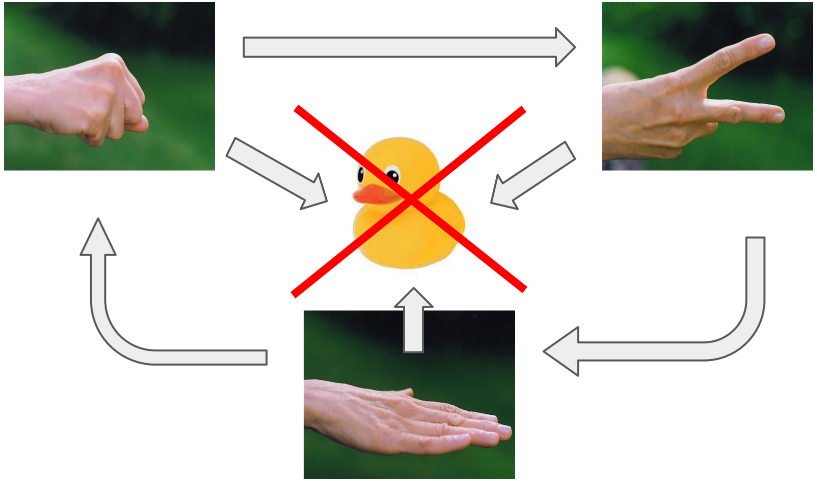
Problem 1: is there a ‘best’ strategy?



Problem 1: is there a ‘best’ strategy?



Problem 1 : is there a ‘best’ strategy?



There is **no best single strategy**.

But there is a best **distribution**

⇩

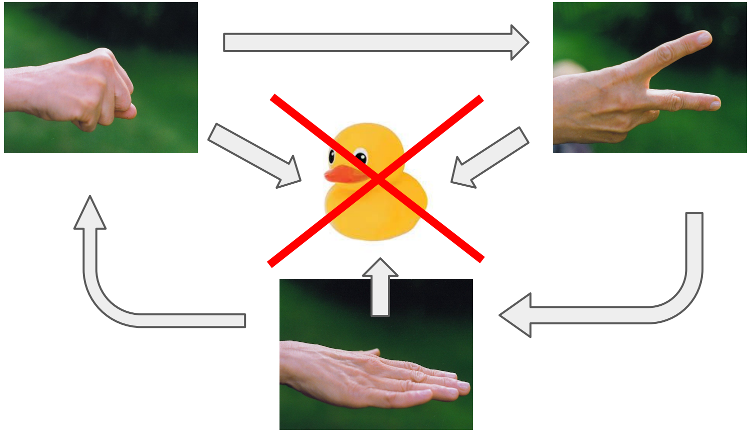
Mixed equilibrium

Mixed Strategies are necessary to play games!!!

Starcraft II

[Vinyals et al. 2019]
(Picture from DeepMind's Blog post)

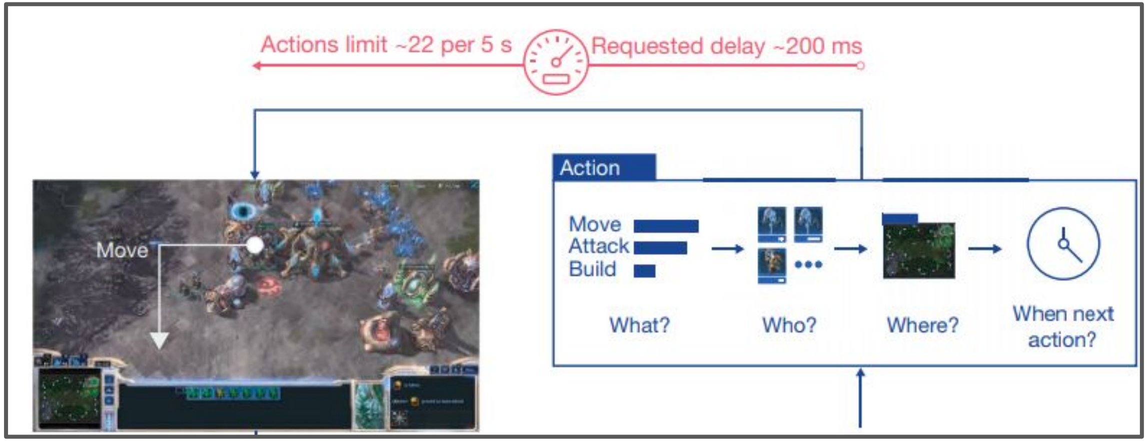
$\approx$



Problem: too many pure strategies to naively consider distributions over strategies (mixed-strategies).

In RL: Pure strategy == deterministic policy
Mixed strategy == stochastic policy

Mixed Strategies are necessary to play games!!!



Picture from [Vinyals et al. 2019]

Problem: we have a **limited capacity**: (we cannot represent some pure or mixed-strategy) ⇨ It changes the (best) way to play the game.

Limited capacity (constraints no imposed by the rules):

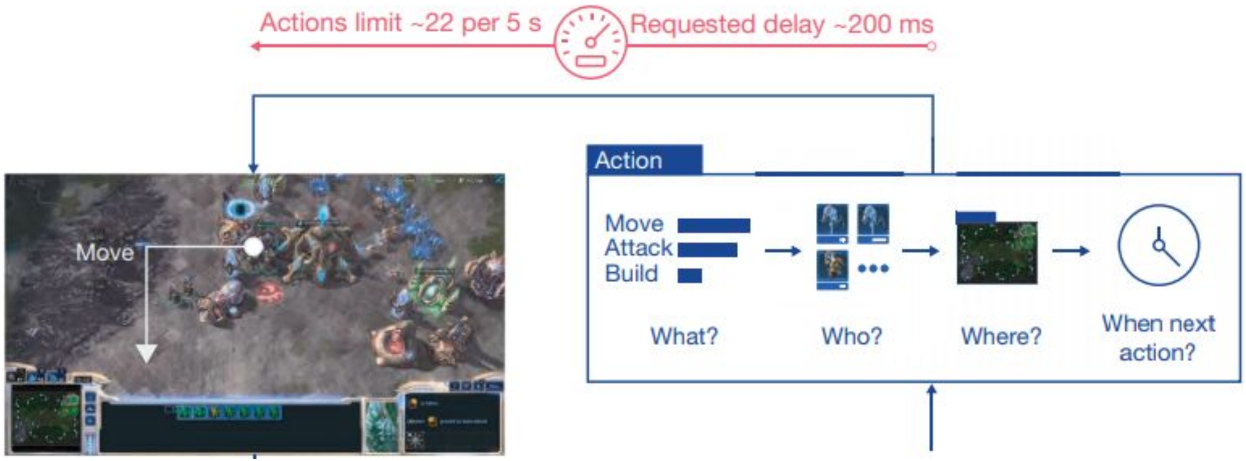
- Physical limitation for the number of action per minute.
- Neural networks cannot represent any function.

Outline

1. **Latent games:** how to leverage function approximation to play games.
2. **Game Optimization:** what are the potential difficulties arising.
3. **The landscape of games:** an empirical study of practical landscapes.
4. **Future Work:** Design of new adversarial formulation for ML.

Outline

1. **Latent games:** how to leverage function approximation to play games.
2. **Game Optim**
3. **The landsca** pes.
4. **Future Work**



Minimax Theorems for Latent games:

or how I learned to stop worrying about mixed-Nash and love neural nets

Gauthier Gidel, David Balduzzi, Wojciech Czarnecki, Marta Garnelo and Yoram Bachrach,

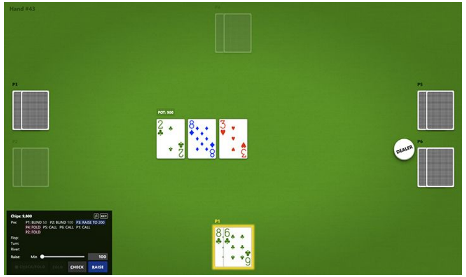
arXiv 2020

Work under review done during an internship at ***DeepMind*** *London*

Lay of the land

Many recent successes to solve what the ML community call (two-player) games:

Poker



[Brown and Sandholm 2018]
(Picture from FAIR's Blog post)

Starcraft II

[Vinyals et al. 2019]
(Picture from DeepMind's Blog post)

Generative Adversarial Nets

[Wu et al. 2020]

Using neural networks

Lay of the land

Theoretical focus (what is our goal)

- Game theory: “one must consider **mixed strategy**”.

- **Previous game theoretic papers** on GANs consider the **networks** as **pure-strategies**: [Arora et al., 2017; Oliehoek et al., 2018; Grnarova et al., 2018; Hsieh et al., 2019]

$$\varphi(\psi, G) := \mathbb{E}_{x\sim data}[\ln(\psi(x))] + \mathbb{E}_{z\sim\mathcal{N}(0,I_d)}[\ln(1-\psi(G(z))]$$

Mixture of networks == distribution over weights (not practical)
In practice: correspond to finite collection of models (**very** costly)

Lay of the land

Theoretical focus (what is our goal)

- Game theory: “one must consider **mixed strategy**”.
- **Previous g** …e GANs consider the **networks** as **pure-strat** …018; Hsieh et al., 2019]

$$\varphi(\psi, G) := \mathbb{E}_{x\sim dat} \quad - \psi(G(z))]$$

Practitioners always used a **single pair (not a collection)** of networks to achieve state-of-the-art (SOTA) results [Brock et al., 2019, Wu et al., 2020]

Mixture of networks == distribution over weights (not practical)
In practice: correspond to finite collection of models (**very** costly)

Bridging the gap between theory and practice

Theoretical focus: can we achieve an equilibrium with a **single pair** of agents???

Previous work:

1. No theoretical work except on GANs. [Arora et al., 2017; Oliehoek et al., 2018; Grnarova et al., 2018; Hsieh et al., 2019]

Our contributions:

1. Unify **“real world games”** (Poker, Starcraft) and **machine learning games** (GANs)

Bridging the gap between theory and practice

Theoretical focus: can we achieve an equilibrium with a **single pair** of agents???

Previous work:

1. No theoretical work except on GANs. [Arora et al., 2017; Oliehoek et al., 2018; Grnarova et al., 2018; Hsieh et al., 2019]

2. Theoretical work on GANs considered **networks** as **pure strategies.**

Our contributions:

1. Unify **“real world games”** (Poker, Starcraft) and **machine learning games** (GANs)

2. Propose a way to see **networks** directly as **mixed-strategies.**

Bridging the gap between theory and practice

Theoretical focus: can we achieve an equilibrium with a **single pair** of agents???

Previous work:

1. No theoretical work except on GANs. [Arora et al., 2017; Oliehoek et al., 2018; Grnarova et al., 2018; Hsieh et al., 2019]
2. Theoretical work on GANs considered **networks** as **pure strategies.**
3. Advocating in practice for a **collection of weights.** (very costly)

Our contributions:

1. Unify **"real world games"** (Poker, Starcraft) and **machine learning games** (GANs)
2. Propose a way to see **networks** directly as **mixed-strategies.**
3. Definition of game/equilibrium that **take into account the practical considerations** (finite capacity and single pair of network):

Bridging the gap between theory and practice

Theoretical focus: can we achieve an equilibrium with a **single pair** of agents???

Previous work:

1. No theoretical work except on GANs. [Arora et al., 2017; Oliehoek et al., 2018; Grnarova et al., 2018; Hsieh et al., 2019]
2. Theoretical work on GANs considered **networks** as **pure strategies.**
3. Advocating in practice for a **collection of weights.** (very costly)
4. Unable to explain why a single pair of networks achieve SOTA.

Our contributions:

1. Unify **"real world games"** (Poker, Starcraft) and **machine learning games** (GANs)
2. Propose a way to see **networks** directly as **mixed-strategies.**
3. Definition of game/equilibrium that **take into account the practical considerations** (finite capacity and single pair of network):
4. Proof that one can reach and **approximate equilibrium with a single pair of networks**.

Structure of the section:

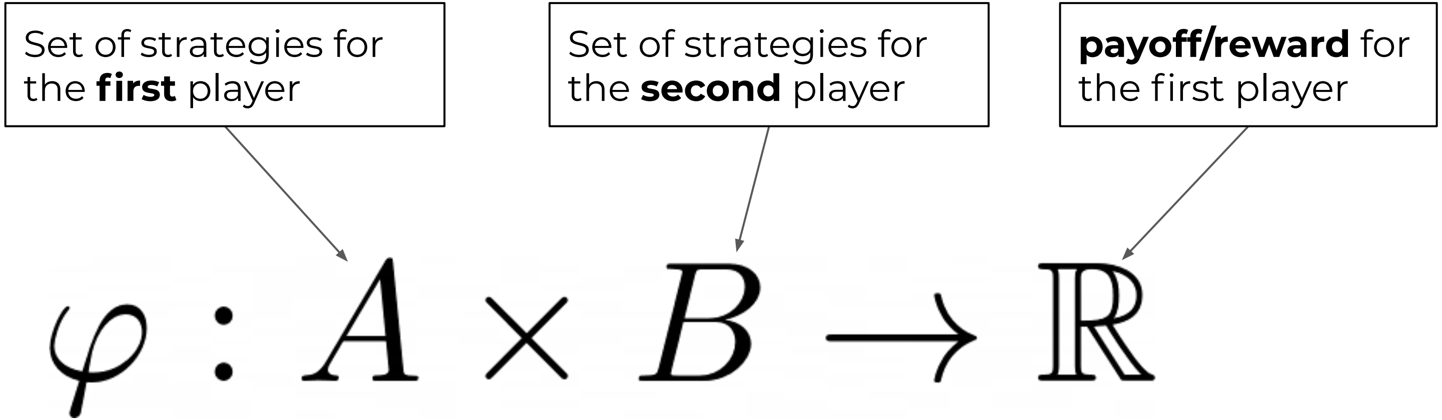
1. Definition of a game (need for mixed strategies)

2. GAN example: represent mixed-strategies with function (neural networks)

3. Generalization to any game!

4. Using these function we can define a **new** concept of equilibrium (limited to the representable mixed-strategies)

A (zero-sum) game, what is this?



Set of strategies for the **first** player

Set of strategies for the **second** player

payoff/reward for the first player

$$\varphi : A \times B \to \mathbb{R}$$

How to reach an equilibrium ?

Solution: play strategies **randomly**

$$\varphi(p, q) = \underbrace{\mathbb{E}_{a \sim p, b \sim q}[\varphi(a, b)]}_{\text{Average payoff}}$$

Probability distributions over strategies

How to reach an equilibrium ?

Fundamental result of game theory
[von Neumann, 1928]:

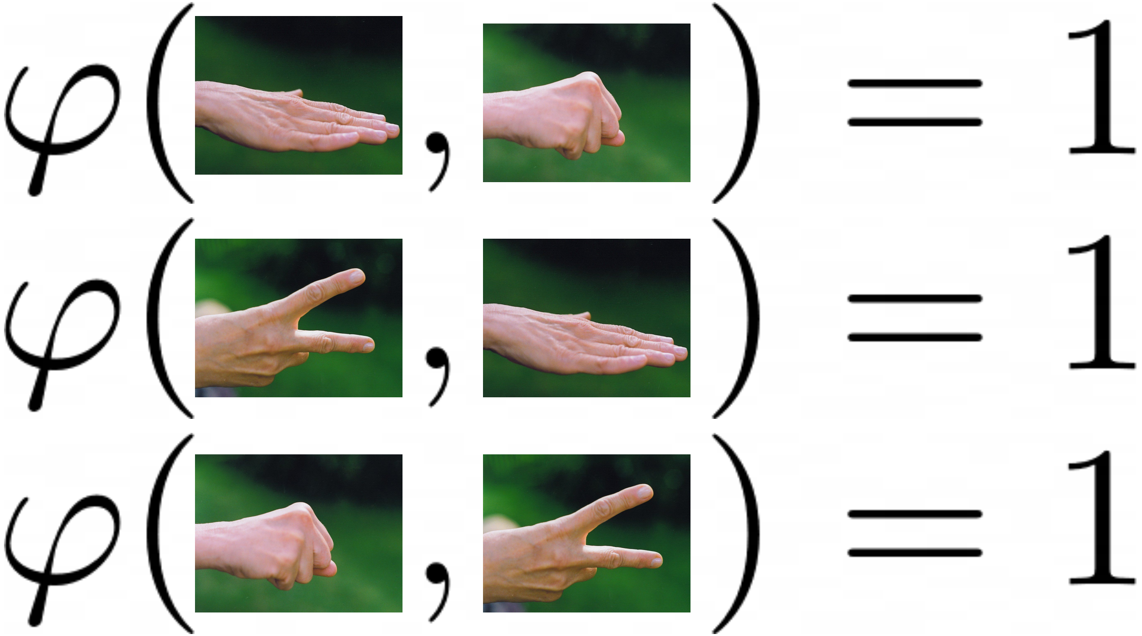
By playing **mixed-strategy** one can achieve an equilibrium.

Not limited capacity !!!

$$\varphi(p, q) = \underbrace{\mathbb{E}_{a\sim p, b\sim q}[\varphi(a, b)]}_{\text{Average payoff}}$$

Any probability distributions over strategies

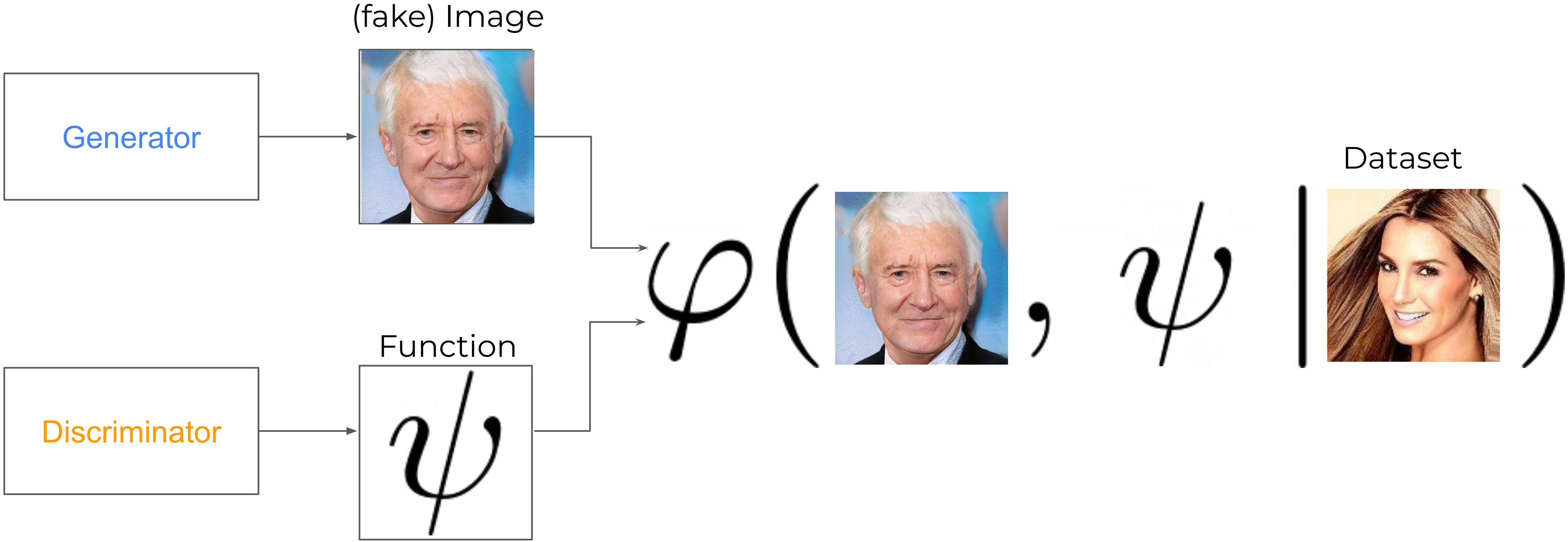
Example: Rock-Paper-Scissors



In that particular example:

- antisymmetric cost
- Winning == 1
- Losing == -1
- Tying == 0
- Zero-sum games are more general

First contribution: (Naive) GANs



(fake) Image

Generator

Function

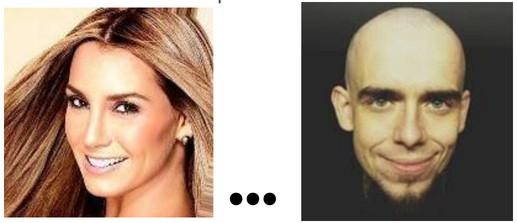
Discriminator

Dataset

First contribution: Payoff of (Naive) GANs

Convention: 0 is “fake” and 1 is “real”

$$\varphi(x,\psi) := \underbrace{\mathbb{E}_{x'\sim data}[\ln\psi(x')]}_{\text{How well the }\textbf{dataset}\text{ is classified as }\textbf{“real”}} + \underbrace{\ln(1-\psi(x))}_{\text{How well the }\textbf{fake image}\text{ is classified as }\textbf{“fake”}}$$



First contribution: Payoff of (Naive) GANs

Going from pure-strategy to mixed-strategies: $x \sim p_G$

$$\varphi(p_G, \psi) := \underbrace{\mathbb{E}_{x' \sim data}[\ln \psi(x')]} + \underbrace{\mathbb{E}_{x \sim p_G}[\ln(1 - \psi(x))]}$$

How well the **dataset** is classified as **“real”**

•••

How well the **fake image** is classified as **“fake”**

Payoff of (Naive) GANs

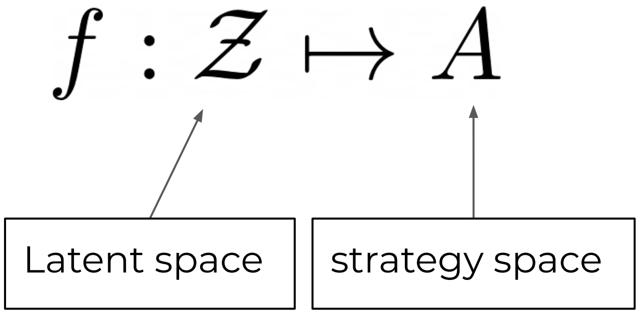
$$\varphi(p_G, \psi) := \mathbb{E}_{x' \sim data}[\ln \psi(x')] + \mathbb{E}_{x \sim p_G}[\ln(1 - \psi(x))]$$

Fact: The Generator **correspond** to a **mixed strategy.**

$$\varphi(G, \psi) := \underbrace{\mathbb{E}_{x' \sim data}[\ln \psi(x')]}_{\text{How well the } \textbf{dataset} \text{ is classified as } \textbf{“real”}} + \underbrace{\mathbb{E}_{z \sim \mathcal{N}(0, I_d)}[\ln(1 - \psi(G(z)))]}_{\text{How well the } \textbf{fake image} \text{ is classified as } \textbf{“fake”}}$$

...

Idea: use function approximation to construct mixtures of strategie



$$f : \mathcal{Z} \mapsto A$$

Latent space

strategy space

$$f(z) = a \sim p_f \, , \; z \sim \pi$$

We can use functions to **represents** a **distribution (i.e. a mixed strategy)!!!**

Idea: use function approximation to construct mixtures of strategie

$$f : \mathcal{Z} \mapsto A$$

Latent space

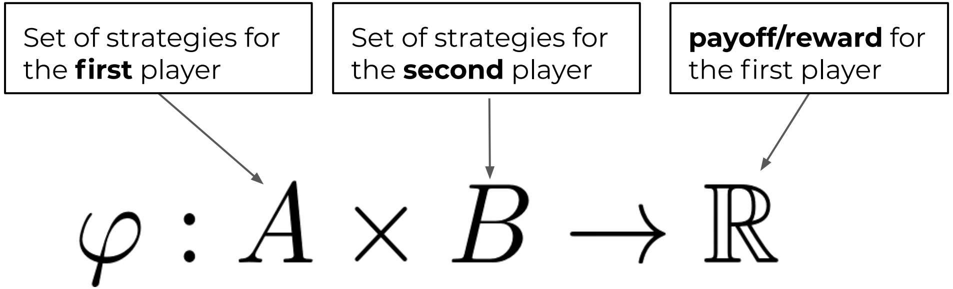
strategy space

Normal or uniform distribution

$$f(z) = a \sim p_f \; , \; z \sim \pi$$

We can use functions to **represents** a **distribution (i.e. a mixed strategy)!!!**

$$\varphi(f, g) = \varphi(p_f, p_g) = \mathbb{E}_{z \sim \pi, z' \sim p'}[\varphi(f(z), g(z'))]$$



Set of strategies for the **first** player

Set of strategies for the **second** player

payoff/reward for the first player

$$\varphi : A \times B \to \mathbb{R}$$

Standard game theory:

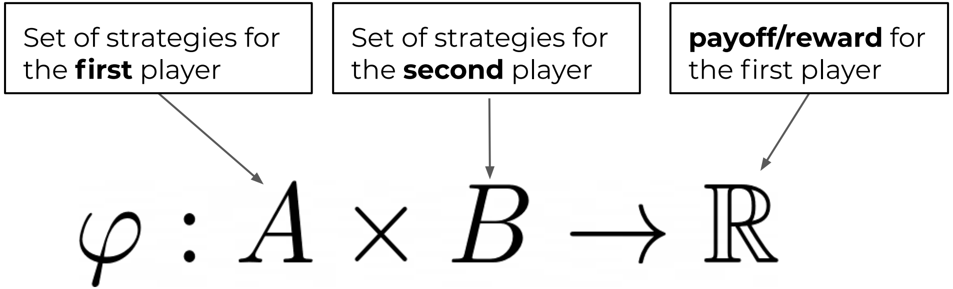
- Consider distributions over A and B. (mixed strategies)

$$\varphi(p, q) = \mathbb{E}_{a\sim p, b\sim q}[\varphi(a, b)]$$

Latent games theory:

- Consider distributions encoded by **limited capacity functions.**

$$\varphi(f, g) := \mathbb{E}_{z\sim\pi, z'\sim\pi'}[\varphi(f(z), g(z'))]$$



Set of strategies for the **first** player

Set of strategies for the **second** player

payoff/reward for the first player

$$\varphi : A \times B \to \mathbb{R}$$

Standard game theory:

- Consider distributions over A and B. (mixed strategies)

$$\varphi(p, q) = \mathbb{E}_{a\sim p, b\sim q}[\varphi(a, b)]$$

- When A infinite (or large), distribution space over A **infinite dimensional!!!**

Latent games theory:

- Consider distributions encoded by **limited capacity functions.**

$$\varphi(f, g) := \mathbb{E}_{z\sim\pi, z'\sim\pi'}[\varphi(f(z), g(z'))]$$

- Tractable even when A infinite.

Set of strategies for the **first** player

Set of strategies for the **second** player

payoff/reward for the first player

$$\varphi : A \times B \to \mathbb{R}$$

Standard game theory:

- Consider distributions over A and B. (mixed strategies)

$$\varphi(p,q) = \mathbb{E}_{a \sim p, b \sim q}[\varphi(a,b)]$$

- When A infinite (or large), distribution space over A **infinite dimensional!!!**
- Not practical

Latent games theory:

- Consider distributions encoded by **limited capacity functions.**

$$\varphi(f,g) := \mathbb{E}_{z \sim \pi, z' \sim \pi'}[\varphi(f(z), g(z'))]$$

- Tractable even when A infinite.
- Correspond to practical GANs
- Extend to any games.

Question: Can we extend von Neumann's Theorem to Latent games?

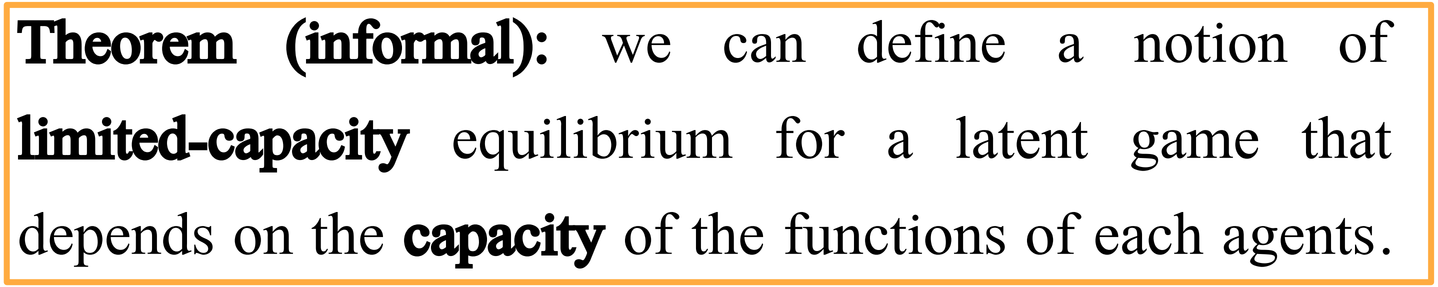
Answer: Yes!

And it provides the notion of a limited capacity equilibrium.

In a **latent game**, the agents leverage **function approximation** to play **mixed strategies**

- Related to the RL policies used to play StarCraft II [Vinyals et al. 2019]
- Related to GANs generators [Goodfellow et al. 2014].
- General and flexible framework that aim to explain why neural nets achieve to approximate equilibria in complex games.

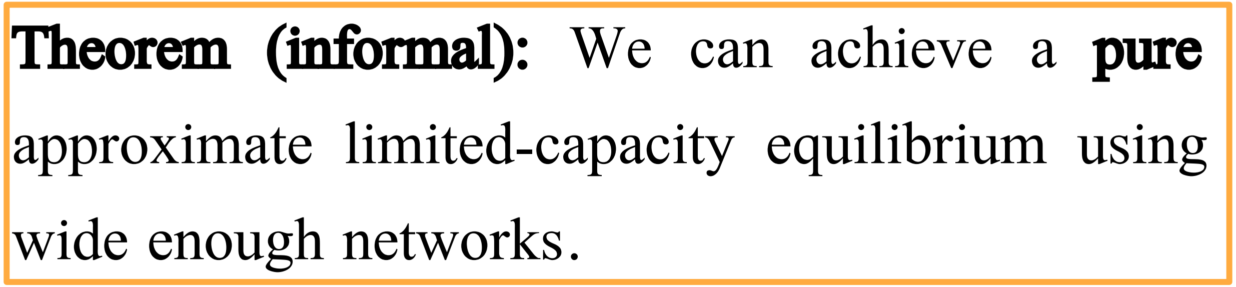
- Each agent updates their function for a **given architecture**.
- **Limited capacity** to play the game.



Theorem (informal): we can define a notion of **limited-capacity** equilibrium for a latent game that depends on the **capacity** of the functions of each agents.

- **Differs** from the **Nash of the game** (unlimited capacity equilibrium)

Achieving Pure-Nash with Neural Nets

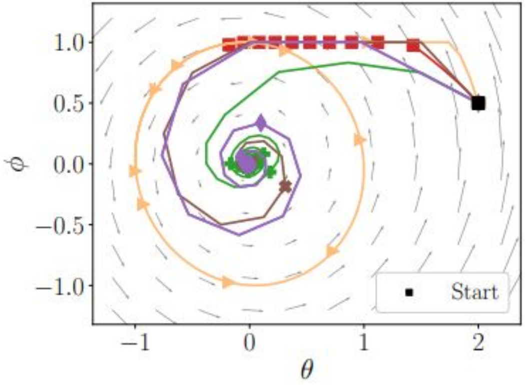


Theorem (informal): We can achieve a **pure** approximate limited-capacity equilibrium using wide enough networks.

Takeaway: This result **bridges the gap** between **theory** and **practice.**

Outline

1. **Latent games:** how to leverage function approximation to play games.
2. **Game Optimization:** what are the potential difficulties arising.
3. **The landscape of games:** an empirical capes.
4. **Future Work:** Design of new adversarial



A Variational Inequality Perspective on GANs

Gauthier Gidel*, Hugo Berard*, Gaëtan Vignoud, Pascal Vincent, Simon Lacoste-Julien

**equal contribution work presented at ICLR 2019*

facebook Artificial Intelligence Research

Inria INVENTEURS DU MONDE NUMÉRIQUE

Université de Montréal

Mila

Game training is ~~hard~~ fascinating !

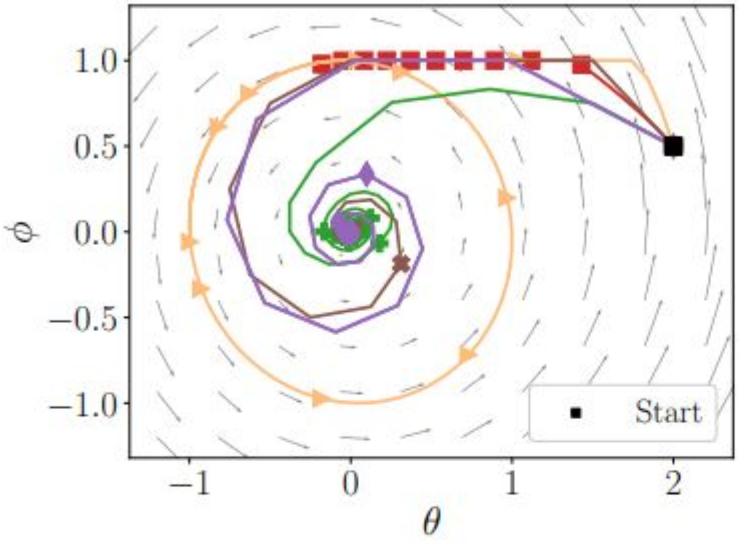
Minimax Training is ~~hard~~ fascinating

Rediscovery of the failure of gradient method in games [Goodfellow, 2016]

Example: WGAN with **linear** discriminator and generator [Mescederer et al., 2018]

$$\min_{\theta} \max_{\phi, ||f_\phi||_L \leq 1} \phi^T \mathbb{E}_{x \sim p_{\mathcal{D}}}[x] - \phi^T \theta \mathbb{E}_{z \sim p_{\mathcal{Z}}}[z]$$

Bilinear saddle point = Linear in θ and ϕ
⇒ "Cycling behavior" (see right).



$$\min_{\theta \in \mathbb{R}} \max_{\phi \in \mathbb{R}} \quad \theta \cdot \phi$$

Our contribution: analysis of gradient, averaging and extragradient for bilinear saddle points.

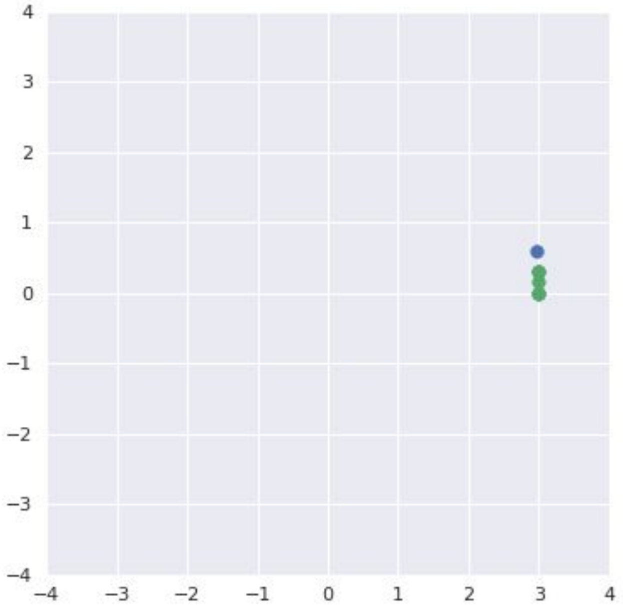
Minimax Training is ~~hard~~ fascinating

Rediscovery of the failure of gradient method in games [Goodfellow, 2016]

<u>Example</u>: WGAN with **linear** discriminator and generator [Mescederer et al., 2018]

$$\min_{\theta} \max_{\phi, ||f_\phi||_L \leq 1} \phi^T \mathbb{E}_{x \sim p_{\mathcal{D}}}[x] - \phi^T \theta \mathbb{E}_{z \sim p_{\mathcal{Z}}}[z]$$

Bilinear saddle point = Linear in θ and ϕ
⇒ "Cycling behavior" (see right).



$$\min_{\theta \in \mathbb{R}} \max_{\phi \in \mathbb{R}} \quad \theta \cdot \phi$$

Our contribution: analysis of gradient, averaging and extragradient for bilinear saddle points.

Generative Adversarial Networks as a Variational Inequality Problem (VIP)

GANs as a Variational Inequality

New perspective for GANs:

- Based on the **vector field** of the game and its **stationary conditions.**
- Relates to vast literature with standard algorithms.

Nash-Equilibrium:

$$\underbrace{L(\theta^*,\phi) \leq \mathrlap{\overbrace{}^{\text{Best strategy for first player (min) is: }\theta^*}}L(\theta^*,\phi^*)}_{\text{Best strategy for second player (max) is: }\phi^*} \leq L(\theta,\phi^*)$$

Stationary Conditions:

$$\nabla_\theta L(\theta^*,\phi^*) = \nabla_\phi L(\theta^*,\phi^*) = 0$$

Gradient of the **first** player at the Nash

Gradient of the **second** player at the Nash

Main takeaways from this perspective:

- The **losses** do **not** matter.

- What matter is the **vector field** followed for the training:

$$F(\theta,\phi) = \begin{pmatrix} \nabla_\theta L(\theta,\phi) \\ -\nabla_\phi L(\theta,\phi) \end{pmatrix}$$

- This vector field may exhibit **rotations.**

- Need for **specific techniques** to **handle rotations.**

Standard Algorithms from Variational Inequality

Method 1: **Averaging**

- Converge even for *"cycling behavior"*.
- Easy to implement. (out of the training loop)
- Can be combined with any method.

General Online averaging:

$$\bar{\boldsymbol{\omega}}_t = (1 - \tilde{\rho}_t)\bar{\boldsymbol{\omega}}_{t-1} + \tilde{\rho}_t \boldsymbol{\omega}_t \quad \text{where} \quad 0 \leq \tilde{\rho}_t \leq 1 \,.$$

Example 1: **Uniform** averaging

$$\tilde{\rho}_t = \frac{1}{t}\,,\; t \geq 0 : \qquad \bar{\boldsymbol{\omega}}_T = \frac{1}{T}\sum_{k=0}^{T-1} \boldsymbol{\omega}_t$$

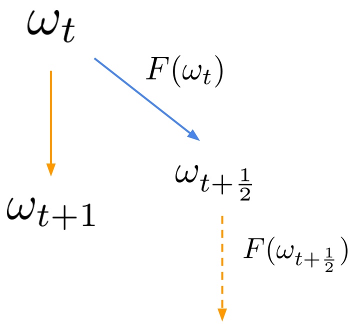
Example 2:
Exponential moving averaging (EMA)

$$\tilde{\rho}_t = 1 - \beta < 1\,,\; t \geq 0 \; : \qquad \bar{\omega}_T = (1-\beta)\sum_{t=1}^{T} \beta^{T-t}\omega_t + \beta^T \omega_0$$

Standard Algorithms from Variational Inequality

Method 2: **Extragradient**

- Step 1: $\omega_{t+\frac{1}{2}} = \omega_t - \gamma_t F(\omega_t)$
- Step 2: $\omega_{t+1} = \omega_t - \gamma_t F(\omega_{t+\frac{1}{2}})$



- Standard in the literature.
- Does not require *averaging.*
- *Theoretically* and *empirically* **faster.**

Intuition:

1. <u>*Game perspective:*</u> *Look one step in the future and anticipate next move of adversary.*
2. *Euler's method: Extrapolation is close to an* ***implicit*** *method because* $\boldsymbol{\omega}_{t+1/2} \approx \boldsymbol{\omega}_{t+1}$

$$\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_{t+1/2} = O(\gamma_t^2)$$

Standard Algorithms from Variational Inequality

Method 2: **Extragradient**

New Intuition: *Extrapolation is close to an* ***implicit*** *method because* $\omega_{t+1/2} \approx \omega_{t+1}$

$$\text{Implicit step:} \quad \omega_{t+1} = \omega_t - \eta F(\omega_{t+1})$$

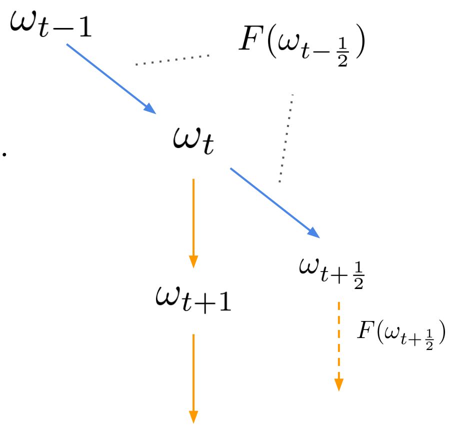
Unknown:
Require to solve a
non-linear system

Extrapolation from the past: Re-using the gradients

Problem: Extragradient requires to compute **two** gradients at each step.

Solution: **Extrapolation from the past** ← **Re-use** gradient.

- Step 1: $\omega_{t+\frac{1}{2}} = \omega_t - \gamma_t F(\omega_{t-\frac{1}{2}})$ ← ***Re-use*** *from previous iteration.*
- Step 2: $\omega_{t+1} = \omega_t - \gamma_t F(\omega_{t+\frac{1}{2}})$ ← (*same as* ***extragradient***).

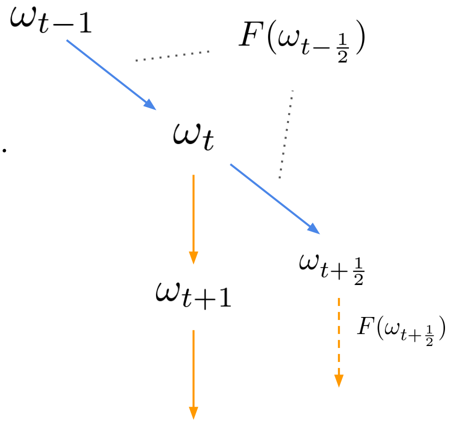


Extrapolation from the past: Re-using the gradients

Problem: Extragradient requires to compute **two** gradients at each step.

Solution: **Extrapolation from the past** ← **Re-use** gradient.

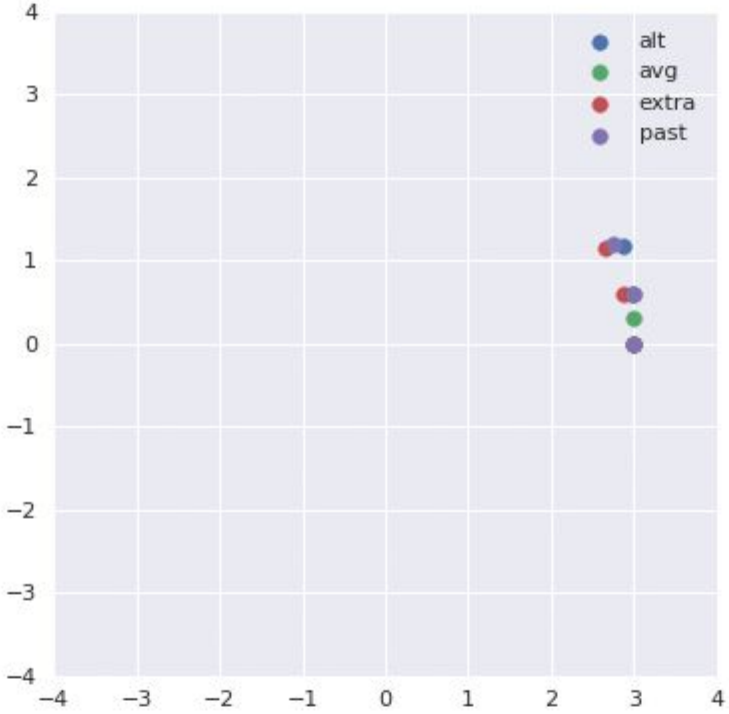
- Step 1: $\omega_{t+\frac{1}{2}} = \omega_t - \gamma_t F(\omega_{t-\frac{1}{2}})$ ← ***Re-use*** *from previous iteration.*
- Step 2: $\omega_{t+1} = \omega_t - \gamma_t F(\omega_{t+\frac{1}{2}})$ ← (*same as* ***extragradient***).



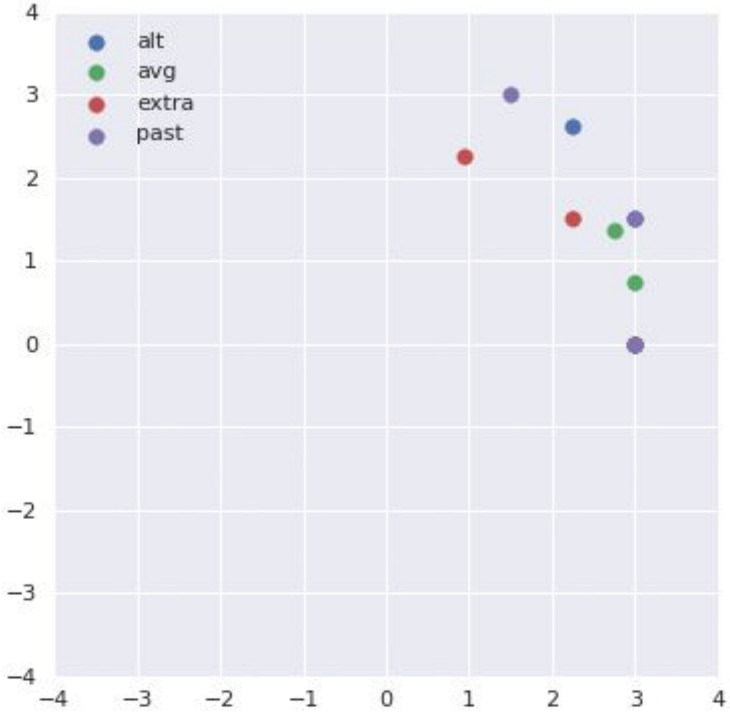
New method with convergence guarantees!!!

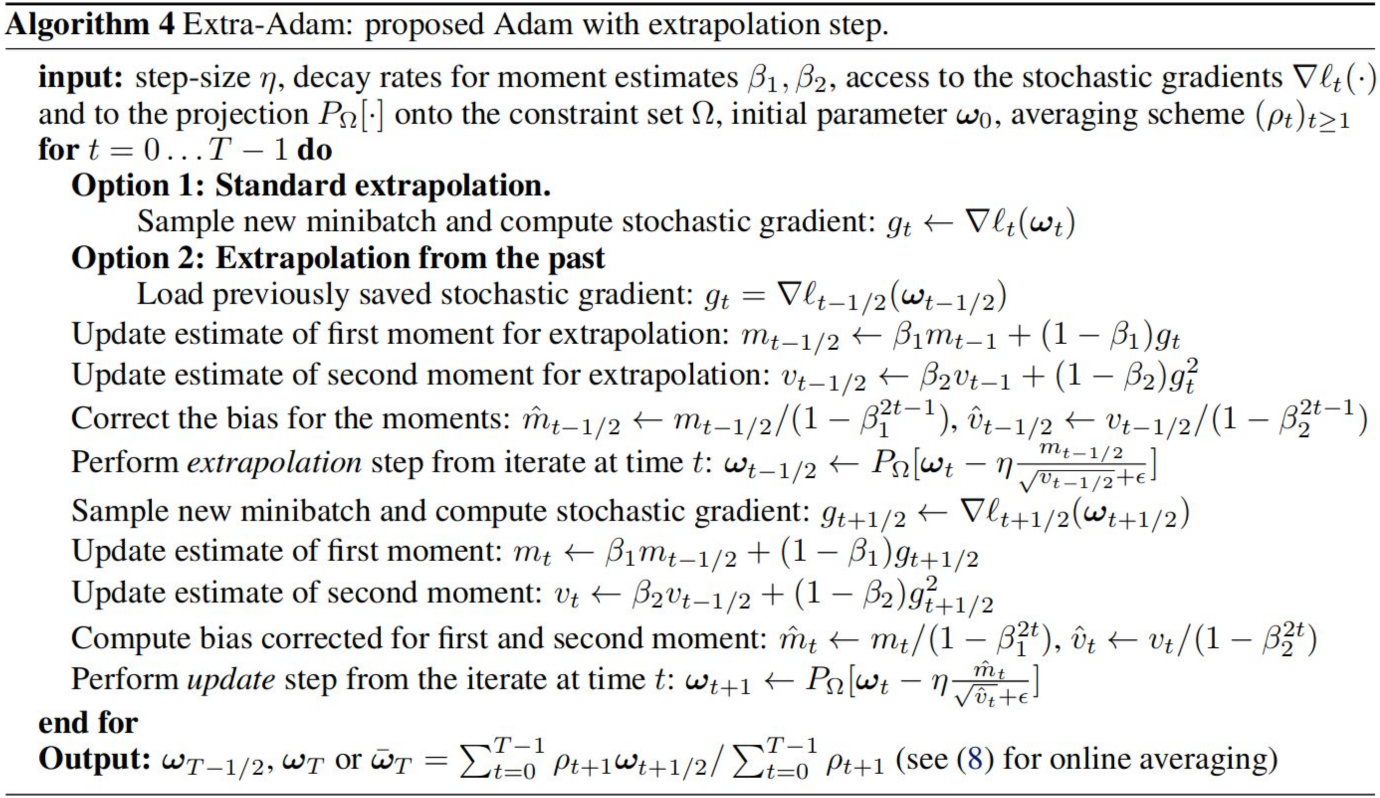
Related to [Daskalakis et al., 2018]

step-size = 0.2



step-size = 0.5





Algorithm 4 Extra-Adam: proposed Adam with extrapolation step.

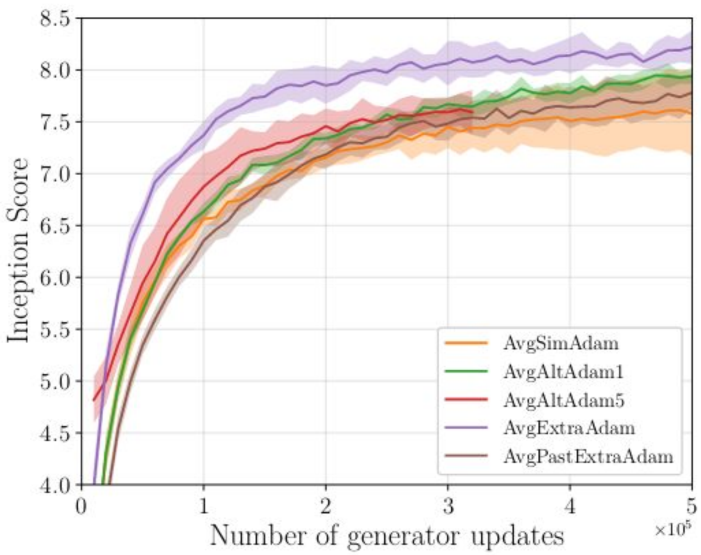
```
input: step-size $\eta$, decay rates for moment estimates $\beta_1, \beta_2$, access to the stochastic gradients $\nabla \ell_t(\cdot)$
and to the projection $P_\Omega[\cdot]$ onto the constraint set $\Omega$, initial parameter $\boldsymbol{\omega}_0$, averaging scheme $(\rho_t)_{t\geq 1}$
for $t = 0 \ldots T-1$ do
    Option 1: Standard extrapolation.
        Sample new minibatch and compute stochastic gradient: $g_t \leftarrow \nabla \ell_t(\boldsymbol{\omega}_t)$
    Option 2: Extrapolation from the past
        Load previously saved stochastic gradient: $g_t = \nabla \ell_{t-1/2}(\boldsymbol{\omega}_{t-1/2})$
    Update estimate of first moment for extrapolation: $m_{t-1/2} \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$
    Update estimate of second moment for extrapolation: $v_{t-1/2} \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$
    Correct the bias for the moments: $\hat{m}_{t-1/2} \leftarrow m_{t-1/2}/(1-\beta_1^{2t-1})$, $\hat{v}_{t-1/2} \leftarrow v_{t-1/2}/(1-\beta_2^{2t-1})$
    Perform extrapolation step from iterate at time $t$: $\boldsymbol{\omega}_{t-1/2} \leftarrow P_\Omega[\boldsymbol{\omega}_t - \eta \frac{m_{t-1/2}}{\sqrt{v_{t-1/2}+\epsilon}}]$
    Sample new minibatch and compute stochastic gradient: $g_{t+1/2} \leftarrow \nabla \ell_{t+1/2}(\boldsymbol{\omega}_{t+1/2})$
    Update estimate of first moment: $m_t \leftarrow \beta_1 m_{t-1/2} + (1-\beta_1) g_{t+1/2}$
    Update estimate of second moment: $v_t \leftarrow \beta_2 v_{t-1/2} + (1-\beta_2) g_{t+1/2}^2$
    Compute bias corrected for first and second moment: $\hat{m}_t \leftarrow m_t/(1-\beta_1^{2t})$, $\hat{v}_t \leftarrow v_t/(1-\beta_2^{2t})$
    Perform update step from the iterate at time $t$: $\boldsymbol{\omega}_{t+1} \leftarrow P_\Omega[\boldsymbol{\omega}_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t+\epsilon}}]$
end for
Output: $\boldsymbol{\omega}_{T-1/2}$, $\boldsymbol{\omega}_T$ or $\bar{\boldsymbol{\omega}}_T = \sum_{t=0}^{T-1} \rho_{t+1} \boldsymbol{\omega}_{t+1/2} / \sum_{t=0}^{T-1} \rho_{t+1}$ (see (8) for online averaging)
```

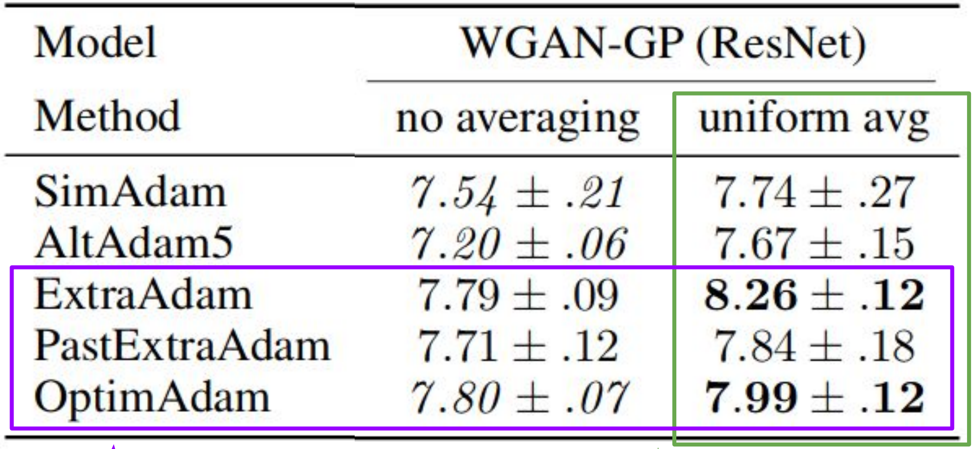
Extrapolation
(Adam style)

Update
(Adam style)

Experimental Results: WGAN-GP (ResNet) on CIFAR10

Inception Score vs
Number of updates





Model	WGAN-GP (ResNet)	
Method	no averaging	uniform avg
SimAdam	*7.54 ± .21*	7.74 ± .27
AltAdam5	*7.20 ± .06*	7.67 ± .15
ExtraAdam	7.79 ± .09	**8.26 ± .12**
PastExtraAdam	7.71 ± .12	7.84 ± .18
OptimAdam	*7.80 ± .07*	**7.99 ± .12**

Extragradient Methods

Averaging

Recall takeaways from VIP perspective:

- What matter is the **vector field** followed for the training.

$$v(\theta,\phi) = \begin{pmatrix} \nabla_\theta L(\theta,\phi) \\ -\nabla_\phi L(\theta,\phi) \end{pmatrix}$$

- This vector field may exhibit **rotations.**

Recall takeaways from VIP perspective:

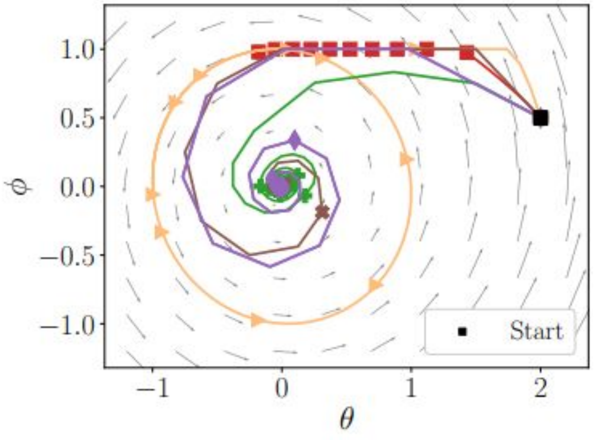
- What matter is the **vector field** followed for the training.

$$v(\theta,\phi) = \begin{pmatrix} \nabla_\theta L(\theta,\phi) \\ -\nabla_\phi L(\theta,\phi) \end{pmatrix}$$

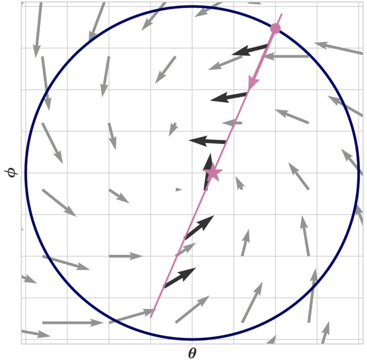
- This vector field **may** exhibit **rotations.**

Is it really the case in practice ?

Outline



1. **Latent games:** how to leverage ... play games.
2. **Game Optimization:** what are t ... ising.
3. **The landscape of games:** an empirical study of practical landscapes.
4. **Future Work:** Design of new adversarial form



A closer look at the landscapes of GANs

Gauthier Gidel*, Hugo Berard*, Amjad Almairi, Pascal Vincent, Simon Lacoste-Julien

*Work Accepted at **ICLR 2020** done during an internship at **ElementAI***

facebook Artificial Intelligence Research

ELEMENT AI

Université de Montréal

Mila

Recall takeaways from VIP perspective:

- What matter is the **vector field** followed for the training.

$$v(\theta,\phi) = \begin{pmatrix} \nabla_\theta L(\theta,\phi) \\ -\nabla_\phi L(\theta,\phi) \end{pmatrix}$$

- This vector field **may** exhibit **rotations.** [Mescheder et al., 2018] [Balduzzi et al., 2018]

Is it really the case in practice ?

Path-Angle: A new visualization tool to detect rotations.

1. **Linear path** between initialization and last iterate.

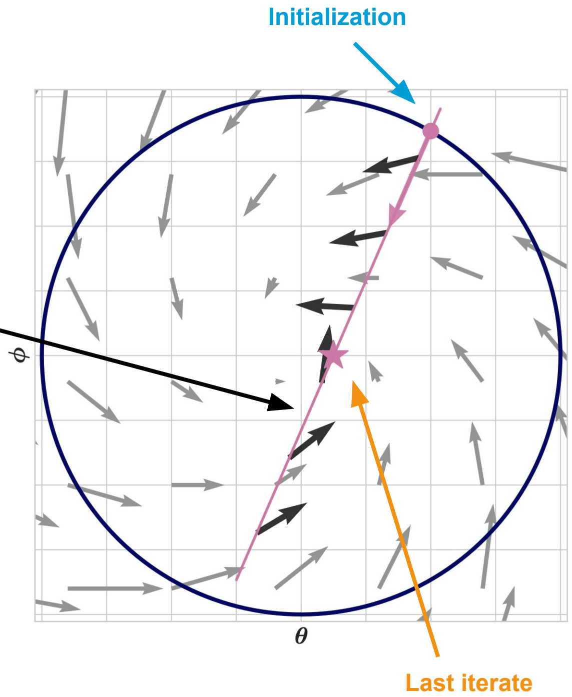
$$\boldsymbol{\omega}_\alpha := \alpha \boldsymbol{\omega}' + (1-\alpha)\boldsymbol{\omega}, \quad \alpha \in [a,b]$$

2. Compute the **norm** of the *game vector field*.

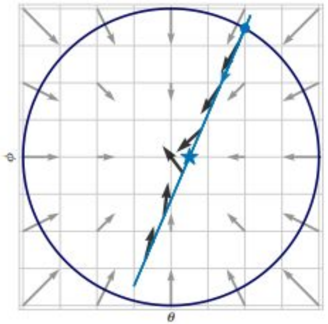
$$:= \|\boldsymbol{v}(\boldsymbol{\omega}_\alpha)\|$$

3. Compute **cosine similarity** between linear path and game vector field.

$$c(\alpha) := \frac{\langle \boldsymbol{\omega}' - \boldsymbol{\omega}, \boldsymbol{v}(\boldsymbol{\omega}_\alpha)\rangle}{\|\boldsymbol{\omega}' - \boldsymbol{\omega}\| \|\boldsymbol{v}(\boldsymbol{\omega}_\alpha)\|}$$



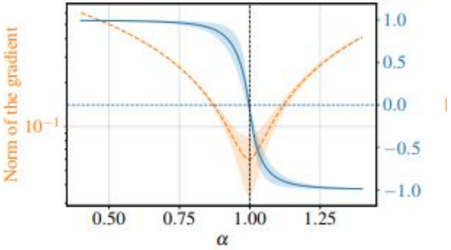
Path-Angle Plots: 3 archetypal behaviors.



(a) Attraction only

Orange: The norm of the game vector field.

Blue: The cosine similarity.

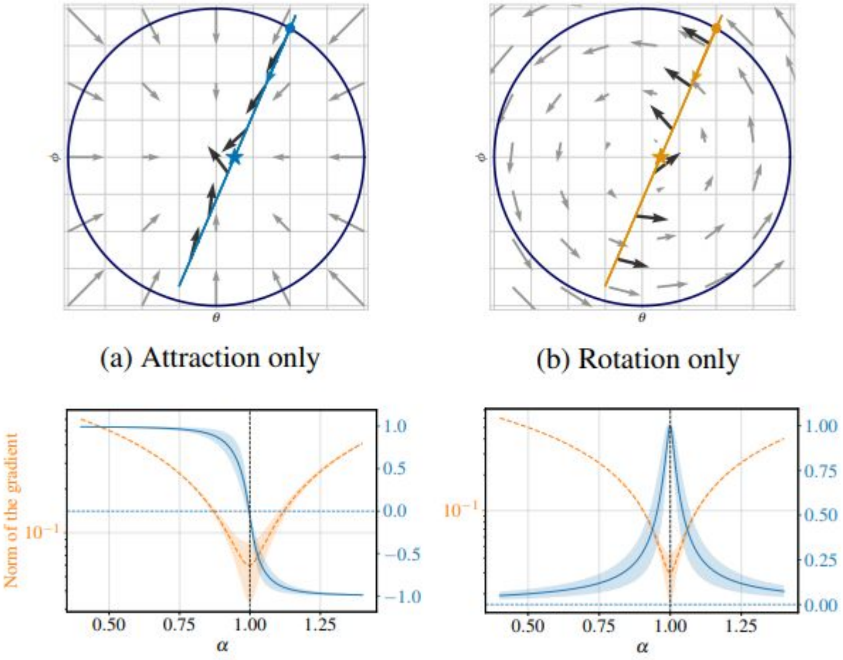


Sign Switch: Indicates attractive behavior.

Path-Angle Plots: 3 archetypal behaviors.

Orange: The norm of the game vector field.

Blue: The cosine similarity.



(a) Attraction only

(b) Rotation only

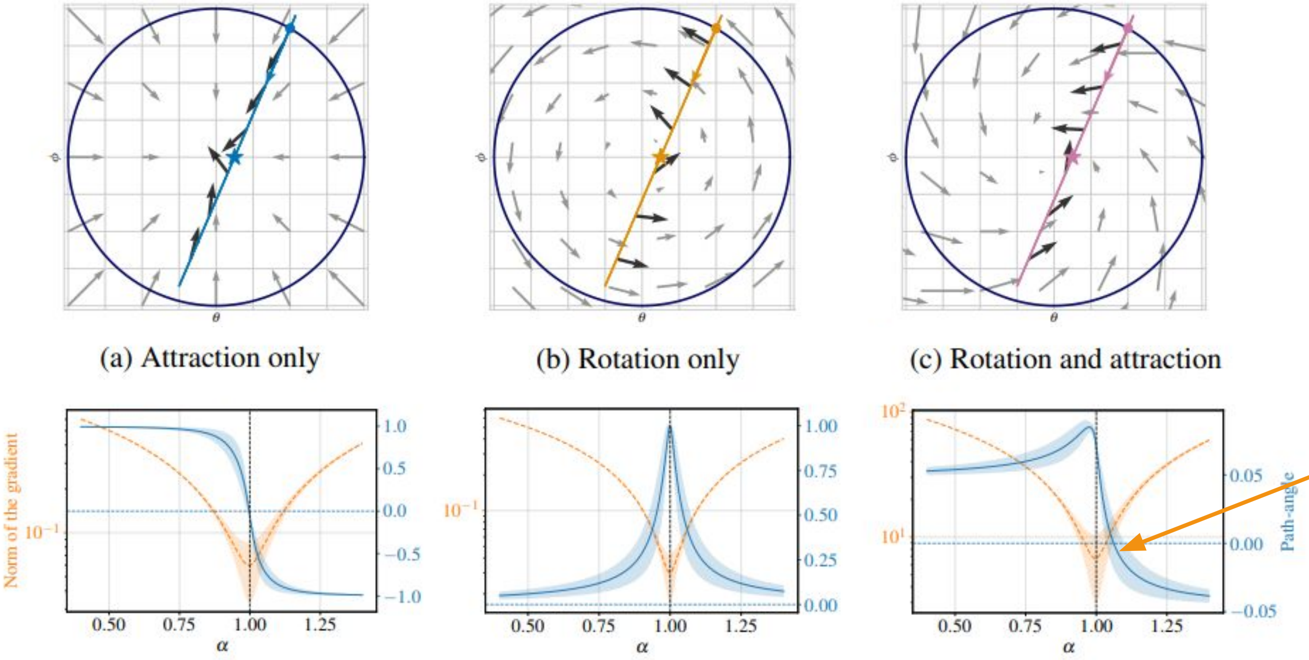
Sign Switch: Indicates attractive behavior.

Bump: Indicates rotational behavior.

Path-Angle Plots: 3 archetypal behaviors.

Orange: The norm of the game vector field.

Blue: The cosine similarity.



(a) Attraction only

(b) Rotation only

(c) Rotation and attraction

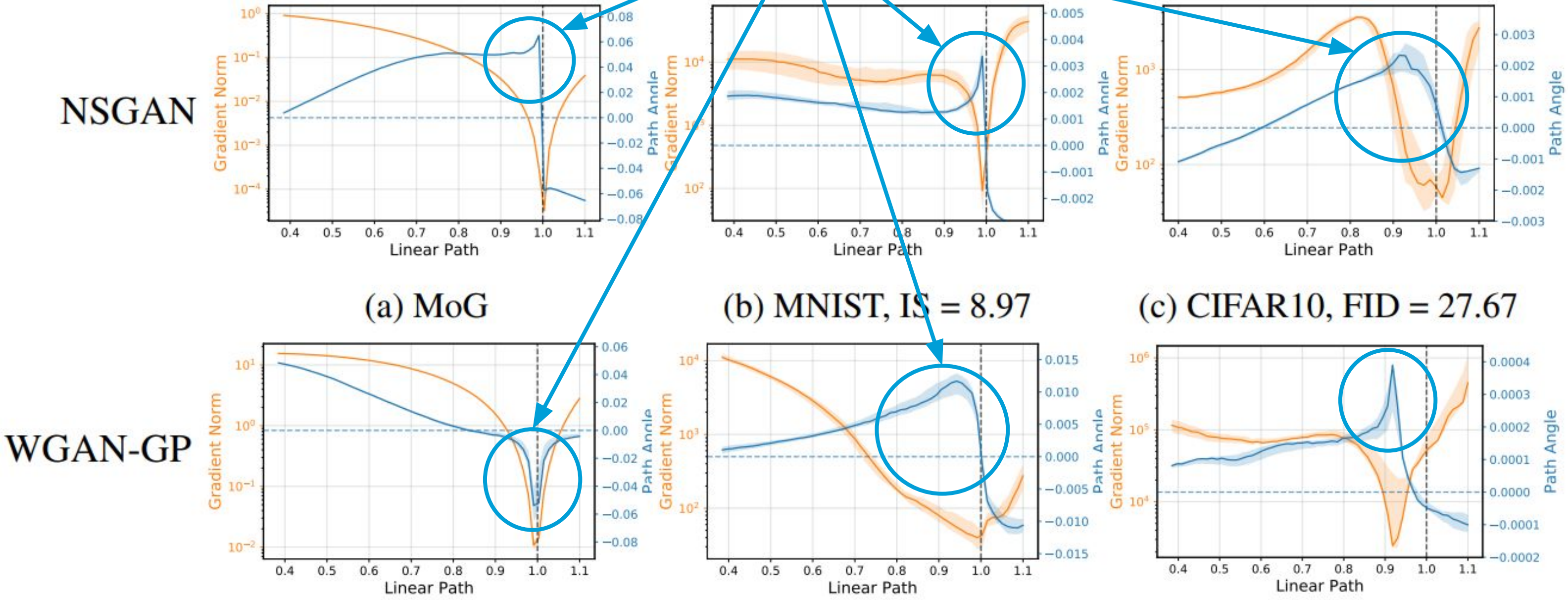
Norm of the gradient is close to zero = Equilibrium point

Sign Switch: Indicates attractive behavior.

Bump: Indicates rotational behavior.

Mix between bump and sign switch: Indicates rotational + attractive behavior.

Large **bump** indicates the presence of rotations.



(a) MoG

(b) MNIST, IS = 8.97

(c) CIFAR10, FID = 27.67

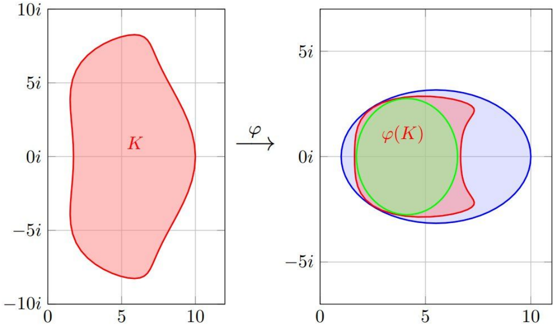
(d) MoG

(e) MNIST, IS = 9.46

(f) CIFAR10, IS = 7.65

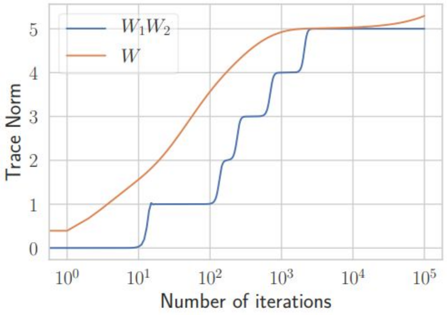
Selected other Publications

- *Accelerating Smooth Games by Manipulating Spectral Shapes,* joint work with Waïss Azizian, Damien Scieur,Ioannis Mitliagkas and Simon Lacoste-Julien. **AISTATS 2020**



- *A Tight and Unified Analysis of Extragradient for a Whole Spectrum of Differentiable Games,* joint work with Waïss Azizian, Ioannis Mitliagkas and Simon Lacoste-Julien. **AISTATS 2020**

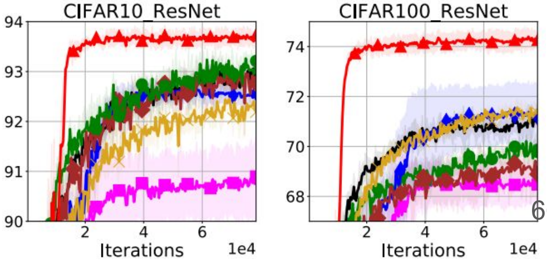
- *Implicit Regularization of Discrete Gradient Dynamics in Deep Linear Neural Networks,* joint work with Francis Bach and Simon Lacoste-Julien. **NeurIPS 2019**



- *Reducing Noise in GAN Training with Variance Reduced Extragradient,* joint work with Tatjana Chavdarova, François Fleuret and Simon Lacoste-Julien. **NeurIPS 2019**

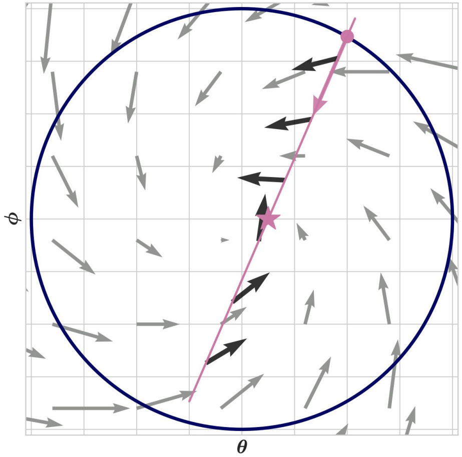
- *Non-normal Recurrent Neural Network (nnRNN): learning long time dependencies while improving expressivity with transient dynamics,* joint work with Giancarlo Kerg, Kyle Goyette, Maximilian Puelma Touzel, Eugene Vorontsov, Yoshua Bengio and Guillaume Lajoie. **NeurIPS 2019**

- *Painless Stochastic Gradient: Interpolation, Line-Search, and Convergence Rates,* joint work with Sharan Vaswani, Aaron Mishkin, Issam Laradji, Mark Schmidt and Simon Lacoste-Julien. **NeurIPS 2019**



Outline

1. **Latent games:** how to leverage fu on to play games.
2. **Game Optimization:** what are the ɛs arising.



3. **The landscape of games:** an empirical study of practical landscapes.
4. **Future Work:** Design of new adversarial formulation for ML.

Path Forward

Communication

Generalization

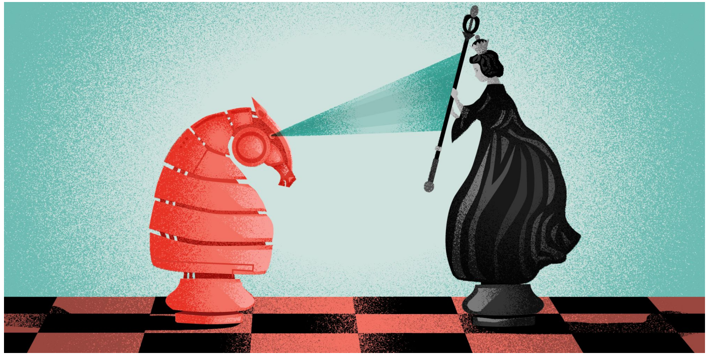
non-convex games

Coordination

Adversarial examples

Building new adversarial formulations for a learning purpose

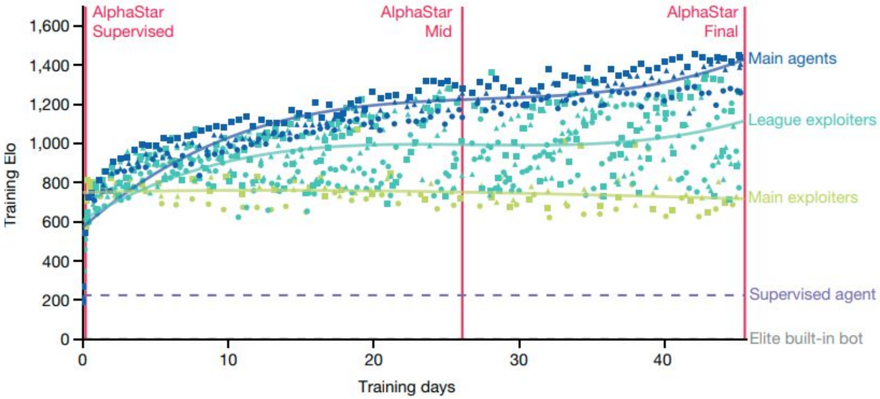
Design **new** adversarial formulation for **pure machine learning purpose.**

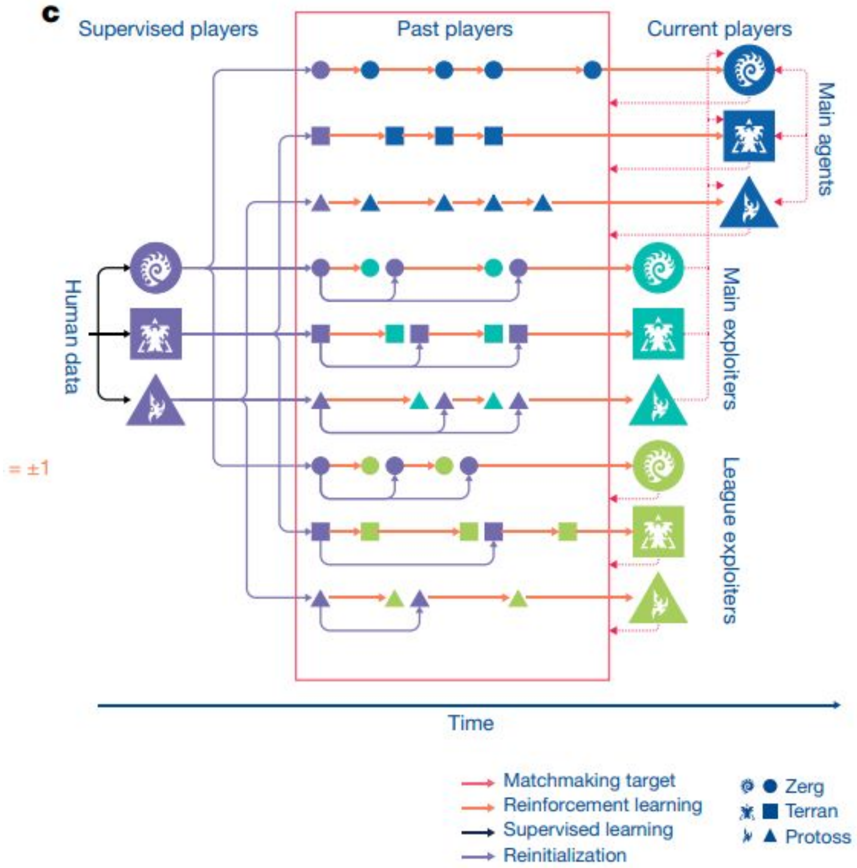


Explore **cooperative** or **coordination** concepts to design **new learning objectives.**

Impact of using a “league” of agents

- Evaluation
- Training
- Definition of diversity





[Vinyals et al. 2019]

Study of non-monotone vector fields

Need for more assumptions

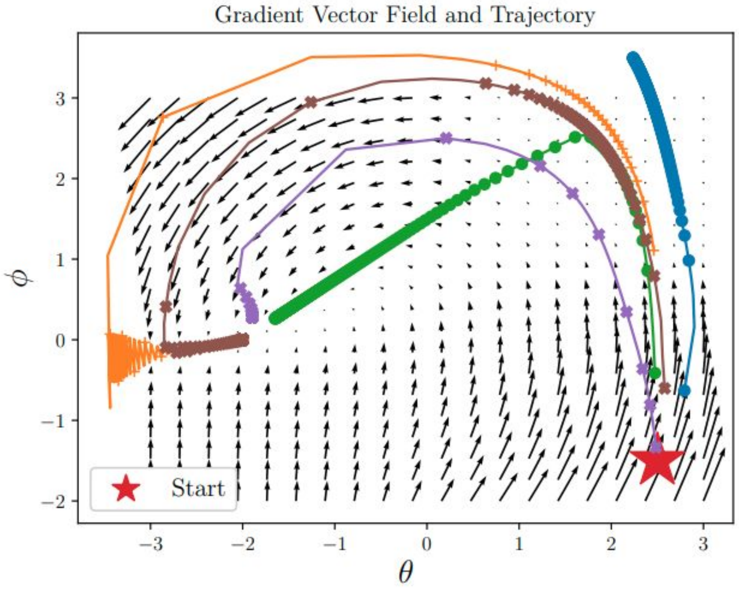


Figure from [Gidel et al. 2019]

COLUMBIA UNIVERSITY

Tony Jebara

facebook Artificial Intelligence Research

Université de Montréal

Mila

Acknowledgements (and many others) !!!

Inria INVENTEURS DU MONDE NUMÉRIQUE

Francis Bach

ELEMENT AI

Mark Schmidt

EPFL

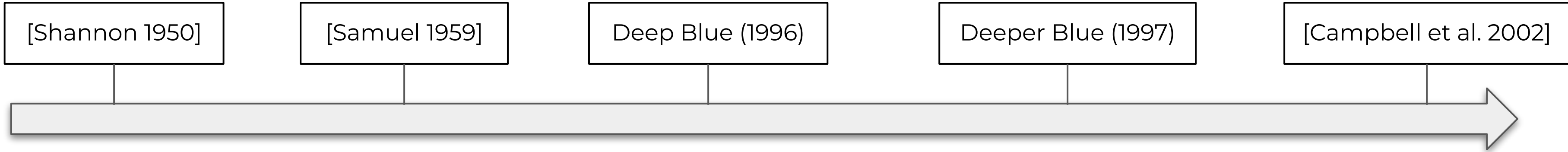
SAMSUNG ADVANCED INSTITUTE OF TECHNOLOGY

1794 ENS ÉCOLE NORMALE SUPÉRIEURE

UBC

Thank you!!!
Any question ?

Achieving super-human performance in Chess has been long standing challenge



[Shannon 1950]

[Samuel 1959]

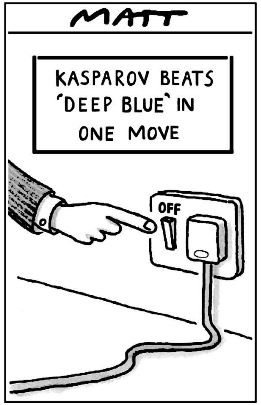
Deep Blue (1996)

Deeper Blue (1997)

[Campbell et al. 2002]

Programming a computer for playing chess.

Some studies in machine learning using the game of checkers



Matthew Pritchett

Photo: EPA

Research paper on Deep Blue

Beyond Chess, achieving super-human performance in multi-player games are great challenges

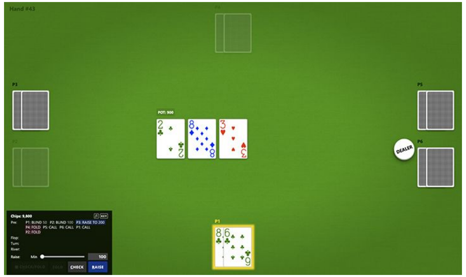
Go

[Silver et al. 2016]
(Picture from DeepMind's blog post)

Dota 2

[OpenAI et al. 2019]
(Picture from OpenAI's Blog post)

Poker



[Brown and Sandholm 2019]
(Picture from FAIR's Blog post)

Starcraft II

[Vinyals et al. 2019]
(Picture from DeepMind's Blog post)

Games specifically designed for Machine learning purposes

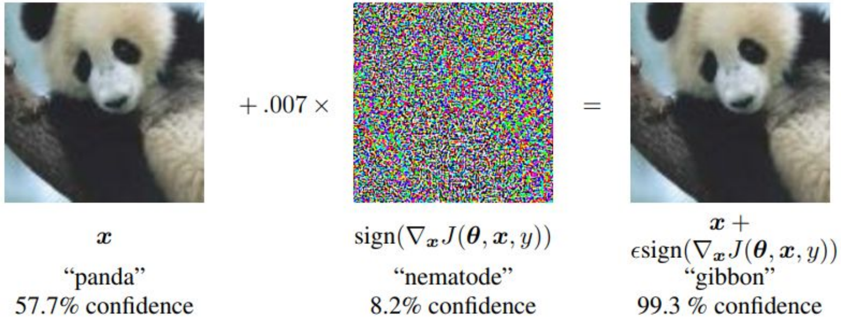
For Generative modeling:

Generative Adversarial Networks
[Goodfellow et al. 2014]

Picture: [Wu et al. 2020]

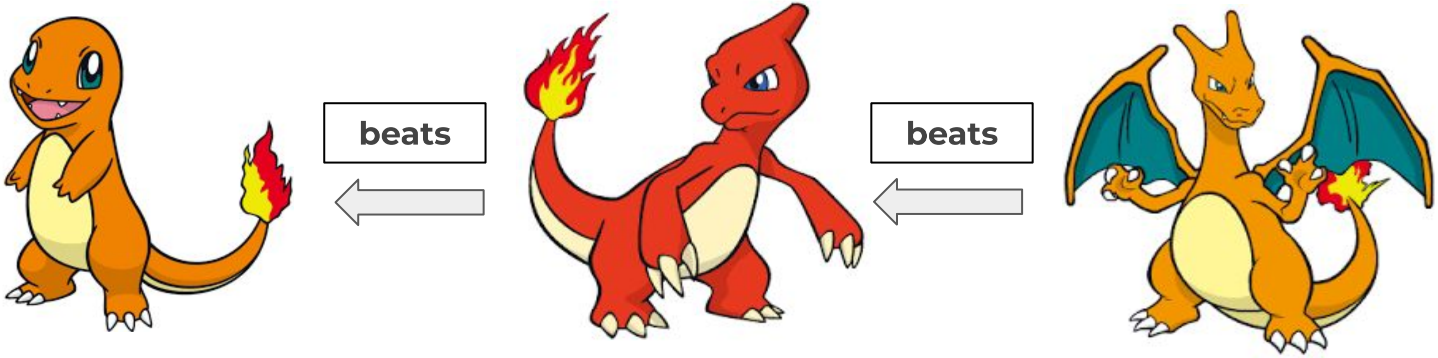
For learning classifier robust to adversarial attacks

Adversarial Training
[Madry et al. 2017]


$+ .007 \times$
$=$
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Picture: [Goodfellow et al. 2014]

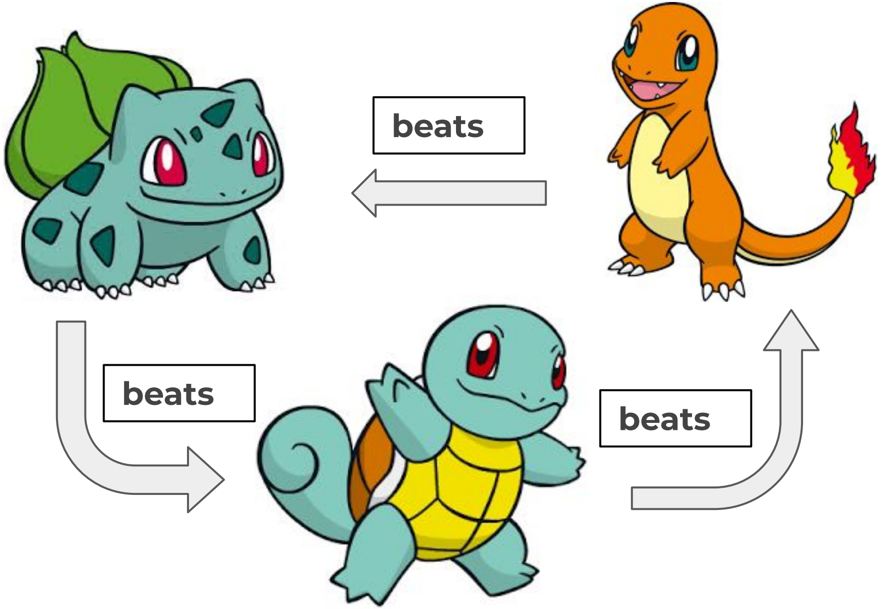
Problem: is there a ‘best’ action?



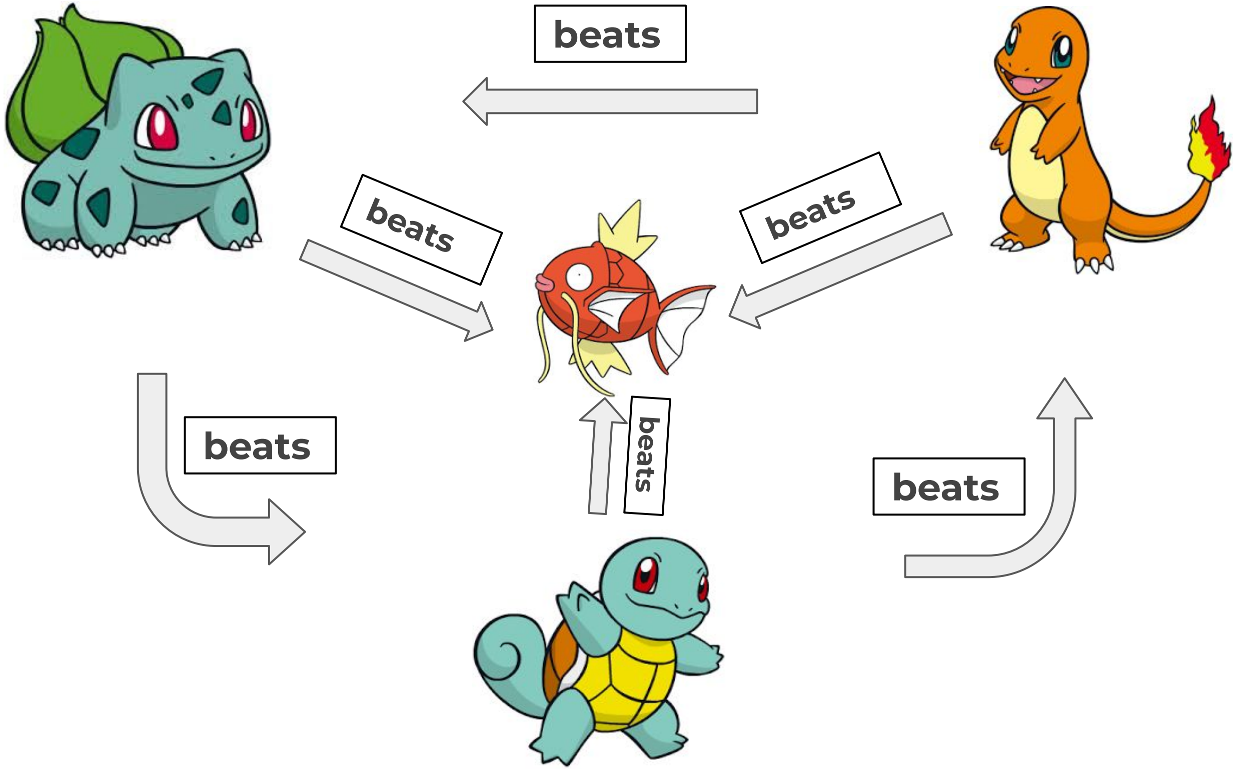
beats

beats

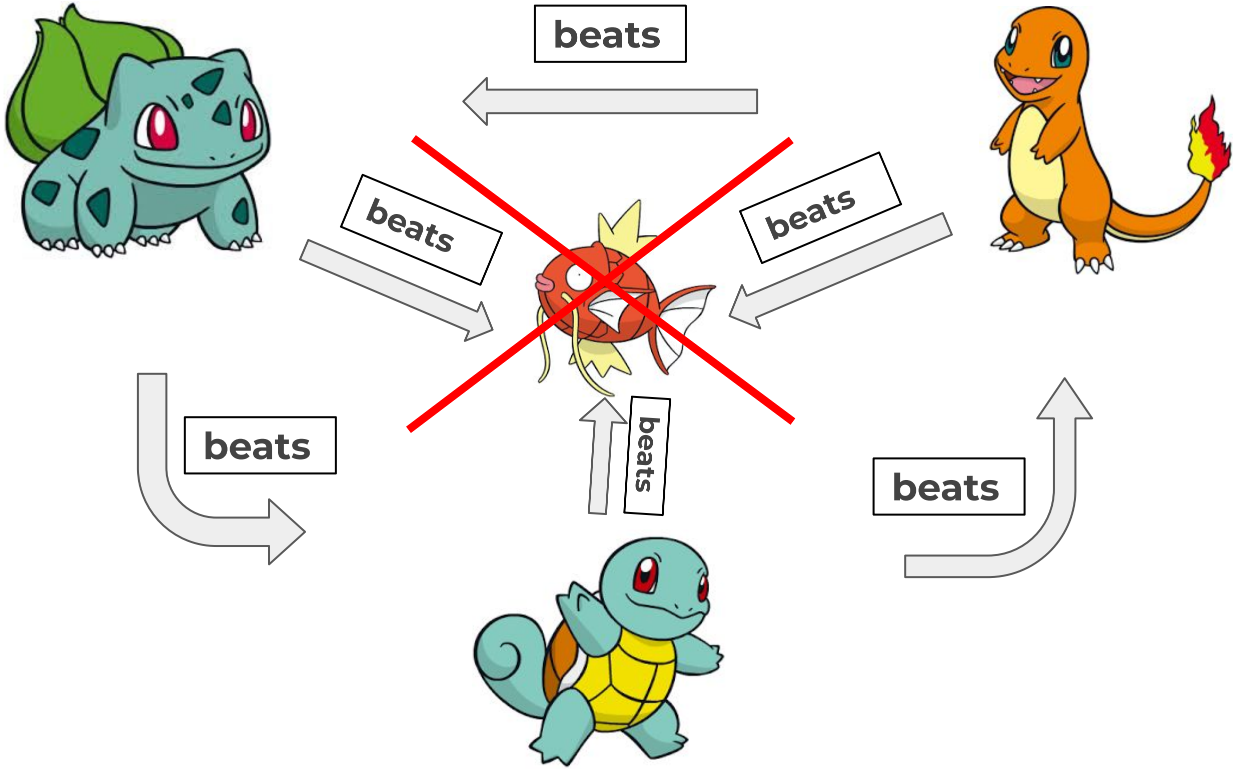
Problem: is there a ‘best’ strategy?



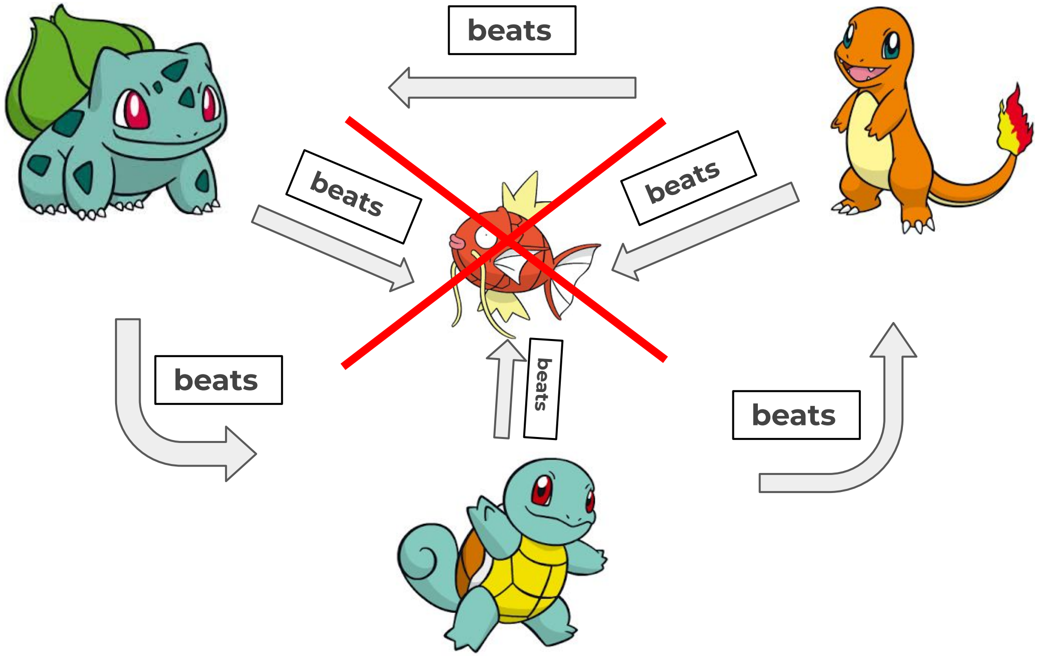
Problem: is there a ‘best’ strategy?



Problem: is there a ‘best’ strategy?



Problem: is there a ‘best’ strategy?



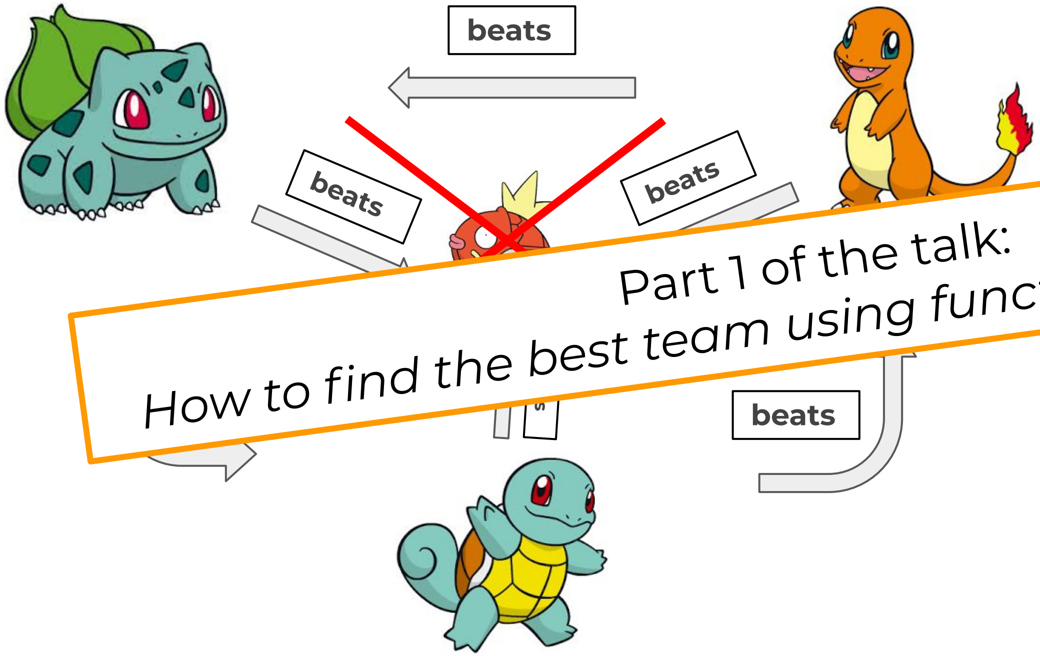
There is **no best action**.

But there is a best **team**

⇩

Mixed equilibrium

Problem: is there a ‘best’ action?



Th **action**.

re is a best **team**

Part 1 of the talk:

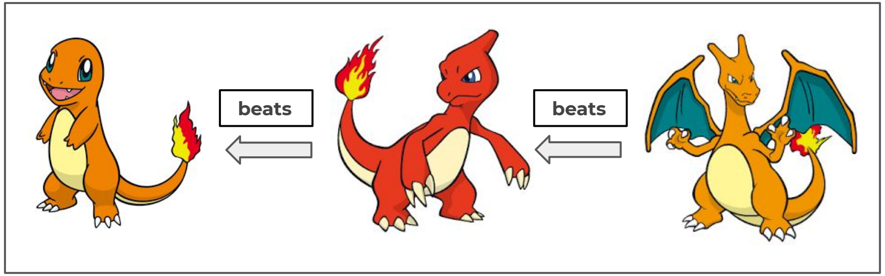
How to find the best team using function approximation

Starcraft II is more challenging to train and evaluate than Go:

Go

[Silver et al. 2016]
(Picture from DeepMind's blog post)

≈



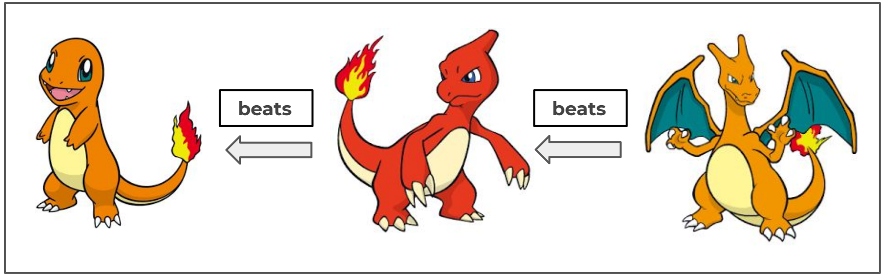
pictures from pokebip.com

Starcraft II is more challenging to train and evaluate than Go:

Go

[Silver et al. 2016]
(Picture from DeepMind's blog post)

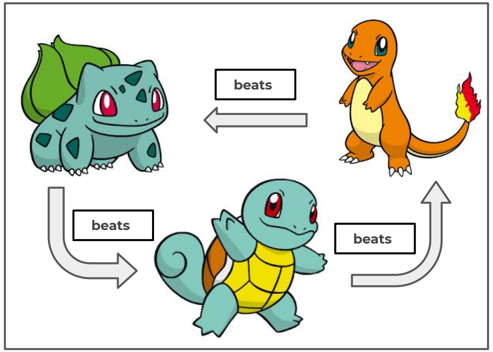
≈



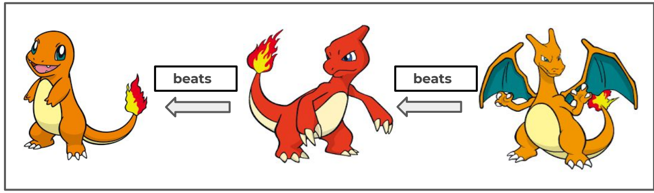
Starcraft II

[Vinyals et al. 2019]
(Picture from DeepMind's Blog post)

≈

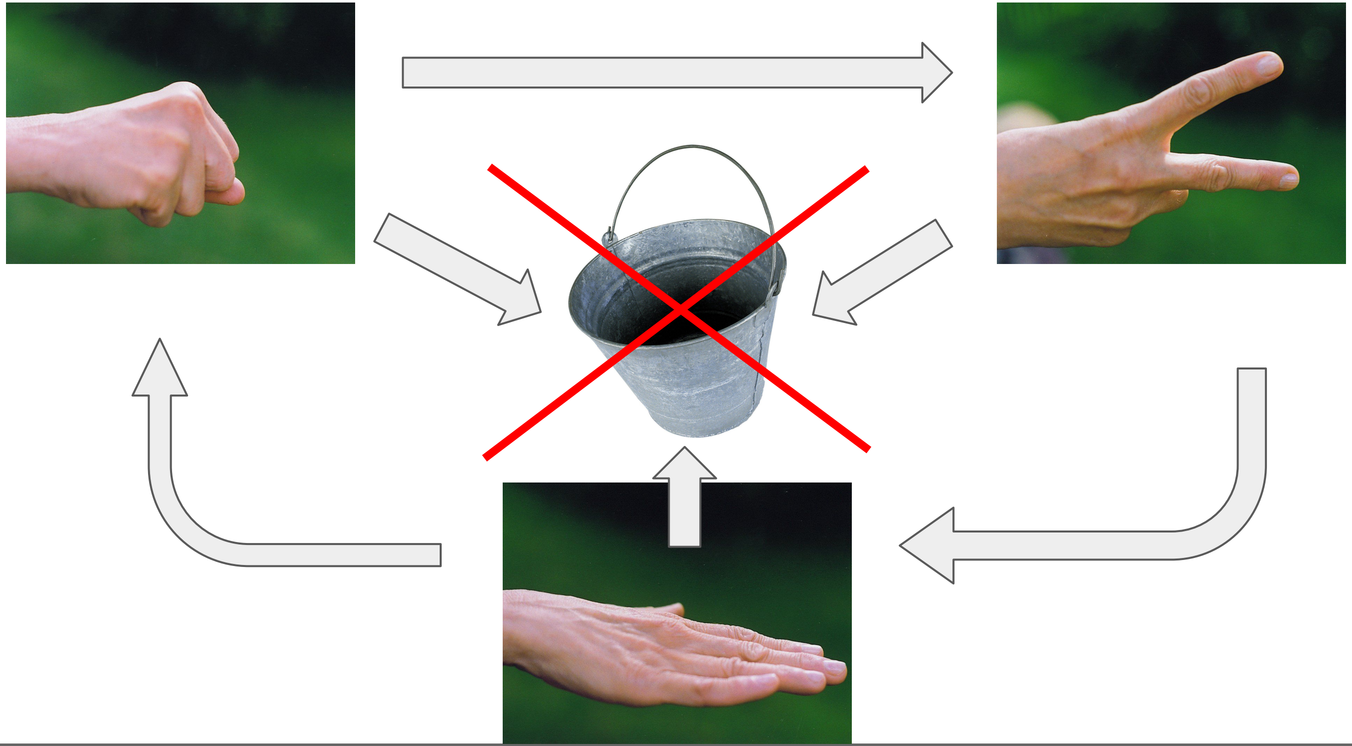


+

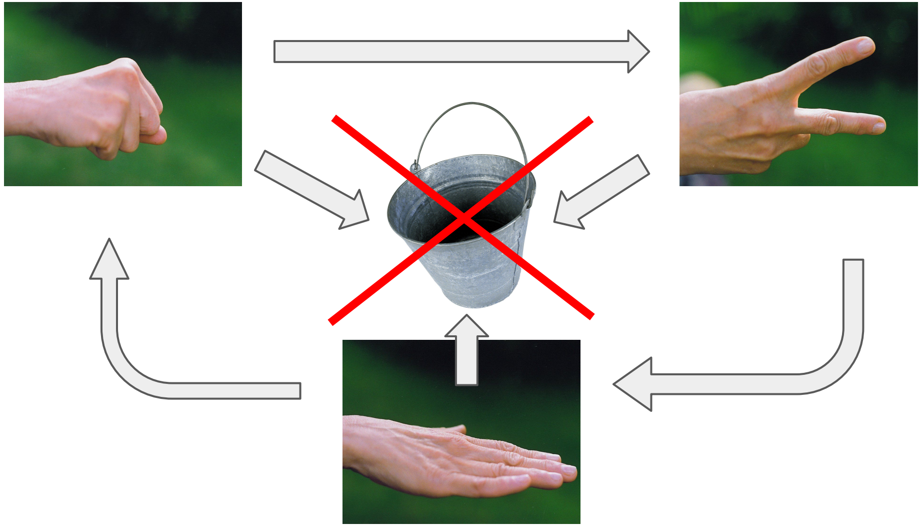


pictures from pokebip.com

Problem 1: is there a ‘best’ action?



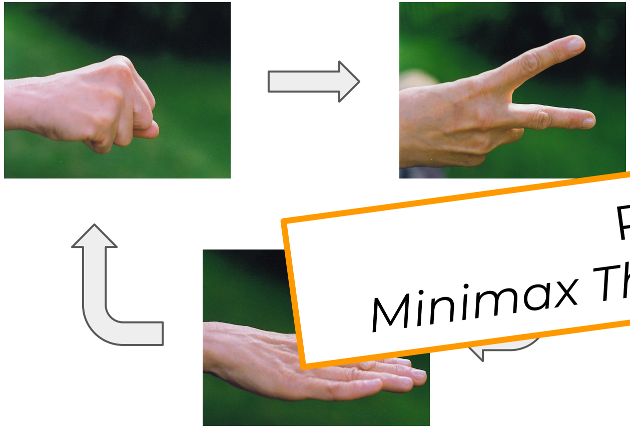
Problem 1: is there a ‘best’ action?



The best agent plays the ‘best’ actions in a “unpredictable” way.

His behavior cannot be <u>exploited</u>

Problem 1: is there a ‘best’ action?



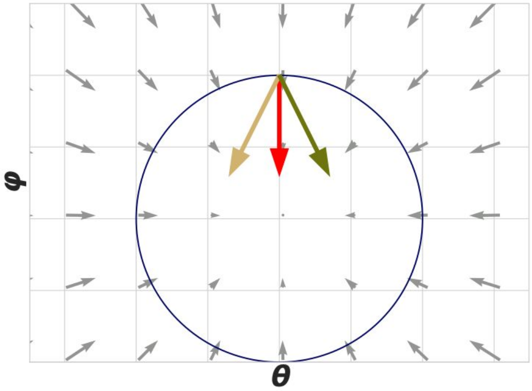
Part 1 of the talk:
Minimax Theorems for Latent games.

The best agent is
unpr e and play
ctions.

His behavior cannot be
<u>exploited</u>

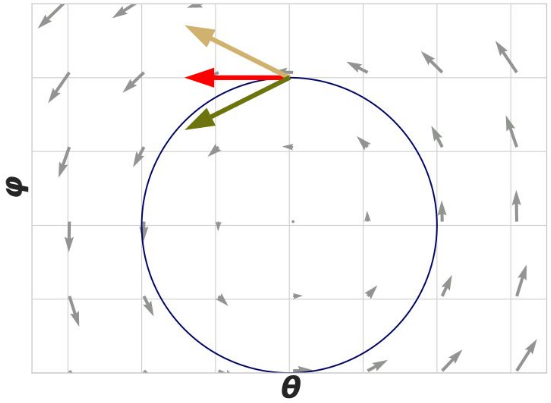
Problem 2: Can we train a reasonably 'good' agent? *(and if yes, how???)*

Standard minimization:
Gradient **descent**



$$\min_{\theta\in\mathbb{R}}\min_{\varphi\in\mathbb{R}} \theta^2+\varphi^2$$

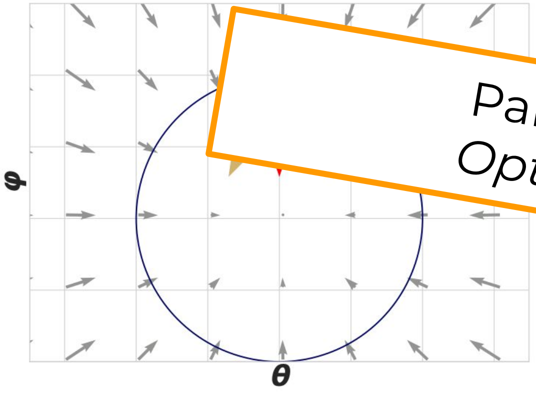
Minimax objective:
Gradient **method**



$$\min_{\theta\in\mathbb{R}}\max_{\phi\in\mathbb{R}} \quad \theta\cdot\phi$$

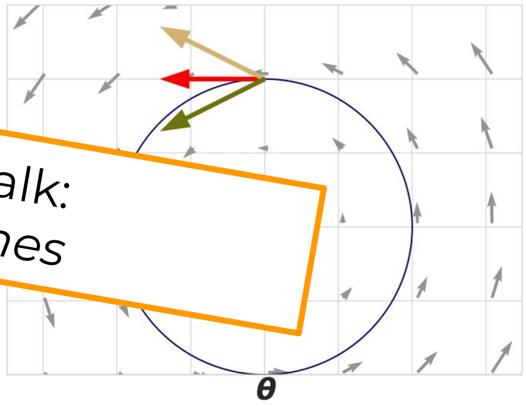
Problem 2: Can we train a reasonably ‘good’ agent? *(and if yes, how???)*

Standard minimization:
Gradient **descent**



$$\min_{\theta,\varphi\in\mathbb{R}} \theta^2 + \varphi^2$$

Minimax objective:
Gradient **method**



$$\min_{\theta\in\mathbb{R}} \max_{\varphi\in\mathbb{R}} \theta \cdot \varphi$$

Part 2 and 3 of the talk:
Optimization of games

Example: Rock-Paper-Scissors

Fundamental result of game theory [von Neumann, 1928]: there exists

1) a number **V**, called the value of the game,
2) a strategy for each player such that their a V) no matter what the other does.

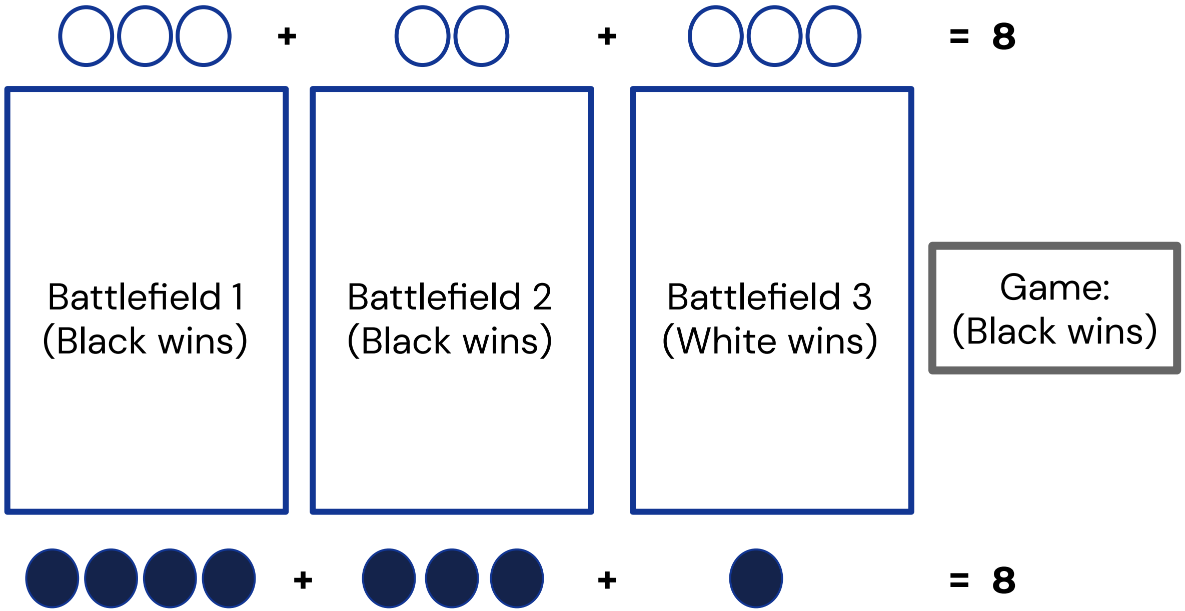
$$V := \min_{q\in\mathcal{P}(B)} \max_{p\in\mathcal{P}(A)} \varphi(p, q$$

> *"As far as I can see, there could be no theory of games [without] the Minimax Theorem"*
>
> *von Neumann (1953)*

$$\varphi(p,q) = \underbrace{\mathbb{E}_{a\sim p,b\sim q}[\varphi(a,b)]}_{\text{Average payoff}}$$

Probability distributions over actions

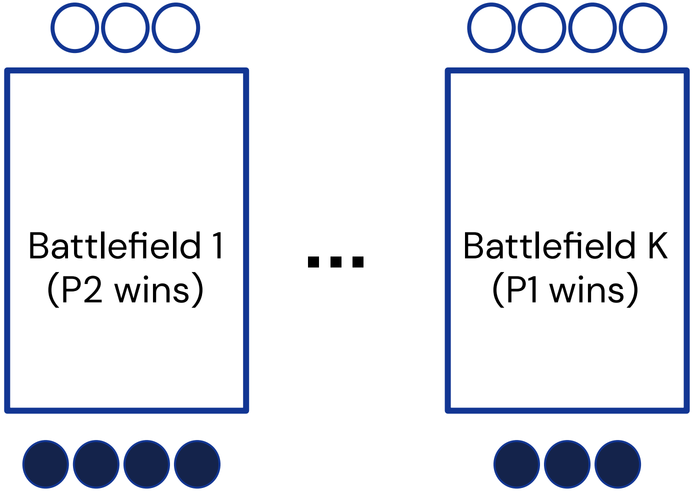
Colonel Blotto Game:



Strategy: One allocation

We want agent to play mixture of strategy, i.e, distribution on allocations

Colonel Blotto Game:



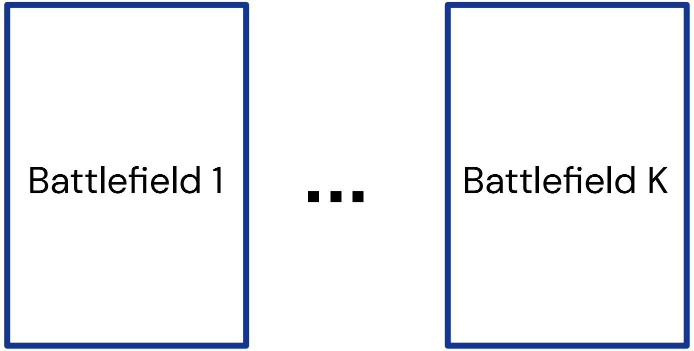
Simplified version: Same number of soldier to allocate.

A strategy is a point in the simplex of dimension K.

A mixed strategy is a distribution over the simplex

Continuous Colonel Blotto Game:

$$p_1 \quad + \quad \dots \quad + \quad p_K \quad = 1$$



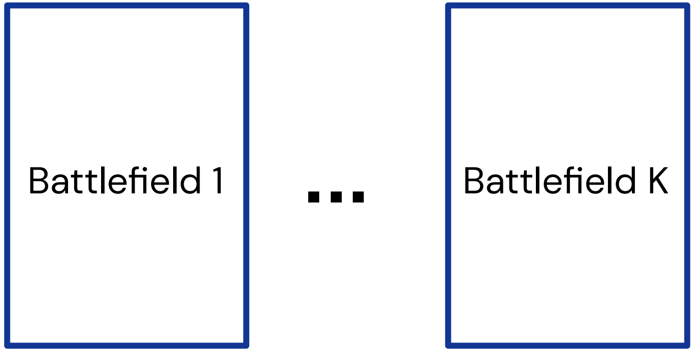
Battlefield 1 ... Battlefield K

$$q_1 \quad + \quad \dots \quad + \quad q_K \quad = 1$$

Payoff = $\mathbf{1}\{p_1 > q_1\} + \dots + \mathbf{1}\{p_K > q_K\}$

Differentiable Colonel Blotto Game:

$p_1 + \dots + p_K = 1$



Battlefield 1

...

Battlefield K

$q_1 + \dots + q_K = 1$

Agents: Latent functions.

$$z \sim \mathcal{N}(0,1) \mapsto A(z) \in \Delta_K$$

Payoff = $\boldsymbol{\sigma}(p_1 - q_1) + \dots + \boldsymbol{\sigma}(p_K - q_K)$

Example: Rock-Paper-Scissors

Fundamental result of game theory [von Neumann, 1928]: there exists

1) a number **V**, called the value of the game,
2) **a strategy for each player such that their average gain is at least V (resp. -V) no matter what the other does.**

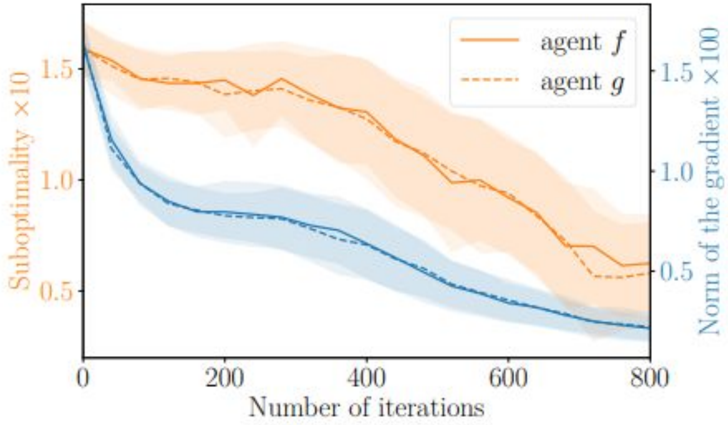
$$V := \min_{q\in\mathcal{P}(B)} \max_{p\in\mathcal{P}(A)} \varphi(p,q) = \max_{p\in\mathcal{P}(A)} \min_{q\in\mathcal{P}(B)} \varphi(p,q)$$

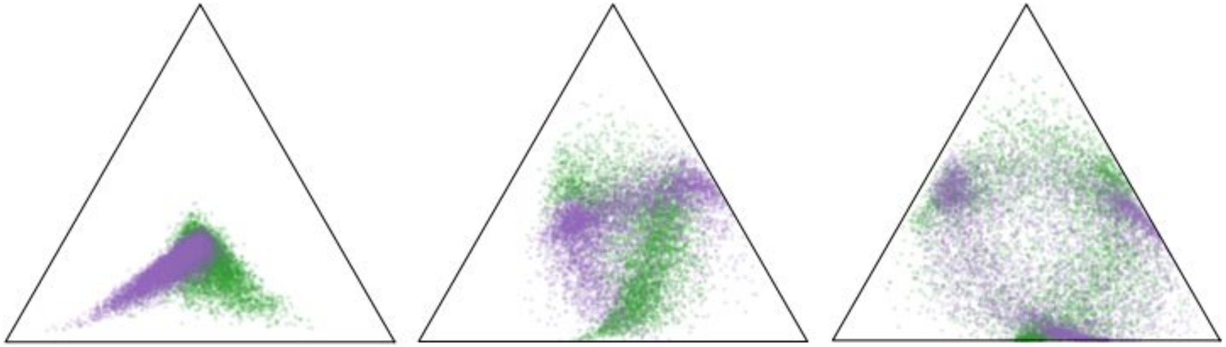
A Goal of the game: find this strategy

$$\varphi(p,q) = \underbrace{\mathbb{E}_{a\sim p, b\sim q}[\varphi(a,b)]}_{\text{Average payoff}}$$

Probability distributions over strategies

Proof of concept:





0 iterations

400 iterations

800 iterations

(Very) Quick reminder on Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs)

[Goodfellow et al. NIPS 2014]

$D_\phi(x)$: Probability of being **real.**

Disciminator: maximize log-likelihood

Example]: Minimax GAN [Goodfellow et al. 2014]

$$\min_\theta \underbrace{\max_\phi \mathbb{E}_{x\sim p_\mathcal{D}}[\log(\overset{\text{Discriminator}}{\overset{\downarrow}{D_\phi}}(x))] + \mathbb{E}_{z\sim p_\mathcal{Z}}[\log(1 - D_\phi(\overset{\text{Generator}}{\overset{\downarrow}{G_\theta}}(z)))]}_{\text{If D is non-parametric: } \quad L(\theta) = \mathrm{JSD}(p_\mathcal{D}||p_\theta)}$$

Generative Adversarial Networks (GANs)

[Goodfellow et al. NIPS 2014]

$D_\phi(x)$: Probability of being **real.**

Discriminator: maximize log-likelihood

Example1: Minimax GAN [Goodfellow et al. 2014]

$$\min_{\theta} \underbrace{\max_{\phi} \mathbb{E}_{x\sim p_{\mathcal{D}}}[\log(\overset{\text{Discriminator}}{D_\phi}(x))] + \mathbb{E}_{z\sim p_{\mathcal{Z}}}[\log(1 - D_\phi(\overset{\text{Generator}}{G_\theta}(z)))]}$$

If D is non-parametric: $L(\theta) = \mathrm{JSD}(p_{\mathcal{D}}||p_\theta)$

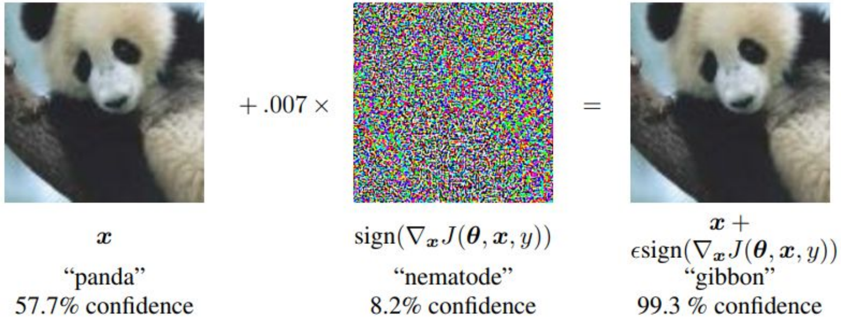
Example2: WGAN formulation [Arjovsky et al. 2017]

$$\min_{\theta} \max_{\phi, ||f_\phi||_L \leq 1} \mathbb{E}_{x\sim p_{\mathcal{D}}}[f_\phi(x)] - \mathbb{E}_{z\sim p_{\mathcal{Z}}}[f_\phi(g_\theta(z)))]$$

Building new adversarial formulations for a learning purpose

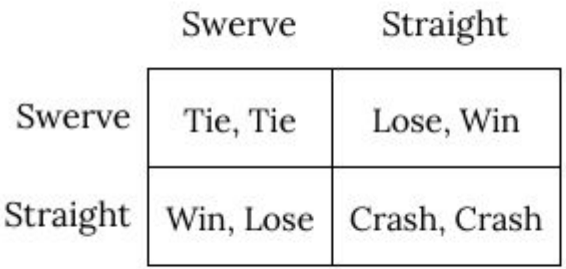
Explore **cooperative** or **coordination** concepts to design **new learning objectives.**

Example: make adversarial training a latent game
[Madry et al. 2017]

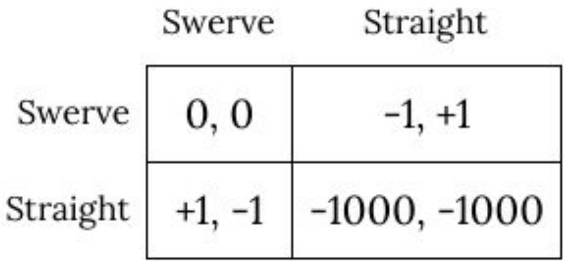

$+ .007 \times$
$=$
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

Picture: [Goodfellow et al. 2014]

Compute Coarse Correlated equilibria for ‘coordination games’



	Swerve	Straight
Swerve	Tie, Tie	Lose, Win
Straight	Win, Lose	Crash, Crash



	Swerve	Straight
Swerve	0, 0	-1, +1
Straight	+1, -1	-1000, -1000

Learning to coordinate by sharing the latent variable.

$$\varphi(f,g) := \mathbb{E}_{z\sim\pi}[\varphi(f(z),g(z))]$$