# Extragradient and Negative Momentum to Optimize GANs

**Simon Lacoste-Julien**

DIRO, Université de Montréal & Mila

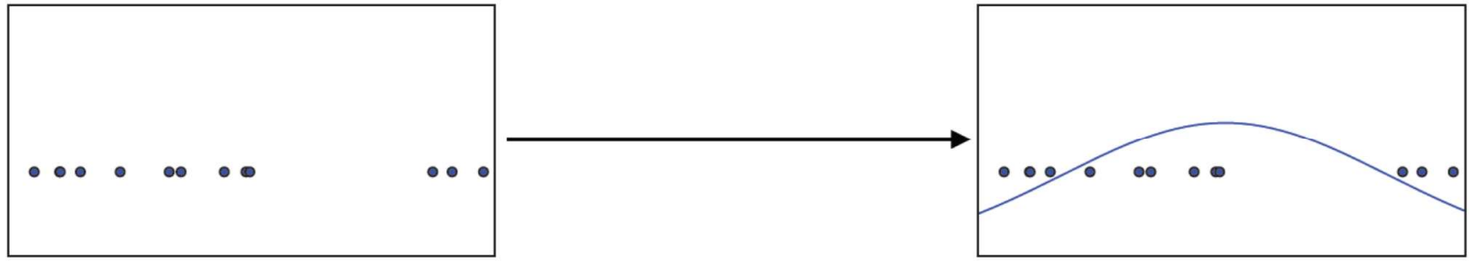


OSL, Les Houches – March 29th, 2019

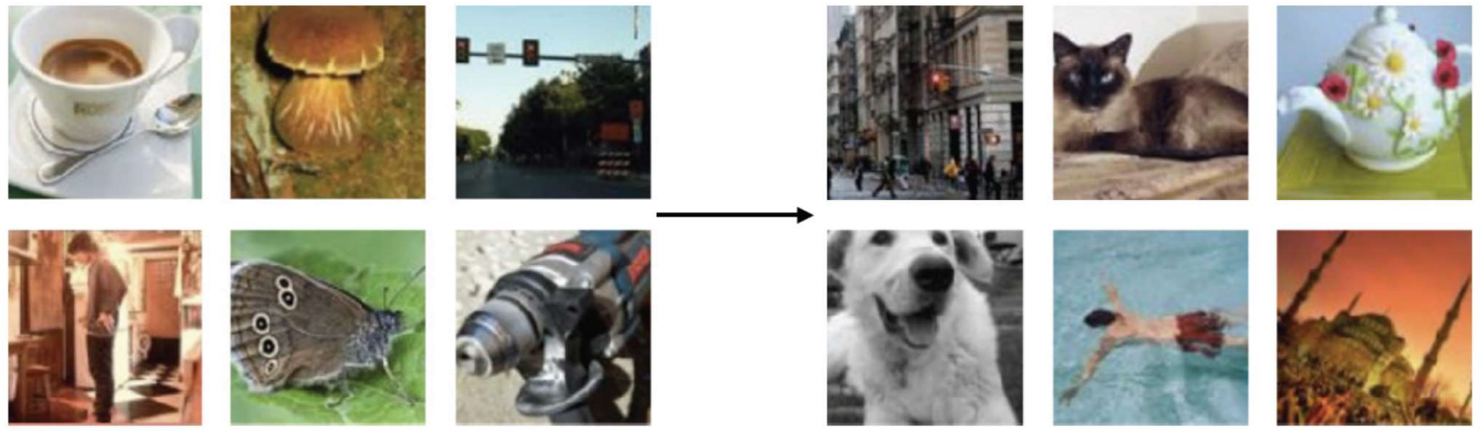
# GAN review

# Generative modeling

- Generative models take training samples from some data distribution and learn a model that represents that distribution.
- Density estimation:



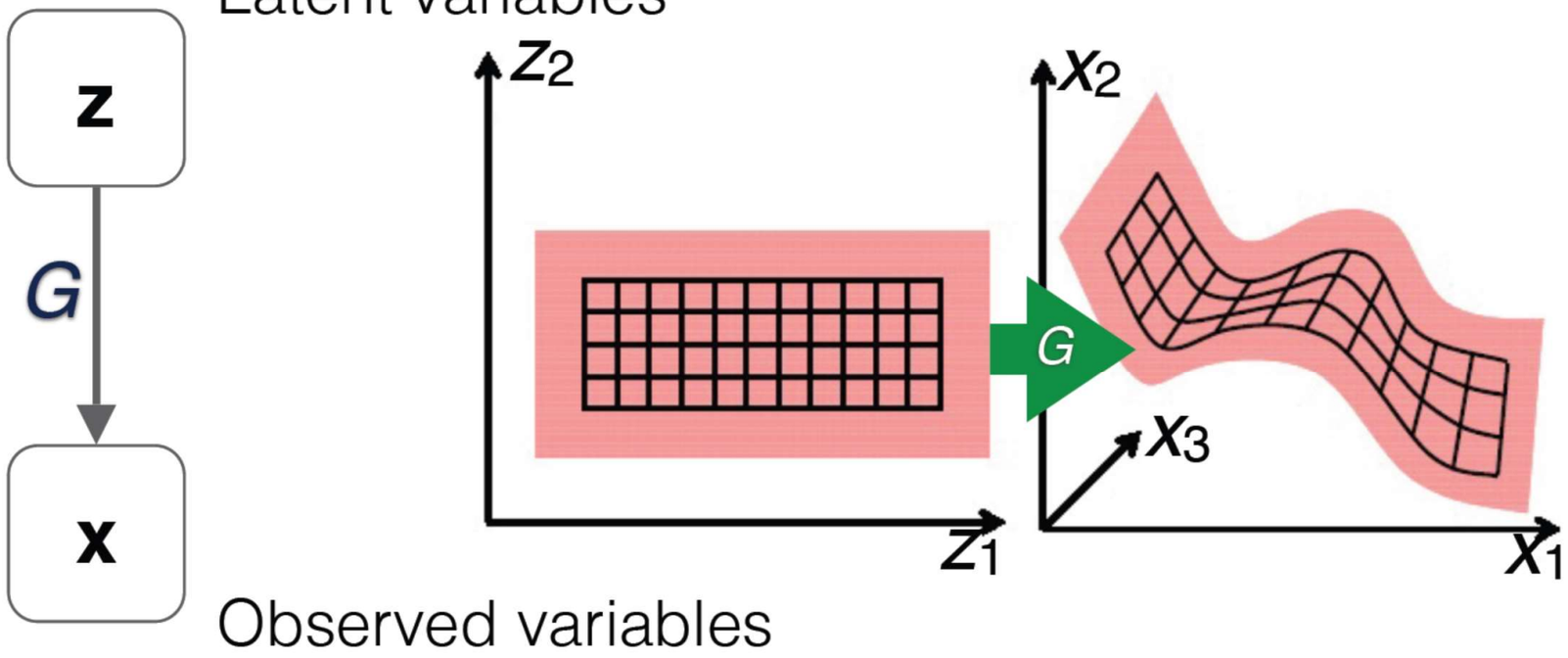
- Sample generation:



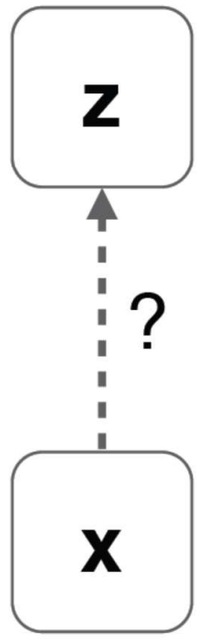

(slide from Aaron Courville's class)

images taken from Goodfellow (2017)

# How to train a latent variable generative model?



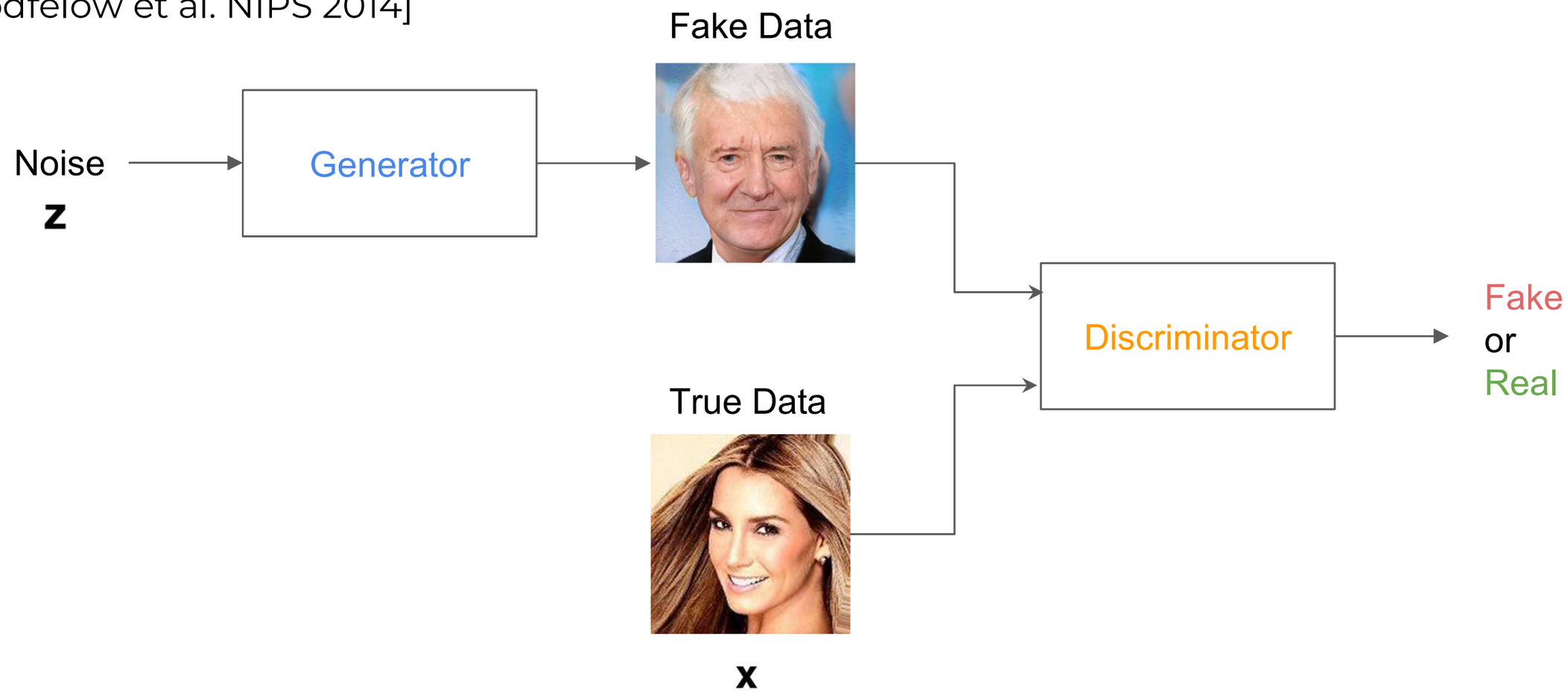

inference




(slide from Aaron Courville's class)

# Generative Adversarial Networks (GANs)

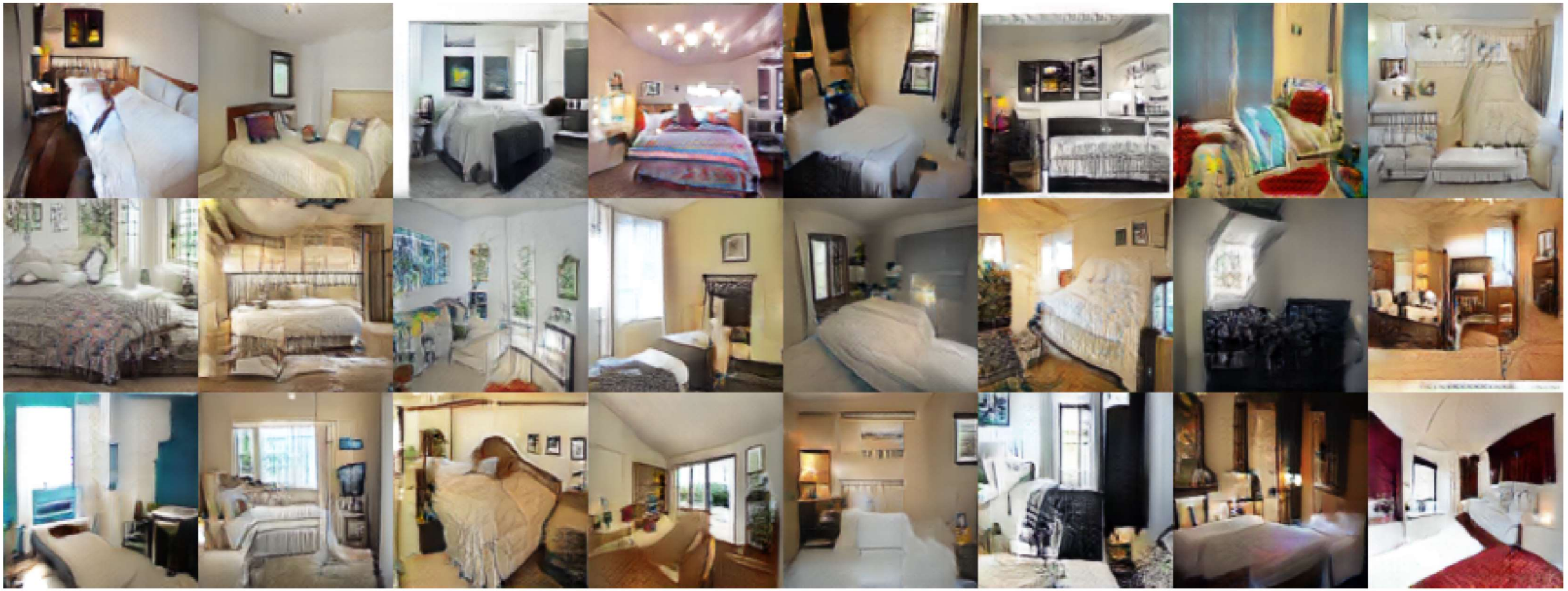
[Goodfelow et al. NIPS 2014]



# Least-Squares GAN

Xudong Mao, Qing Li†, Haoran Xie, Raymond Y.K. Lau and Zhen Wang, ArXiv, Feb. 2017

MILA



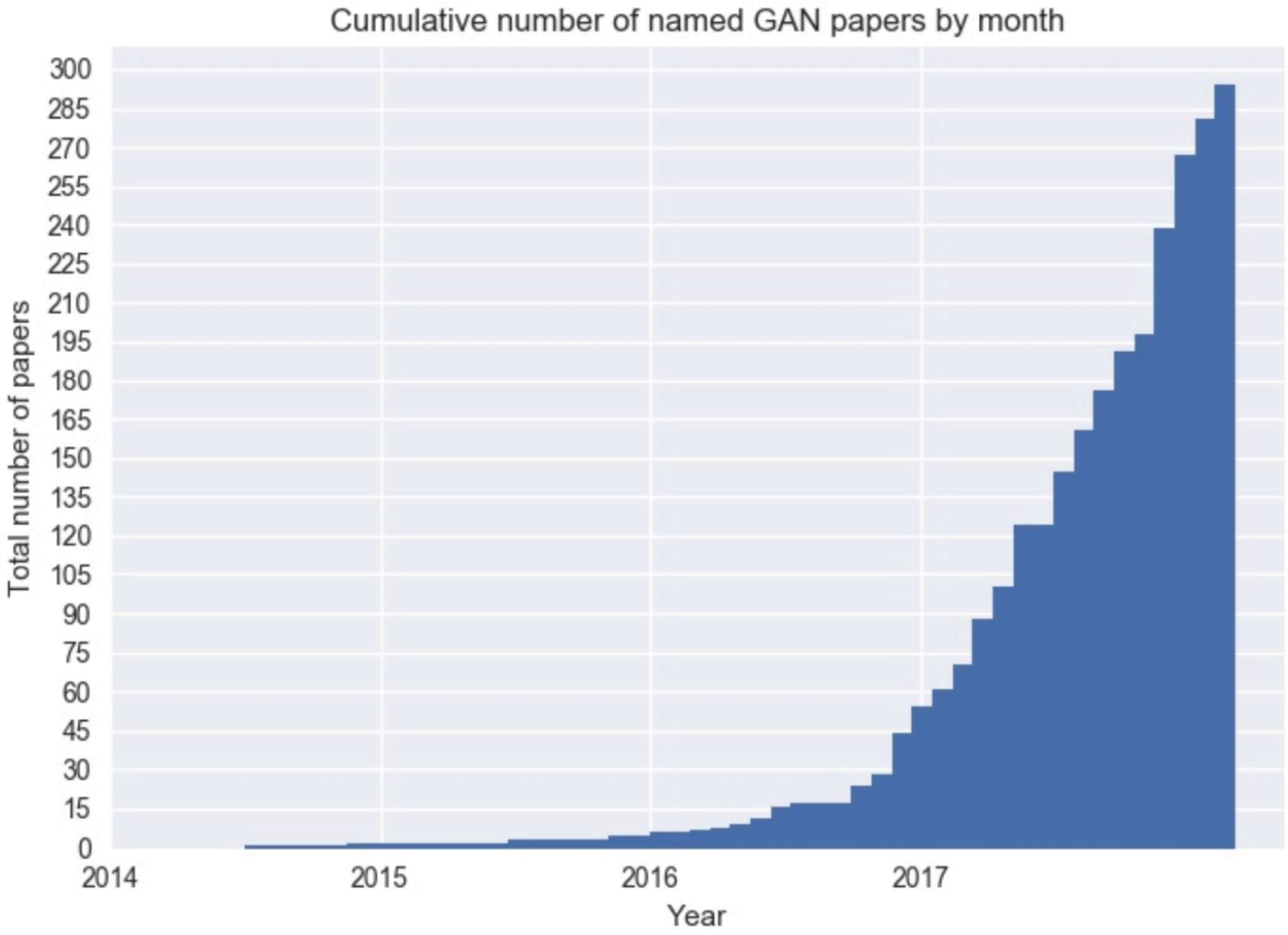
128x128 LSUN bedroom scenes

(slide from Aaron Courville's class)

# 3.5 Years of Progress on Faces


(Brundage et al, 2018)

# An explo-GAN of papers




(slide from Aaron Courville's class)

Explosive growth—All the named GAN variants cumulatively since 2014.

Credit: Bruno Gavranović

(from Deep Hunt, blog by Avinash Hindupur)

# The GAN Zoo – the LDA of deep learning!

- 3D-ED-GAN - Shape Inpainting using 3D Generative Adversarial Network and Recurrent Convolutional Networks
- 3D-GAN - Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling (github)
- 3D-IWGAN - Improved Adversarial Systems for 3D Object Generation and Reconstruction (github)
- 3D-PhysNet - 3D-PhysNet: Learning the Intuitive Physics of Non-Rigid Object Deformations
- 3D-RecGAN - 3D Object Reconstruction from a Single Depth View with Adversarial Learning (github)
- ABC-GAN - ABC-GAN: Adaptive Blur and Control for improved training stability of Generative Adversarial Networks (github)
- ABC-GAN - GANs for LIFE: Generative Adversarial Networks for Likelihood Free Inference
- AC-GAN - Conditional Image Synthesis With Auxiliary Classifier GANs
- acGAN - Face Aging With Conditional Generative Adversarial Networks
- ACGAN - Coverless Information Hiding Based on Generative adversarial networks
- acGAN - On-line Adaptative Curriculum Learning for GANs
- ACtuAL - ACtuAL: Actor-Critic Under Adversarial Learning
- AdaGAN - AdaGAN: Boosting Generative Models
- Adaptive GAN - Customizing an Adversarial Example Generator with Class-Conditional GANs
- AdvEntuRe - AdvEntuRe: Adversarial Training for Textual Entailment with Knowledge-Guided Examples
- AdvGAN - Generating adversarial examples with adversarial networks
- AE-GAN - AE-GAN: adversarial eliminating with GAN
- AE-OT - Latent Space Optimal Transport for Generative Models
- AEGAN - Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets
- AF-DCGAN - AF-DCGAN: Amplitude Feature Deep Convolutional GAN for Fingerprint Construction in Indoor Localizatior System
- AffGAN - Amortised MAP Inference for Image Super-resolution
- AIM - Generating Informative and Diverse Conversational Responses via Adversarial Information Maximization
- AL-CGAN - Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts
- ALI - Adversarially Learned Inference (github)
- AlignGAN - AlignGAN: Learning to Align Cross-Domain Images with Conditional Generative Adversarial Networks
- AlphaGAN - AlphaGAN: Generative adversarial networks for natural image matting
- AM-GAN - Activation Maximization Generative Adversarial Nets
- AmbientGAN - AmbientGAN: Generative models from lossy measurements (github)
- AMC-GAN - Video Prediction with Appearance and Motion Conditions
- AnoGAN - Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery
- APD - Adversarial Distillation of Bayesian Neural Network Posteriors
- UV-GAN - UV-GAN: Adversarial Facial UV Map Completion for Pose-invariant Face Recognition
- VA-GAN - Visual Feature Attribution using Wasserstein GANs
- VAC+GAN - Versatile Auxiliary Classifier with Generative Adversarial Network (VAC+GAN), Multi Class Scenarios
- VAE-GAN - Autoencoding beyond pixels using a learned similarity metric
- VariGAN - Multi-View Image Generation from a Single-View
- VAW-GAN - Voice Conversion from Unaligned Corpora using Variational Autoencoding Wasserstein Generative Adversarial Networks
- VEEGAN - VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning (github)
- VGAN - Generating Videos with Scene Dynamics (github)
- VGAN - Generative Adversarial Networks as Variational Training of Energy Based Models (github)
- VGAN - Text Generation Based on Generative Adversarial Nets with Latent Variable
- ViGAN - Image Generation and Editing with Variational Info Generative Adversarial Networks
- VIGAN - VIGAN: Missing View Imputation with Generative Adversarial Networks
- VoiceGAN - Voice Impersonation using Generative Adversarial Networks
- VOS-GAN - VOS-GAN: Adversarial Learning of Visual-Temporal Dynamics for Unsupervised Dense Prediction in Videos
- VRAL - Variance Regularizing Adversarial Learning
- WaterGAN - WaterGAN: Unsupervised Generative Network to Enable Real-time Color Correction of Monocular Underwater Images
- WaveGAN - Synthesizing Audio with Generative Adversarial Networks
- WaveletGLCA-GAN - Global and Local Consistent Wavelet-domain Age Synthesis
- weGAN - Generative Adversarial Nets for Multiple Text Corpora
- WGAN - Wasserstein GAN (github)
- WGAN-CLS - Text to Image Synthesis Using Generative Adversarial Networks
- WGAN-GP - Improved Training of Wasserstein GANs (github)
- WGAN-L1 - Subsampled Turbulence Removal Network
- WS-GAN - Weakly Supervised Generative Adversarial Networks for 3D Reconstruction
- X-GANs - X-GANs: Image Reconstruction Made Easy for Extreme Cases
- XGAN - XGAN: Unsupervised Image-to-Image Translation for many-to-many Mappings
- ZipNet-GAN - ZipNet-GAN: Inferring Fine-grained Mobile Traffic Patterns via a Generative Adversarial Neural Network
- α-GAN - Variational Approaches for Auto-Encoding Generative Adversarial Networks (github)
- β-GAN - Annealed Generative Adversarial Networks
- Δ-GAN - Triangle Generative Adversarial Networks

# Optimization for GANs

# GAN and effective task-loss

[Goodfellow et al. NIPS 2014]

- Generative Adversarial Networks (GAN):
  - generator gives $q_\theta \in \mathcal{Q}$
  - discriminator tries to distinguish $q_\theta$ from $p$
    - let $D_\phi(x)$ be belief $\in [0,1]$ that $x$ came from true distribution $p$
  - GAN (population) objective:

$$\min_{q_\theta} \underbrace{\max_{D_\phi} \mathbb{E}_{x\sim p}[\log D_\phi(x)] + \mathbb{E}_{x\sim q_\theta}[\log(1 - D_\phi(x))]}$$

- if D non-parametric: get $L_p(\theta) \approx \mathrm{JSD}(p||q_\theta)$
  - (KL-like) => no good!
- but D in practice has limited capacity (e.g. NN): $\phi \in \Phi$
  - => induces a meaningful task loss $L_p(\theta)$ !
  - "parametric adversarial divergence" [Huang et al. arxiv 2017]

# A Variational Inequality Perspective on GANs

[G. Gidel*, H. Berard*, Gaëtan Vignoud, P. Vincent, S. Lacoste-Julien,
arXiv 2018 – to appear at ICLR 2019]

with:

Gauthier Gidel

Hugo Berard

Pascal Vincent

Mila

Université de Montréal

facebook Artificial Intelligence Research

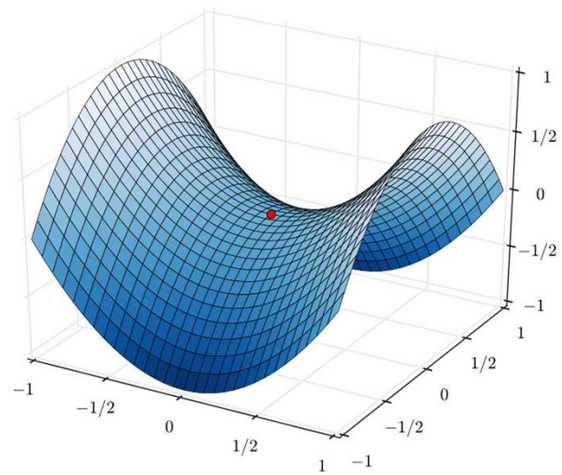
# Optimization perspective of GAN

- Min-Max – two player (zero-sum) game:

generator params

discriminator params

$$\min_{\theta \in \Theta} \max_{\phi \in \Phi} \bar{\mathcal{L}}(\theta, \phi) \, .$$




- Min-max also called *saddle point*

in math. programming

example: WGAN formulation:

$$\min_{\theta \in \Theta} \max_{\phi \in \Phi, ||f_\phi||_L \leq 1} \mathbb{E}_{\boldsymbol{x} \sim p}[f_\phi(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}' \sim q_\theta}[f_\phi(\boldsymbol{x}')].$$

# More general...

as noted by [Mescheder, Nowocin, Geiger; NIPS 17]...

- Non-zero sum game: **different cost functions**

$$\begin{cases} \theta^* \in \underset{\theta \in \Theta}{\arg\min}\, \mathcal{L}^{(\theta)}(\theta, \phi^*) \\ \phi^* \in \underset{\phi \in \Phi}{\arg\min}\, \mathcal{L}^{(\phi)}(\theta^*, \phi)\,. \end{cases}$$

$\mathcal{L}^{(\theta)} = -\mathcal{L}^{(\phi)}$ , then **zero-sum** i.e. min-max)

- Example: non-saturating GAN:

(original working version from [Goodfellow et al. NIPS 14])

$$\mathcal{L}^{(\theta)}(\theta, \phi) \overset{\text{def}}{=} -\mathop{\mathbb{E}}_{\boldsymbol{x}' \sim q_\theta} \log f_\phi(\boldsymbol{x}')$$

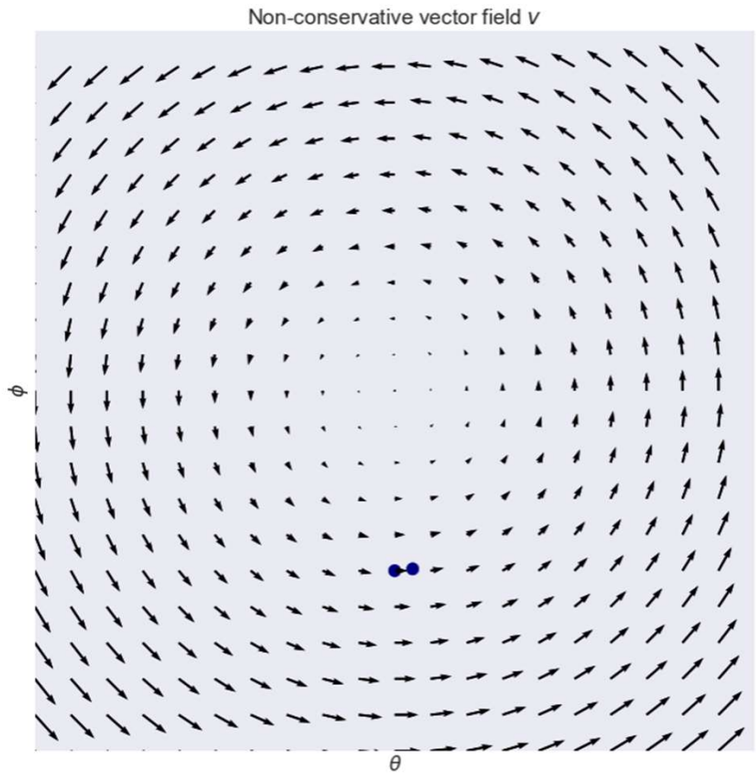
$$\mathcal{L}^{(\phi)}(\theta, \phi) \overset{\text{def}}{=} -\mathop{\mathbb{E}}_{\boldsymbol{x} \sim p} \log f_\phi(\boldsymbol{x}) - \mathop{\mathbb{E}}_{\boldsymbol{x}' \sim q_\theta} \log(1 - f_\phi(\boldsymbol{x}'))\,.$$

# Pet peeve from GAN literature...

- "Saddle points are hard to optimize"
- example from
  Goodfellow NIPS 16 tutorial:
  *simple bilinear objective*

- WGAN with *linear* discriminator
  and generator:

$$\min_{\theta \in \Theta} \max_{\phi \in \Phi, ||\phi|| \leq 1} \phi^T \mathbb{E}[\boldsymbol{X}] - \phi^T \theta \mathbb{E}[\boldsymbol{Z}] \, .$$




http://www.inference.vc/my-notes-on-the-numerics-of-gans/

# Standard methods from Math. Prog.

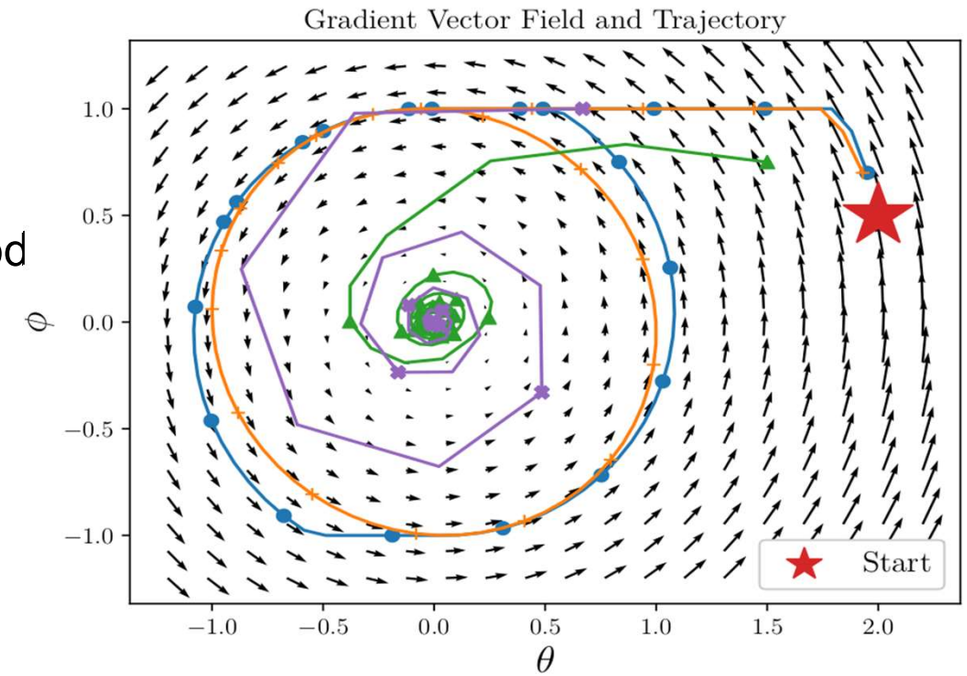
- But saddle points can be optimized!

**non-convergent**

- Blue: Standard gradient method.
- Orange: Alternating gradient method

**convergent**

- Green: Gradient method with **avg**.
- Purple: **Extragradient** method.




from ***variational inequality*** literature

# GAN as a variational inequality

- new perspective on GANs
  - based on stationary conditions
  - relate to vast literature in math. opt. with standard algorithms

- stationary conditions for game:

$$\begin{cases} \theta^* \in \underset{\theta \in \Theta}{\arg\min}\, \mathcal{L}^{(\theta)}(\theta, \phi^*) \\ \phi^* \in \underset{\phi \in \Phi}{\arg\min}\, \mathcal{L}^{(\phi)}(\theta^*, \phi)\,. \end{cases}$$

$$\begin{cases} \nabla_\theta \mathcal{L}^{(\theta)}(\theta^*, \phi^*)^\top (\theta - \theta^*) \geq 0\,, \\ \nabla_\phi \mathcal{L}^{(\phi)}(\theta^*, \phi^*)^\top (\phi - \phi^*) \geq 0 \end{cases} \quad \forall\, (\theta, \phi) \in \Theta \times \Phi.$$

  - interpretation: each player can’t improve cost function

# GAN as a variational inequality

- stationary conditions:

$$\begin{cases} \nabla_\theta \mathcal{L}^{(\theta)}(\theta^*, \phi^*)^\top (\theta - \theta^*) \geq 0\,, \\ \nabla_\phi \mathcal{L}^{(\phi)}(\theta^*, \phi^*)^\top (\phi - \phi^*) \geq 0 \end{cases} \quad \forall\, (\theta, \phi) \in \Theta \times \Phi.$$

- rewrite as:

Defining $\boldsymbol{x} \stackrel{\text{def}}{=} (\theta, \phi)$, $\boldsymbol{x}^* \stackrel{\text{def}}{=} (\theta^*, \phi^*)$, $\mathcal{X} \stackrel{\text{def}}{=} \Theta \times \Phi$ and,

$$F(\boldsymbol{x}) \stackrel{\text{def}}{=} \begin{pmatrix} \nabla_\theta \mathcal{L}^{(\theta)}(\theta, \phi) \\ \nabla_\phi \mathcal{L}^{(\phi)}(\theta, \phi) \end{pmatrix}, \tag{7}$$

the stationary conditions can be compactly formulated as:

$$F(\boldsymbol{x}^*)^\top (\boldsymbol{x} - \boldsymbol{x}^*) \geq 0\,, \quad \forall \boldsymbol{x} \in \mathcal{X}\,. \tag{8}$$

say "$x^*$ solves the **variational inequality**"

# GAN as a variational inequality

takeaways:

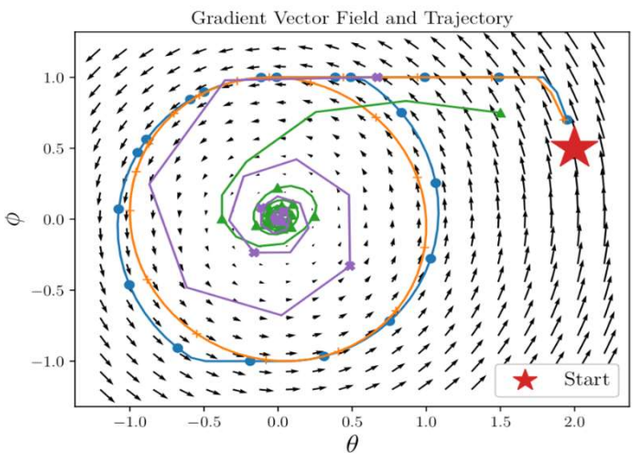
- GAN objective even with different cost functions for the discriminator and the generator,

$$\begin{cases} \theta^* \in \arg\min_{\theta \in \Theta} \mathcal{L}^{(\theta)}(\theta, \phi^*) \\ \phi^* \in \arg\min_{\phi \in \Phi} \mathcal{L}^{(\phi)}(\theta^*, \phi)\,. \end{cases}$$

- Can be formulated as a **variational inequality**.

$$F(\boldsymbol{x}^*)^\top (\boldsymbol{x} - \boldsymbol{x}^*) \geq 0\,, \quad \forall \boldsymbol{x} \in \mathcal{X}\,.$$

- Can import to GANs standard variational inequality algorithms (+ make their stochastic extensions)
- With theoretical guarantees (at least for “convex” cost functions)



# Convergent methods for V.I.

**gradient method**

**with averaging**

[Bruck 1977]

stochastic: [Nemirovski et al. 09]

**extragradient method**

[Korpelich 1976]

stochastic: [Juditsky et al. 11]

- smaller variance

**Algorithm 1** Stochastic Gradient Method (AvgSGD)

Let $x_0 \in \mathcal{X}$
**for** $t = 0 \dots T-1$ **do**
  Sample $\xi_t \sim P$
  Update $\boldsymbol{x}_{t+1} := P_{\mathcal{X}}\big(\boldsymbol{x}_t - \gamma_t F(\boldsymbol{x}_t, \xi_t)\big)$
**end for**
Return $\bar{\boldsymbol{x}}_T = \sum_{t=0}^{T-1} \gamma_t \boldsymbol{x}_t / \sum_{t=0}^{T-1} \gamma_t$

**Algorithm 2** Stochastic Extragradient Method (SEM)

Let $\boldsymbol{x}_0 \in \mathcal{X}$
**for** $t = 0 \dots T-1$ **do**
  Sample $\xi_t \sim P$ ▷ *Extrapolation step*
  Compute $\boldsymbol{x}_{t+\frac{1}{2}} := P_{\mathcal{X}}(\boldsymbol{x}_t - \gamma_t F(\boldsymbol{x}_t, \xi_t))$
  Sample $\xi_{t+\frac{1}{2}} \sim P$ ▷ *Update step*
  Update $\boldsymbol{x}_{t+1} := P_{\mathcal{X}}\big(\boldsymbol{x}_t - \gamma_t F\big(\boldsymbol{x}_{t+\frac{1}{2}}, \xi_{t+\frac{1}{2}}\big)\big)$
**end for**
Return $\bar{\boldsymbol{y}}_T = \sum_{t=0}^{T-1} \gamma_t \boldsymbol{x}_{t+1/2} / \sum_{t=0}^{T-1} \gamma_t$

# More into extragradient

Extragradient update:

$$\begin{cases} \boldsymbol{\omega}_{t+\frac{1}{2}} = \boldsymbol{\omega}_t - \gamma_t F(\boldsymbol{\omega}_t) & \text{(extrapolation step)} \\ \boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t - \gamma_t F(\boldsymbol{\omega}_{t+\frac{1}{2}}) & \text{(update step)} \end{cases}$$

Can be seen as first-order approximation of **implicit** scheme:

$$\text{Implicit step:} \quad \boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t - \eta F(\boldsymbol{\omega}_{t+1})$$

# More into extragradient

Extragradient update:

$$\begin{cases} \boldsymbol{\omega}_{t+\frac{1}{2}} = \boldsymbol{\omega}_t - \gamma_t F(\boldsymbol{\omega}_t) & \text{(extrapolation step)} \\ \boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t - \gamma_t F(\boldsymbol{\omega}_{t+\frac{1}{2}}) & \text{(update step)} \end{cases}$$

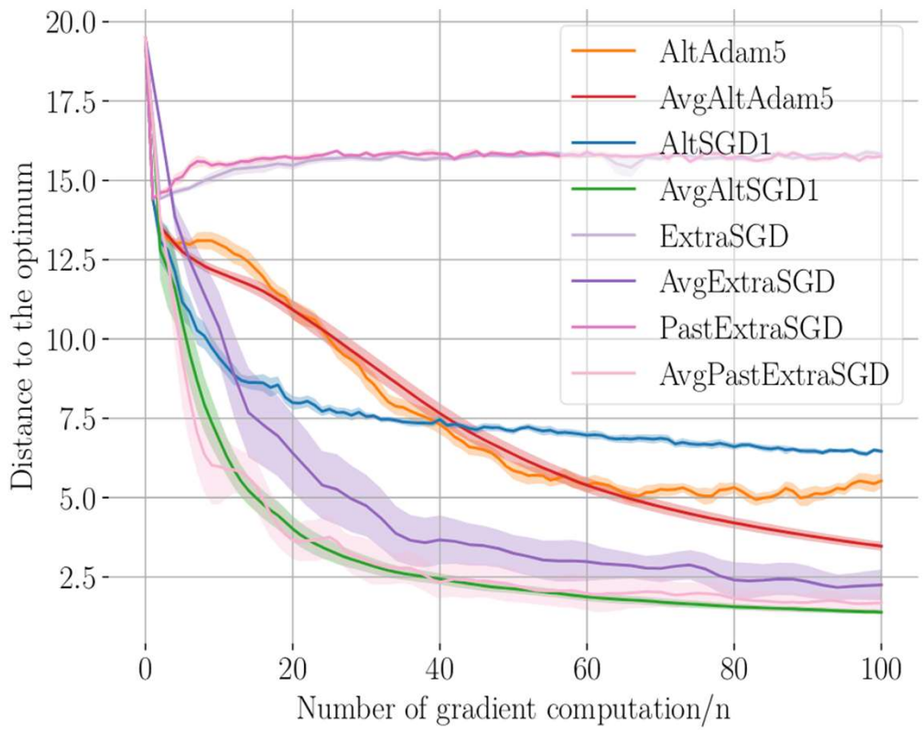
Extrapolation *from the past* alternative (save factor of 2):

$$\begin{cases} \boldsymbol{\omega}_{t+\frac{1}{2}} = \boldsymbol{\omega}_t - \gamma_t F(\boldsymbol{\omega}_{t-\frac{1}{2}}) & \text{(re-use from previous step)} \\ \boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t - \gamma_t F(\boldsymbol{\omega}_{t+\frac{1}{2}}) & \text{(same as extragradient)} \end{cases}$$

Equivalent to methods proposed in the online learning literature:
*"optimistic mirror descent"*
[Chiang et al. 2012], [Rakhlin & Sridharan 2013],..., [Daskalakis et al. 2018]

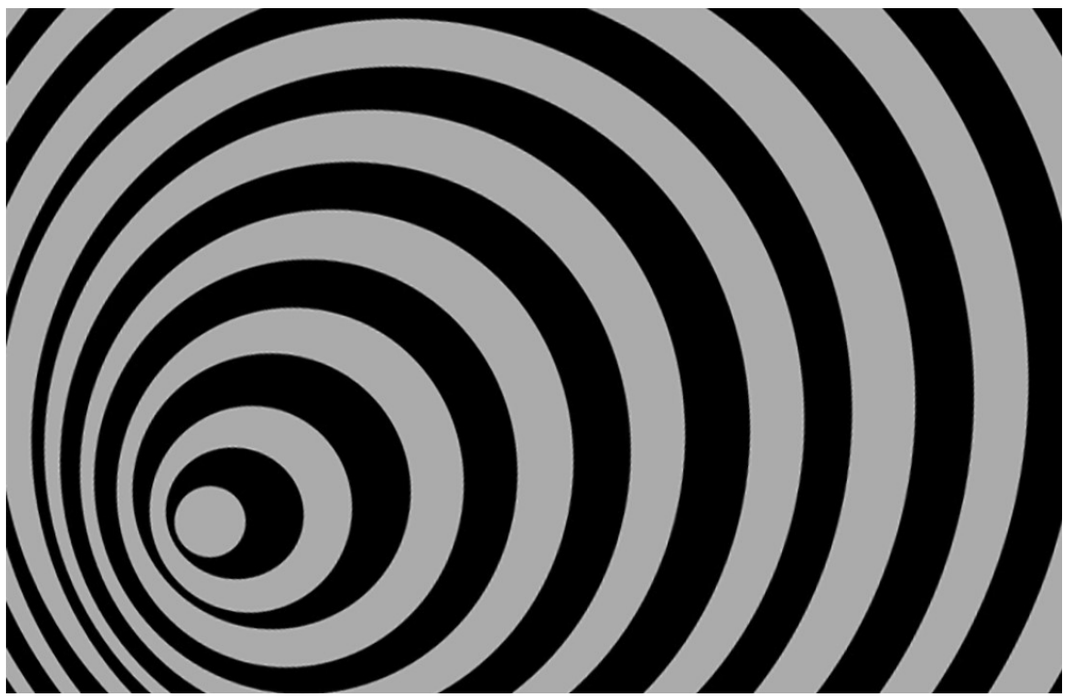
# Bilinear stochastic experiment

bilinear stochastic objective: $$\frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{x}^\top \boldsymbol{M}^{(i)}\boldsymbol{y} + \boldsymbol{x}^\top \boldsymbol{a}^{(i)} + \boldsymbol{y}^\top \boldsymbol{b}^{(i)}\right).$$



# Entering the deep learning hacks

## twilight zone!



**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** $t = 0 \dots T-1$ **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 $t$: $\boldsymbol{\omega}_{t-1/2} \leftarrow P_\Omega[\boldsymbol{\omega}_t - \eta \frac{m_{t-1/2}}{\sqrt{v_{t-1/2}}+\epsilon}]$

Extrapolation (Adam style)

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: $\hat{m}_t \leftarrow m_t/(1-\beta_1^{2t})$, $\hat{v}_t \leftarrow v_t/(1-\beta_2^{2t})$

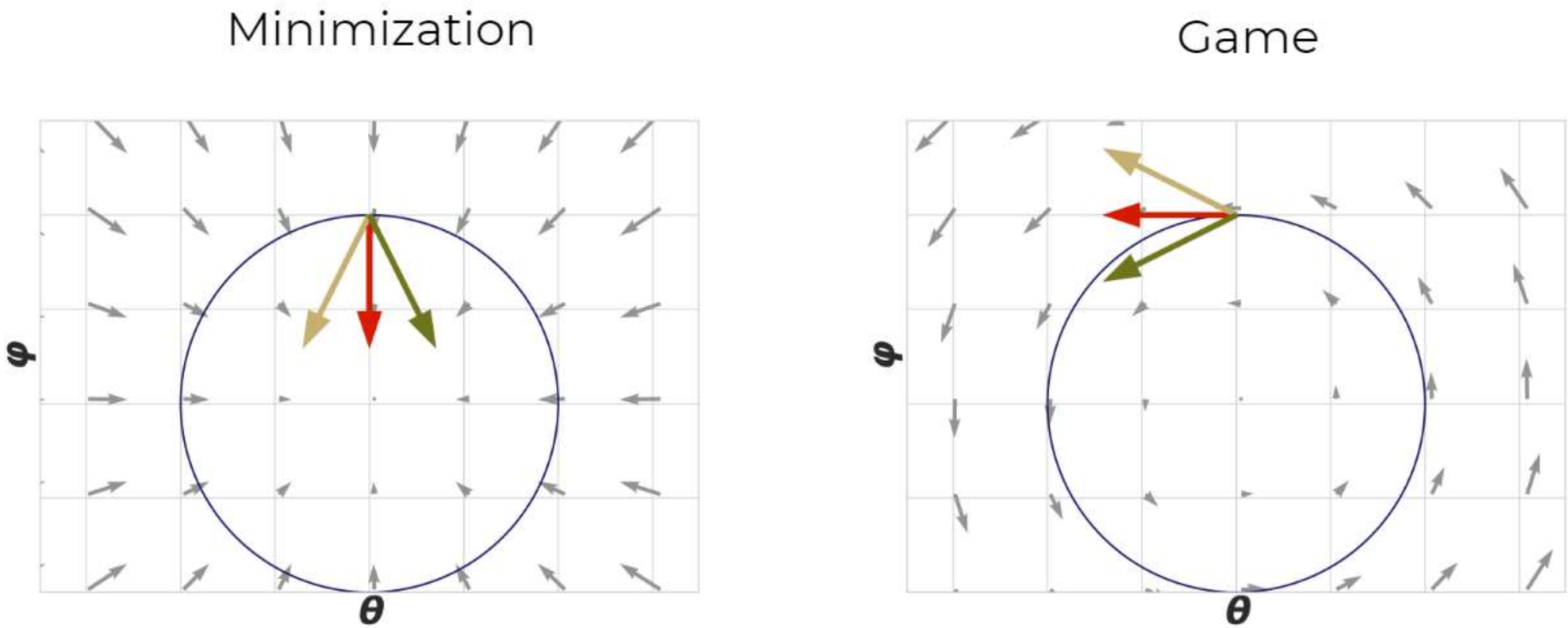
Perform *update* step from the iterate at time $t$: $\boldsymbol{\omega}_{t+1} \leftarrow P_\Omega[\boldsymbol{\omega}_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon}]$

Update (Adam style)

**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)

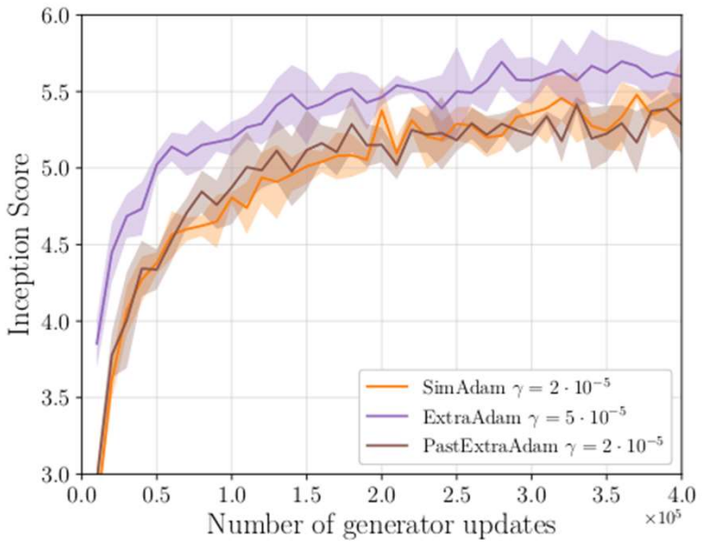
# Sidenote: issue of noise in games...




Using *variance reduction*, were able to make GANs work with SGD!
See [Chavdarova et al., on arXiv hopefully soon!]

# Experimental Results: WGAN on CIFAR10

Inception Score vs
**nb of generator updates**




Inception Score on CIFAR10

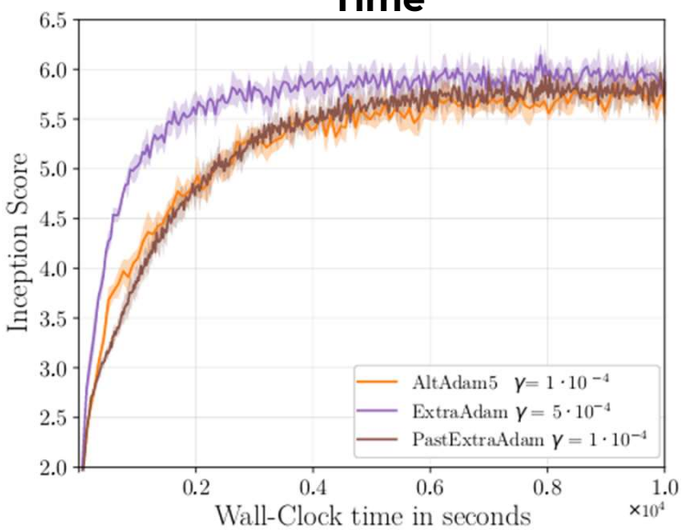
| Model | WGAN | | |
|---|---|---|---|
| Method | no averaging | uniform avg | EMA |
| SimAdam | *6.05 ± .12* | 5.83 ± .16 | 6.08 ± .10 |
| AltAdam5 | *5.45 ± .08* | 5.72 ± .06 | 5.49 ± .05 |
| ExtraAdam | **6.38 ± .09** | **6.38 ± .20** | **6.37 ± .08** |
| PastExtraAdam | 5.98 ± 0.15 | 6.07 ± 0.19 | 6.01 ± 0.11 |
| OptimAdam | 5.74 ± 0.10 | 5.80 ± 0.08 | 5.78 ± 0.05 |

**Extragradient Methods**

**Averaging**

# Experimental Results: WGAN-GP on CIFAR10

Inception Score vs **Time**




Inception Score on CIFAR10

| Model | WGAN-GP | | |
|---|---|---|---|
| Method | no averaging | no averaging | uniform avg |
| SimAdam | *6.05 ± .12* | *6.00 ± .07* | 6.01 ± .08 |
| AltAdam5 | *5.45 ± .08* | *6.25 ± .05* | **6.51 ± .05** |
| ExtraAdam | **6.38 ± .09** | 6.22 ± .04 | 6.35 ± .05 |
| PastExtraAdam | 5.98 ± 0.15 | 6.27 ± 0.06 | 6.23 ± 0.13 |
| OptimAdam | 5.74 ± 0.10 | - | - |

**Extragradient Methods**

**Averaging**

# Latest results

| Model | WGAN (DCGAN) | | | WGAN-GP (ResNet) | | |
|---|---|---|---|---|---|---|
| Method | no avg | uniform avg | EMA | no avg | uniform avg | EMA |
| SimAdam | *6.05 ± .12* | 5.85 ± .16 | 6.08 ± .10 | *7.51 ± .17* | 7.68 ± .43 | 7.60 ± .17 |
| AltAdam5 | *5.45 ± .08* | 5.72 ± .06 | 5.49 ± .05 | *7.57 ± .02* | 8.01 ± .05 | 7.66 ± .03 |
| ExtraAdam | **6.38 ± .09** | **6.38 ± .20** | **6.37 ± .08** | 7.90 ± .11 | **8.47 ± .10** | 8.13 ± .07 |
| PastExtraAdam | 5.98 ± .15 | 6.07 ± .19 | 6.01 ± .11 | 7.84 ± .06 | 8.01 ± .09 | 7.99 ± .03 |
| OptimAdam | *5.74 ± .10* | 5.80 ± .08 | 5.78 ± .05 | *7.98 ± .08* | 8.18 ± .09 | 8.10 ± .06 |

Frechet Inception Distance:

| Model | WGAN-GP (ResNet) | | |
|---|---|---|---|
| Method | no averaging | uniform avg | EMA |
| SimAdam | *23.74 ± 2.79* | 26.29 ± 5.56 | 21.89 ± 2.51 |
| AltAdam5 | *21.65 ± .66* | 19.91 ± .43 | 20.69 ± .37 |
| ExtraAdam | 19.42 ± .15 | 18.13 ± .51 | **16.78 ± .21** |
| PastEAdam | 19.95 ± .38 | 22.45 ± .93 | 17.85 ± .40 |
| OptimAdam | *18.88 ± .55* | 21.23 ± 1.19 | 16.91 ± .32 |

# Discussion

- New framework called variational inequality to consider GAN objectives.
- Tapped into the optimization literature to bring new techniques to train GANs
- With caveats:
  - Averaging seems to consistently improve the inception score at convergence,
  - Extragradient looks to be more stable for step size tuning.
- Just considered two standard algorithms.
- Could design algorithm specific to GANs inspired from variational inequalities.
  - current work: variance reduction!

Mila

Université de Montréal

# Negative Momentum for Improved Game Dynamics

Gauthier Gidel*, Reyhane Askari Hemmat*, Mohammad Pezeshki, Gabriel Huang, Remi Lepriol, Simon Lacoste-Julien, Ioannis Mitliagkas

https://arxiv.org/abs/1807.04740
to appear at AISTATS 2019

Game dynamics are ~~weird~~

fascinating

# Start with optimization dynamics

# Optimization

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta} \in \boldsymbol{\theta}} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta})$$

Smooth, differentiable cost function, L

→ Looking for stationary (fixed) points (gradient is 0)

→ Gradient descent

# Optimization

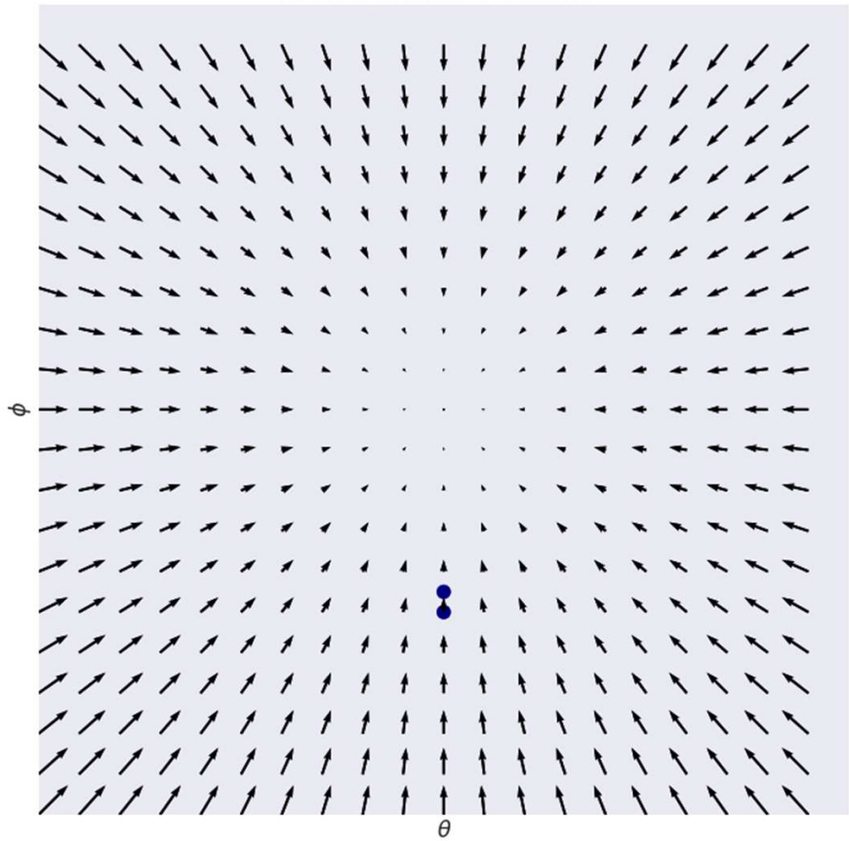
$$\boldsymbol{v}(\boldsymbol{\theta}) = \nabla \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta})$$

Conservative vector field

→

Straightforward dynamics

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \boldsymbol{v}(\boldsymbol{\theta}_t)$$



# Gradient descent

$$\boldsymbol{v}(\boldsymbol{\theta}) = \nabla \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta})$$

Conservative vector field

→

Straightforward dynamics

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \boldsymbol{v}(\boldsymbol{\theta}_t)$$

Fixed-point analysis

$$F_\eta(\boldsymbol{\theta}) = \boldsymbol{\theta} - \eta \boldsymbol{v}(\theta)$$

Jacobian of operator

$$\nabla F_\eta(\boldsymbol{\theta}) = I - \eta \nabla \boldsymbol{v}(\theta)$$

**Hessian of objective, L**

# Local convergence

**Theorem 1** (Prop. 4.4.1 Bertsekas [1999]). *If the spectral radius* $\rho_{\max} \overset{def}{=} \rho(\nabla F_\eta(\boldsymbol{\omega}^*)) < 1$, *then, for* $\boldsymbol{\omega}_0$ *in a neighborhood of* $\boldsymbol{\omega}^*$, *the distance of* $\boldsymbol{\omega}_t$ *to the stationary point* $\boldsymbol{\omega}^*$ *converges at a linear rate of* $\mathcal{O}\left((\rho_{\max} + \epsilon)^t\right)$, $\forall \epsilon > 0$.

Eigenvalues of op. Jacobian

$$\lambda(\nabla F_\eta(\boldsymbol{\theta})) = 1 - \eta\lambda(\nabla \boldsymbol{v}(\theta))$$

If $\rho(\theta^*)=\max |\lambda(\theta^*)|<1$, then fast local convergence

Jacobian of operator

$$\nabla F_\eta(\boldsymbol{\theta}) = I - \eta \nabla \boldsymbol{v}(\theta)$$

**Hessian of objective, L**
**Symmetric, real-eigenvalues**

# Momentum (heavy ball, Polyak 1964)

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \boldsymbol{v}(\boldsymbol{\theta}_t) + \beta(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1})$$

Jacobian of momentum operator

$$\nabla F_{\eta,\beta}(\boldsymbol{\theta}_t, \boldsymbol{\theta}_{t-1}) = \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{0}_n \\ \boldsymbol{I}_n & \boldsymbol{0}_n \end{bmatrix} - \eta \begin{bmatrix} \nabla \boldsymbol{v}(\boldsymbol{\theta}_t) & \boldsymbol{0}_n \\ \boldsymbol{0}_n & \boldsymbol{0}_n \end{bmatrix} + \beta \begin{bmatrix} \boldsymbol{I}_n & -\boldsymbol{I}_n \\ \boldsymbol{0}_n & \boldsymbol{0}_n \end{bmatrix}$$

**Non-symmetric, with complex eigenvalues**
**→ Rotations in augmented state-space**

# Games

# Games

Nash Equilibrium

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta} \in \boldsymbol{\theta}} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\theta}, \boldsymbol{\varphi}^*)$$

$$\boldsymbol{\varphi}^* \in \arg\min_{\boldsymbol{\varphi} \in \boldsymbol{\varphi}} \mathcal{L}^{(\boldsymbol{\varphi})}(\boldsymbol{\theta}^*, \boldsymbol{\varphi})$$

Smooth, differentiable L

→ Looking for local Nash eq

→ Gradient descent

→ **Simultaneous**

→ **Alternating**

# Game dynamics

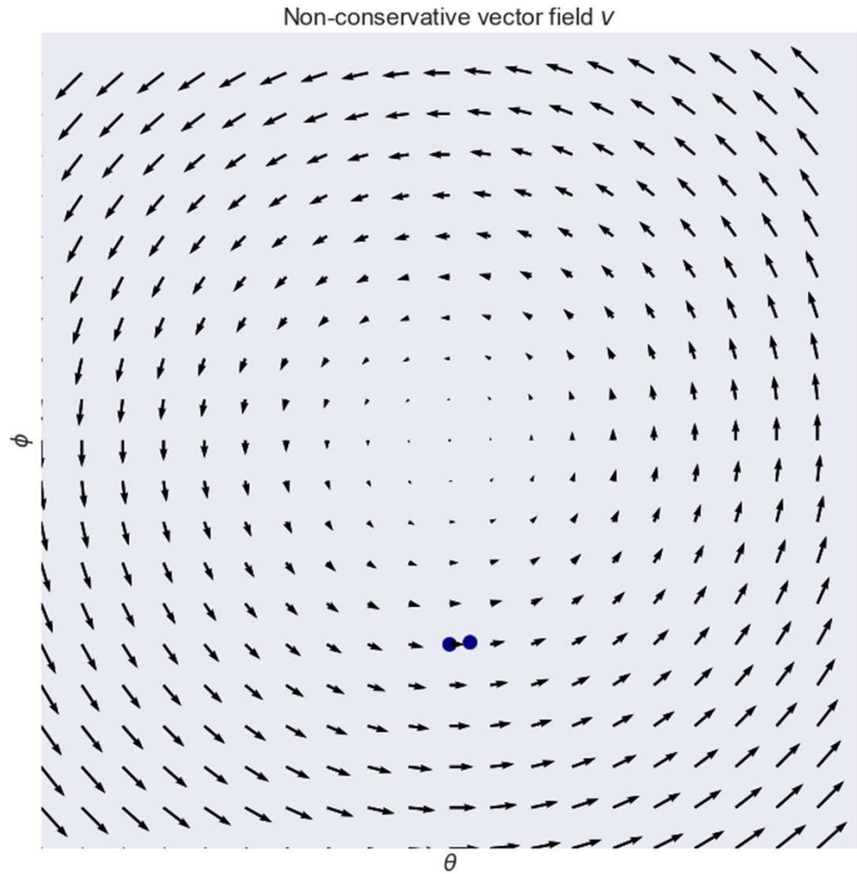
$$\boldsymbol{v}(\boldsymbol{\varphi}, \boldsymbol{\theta}) := \begin{bmatrix} \nabla_{\boldsymbol{\varphi}} \mathcal{L}^{(\boldsymbol{\varphi})}(\boldsymbol{\varphi}, \boldsymbol{\theta}) \\ \nabla_{\boldsymbol{\theta}} \mathcal{L}^{(\boldsymbol{\theta})}(\boldsymbol{\varphi}, \boldsymbol{\theta}) \end{bmatrix}$$

Non-conservative vector field

→

Rotational dynamics

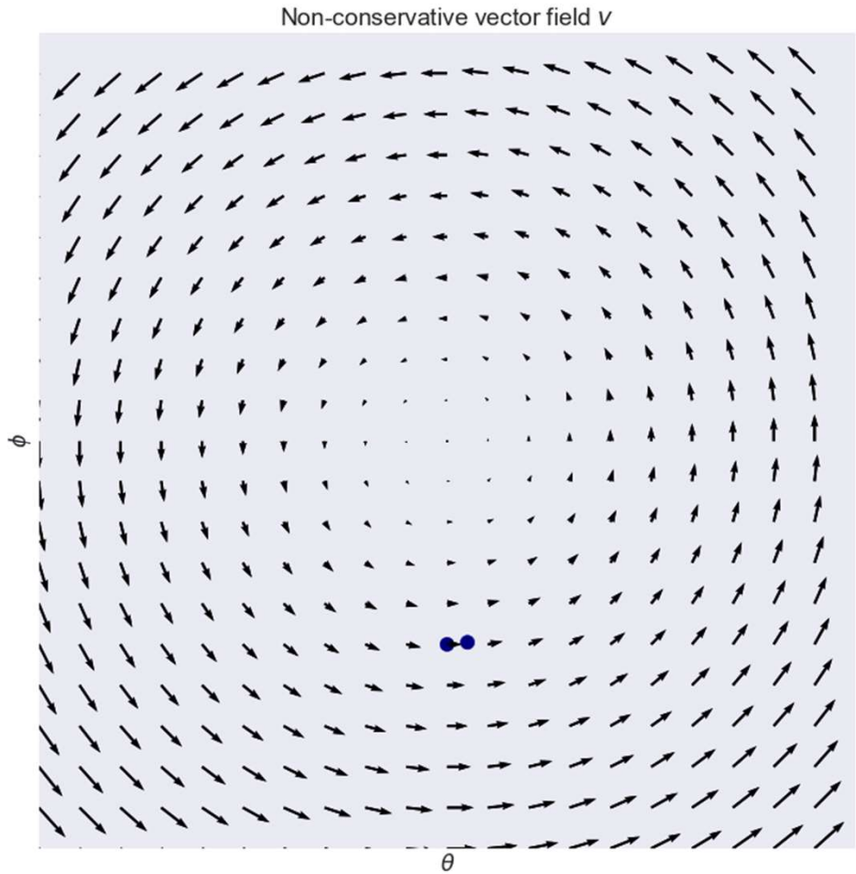
$$F_\eta(\boldsymbol{\varphi}, \boldsymbol{\theta}) \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{\varphi} & \boldsymbol{\theta} \end{bmatrix}^\top - \eta \, \boldsymbol{v}(\boldsymbol{\varphi}, \boldsymbol{\theta})$$



# Game dynamics under gradient descent

$$F_\eta(\boldsymbol{\varphi}, \boldsymbol{\theta}) \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{\varphi} & \boldsymbol{\theta} \end{bmatrix}^\top - \eta \, \boldsymbol{v}(\boldsymbol{\varphi}, \boldsymbol{\theta})$$

**Jacobian is non-symmetric, with complex eigenvalues → Rotations in decision space**




Games already demonstrate rotational dynamics.

Can momentum help/hurt??

# Spoiler

Positive momentum can be bad for adversarial games

Recent work reduced the momentum.

-> Not an accident

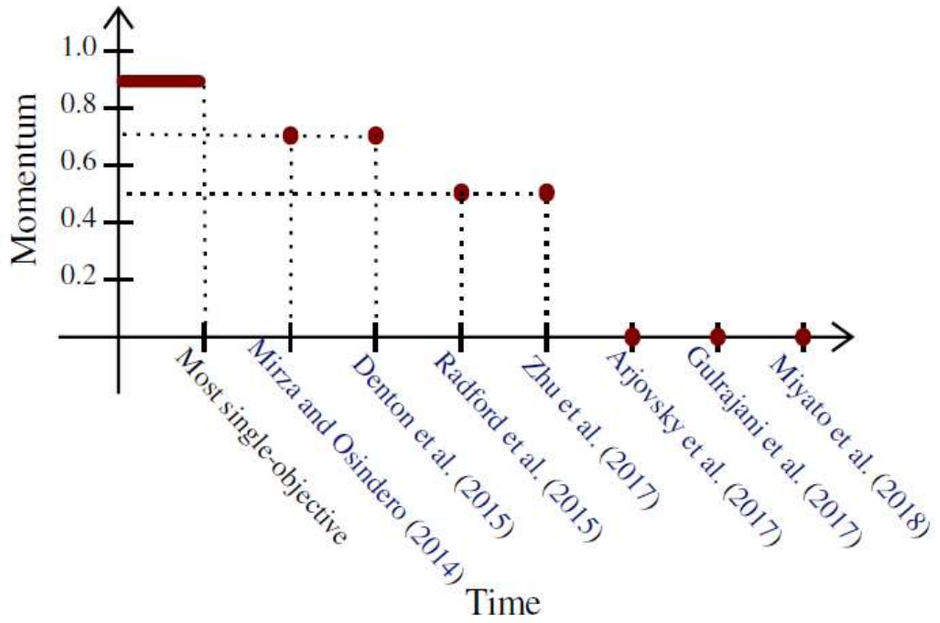



Figure 1: Decreasing trend in the value of momentum used for training GANs across time.

# Our results

Negative momentum is “optimal”on simple bilinear game

Negative momentum values are locally preferrable near 0 on a more general class of games

Negative momentum is empirically best for certain zero sum games like “saturating GANs”

# Momentum on games

Recall Polyak's momentum (on top of simultaneous grad. desc.):

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta \boldsymbol{v}(\boldsymbol{x}_t) + \beta(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}), \quad \boldsymbol{x}_t = (\boldsymbol{\theta}_t, \boldsymbol{\phi}_t)$$

Fixed point operator requires a **state augmentation**:
(because we need previous iterate)

$$F_{\eta,\beta}(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) := \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{0}_n \\ \boldsymbol{I}_n & \boldsymbol{0}_n \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{x}_{t-1} \end{bmatrix} - \eta \begin{bmatrix} \boldsymbol{v}(\boldsymbol{x}_t) \\ \boldsymbol{0}_n \end{bmatrix} + \beta \begin{bmatrix} \boldsymbol{I}_n & -\boldsymbol{I}_n \\ \boldsymbol{0}_n & \boldsymbol{0}_n \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{x}_{t-1} \end{bmatrix}$$

# Bilinear game

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\varphi}} \boldsymbol{\theta}^\top \boldsymbol{A} \boldsymbol{\varphi}$$

| Method | $\beta$ | Bounded | Converges |
|---|---|---|---|
| Simultaneous | $\beta \in \mathbb{R}$ | ✗ | ✗ |
| Alternated | $>0$ | ✗ | ✗ |
| | $0$ | ✓ | ✗ |
| | $<0$ | ✓ | ✓ |

# “Proof by picture”

Gradient descent

→

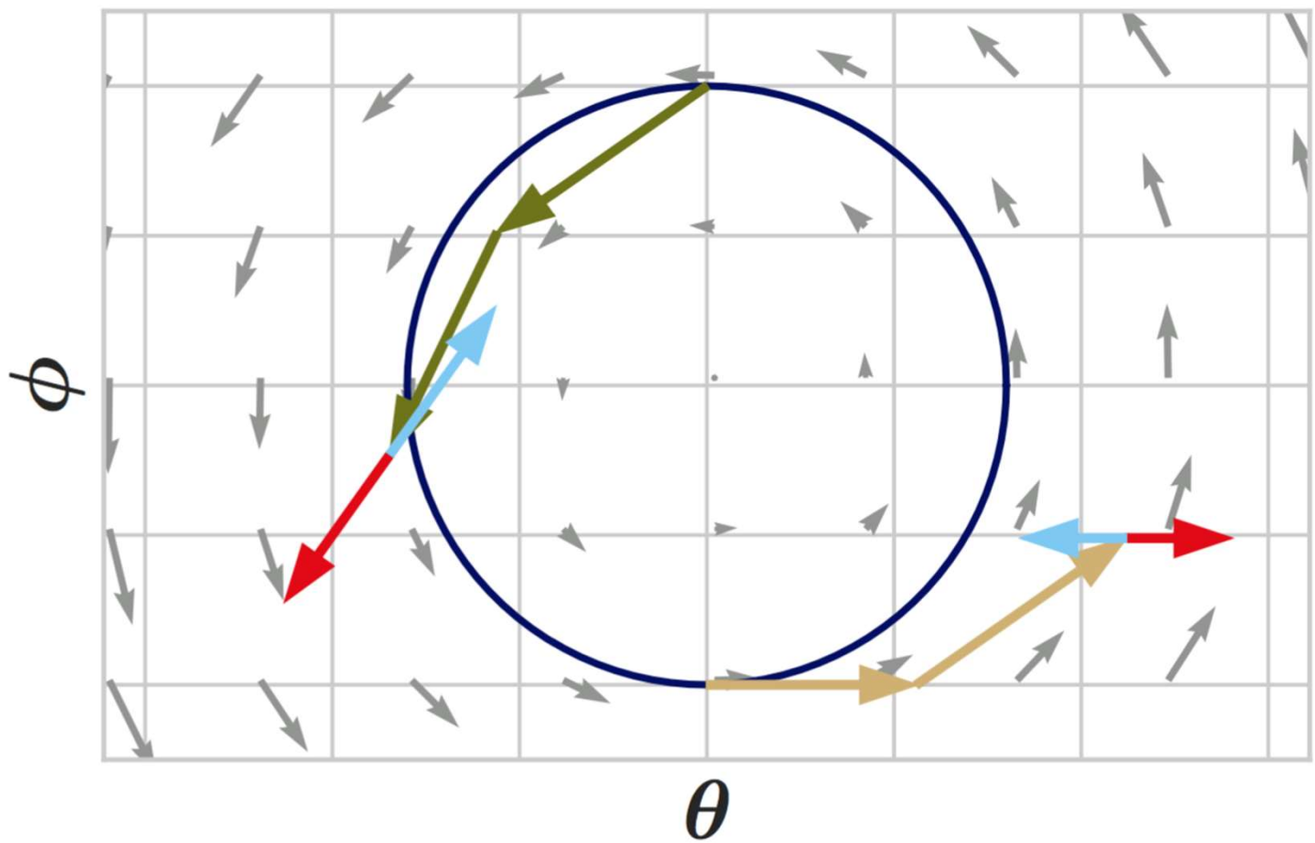
**Simultaneous**

→ **Alternating**

Momentum

→ **Positive**

→ **Negative**



# General games

## Without momentum

$$F_\eta(\boldsymbol{\varphi}, \boldsymbol{\theta}) \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{\varphi} & \boldsymbol{\theta} \end{bmatrix}^\top - \eta\, \boldsymbol{v}(\boldsymbol{\varphi}, \boldsymbol{\theta})$$

$$\nabla F_\eta(\boldsymbol{\theta}) = I - \eta \nabla \boldsymbol{v}(\theta)$$

$$\mathrm{Sp}(\nabla F_\eta(\boldsymbol{\omega}^*)) = \{1 - \eta\lambda | \lambda \in \mathrm{Sp}(\nabla \boldsymbol{v}(\boldsymbol{\omega}^*))\}$$

## With momentum

$$F_{\eta,\beta}(\boldsymbol{\omega}_t, \boldsymbol{\omega}_{t-1}) = (\boldsymbol{\omega}_{t+1}, \boldsymbol{\omega}_t)$$
$$\boldsymbol{\omega}_{t+1} := \boldsymbol{\omega}_t - \eta \boldsymbol{v}(\boldsymbol{\omega}_t) + \beta(\boldsymbol{\omega}_t - \boldsymbol{\omega}_{t-1})$$

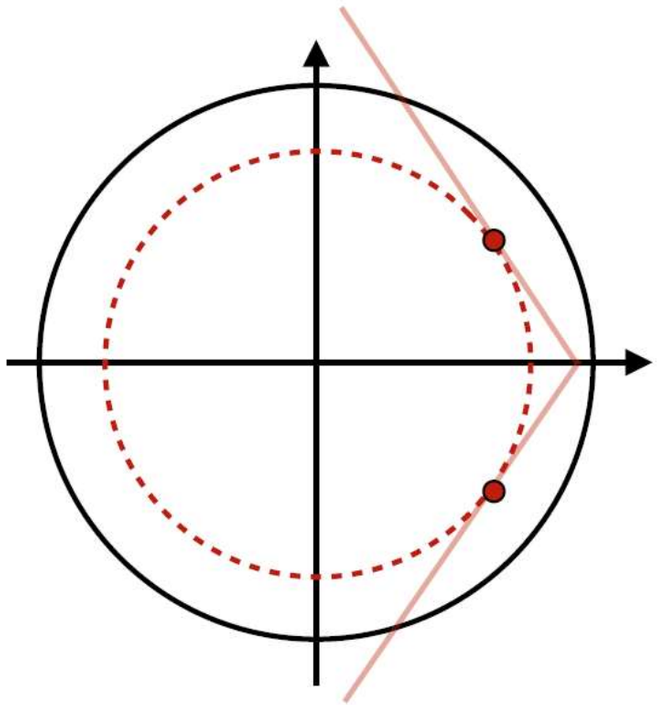
$$\begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{0}_n \\ \boldsymbol{I}_n & \boldsymbol{0}_n \end{bmatrix} - \eta \begin{bmatrix} \nabla \boldsymbol{v}(\boldsymbol{\omega}_t) & \boldsymbol{0}_n \\ \boldsymbol{0}_n & \boldsymbol{0}_n \end{bmatrix} + \beta \begin{bmatrix} \boldsymbol{I}_n & -\boldsymbol{I}_n \\ \boldsymbol{0}_n & \boldsymbol{0}_n \end{bmatrix}$$

$$\mu_\pm(\beta, \eta, \lambda) := (1 - \eta\lambda + \beta)\frac{1 \pm \Delta^{\frac{1}{2}}}{2},$$

Shifting the eigen-values

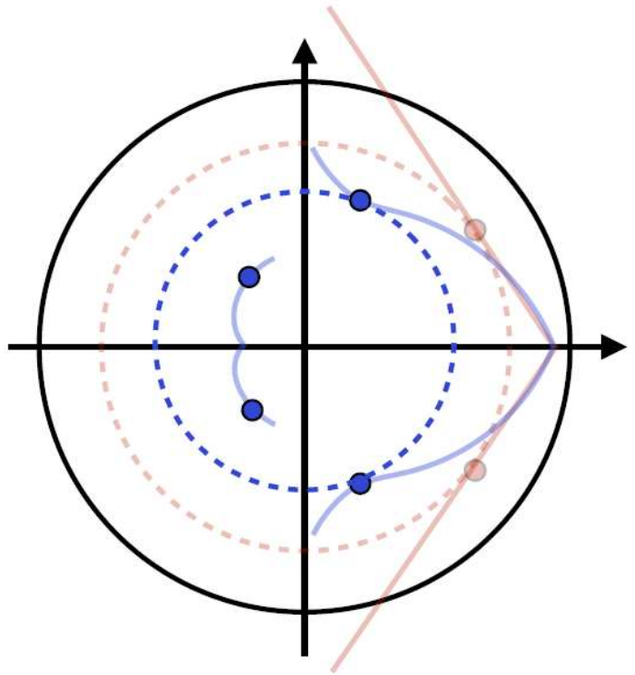
Theorem 4: Given a large class of games, a small negative momentum can improve the magnitude of the limiting eigenvalues

## Without momentum



$$\mathrm{Sp}(\nabla F_\eta(\boldsymbol{\omega}^*)) = \left\{1 - \eta\lambda | \lambda \in \mathrm{Sp}(\nabla \boldsymbol{v}(\boldsymbol{\omega}^*))\right\}$$

## With negative momentum

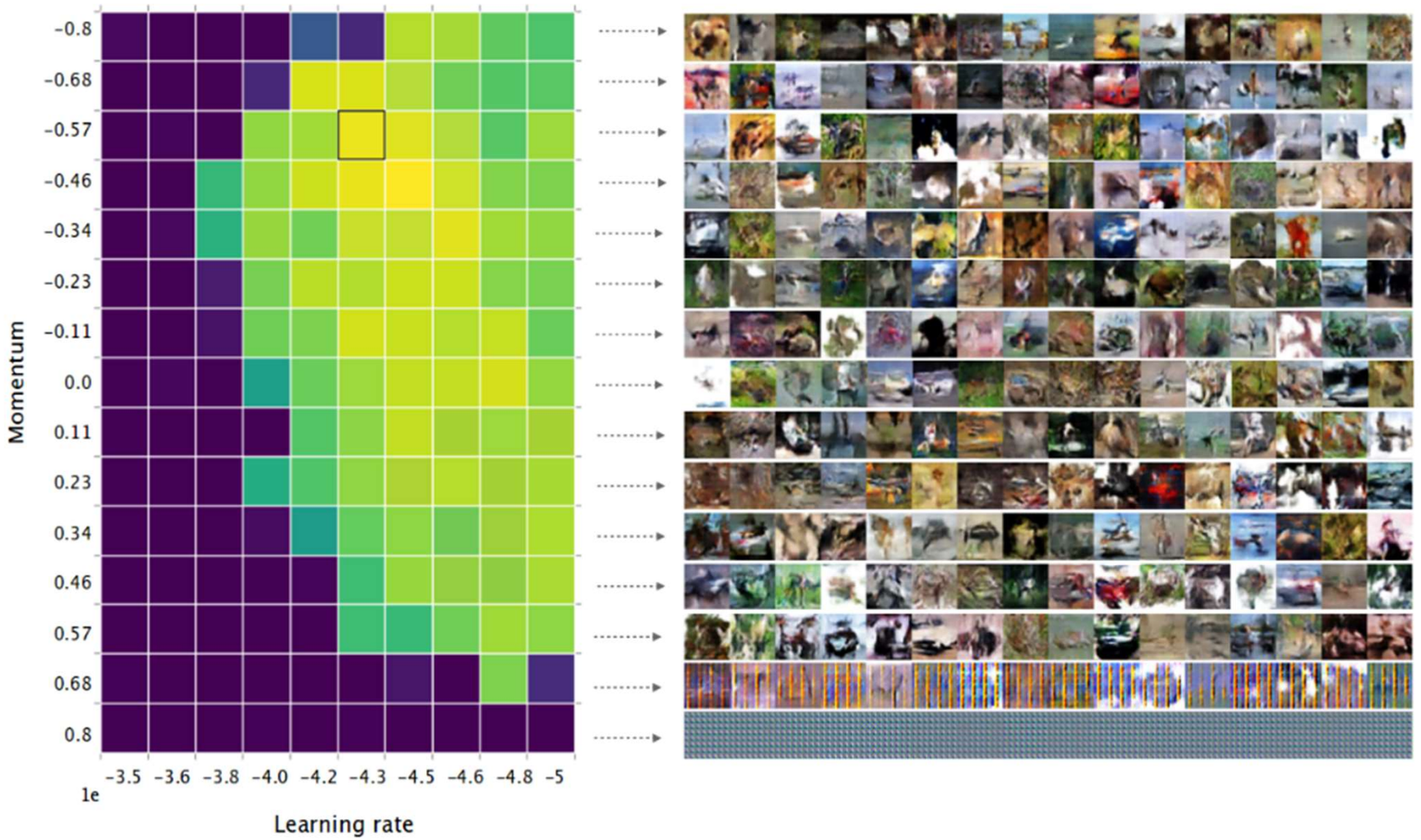


$$\mu_\pm(\beta, \eta, \lambda) := (1 - \eta\lambda + \beta)\frac{1 \pm \Delta^{\frac{1}{2}}}{2},$$

# Empirical results

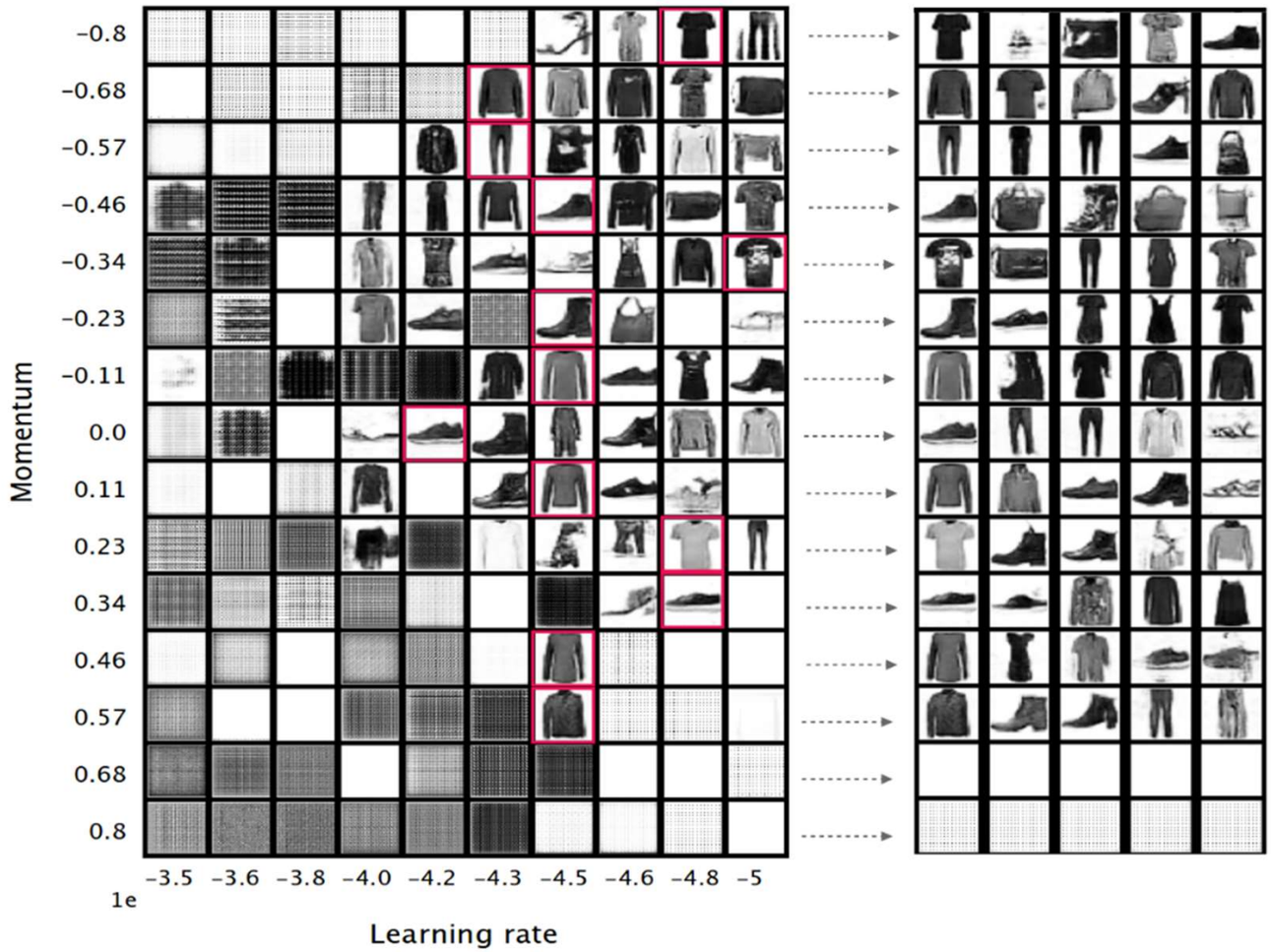
# What happens in practice ?

CIFAR-10:



# What happens in practice ?

Fashion MNIST:



# Negative Momentum

To sum up:

- Negative momentum seems to improve the behaviour due to “bad” eigenvalues.
- ‘’Optimal’’ for a class of games
- Empirically optimal on “saturating” GANs


https://sgo-workshop.github.io/

Thank you!