# Learning Dynamic Generative Models via Causal Optimal Transport

Beatrice Acciaio
London School of Economics

with Michael Munn (Google NY), and Tianlin Xu (LSE)

Model Uncertainty in Risk Management
31 January 2020, Natixis, Paris

# Idea in a nutshell

Let me tell you where we are heading to...

- We observe an **i.i.d. sample**, where every element is a path/evolution of some process of interest (asset price process, LOB, volatility surface, audio data,...)
- We want to understand the **distribution** underlying the sample
- We want to train a generator to:
  - **generate** new real-looking samples (e.g. to extend the available data set for training and evaluation of trading strategies)
  - **predict** the evolution of the path given that we observe part of it
- For this we use some **dynamic modification of GANs**

# Outline

- Introduction to Generative Adversarial Networks (GANs)
- Our toolkit: Causal Optimal Transport (COT)
- Dynamic GANs via COT
- Applications

# Outline

- Introduction to Generative Adversarial Networks (GANs)
- Our toolkit: Causal Optimal Transport (COT)
- Dynamic GANs via COT
- Applications

# Generative Adversarial Networks

Generative: train a **Generator G** to learn data distribution from an i.i.d. sample of observations (training data)

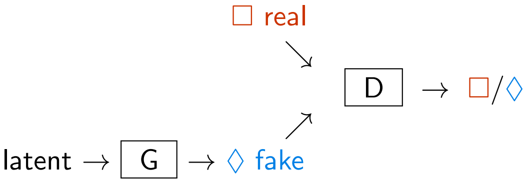
Adversarial: set a **Discriminator D** against G, to stimulate G to do a better job

# Generative Adversarial Networks

Generative: train a **Generator G** to learn data distribution from an i.i.d. sample of observations (training data)

Adversarial: set a **Discriminator D** against G, to stimulate G to do a better job

- In a loop, we train: **G** to generate real-looking samples, and
  **D** to recognize whether an element comes from real data or is fake (generated by G).
- G and D compete with each other, which drives both of them to improve, until the generated samples are indistinguishable from the genuine data (zero-sum game).



# Generative Adversarial Networks

- training data $\{x^i\}_{i=1}^N$ on $\mathcal{X}$, empirical distribution $\mu = \frac{1}{N}\sum_{i=1}^N \delta_{x^i}$
- latent space $\mathcal{Z}$, $\dim(\mathcal{Z}) \ll \dim(\mathcal{X})$, noise distribution $\zeta \in \mathcal{P}(\mathcal{Z})$
- $g_\theta : \mathcal{Z} \to \mathcal{X}$ generates samples, $\nu_\theta = g_{\theta\#}\zeta \in \mathcal{P}(\mathcal{X})$ (cf. $\mu$)
- $f_\varphi : \mathcal{X} \to [0,1]$ outputs high value if D believes input likely to be real

# Generative Adversarial Networks

- training data $\{x^i\}_{i=1}^N$ on $\mathcal{X}$, empirical distribution $\mu = \frac{1}{N}\sum_{i=1}^N \delta_{x^i}$
- latent space $\mathcal{Z}$, $\dim(\mathcal{Z}) \ll \dim(\mathcal{X})$, noise distribution $\zeta \in \mathcal{P}(\mathcal{Z})$
- $g_\theta : \mathcal{Z} \to \mathcal{X}$ generates samples, $\nu_\theta = g_{\theta\#}\zeta \in \mathcal{P}(\mathcal{X}) \quad$ (cf. $\mu$)
- $f_\varphi : \mathcal{X} \to [0,1]$ outputs high value if D believes input likely to be real

**Problem formulation** in original GANs (Goodfellows et al. 2014):

$$\inf_\theta \sup_\varphi \left\{ \mathbb{E}^{x\sim\mu}[\ln f_\varphi(x)] + \mathbb{E}^{z\sim\zeta}[\ln(1 - f_\varphi(g_\theta(z)))] \right\}$$

**D:** learn $f_\varphi$ (via NN) s.t. $f_\varphi(\text{real}) \sim 1$, $f_\varphi(\text{fake}) \sim 0$

**G:** learn decoding map $g_\theta$ (via NN) to maximally confuse D

# Generative Adversarial Networks

**Why not Maximum Likelihood Estimation?**

- Density fitting: $d\nu_\theta(x) = p_\theta(x)dx$
- MLE: $\sup_\theta \frac{1}{N}\sum_{i=1}^N \ln p_\theta(x^i) \;\longleftrightarrow\; \inf_\theta H(\mu|\nu_\theta)$, where $H(.|.)$ relative entropy
- But $\nu_\theta$ has no density in $\mathcal{X}$, supports of $\nu_\theta$ and $\mu$ may even be non-overlapping (MLE not well defined)

$\Rightarrow$ look for a more flexible discrepancy to compare $\mu$ and $\nu_\theta$. Use a metric that can handle measures with non-overlapping supports.

# Generative Adversarial Networks: moving on

**Problems** (with original GANs):

- Continuity w.r.t. parameters
- Convergence
- Stability

# Generative Adversarial Networks: moving on

**Problems** (with original GANs):

- Continuity w.r.t. parameters
- Convergence
- Stability

**Some ways out:**

- Gradient-based regularizations
- D calculates some other divergence between $\mu$ and $\nu$: Integral Probability Metrics, Maximum Mean Discrepancy, Wasserstein distance, energy distance

# Generative Adversarial Networks: moving on

**Problems** (with original GANs):

- Continuity w.r.t. parameters
- Convergence
- Stability

**Some ways out:**

- Gradient-based regularizations
- D calculates some other divergence between $\mu$ and $\nu$: Integral Probability Metrics, Maximum Mean Discrepancy, Wasserstein distance, energy distance

Example: Wasserstein distance $\mathcal{W}_1(\mu, \nu_\theta) = \inf\{\mathbb{E}^\pi[\|x - y\|] : \pi_1 = \mu, \pi_2 = \nu_\theta\}$

$$\Longrightarrow \qquad \underbrace{\inf_\theta}_{G} \; \underbrace{\mathcal{W}_1(\mu, \nu_\theta)}_{D}$$

# Wasserstein GANs (Arjovsky et al., Gulrajani et al. 2017)

Dual formulation of the Wasserstein distance:

$$\mathcal{W}_1(\mu, \nu_\theta) = \sup_{f \ \mathrm{Lip}_1} \{\mathbb{E}^{\mu}[f] - \mathbb{E}^{\nu_\theta}[f]\}$$

# Wasserstein GANs (Arjovsky et al., Gulrajani et al. 2017)

Dual formulation of the Wasserstein distance:

$$\mathcal{W}_1(\mu, \nu_\theta) = \sup_{f \ \text{Lip}_1} \{\mathbb{E}^{\mu}[f] - \mathbb{E}^{\nu_\theta}[f]\}$$

$\rightarrow$ enforce Lip constraint via gradient penalization (easier and regularized)

$$\inf_{\theta} \sup_{\varphi} \left\{\mathbb{E}^{\mu}[f_\varphi(x)] - \mathbb{E}^{\nu_\theta}[f_\varphi(y)] \ + \text{Lip. penalization}\right\}$$

# Wasserstein GANs (Arjovsky et al., Gulrajani et al. 2017)

Dual formulation of the Wasserstein distance:

$$\mathcal{W}_1(\mu, \nu_\theta) = \sup_{f \ \text{Lip}_1} \{\mathbb{E}^{\mu}[f] - \mathbb{E}^{\nu_\theta}[f]\}$$

$\rightarrow$ enforce Lip constraint via gradient penalization (easier and regularized)

$$\inf_{\theta} \sup_{\varphi} \left\{\mathbb{E}^{\mu}[f_\varphi(x)] - \mathbb{E}^{\nu_\theta}[f_\varphi(y)] \ + \text{Lip. penalization}\right\}$$

- Continuity: if $\theta \mapsto g_\theta$ cont. $\Rightarrow$ $\theta \mapsto \mathcal{W}_1(\mu, \nu_\theta)$ cont.
- Convergence: WGANs converge if D always trained till optimality
- WGANs outperform MLE and MLE-NN unless exact parametric form of data is known

# Sinkhorn Divergences (Genevay, Peyré and Cuturi 2017)

$\rightarrow$ Genevay et al. suggest to consider the primal formulation: numerically more stable (in the dual, gradient requires differentiating the potential:difficult to compute, unstable)

# Sinkhorn Divergences (Genevay, Peyré and Cuturi 2017)

$\rightarrow$ Genevay et al. suggest to consider the primal formulation: numerically more stable (in the dual, gradient requires differentiating the potential:difficult to compute, unstable)

**GANs**: continuity w.r.t. $g$ $\times$, convergence $\times$, stability $\times$

**WGANs** (Optimal Transport):

$\hookrightarrow$ dual OT (Arjovsky et al. 2017, Gulrajani et al. 2017)
continuity $\checkmark$, convergence $\checkmark$, stability $\times$

$\hookrightarrow$ primal OT (Genevay, Peyré, Cuturi 2017)
continuity $\checkmark$, convergence $\checkmark$, stability $\checkmark$

# Sinkhorn Divergences (Genevay, Peyré and Cuturi 2017)

$\rightarrow$ Genevay et al. suggest to consider the primal formulation: numerically more stable (in the dual, gradient requires differentiating the potential:difficult to compute, unstable)

**GANs**: continuity w.r.t. $g$ $\times$, convergence $\times$, stability $\times$

**WGANs** (Optimal Transport):

$\hookrightarrow$ dual OT (Arjovsky et al. 2017, Gulrajani et al. 2017)
continuity $\checkmark$, convergence $\checkmark$, stability $\times$

$\hookrightarrow$ primal OT (Genevay, Peyré, Cuturi 2017)
continuity $\checkmark$, convergence $\checkmark$, stability $\checkmark$

$\rightarrow$ We consider a **dynamic framework**: to **generate paths** (data = time series)

- we mimic primal OT approach by Genevay et al.
- we need a **good distance for sequential data**

# Outline

- Introduction to Generative Adversarial Networks (GANs)
- Our toolkit: Causal Optimal Transport (COT)
- Dynamic GANs via COT
- Applications

# Classical Monge-Kantorovich Optimal Transport

Given Polish probability spaces $(\mathcal{X}, \mu), (\mathcal{Y}, \nu)$, move the mass from $\mu$ to $\nu$ minimizing the cost of transportation $c : \mathcal{X} \times \mathcal{Y} \to [0, \infty]$:

$$\mathrm{OT}(\mu, \nu, c) := \inf \left\{ \mathbb{E}^{\pi}[c(x, y)] : \pi \in \Pi(\mu, \nu) \right\},$$

where $\Pi(\mu, \nu)$ probability measures on $\mathcal{X} \times \mathcal{Y}$ with marginals $\mu$ and $\nu$

# Classical Monge-Kantorovich Optimal Transport

Given Polish probability spaces $(\mathcal{X}, \mu), (\mathcal{Y}, \nu)$, move the mass from $\mu$ to $\nu$ minimizing the cost of transportation $c : \mathcal{X} \times \mathcal{Y} \to [0, \infty]$:

$$\mathrm{OT}(\mu, \nu, c) := \inf \left\{ \mathbb{E}^{\pi}[c(x, y)] : \pi \in \Pi(\mu, \nu) \right\},$$

where $\Pi(\mu, \nu)$ probability measures on $\mathcal{X} \times \mathcal{Y}$ with marginals $\mu$ and $\nu$

e.g. $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$, $c(x, y) = \|x - y\|$ $\to$ Wasserstein distance

# Classical Monge-Kantorovich Optimal Transport

Given Polish probability spaces $(\mathcal{X}, \mu), (\mathcal{Y}, \nu)$, move the mass from $\mu$ to $\nu$ minimizing the cost of transportation $c : \mathcal{X} \times \mathcal{Y} \to [0, \infty]$:

$$\mathrm{OT}(\mu, \nu, c) := \inf \left\{ \mathbb{E}^{\pi}[c(x, y)] : \pi \in \Pi(\mu, \nu) \right\},$$

where $\Pi(\mu, \nu)$ probability measures on $\mathcal{X} \times \mathcal{Y}$ with marginals $\mu$ and $\nu$

e.g. $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$, $c(x, y) = \|x - y\|$ $\to$ Wasserstein distance

$\to$ **Dynamic framework** (e.g. $\mathcal{X} = \mathcal{Y} = \mathbb{R}^{d \times T}$) something that evolves in time:
"move distribution of a process $X$ into distribution of a process $Y$"

# Causal Optimal Transport

→ What is a **good distance in a dynamic framework**?

# Causal Optimal Transport

→ What is a **good distance in a dynamic framework**?

→ Idea: move the mass in a non-anticipative way ($Y$ is $X$−adapted, modulo external randomization)

# Causal Optimal Transport

→ What is a **good distance in a dynamic framework**?

→ Idea: move the mass in a non-anticipative way ($Y$ is $X$−adapted, modulo external randomization)

→ Mathematically: $\pi \in \mathcal{P}(\mathbb{R}^{d\times T} \times \mathbb{R}^{d\times T})$ is causal if

$$\pi(dy_t | dx_1, \cdots, dx_T) = \pi(dy_t | dx_1, \cdots, dx_t) \quad \forall t$$

# Causal Optimal Transport

→ What is a **good distance in a dynamic framework**?

→ Idea: move the mass in a non-anticipative way ($Y$ is $X$−adapted, modulo external randomization)

→ Mathematically: $\pi \in \mathcal{P}(\mathbb{R}^{d\times T} \times \mathbb{R}^{d\times T})$ is causal if

$$\pi(dy_t|dx_1, \cdots, dx_T) = \pi(dy_t|dx_1, \cdots, dx_t) \quad \forall t$$

**Causal Optimal Transport problem:**

$$\mathrm{COT}(\mu, \nu, c) := \inf \left\{ \mathbb{E}^{\pi}[c(X, Y)] : \pi \in \Pi^{\text{causal}}(\mu, \nu) \right\},$$

where $\Pi^{\text{causal}}(\mu, \nu) = \{\pi \in \Pi(\mu, \nu) : \pi \text{ causal}\}$

# Outline

- Introduction to Generative Adversarial Networks (GANs)
- Our toolkit: Causal Optimal Transport (COT)
- Dynamic GANs via COT
- Applications

# Dynamic Generative Adversarial Networks via COT

**Our approach:**

- We want to train a generator to produce sequential data
- We build some modification of GAN where D computes distance between the real distribution $\mu$ and the generated distribution $\nu_\theta$ via causal optimal transport

# Dynamic Generative Adversarial Networks via COT

**Our approach:**

- We want to train a generator to produce sequential data
- We build some modification of GAN where D computes distance between the real distribution $\mu$ and the generated distribution $\nu_\theta$ via causal optimal transport
- Inspired by Genevay, Peyré and Cuturi, we consider the primal COT problem and add an entropic regularization $\rightarrow$ Sinkhorn algorithm
- Causality provides a family of cost functions $\rightarrow$ D computes the worst case OT distance w.r.t. these cost functions (D is learning the cost function)

# Dynamic Generative Adversarial Networks via COT

1. We regularize the causal transport:

$$\mathrm{COT}^{\varepsilon}(\mu, \nu, c) := \inf_{\pi \in \Pi^{\text{causal}}(\mu,\nu)} \left\{ \mathbb{E}^{\pi}[c(x, y)] + \varepsilon H(\pi | \mu \otimes \nu) \right\},$$

$$\mathrm{COT}^{\varepsilon}(\mu, \nu, c) \xrightarrow[\varepsilon \to 0]{} \mathrm{COT}(\mu, \nu, c) \quad \text{(A., Backhoff, Jia 2020)}$$

# Dynamic Generative Adversarial Networks via COT

1. We regularize the causal transport:

$$\mathrm{COT}^{\varepsilon}(\mu,\nu,c) := \inf_{\pi\in\Pi^{\text{causal}}(\mu,\nu)} \left\{\mathbb{E}^{\pi}[c(x,y)] + \varepsilon H(\pi|\mu\otimes\nu)\right\},$$

$$\mathrm{COT}^{\varepsilon}(\mu,\nu,c) \xrightarrow[\varepsilon\to 0]{} \mathrm{COT}(\mu,\nu,c) \quad \text{(A., Backhoff, Jia 2020)}$$

2. We dualize the causality constraint, and obtain:

$$\mathrm{COT}^{\varepsilon}(\mu,\nu,c) = \sup_{s\in\mathbb{S}} \mathrm{OT}^{\varepsilon}(\mu,\nu,c+s)$$

# Dynamic Generative Adversarial Networks via COT

1. We regularize the causal transport:

$$\mathrm{COT}^{\varepsilon}(\mu,\nu,c) := \inf_{\pi\in\Pi^{\text{causal}}(\mu,\nu)} \left\{\mathbb{E}^{\pi}[c(x,y)] + \varepsilon H(\pi|\mu\otimes\nu)\right\},$$

$$\mathrm{COT}^{\varepsilon}(\mu,\nu,c) \xrightarrow[\varepsilon\to 0]{} \mathrm{COT}(\mu,\nu,c) \quad \text{(A., Backhoff, Jia 2020)}$$

2. We dualize the causality constraint, and obtain:

$$\mathrm{COT}^{\varepsilon}(\mu,\nu,c) = \sup_{s\in\mathbb{S}} \mathrm{OT}^{\varepsilon}(\mu,\nu,c+s)$$

3. We remove the bias $\rightarrow$ Sinkhorn divergence:

$$\mathcal{W}_{c+s,\varepsilon}(\mu,\nu) := \mathrm{OT}^{\varepsilon}(\mu,\nu,c+s) - \tfrac{1}{2}\mathrm{OT}^{\varepsilon}(\mu,\mu,c+s) - \tfrac{1}{2}\mathrm{OT}^{\varepsilon}(\nu,\nu,c+s)$$

# Dynamic Generative Adversarial Networks via COT

**Causal Wasserstein GAN**:

$$\boxed{\inf_{\theta} \sup_{\varphi} \mathcal{W}_{c_\varphi,\varepsilon}(\mu, \nu_\theta)}$$

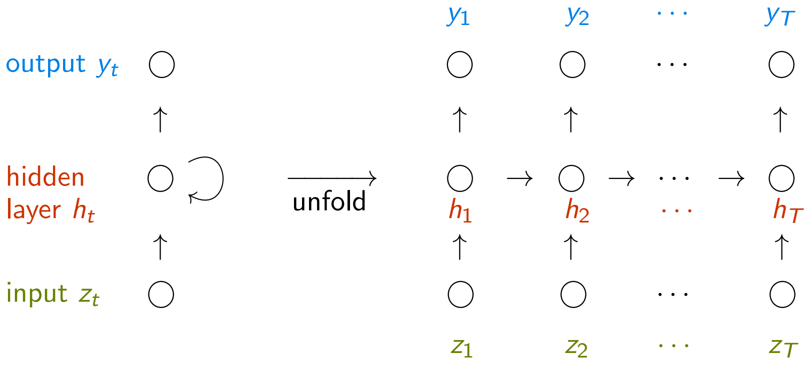
$\rightarrow$ D learns $c_\varphi$, that is, the cost function (worst-case distance)

$\rightarrow$ G learns $\nu_\theta$, that is, the best generating function $g_\theta$ ($\nu_\theta = g_{\theta\#}\zeta$)

- the cost functions $c_\varphi$ are of the form appearing in the dualization of causality
- $\varphi$ ad $\theta$ learned through "dynamic architectures", such as Recurrent Neural Networks and Convolutionals Neural Networks

# Training architecture: example

## Basic Recurrent Neural Network




$h_t = \sigma(Az_t + Bh_{t-1} + a)$ network memory, $\sigma$ activation functions

$y_t = Cs_t$, $\theta = \{A, B, C, a\}$ parameters: weight matrices and bias vectors

# The algorithm

To solve the min-max problem, we approximate $\mathcal{W}_{c_\varphi,\epsilon}(\mu,\nu_\theta)$:

- sample mini-batches from real data and from latent space $\hookrightarrow$ emp.distr. $\hat{\mu},\hat{\nu}_\theta$
- penalize cost functions $c_\varphi = c + s$ for which $s \notin \mathbb{S}$
- compute $\inf_{\pi\in\Pi(\hat{\mu},\hat{\nu}_\theta)} \left\{\mathbb{E}^\pi[c_\varphi] + \epsilon H(\pi|\hat{\mu}\otimes\hat{\nu}_\theta)\right\}$ by Sinkhorn algorithm (Cuturi 2013), by considering a pre-determined # iterations

$$\Rightarrow \quad \widehat{\mathcal{W}}_{c_\varphi,\epsilon}(\hat{\mu},\hat{\nu}_\theta)$$

Use stochastic Gradient Ascent/Descent to update parameters:

$$\varphi_{n+1} = \varphi_n + \alpha\,\nabla_\varphi \widehat{\mathcal{W}}_{c_\varphi,\epsilon}(\hat{\mu},\hat{\nu}_\theta)$$

$$\theta_{n+1} = \theta_n - \alpha\,\nabla_\theta \widehat{\mathcal{W}}_{c_\varphi,\epsilon}(\hat{\mu},\hat{\nu}_\theta)$$

# Pseudo-code

**Data:** $\theta_0$, $\varphi_0$, $\{x^i\}_{i=1}^N$, $\zeta$, $\epsilon$, batch size $m$, Sinkhorn iter., learning rate $\alpha$, critic iter. $n_c$
**Result:** $\theta$, $\varphi$
$\theta \leftarrow \theta_0$, $\varphi \leftarrow \varphi_0$
**for** $k = 1, 2, \dots$ **do**
  **for** $l = 1, 2, \dots, n_c$ **do**
    Sample: $\{x^i\}_{i=1}^m$ from real data, and $\{z^i\}_{i=1}^m$ from $\zeta$
    $y^i \leftarrow g_\theta(z^i)$
    $\varphi \leftarrow \varphi + \alpha \nabla_\varphi \left( \widehat{\mathcal{W}}_{c_\varphi, \epsilon}(\hat{\mu}, \hat{\nu}_\theta) \right)$
  **end**
  Sample: $\{x^i\}_{i=1}^m$ from real data, and $\{z^i\}_{i=1}^m$ from $\zeta$
  $y^i \leftarrow g_\theta(z^i)$
  $\theta \leftarrow \theta - \alpha \nabla_\theta \left( \widehat{\mathcal{W}}_{c_\varphi, \epsilon}(\hat{\mu}, \hat{\nu}_\theta) \right)$
**end**

# Prediction rather than generation

→ Causal Wasserstein GANs: learn how to **generate real-looking evolutions** given an observed dataset.

→ WIP: develop a conditional modification of the algorithm, for **time-series trend prediction**, so that we feed the beginning of a sequence and the generator produces some reasonable continuation.

- Mathematically: easy modification
- But may require different choice of architectures

# Outline

- Introduction to Generative Adversarial Networks (GANs)
- Our toolkit: Causal Optimal Transport (COT)
- Dynamic GANs via COT
- Applications

# Applications

**Initial testing:**

- We have been testing some easy-to check features on simulated data, e.g. reproducing periodic curves.
- Now we are testing on standard datasets, such as audio datasets (NSynth); and MNIST (yes, I know, not truly sequential..)

# Applications

**Initial testing:**

- We have been testing some easy-to check features on simulated data, e.g. reproducing periodic curves.
- Now we are testing on standard datasets, such as audio datasets (NSynth); and MNIST (yes, I know, not truly sequential..)

**Financial applications: data-driven model-independent analysis**

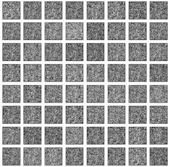
- Robust pricing of financial derivatives
- Volatility prediction
- Prediction of evolution of LOB

# Applications

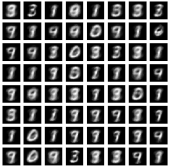
As we are still working on the audio / financial applications - and since I cannot finish such a talk without showing something we generated...

# Applications

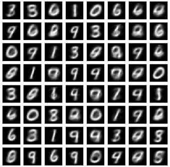
As we are still working on the audio / financial applications - and since I cannot finish such a talk without showing something we generated...



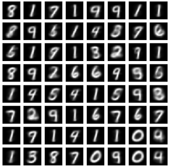
1 iteration (1.1 sec)



300 iterations (4.5 min)



15k iterations (5h)



40k iterations (14h)

MNIST: batch size 32, critic 1, $\epsilon = 0.8$, Sinkhorn iter. 30, learning rate 0.0001

# Literature


- Acciaio, Backhoff, Jia: Cournot-Nash equilibrium and optimal transport in a dynamic setting, 2020
- Arjovsky, Chintala, Bottou: Wasserstein GAN, 2017
- Cuturi: Sinkhorn distances: Lightspeed computation of OT, 2013
- Genevay, Peyré, Cuturi: Learning Generative Models with Sinkhorn Divergences, 2017
- Goodfellows et al.: Generative Adversarial Networks, 2014
- Gulrajani et al.: Improved Training of Wasserstein GANs, 2017

**Thank you for your attention!**