

# Deep Generative Models

## Lecture 10

Roman Isachenko



Ozon Masters

Spring, 2021

## Recap of previous lecture

### Likelihood-free learning

- ▶ Likelihood is not a perfect measure quality measure for generative model.
- ▶ Likelihood could be intractable.

Imagine we have two sets of samples

- ▶  $\mathcal{S}_1 = \{\mathbf{x}_i\}_{i=1}^{n_1} \sim \pi(\mathbf{x})$  – real samples;
- ▶  $\mathcal{S}_2 = \{\mathbf{x}_i\}_{i=1}^{n_2} \sim p(\mathbf{x}|\theta)$  – generated (or fake) samples.

### Two sample test

$$H_0 : \pi(\mathbf{x}) = p(\mathbf{x}|\theta), \quad H_1 : \pi(\mathbf{x}) \neq p(\mathbf{x}|\theta)$$

If test statistic  $T(\mathcal{S}_1, \mathcal{S}_2) < \alpha$ , then accept  $H_0$ , else reject it.

- ▶  $p(\mathbf{x}|\theta)$  minimizes the value of test statistic  $T(\mathcal{S}_1, \mathcal{S}_2)$ .
- ▶ It is hard to find an appropriate test statistic in high dimensions.  $T(\mathcal{S}_1, \mathcal{S}_2)$  could be learnable.

## Recap of previous lecture

- ▶ **Generator:** generative model  $\mathbf{x} = G(\mathbf{z})$ , which makes generated sample more realistic.
- ▶ **Discriminator:** a classifier  $D(\mathbf{x}) \in [0, 1]$ , which distinguishes real samples from generated samples.

### GAN optimality theorem

The minimax game

$$\min_G \max_D V(G, D) = \min_G \max_D [\mathbb{E}_{\pi(x)} \log D(\mathbf{x}) + \mathbb{E}_{p(z)} \log(1 - D(G(\mathbf{z})))]$$

has the global optimum  $\pi(\mathbf{x}) = p(\mathbf{x}|\theta)$ , in this case  $D^*(\mathbf{x}) = 0.5$ .

$$\min_G V(G, D^*) = \min_G [2JSD(\pi || p) - \log 4] = -\log 4, \quad \pi(\mathbf{x}) = p(\mathbf{x}|\theta).$$

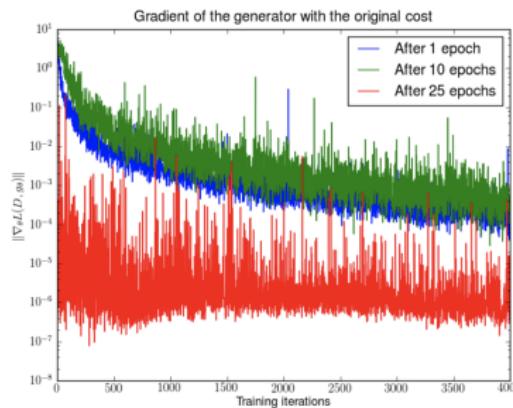
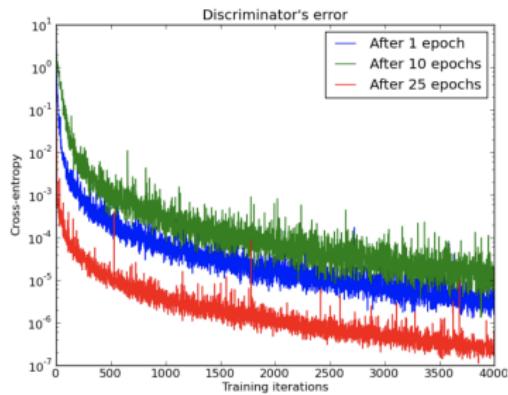
If the generator could be any function and the discriminator is optimal at every step, then the generator is guaranteed to converge to the data distribution.

# Vanishing gradients

## Objective

$$\min_G \max_D V(G, D) = \min_G \max_D [\mathbb{E}_{\pi(x)} \log D(x) + \mathbb{E}_{p(z)} \log(1 - D(G(z)))]$$

Early in learning,  $G$  is poor,  $D$  can reject samples with high confidence. In this case,  $\log(1 - D(G(z)))$  saturates.



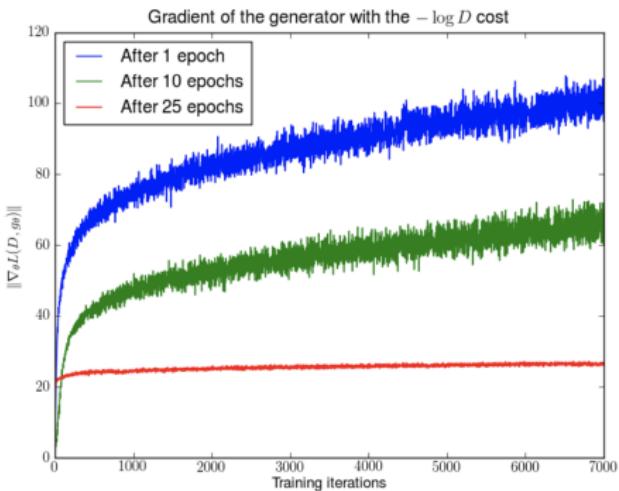
# Vanishing gradients

## Objective

$$\min_G \max_D V(G, D) = \min_G \max_D [\mathbb{E}_{\pi(x)} \log D(x) + \mathbb{E}_{p(z)} \log(1 - D(G(z)))]$$

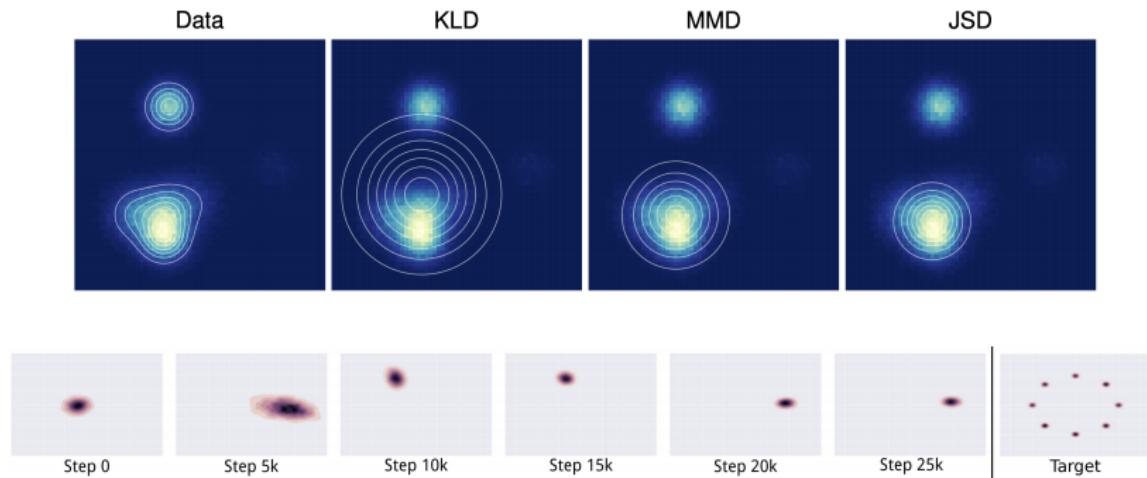
## Non-saturating GAN

- ▶ Maximize  $\log D(G(z))$  instead of  $\log(1 - D(G(z)))$ .
- ▶ Gradients are getting much stronger, but the training is unstable (with increasing mean and variance).



## Mode collapse

The phenomena where the generator of a GAN collapses to one or few distribution modes.



Alternate architectures, adding regularization terms, injecting small noise perturbations and other millions bags and tricks are used to avoid the mode collapse.

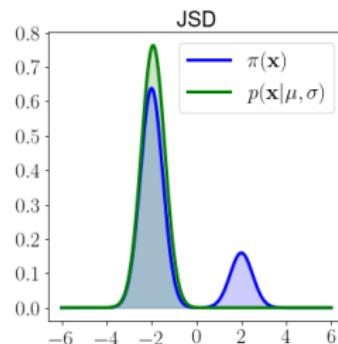
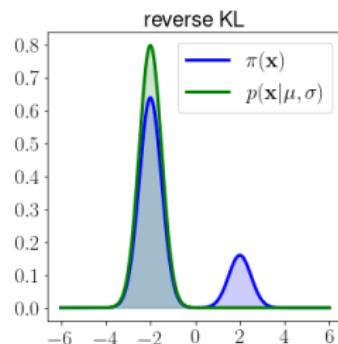
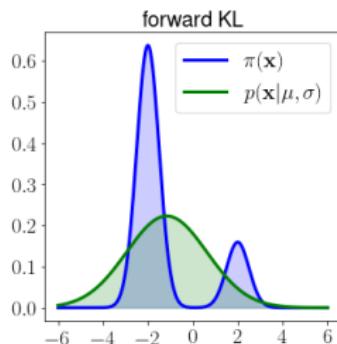
Goodfellow I. J. et al. *Generative Adversarial Networks*, 2014  
Metz L. et al. *Unrolled Generative Adversarial Networks*, 2016

# Jensen-Shannon vs Kullback-Leibler

Mode covering vs mode seeking

$$KL(\pi||p) = \int \pi(\mathbf{x}) \log \frac{\pi(\mathbf{x})}{p(\mathbf{x})} d\mathbf{x}, \quad KL(p||\pi) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{\pi(\mathbf{x})} d\mathbf{x}$$

$$JSD(\pi||p) = \frac{1}{2} \left[ KL \left( \pi(\mathbf{x}) || \frac{\pi(\mathbf{x}) + p(\mathbf{x})}{2} \right) + KL \left( p(\mathbf{x}) || \frac{\pi(\mathbf{x}) + p(\mathbf{x})}{2} \right) \right]$$

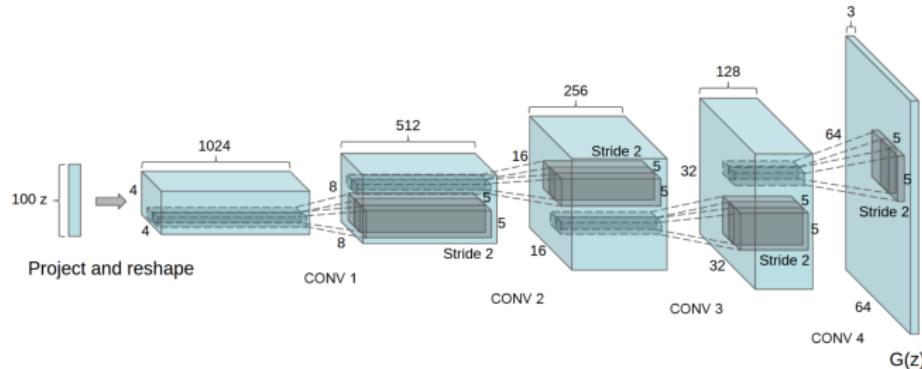


# Vanilla GAN results



# Deep Convolutional GAN

## Architecture

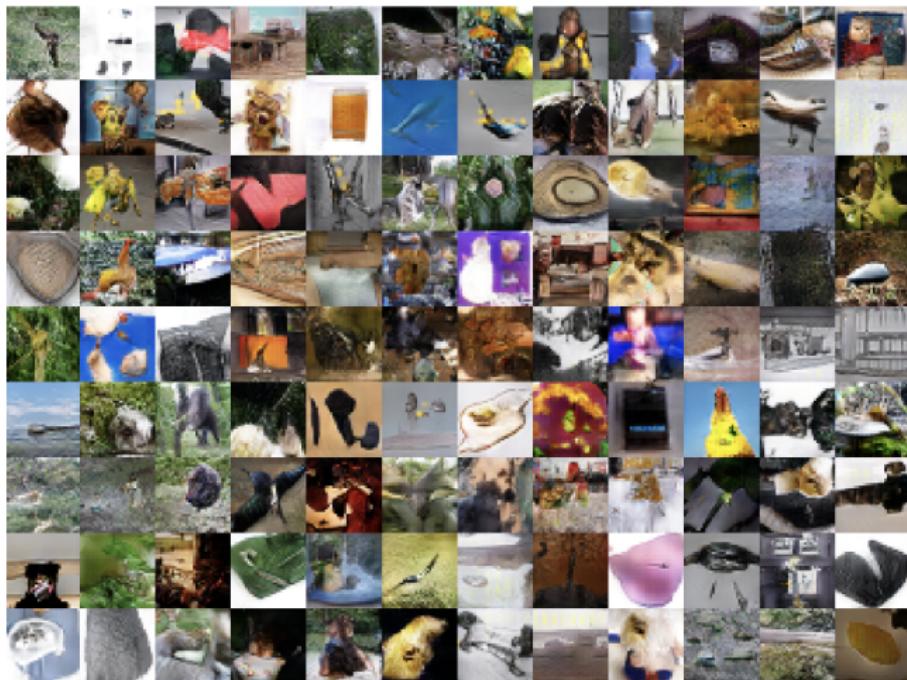


- ▶ Mean-pooling instead of max-pooling.
- ▶ Transposed convolutions in the generator for upsampling.
- ▶ Downsample with strided convolutions and average pooling.
- ▶ ReLU for generator, Leaky-ReLU (0.2) for discriminator.
- ▶ Output nonlinearity: tanh for Generator, sigmoid for discriminator.
- ▶ Batch Normalization used to prevent mode collapse (not applied at the output of  $G$  and input of  $D$ ).
- ▶ Adam: small LR = 2e-4; small momentum: 0.5, batch-size: 128.

*Radford A., Metz L., Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2015*

# Deep Convolutional GAN

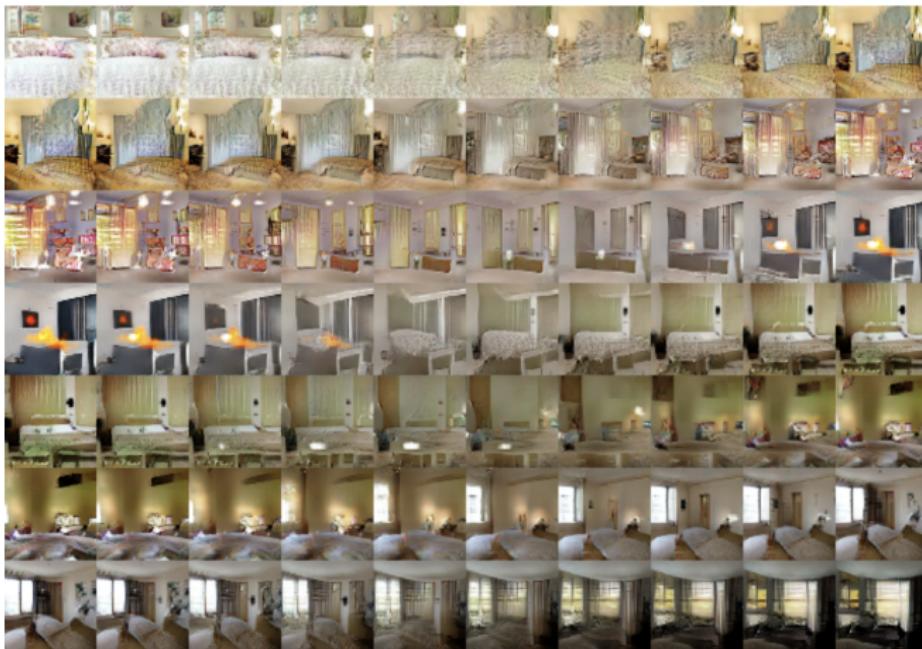
## ImageNet samples



*Radford A., Metz L., Chintala S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2015*

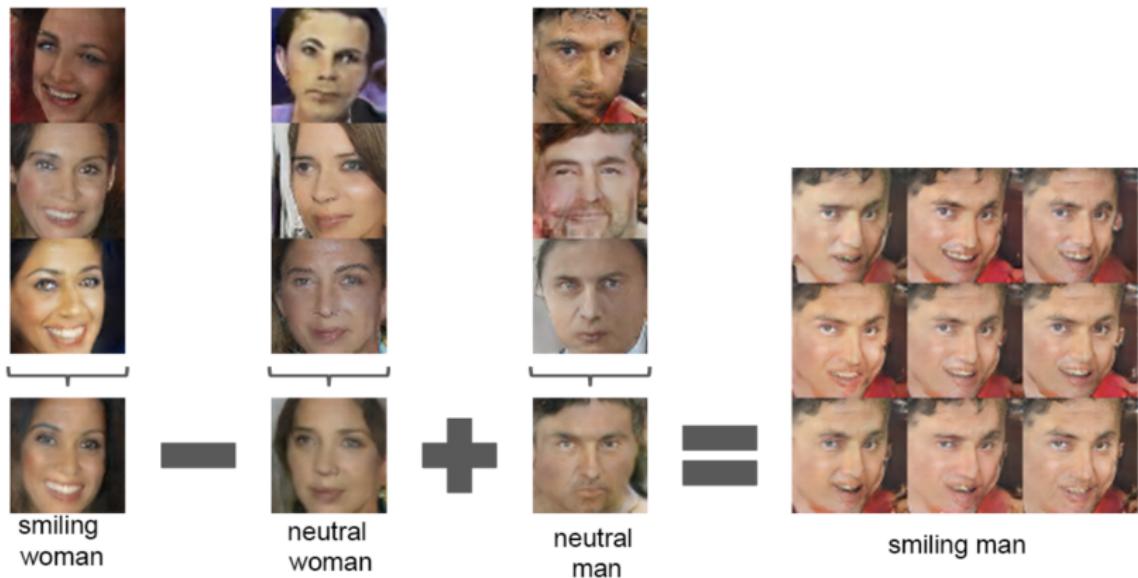
# Deep Convolutional GAN

## Smooth interpolations



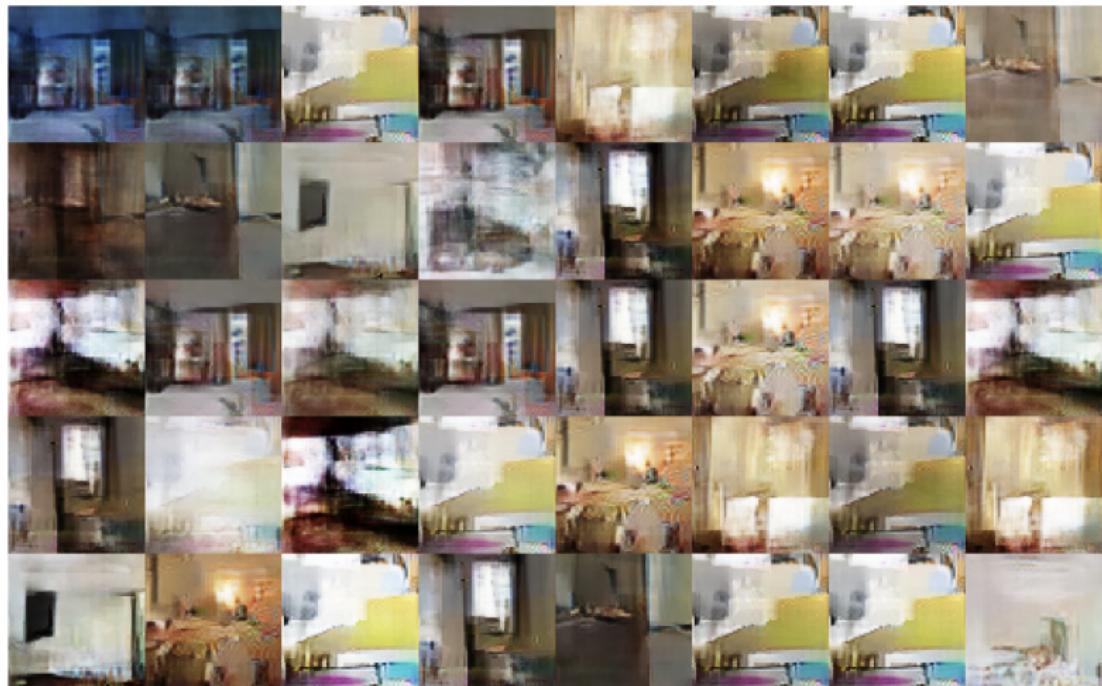
# Deep Convolutional GAN

## Vector arithmetic



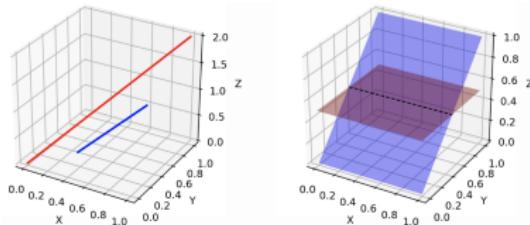
# Deep Convolutional GAN

## Mode collapse



# Informal theoretical results

- ▶ Since  $z$  usually has lower dimensionality compared to  $x$ , manifold  $G(z)$  has a measure 0 in  $x$  space. Hence, support of  $p(x|\theta)$  lies on low-dimensional manifold.
- ▶ Distribution of real images  $\pi(x)$  is also concentrated on a low dimensional manifold.



- ▶ If  $\pi(x)$  and  $p(x|\theta)$  have disjoint supports, then there is a smooth optimal discriminator. We are not able to learn anything by backproping through it.
- ▶ For such low-dimensional disjoint manifolds

$$KL(\pi||p) = KL(p||\pi) = \infty, \quad JSD(\pi||p) = \log 2$$

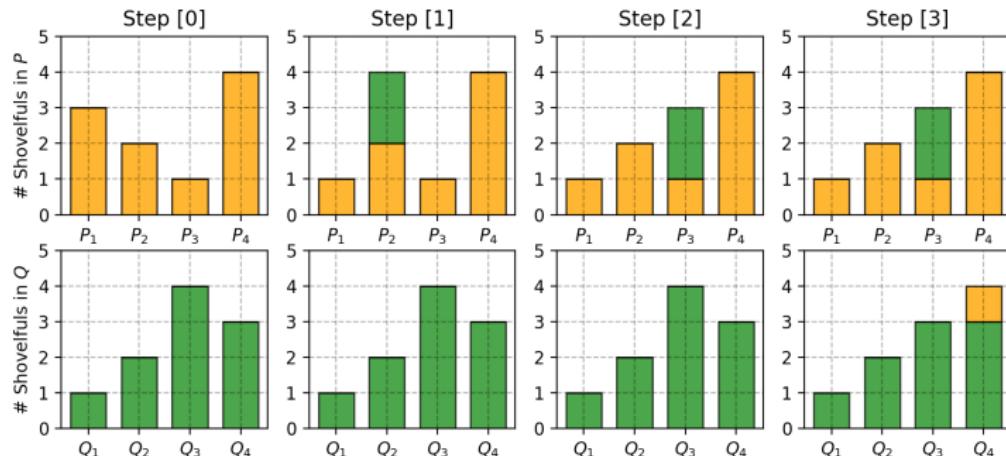
- ▶ Adding continuous noise to the inputs of the discriminator smoothes the distributions of the probability mass.

Weng L. From GAN to WGAN, 2019

Arjovsky M., Bottou L. Towards Principled Methods for Training Generative Adversarial Networks, 2017

# Wasserstein distance (discrete)

Also called Earth Mover's distance. The minimum cost of moving and transforming a pile of dirt in the shape of one probability distribution to the shape of the other distribution.



$$W(P, Q) = 2(\text{step 1}) + 2(\text{step 2}) + 1(\text{step 3}) = 5$$

# Wasserstein distance

$$W(\pi, p) = \inf_{\gamma \in \prod(\pi, p)} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \gamma} \|\mathbf{x} - \mathbf{y}\| = \inf_{\gamma \in \prod(\pi, p)} \int \|\mathbf{x} - \mathbf{y}\| \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}$$

- ▶  $\prod(\pi, p)$  – the set of all joint distributions  $\gamma(\mathbf{x}, \mathbf{y})$  with marginals  $\pi$  and  $p$  ( $\int \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} = p(\mathbf{y})$ ,  $\int \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{y} = \pi(\mathbf{x})$ )
- ▶  $\gamma(\mathbf{x}, \mathbf{y})$  – transportation plan (the amount of "dirt" that should be transported from point  $\mathbf{x}$  to point  $\mathbf{y}$ ).
- ▶  $\gamma(\mathbf{x}, \mathbf{y})$  – the amount,  $\|\mathbf{x} - \mathbf{y}\|$  – the distance.

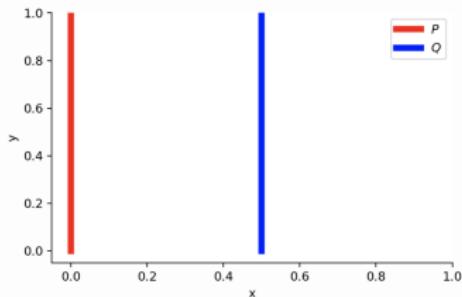
For better understanding of transportation plan function  $\gamma$ , try to write down the plan for previous discrete case.

# Wasserstein distance vs KL vs JSD

Consider 2d distributions

$$\pi(x, y) = (0, U[0, 1])$$

$$p(x, y|\theta) = (\theta, U[0, 1])$$



- $\theta = 0$ . Distributions are the same

$$KL(\pi||p) = KL(p||\pi) = JSD(p||\pi) = W(\pi, p) = 0$$

- $\theta \neq 0$

$$KL(\pi||p) = \int_{U[0,1]} 1 \log \frac{1}{0} dy = \infty = KL(p||\pi)$$

$$JSD(\pi||p) = \frac{1}{2} \left( \int_{U[0,1]} 1 \log \frac{1}{1/2} dy + \int_{U[0,1]} 1 \log \frac{1}{1/2} dy \right) = \log 2$$

$$W(\pi, p) = |\theta|$$

# Wasserstein distance vs KL vs JSD

## Theorem 1

Let  $G(\mathbf{z}, \theta)$  be (almost) any feedforward neural network, and  $p(\mathbf{z})$  a prior over  $\mathbf{z}$  such that  $\mathbb{E}_{p(\mathbf{z})} \|\mathbf{z}\| < \infty$ . Then therefore  $W(\pi, p)$  is continuous everywhere and differentiable almost everywhere.

## Theorem 2

Let  $\pi$  be a distribution on a compact space  $\mathcal{X}$  and  $\{p_t\}_{t=1}^{\infty}$  be a sequence of distributions on  $\mathcal{X}$ .

$$KL(\pi || p_t) \rightarrow 0 \text{ (or } KL(p_t || \pi) \rightarrow 0) \quad (1)$$

$$JSD(\pi || p_t) \rightarrow 0 \quad (2)$$

$$W(\pi || p_t) \rightarrow 0 \quad (3)$$

Then, considering limits as  $t \rightarrow \infty$ , (1) implies (2), (2) implies (3).

# Wasserstein GAN

## Wasserstein distance

$$W(\pi||p) = \inf_{\gamma \in \prod(\pi, p)} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \gamma} \|\mathbf{x} - \mathbf{y}\| = \inf_{\gamma \in \prod(\pi, p)} \int \|\mathbf{x} - \mathbf{y}\| \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}$$

The infimum across all possible joint distributions in  $\prod(\pi, p)$  is intractable.

## Kantorovich-Rubinstein duality

$$W(\pi||p) = \frac{1}{K} \max_{\|f\|_L \leq K} [\mathbb{E}_{\pi(\mathbf{x})} f(\mathbf{x}) - \mathbb{E}_{p(\mathbf{x})} f(\mathbf{x})],$$

where  $\|f\|_L \leq K$  are  $K$ -Lipschitz continuous functions  
 $(f : \mathcal{X} \rightarrow \mathbb{R})$

$$|f(\mathbf{x}_1) - f(\mathbf{x}_2)| \leq K \|\mathbf{x}_1 - \mathbf{x}_2\|, \quad \text{for all } \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}.$$

# Wasserstein GAN

## Kantorovich-Rubinstein duality

$$W(\pi || p) = \frac{1}{K} \max_{\|f\|_L \leq K} [\mathbb{E}_{\pi(x)} f(x) - \mathbb{E}_{p(x)} f(x)],$$

- ▶ Now we have to ensure that  $f$  is  $K$ -Lipschitz continuous.
- ▶ Let  $f(x, \phi)$  be a feedforward neural network parametrized by  $\phi$ .
- ▶ If parameters  $\phi$  lie in a compact set  $\Phi$  then  $f(x, \phi)$  will be  $K$ -Lipschitz continuous function.
- ▶ Let the parameters be clamped to a fixed box  $\Phi \in [-0.01, 0.01]^d$  after each gradient update.

$$\begin{aligned} K \cdot W(\pi || p) &= \max_{\|f\|_L \leq K} [\mathbb{E}_{\pi(x)} f(x) - \mathbb{E}_{p(x)} f(x)] \geq \\ &\geq \max_{\phi \in \Phi} [\mathbb{E}_{\pi(x)} f(x, \phi) - \mathbb{E}_{p(x)} f(x, \phi)] \end{aligned}$$

# Wasserstein GAN

## Vanilla GAN objective

$$\min_G \max_D \mathbb{E}_{\pi(x)} \log D(x) + \mathbb{E}_{p(z)} \log(1 - D(G(z)))$$

## WGAN objective

$$\min_G W(\pi || p) = \min_G \max_{\phi \in \Phi} [\mathbb{E}_{\pi(x)} f(x, \phi) - \mathbb{E}_{p(z)} f(G(z), \phi)].$$

- ▶ Discriminator  $D$  is similar to the function  $f$ , but not the same (it is not a classifier anymore). In the WGAN model, function  $f$  is usually called *critic*.
- ▶ "Weight clipping is a clearly terrible way to enforce a Lipschitz constraint". If the clipping parameter is large, it is hard to train the critic till optimality. If the clipping parameter is too small, it could lead to vanishing gradients.

# Wasserstein GAN

---

**Algorithm 1** WGAN, our proposed algorithm. All experiments in the paper used the default values  $\alpha = 0.00005$ ,  $c = 0.01$ ,  $m = 64$ ,  $n_{\text{critic}} = 5$ .

---

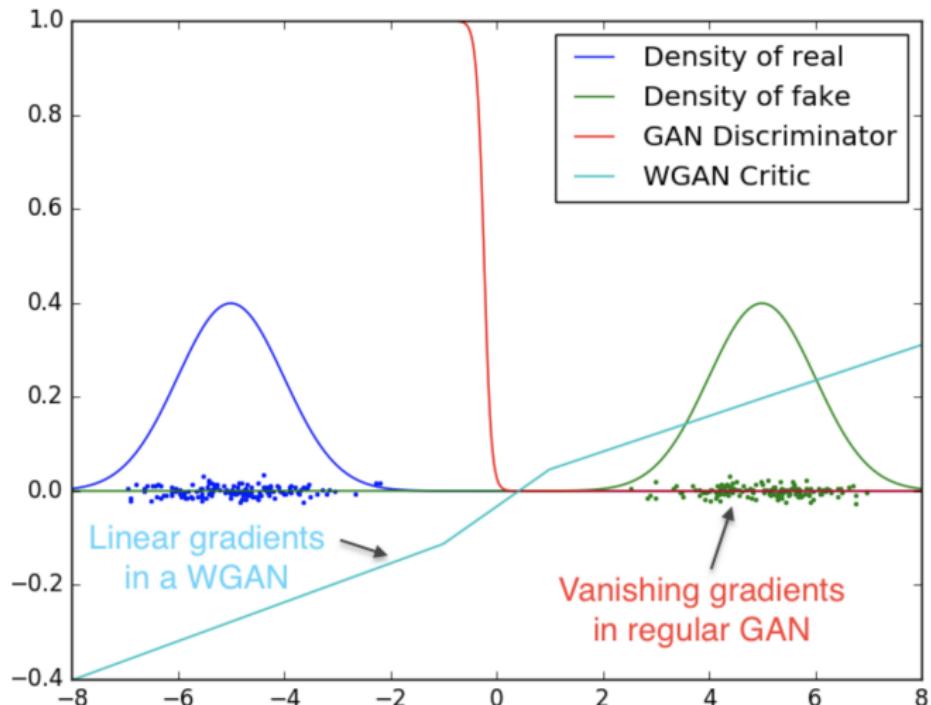
**Require:** :  $\alpha$ , the learning rate.  $c$ , the clipping parameter.  $m$ , the batch size.  
 $n_{\text{critic}}$ , the number of iterations of the critic per generator iteration.

**Require:** :  $w_0$ , initial critic parameters.  $\theta_0$ , initial generator's parameters.

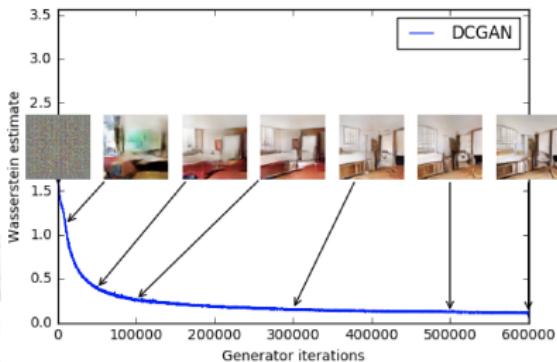
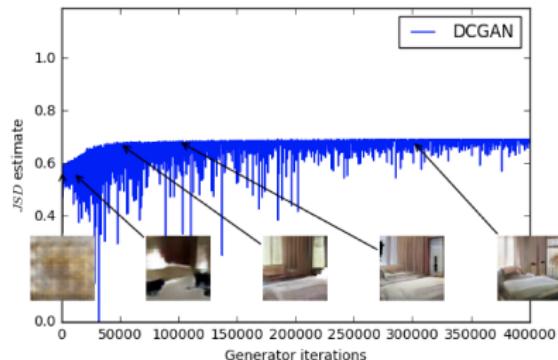
```
1: while  $\theta$  has not converged do
2:   for  $t = 0, \dots, n_{\text{critic}}$  do
3:     Sample  $\{x^{(i)}\}_{i=1}^m \sim \mathbb{P}_r$  a batch from the real data.
4:     Sample  $\{z^{(i)}\}_{i=1}^m \sim p(z)$  a batch of prior samples.
5:      $g_w \leftarrow \nabla_w \left[ \frac{1}{m} \sum_{i=1}^m f_w(x^{(i)}) - \frac{1}{m} \sum_{i=1}^m f_w(g_\theta(z^{(i)})) \right]$ 
6:      $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$ 
7:      $w \leftarrow \text{clip}(w, -c, c)$ 
8:   end for
9:   Sample  $\{z^{(i)}\}_{i=1}^m \sim p(z)$  a batch of prior samples.
10:   $g_\theta \leftarrow -\nabla_\theta \frac{1}{m} \sum_{i=1}^m f_w(g_\theta(z^{(i)}))$ 
11:   $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$ 
12: end while
```

---

# Wasserstein GAN



# Wasserstein GAN



- ▶  $JSD$  correlates poorly with the sample quality. Stays constant nearly maximum value  $\log 2 \approx 0.69$ .
- ▶  $W$  is highly correlated with the sample quality.

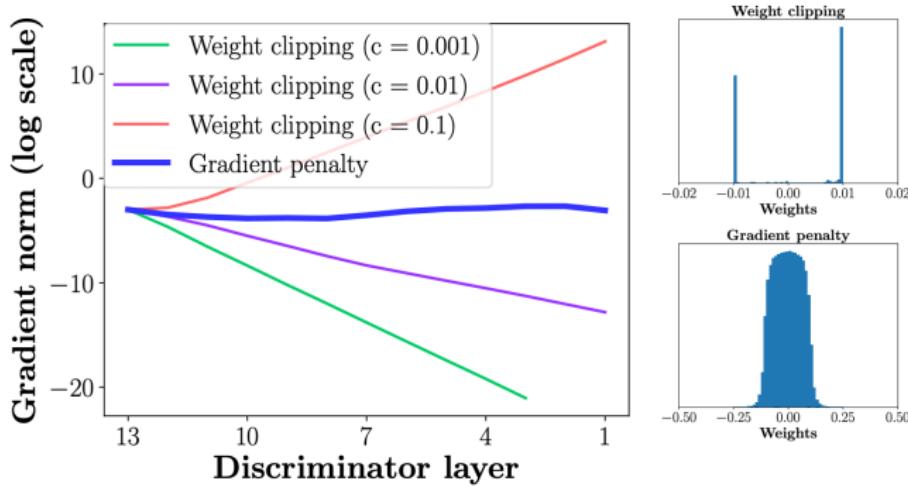


"In no experiment did we see evidence of mode collapse for the WGAN algorithm."

# Wasserstein GAN with Gradient Penalty

## Weight clipping analysis

- ▶ The critic ignores higher moments of the data distribution.
- ▶ The gradients either grow or decay exponentially.



Gradient penalty makes the gradients more stable.

# Wasserstein GAN with Gradient Penalty

## Theorem

Let  $\pi(\mathbf{x})$  and  $p(\mathbf{x})$  be two distribution in  $\mathcal{X}$ , a compact metric space. Then, there is 1-Lipschitz function  $f^*$  which is the optimal solution of

$$\max_{\|f\|_L \leq 1} [\mathbb{E}_{\pi(\mathbf{x})} f(\mathbf{x}) - \mathbb{E}_{p(\mathbf{x})} f(\mathbf{x})].$$

Let  $\gamma$  be the optimal transportation plan between  $\pi(\mathbf{x})$  and  $p(\mathbf{x})$ . Then, if  $f^*$  is differentiable,  $\gamma(\mathbf{x} = \mathbf{y}) = 0$  and  $\hat{\mathbf{x}}_t = t\mathbf{x} + (1 - t)\mathbf{y}$  with  $t \in [0, 1]$  it holds that

$$\mathbb{P}_{(\mathbf{x}, \mathbf{y}) \sim \gamma} \left[ \nabla f^*(\hat{\mathbf{x}}_t) = \frac{\mathbf{y} - \hat{\mathbf{x}}_t}{\|\mathbf{y} - \hat{\mathbf{x}}_t\|} \right] = 1.$$

## Corollary

$f^*$  has gradient norm 1 almost everywhere under  $\pi(\mathbf{x})$  and  $p(\mathbf{x})$ .

# Wasserstein GAN with Gradient Penalty

A differentiable function is 1-Lipschitz if and only if it has gradients with norm at most 1 everywhere.

## Gradient penalty

$$W(\pi || p) = \underbrace{\mathbb{E}_{\pi(x)} f(x) - \mathbb{E}_{p(x)} f(x)}_{\text{original critic loss}} + \lambda \underbrace{\mathbb{E}_{\hat{x}} \left[ (\|\nabla_{\hat{x}} f(\hat{x})\|_2 - 1)^2 \right]}_{\text{gradient penalty}},$$

- ▶ Samples  $\hat{x}$  are uniformly sampled along straight lines between pairs of points from the data distribution  $\pi(x)$  and the generator distribution  $p(x|\theta)$ .
- ▶ Enforcing the unit gradient norm constraint everywhere is intractable, it turns out to be sufficient to enforce it only along these straight lines.

# Wasserstein GAN with Gradient Penalty

---

**Algorithm 1** WGAN with gradient penalty. We use default values of  $\lambda = 10$ ,  $n_{\text{critic}} = 5$ ,  $\alpha = 0.0001$ ,  $\beta_1 = 0$ ,  $\beta_2 = 0.9$ .

---

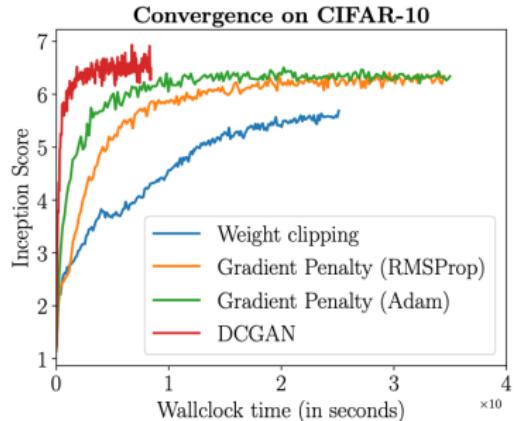
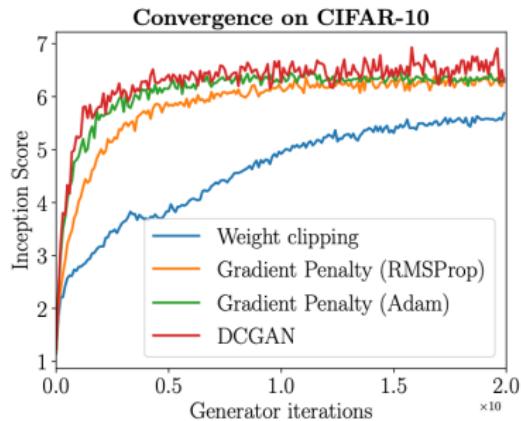
**Require:** The gradient penalty coefficient  $\lambda$ , the number of critic iterations per generator iteration  $n_{\text{critic}}$ , the batch size  $m$ , Adam hyperparameters  $\alpha, \beta_1, \beta_2$ .

**Require:** initial critic parameters  $w_0$ , initial generator parameters  $\theta_0$ .

```
1: while  $\theta$  has not converged do
2:   for  $t = 1, \dots, n_{\text{critic}}$  do
3:     for  $i = 1, \dots, m$  do
4:       Sample real data  $\mathbf{x} \sim \mathbb{P}_r$ , latent variable  $\mathbf{z} \sim p(\mathbf{z})$ , a random number  $\epsilon \sim U[0, 1]$ .
5:        $\tilde{\mathbf{x}} \leftarrow G_\theta(\mathbf{z})$ 
6:        $\hat{\mathbf{x}} \leftarrow \epsilon \mathbf{x} + (1 - \epsilon) \tilde{\mathbf{x}}$ 
7:        $L^{(i)} \leftarrow D_w(\tilde{\mathbf{x}}) - D_w(\mathbf{x}) + \lambda(\|\nabla_{\hat{\mathbf{x}}} D_w(\hat{\mathbf{x}})\|_2 - 1)^2$ 
8:     end for
9:      $w \leftarrow \text{Adam}(\nabla_w \frac{1}{m} \sum_{i=1}^m L^{(i)}, w, \alpha, \beta_1, \beta_2)$ 
10:   end for
11:   Sample a batch of latent variables  $\{\mathbf{z}^{(i)}\}_{i=1}^m \sim p(\mathbf{z})$ .
12:    $\theta \leftarrow \text{Adam}(\nabla_\theta \frac{1}{m} \sum_{i=1}^m -D_w(G_\theta(\mathbf{z})), \theta, \alpha, \beta_1, \beta_2)$ 
13: end while
```

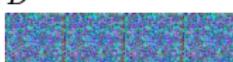
---

# Wasserstein GAN with Gradient Penalty



Min. score	Only GAN	Only WGAN-GP	Both succeeded	Both failed
1.0	0	8	192	0
3.0	1	88	110	1
5.0	0	147	42	11
7.0	1	104	5	90
9.0	0	0	0	200

# Wasserstein GAN with Gradient Penalty

DCGAN	LSGAN	WGAN (clipping)	WGAN-GP (ours)	
Baseline ( $G$ : DCGAN, $D$ : DCGAN)				
$G$ : No BN and a constant number of filters, $D$ : DCGAN				
$G$ : 4-layer 512-dim ReLU MLP, $D$ : DCGAN				
No normalization in either $G$ or $D$				
Gated multiplicative nonlinearities everywhere in $G$ and $D$				
tanh nonlinearities everywhere in $G$ and $D$				
101-layer ResNet $G$ and $D$				

## Summary

- ▶ Mode collapse and vanishing gradients are the two main problems of vanilla GAN. Lots of tips and tricks has to be used to make the GAN training is stable and scalable.
- ▶ DCGAN is the first GAN with deep convolutional architecture.
- ▶ Earth-Mover distance is a more appropriate objective function for distribution matching problem.
- ▶ Wasserstein GAN uses Kantorovich-Rubinstein duality to obtain EM distance.
- ▶ Weight clipping and gradient penalty are the ways to enforce Lipschitzness of the critic.