Generative Adversarial Networks

Wang Chao, Group of DL

Talk of 2017Spring CV

2017-6-1

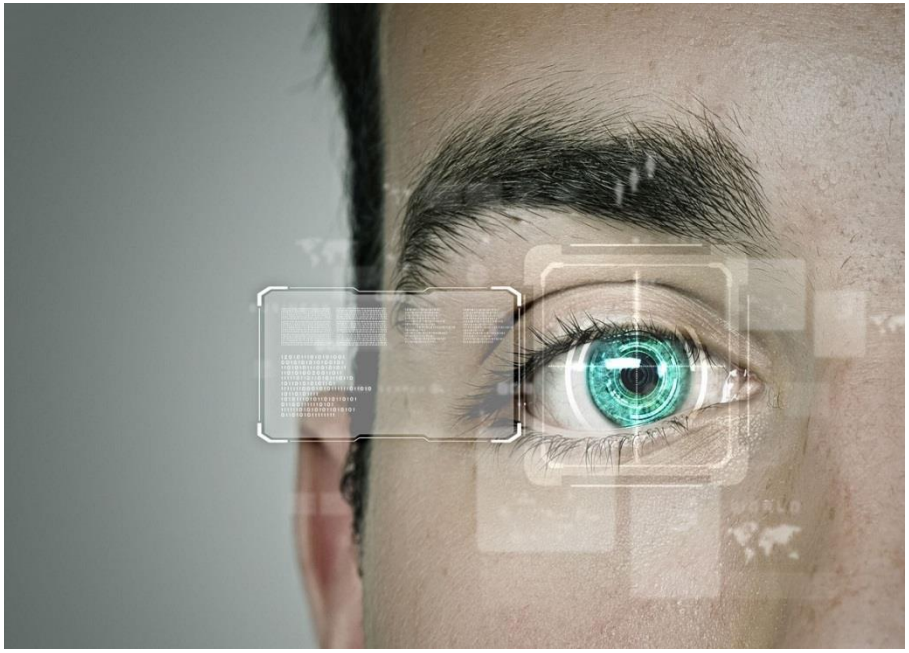
Overview

- Introduction
- Why learn GAN
- GANs in details
- Research Fronts
- Q&A

Introduction

Understand the world

Visual system



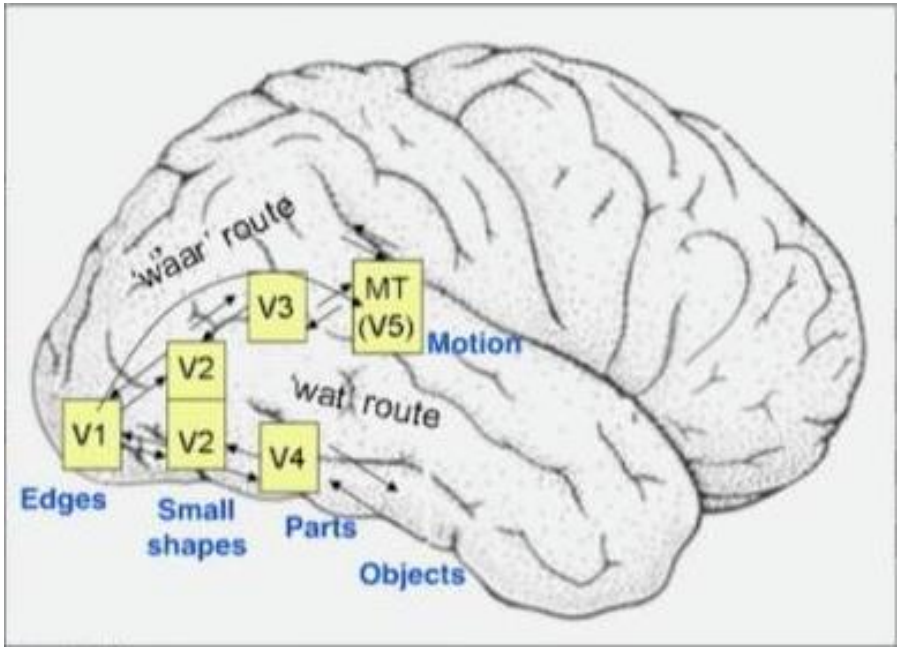
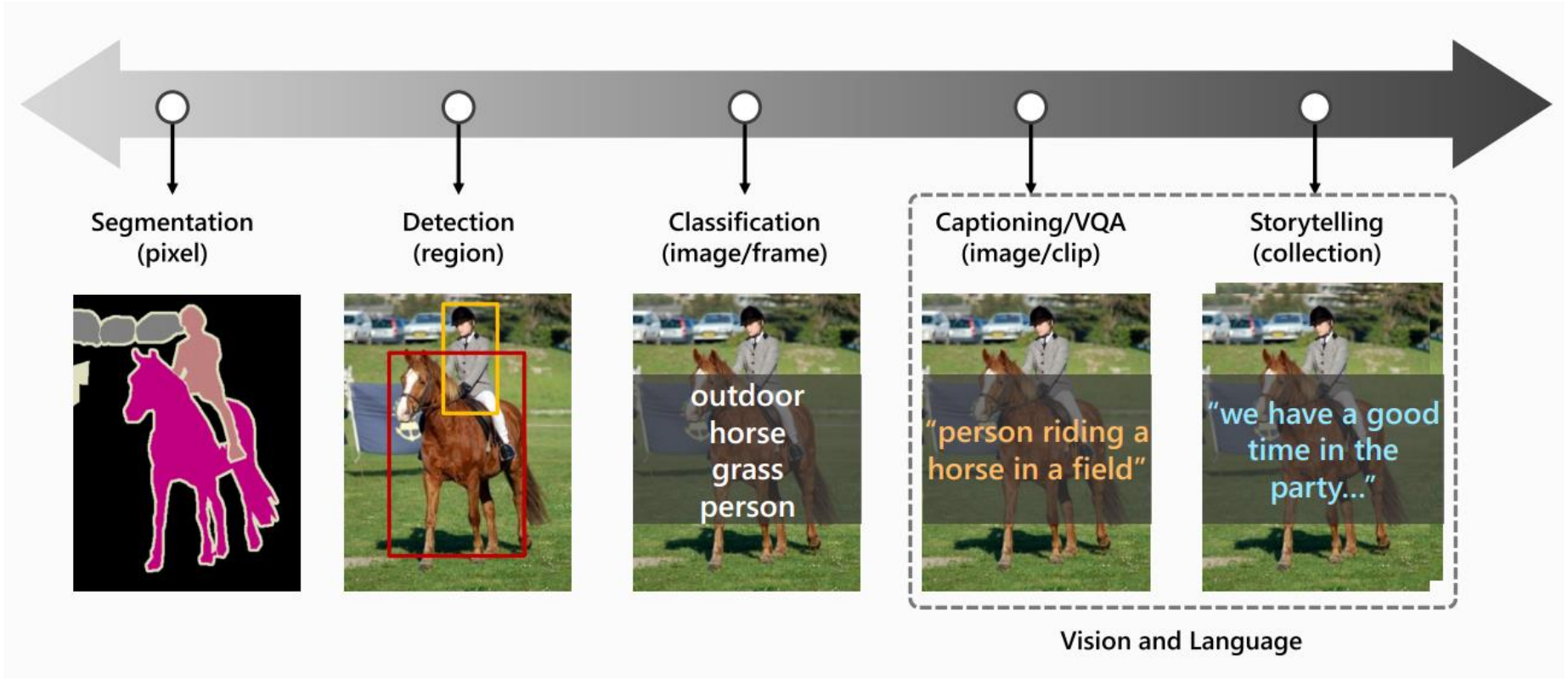


Image and video understanding



“What I cannot create, I do not understand.”

--- Richard Feynman

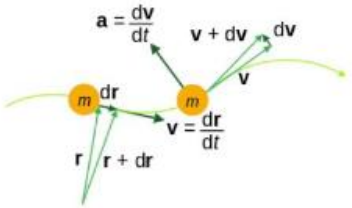
Representation learning

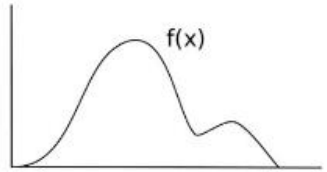
Things...

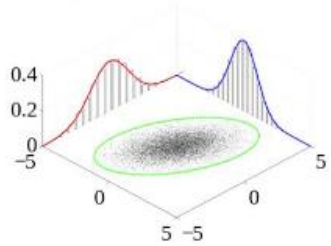
My heart beats as if the world is dropping, you may not feel the love but i do its a heart breaking moment of your life. enjoy the times that we have, it might not sound good but one thing it rhymes it might not be romantic but i think it is great,the best rhyme i've ever heard.

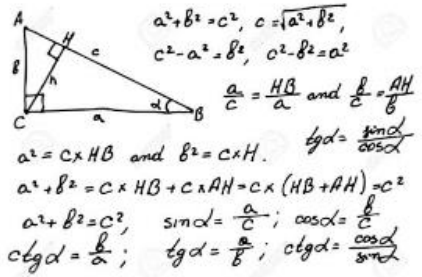
Representation

Engineering Knowledge...

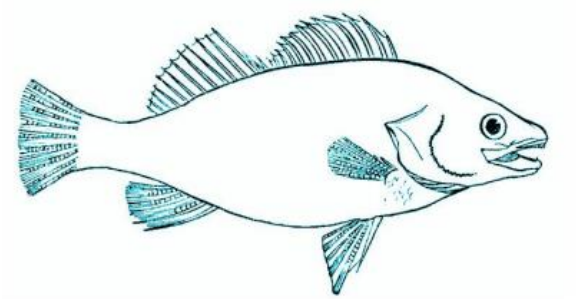




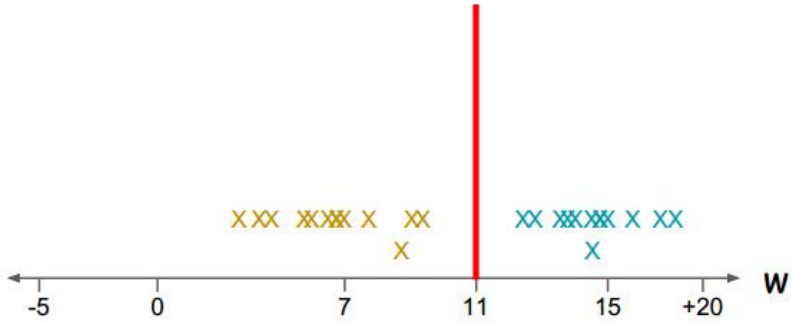


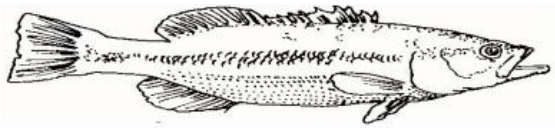


Representation Learning

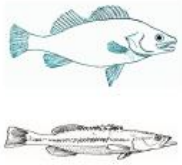


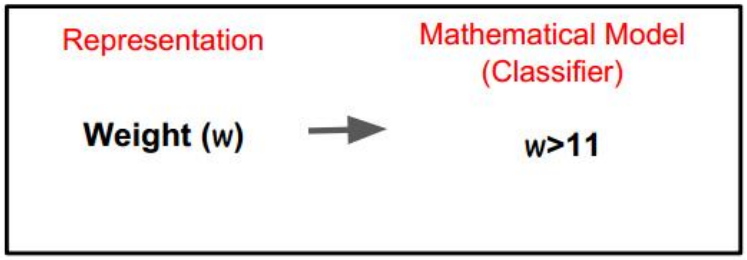
~12 lbs





~8 lbs

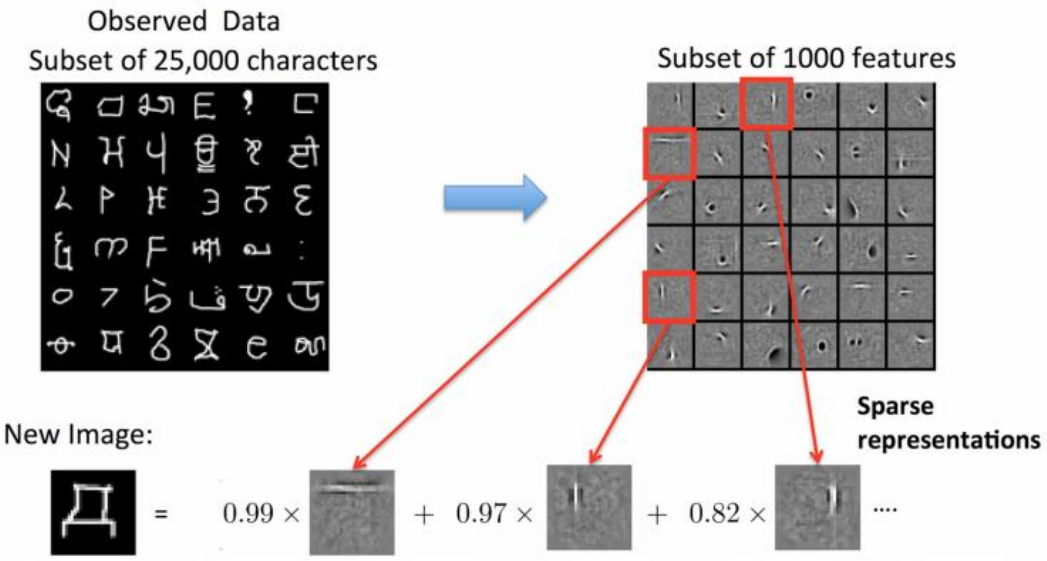


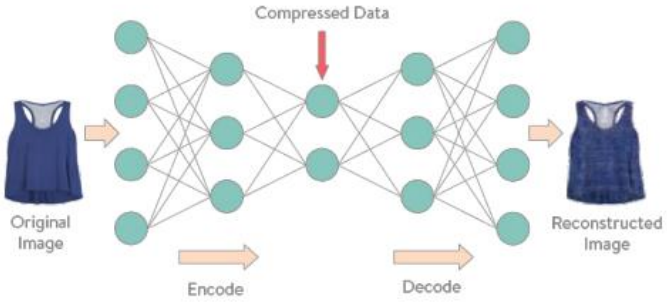


Representation Learning

Represent the cat for a cat detector!

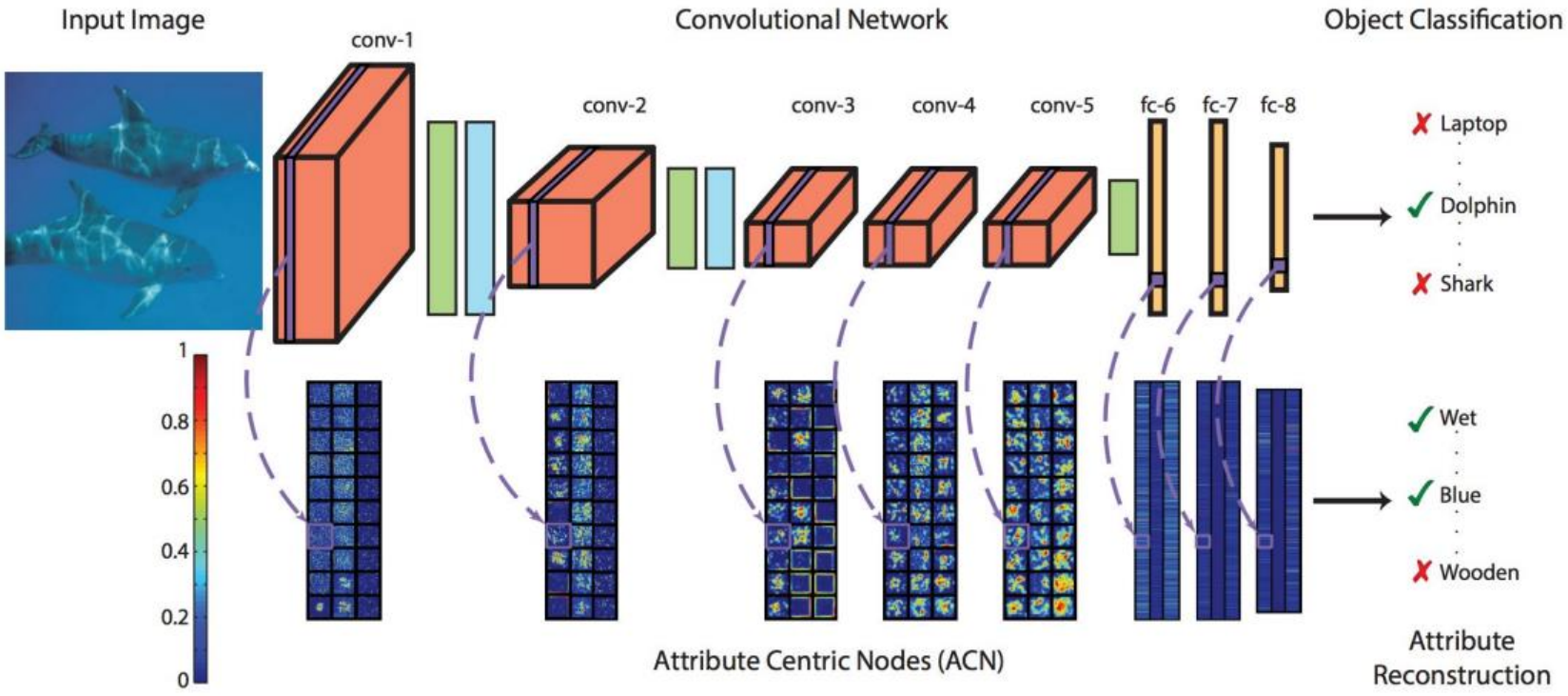
Representation Learning





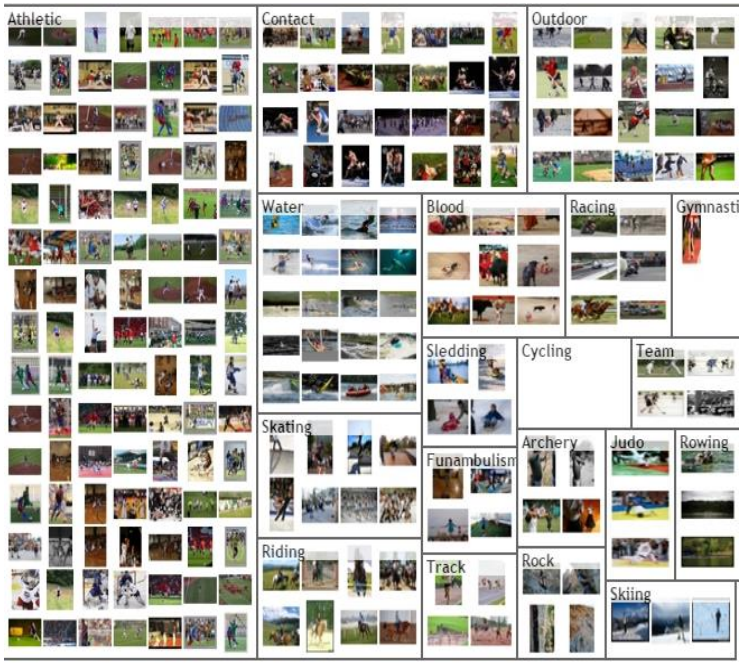
Unsupervised learning

Representation Learning



Supervised learning

GAN



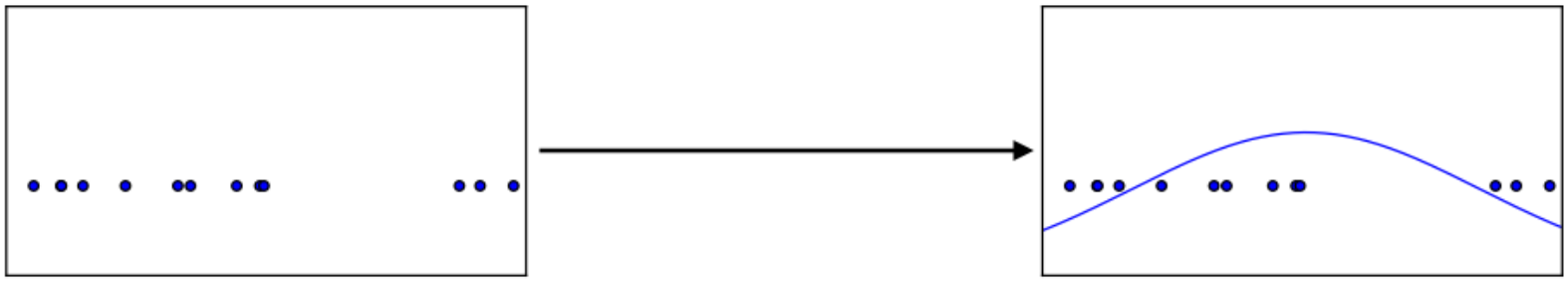
GAN model
Total 8 layers
Only 145MB

VS

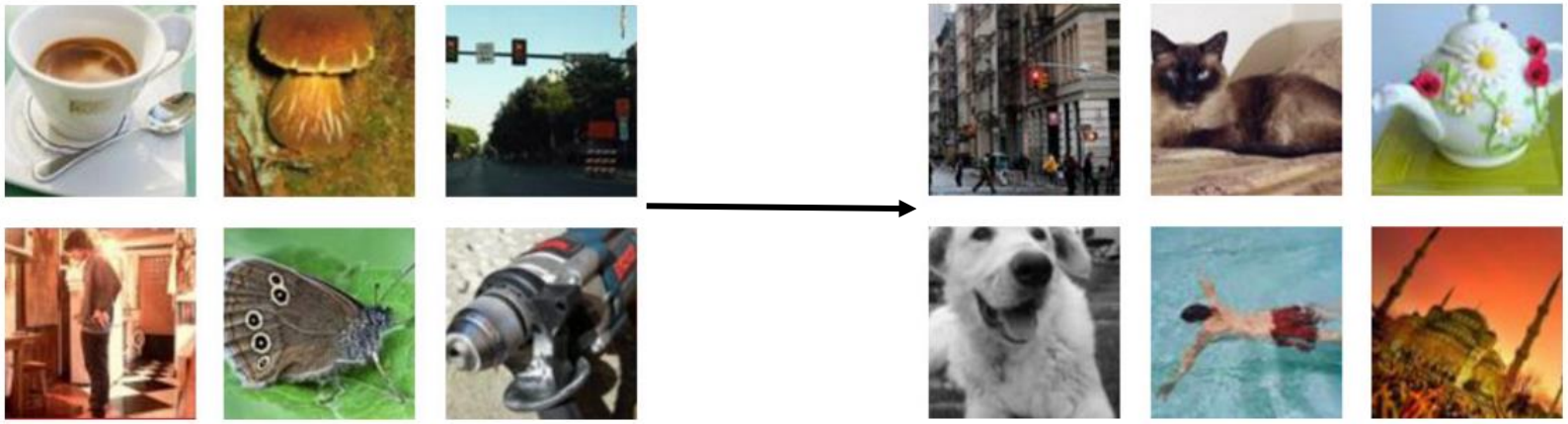
ImageNet
1.3 million images
200GB

Generative Modeling

- Probability density estimation



- Sample generation

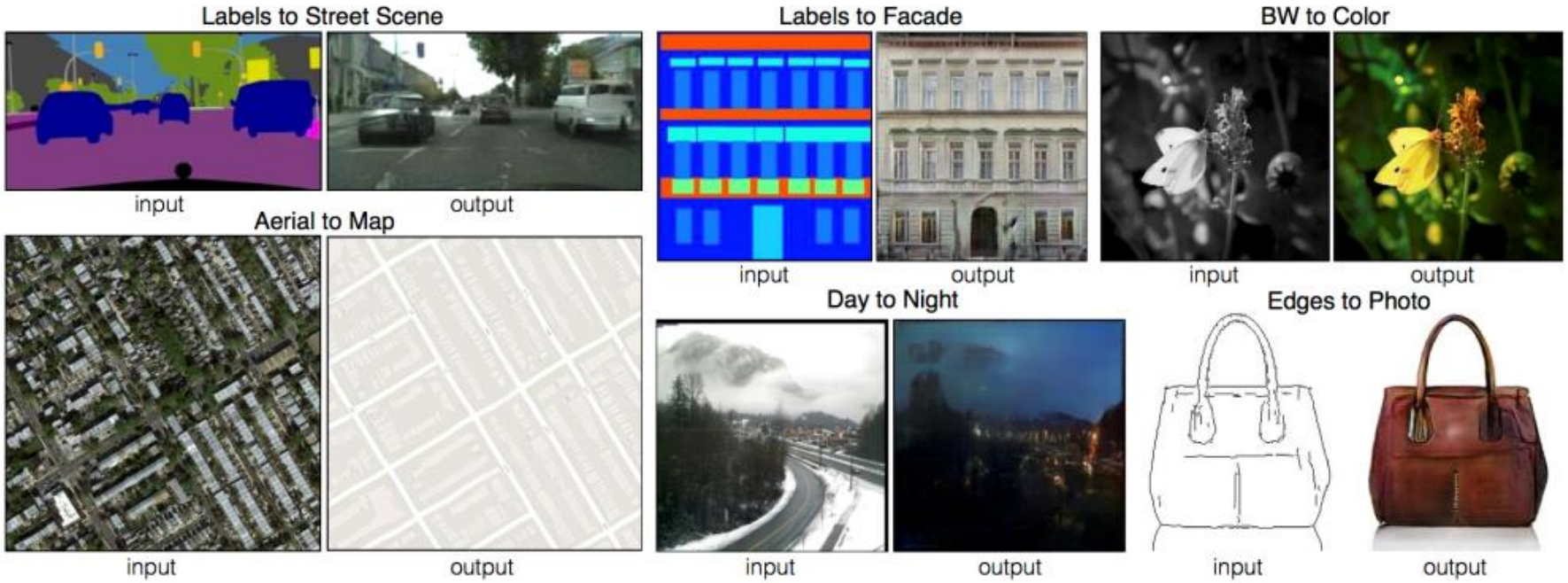


Why Generative Model?

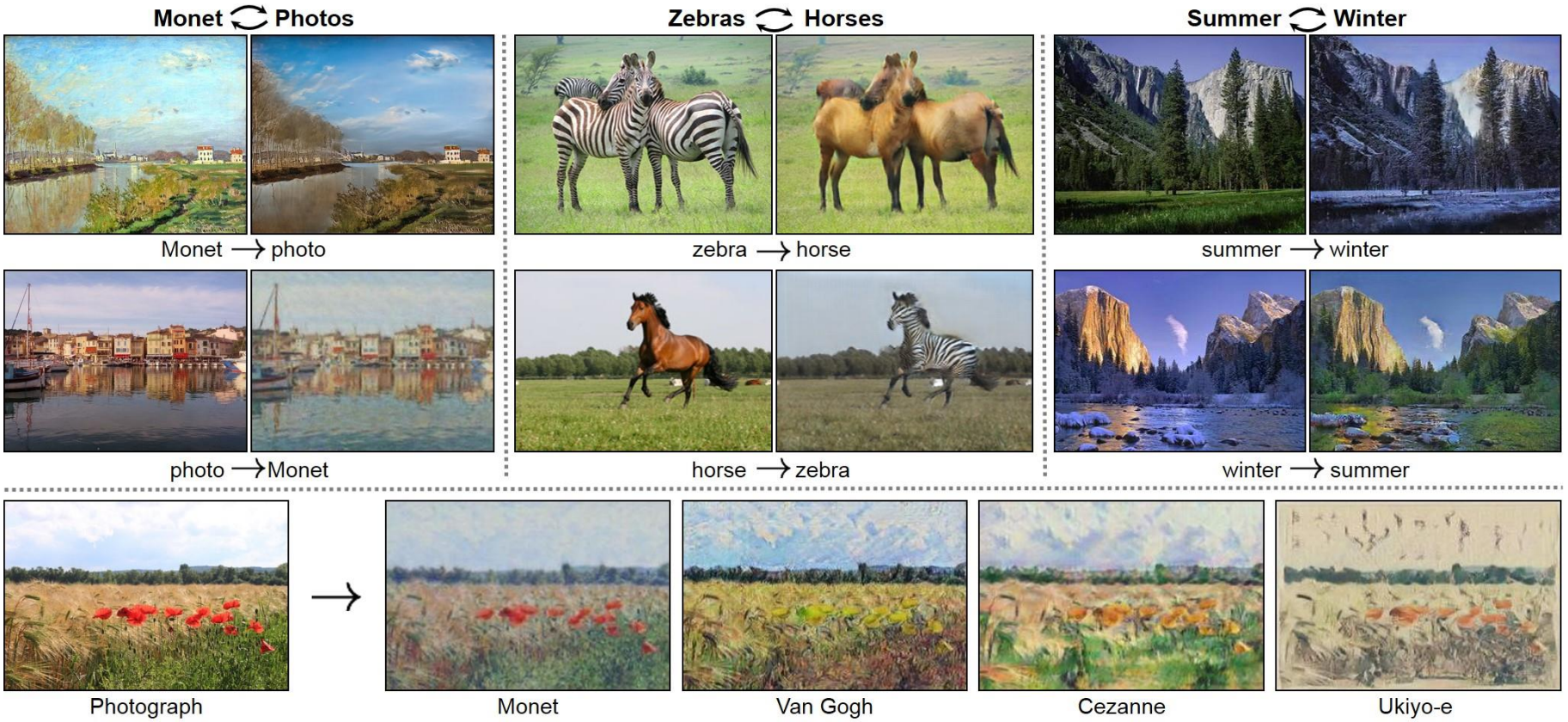
- Test of our ability to use high-dimensional, complicated data
- RL
- Missing data
- Any situation need “generative model”

Applications

Image to image translation



Isola *et al.* "Image-to-Image Translation with Conditional Adversarial Networks". CVPR 2017.



Zhu *et al.* "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks ". arXiv 2017.

Text to image

Text descriptions (content)

Images (style)

The bird has a **yellow breast** with **grey** features and a small beak.

This is a large **white** bird with **black wings** and a **red head**.

A small bird with a **black head and wings** and features grey wings.

This bird has a **white breast**, brown and white coloring on its head and wings, and a thin pointy beak.

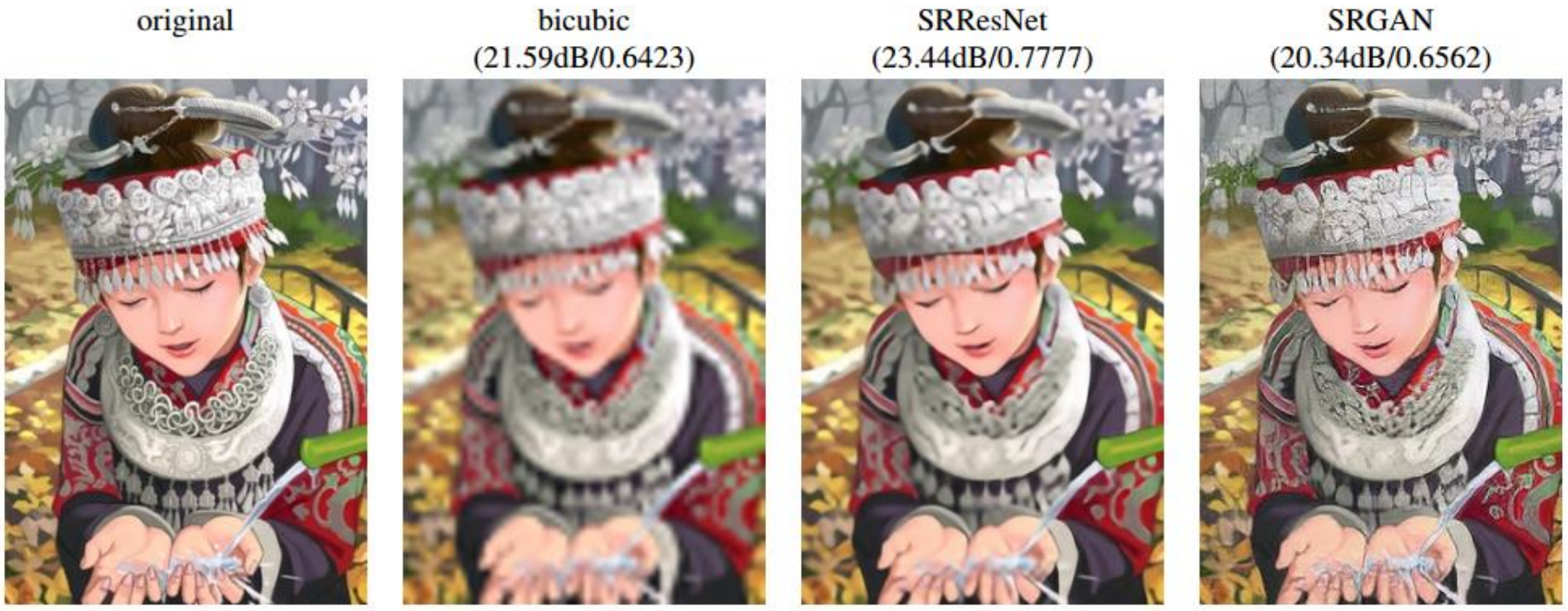
A small bird with **white base** and **black stripes** throughout its belly, head, and feathers.

A small sized bird that has a cream belly and a short pointed bill.

This bird is **completely red**.

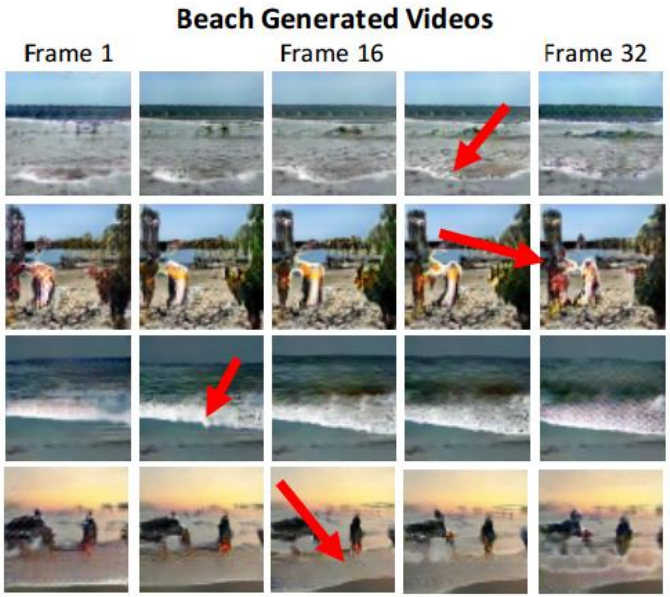
Scott *et al.* "Generative Adversarial Text to Image Synthesis". ICML 2016.

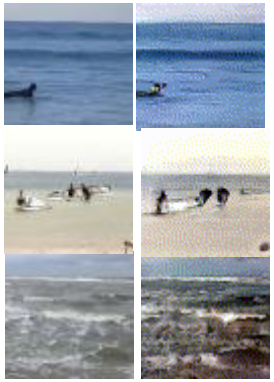
Super resolution



Ledig *et al.* "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network". CVPR 2017.

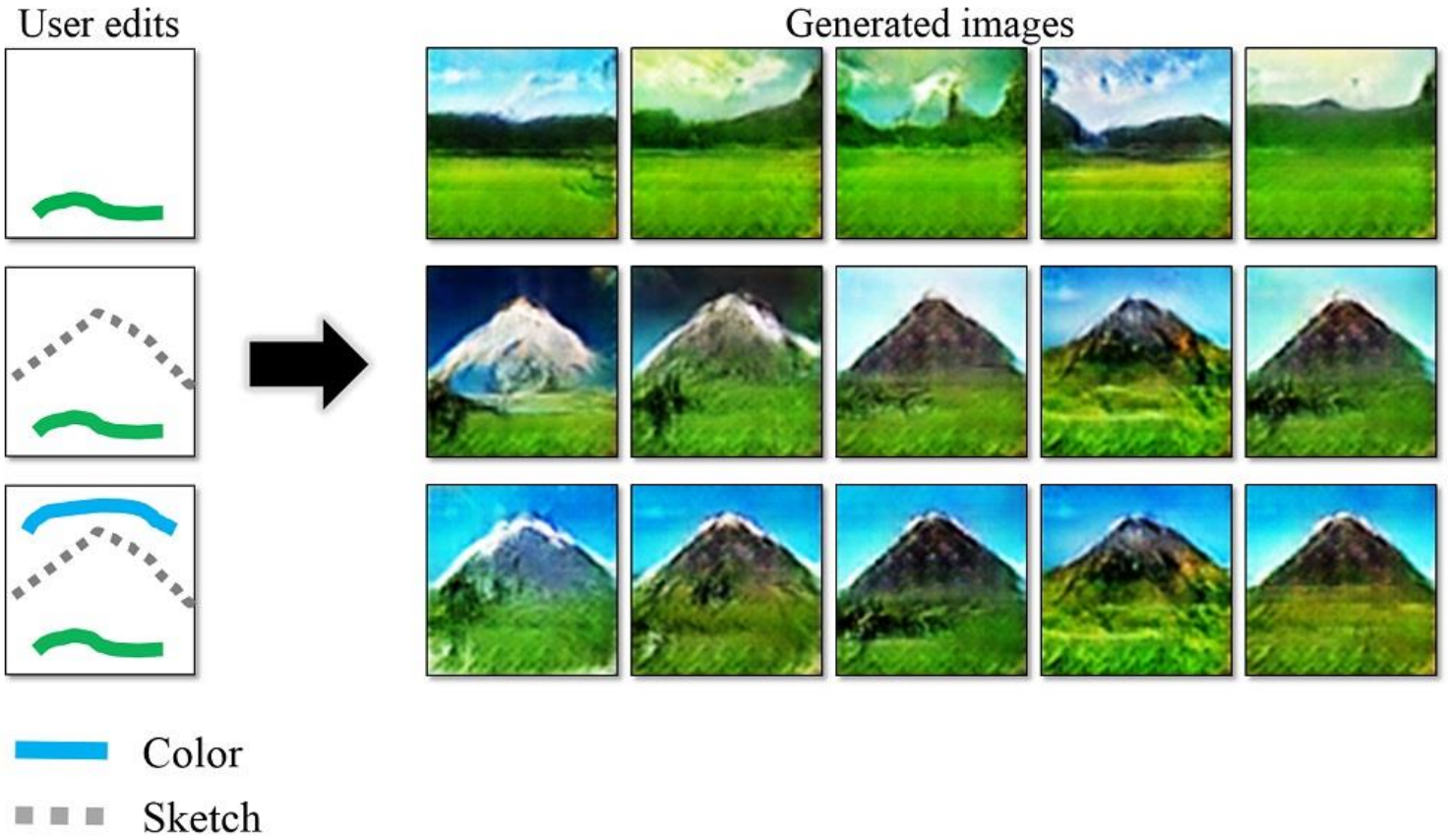
Video Prediction





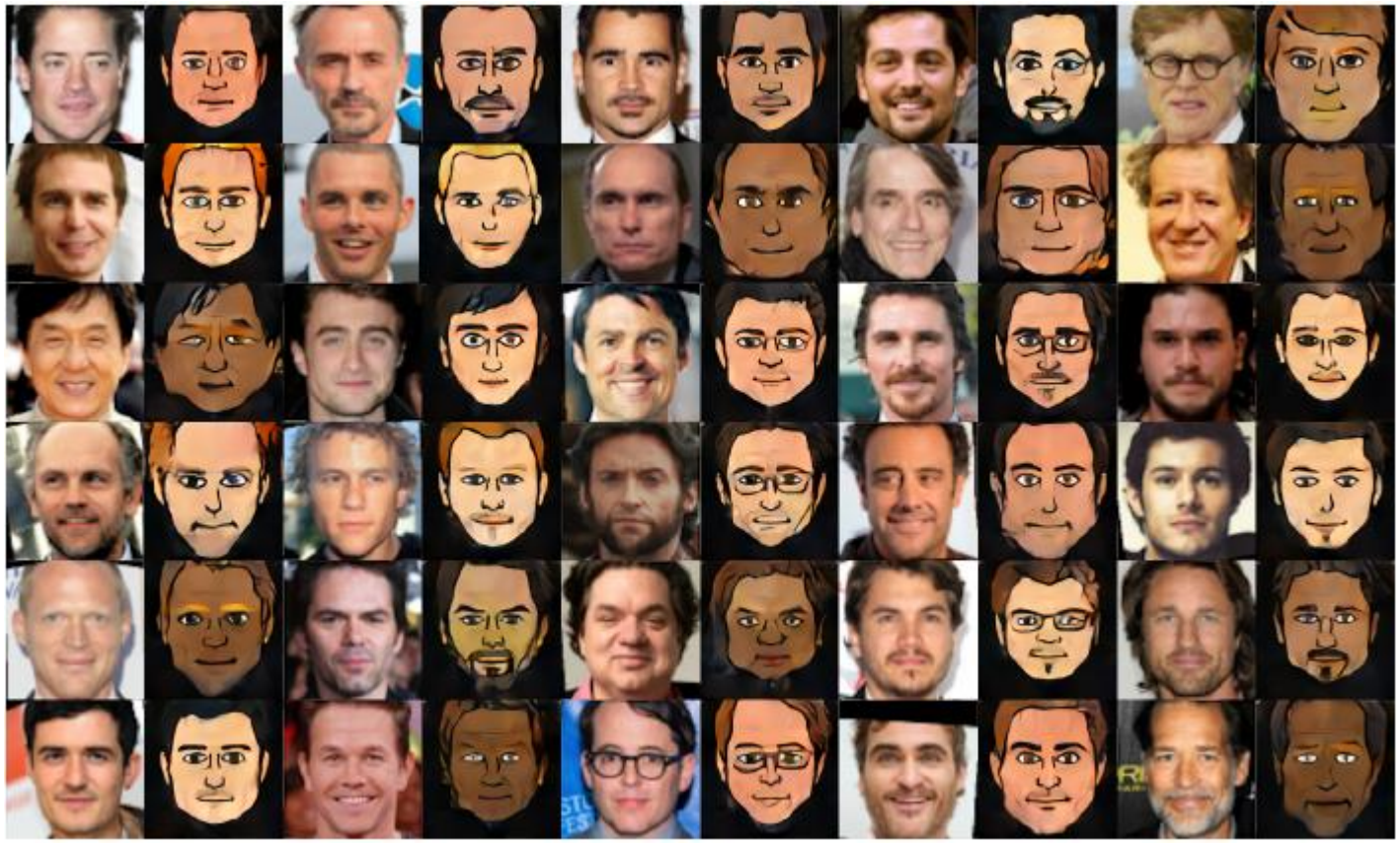
Vondrick *et al.* "Generating Videos with Scene Dynamics". NIPS 2016.

iGAN



Zhu *et al.* "Generative Visual Manipulation on the Natural Image Manifold". ECCV 2016.

Unsupervised Cross-Domain Image Generation



Taigman *et al.* "Unsupervised Cross-Domain Image Generation". arXiv 2016.

GAN application zoo

Object Detection

Image editing

Semi-supervised learning

Music Generation

Semantic Segmentation

SeqGAN

Style transfer

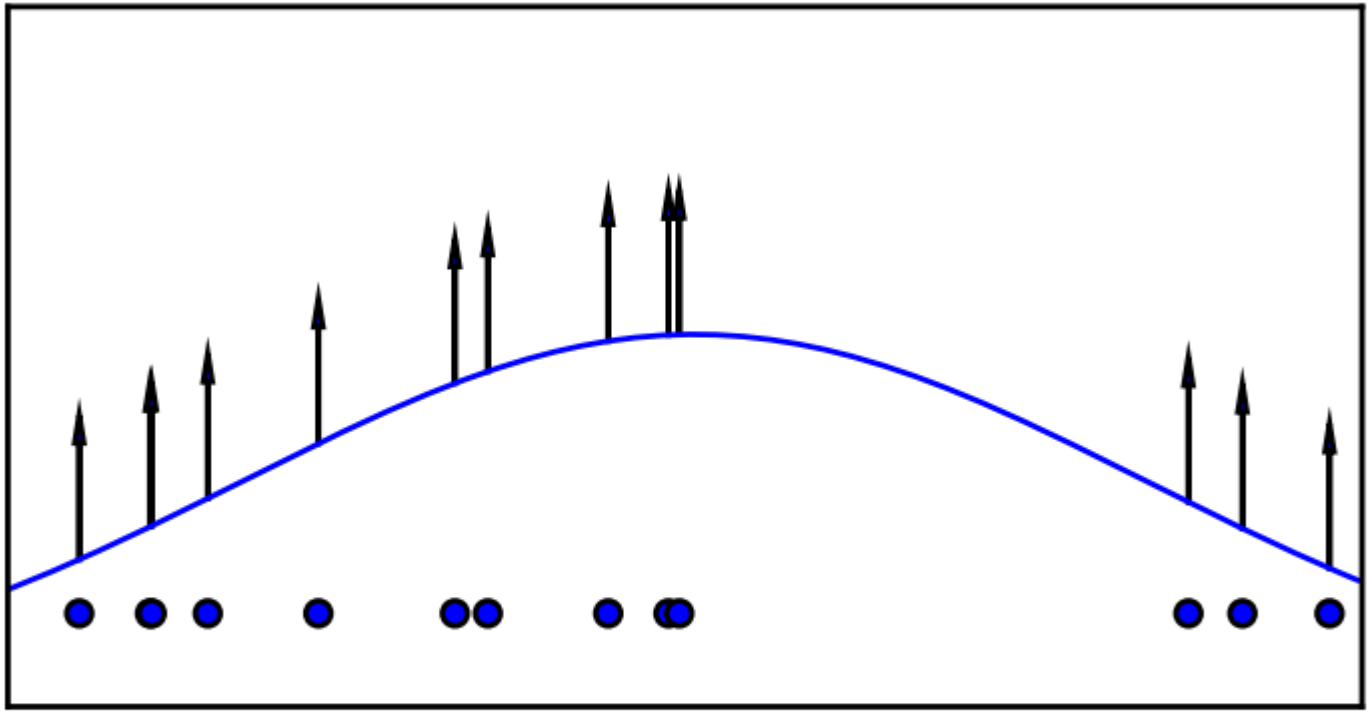
Drug concentration

RNN-GAN

Clustering

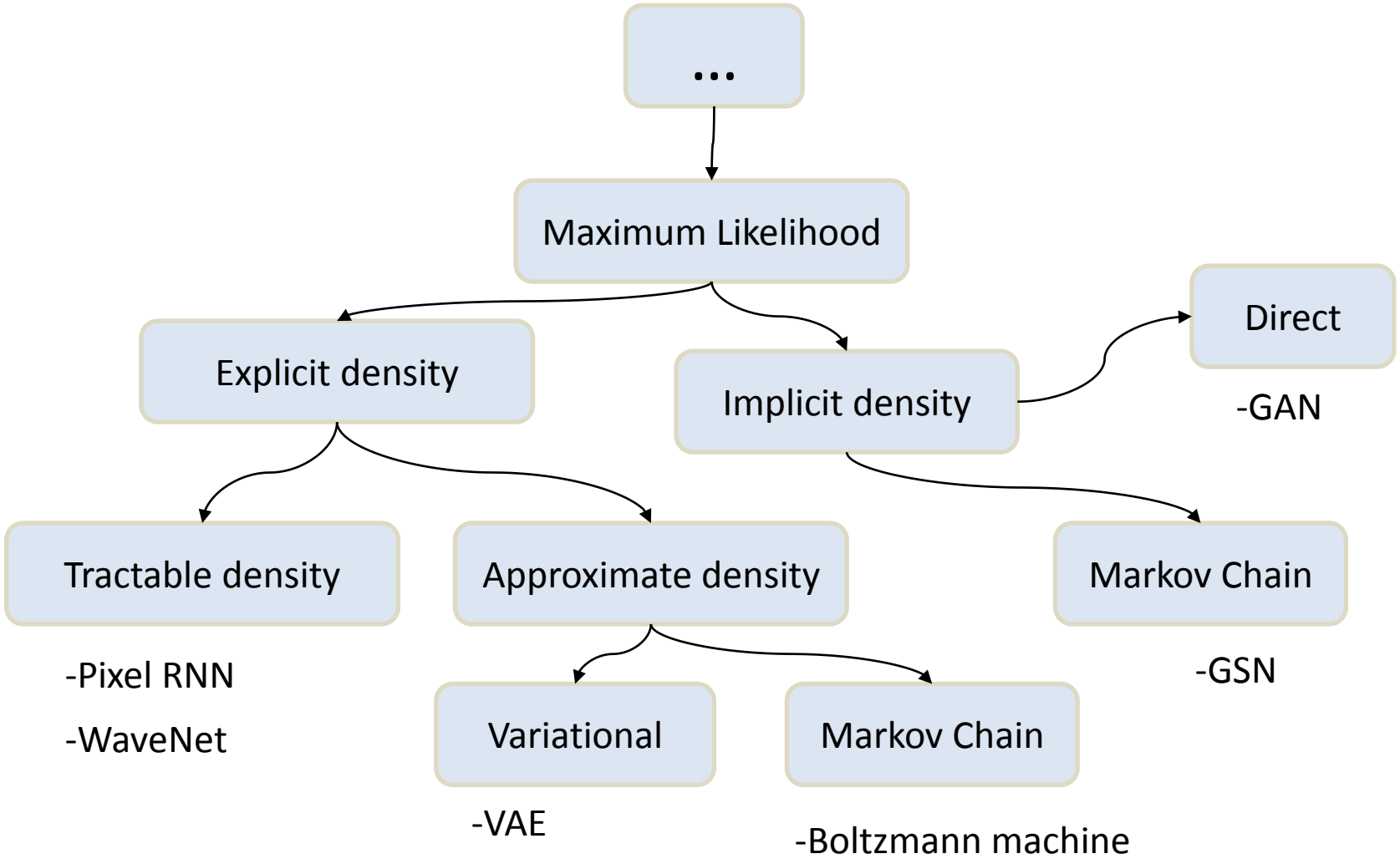
How Generative model works?

Maximum Likelihood



$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(\boldsymbol{x} \mid \boldsymbol{\theta})$$

Generative Models



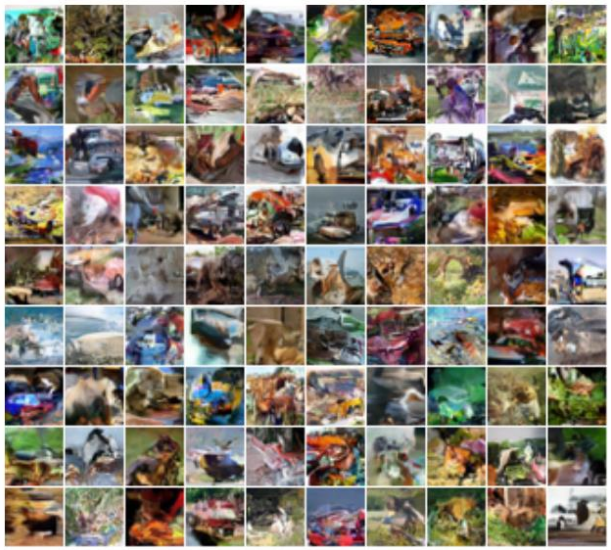
Pixel RNN

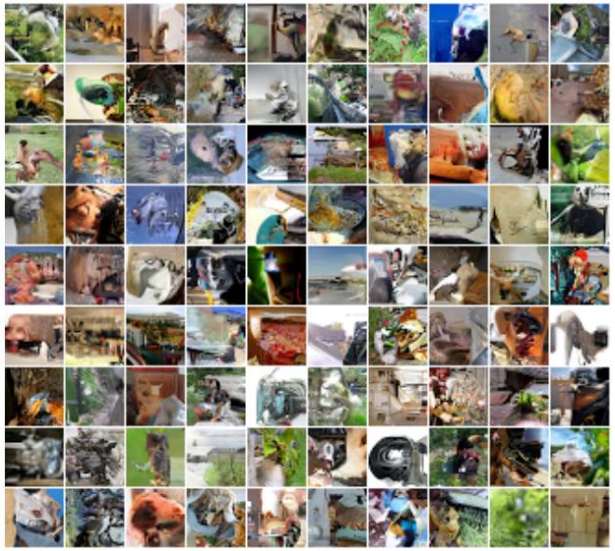
- Explict formula based on chain rule:

$$P_{model}(x) = P_{model}(x_1) \prod_{i=2}^{n} P_{model}(x_i | x_1, \ldots, x_{i-1})$$

Disadvantages:

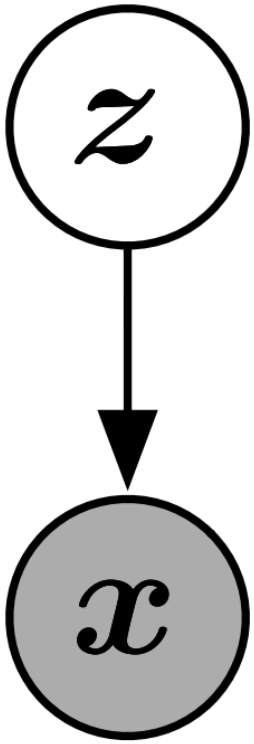
- O(n) sample generation cost
- Generation not controlled by a latent code





Oord *et al.* "Pixel Recurrent Neural Networks". ICML 2016.

Variational Autoencoder(VAE)



- $logp(x) \geq \log p(x) - D_{KL}(q(z)||p(z|x))$
  $= \mathbb{E}_{z\sim q} logp(x,z) + H(q)$
- Disadvantages:
  - Not asymptotically consistent unless q is perfect
  - Sample tend to have lower quality

CIFAR10 samples

Kingma *et al.* "Auto-Encoding Variational Bayes". ICLR 2016.

Adversarial training

- A phrase whose usage is in flux; a new term that applies to both new and old ideas.

- **Examples**:
- An agent playing against a copy of itself in a board game
- Robust optimization / robust control
- Training neural networks on adversarial examples

GANs in details

- Architecture
- Object function
- Theory(Minimax Game)
- Problems

GAN

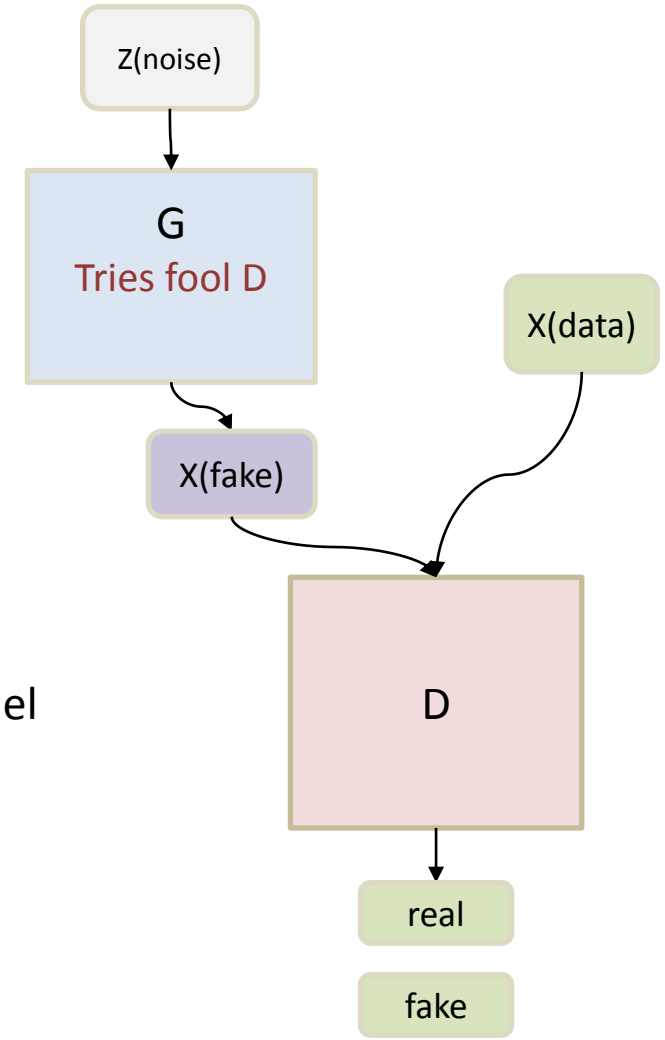
[0.1,0.5,1.2,1.1,0.6..]

Games

- A counterfeiter-police game between two components： a generator G and a discriminator D
- **G:** counterfeiter -> trying to fool police with fake currency
- **D:** policy, trying to detect the counterfeit currency

GAN Framework

Samples from G model



Samples from X data

Object Function

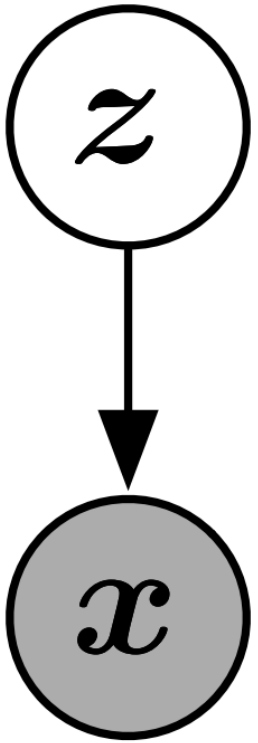
$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1-D(G(z))]$$

- D(x)： predict that real data's label is "1"
- D(G(z))： predict that fake data's label is "0"

Goodfellow *et al.* "Generative Adversarial Networks". NIPS 2014.

Generator Network

- Use a latent code
- Unlike variational method
- No Markov chains needed
- Regards as producing best results



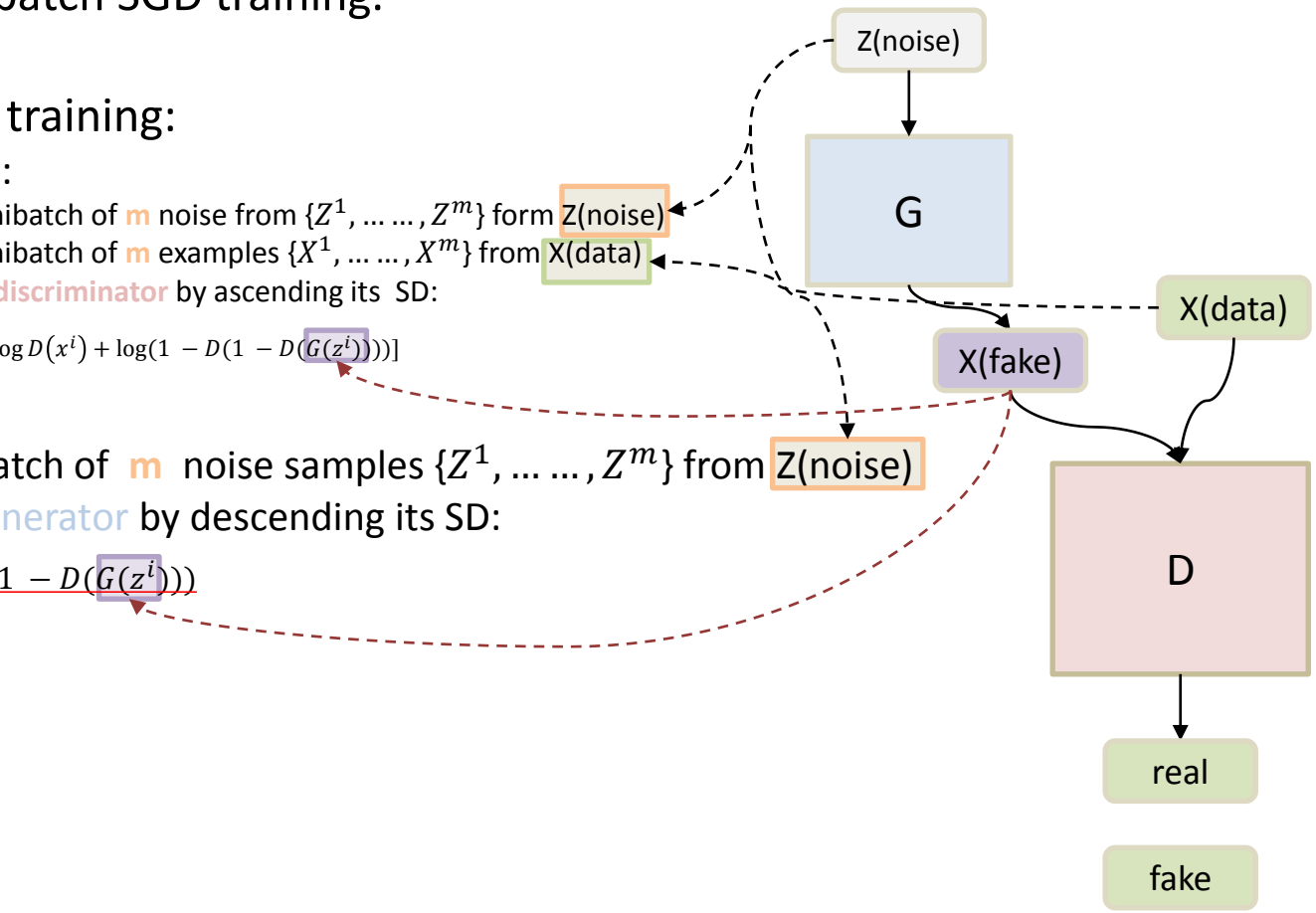
Goodfellow *et al.* "Generative Adversarial Networks". NIPS 2014.

Training algrothim

- Algorithm: minibatch SGD training.
- For iterations in training:
  - For K steps do:
    - Samples minibatch of m noise from $\{Z^1, \dots \dots, Z^m\}$ form Z(noise)
    - Samples minibatch of m examples $\{X^1, \dots \dots, X^m\}$ from X(data)
    - Update the discriminator by ascending its SD:

$$\nabla_{\theta d} \frac{1}{m} \sum_{i=1}^{m} [\log D(x^i) + \log(1 - D(1 - D(G(z^i))))]$$

  - End for
  - Sample minibatch of m noise samples $\{Z^1, \dots \dots, Z^m\}$ from Z(noise)
  - Update the generator by descending its SD:
    - $\nabla_{\theta d} \frac{1}{m} \sum_{i=1}^{m} (1 - D(G(z^i)))$
- End for



Goodfellow *et al.* "Generative Adversarial Networks". NIPS 2014.

Minimax Game

- $J^d = -\frac{1}{2}\mathbb{E}_{x\sim P_{data}(x)}[\log D(x)] - \frac{1}{2}\mathbb{E}_z \log\left(1 - D\left(G(z)\right)\right)$
- $J^g = -J^d$

- **Zero-sum** game
- **D**'s goal: maximize V(D,G)
- **G**'s goal: minimize V(D,G)

Goodfellow *et al.* "Generative Adversarial Networks". NIPS 2014.

Discriminator

- *Global Optimality*: $P_{\text{data}} = P_g$
- *Fix G*: $D(x) = \dfrac{P_{data}(x)}{P_{data}(x)+P_{model}(x)}$
- Maximize: $V(G, D)$

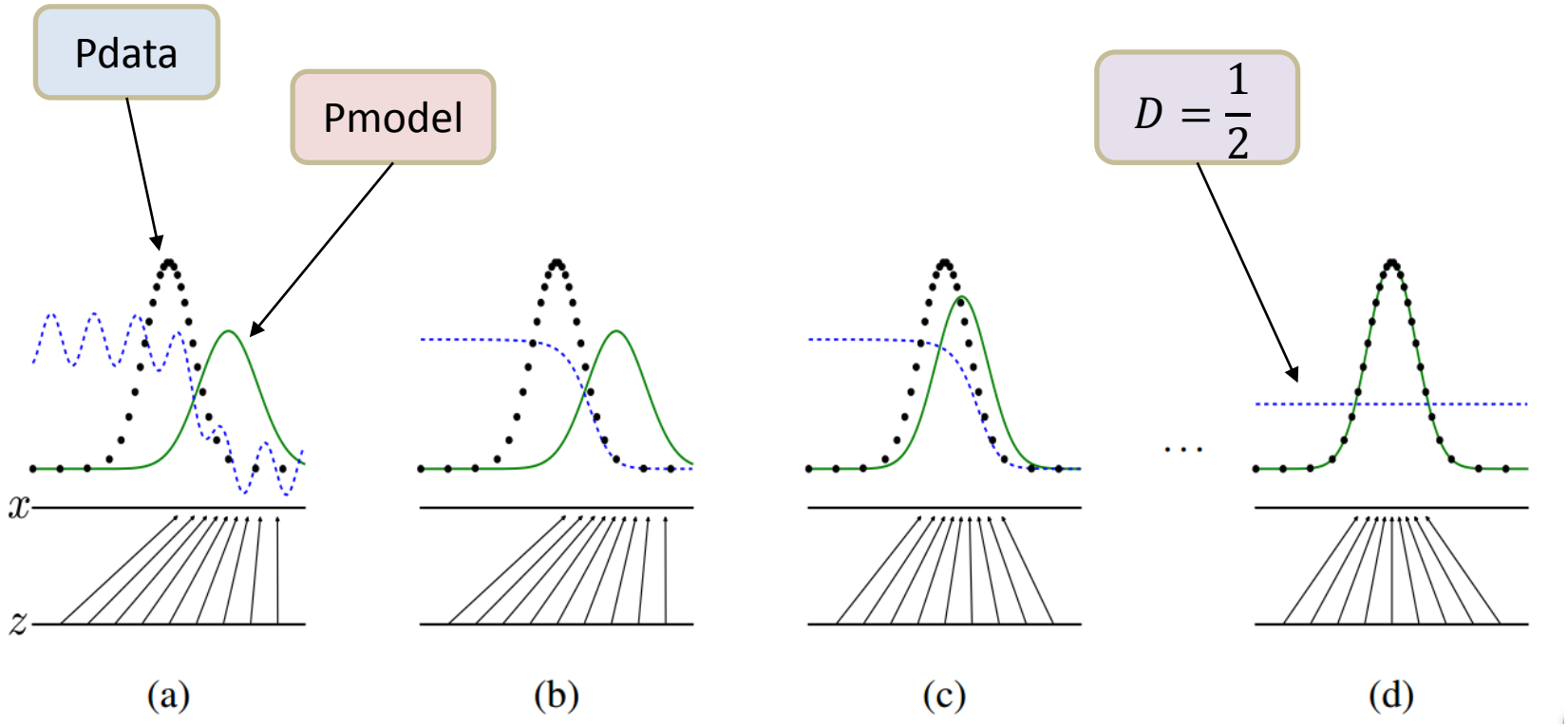
$$V(G,D) = \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x}))dx + \int_{z} p_{\boldsymbol{z}}(\boldsymbol{z}) \log(1 - D(g(\boldsymbol{z})))dz$$
$$= \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}) \log(D(\boldsymbol{x})) + p_g(\boldsymbol{x}) \log(1 - D(\boldsymbol{x}))dx$$

$$y \to a \log(y) + b \log(1 - y)$$

$$y = \frac{a}{a+b}$$

Goodfellow *et al.* "Generative Adversarial Networks". NIPS 2014.

Discriminator Strategy



Goodfellow *et al.* "Generative Adversarial Networks". NIPS 2014.

Non-Saturating Game

- $J^d = -\frac{1}{2}\mathbb{E}_{x\sim P_{data}(x)}[\log D(x)] - \frac{1}{2}\mathbb{E}_z \log\Big(1 - D\big(G(z)\big)\Big)$
- $J^g = -\frac{1}{2}\mathbb{E}_z \log D\big(G(z)\big)$

- No longer a single loss
- G model maximizes the log-probability of D being mistaken
- Heuristically, G can still learn even though D rejects all G samples

Goodfellow *et al.* "Generative Adversarial Networks". NIPS 2014.

Examples

Goodfellow *et al.* "Generative Adversarial Networks". NIPS 2014.

“There’s no free lunch.”

original

generated

Problems

- No convergence
- Mode collapse
- Mode missing
- Unstabitily

Non convergence

- Optimization algorithms often approach a saddle point or local minimum rather than a global minimum

- Game solving algorithms may not approach an equilibrium at all

Objective Function

- D function:

$$\log(D, g_\theta) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z))]$$

- G function:
  - Original: $E_{z\sim p_z(z)}[\log(1 - D(G(z))]$
  - Alternative: $E_{z\sim p_z(z)}[-\log D(g_\theta(z))]$
- Zero-sum Game -> non-saturating game


中国人民银行
100
壹佰圆
Z3E9172945
100


中国人民银行
100
壹佰圆
AA00000000
100

Non convergence

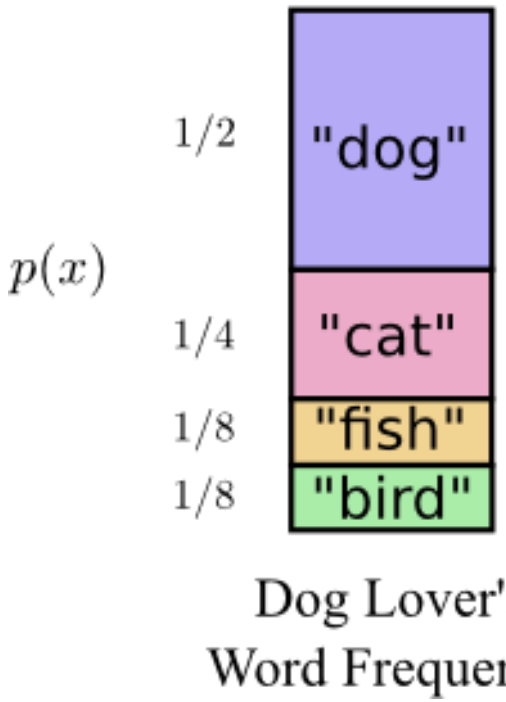
- G original: $E_{x\sim p_{data}(x)}[\log D(x)]+E_{z\sim p_z(z)}[\log(1-D(G(z))]$
- D optimal: $D^*(x)=\frac{P_r(x)}{P_r(x)+P_g(x)}$
- G become: $E_{x\sim p_{data}(x)}[\log\frac{P_r(x)}{\frac{1}{2}[P_r(x)+P_g(x)]}]+E_{z\sim p_z(z)}[\log\frac{P_g(x)}{\frac{1}{2}[P_r(x)+P_g(x)]}]$

$-2\log 2$

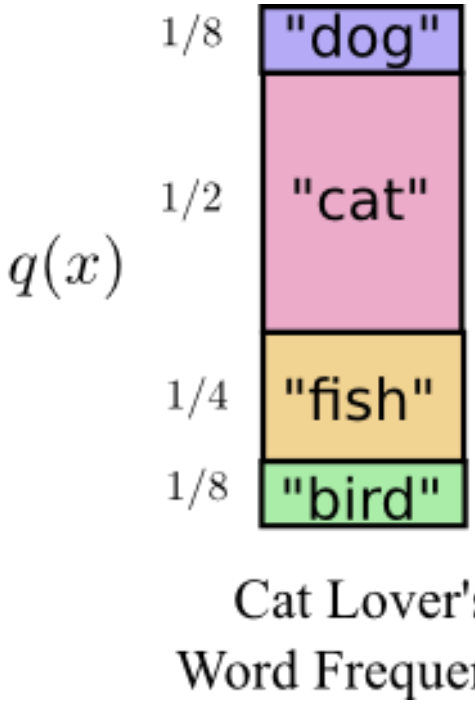
$$L(D^*,g_\theta)=2JSD[P_r\,\|\,P_g]-2\log 2$$

Understanding KL&JS

Bob



Dog Lover's
Word Frequency

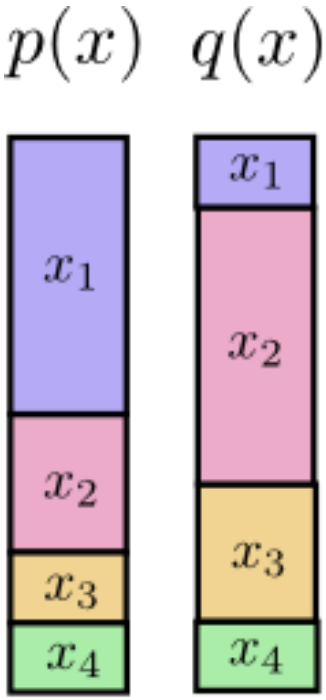


Cat Lover's
Word Frequency

Alice

(Colah's blog)

Understanding KL&JS

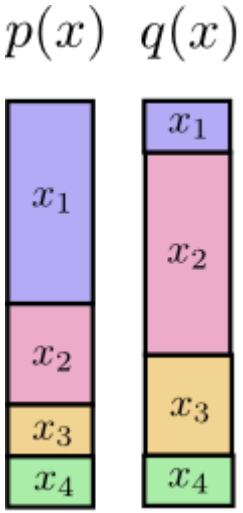


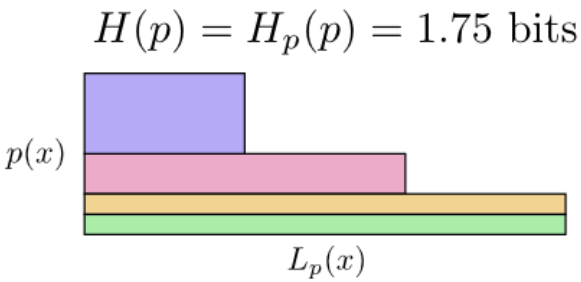
Cross-Entropy: $H_p(q)$

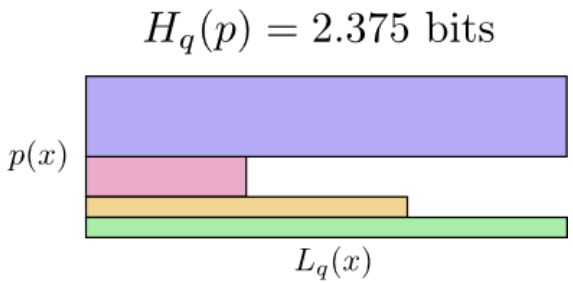
Average Length
of message from q(x)
using code for p(x).

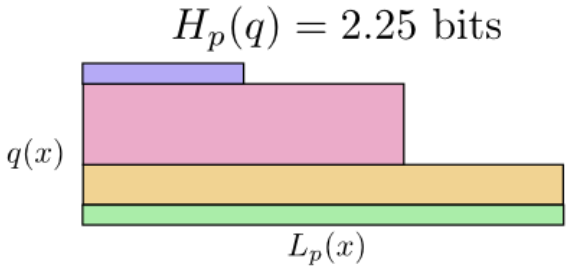
$$\mathrm{H_p}(q) = \sum_x q(x) \log_2 (\frac{1}{p(x)})$$

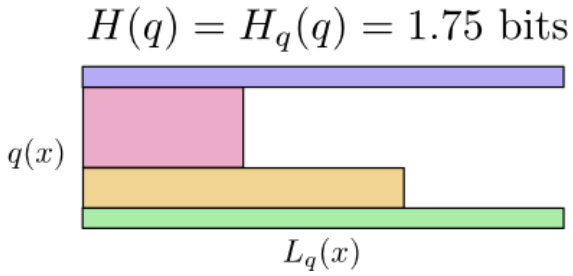
Understanding KL&JS





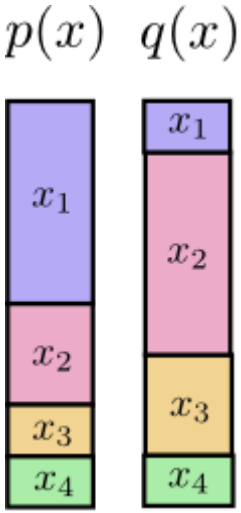






$$\mathrm{H_p}(q) \neq \mathrm{H_q}(p)$$

Understanding KL&JS



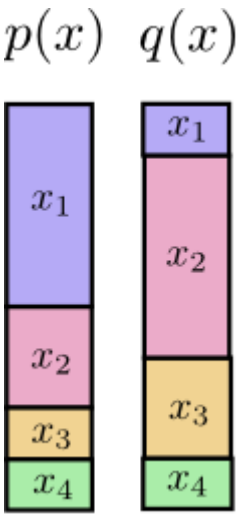
JS Divergence:

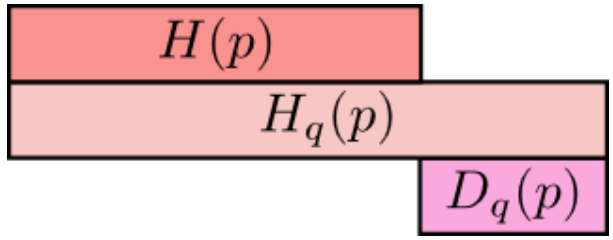
$$D_{JS}(\mathrm{p}\,|\,\mathrm{q}) = D_{JS}(\mathrm{q}\,|\,\mathrm{p}) = \frac{1}{2}D_{KL}(p\,|\,r) + \frac{1}{2}D_{KL}(q\,|\,r)$$

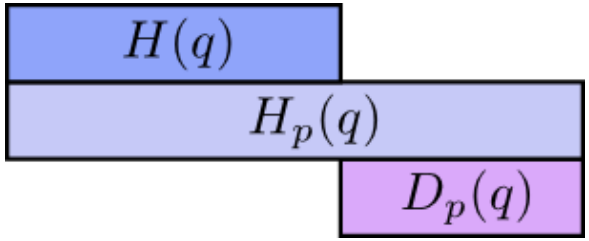
$$r = \frac{1}{2}(p+q)$$

KL Divergence: $D_q(\mathrm{p}) = \mathrm{H}_\mathrm{q}(p) - \mathrm{H}(p)$

Understanding KL&JS



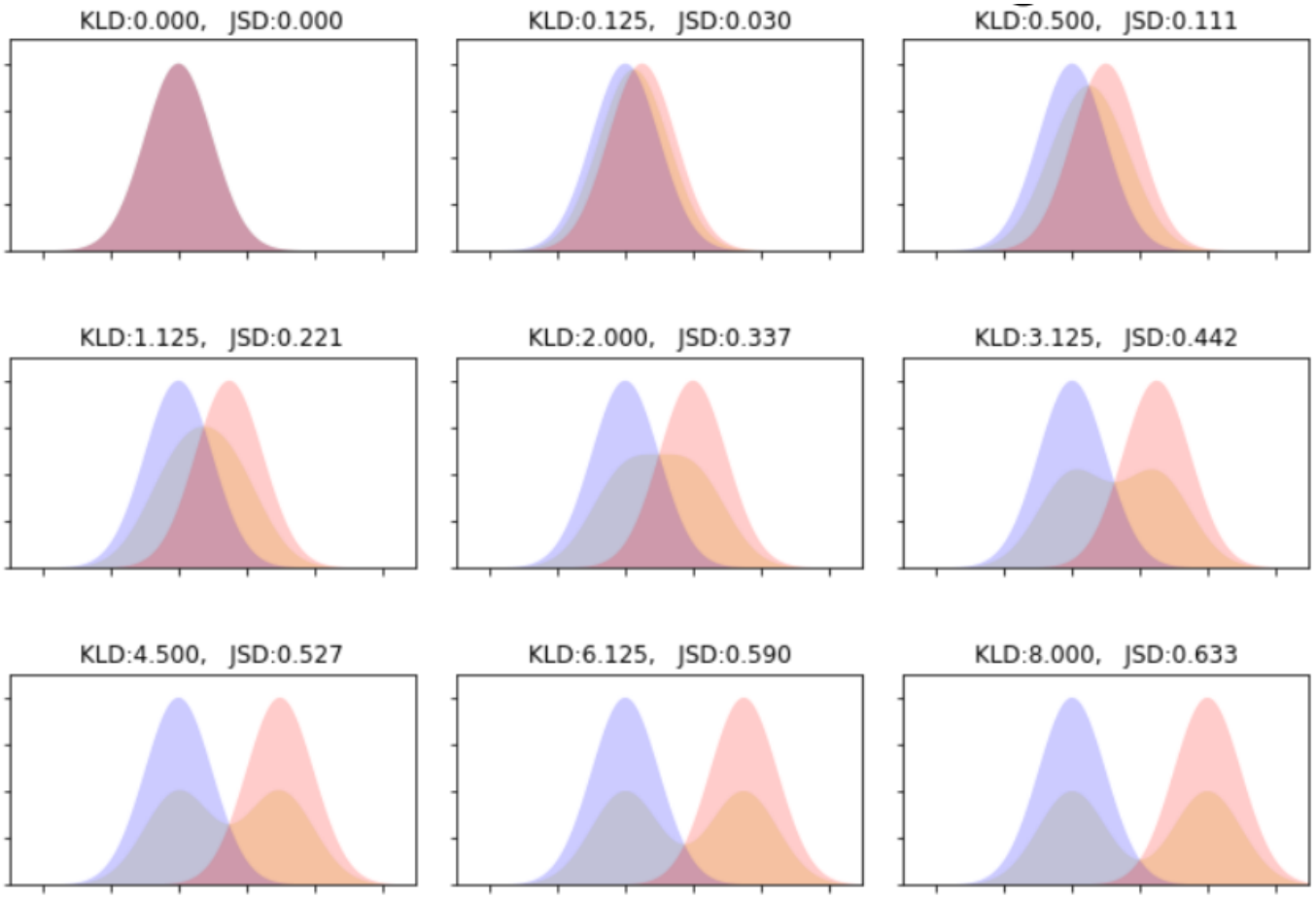




$$\mathrm{H_p}(q) \neq \mathrm{H_q}(p)$$

KL Divergence: $D_q(\mathrm{p}) = \mathrm{H_q}(p) - \mathrm{H}(p)$

Understanding KL&JS



Non convergence(original G function)

$$L(D^*, g_\theta) = 2JSD[P_r \parallel P_g] - 2\log 2$$

$$P_r(x) = 0 \text{且} P_g(x) = 0$$

$$P_r(x) \neq 0 \text{且} P_g(x) \neq 0$$

$$P_r(x) = 0 \text{且} P_g(x) \neq 0$$

$$P_r(x) \neq 0 \text{且} P_g(x) = 0$$

$$2\log 2$$

Unstable &Mode missing

- G alternative: $E_{z\sim p_z(z)}[-\log D(G(z))]$
- D optimal: $D^*(x)=\frac{P_r(x)}{P_r(x)+P_g(x)}$
- G become: $KL(P_g \| P_r)-2JS(P_r \| P_g)$

MIN

MAX

Unstable problem

- Minimizing the KL divergence whie

  Maximizing the JS divergence is crazy：

$$\mathrm{K}L(P_g \| P_r) - 2JS(P_r \| P_g)$$

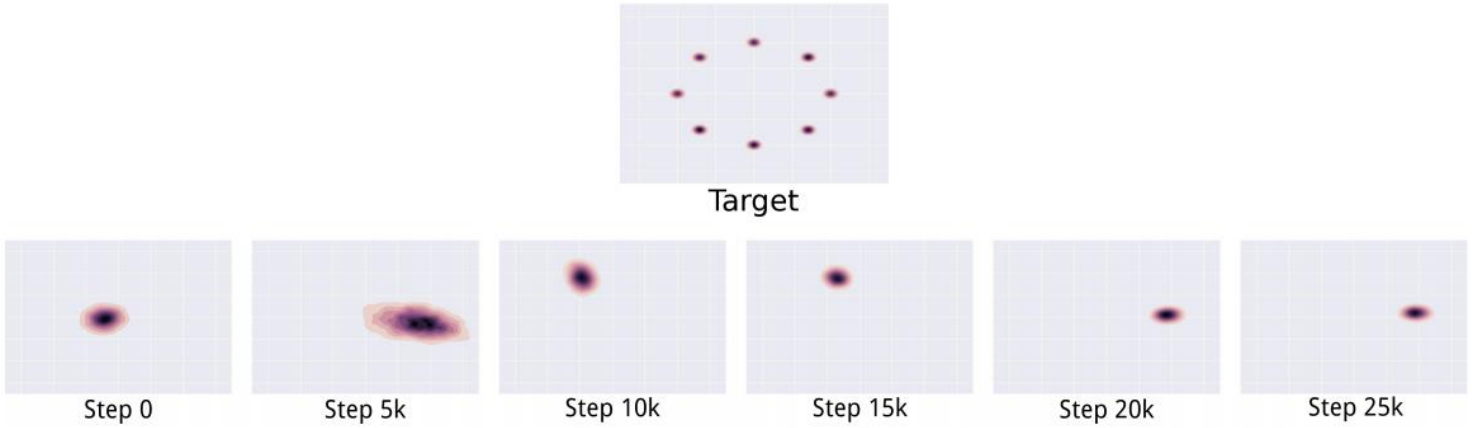
- Result in gradient unstable

Mode collapse

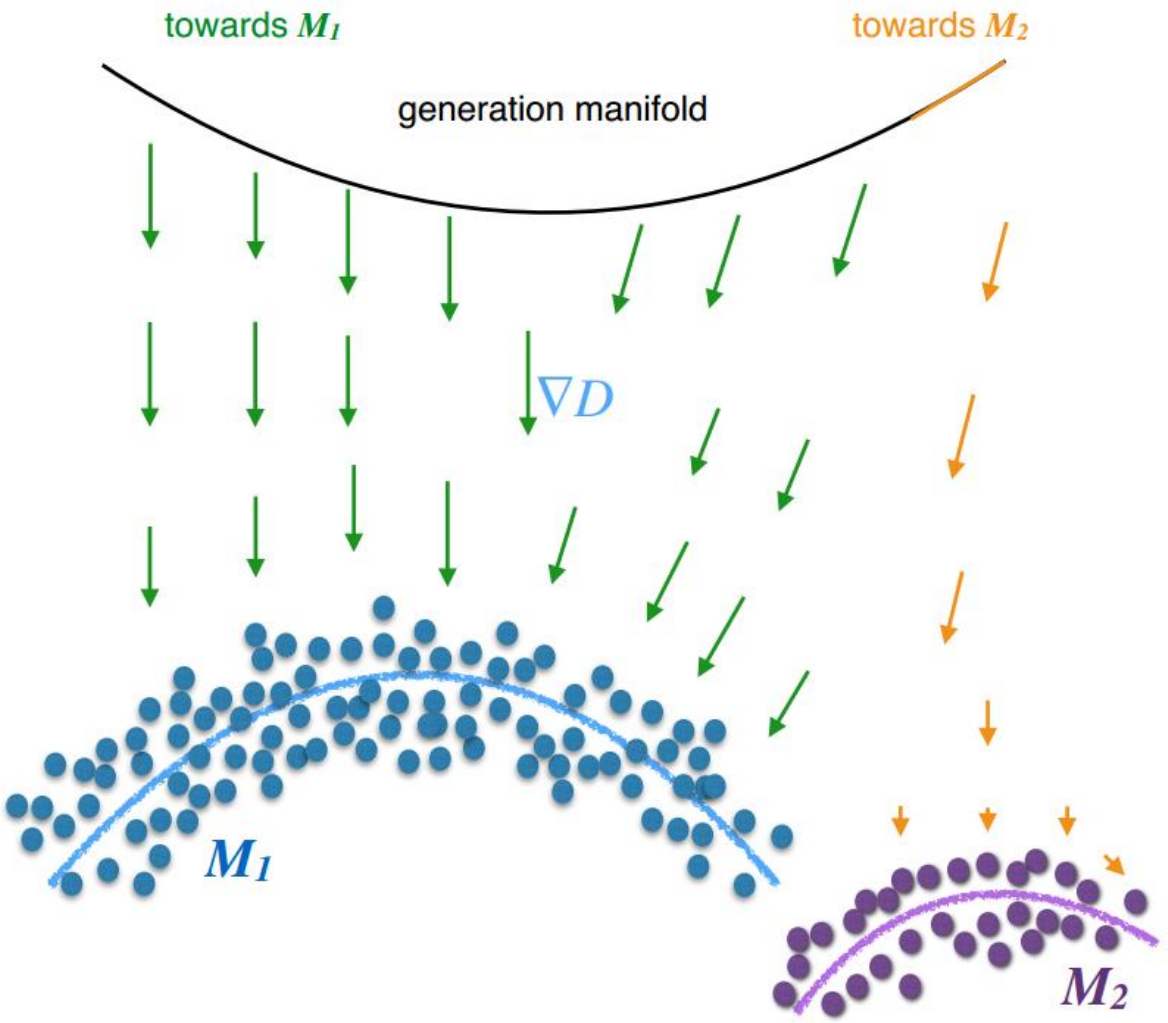
- Minimizing KL divergence only:

$$\text{When } P_g(x) \to 0, \quad P_r(x) \to 1; P_g(x)\log\frac{P_g(x)}{P_r(x)} \to 0, \mathrm{KL}(P_g \,\|\, P_r) \to 0$$

$$\text{When } P_g(x) \to 1, \quad P_r(x) \to 0; P_g(x)\log\frac{P_g(x)}{P_r(x)} \to +\infty, \mathrm{KL}(P_g \,\|\, P_r) \to +\infty$$

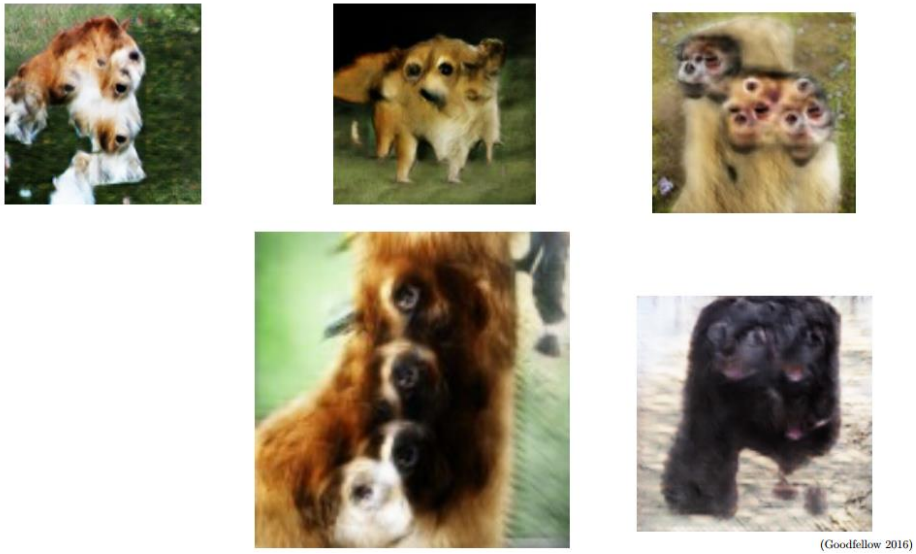


Mode missing



Che *et al.* "Mode Regularized Generative Adversarial Networks". ICLR 2017.

Other problems



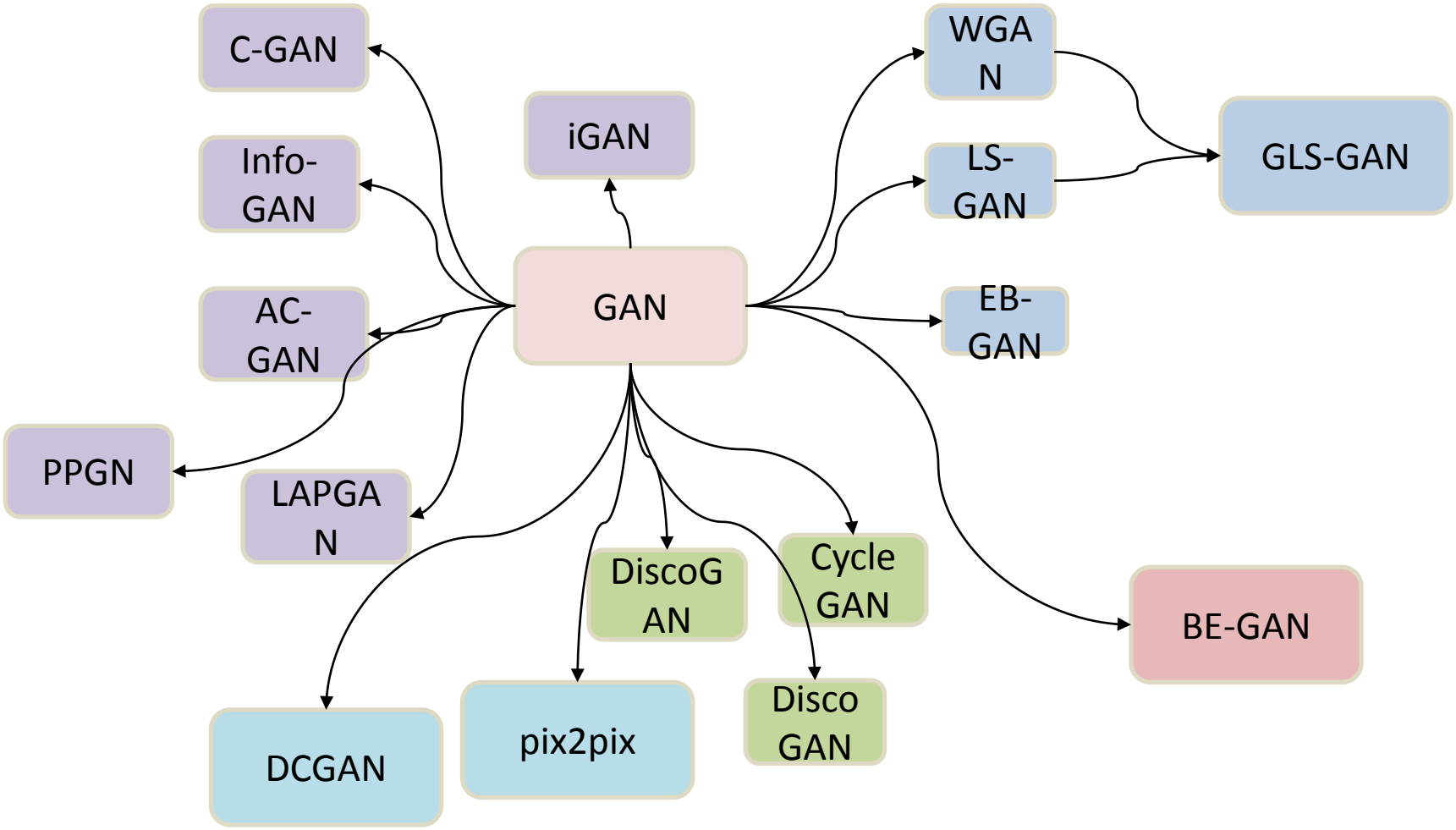
(Goodfellow 2016)

(Goodfellow 2016)

Counting

Global structure

Research

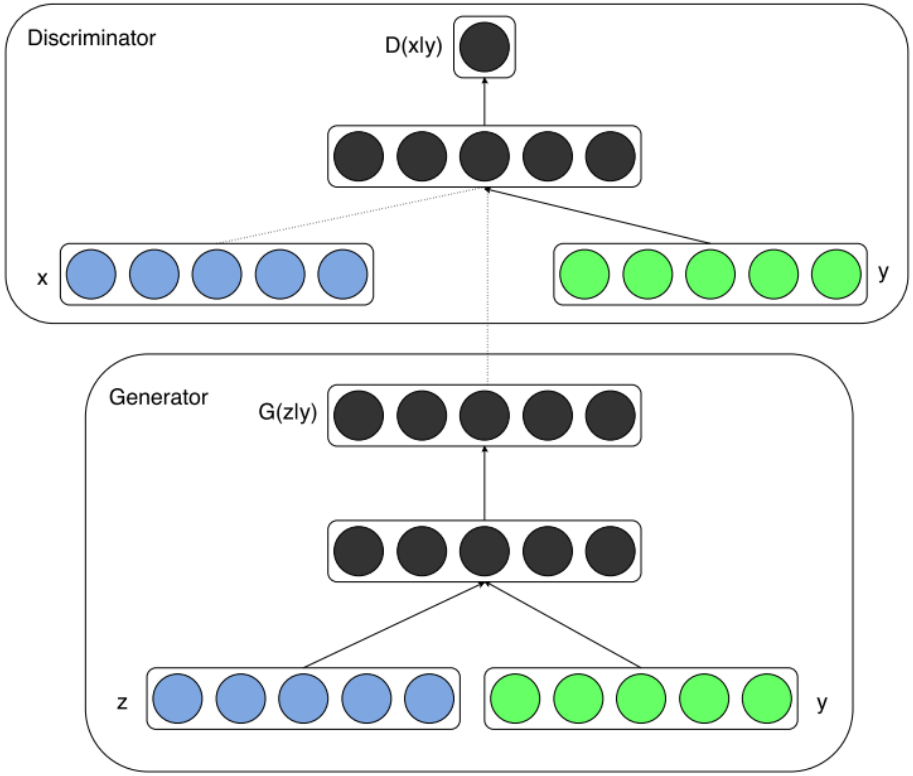


Girshick R, Donahue J, Darrell T, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation。In，ICCV，2015.

Add More infomation

- CGAN
- InfoGAN
- iGAN

Conditional GAN



$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[\log D(x \mid y)] + E_{x\sim p_z(x)}[\log(1 - D(G(z \mid y)))]$$

Mirza *et al.* "Conditional Generative Adversarial Nets". arXiv 2014.

InfoGAN

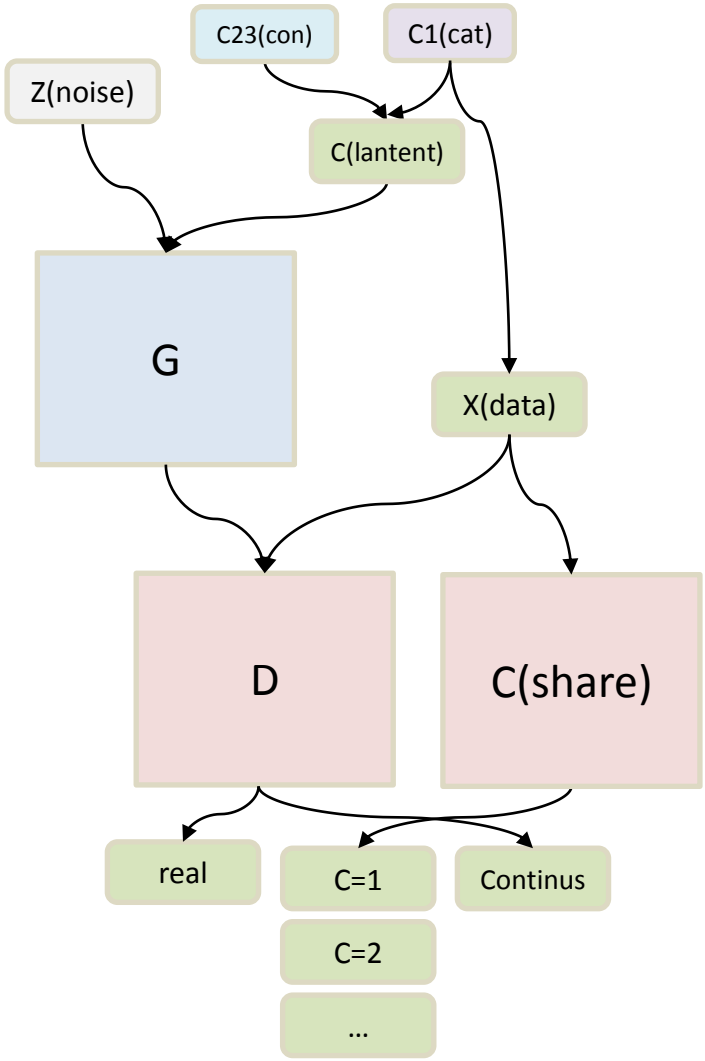
- Information theory

Mutual information:

$$I(X;Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid \mathrm{X})$$

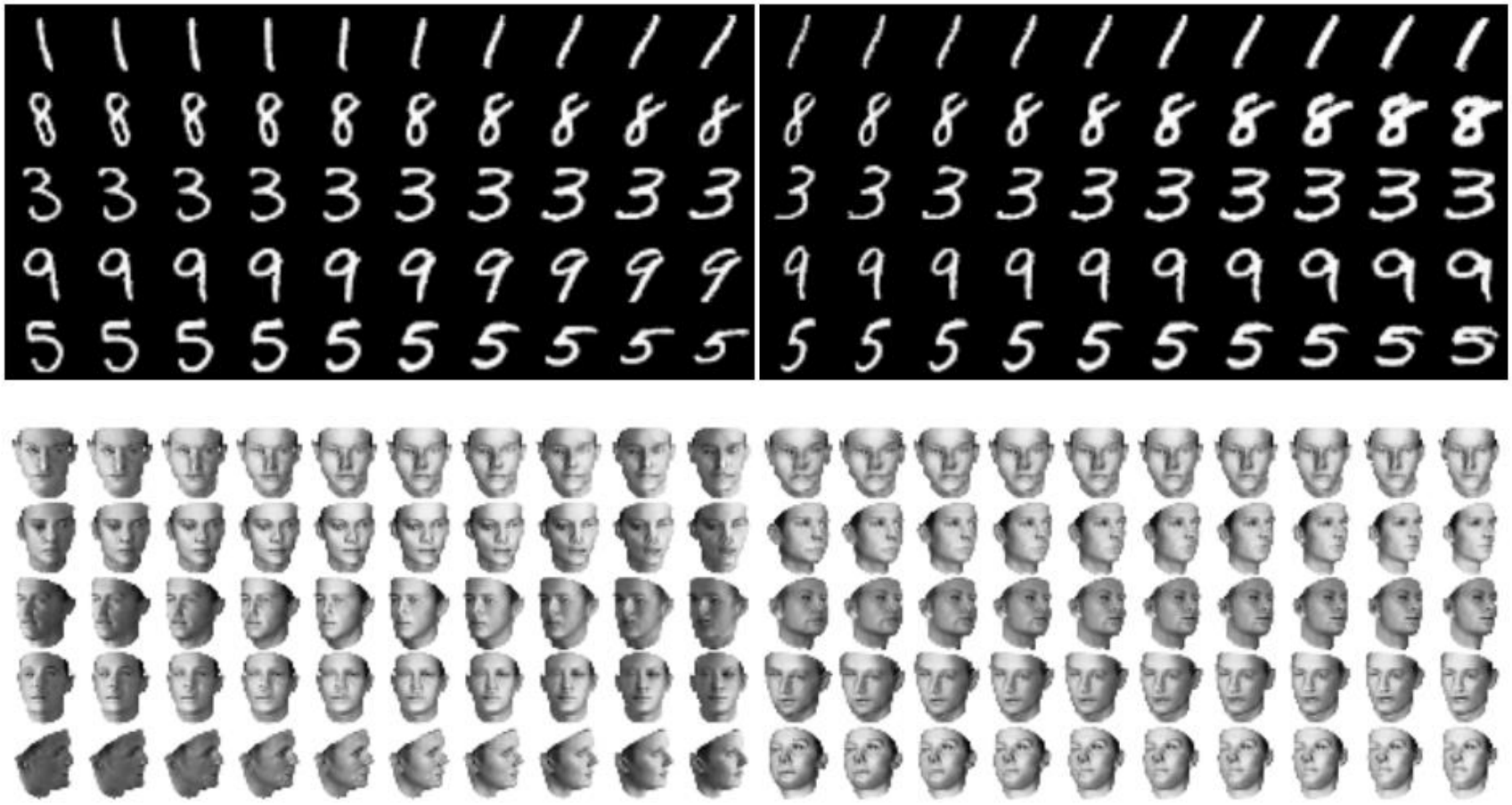
Objective function:

$$\min_G \max_D V_1(D,G) = V(D,G) - \lambda I(c;G(z,c))$$



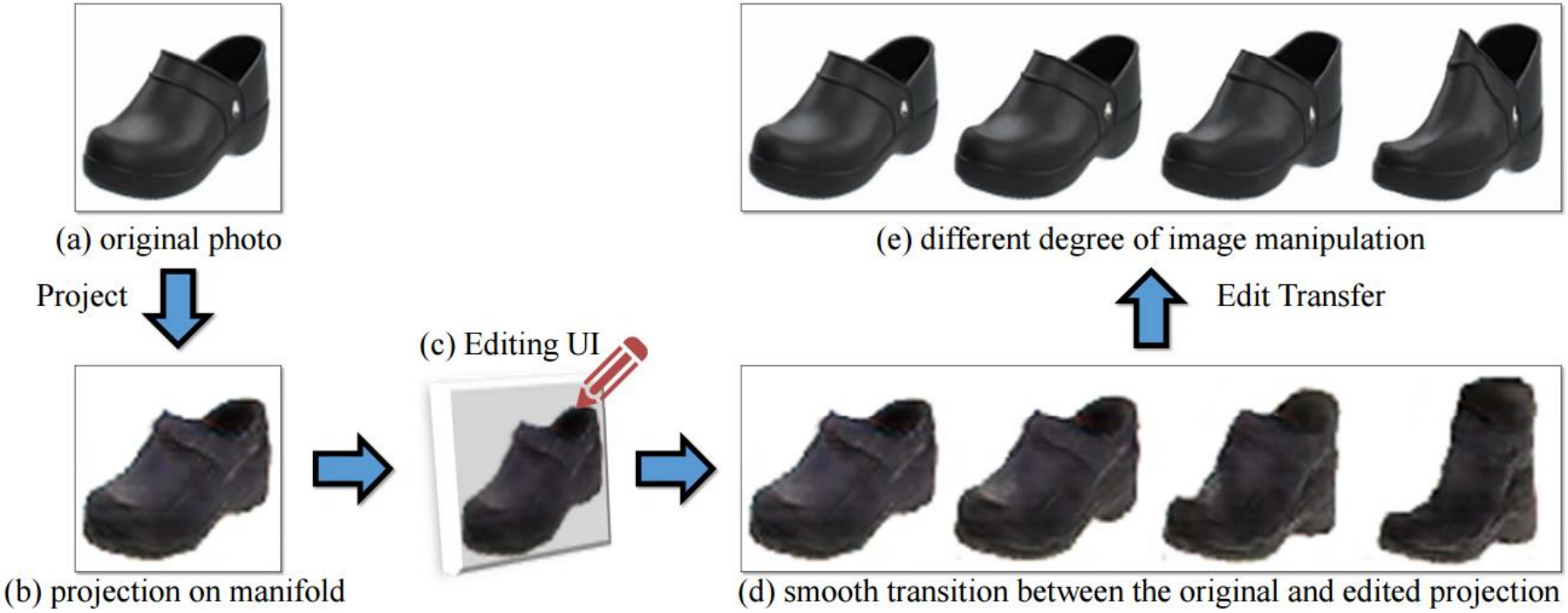
Chen *et al.* "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets". arXiv 2016.

InfoGAN



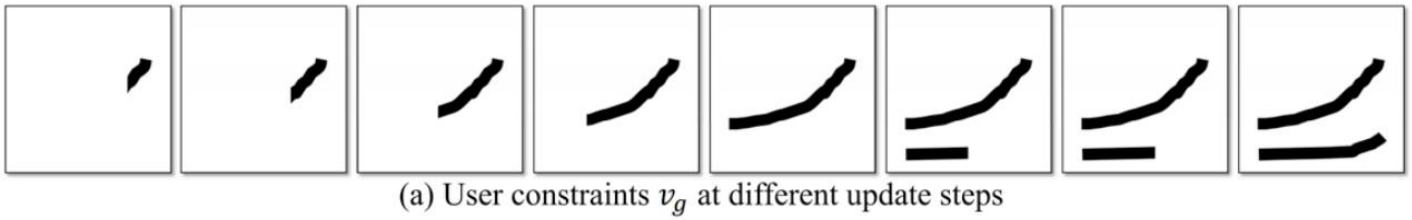
Chen *et al.* "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets". arXiv 2016.

iGAN



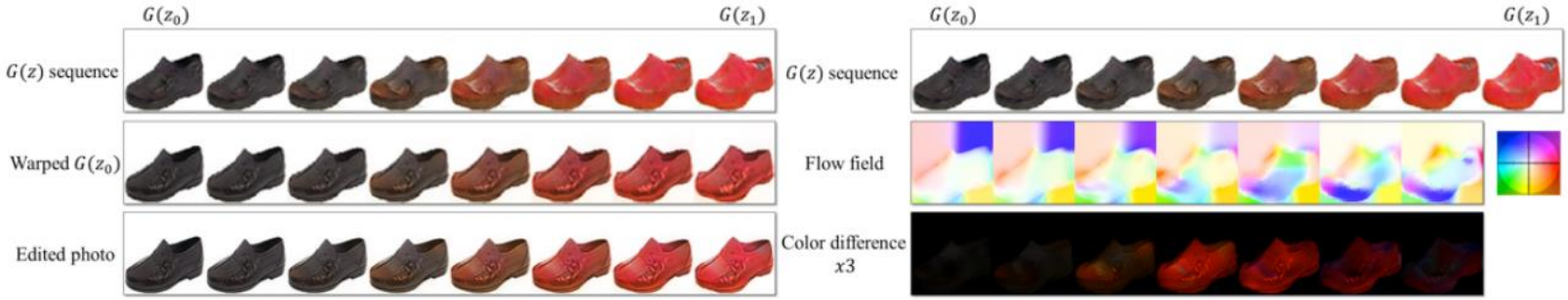
Zhu *et al.* "Generative Visual Manipulation on the Natural Image Manifold". ECCV 2016.

iGAN



(a) User constraints v_g at different update steps

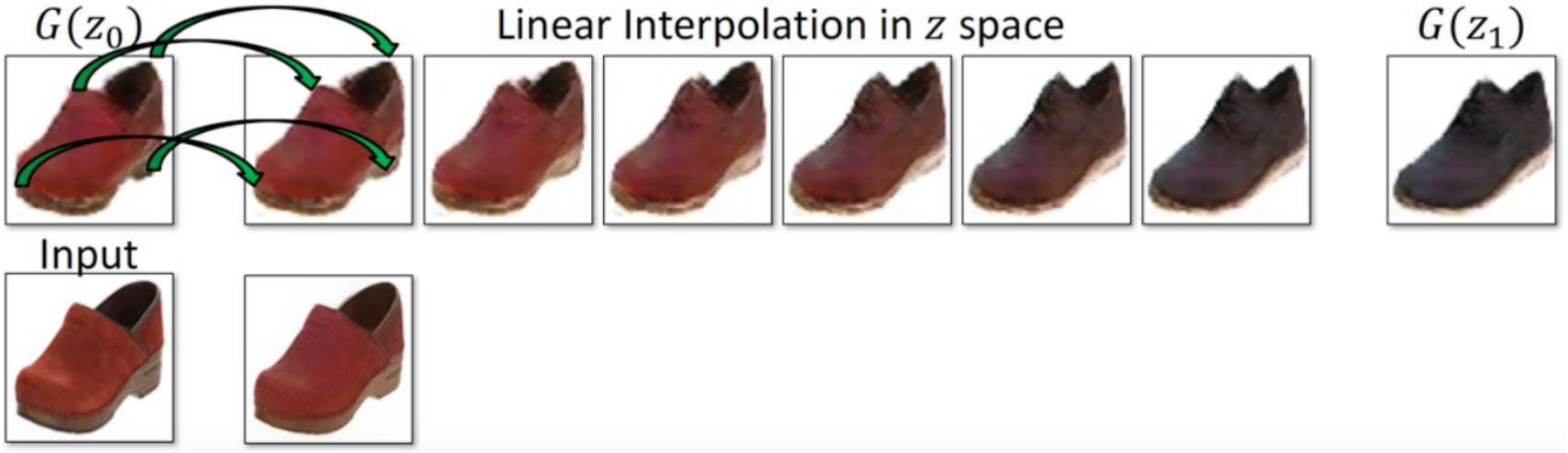
$G(z_0)$ (b) Updated images according to user edits $G(z_1)$



Zhu *et al.* "Generative Visual Manipulation on the Natural Image Manifold". ECCV 2016.

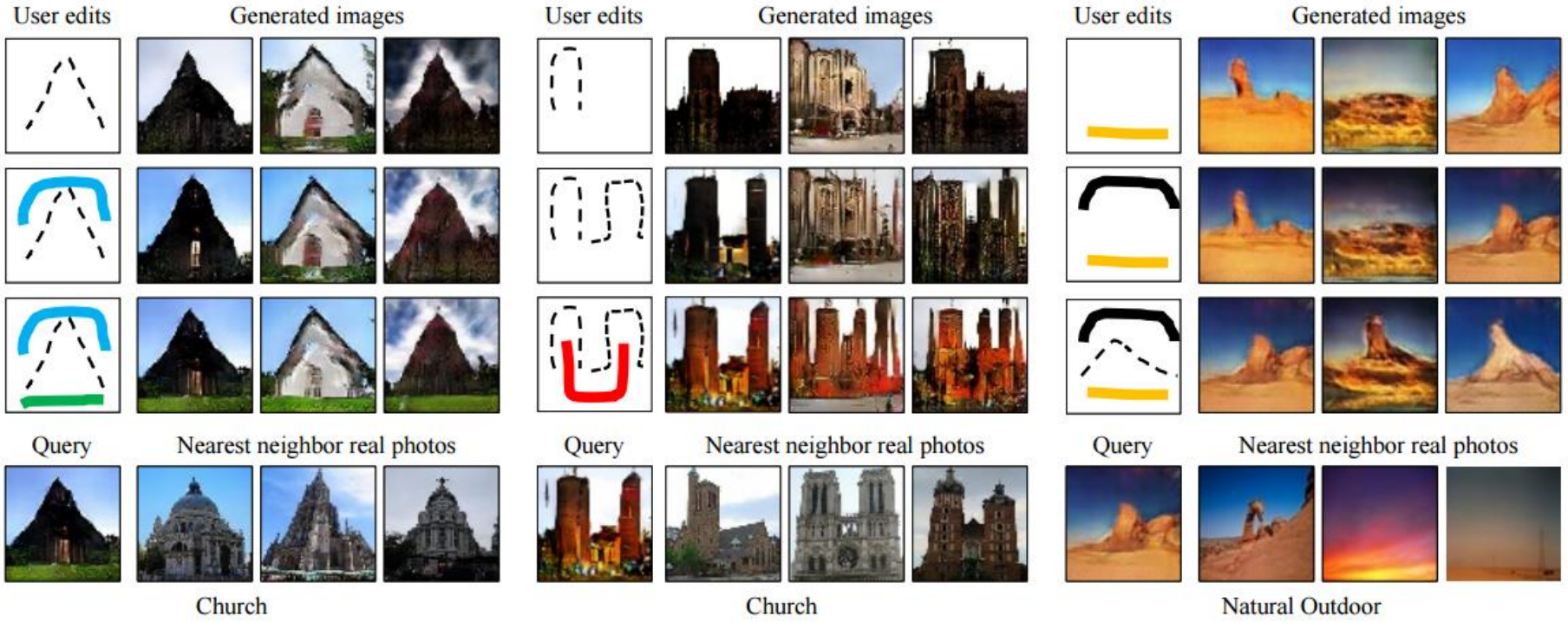
iGAN

$$\iint \underbrace{\|I(x,y,t) - A \cdot I(x+u, y+v, t+1)\|^2}_{\text{data term}} + \underbrace{\sigma_s(\|\nabla u\|^2 + \|\nabla v\|^2)}_{\text{spatial reg}} + \underbrace{\sigma_c \|\nabla A\|^2}_{\text{color reg}} dx dy$$



Zhu *et al.* "Generative Visual Manipulation on the Natural Image Manifold". ECCV 2016.

iGAN

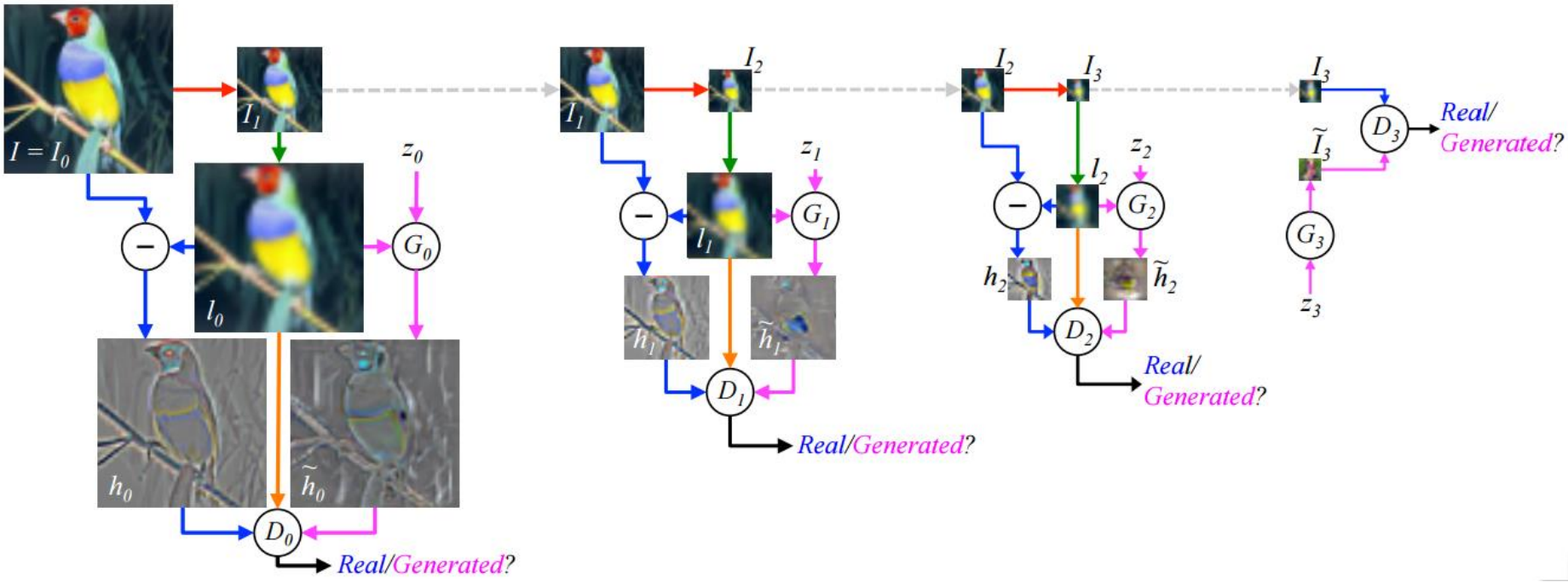


Zhu *et al.* "Generative Visual Manipulation on the Natural Image Manifold". ECCV 2016.

Fine-grained Guidance

- LAPGAN
- Matching-aware Discriminator
- PPGN

LAPGAN



$$\min_G \max_D \mathrm{E}_{\mathrm{h,l}\sim p_d(h,l)}[\log D(h,l)] + \mathrm{E}_{\mathrm{z}\sim p_{noise}(z),l\sim p_l(l)}[\log(1-D(G(z,l),l))]$$

Denton *et al.* "Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks". NIPS 2015.

Text to image

- Matching-aware Discriminator
  - Implicitly separate two source of error:unrealistic
  - Images, and realistic images of the wrong class that mismatch the conditioning

- $\mathrm{x} \leftarrow G(z, h)$ {Forward through generator}
- $s_r \leftarrow D(x, h)$ {right image, right text}
- $s_w \leftarrow D(x, h)$ {real image, wrong text}
- $s_f \leftarrow D(x, h)$ {fake image, right text}

$$L_D \leftarrow \log(s_r) + (\log(1 - s_w) + \log(1 - s_f)) / 2$$

Scott *et al.* "Generative Adversarial Text to Image Synthesis". ICML 2016.

Text to image

Text descriptions (content)

Images (style)

The bird has a **yellow breast** with **grey** features and a small beak.

This is a large **white** bird with **black wings** and a **red head**.

A small bird with a **black head and wings** and features grey wings.

This bird has a **white breast**, brown and white coloring on its head and wings, and a thin pointy beak.

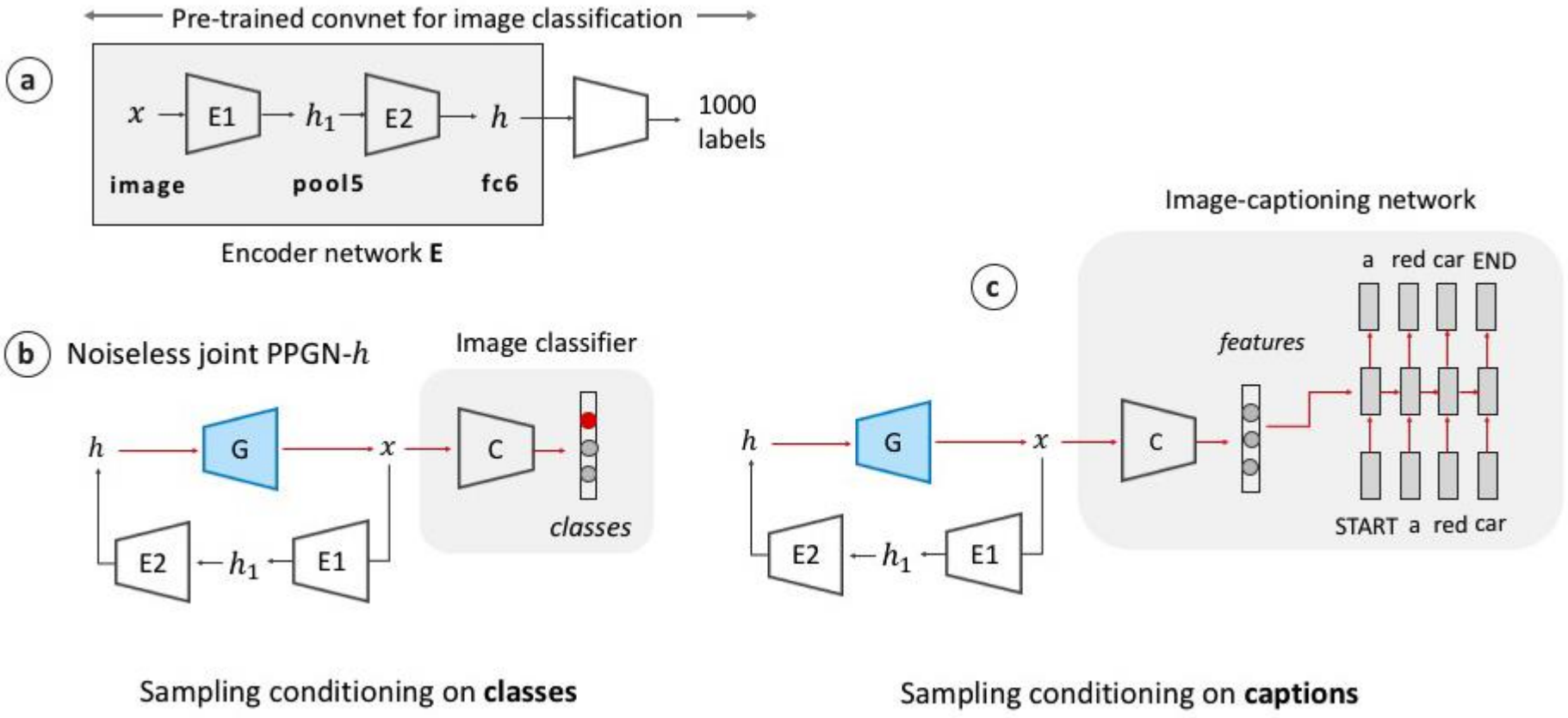
A small bird with **white base** and **black stripes** throughout its belly, head, and feathers.

Scott *et al.* "Generative Adversarial Text to Image Synthesis". ICML 2016.

PPGN

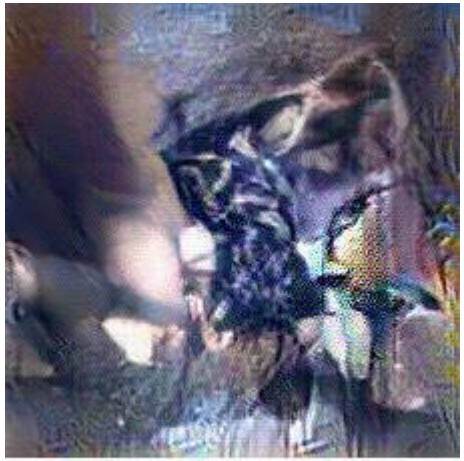
Nguyen *et al.* "Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space". CVPR 2017.

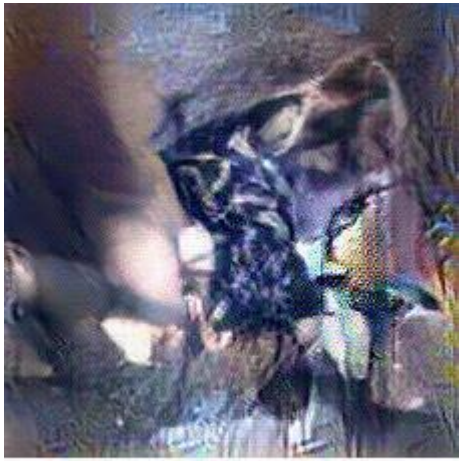
PPGN

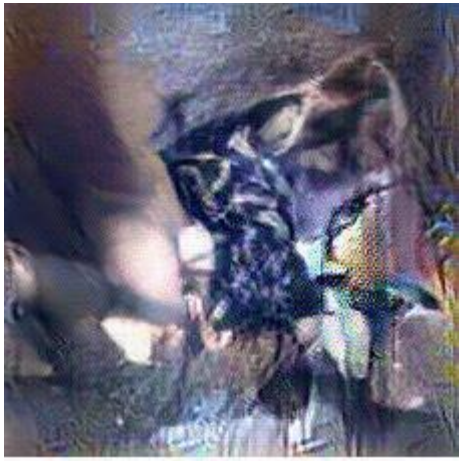


Nguyen *et al.* "Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space". CVPR 2017.

PPGN



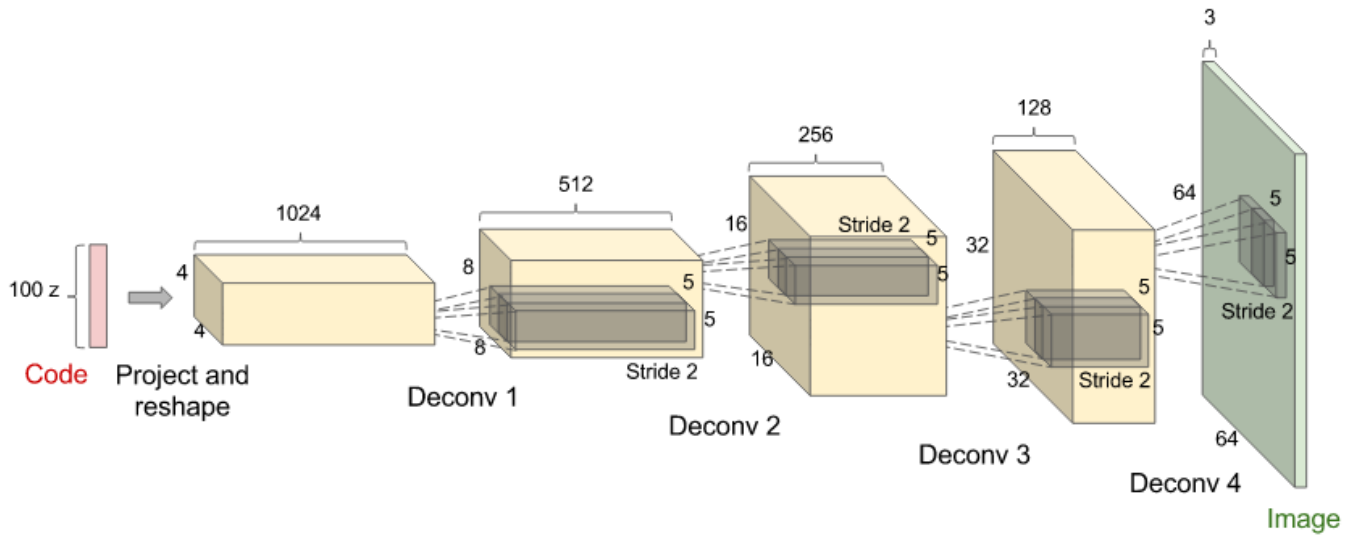


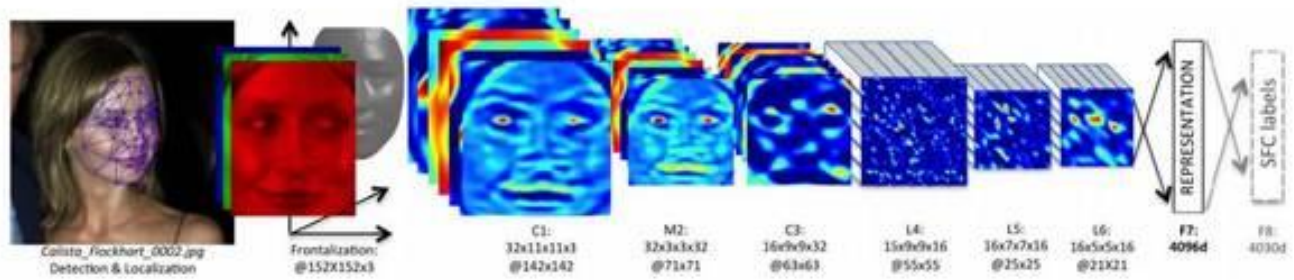


Nguyen *et al.* "Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space". CVPR 2017.

Architectures

DCGAN&tricks

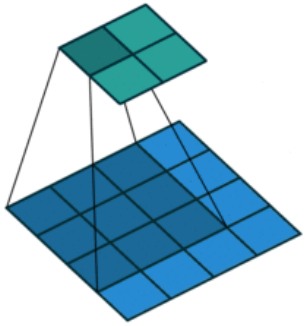




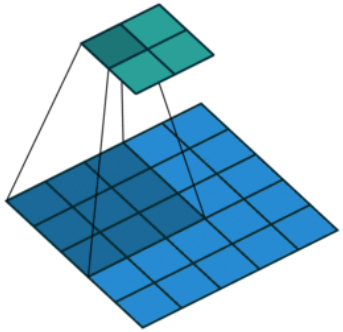
[Taigman et al. 2014]

Radford *et al.* "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". ICLR 2016.

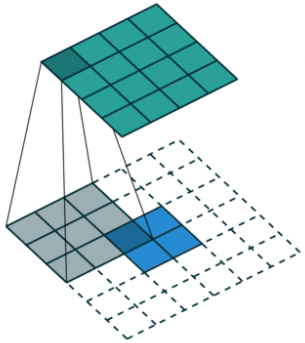
Unstranding ‘Deconvolution’



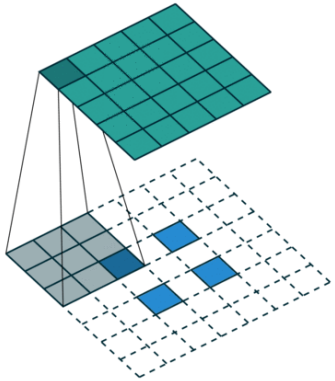
No padding, no stride



No padding, stride



No padding, stride transposed



No padding, stride transposed

In details

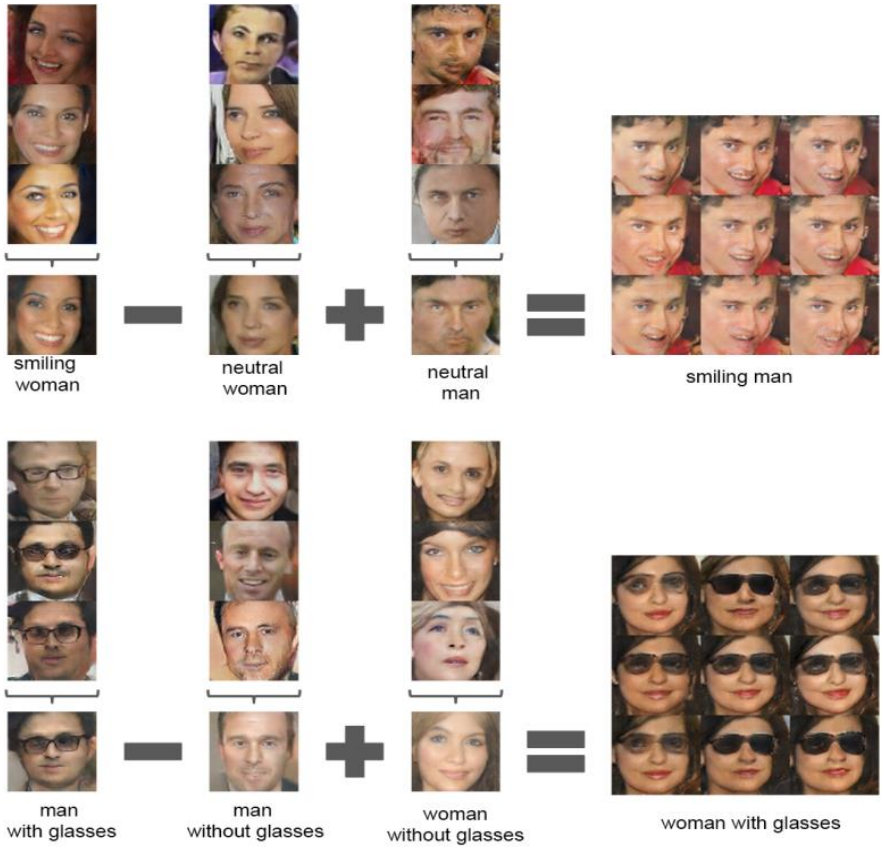
- For stable Deep Convolutional GANs
  - Replace any pooling with Convolutions (strided; fractional-strided)
  - Use batchnorm
  - Remove fully connected hidden layers
  - Use ReLu activation in G
  - Use leakyReLu activation in D

Radford *et al.* "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". ICLR 2016.

Results of DCGAN

On Lsun bedroom

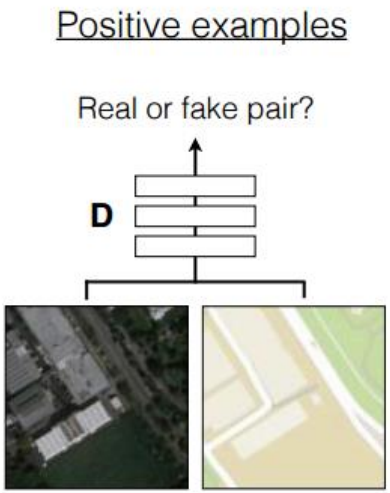
Results of DCGAN

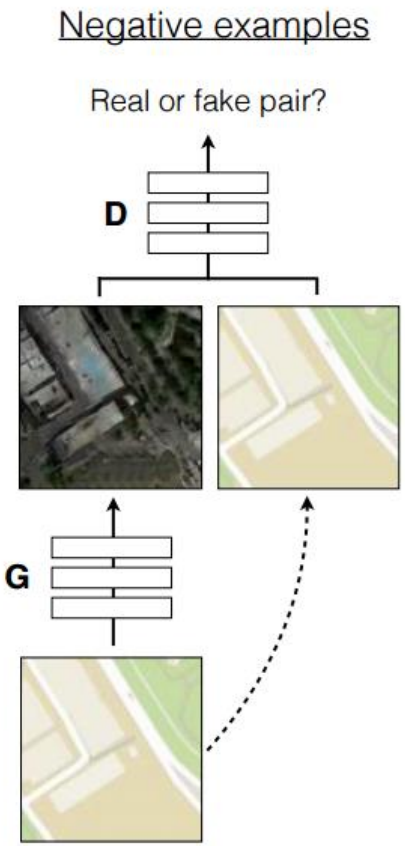


Face editing

Radford *et al.* "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". ICLR 2016.

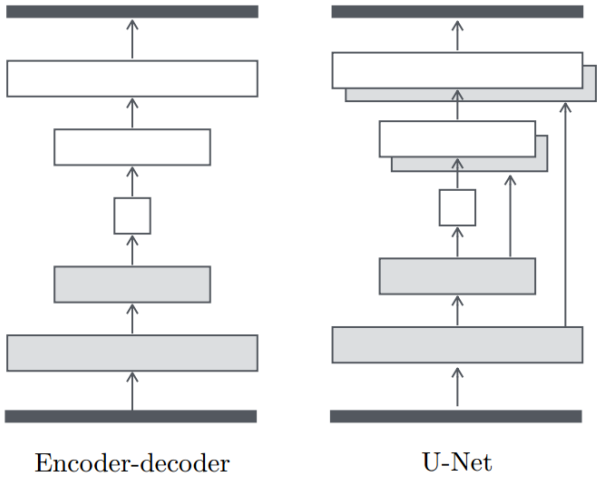
Pix2pix





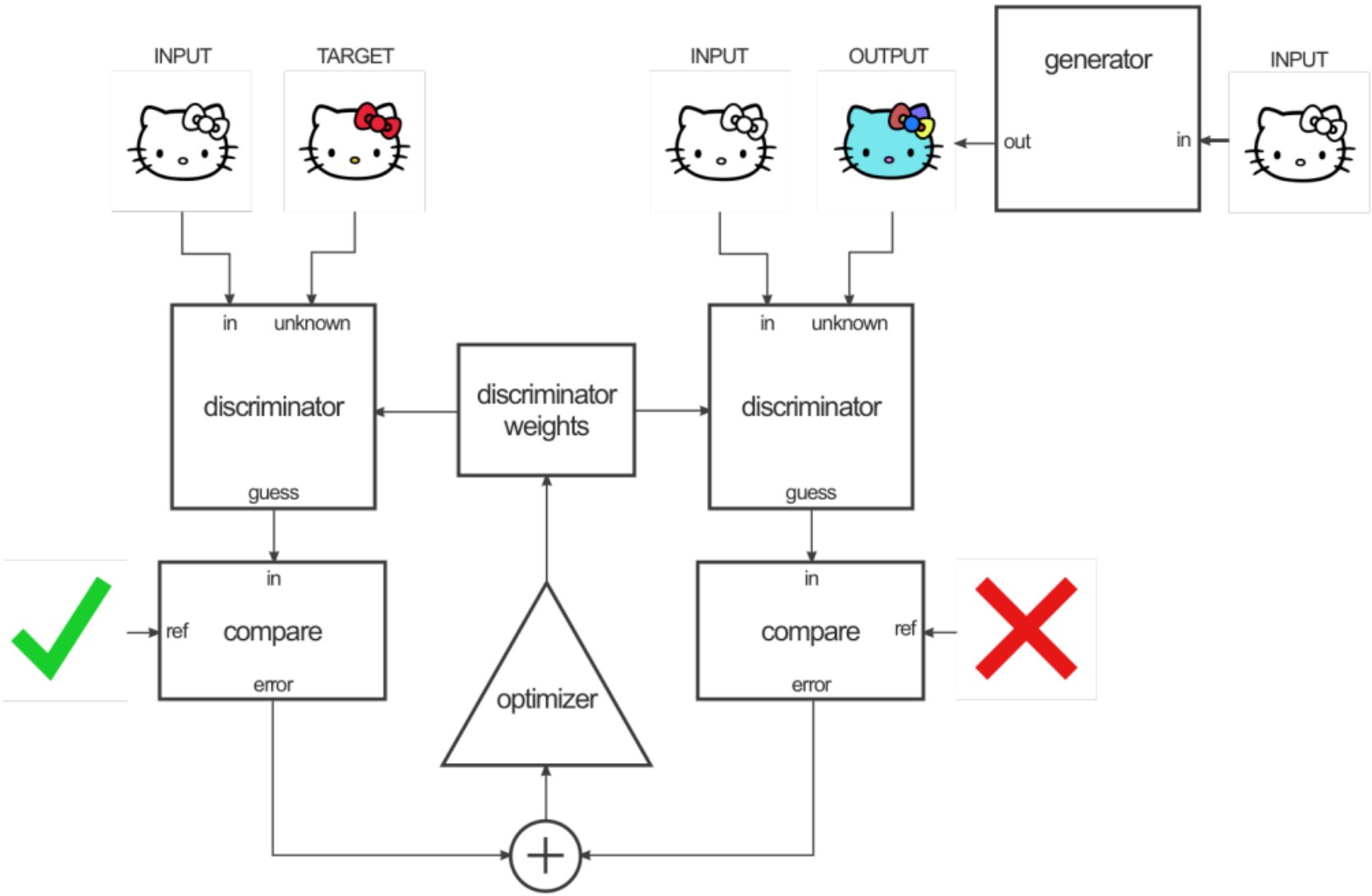
G tries to synthesize fake images that fool **D**

D tries to identify the fakes

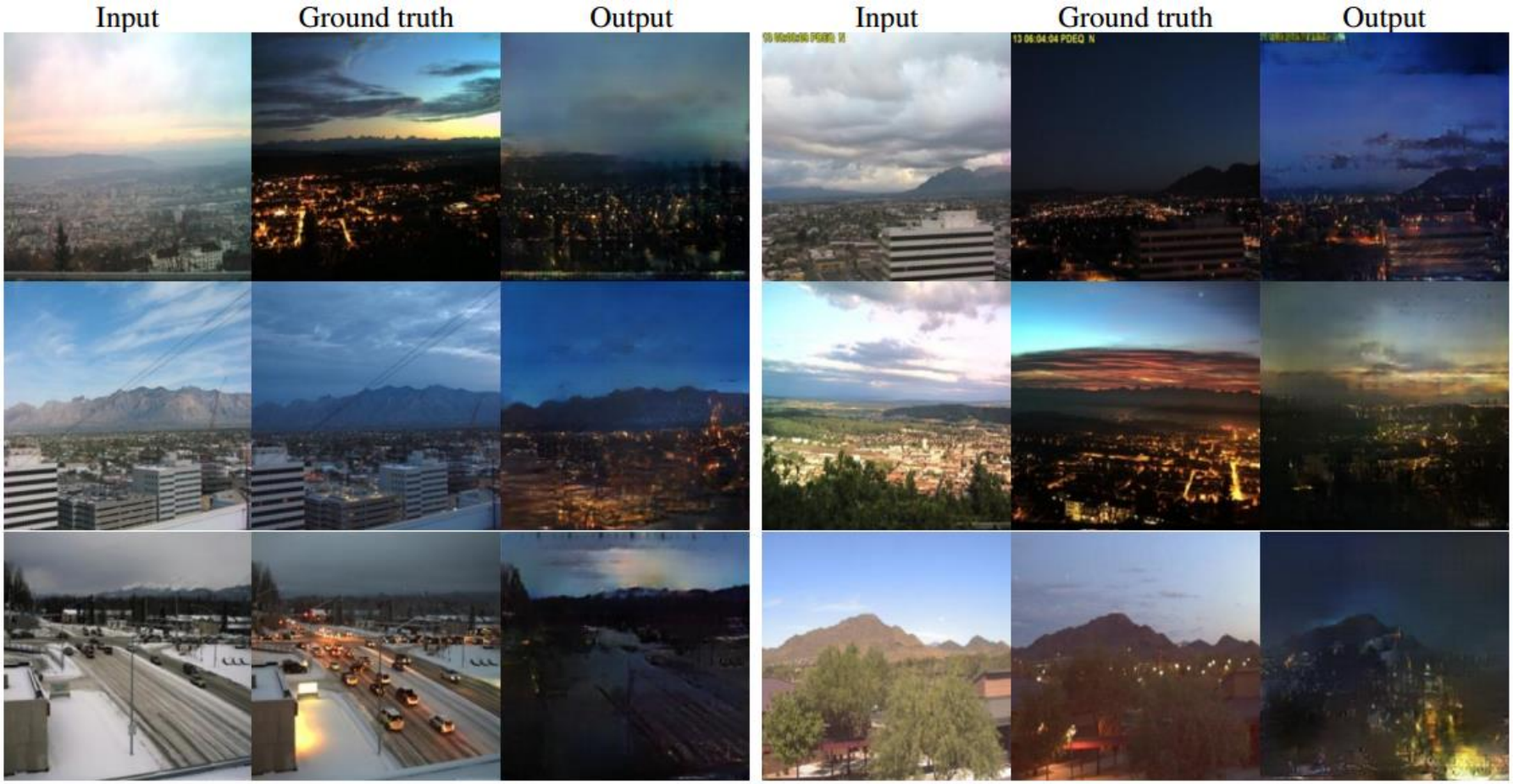


Isola *et al.* "Image-to-Image Translation with Conditional Adversarial Networks". CVPR 2017.

Pix2pix



Pix2pix



Isola *et al.* "Image-to-Image Translation with Conditional Adversarial Networks". CVPR 2017.

Pix2pix

Isola *et al.* "Image-to-Image Translation with Conditional Adversarial Networks". CVPR 2017.

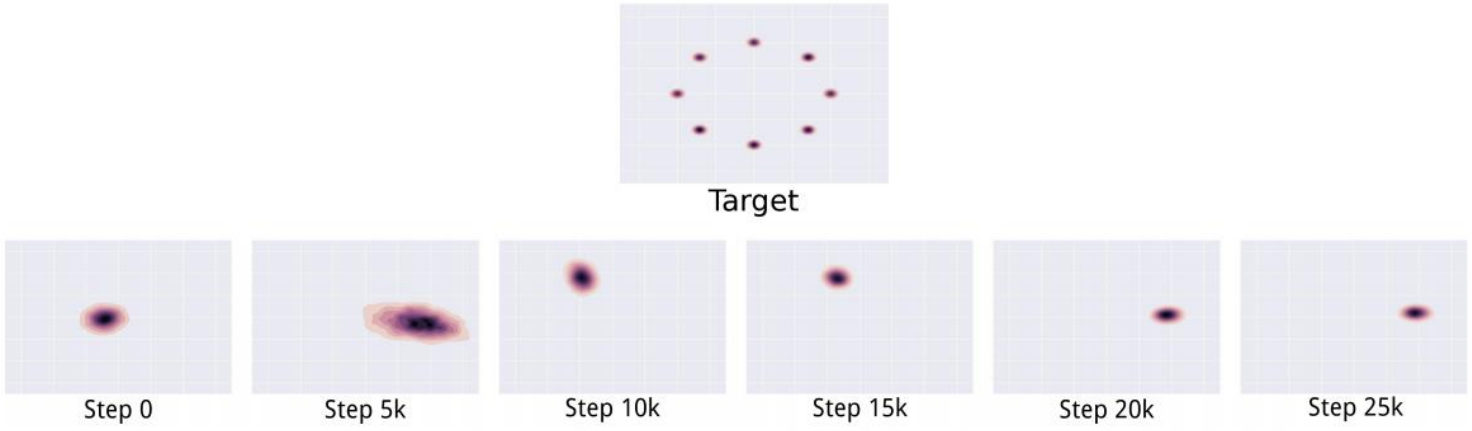
Encode effectives

- MRGAN
- EBGAN
- BEGAN

Review mode collapse

$$\min_G \max_D V(G,D) \neq \max_D \min_G V(G,D)$$

- D in inner loop: convergence to correct distribution
- G in inner loop: place all mass on most likely point



Mode Regularized GANs

- Regularized GANs
  - encoder E:

$$\mathrm{E}_{\mathrm{x}\sim p_d}[\lambda_1 d(x, G\circ E(x)) + \lambda_2 \log D(G\circ E(x))]$$

  - generator G:

$$-\mathrm{E}_z[\log D(G(z))] + \mathrm{E}_{\mathrm{x}\sim p_d}[\lambda_1 d(x, G\circ E(x)) + \lambda_2 \log D(G\circ E(x))]$$

  - discriminator D： same as vanilla GAN

- Still Suffer from gradient vanishing

Che *et al.* "Mode Regularized Generative Adversarial Networks". ICLR 2017.

Mode Regularized GANs

Manifold-Diffusion GANs:

– Manifold-step:

- Match generation manifold and real data manifold

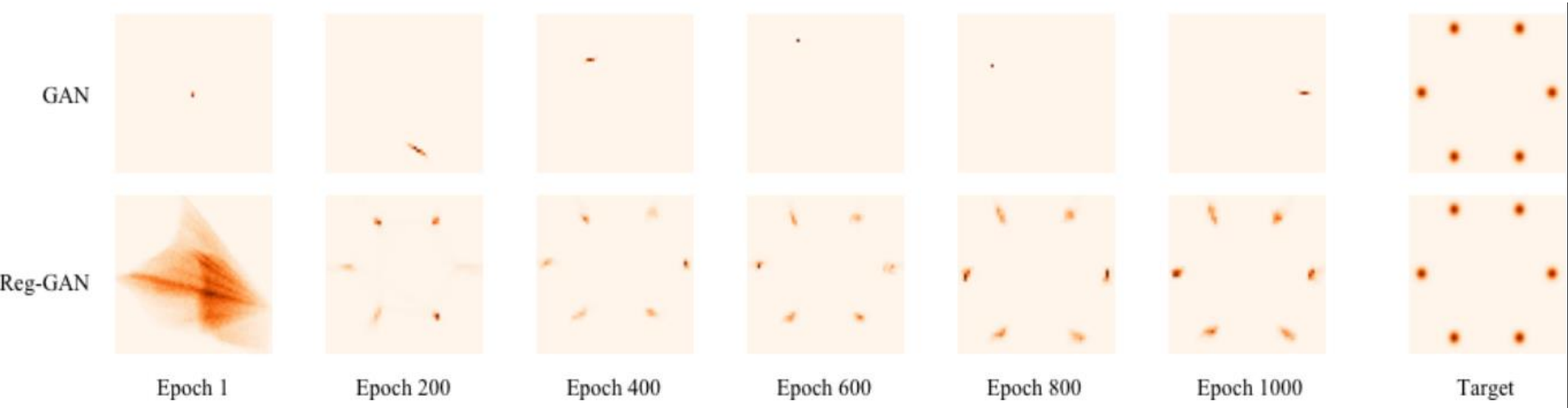
– Diffusion step：

- Distribute the probability mass on the generation manifold fairly

- D: firstly comparing between real data and the encoded data

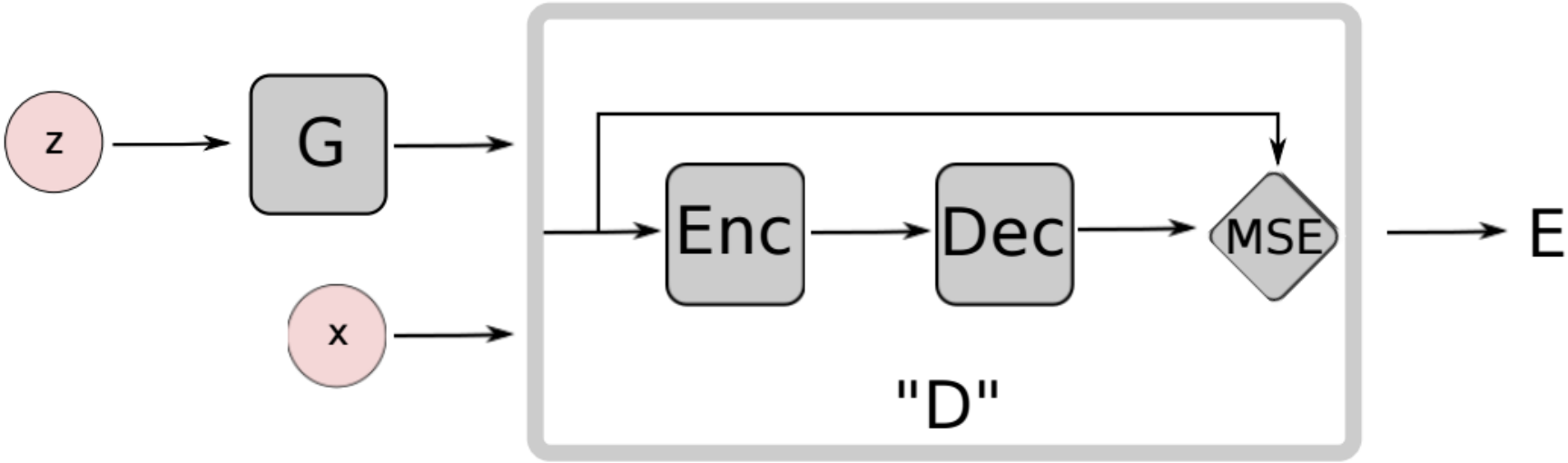
Che *et al.* "Mode Regularized Generative Adversarial Networks". ICLR 2017.

Mode Regularized GANs



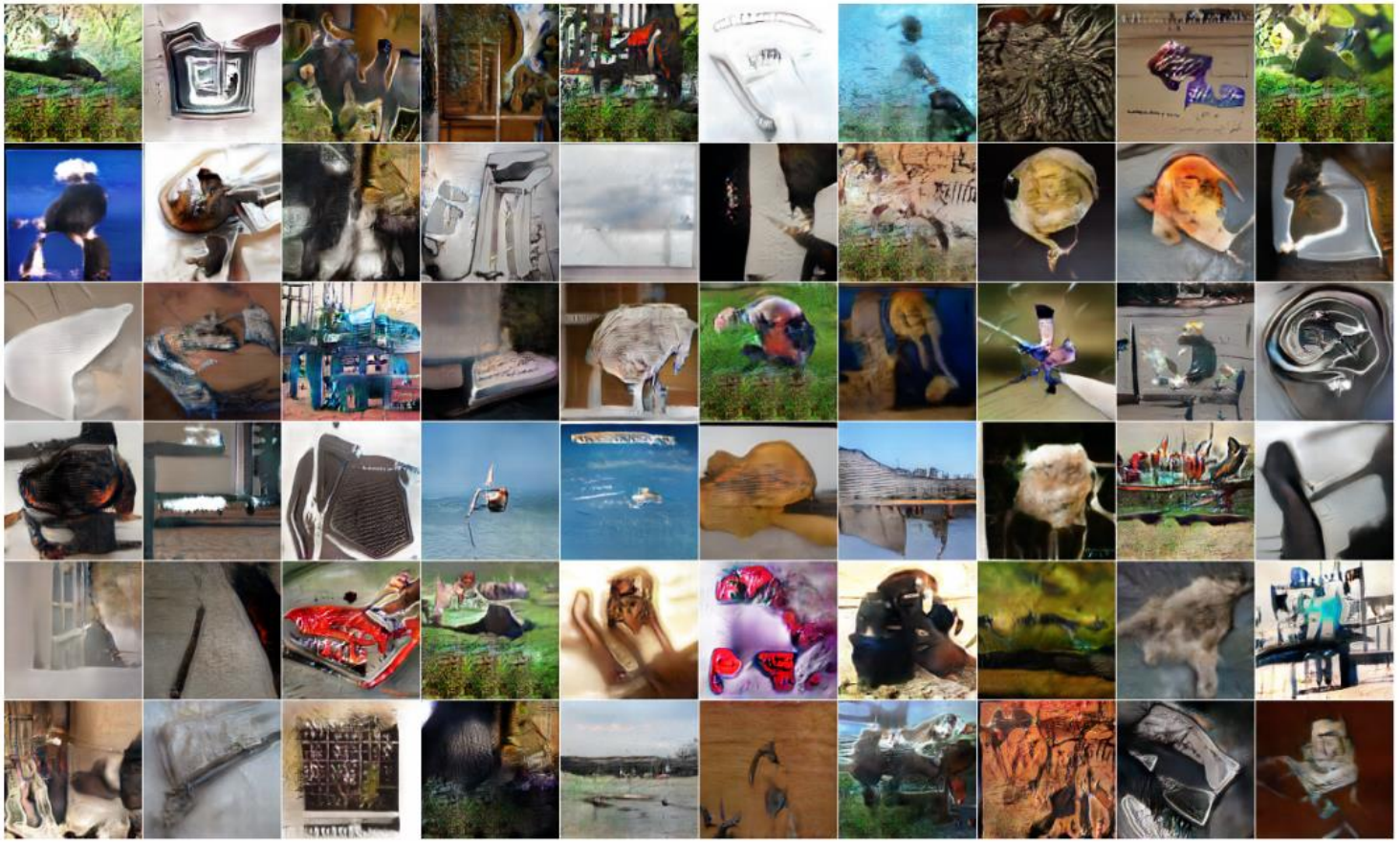
Che *et al.* "Mode Regularized Generative Adversarial Networks". ICLR 2017.

EBGAN



Zhao *et al.* "Energy-based Generative Adversarial Network". ICLR 2017.

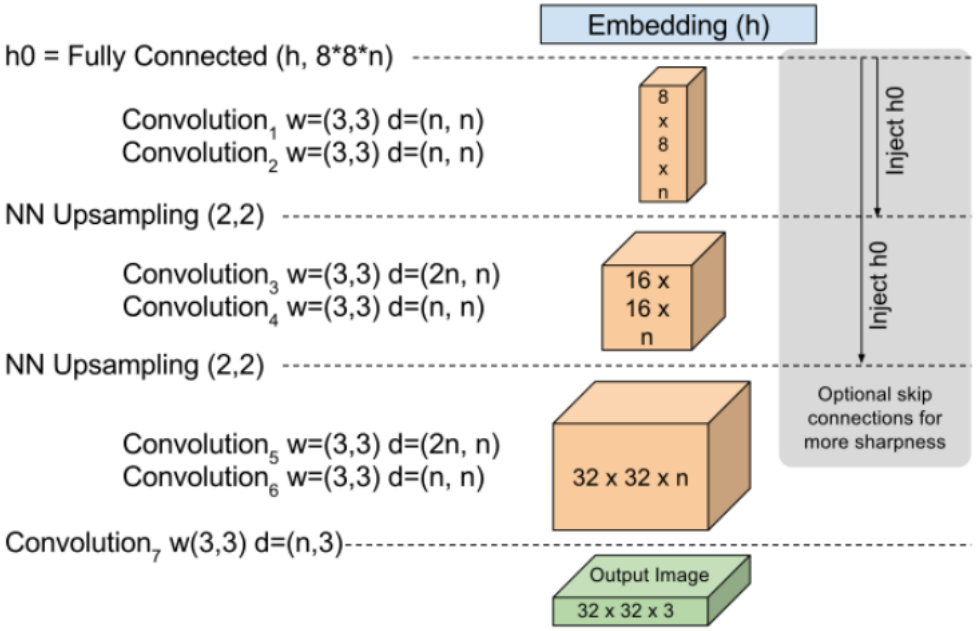
EBGAN



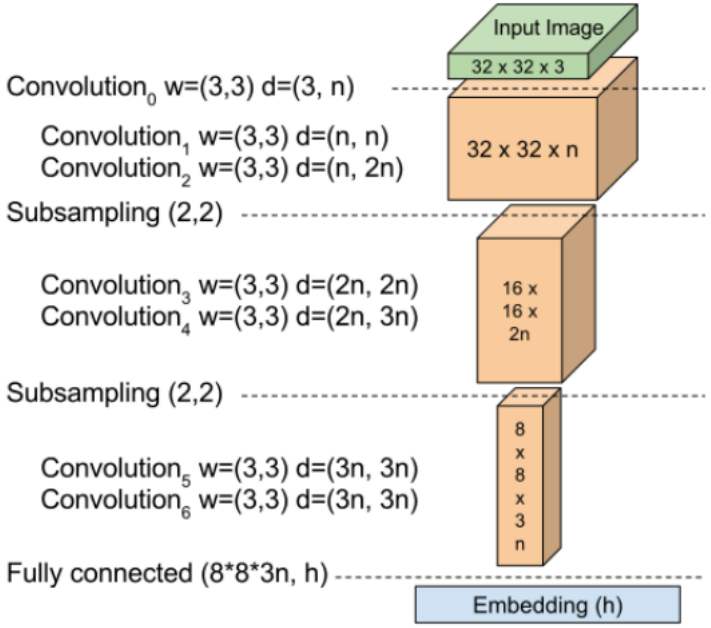
On ImageNet

Zhao *et al.* "Energy-based Generative Adversarial Network". ICLR 2017.

BEGAN



(a) Generator/Decoder



(b) Encoder

Berthelot *et al.* "BEGAN: Boundary Equilibrium Generative Adversarial Networks". arXiv 2017.

BEGAN

(a) EBGAN (64x64) (b) Our results (128x128)

Figure 2: Random samples comparison

Figure 3: Random 64x64 samples at varying diversities $\gamma \in \{0.3, 0.5, 0.7\}$

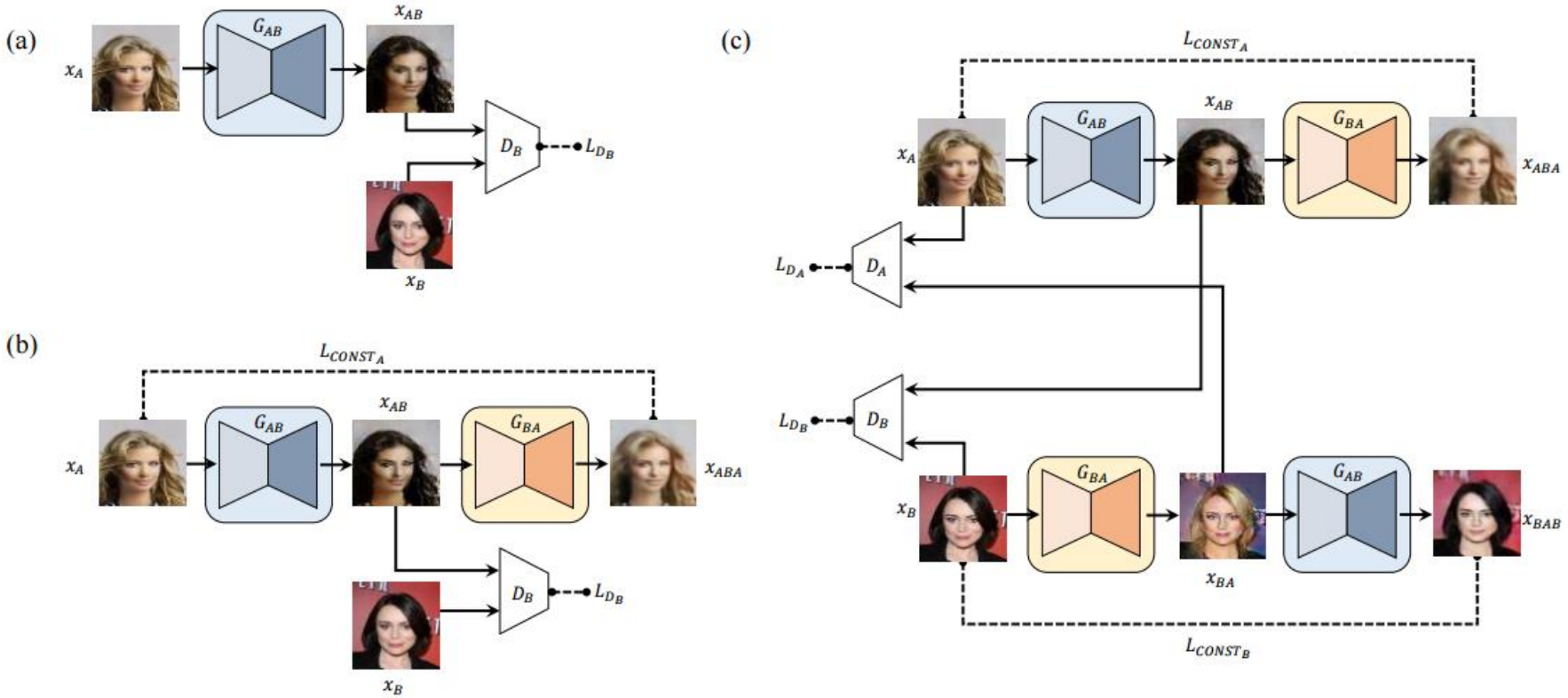
BEGAN

Berthelot *et al.* "BEGAN: Boundary Equilibrium Generative Adversarial Networks". arXiv 2017.

GAN “三兄弟”

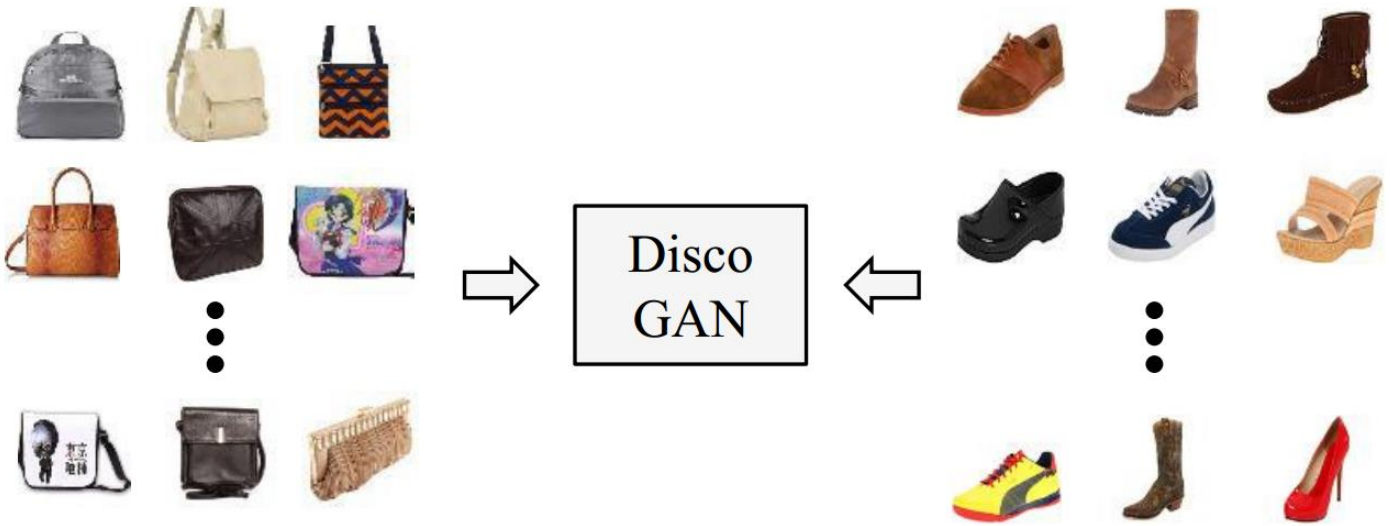
- CycleGAN
- DiscoGAN
- DualGAN

DiscoGAN



Learning to discover cross-domain relations

Kim *et al.* *"Learning to Discover Cross-Domain Relations with Generative Adversarial Networks"*. ICML 2017.

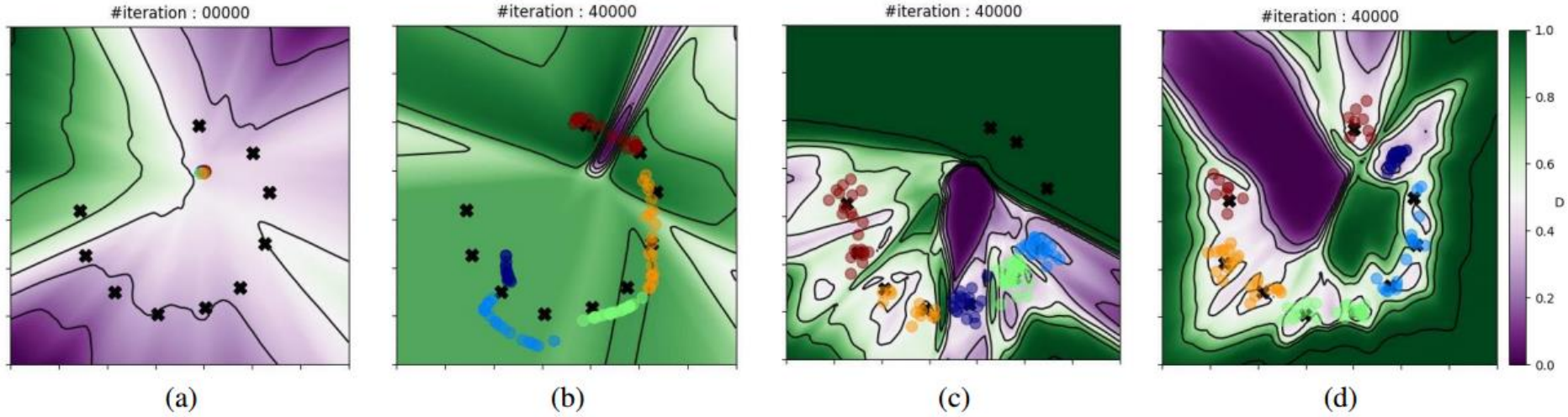


(a) Learning cross-domain relations **without any extra label**

(b) Handbag images (input) & **Generated** shoe images (output)

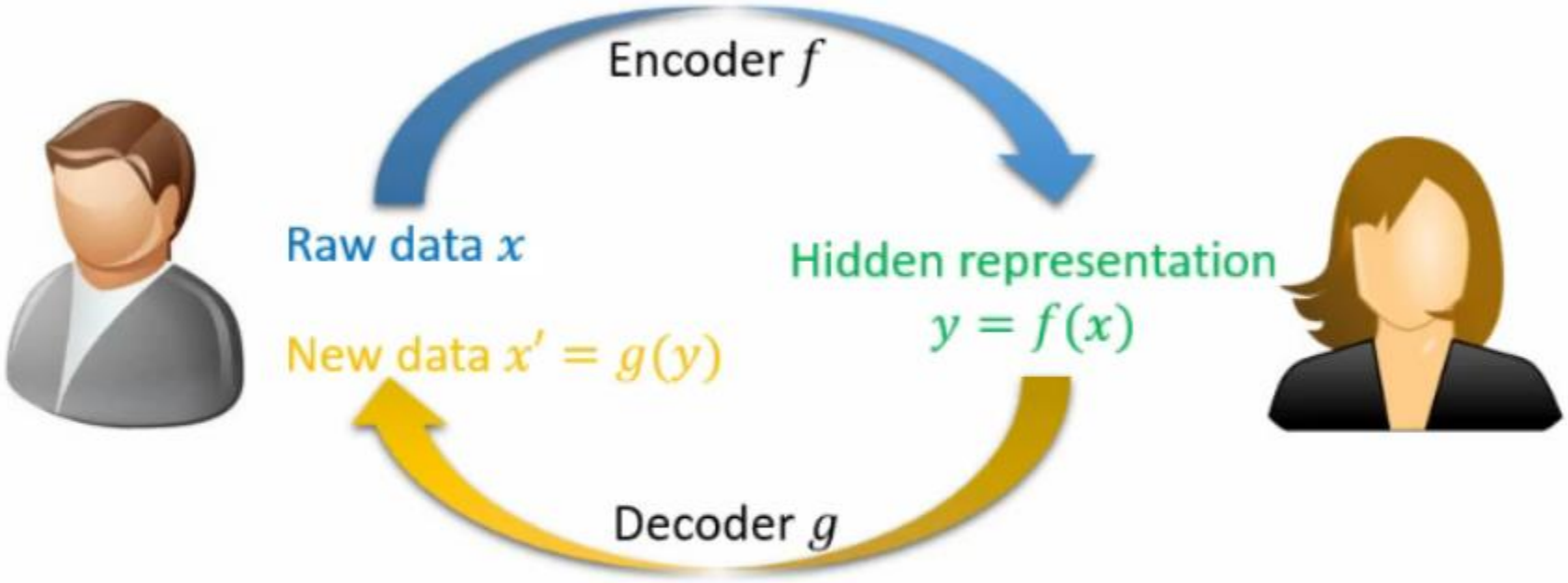
Kim *et al.* *"Learning to Discover Cross-Domain Relations with Generative Adversarial Networks"*. ICML 2017.

Mode missing



Kim *et al.* *"Learning to Discover Cross-Domain Relations with Generative Adversarial Networks"*. ICML 2017.

Dual Learning

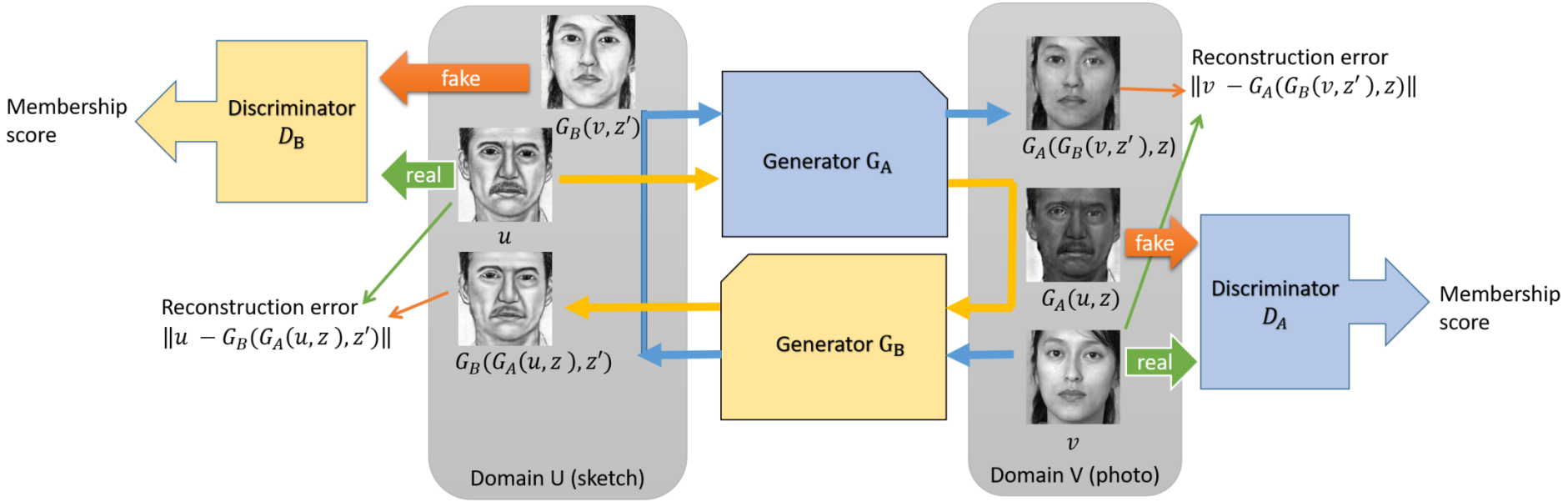


Feedback signals during the loop:

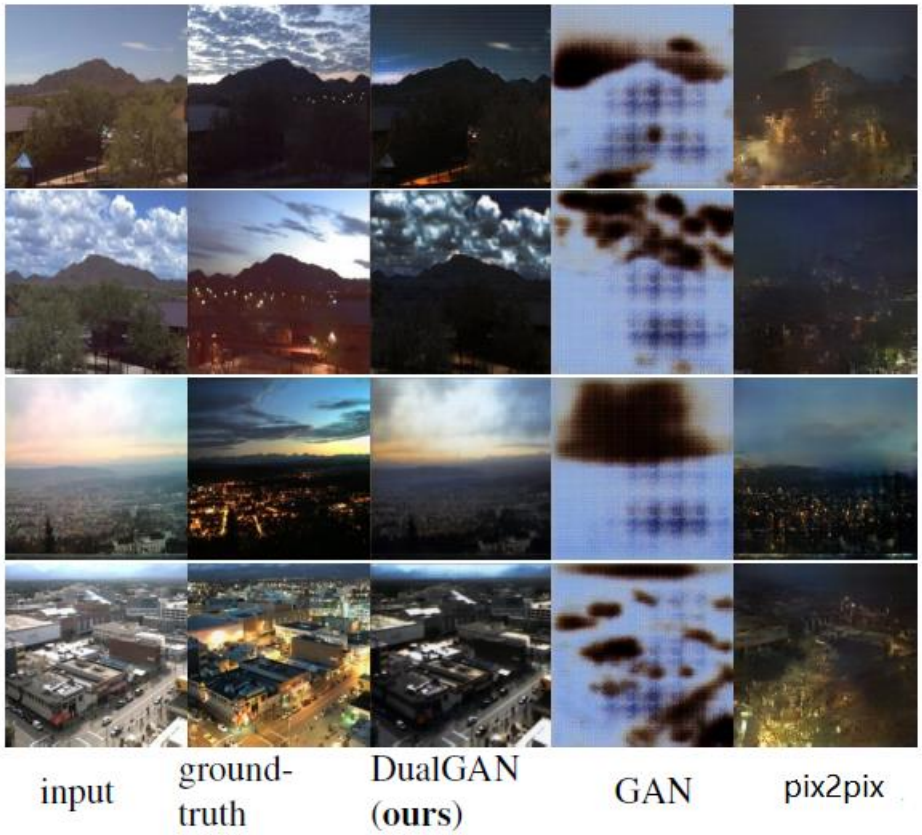
- $R(x, f, g) = s(x, x')$: similarity between x and x'

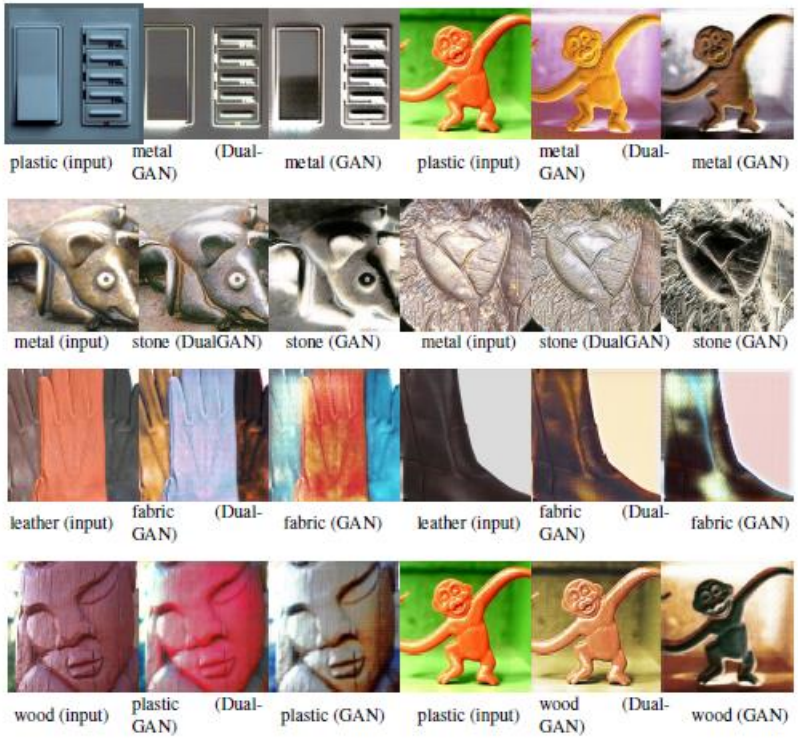
Xia *et al.* "Dual Learning for Machine Translation". ICCV2017.

DualGAN



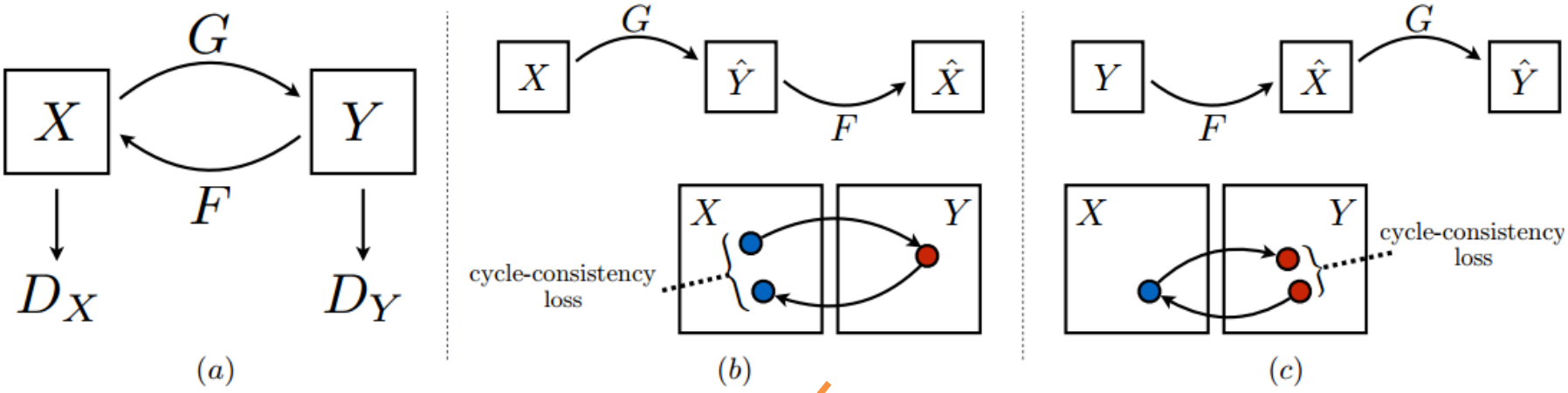
Yi *et al.* "*DualGAN: Unsupervised Dual Learning for Image-to-Image Translation*". ICCV2017.





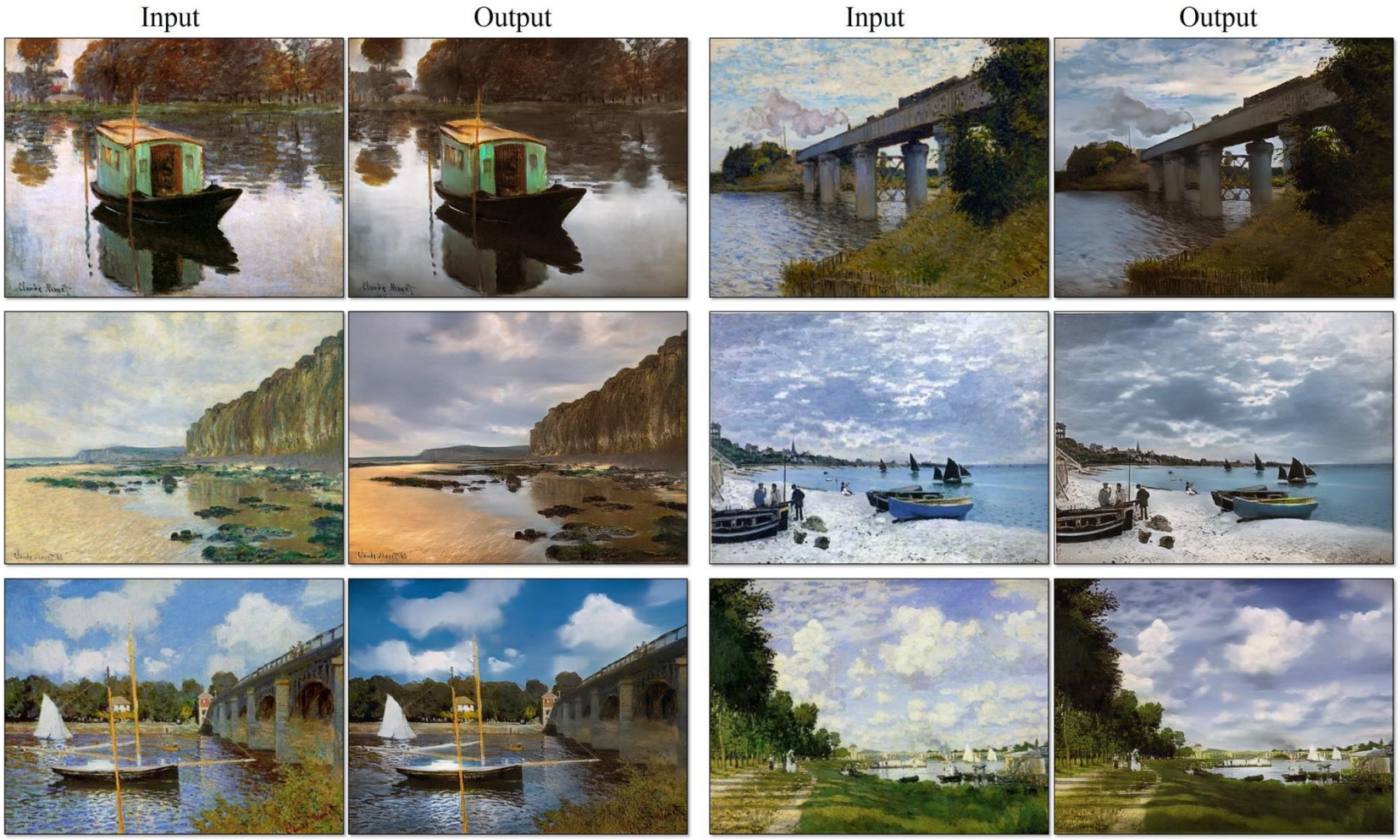
Yi *et al.* "*DualGAN: Unsupervised Dual Learning for Image-to-Image Translation*". ICCV2017.

CycleGAN

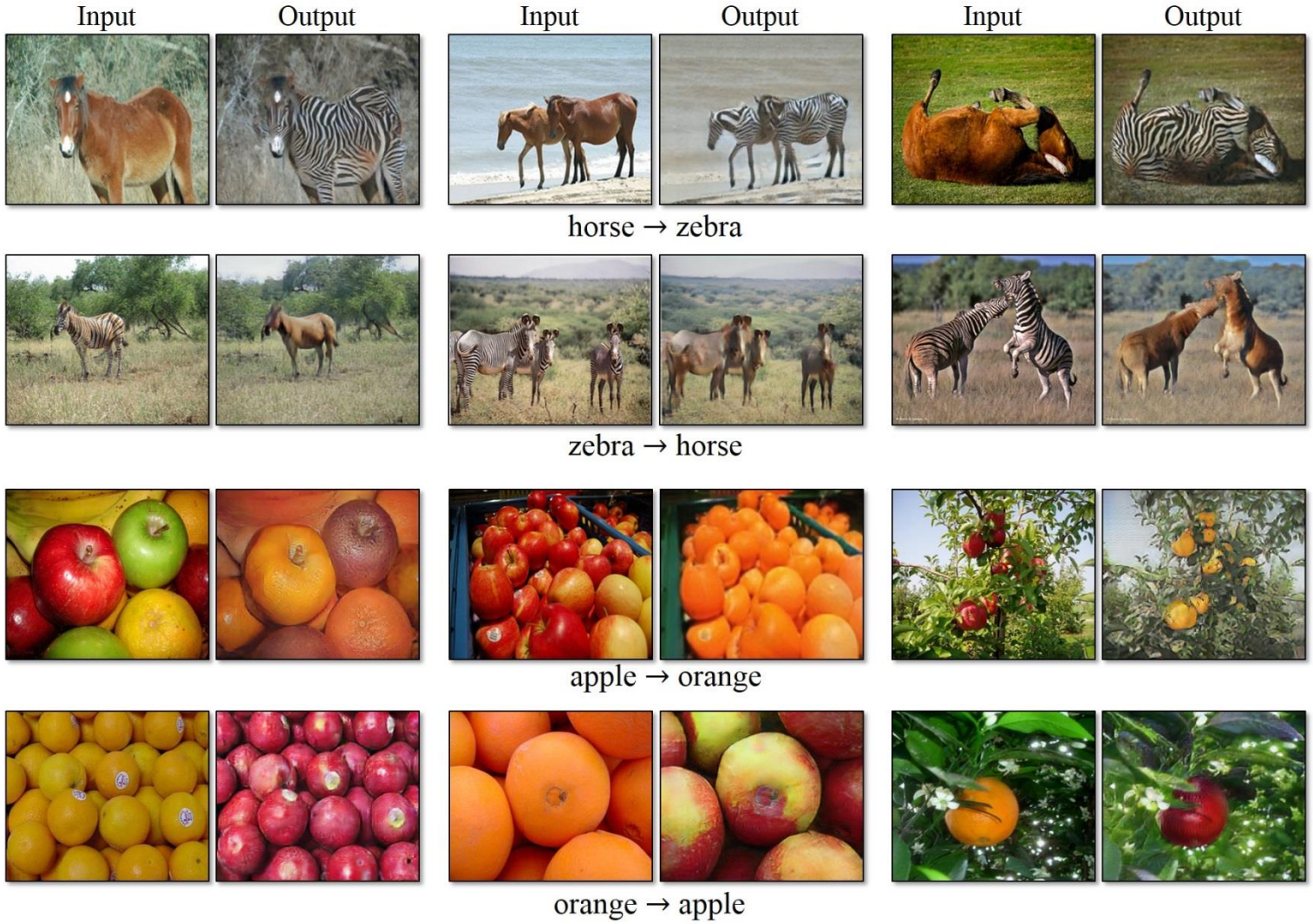


- $L_{cyc}(G,F) = E_{x\sim p_{data(x)}}\left[\left||F(G(x)) - x\right|\right] + E_{y\sim p_{data(y)}}\left[\left||G(F(y)) - y\right|\right]$

Yi *et al.* "*DualGAN: Unsupervised Dual Learning for Image-to-Image Translation*". ICCV2017.

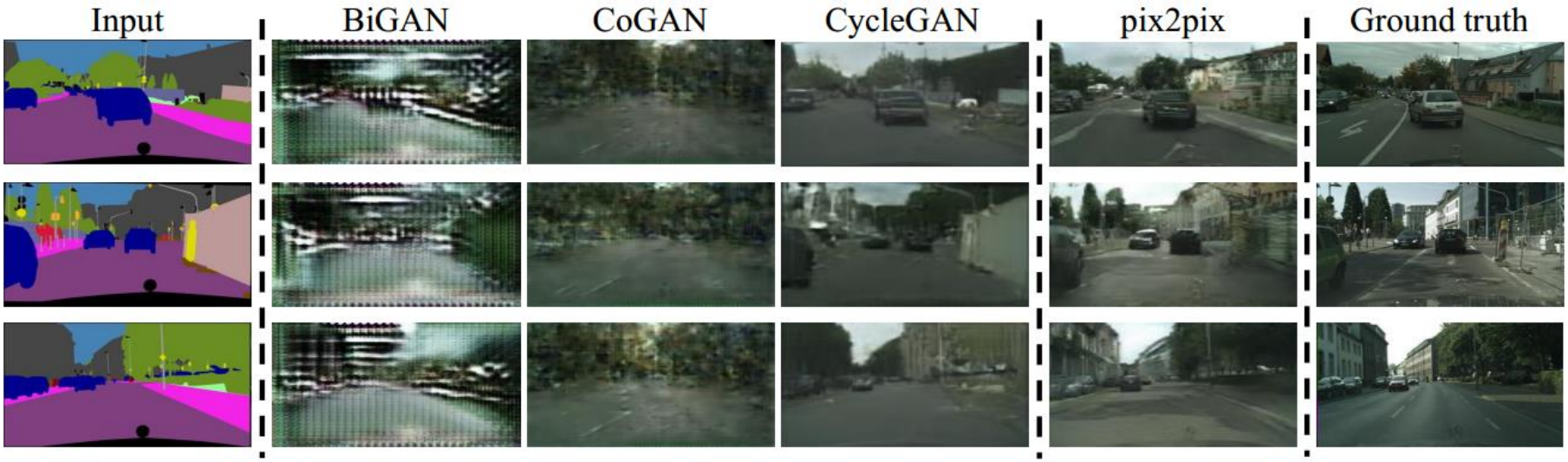


Zhu *et al.* "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks ". arXiv 2017.

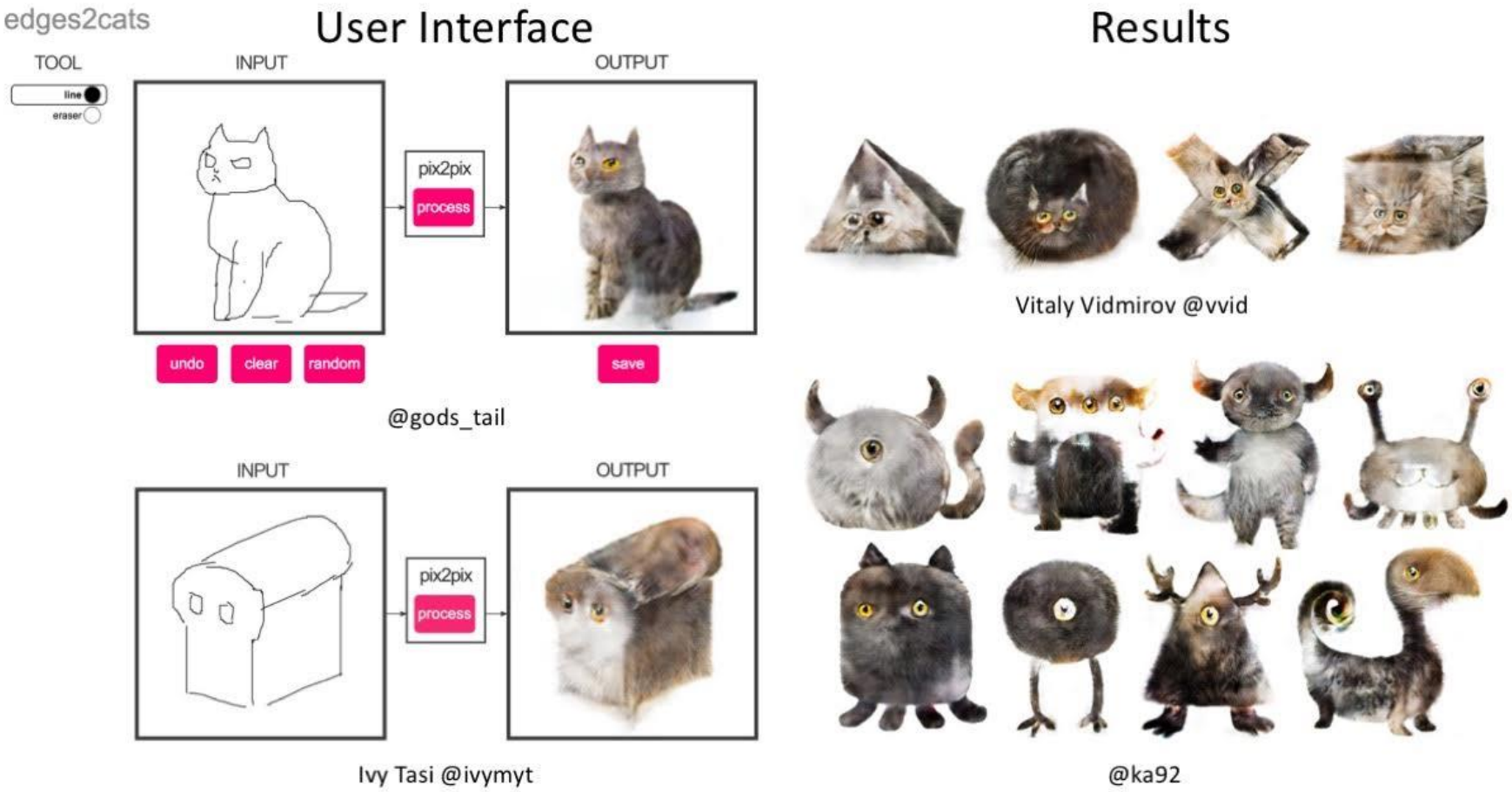


Zhu *et al.* "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks ". arXiv 2017.

Encoders-incorporated



Zhu *et al.* "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks ". arXiv 2017.



- https://affinelayer.com/pixsrv/

WGAN

- Wasserstein Distance:

$$W(\mathrm{P}_r, \mathrm{P}_g) = \inf_{\gamma \in \prod(\mathrm{P}_r, \mathrm{P}_g)} E_{(x,y)}[\| x - y \|]$$

- Why is it superior to KL & JS divergence?

Arjovsky *et al.* "Wasserstein GAN ". arXiv 2017.

Wasserstein Distance

- Wasserstein Distance:

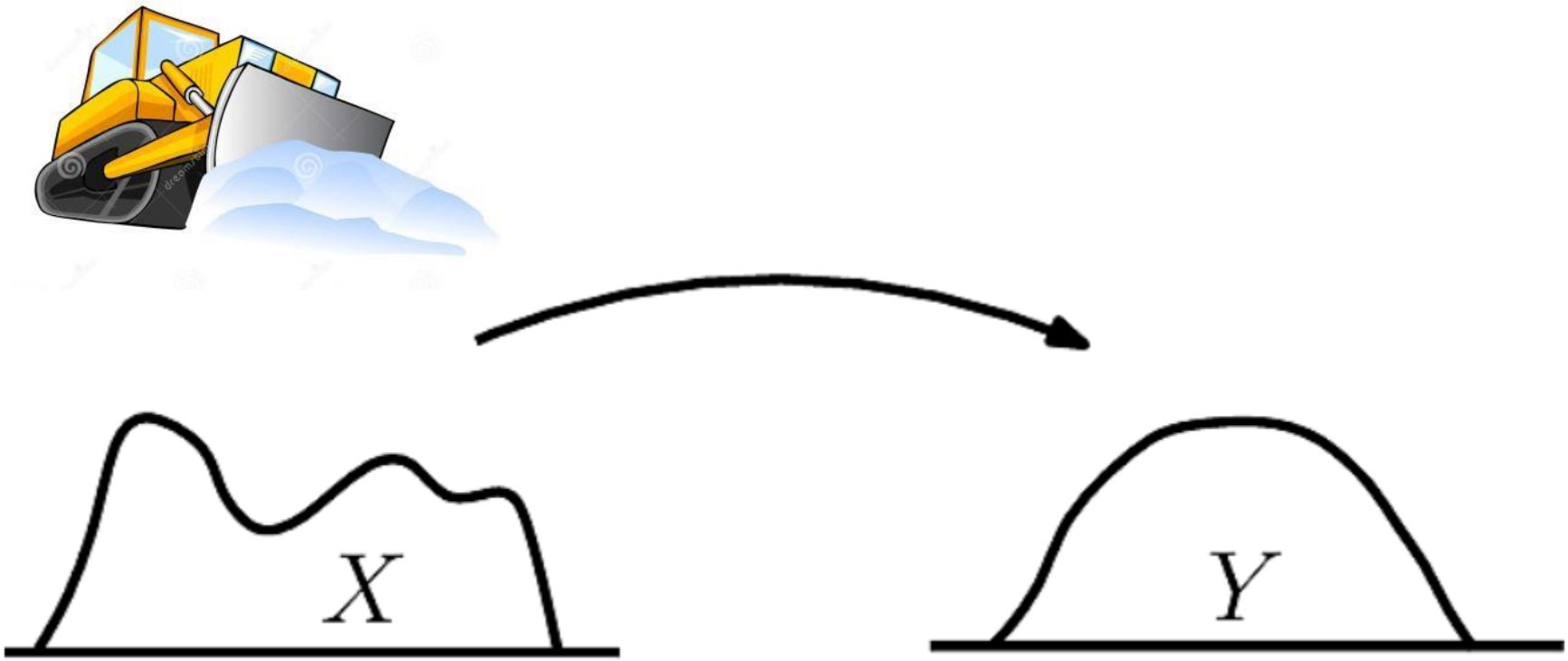
$$W(\mathrm{P}_r, \mathrm{P}_g) = \inf_{\gamma \in \prod(\mathrm{P}_r, \mathrm{P}_g)} E_{(x,y)}[\| x - y \|]$$

- $\prod(P_r, P_g)$ denotes the set of all joint distribution $\gamma(x, y)$ whose marginals are respectively P_r and P_g.
- $\gamma(x, y)$ indicates how much "mass" must be transported from x to y in order to transform the distribution P_r to P_g.
- EM distance is the cost of the optimal transport plan

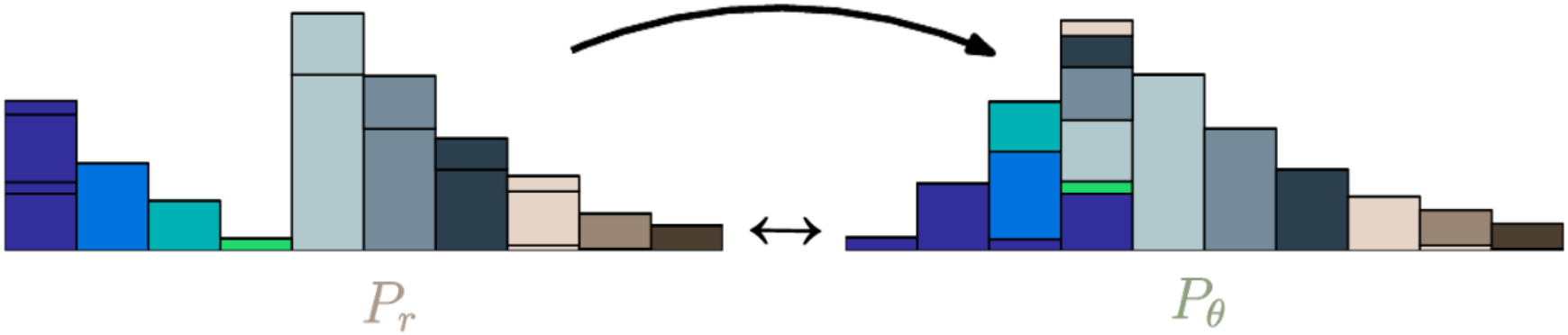
Arjovsky *et al.* "Wasserstein GAN ". arXiv 2017.

Wasserstein distance

- “吊打” KL & KS



Wasserstein Distance



Wasserstein Distance

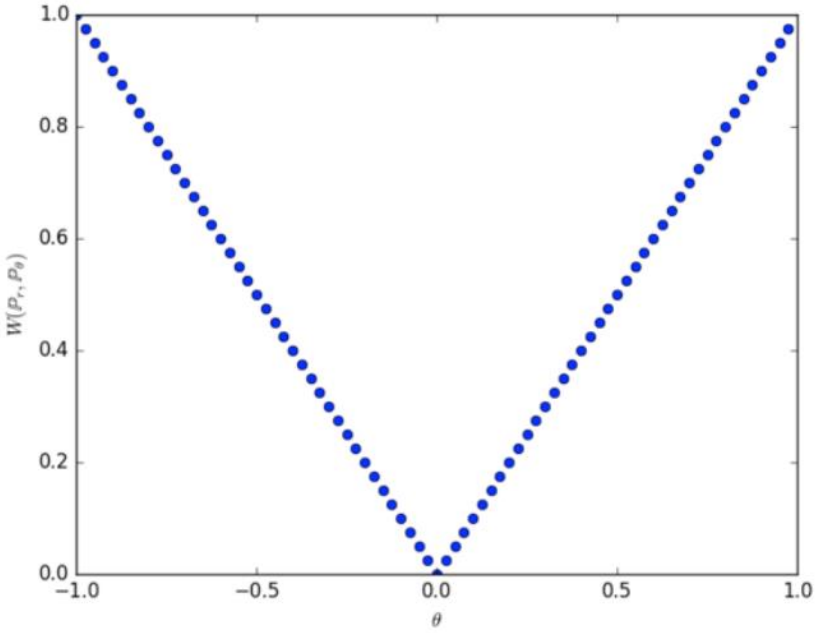
- Wasserstein Distance:

$$W(\mathrm{P}_r, \mathrm{P}_g) = \inf_{\gamma \in \prod(\mathrm{P}_r, \mathrm{P}_g)} E_{(x,y)}[\| x - y \|]$$

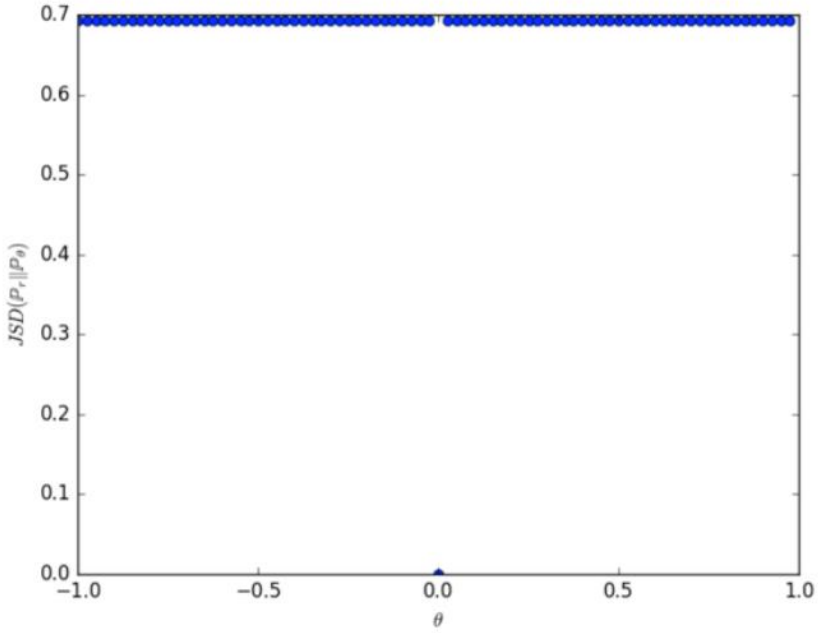
- The distance is shown to have the desirable property
- Continuous everywhere
- Differentiable almost everywhere
- Most importantly:
  - It can reflect the distance of two distributions even if they do not overlap, and thus can provide meaningful gradient.

Arjovsky *et al.* "Wasserstein GAN ". arXiv 2017.

Wasserstein Distance



Wasserstein distance



JS Divergence

Arjovsky *et al.* "Wasserstein GAN ". arXiv 2017.

From Wasserstein Distance to WGAN

- Wasserstein Distance:

$$W(\mathrm{P}_r, \mathrm{P}_g) = \inf_{\gamma \in \prod(\mathrm{P}_r, \mathrm{P}_g)} E_{(x,y)}[\| x - y \|]$$

- By applying the kantorovich-Rubinstein duality it becomes:

$$\min_G \max_{D \in D} E_{x \sim P_r}[D(x)] - E_{\tilde{x} \sim P_g}[D(\tilde{x})]$$

Arjovsky *et al.* "Wasserstein GAN ". arXiv 2017.

Lipschitz Continuity

- Real-value function: $f: R \rightarrow R$
- Positive constant: K

$$|f(x_1) - f(x_2)| \leq K|x_1 - x_2|$$

- In other words, a Lipschitz continuous function has bounded first derivative. Intuitively, the slope of a KK Lipschitz function never exceeds KK.

$$d_Y\big(f(x_1), f(x_2)\big) \leq K d_X(x_1, x_2)$$

From Wasserstein Distance to WGAN

- Wasserstein Distance:

$$\min_{G} \max_{D \in D} E_{x \sim P_r}[D(x)] - E_{\tilde{x} \sim P_g}[D(\tilde{x})]$$

- To satisfy the requirement, WGAN apply weight clipping [-c,c]

Arjovsky *et al.* "Wasserstein GAN ". arXiv 2017.

From Wasserstein Distance to WGAN

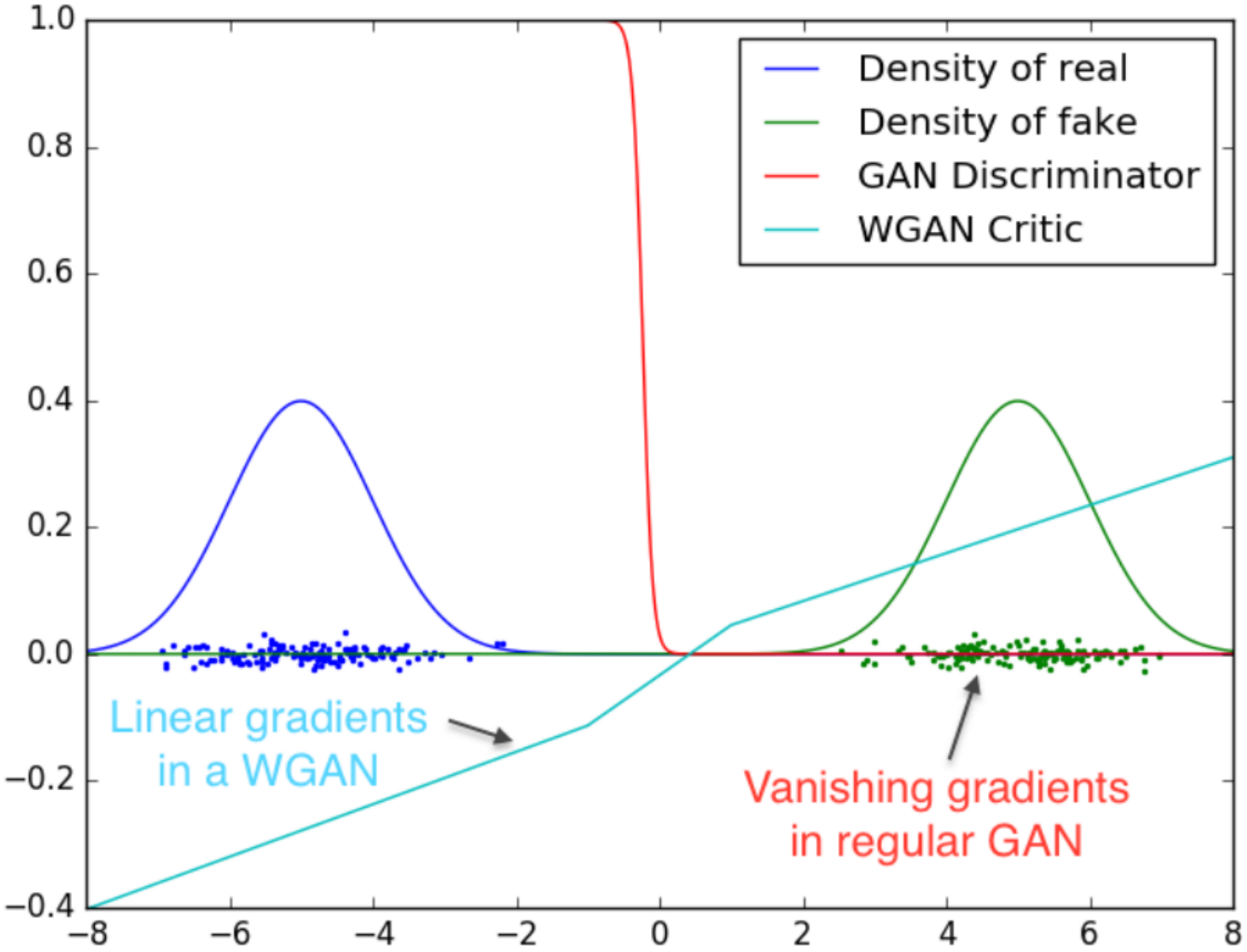
- Wasserstein Distance:

$$\min_{G} \max_{D \in D} E_{x \sim P_r}[D(x)] - E_{\tilde{x} \sim P_g}[D(\tilde{x})]$$

- WGAN remove s the sigmoid layer in **D** because by using Wasserstein distance, **D** is ding regression rather than classifiaction.

Arjovsky *et al.* "Wasserstein GAN ". arXiv 2017.

From Wasserstein Distance to WGAN



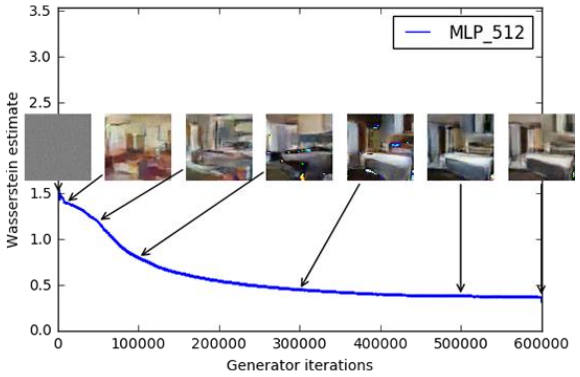
Arjovsky *et al.* "Wasserstein GAN ". arXiv 2017.

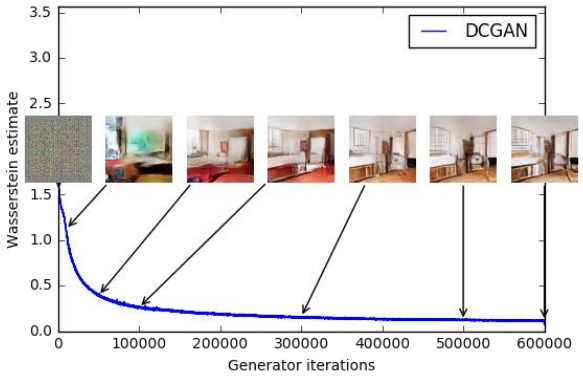
Efficient of WGAN

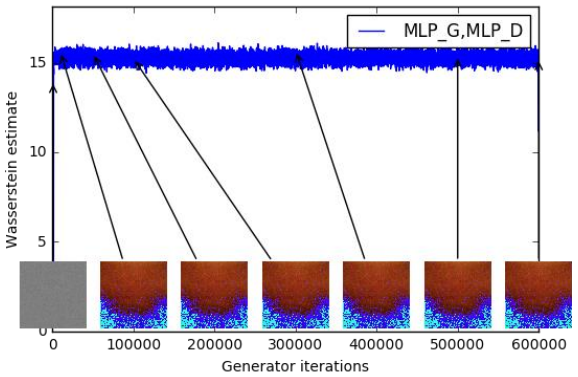
- GAN loss **corresponds** with image quality
- **Converge**, so you can tune the hyper-parameters
- **Stable** training; even without BN
- Way **less mode collapse**
- Theory about why it works and JS is terrible

Arjovsky *et al.* "Wasserstein GAN ". arXiv 2017.

Efficient of WGAN



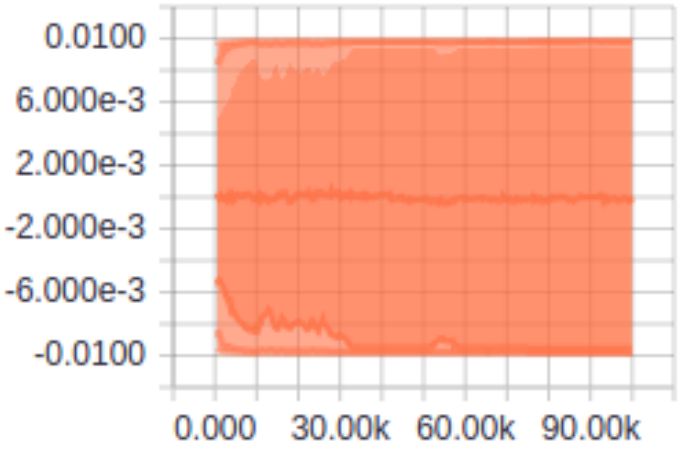




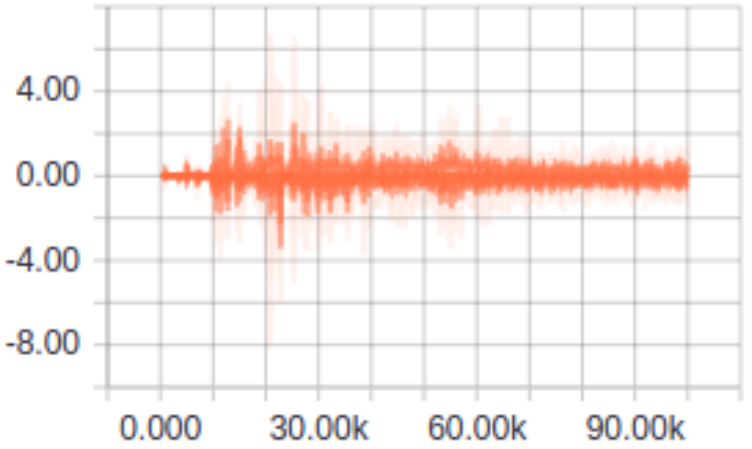
Arjovsky *et al.* "Wasserstein GAN ". arXiv 2017.

Efficient of WGAN

discriminator/W_0



discriminator/W_0/gradient



Arjovsky *et al.* "Wasserstein GAN ". arXiv 2017.

Training WGAN

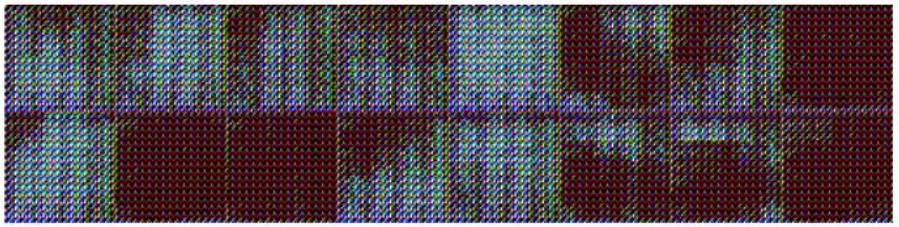
GAN & DCGAN

DCGAN

GAN

Without BN

WGAN

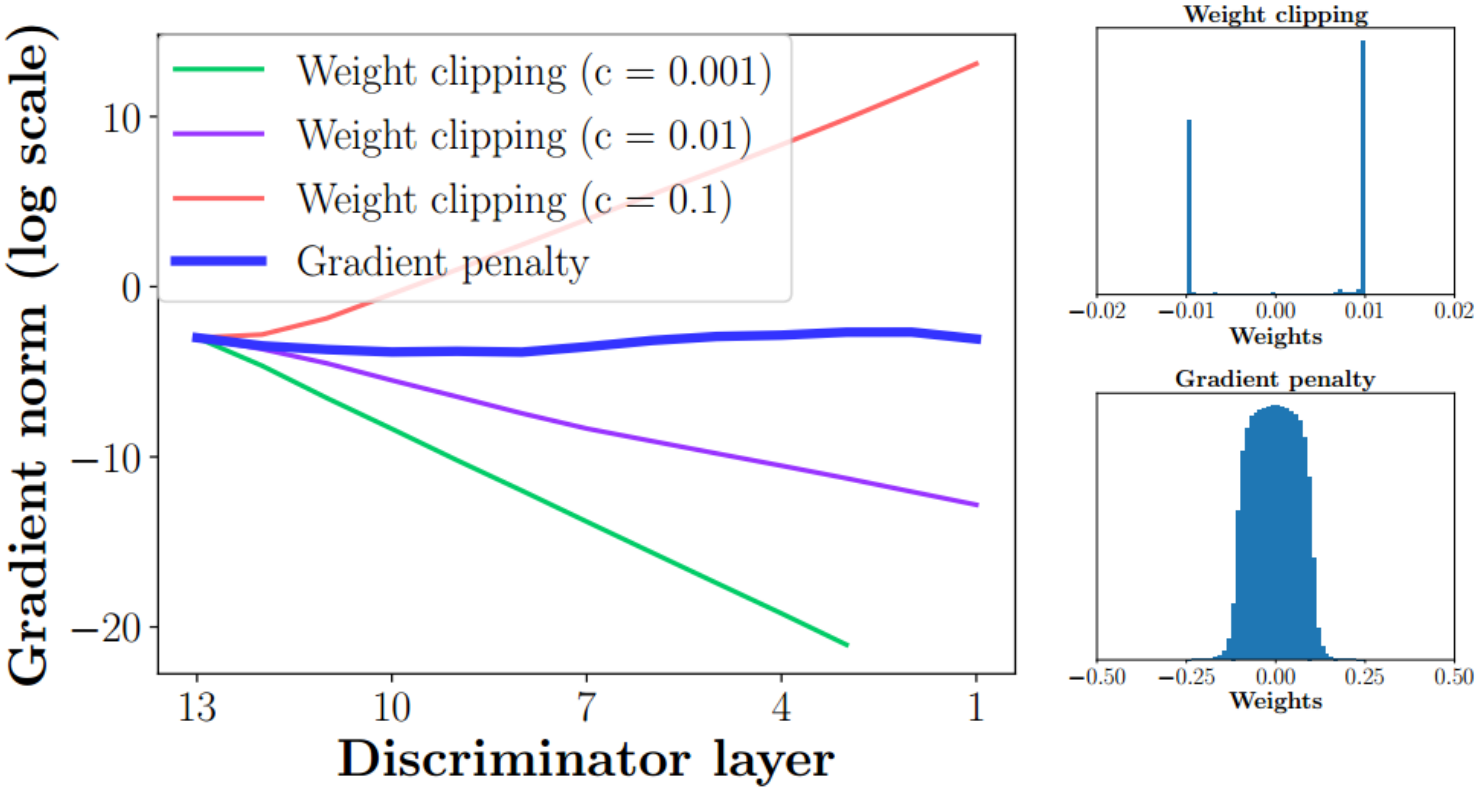


DCGAN

Arjovsky *et al.* "Wasserstein GAN ". arXiv 2017.

Improved WGAN

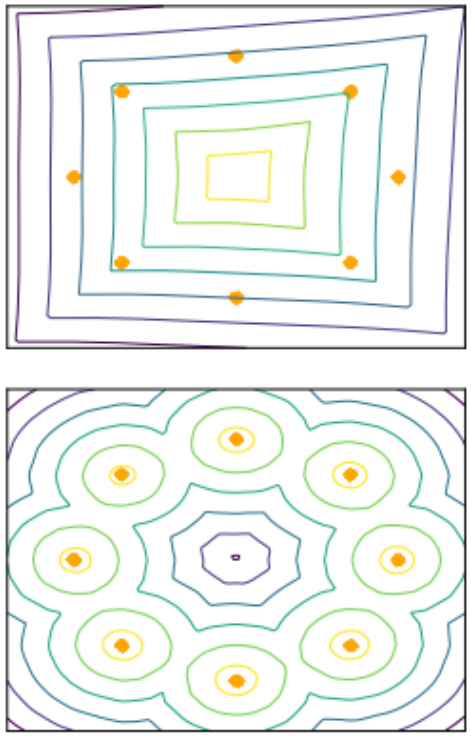
- Weight clipping



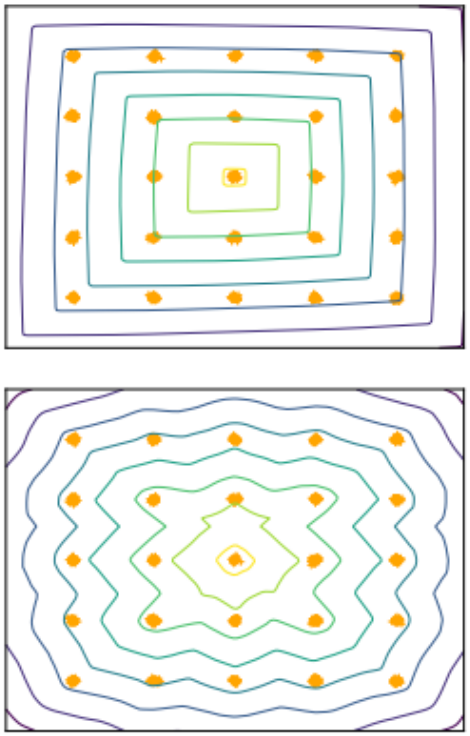
I Gulrajani *et al.* "Improved Training of Wasserstein GANs". arXiv 2017.

Drawback of weight clipping

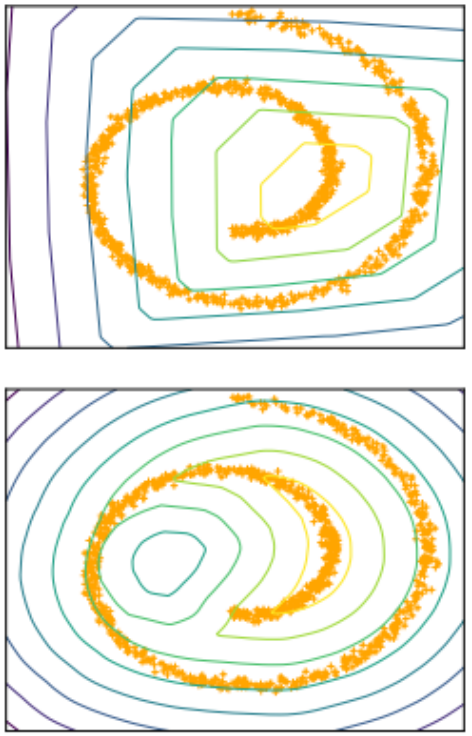
8 Gaussians



25 Gaussians



Swiss Roll



I Gulrajani *et al.* "Improved Training of Wasserstein GANs". arXiv 2017.

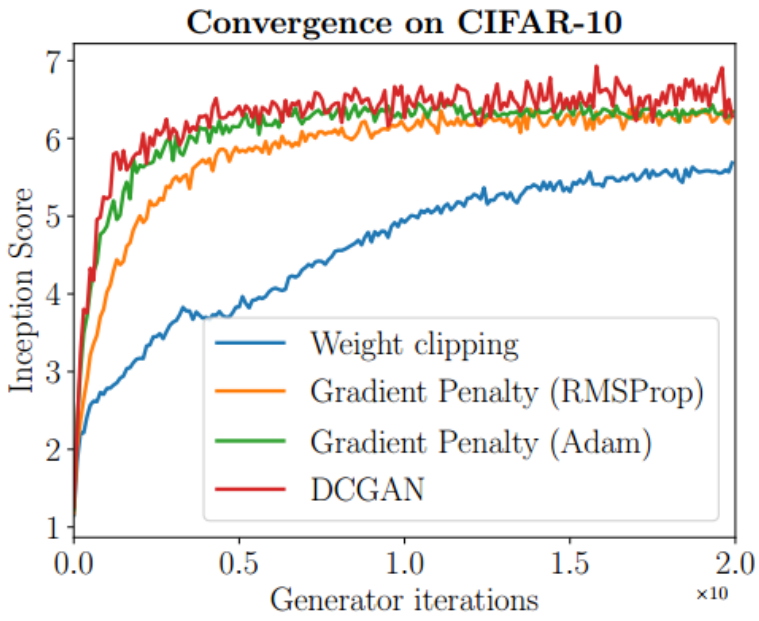
Improved WGAN

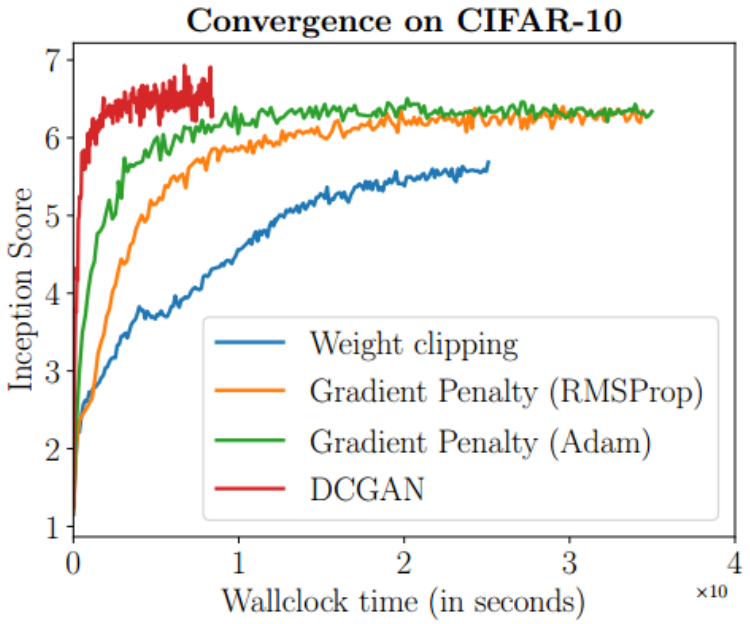
- D of WGAN
  - Has gradients with norm 1 almost everywhere under P_r and P_g
- The objective of improved WGAN

$$L = \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}\left[D(\tilde{\boldsymbol{x}})\right] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}\left[D(\boldsymbol{x})\right]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{\boldsymbol{x}}}}\left[\left(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1\right)^2\right]}_{\text{Our gradient penalty}}.$$

I Gulrajani *et al.* "Improved Training of Wasserstein GANs". arXiv 2017.

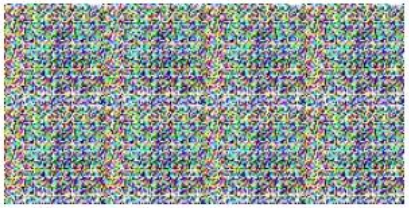
Improved WGAN





I Gulrajani *et al.* "Improved Training of Wasserstein GANs". arXiv 2017.

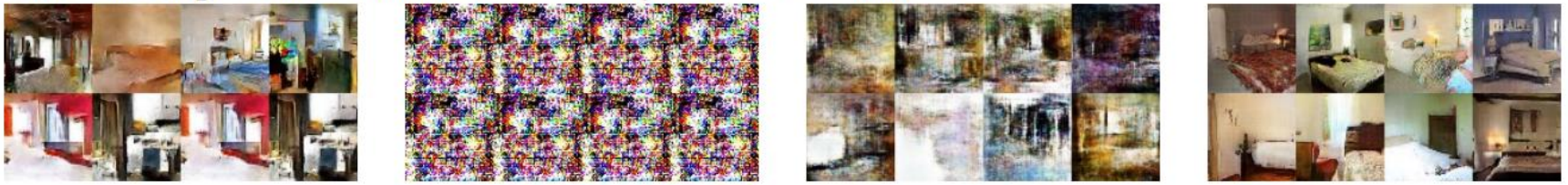
Improved WGAN

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			
			

I Gulrajani *et al.* "Improved Training of Wasserstein GANs". arXiv 2017.

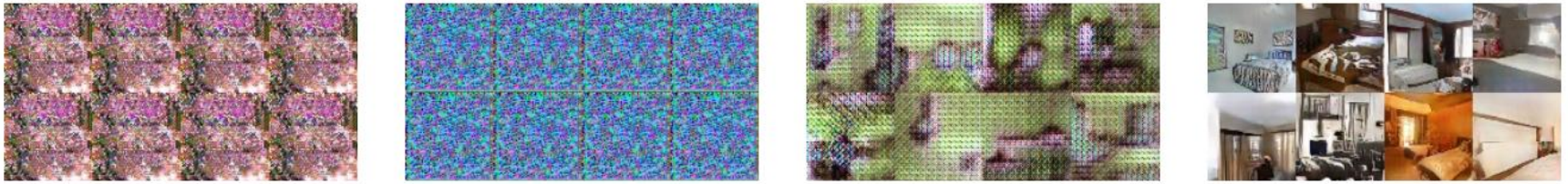
Improved WGAN

Gated multiplicative nonlinearities everywhere in G and D



$tanh$ nonlinearities everywhere in G and D

101-layer ResNet G and D



I Gulrajani *et al.* "Improved Training of Wasserstein GANs". arXiv 2017.

What's **Next**?

See you two weeks later