Northeastern

CS 7140:
ADVANCED MACHINE LEARNING

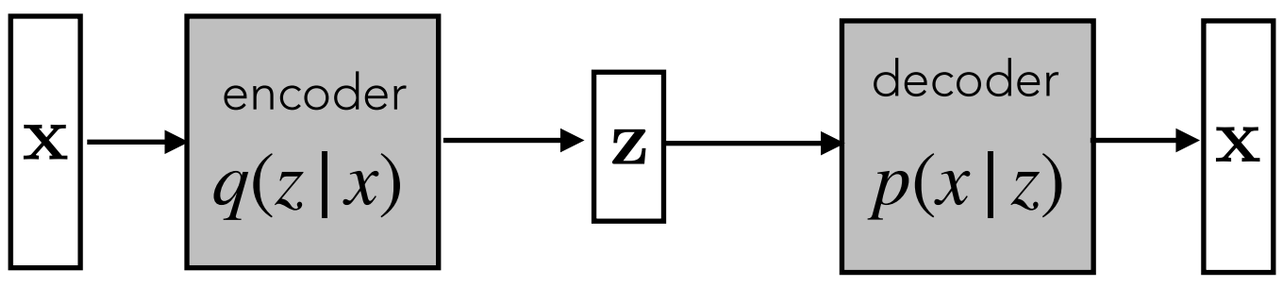
Recap: Variational Autoencoder (VAE)

- Optimize the ELBO

$$\log p(x) \geq \mathbb{E}_q[\log p(x|z)] - \mathrm{KL}(q(z|x)||p(z))$$

generative net (decoder) — *inference net (encoder)* — *prior*

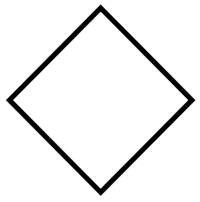
- Approximate the distributions with neural networks



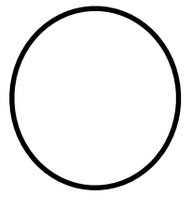
How to backprop through a stochastic random variable?

Recap: Reparameterization Trick

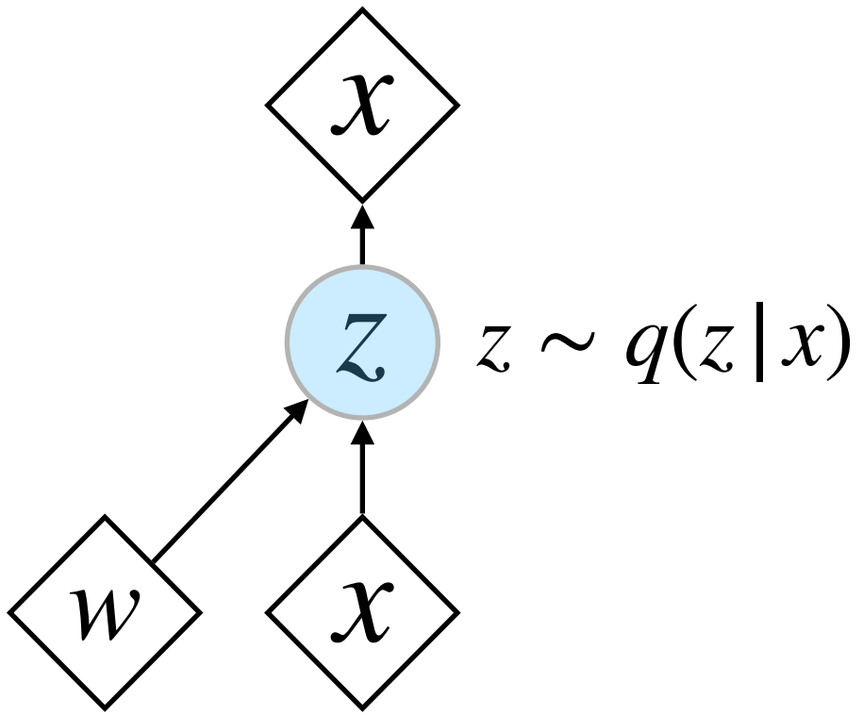
- Back-propagation through *stochastic* transformation
- Introduce a new random variable ϵ



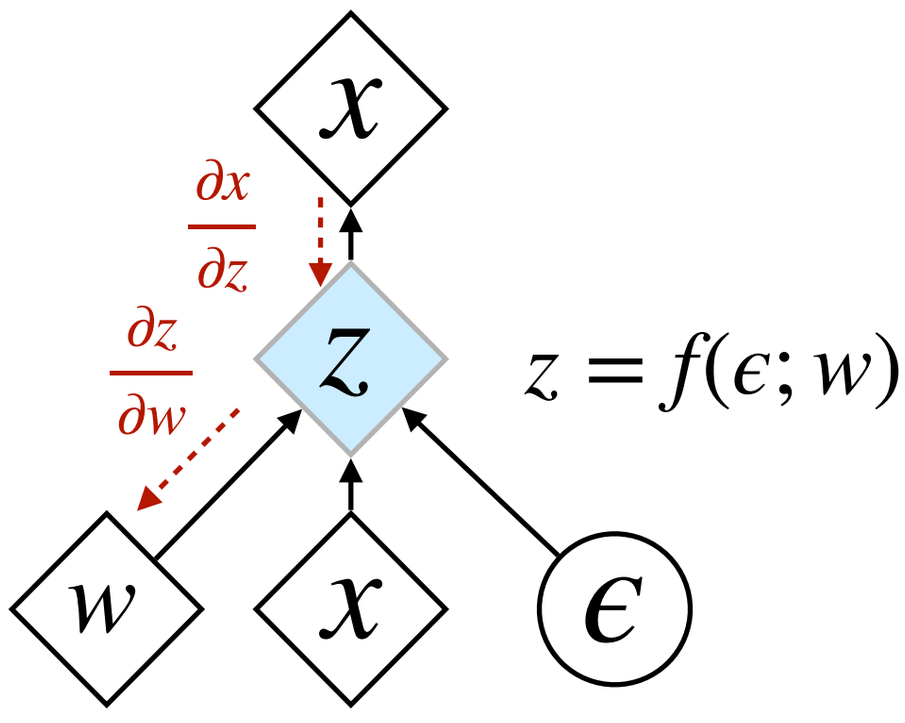
deterministic variable



stochastic variable



$z \sim q(z \mid x)$

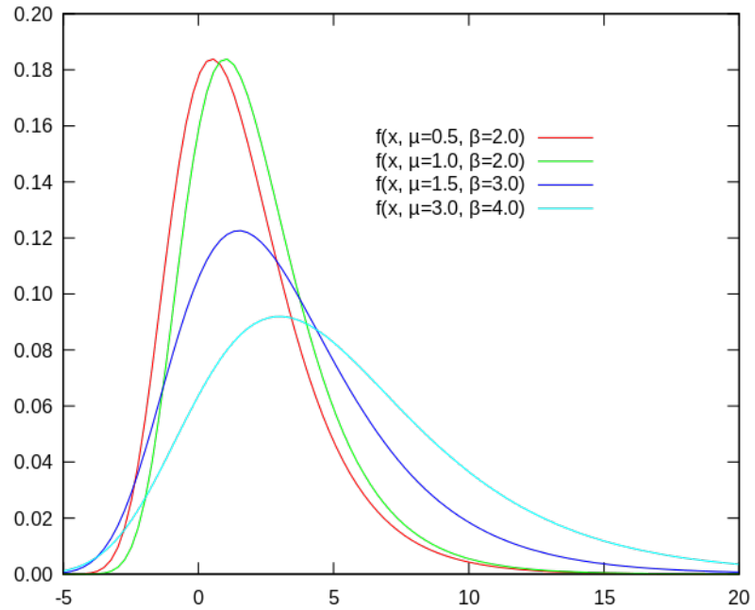


$z = f(\epsilon; w)$

Recap: Gumbel-Soft max

Gumbel distribution

PDF: $\frac{1}{\beta}e^{z+e^{-z}}$ $z = \frac{x-\mu}{\beta}$



Gumbel random variable

$U \sim \text{uniform}(0,1)$ $G = -\log(-\log U)$

Re-parametrize discrete variable

$P(Z = k) = \pi_k$ $Z = \text{argmax}_k(\log \pi_k + G_k)$

Softmax

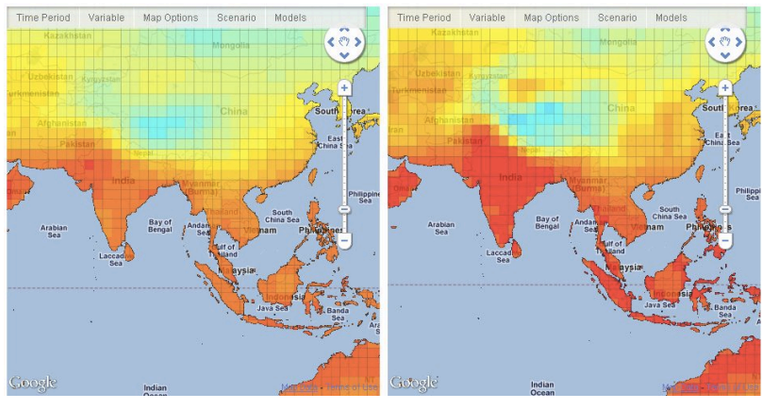
$$\sigma(\pi) = \frac{e^{\pi_k}}{\sum_k e^{\pi_k}}$$

GENERATIVE MODELING

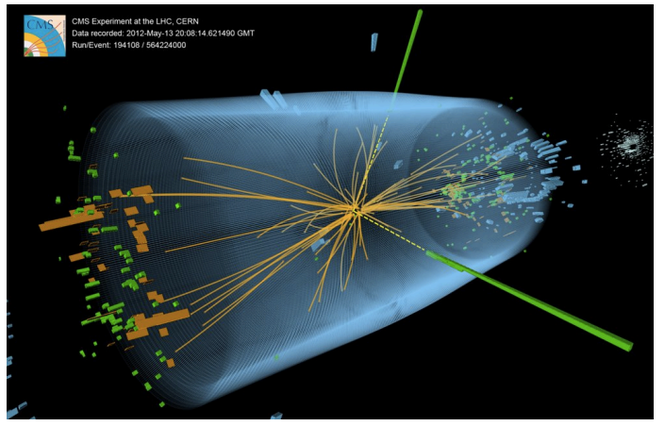
Overview

What is a generative model? $P(x, y)$

- A model that allows us to learn a **simulator** of data
- Models that allow for (conditional) **density estimation**
- Approaches for **unsupervised learning** of data



climate simulations



physics experiments

Why Generative Model

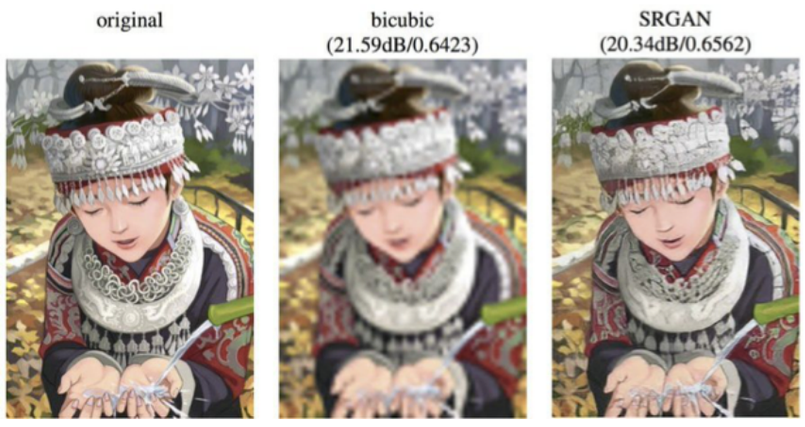
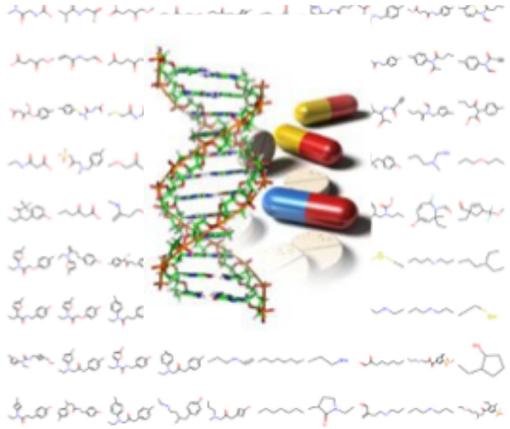
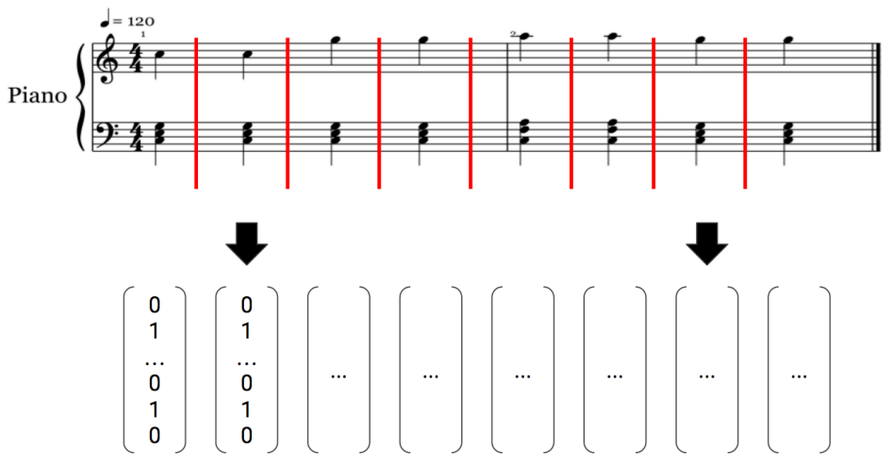


image super-resolution

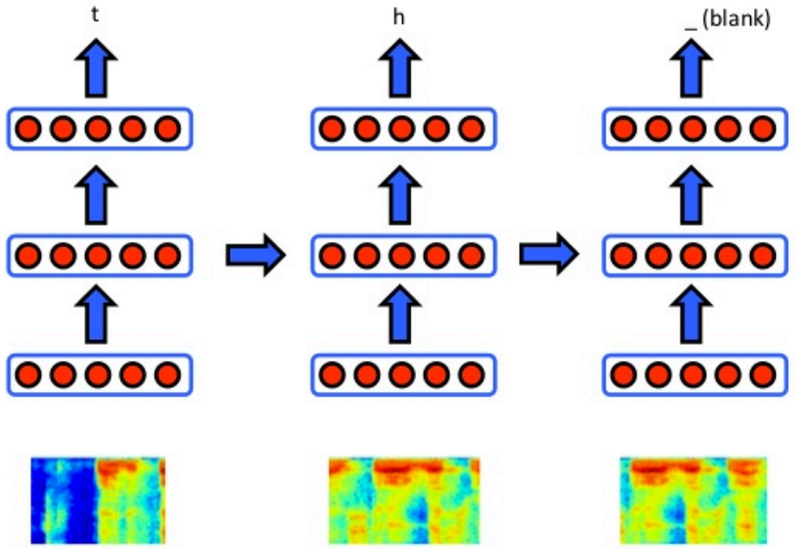


drug design

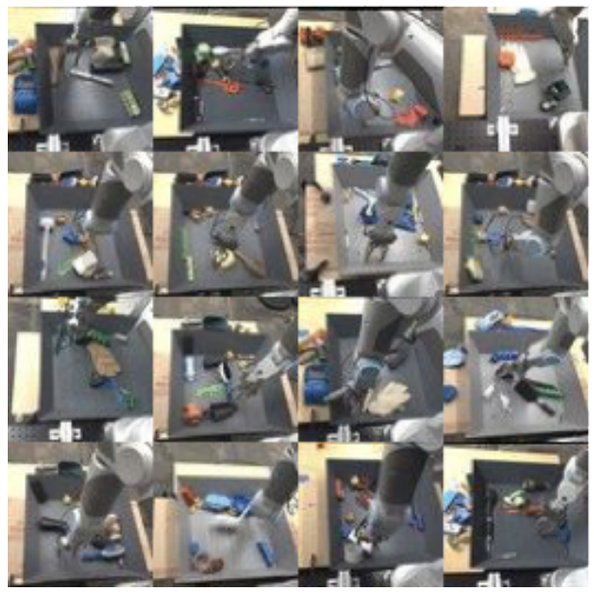
image compression



music composition



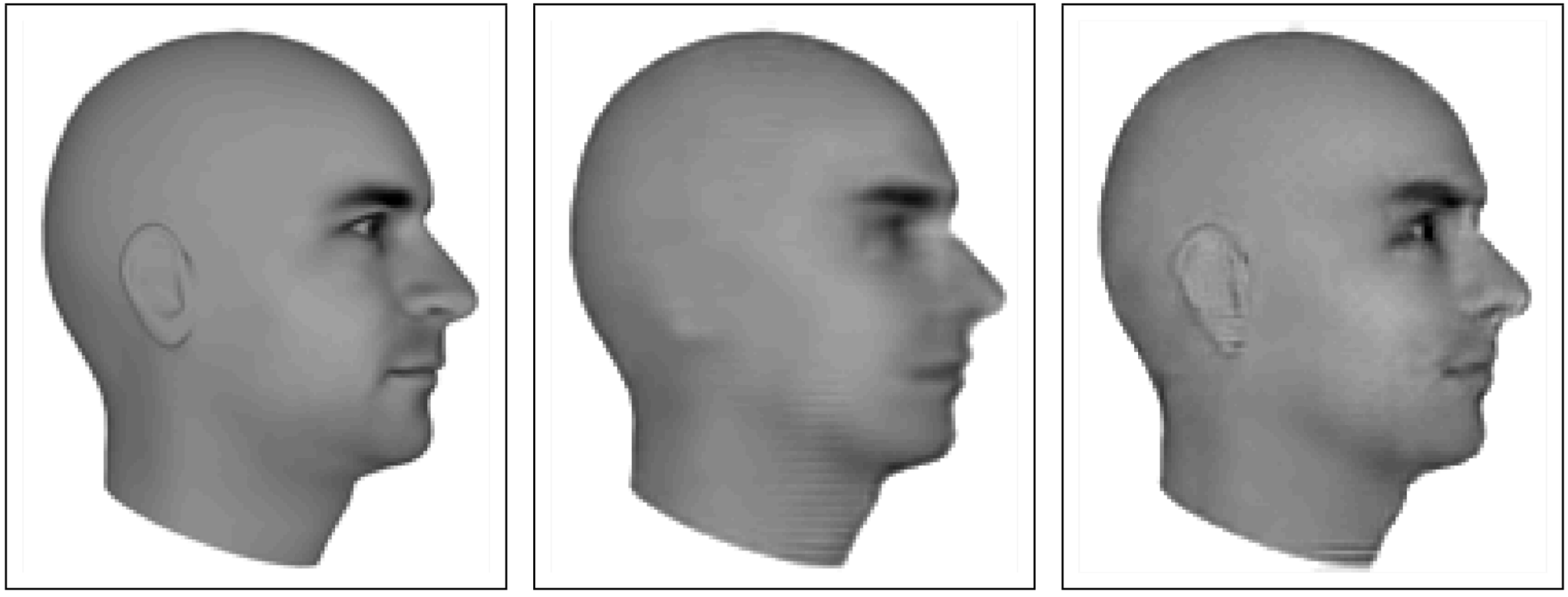
speech to text



physical simulator

GENERATIVE ADVERSARIAL NETWORK

Next Video Frame Prediction

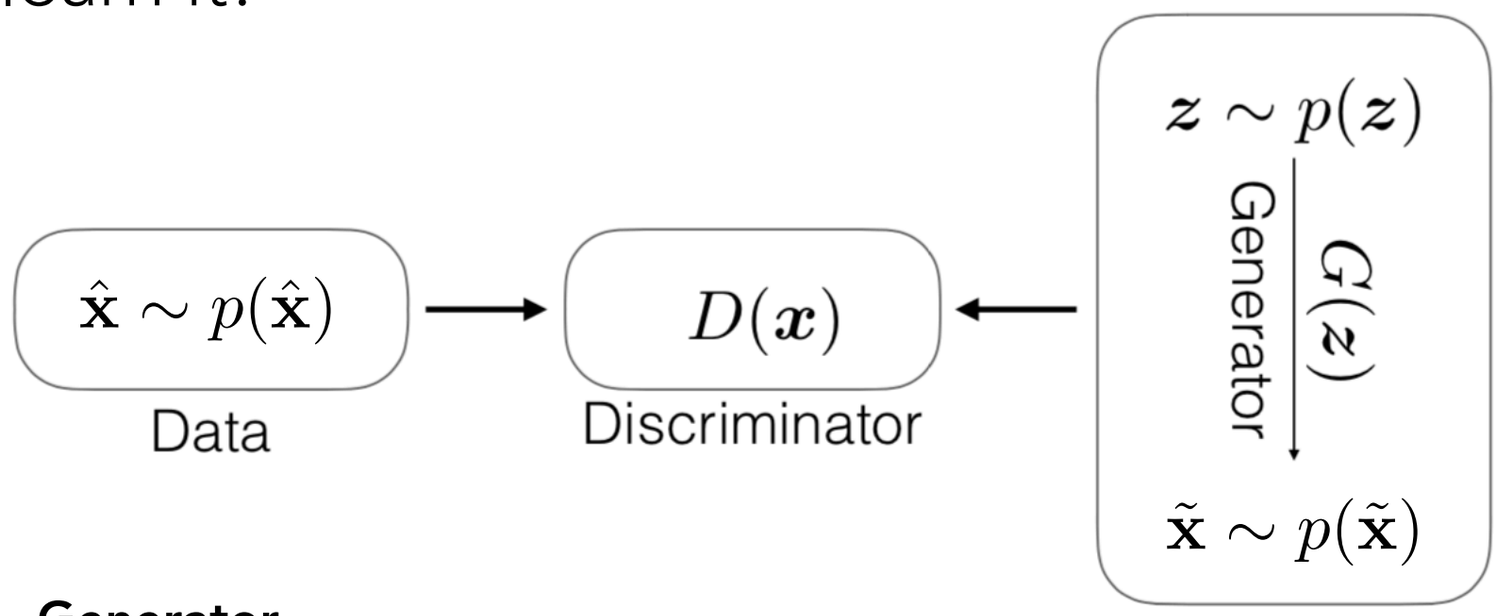


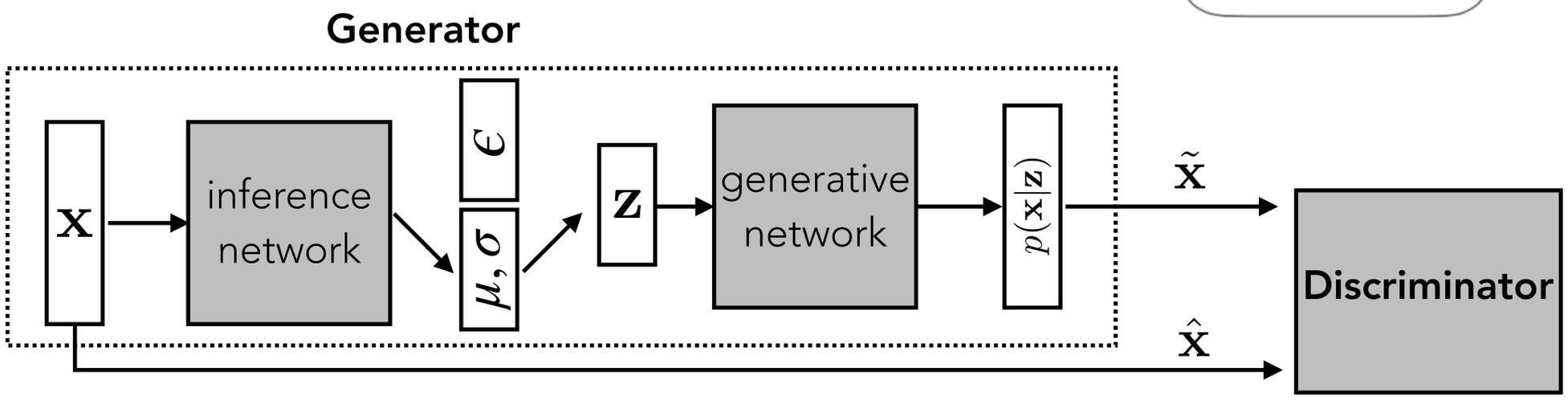
Ground Truth | VAE | GAN

Lotter et al 2016

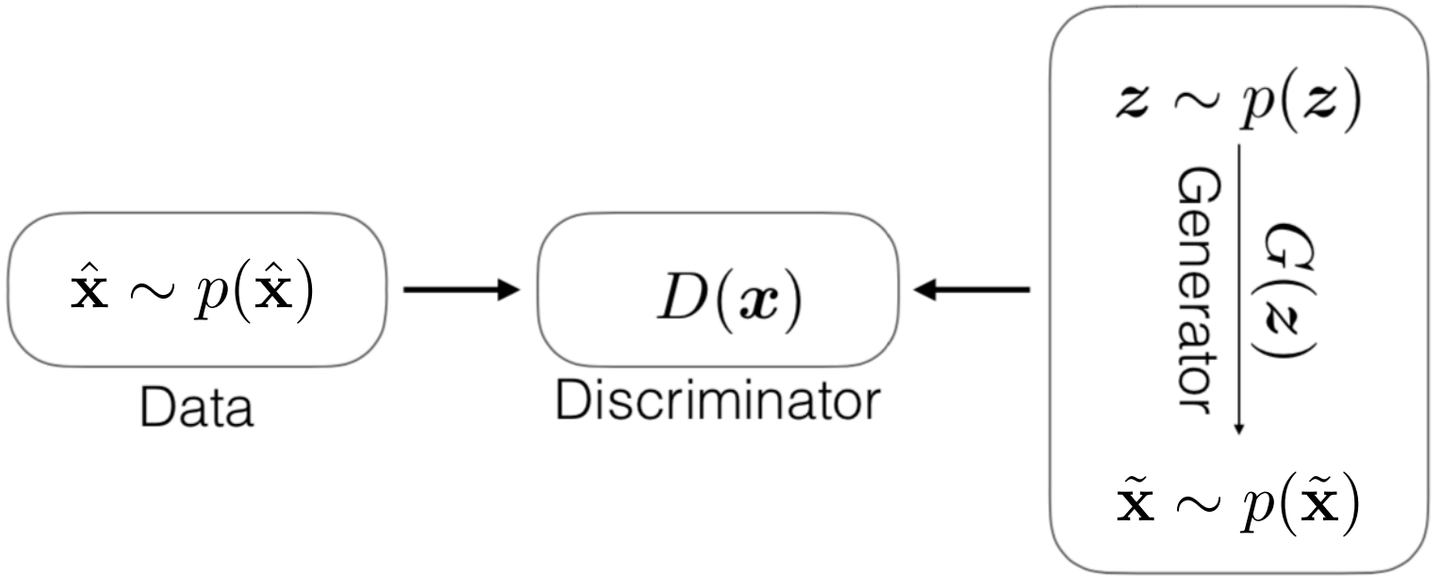
Generative Adversarial Networks (GANs)

Instead of optimizing the log-likelihood (ELBO)
Let the model learn it!





Generative Adversarial Networks (GANs)



learn the discriminator:

$$p(\text{data}|\mathbf{x}) = D(\mathbf{x}) \qquad\qquad p(\text{gen.}|\mathbf{x}) = 1 - D(\mathbf{x})$$

Bernoulli outcome: $y \in \{\text{data}, \text{gen.}\}$

$$\log p(y|\mathbf{x}) = \log D(\hat{\mathbf{x}}) + \log(1 - D(\tilde{\mathbf{x}}))$$

zero-sum game:

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

Goodfellow, 2016

Mohamed, 2016

Zero Sum Game

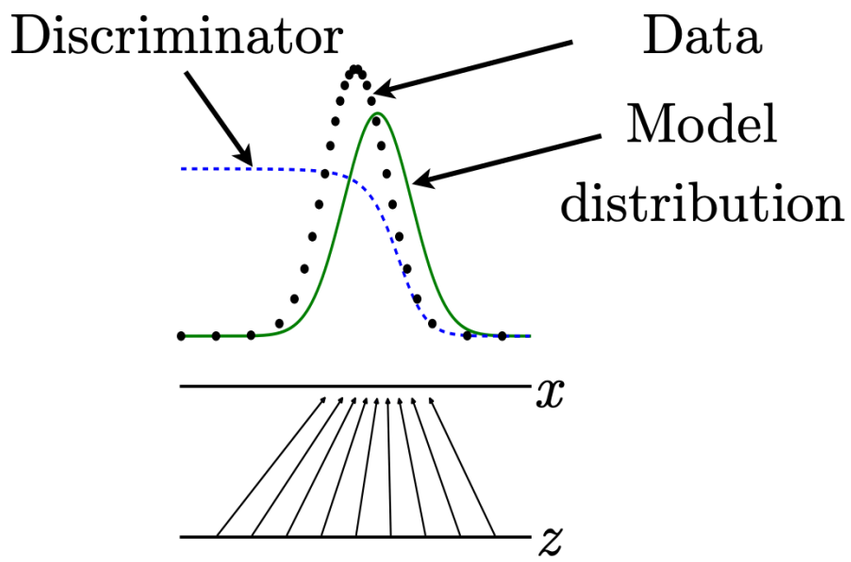
- Generator minimizes the log-probability of the discriminator being correct

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{p(\hat{\mathbf{x}})}[\log D(\hat{\mathbf{x}})] - \frac{1}{2}\mathbb{E}_{\mathbf{z}} \log(1 - D(G(\mathbf{z}))$$

$$J^{(G)} = -J^{(D)}$$

- Optimal discriminator for any data and model is always

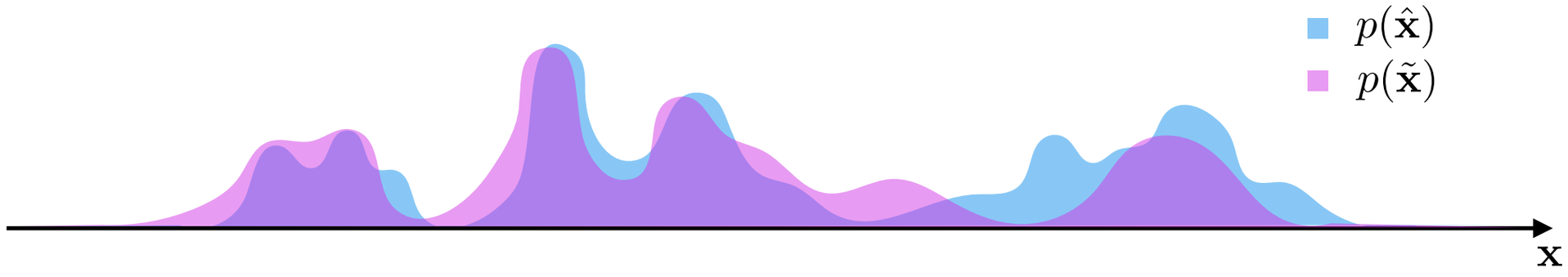
$$D(x) = \frac{p(\hat{\mathbf{x}})}{p(\hat{\mathbf{x}}) + p(\tilde{\mathbf{x}})}$$



Training GAN

- Repeat
  - Train Discriminator for k steps
    - sample z, x
    - update discriminator $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))$
  - Train Generator
    - sample z
    - update generator $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(z^{(i)})))$
- Until converge

Alternative Interpretation: Hypothesis Testing



estimate density ratio through *Bayesian two-sample test*

data distribution $p(\hat{\mathbf{x}})$ generated distribution $p(\tilde{\mathbf{x}})$

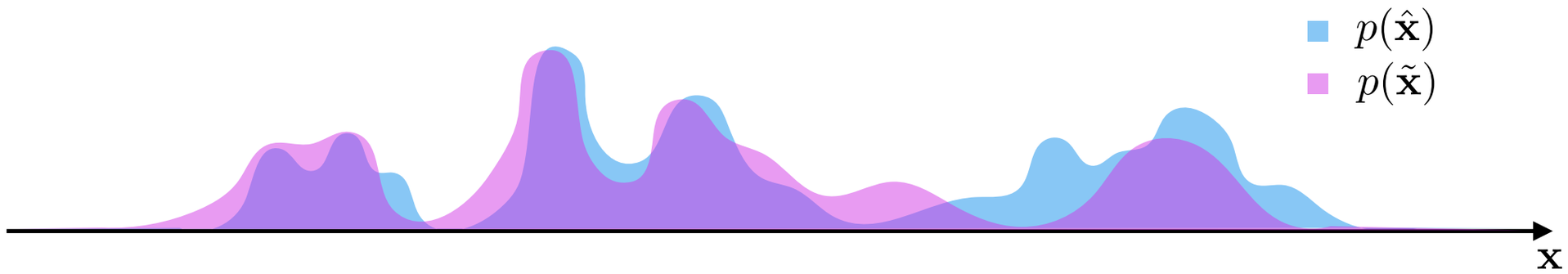
$$\frac{p(\hat{\mathbf{x}})}{p(\tilde{\mathbf{x}})} = \frac{p(\mathbf{x}|\text{data})}{p(\mathbf{x}|\text{gen.})}$$

$$\frac{p(\hat{\mathbf{x}})}{p(\tilde{\mathbf{x}})} = \frac{p(\text{data}|\mathbf{x})p(\mathbf{x})/p(\text{data})}{p(\text{gen.}|\mathbf{x})p(\mathbf{x})/p(\text{gen.})} \quad \text{(Bayes' rule)}$$

$$\frac{p(\hat{\mathbf{x}})}{p(\tilde{\mathbf{x}})} = \frac{p(\text{data}|\mathbf{x})}{p(\text{gen.}|\mathbf{x})} \quad \text{(assuming equal dist. prob.)}$$

density estimation becomes a sample discrimination task

Hypothesis Testing



estimate density ratio through *Bayesian two-sample test*

data distribution $p(\hat{\mathbf{x}})$ generated distribution $p(\tilde{\mathbf{x}})$

under an *"ideal"* discriminator, we can prove that
the generator minimizes the **Jensen-Shannon divergence**

$$\min_G \max_D V(D^\star, G) = 2D_{JS}(p(\hat{x}) \,||\, p(\tilde{x})) - 2\log 2$$

$$D_{JS}(p(\hat{\mathbf{x}})||p(\tilde{\mathbf{x}})) = \frac{1}{2}D_{KL}(p(\hat{\mathbf{x}})||\frac{1}{2}(p(\hat{\mathbf{x}}) + p(\tilde{\mathbf{x}}))) + \frac{1}{2}D_{KL}(p(\tilde{\mathbf{x}})||\frac{1}{2}(p(\hat{\mathbf{x}}) + p(\tilde{\mathbf{x}})))$$

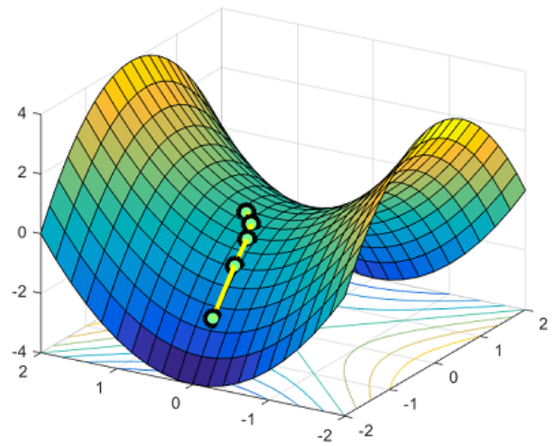
Difficulty of Training GANs

- Mini-max game

$$\min_{G} \max_{D} V(D, G)$$

optima is a saddle point

non-convex optimization



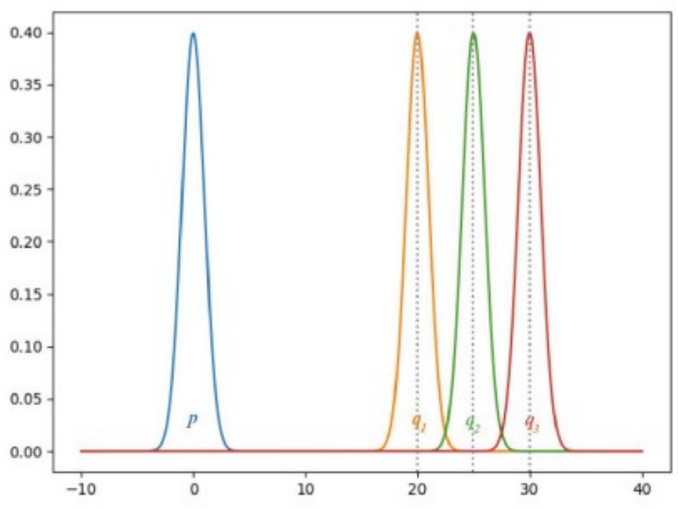
saddle point

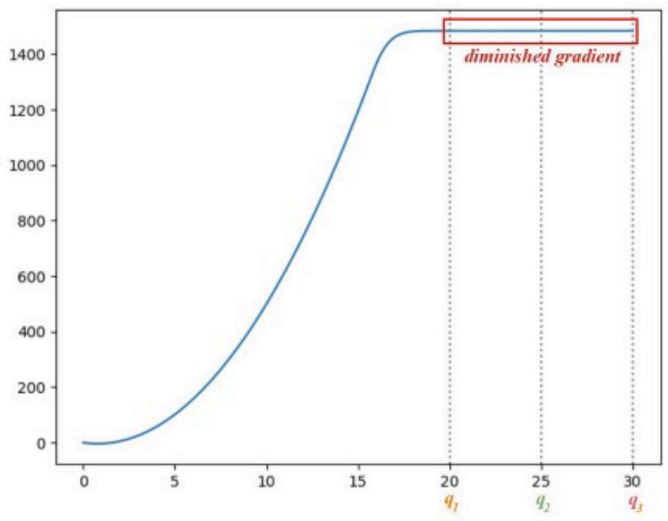
- Vanishing Gradient

$$D_{JS}(p(x)||q(x))$$

generated distribution is far away

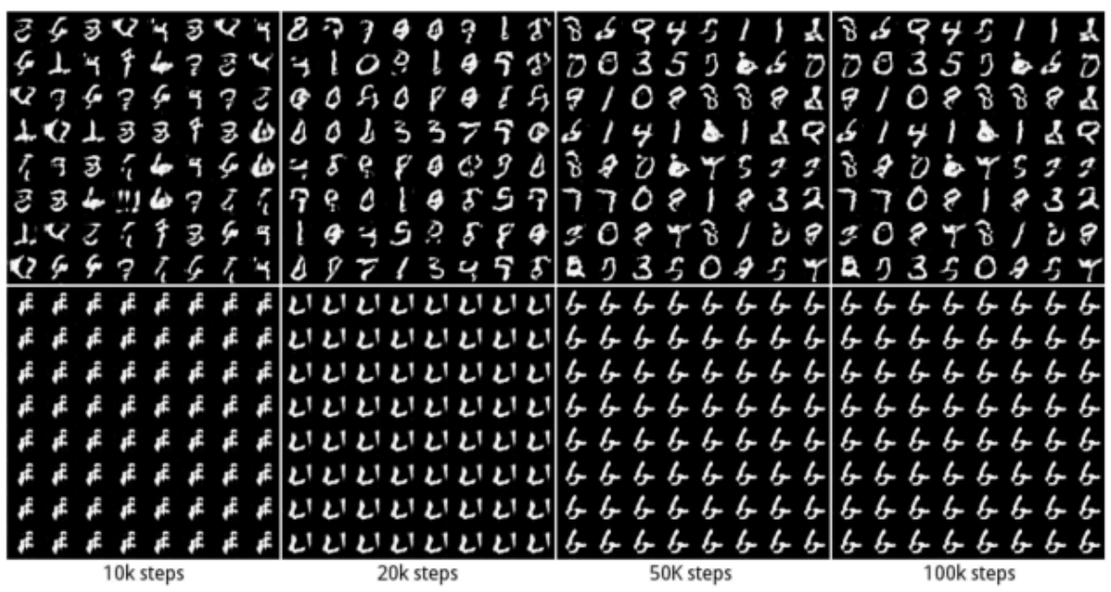
zero gradient



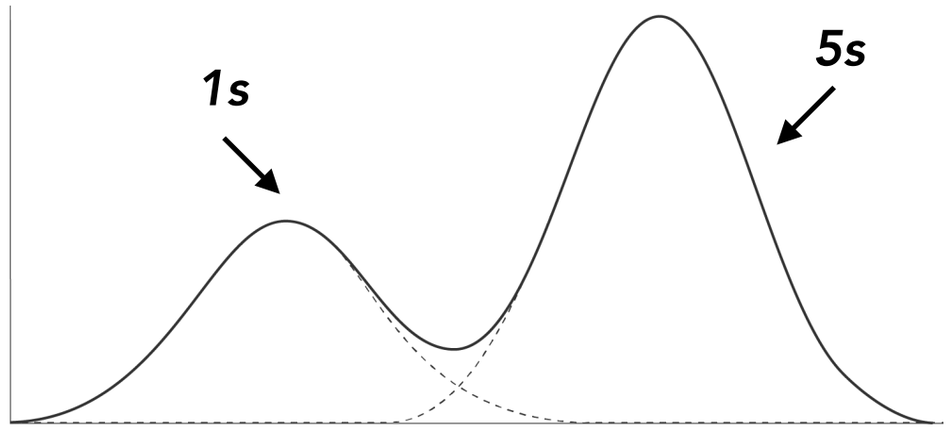


Mode Collapse

- Most data distributions are multimodal
- Generator is independent of the latent variable



discriminator fails to detect new mode

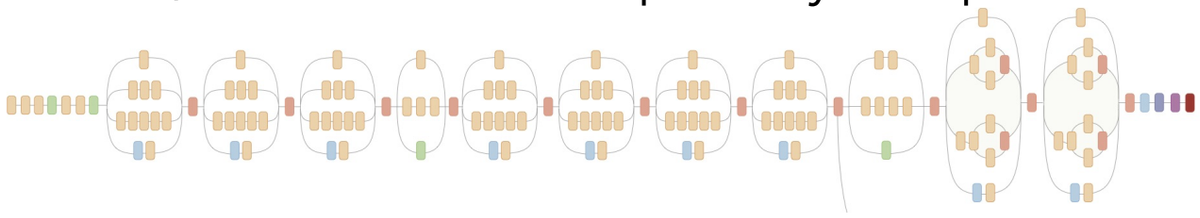


Metz, Luke, et al. "Unrolled generative adversarial networks." *arXiv preprint arXiv:1611.02163* (2016).

Evaluation

without an explicit likelihood, it is difficult to quantify the performance

inception score



use a pre-trained Inception v3 model to quantify class and distribution entropy

$$\mathrm{IS}(G) = \exp\left(\mathbb{E}_{p(\tilde{\mathbf{x}})} D_{KL}(p(y|\tilde{\mathbf{x}})||p(y))\right)$$

$p(y|\tilde{\mathbf{x}})$ is the class distribution for a given image

⟶ should be highly peaked (low entropy)

$p(y) = \int p(y|\tilde{\mathbf{x}}) d\tilde{\mathbf{x}}$ is the marginal class distribution

⟶ want this to be uniform (high entropy)

in practice

$$\mathrm{IS}(G) \approx \exp\left(\frac{1}{M} D_{\mathrm{KL}}(p(y\,|\,x^{(i)})\,||\,\tilde{p}(y)\right) \qquad \tilde{p}(y) = \frac{1}{N}\sum_{i}^{N} p(y\,|\,x^{(i)})$$

Improved Techniques for Training GANs, *Salimans et al.*, 2016

A Note on the Inception Score, *Barratt & Sharma*, 2018

Difficulty of Training GAN

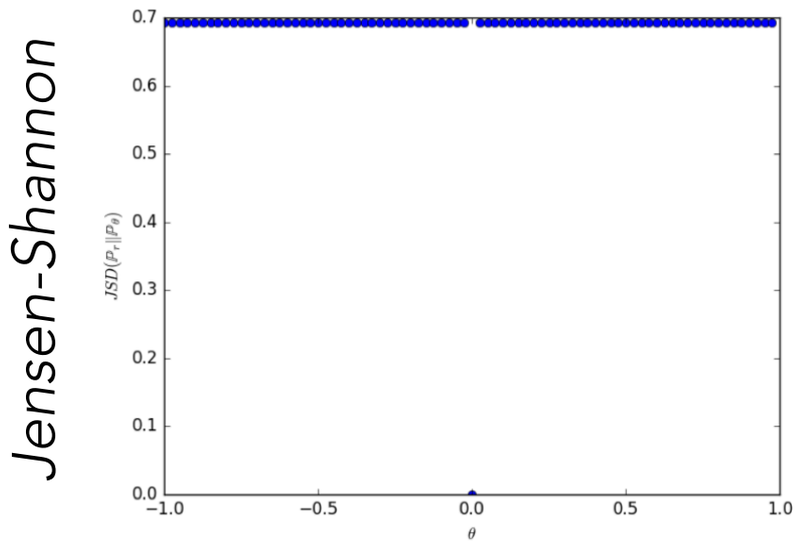
under an *"ideal"* discriminator, we can prove that
the generator minimizes the **Jensen-Shannon divergence**

$$D_{JS}(p(\hat{\mathbf{x}})||p(\tilde{\mathbf{x}})) = \frac{1}{2}D_{KL}(p(\hat{\mathbf{x}})||\frac{1}{2}(p(\hat{\mathbf{x}}) + p(\tilde{\mathbf{x}}))) + \frac{1}{2}D_{KL}(p(\tilde{\mathbf{x}})||\frac{1}{2}(p(\hat{\mathbf{x}}) + p(\tilde{\mathbf{x}})))$$

however, this metric can be
<u>discontinuous</u>, making it difficult to train

$$Z \sim U[0,1] \quad p(\hat{x}) \sim (0,Z) \quad p(\tilde{x}) \sim (\theta, z)$$

$$JS(p(\hat{x})\,||\,p(\tilde{x})) = \begin{cases} \log 2 & if\ \theta \neq 0 \\ 0 & if\ \theta = 0 \end{cases}$$



θ is a gen. model parameter

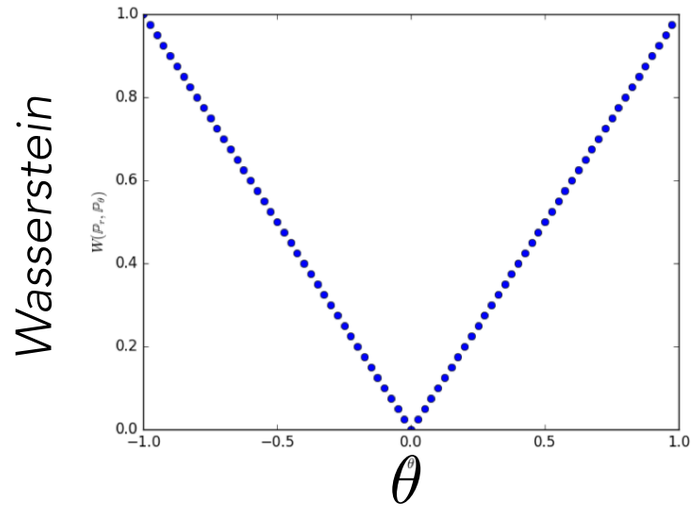
Wasserstein GANs, *Arjovsky et al.*, 2017

Improved Training of Wasserstein GANs, Gulrajani et al., 2017

Wasserstein GAN

Use the Wasserstein (Earth Mover's) distance (continuous and diff. almost everywhere):

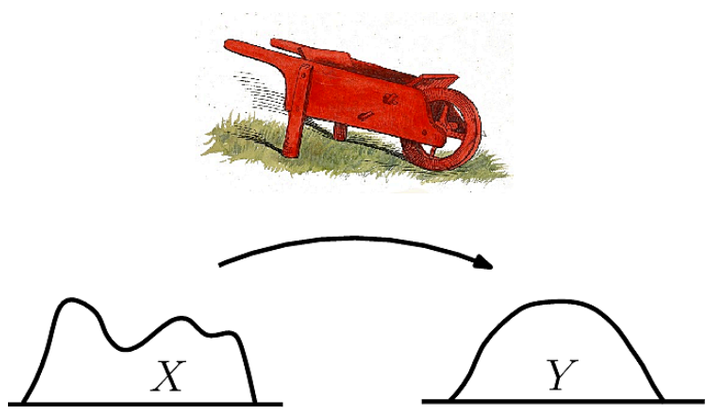
$$W(p(\hat{\mathbf{x}}), p(\tilde{\mathbf{x}})) = \inf_{\gamma \in \Pi(p(\hat{\mathbf{x}}), p(\tilde{\mathbf{x}}))} \mathbb{E}_{(\hat{\mathbf{x}}, \tilde{\mathbf{x}}) \sim \gamma} \left[||\hat{\mathbf{x}} - \tilde{\mathbf{x}}|| \right]$$

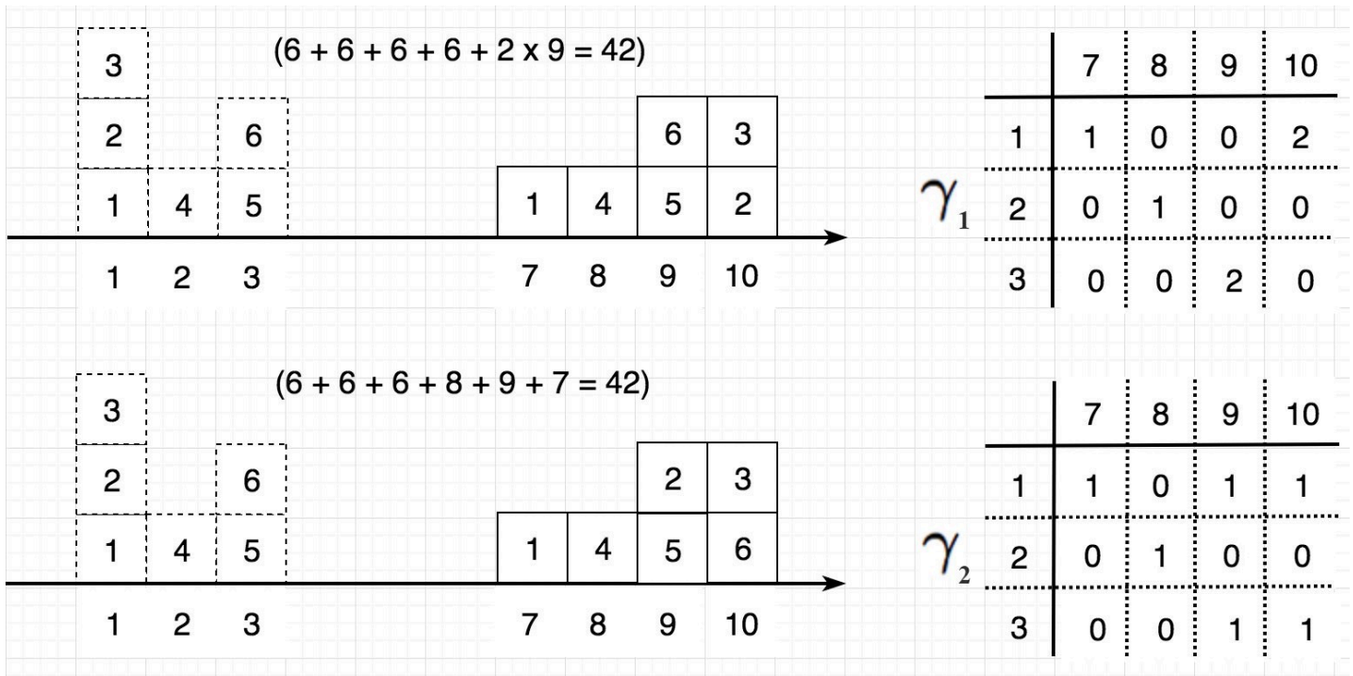


"minimum cost of transporting points between two distributions"

θ is a gen. model parameter

$$W(p(\hat{x}) || p(\tilde{x})) = |\theta|$$





Wasserstein GANs, *Arjovsky et al.*, 2017

Improved Training of Wasserstein GANs, Gulrajani et al., 2017

Wasserstein GAN

Intractable to actually evaluate Wasserstein distance,
but by Kantorovich-Rubinstein duality

$$W(p(\hat{x}), p(\tilde{(x)}) = \sup_{\|f\|_L \leq 1} \mathbb{E}[f(\hat{x})] - \mathbb{E}[f(\tilde{x})]$$

$$|f(x) - f(y)| \leq |x - y|$$

We can evaluate

$$\min_G \max_{D \in \mathcal{D}} \mathbb{E}_{p(\hat{\mathbf{x}})} [D(\hat{\mathbf{x}})] - \mathbb{E}_{p(\tilde{\mathbf{x}})} [D(\tilde{\mathbf{x}})]$$

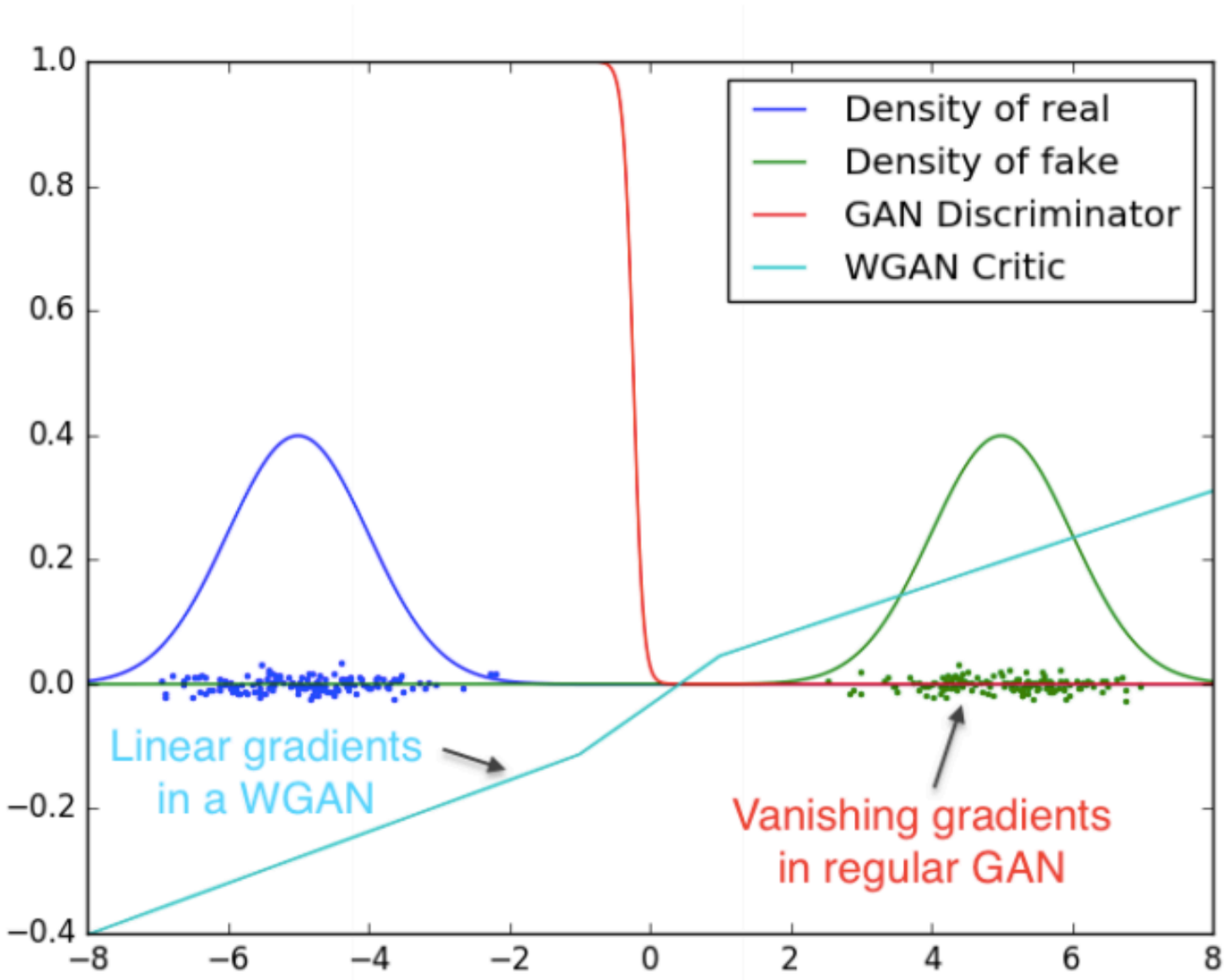
$\mathcal{D}$ is the set of Lipschitz functions, which can be enforced through **weight clipping** or **gradient penalty**

$$w = \text{clip}(w, -c, c)$$

$$x = t\hat{x} + (1 - t)\tilde{x}$$

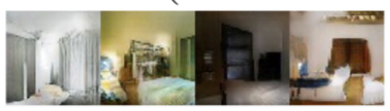
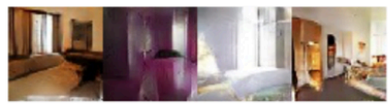
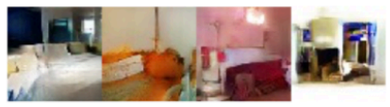
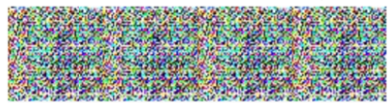
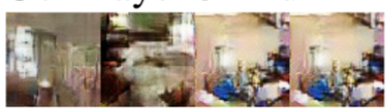
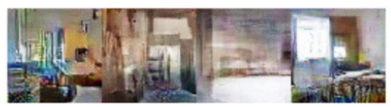
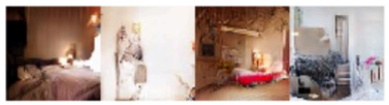
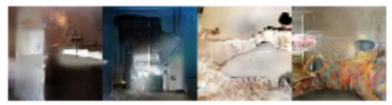
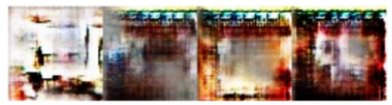
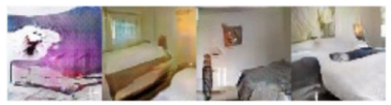
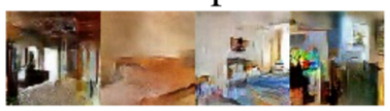
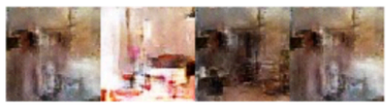
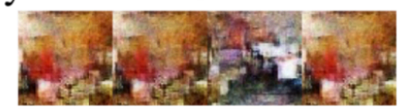
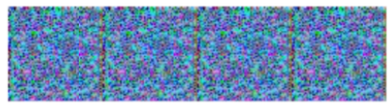
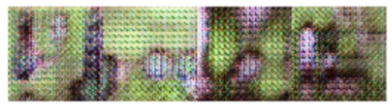
$$V = V + \lambda \mathbb{E}[(\|\nabla D(x)\| - 1)^2]$$

Wasserstein GAN



The discriminator of a minimax GAN saturates and results in vanishing gradients.WGAN critic provides very clean gradients on all parts of the space

Wasserstein GAN

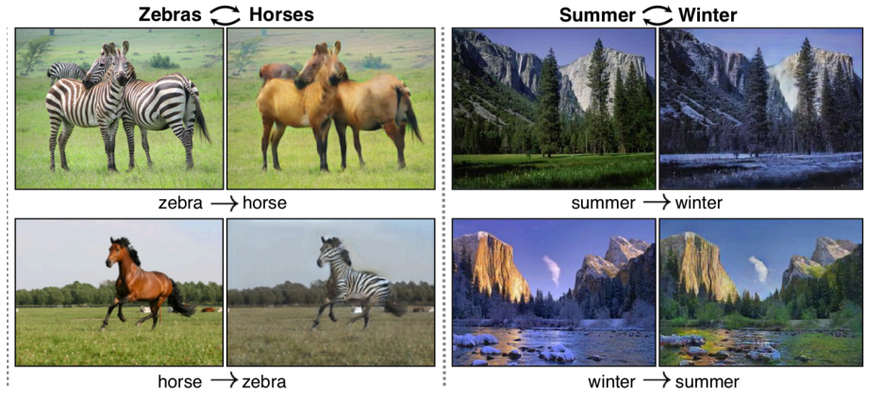
DCGAN	LSGAN	WGAN (clipping)	WGAN-GP (ours)
Baseline (G: DCGAN, D: DCGAN)				
G: No BN and a constant number of filters, D: DCGAN				
G: 4-layer 512-dim ReLU MLP, D: DCGAN				
No normalization in either G or D				
Gated multiplicative nonlinearities everywhere in G and D				
tanh nonlinearities everywhere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

Improved Training of Wasserstein GANs, Gulrajani et al., 2017

Applications of GAN

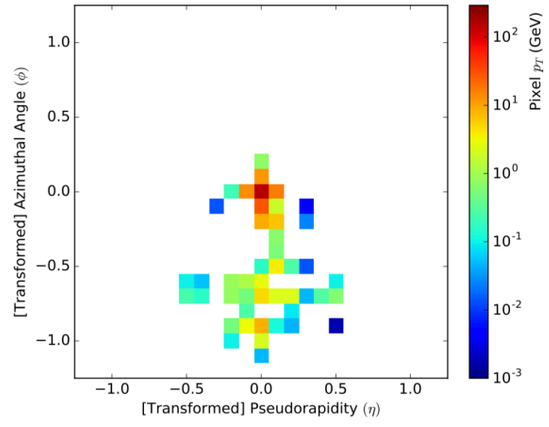
image to image translation

Image-to-Image Translation with Conditional Adversarial Networks, *Isola et al.*, 2016



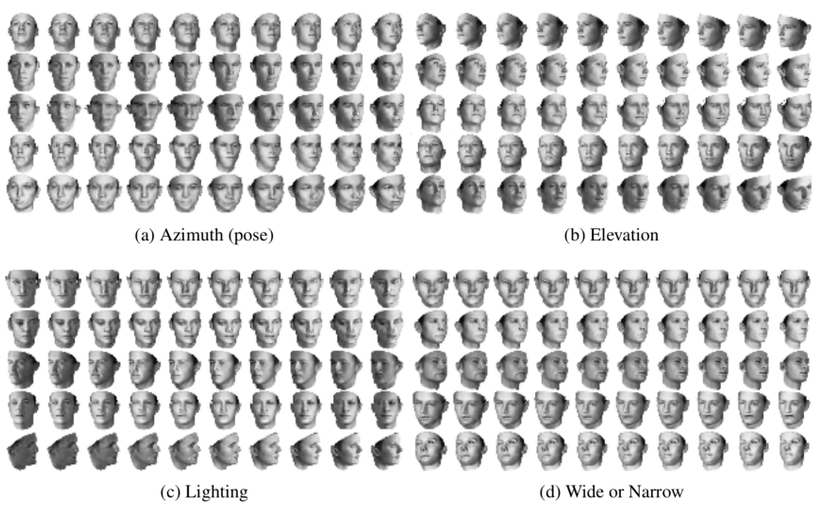
Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, *Zhu et al.*, 2017

experimental simulation



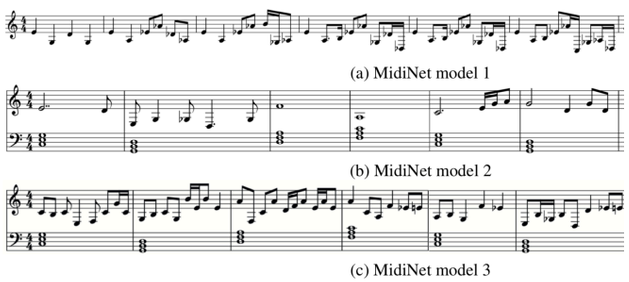
Learning Particle Physics by Example, *de Oliveira et al.*, 2017

interpretable representations



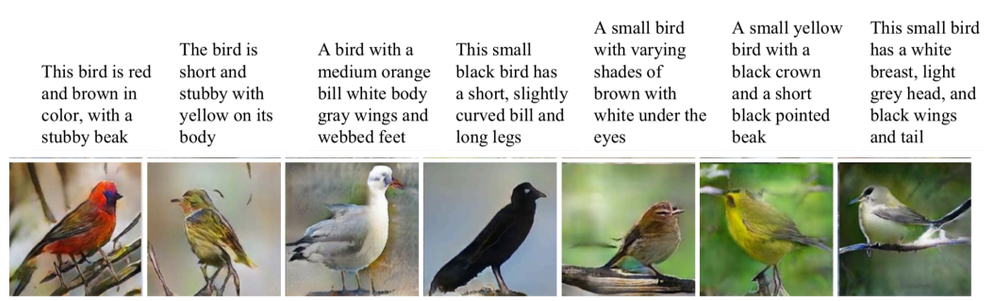
InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets, *Chen et al.*, 2016

music synthesis



MIDINET: A CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORK FOR SYMBOLIC-DOMAIN MUSIC GENERATION, *Yang et al.*, 2017

text to image synthesis



StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks, *Zhang et al.*, 2016