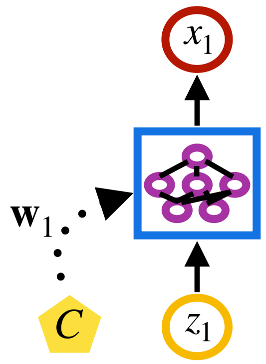
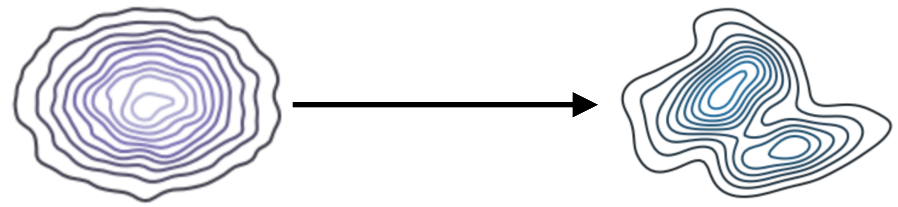
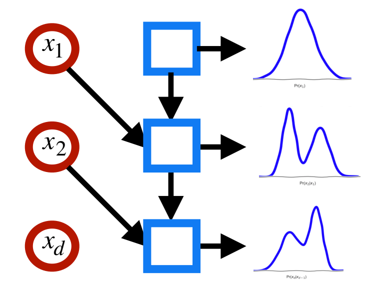






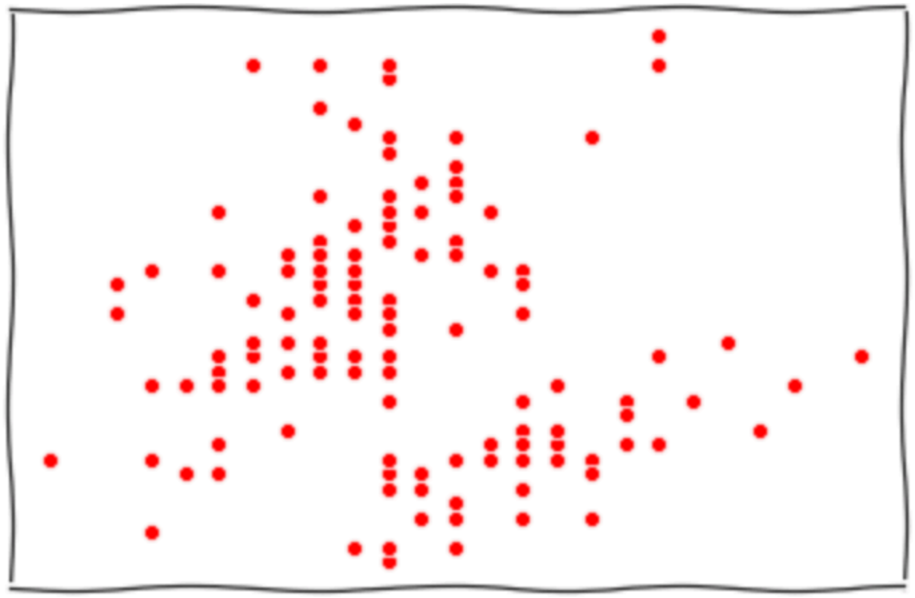
Lec 19: Generative Adversarial Networks

Yaoliang Yu

July 14, 2020

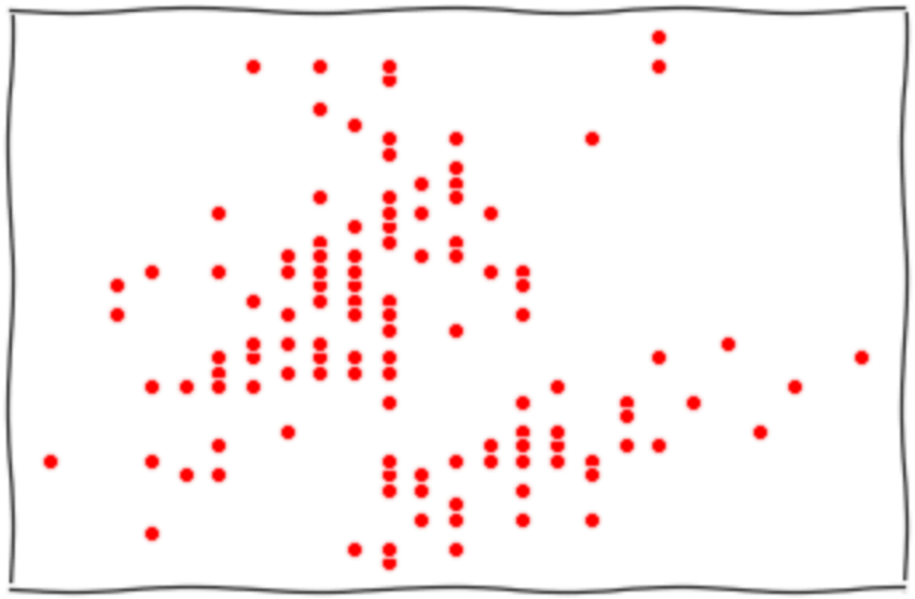
UNIVERSITY OF WATERLOO

density estimation

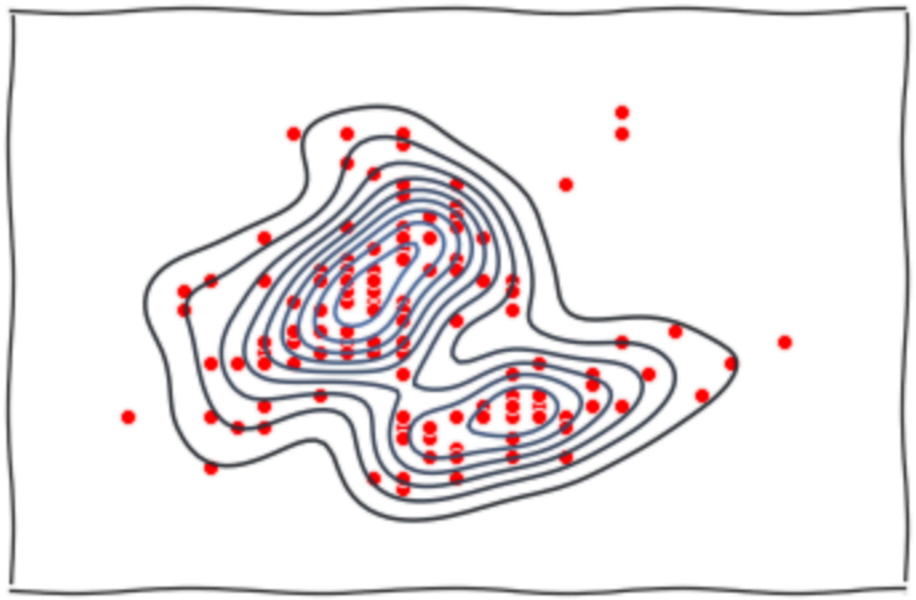


data $= \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$

density estimation



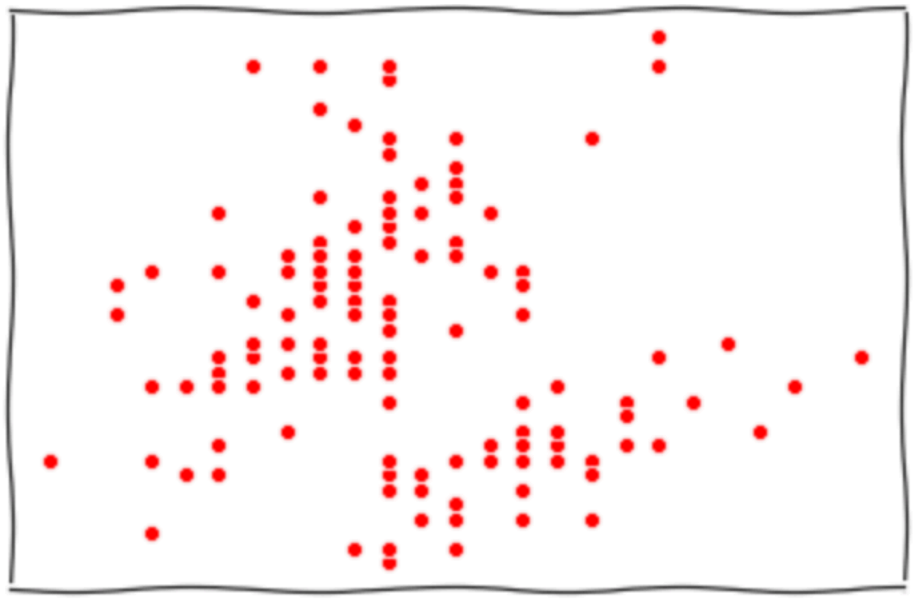
data $= \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$



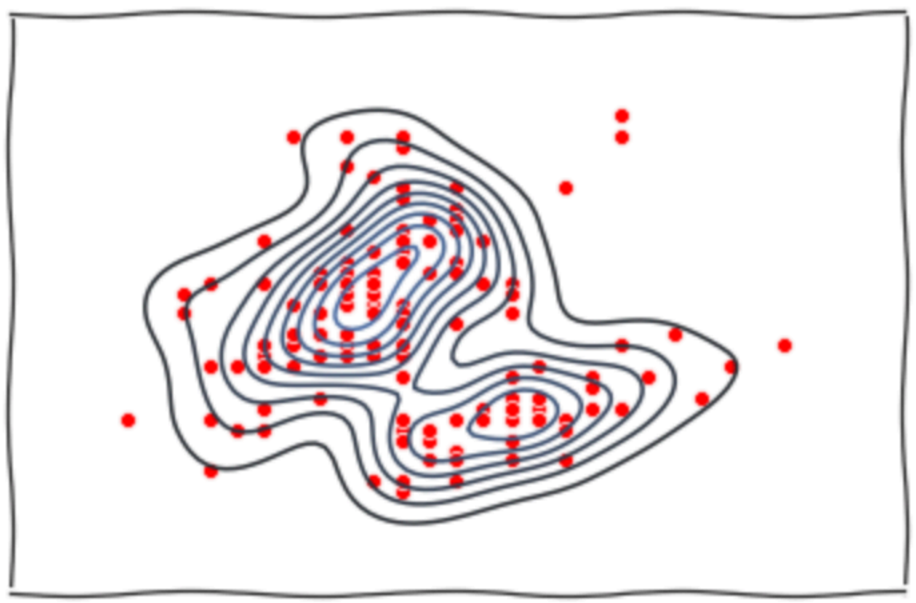
estimate $p(\mathbf{x})$

If we can generate, then we can classify

Review: Maximum Likelihood



data $= \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$



estimate $p(\mathbf{x})$

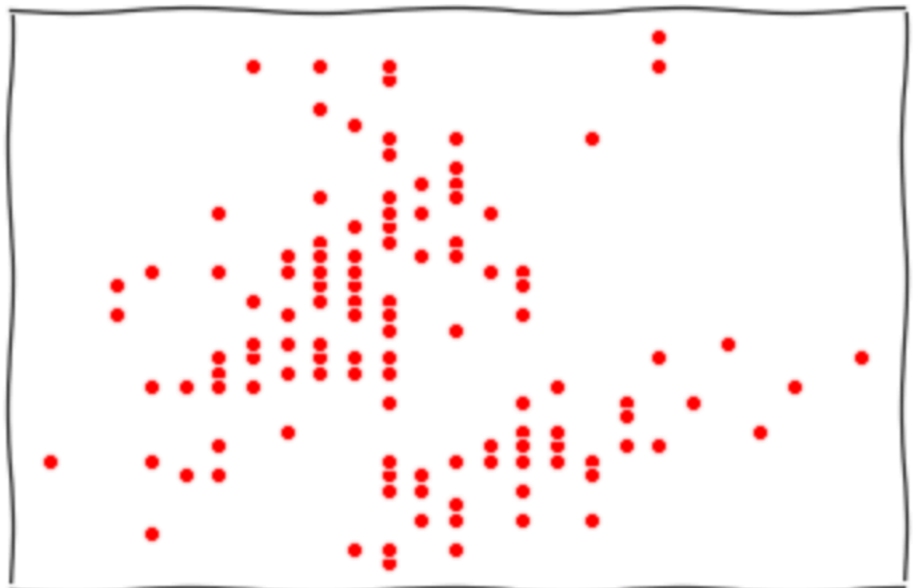
$$\min_{\boldsymbol{\theta}} \ \mathrm{KL}\Big(p(\mathbf{x}) \| q_{\boldsymbol{\theta}}(\mathbf{x})\Big) \equiv \int -\log q_{\boldsymbol{\theta}}(\mathbf{x}) \cdot p(\mathbf{x}) \mathrm{d}\mathbf{x} \approx -\frac{1}{n} \sum_{i=1}^{n} \log q_{\boldsymbol{\theta}}(\mathbf{x}_i)$$

“distance” **unknown truth** **model** **likelihood**

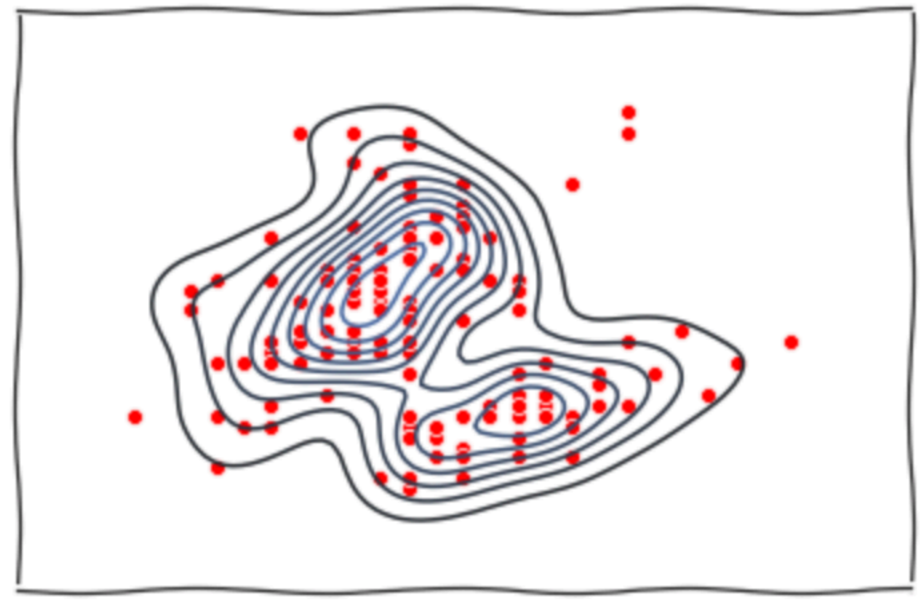
0 iff q=p

explicitly evaluating $q_{\theta}(\mathbf{x})$

Preview: Generative Adversarial Network



data $= \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$

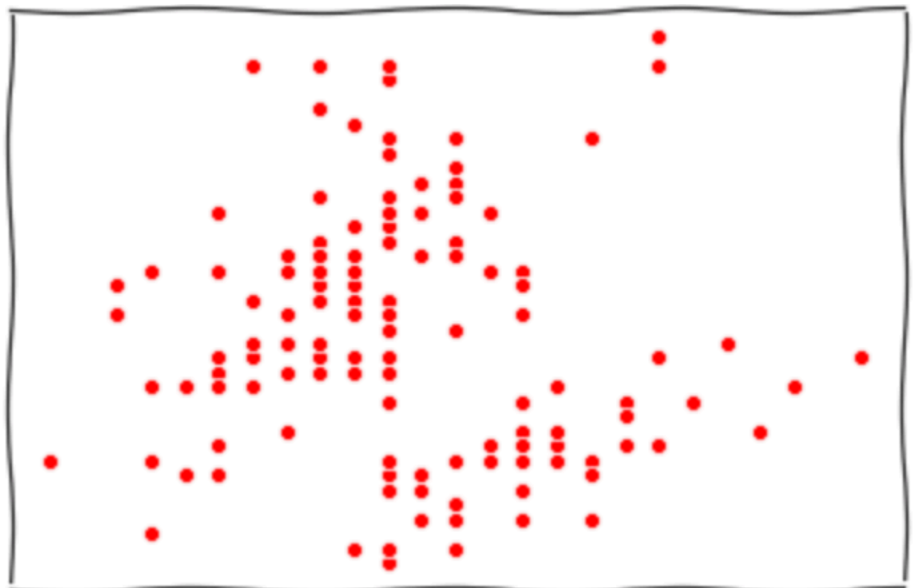


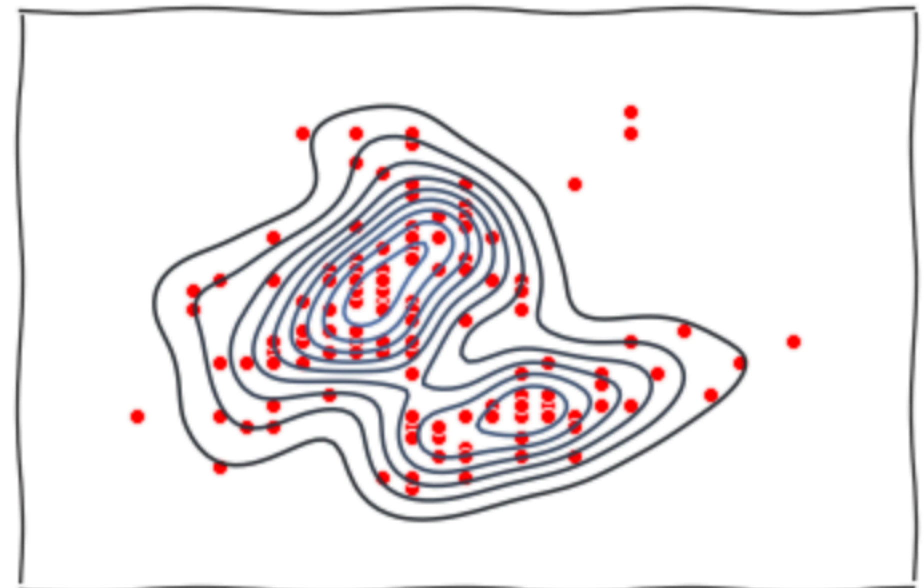
estimate $q(\mathbf{x})$

$$\min_{\boldsymbol{\theta}} \ \mathrm{KL}\Big(p(\mathbf{x}) \| q_{\boldsymbol{\theta}}(\mathbf{x})\Big) \equiv \int -\log q_{\boldsymbol{\theta}}(\mathbf{x}) \cdot p(\mathbf{x}) \mathrm{d}\mathbf{x} \approx -\frac{1}{n} \sum_{i=1}^{n} \log q_{\boldsymbol{\theta}}(\mathbf{x}_i)$$

sampling
explicitly ~~evaluating~~ $q_\theta(\mathbf{x})$

Preview: Generative Adversarial Network





data $= \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$

estimate $p(\mathbf{x})$

$$\min_{\boldsymbol{\theta}} \ \mathrm{KL}\Big(p(\mathbf{x}) \| q_{\boldsymbol{\theta}}(\mathbf{x})\Big) \equiv \int -\log q_{\boldsymbol{\theta}}(\mathbf{x}) \cdot p(\mathbf{x}) \mathrm{d}\mathbf{x} \approx -\frac{1}{n}\sum_{i=1}^{n} \log q_{\boldsymbol{\theta}}(\mathbf{x}_i)$$

$\parallel$

discriminator

$$\max_{S \in \mathcal{S} \subseteq \mathbb{R}^{\mathbb{R}^d}} \int S(\mathbf{x}) \cdot p(\mathbf{x}) \mathrm{d}\mathbf{x} - \int f^*(S(\mathbf{x})) \cdot q_{\boldsymbol{\theta}}(\mathbf{x}) \mathrm{d}\mathbf{x}$$

$\wr\wr$

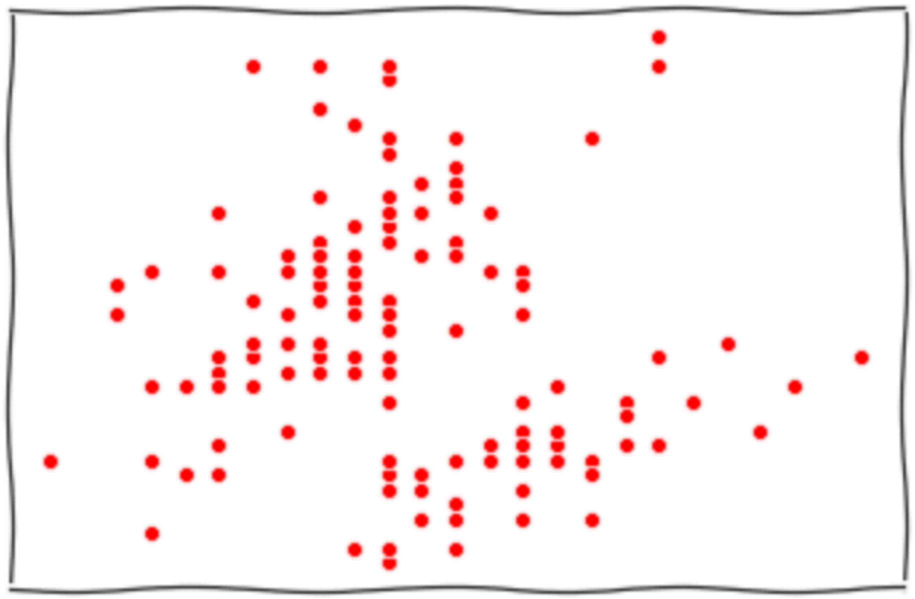
generator

$$\min_{T} \max_{S} \ \frac{1}{n}\sum_{i=1}^{n} S(\mathbf{x}_i) - \frac{1}{m}\sum_{j=1}^{m} f^*(S(T(\mathbf{z}_j)))$$

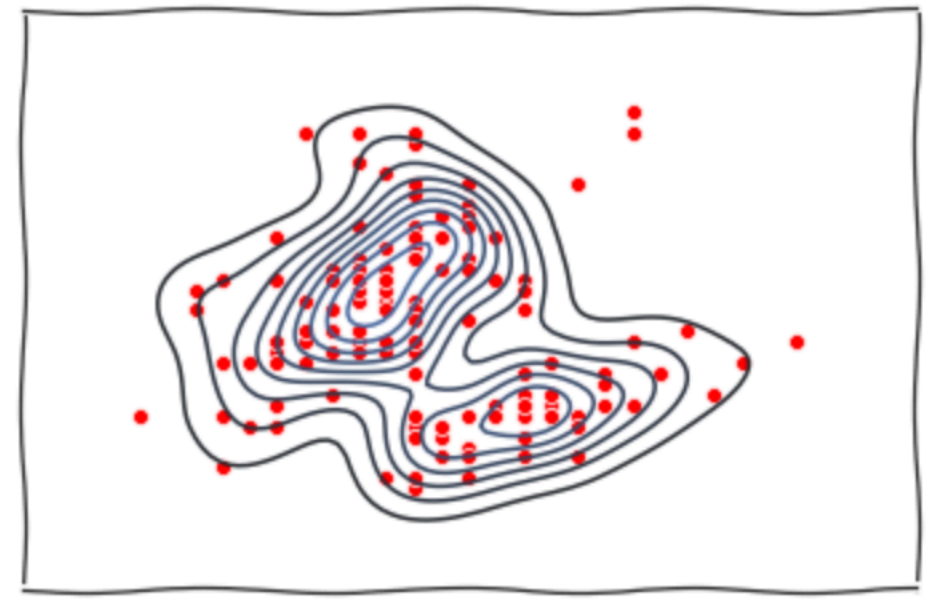
sampling

explicitly ~~evaluating~~ $q_{\theta}(\mathbf{x})$

Review: Expectation-Maximization



data $= \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$



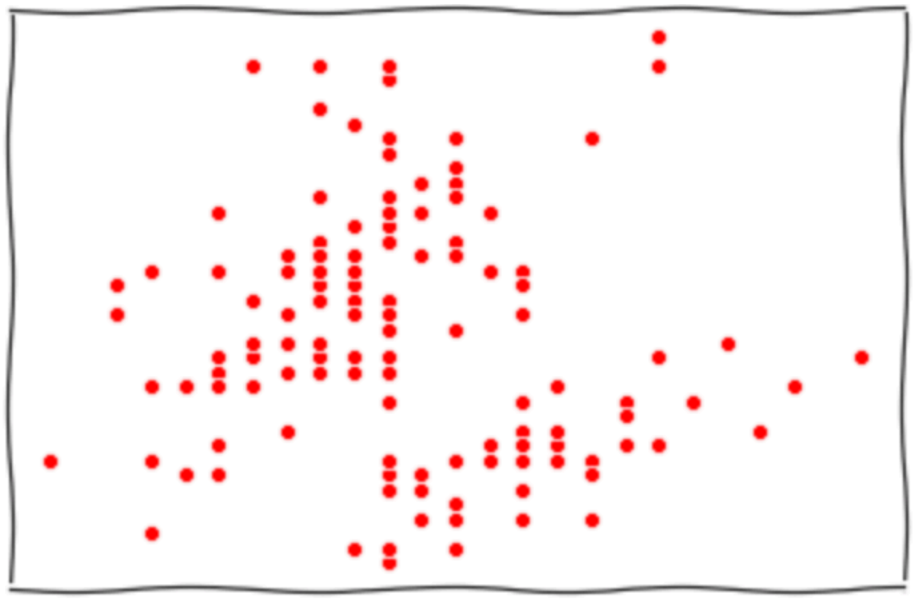
estimate $p(\mathbf{x})$

$$\min_{\boldsymbol{\theta}} \min_{p(\mathbf{z}|\mathbf{x})} \mathrm{KL}\Big(p(\mathbf{x}, \mathbf{z}) \| q_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})\Big) \approx \frac{1}{n}\sum_{i=1}^{n} \int [\log p(\mathbf{z} \,|\, \mathbf{x}_i) - \log q_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{z})] p(\mathbf{z} \,|\, \mathbf{x}_i) \mathrm{d}\mathbf{z}$$

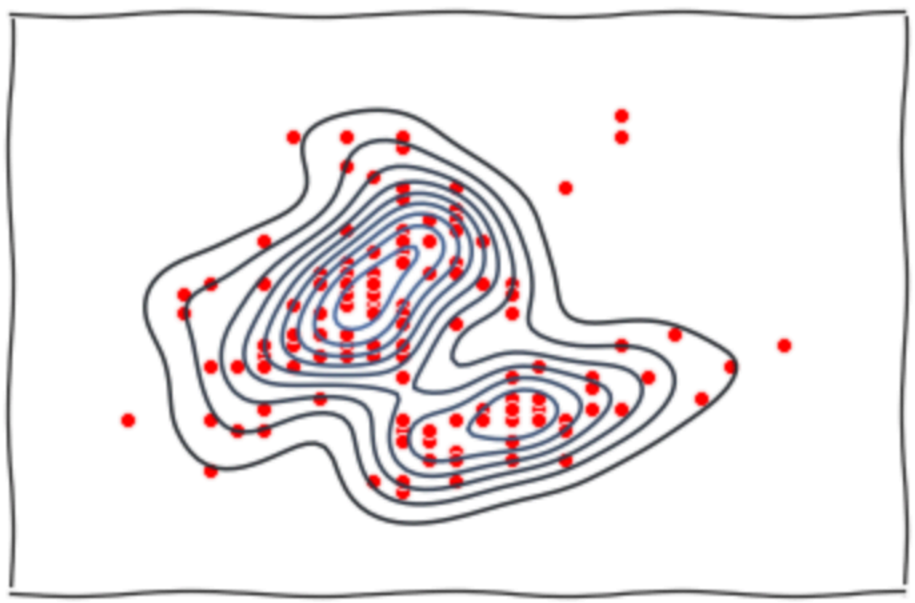
E-step: $p(\mathbf{z} \,|\, \mathbf{x}) = q_{\boldsymbol{\theta}}(\mathbf{z} \,|\, \mathbf{x})$ **explicitly evaluating** $q_{\boldsymbol{\theta}}(\mathbf{z} \,|\, \mathbf{x})$

M-step: $\min_{\boldsymbol{\theta}} -\frac{1}{n}\sum_{i=1}^{n} \int \log q_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{z}) \cdot p(\mathbf{z} \,|\, \mathbf{x}_i) \mathrm{d}\mathbf{z}$ **explicitly evaluating** $q_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$

Monte Carlo EM



data $= \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$



estimate $p(\mathbf{x})$

$$\min_{\boldsymbol{\theta}} \min_{p(\mathbf{z}|\mathbf{x})} \mathrm{KL}\Big(p(\mathbf{x}, \mathbf{z}) \| q_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})\Big) \approx \frac{1}{n}\sum_{i=1}^{n} \int [\log p(\mathbf{z} \,|\, \mathbf{x}_i) - \log q_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{z})] p(\mathbf{z} \,|\, \mathbf{x}_i) \mathrm{d}\mathbf{z}$$

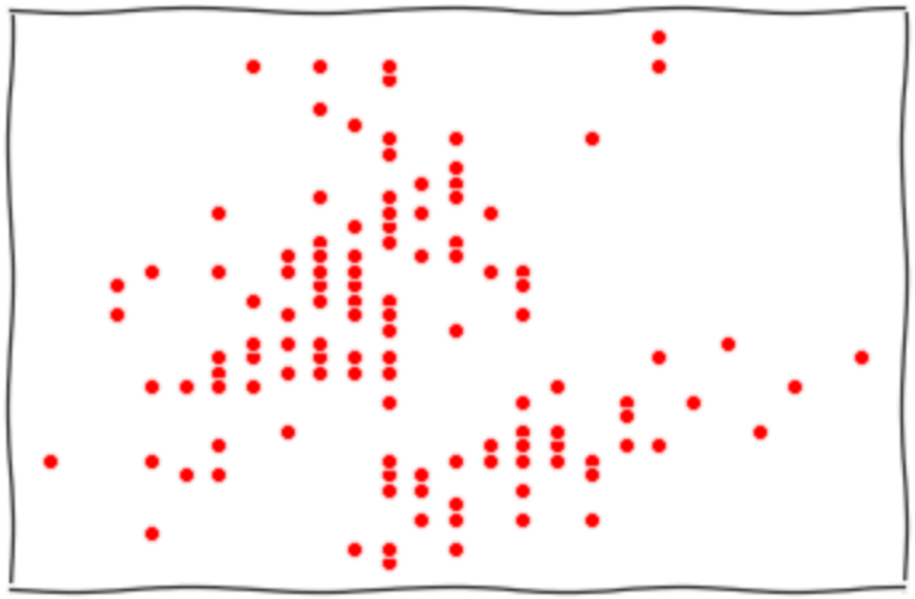
sampling

E-step: $p(\mathbf{z} \,|\, \mathbf{x}) = q_{\boldsymbol{\theta}}(\mathbf{z} \,|\, \mathbf{x})$ explicitly ~~evaluating~~ $q_{\boldsymbol{\theta}}(\mathbf{z} \,|\, \mathbf{x})$

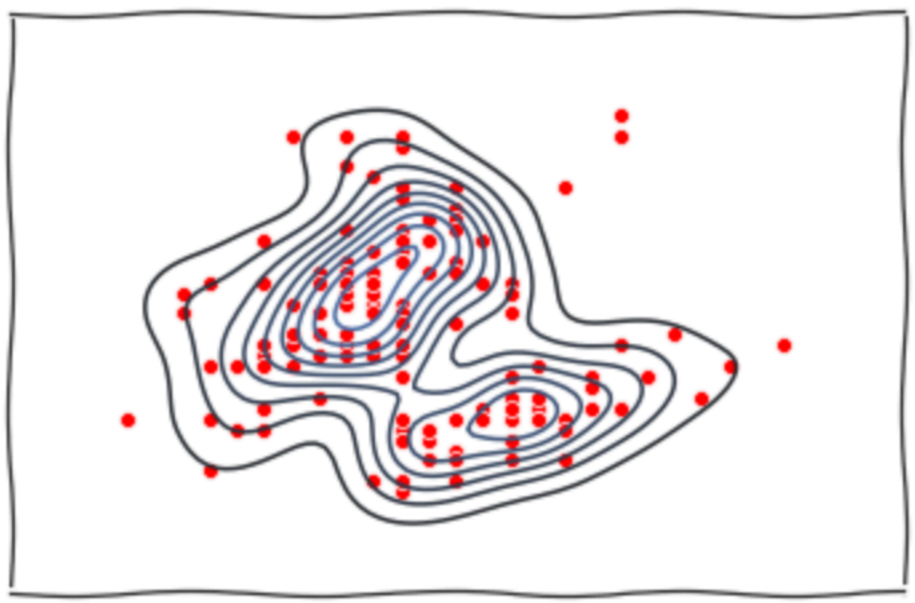
M-step: $\min_{\boldsymbol{\theta}} -\frac{1}{n}\sum_{i=1}^{n} \int \log q_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{z}) \cdot p(\mathbf{z} \,|\, \mathbf{x}_i) \mathrm{d}\mathbf{z} \approx -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} \log q_{\boldsymbol{\theta}}(\mathbf{x}_i, \mathbf{z}_j)$

explicitly evaluating $q_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$

Preview: Variational Auto-Encoder



data $= \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$



estimate $p(\mathbf{x})$

$$\min_{q_\theta(\mathbf{x}|\mathbf{z})} \min_{p_\varphi(\mathbf{z}|\mathbf{x})} \mathrm{KL}\Big(p(\mathbf{x})p_\varphi(\mathbf{z} \mid \mathbf{x}) \| q_\theta(\mathbf{x} \mid \mathbf{z})q(\mathbf{z})\Big)$$

unknown truth **encoder** **decoder** **fixed normal**

explicitly evaluating $q_\theta(\mathbf{x} \mid \mathbf{z})$

explicitly evaluating $p_\varphi(\mathbf{z} \mid \mathbf{x})$

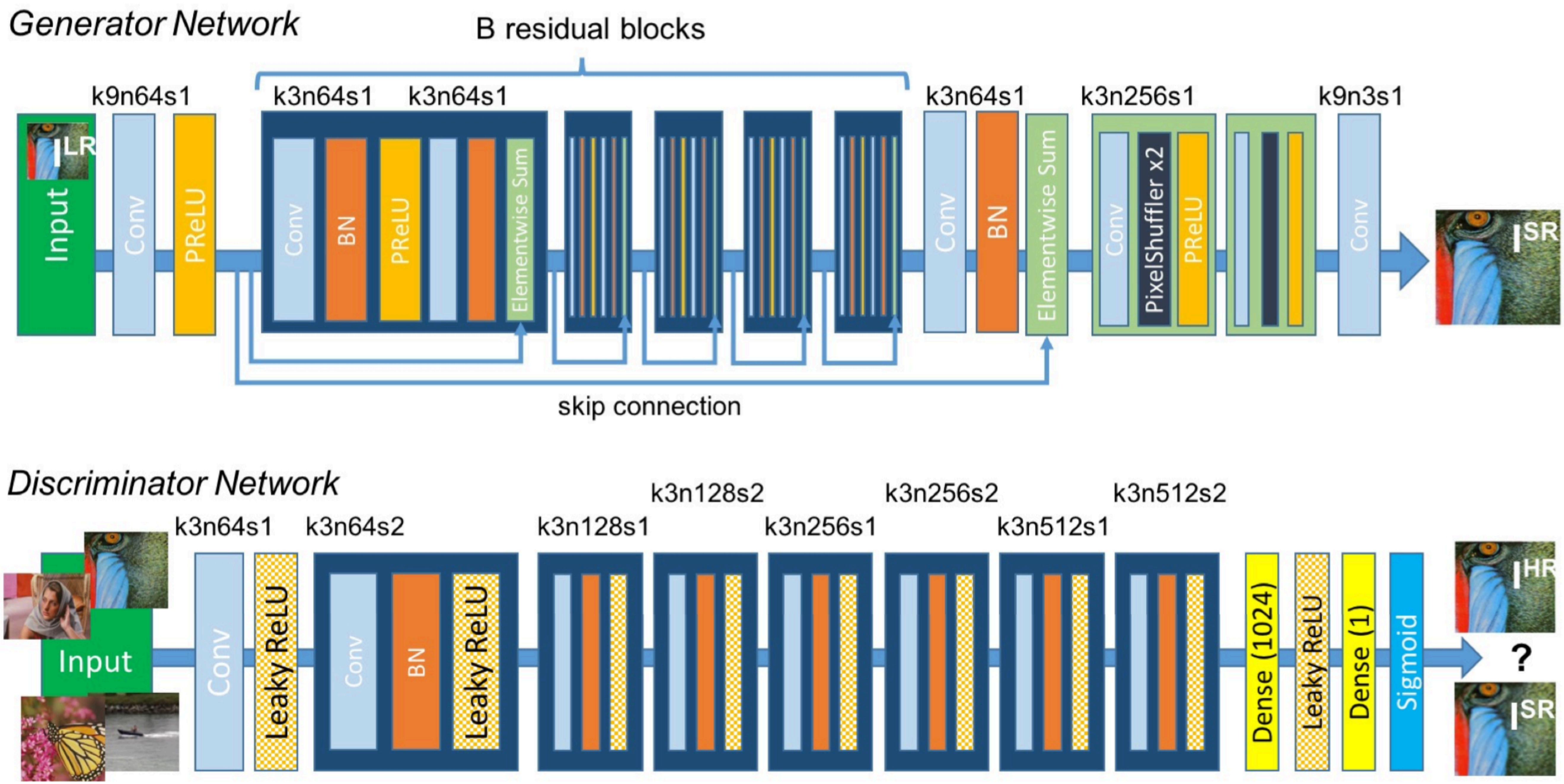
Applications

Image Super-Resolution

bicubic (21.59dB/0.6423) | SRResNet (23.53dB/0.7832) | SRGAN (21.15dB/0.6868) | original

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

Image Super-Resolution



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

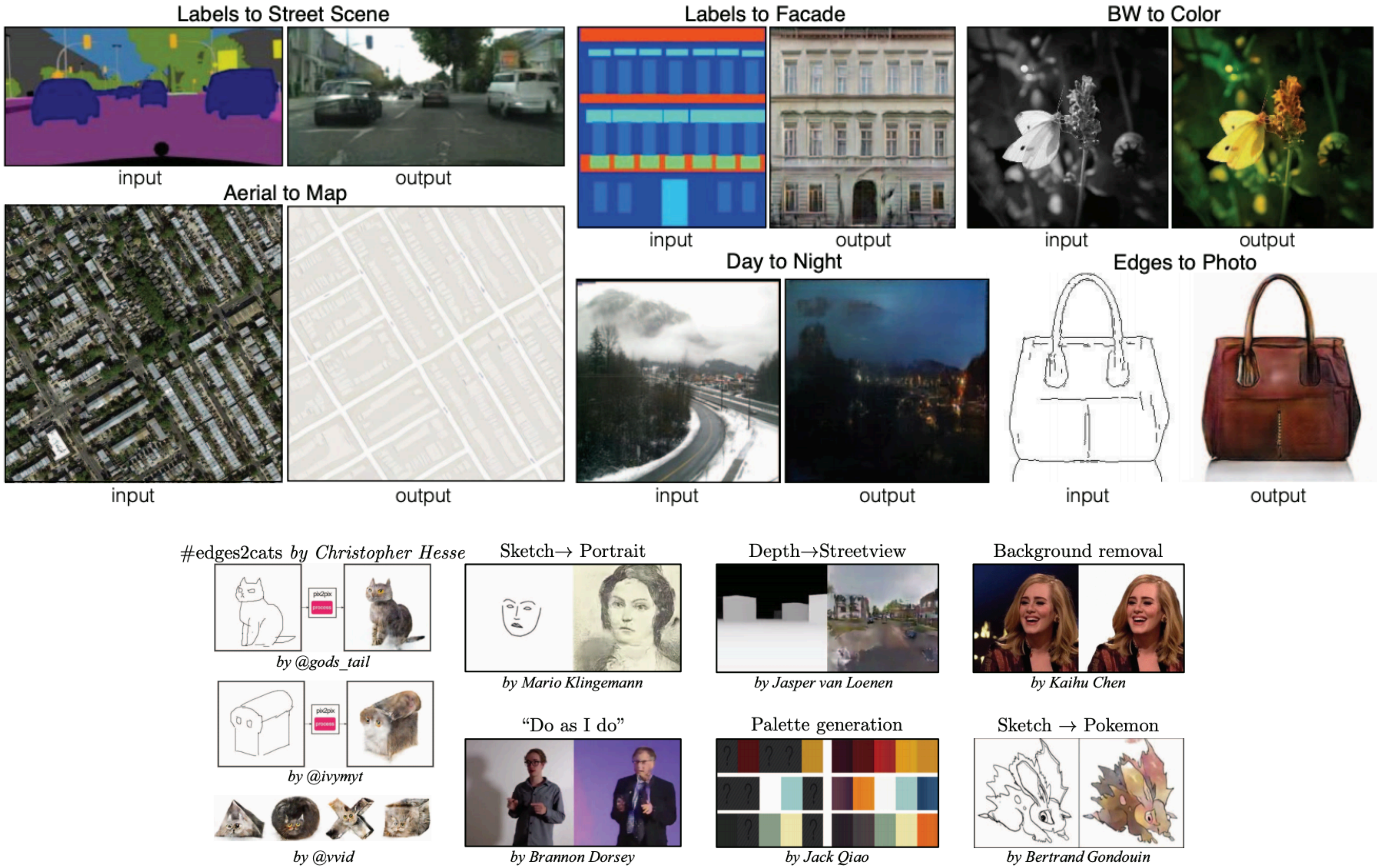
Interactive Image Generation

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

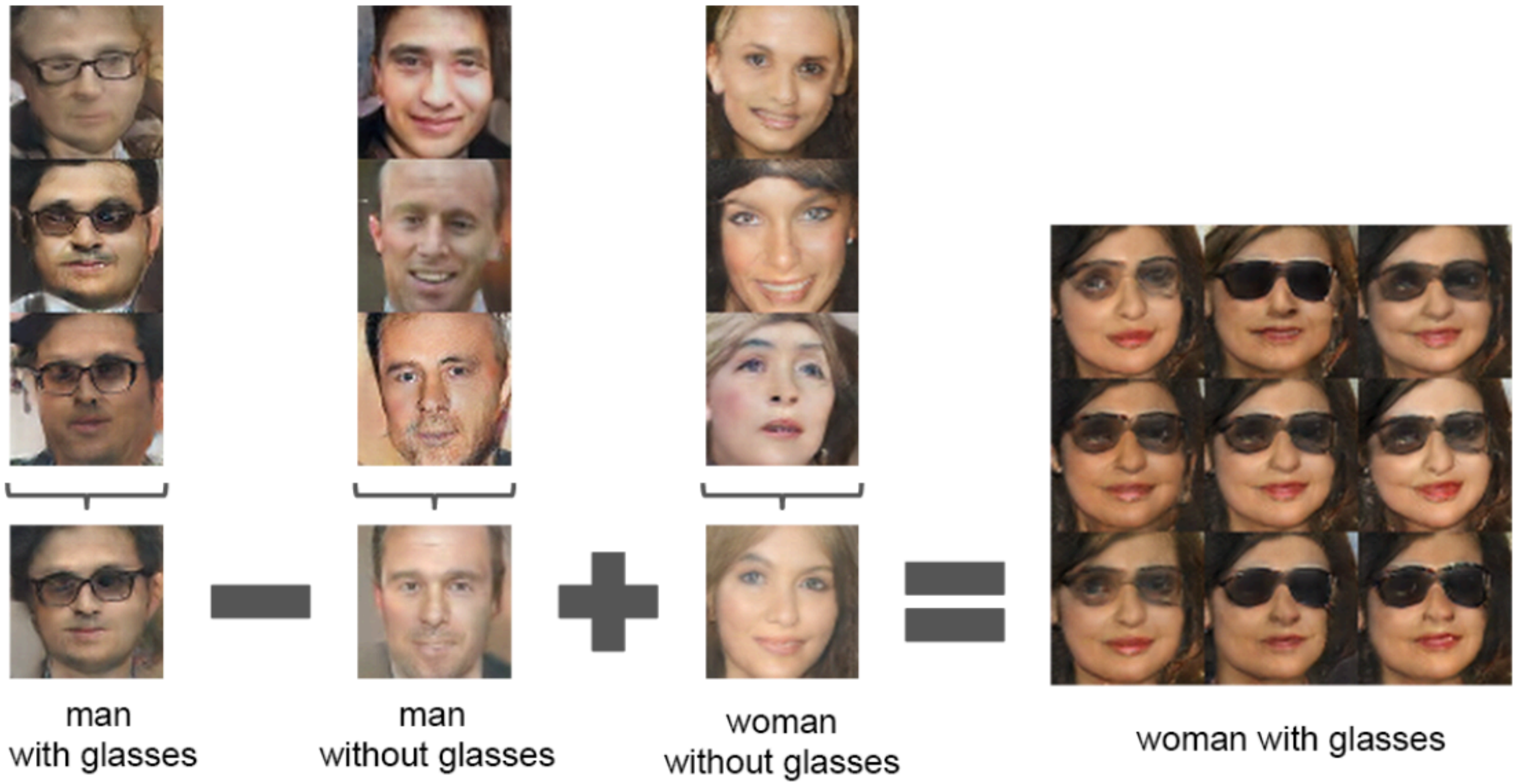
Neural photo editing

Brock et. al. Neural Photo Editing with Introspective Adversarial Networks. ICLR, 2017

Image to Image Translation



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



Radford et. al. Language Models are Unsupervised Multitask Learners. ICLR, 2016

Kingma et. al. Glow:Generative Flow with Invertible 1x1 Convolutions, NeurIPS 2018

Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

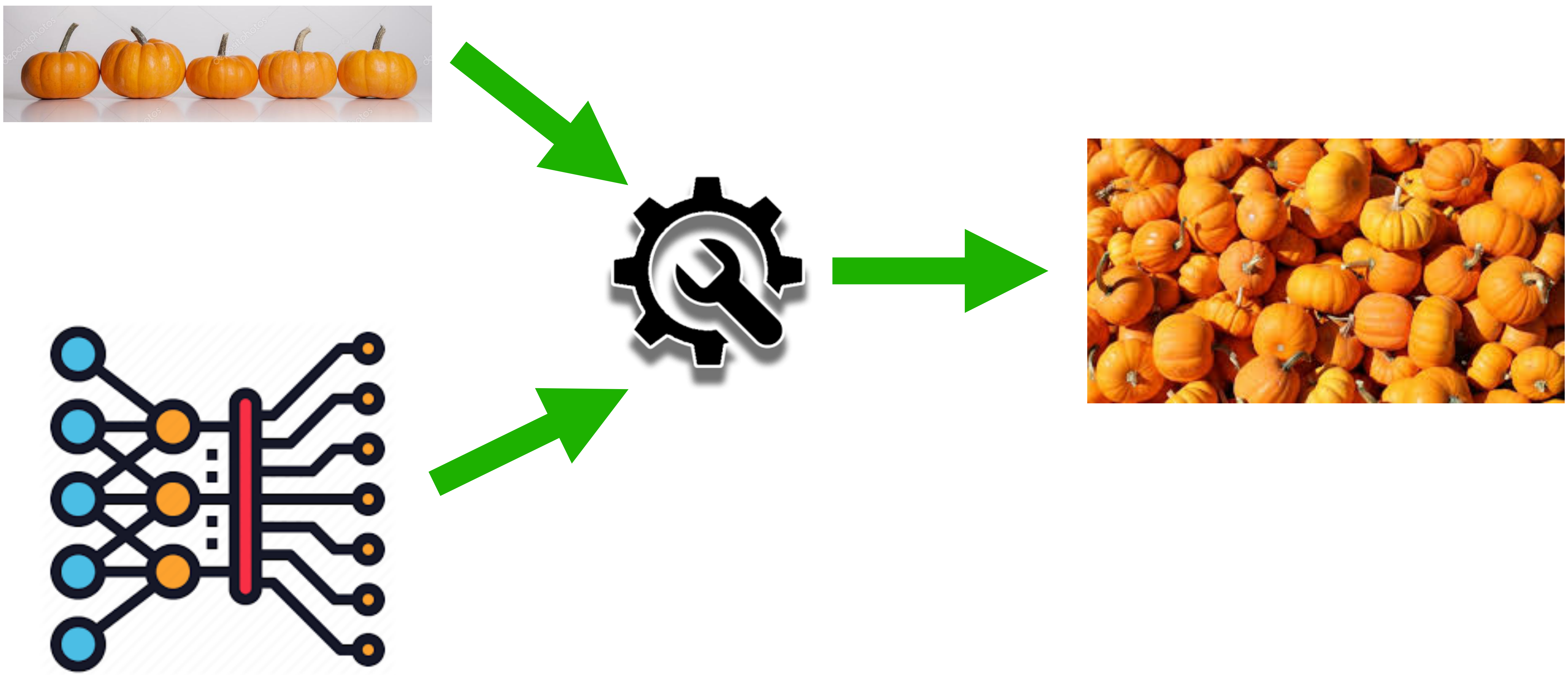
Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America.

While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. According to Pérez, "In South America, such incidents seem to be quite common."

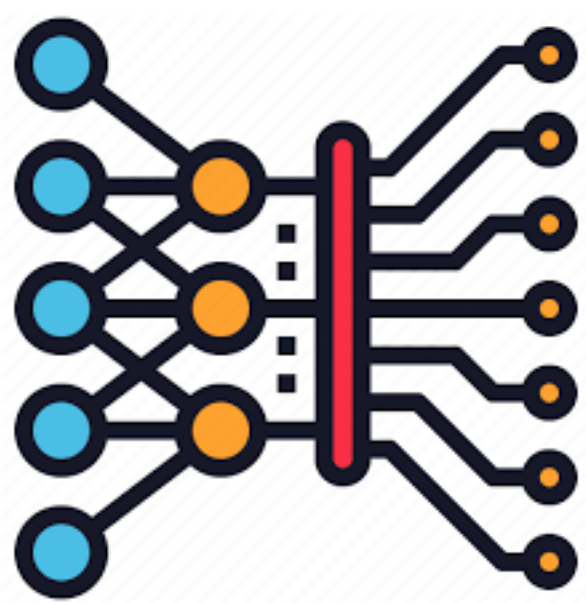
However, Pérez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA. "But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization," said the scientist.

Radford et. al. Language Models are Unsupervised Multitask Learners. 2019

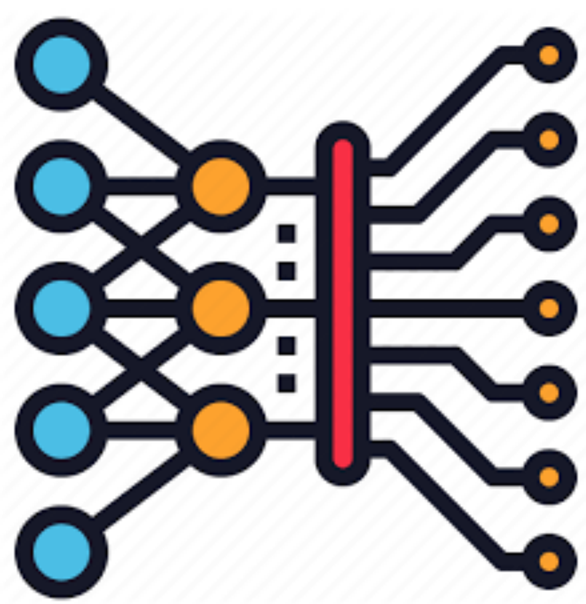
Generative Training?



Generative Training?

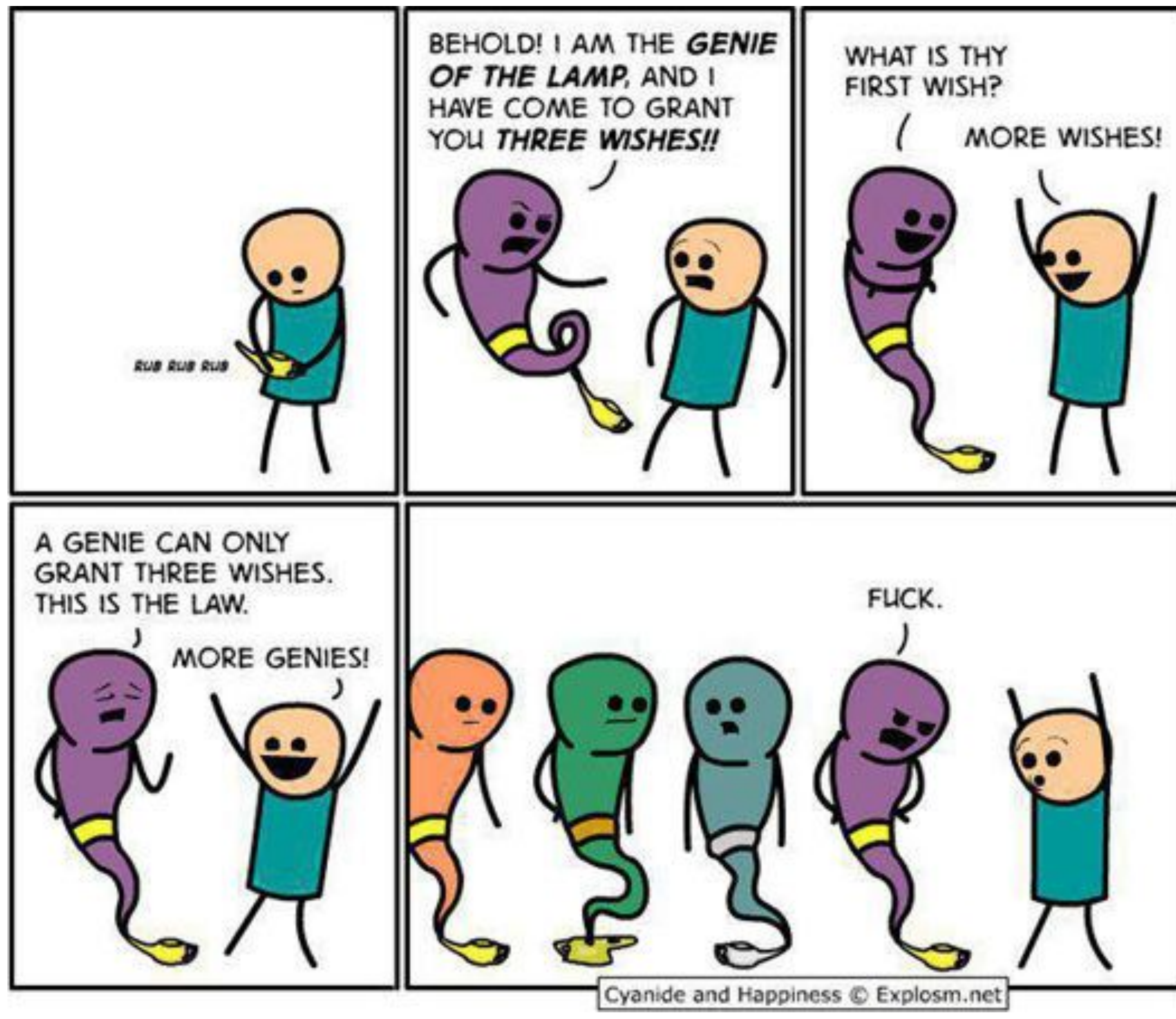


Generative Training?



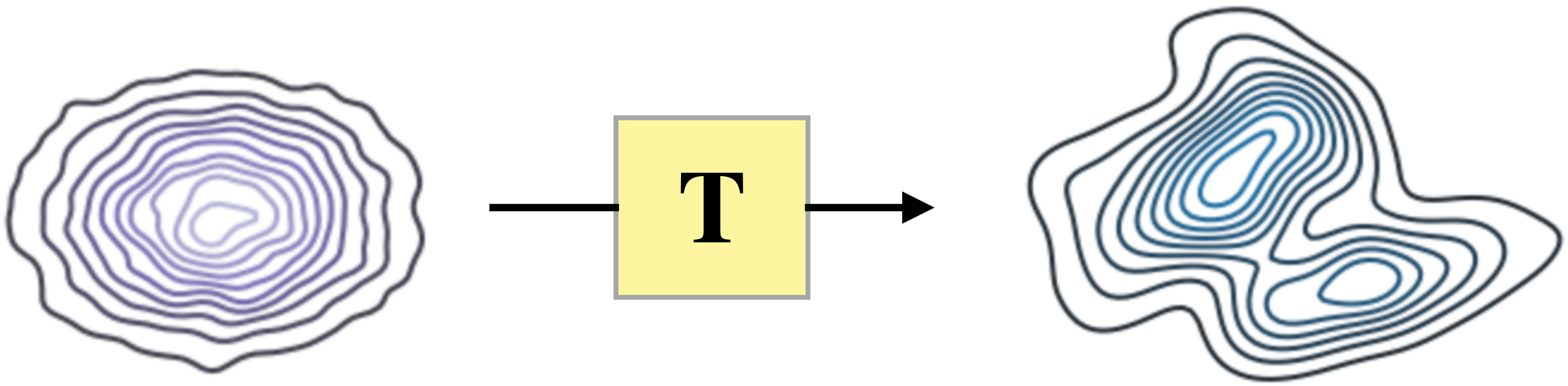
Charles Ponzi
(1882-1949)

Generative Training?



A simple trick

Push-forward



source

$q(\mathbf{z})$

target

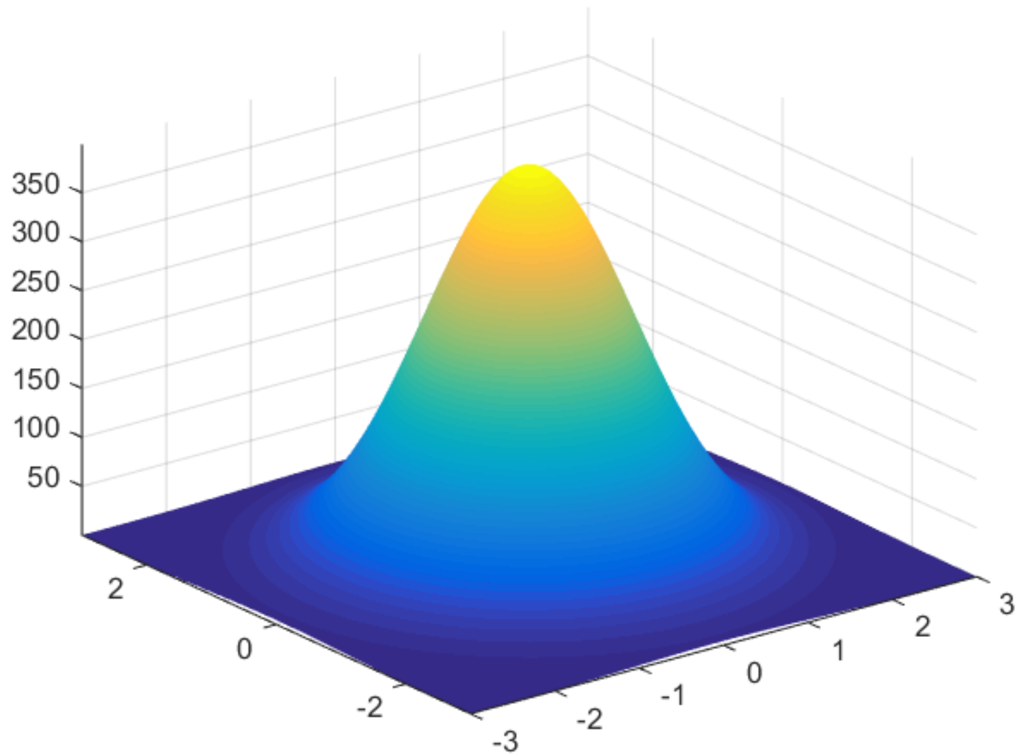
$p(\mathbf{x}) := (\mathbf{T}_{\#}q)(\mathbf{x})$

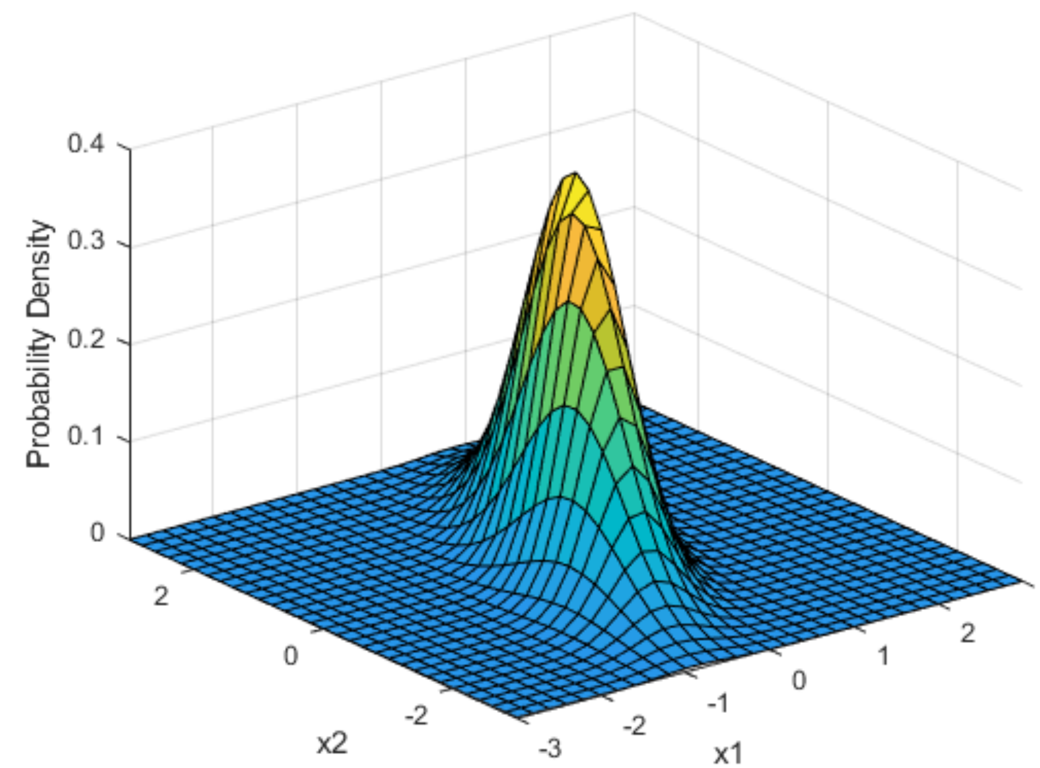
change-of-variable formula: $p(\mathbf{x}) = q(\mathbf{T}^{-1}(\mathbf{x})) \cdot \left| \nabla \mathbf{T}\left(\mathbf{T}^{-1}(\mathbf{x})\right) \right|^{-1}$

to sample x from p(x): sample z from q, then set x = T(z)

parameterize T through a deep network: T(x) = DNN(x; w)

Linear Example



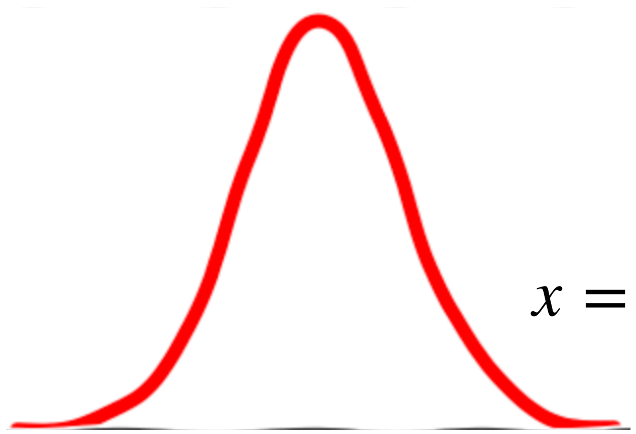


$$TT^{\top} = \Sigma$$

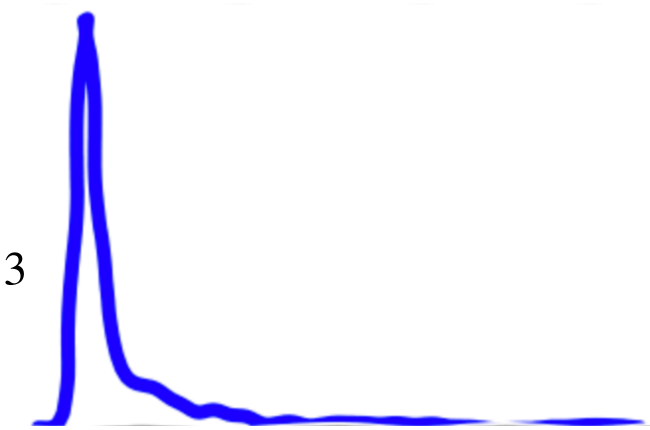
$$Z \sim \mathcal{N}(0, I) \longrightarrow TZ =: X \sim \mathcal{N}(0, \Sigma)$$

unique increasing triangular T = chol(Σ)

Nonlinear Example



$$x = a_0 + a_1 z + a_2 z^2 + a_3 z^3$$

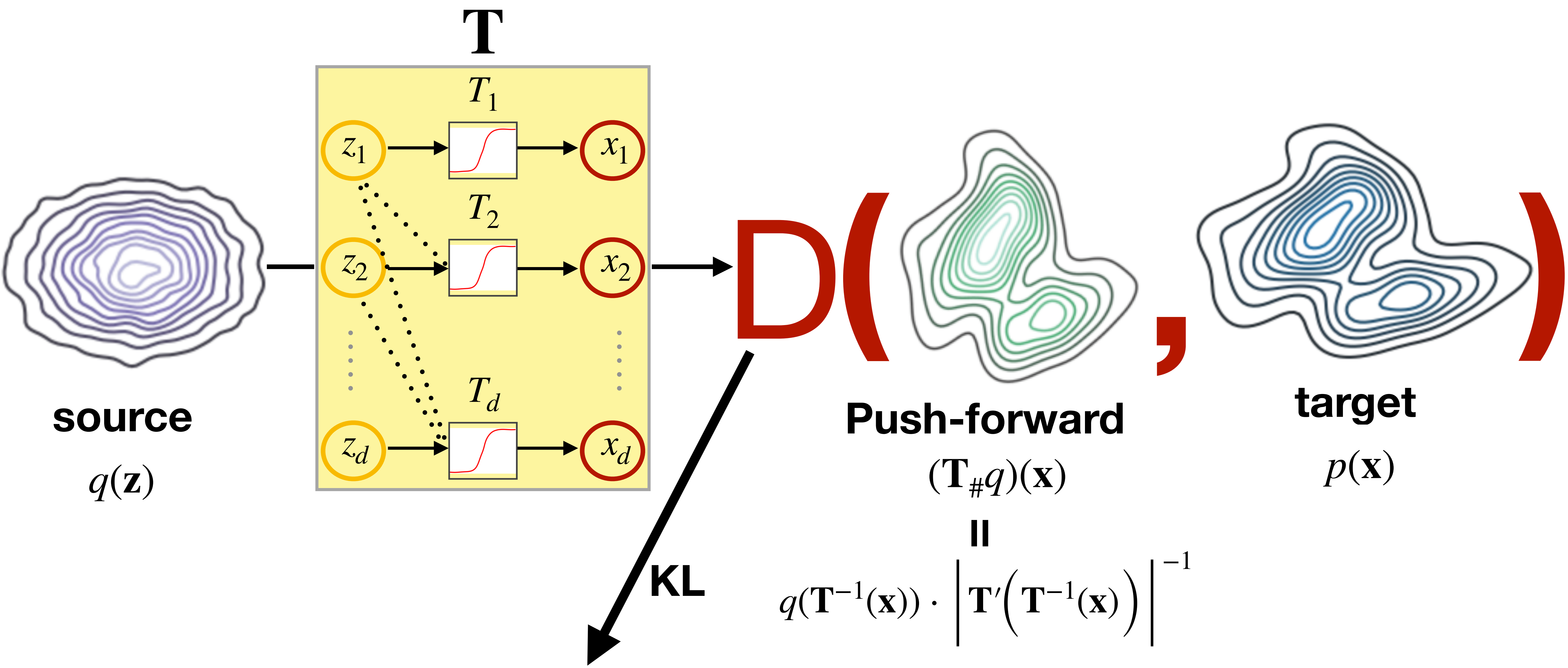


$z \sim \mathcal{N}(0,1)$

$x = \frac{1}{4} z^3 + z^2 + z + 2$

Theorem (roughly): there always exists a (unique increasing triangular) map T that pushes any source density to any target density.

Maximum Likelihood revisited



learn T by maximizing likelihood

$$\min_{\mathbf{T}} \sum_{i=1}^{n} \left[-\log q\left(\mathbf{T}^{-1}(\mathbf{x}_i)\right) + \sum_j \log \partial_j T_j\left(\mathbf{T}^{-1}(\mathbf{x}_i)\right) \right]$$

explicitly evaluating $q_\theta(\mathbf{x})$

Marzouk, Y. et.al. Sampling via Measure Transport: An Introduction, Springer, 2016

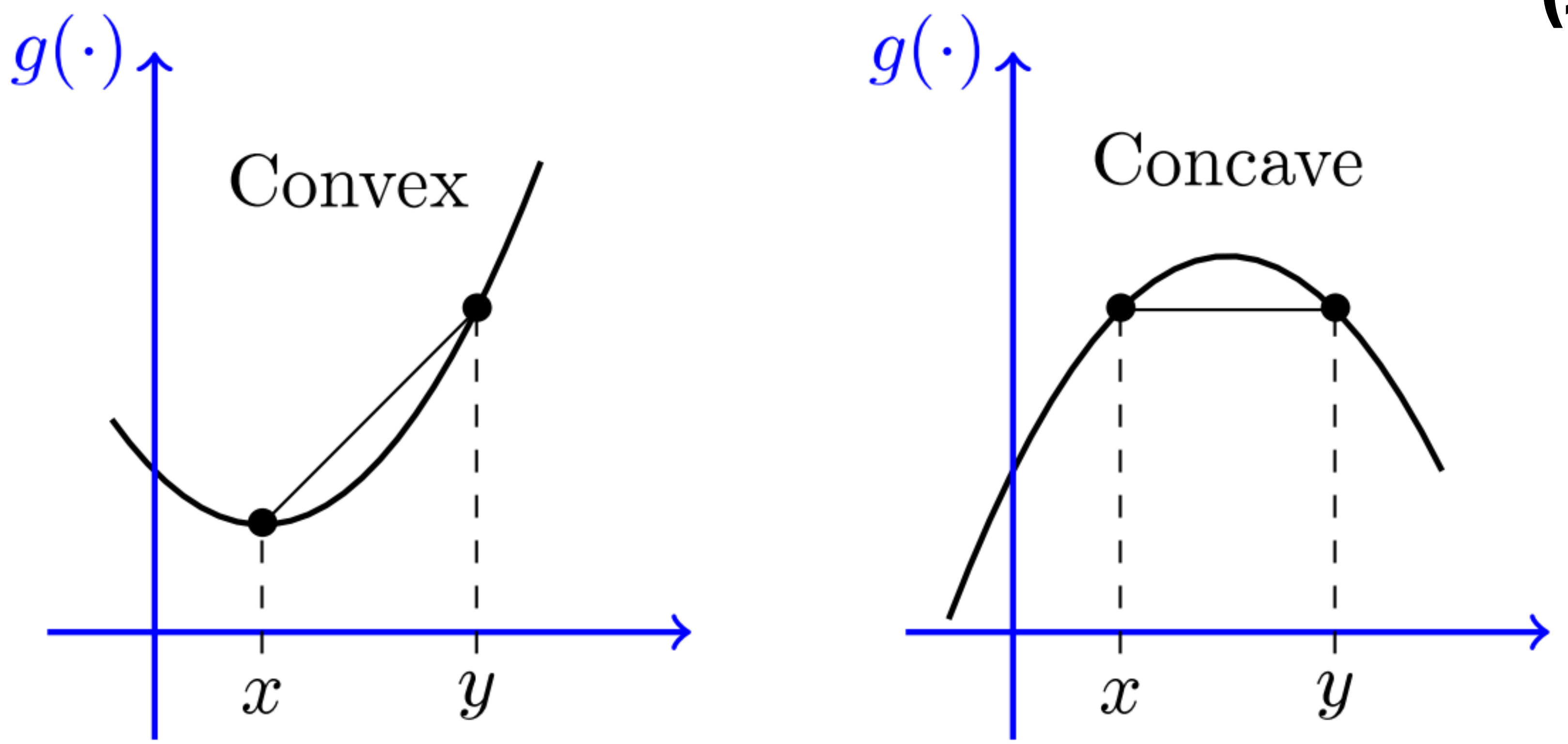
[JSY]. Sum-of-squares Polynomial Flow. ICML, 2019

Another nice tool

Fenchel Conjugate

A univariate real-valued function f is (strictly) convex if $f'' \geq 0$

(>)



The Fenchel conjugate of f is : $f^*(t) = \max_s st - f(s)$, always convex

Theorem: f is convex iff $f^{**} := (f^*)^* = f$

f-divergence

$$D_f(p\|q) := \int f\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) \cdot q(\mathbf{x})\mathrm{d}\mathbf{x}$$

where $f : \mathbb{R}_+ \to \mathbb{R}$ strictly convex and $f(1) = 0$

$$D_f(p\|q) \geq f\left(\int \frac{p(\mathbf{x})}{q(\mathbf{x})} \cdot q(\mathbf{x})\mathrm{d}\mathbf{x}\right) = f(1) = 0$$

equality iff p = q

$$D_f(p\|q) = \max_{S:\mathbb{R}^d \to \mathbb{R}} \int S(\mathbf{x}) \cdot p(\mathbf{x})\mathrm{d}\mathbf{x} - \int f^*(S(\mathbf{x})) \cdot q(\mathbf{x})\mathrm{d}\mathbf{x}$$

$$= \int \max_{S(\mathbf{x})} \left[S(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})} - f^*(S(\mathbf{x}))\right] \cdot q(\mathbf{x})\mathrm{d}\mathbf{x}$$

Examples

Kullback–Leibler

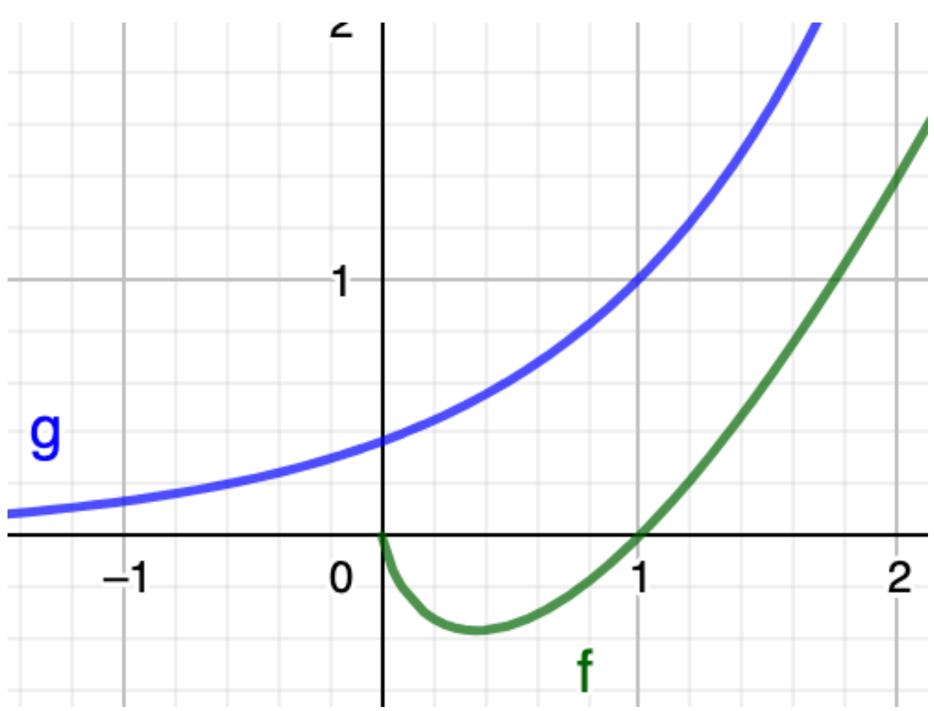
$$f(s) = s \log s$$

$$f''(s) = \frac{1}{s} > 0$$

$$f(1) = 1 \log 1 = 0$$

$$D_f(p\|q) = \int \log\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) \cdot p(\mathbf{x}) \mathrm{d}\mathbf{x}$$

$$f^*(t) = \exp(t-1)$$



Jensen–Shannon

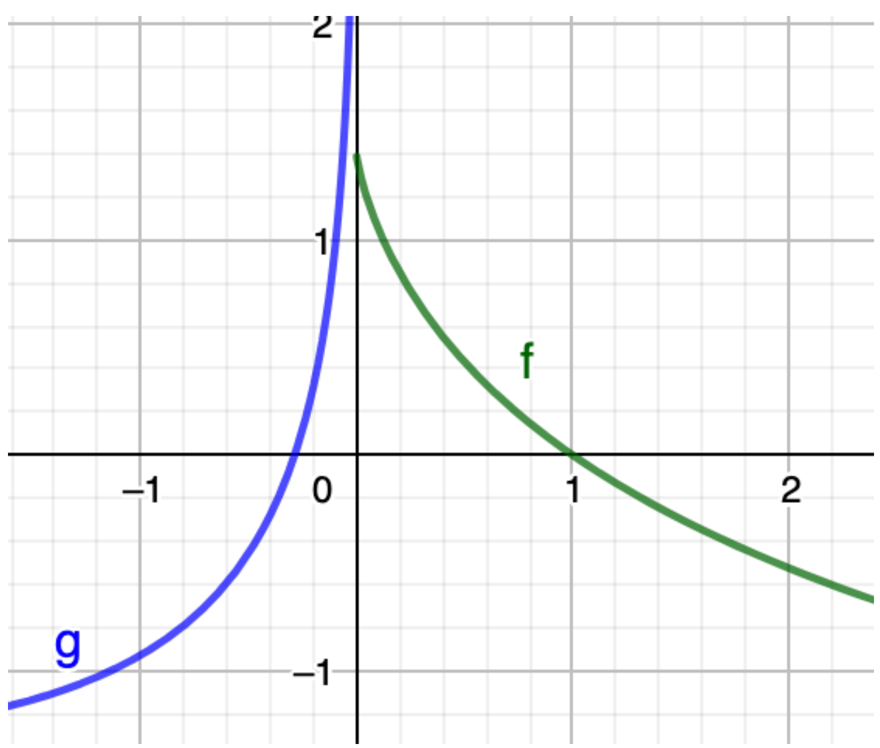
$$f(s) = s \log s - (s+1)\log(s+1) + \log 4$$

$$f''(s) = \frac{1}{s} - \frac{1}{s+1} = \frac{1}{s(s+1)} > 0$$

$$f(1) = -2\log 2 + \log 4 = 0$$

$$D_f(p\|q) = \mathrm{KL}(p\|\frac{p+q}{2}) + \mathrm{KL}(q\|\frac{p+q}{2})$$

$$f^*(t) = -\log(1-\exp(t)) - \log 4$$



Generative Adversarial Networks (GAN)

Goodfellow, I. et.al. Generative Adversarial Networks, NeurIPS, 2014

Putting things together

$$\min_{\boldsymbol{\theta}} \; D_f\Big(p(\mathbf{x}) \| q_{\boldsymbol{\theta}}(\mathbf{x})\Big) = \max_{S \in \mathcal{S} \subseteq \mathbb{R}^{\mathbb{R}^d}} \int S(\mathbf{x}) \cdot p(\mathbf{x}) \mathrm{d}\mathbf{x} - \int f^*(S(\mathbf{x})) \cdot q_{\boldsymbol{\theta}}(\mathbf{x}) \mathrm{d}\mathbf{x}$$

$$\approx$$

$$\min_{T_{\boldsymbol{\theta}}} \max_{S_{\boldsymbol{\varphi}}} \; \frac{1}{n} \sum_{i=1}^{n} S_{\boldsymbol{\varphi}}(\mathbf{x}_i) - \frac{1}{m} \sum_{j=1}^{m} f^*(S_{\boldsymbol{\varphi}}(T_{\boldsymbol{\theta}}(\mathbf{z}_j)))$$

$\tilde{\mathbf{x}} = T_{\theta}(\mathbf{z})$

generator **discriminator** **true sample** **generated "fake" sample**

both parameterized as DNN

sampling

explicitly ~~evaluating~~ $q_\theta(\mathbf{x})$

Example: JS-GAN

$$\min_{T_\theta} \max_{S_\varphi} \frac{1}{n} \sum_{i=1}^{n} S_\varphi(\mathbf{x}_i) - \frac{1}{m} \sum_{j=1}^{m} f^*(S_\varphi(T_\theta(\mathbf{z}_j)))$$

$$D_f(p \| q) = \mathrm{KL}(p \| \frac{p+q}{2}) + \mathrm{KL}(q \| \frac{p+q}{2})$$

$$f^*(t) = -\log(1 - \exp(t)) - \log 4 \qquad t \le 0$$

$S \leftarrow \log S$

$$\min_{T_\theta} \max_{S_\varphi} \frac{1}{n} \sum_{i=1}^{n} \log S_\varphi(\mathbf{x}_i) + \frac{1}{m} \sum_{j=1}^{m} \log(1 - S_\varphi(T_\theta(\mathbf{z}_j)))$$

$0 \le S_\varphi \le 1$

sampling
explicitly ~~evaluating~~ $q_\theta(\mathbf{x})$

Interpreting JS-GAN

Generator tries to "fool" discriminator

$$\min_{T_\theta} \max_{S_\varphi} \; \frac{1}{n}\sum_{i=1}^{n} \log S_\varphi(\mathbf{x}_i) + \frac{1}{m}\sum_{j=1}^{m} \log(1 - S_\varphi(T_\theta(\mathbf{z}_j))) \qquad 0 \le S_\varphi \le 1$$

Discriminator performs nonlinear logistic regression, where $p = S_\varphi$

$$y = \begin{cases} 1, & \text{true sample} \\ -1, & \text{generated "fake" sample} \end{cases}$$

Two-player zero-sum game: at equilibrium true sample = generated "fake" sample

sampling

explicitly ~~evaluating~~ $q_\theta(\mathbf{x})$

Interpreting JS-GAN

$$\min_{T_{\boldsymbol{\theta}}} \max_{S_{\boldsymbol{\varphi}}} \ \frac{1}{n}\sum_{i=1}^{n} \log S_{\boldsymbol{\varphi}}(\mathbf{x}_i) + \frac{1}{m}\sum_{j=1}^{m} \log(1 - S_{\boldsymbol{\varphi}}(T_{\boldsymbol{\theta}}(\mathbf{z}_j))$$

Two-player zero-sum game: at equilibrium true sample = generated “fake” sample

After training

$$\min_{T_\theta} \max_{S_\varphi} \; \frac{1}{n}\sum_{i=1}^{n} \log S_\varphi(\mathbf{x}_i) + \frac{1}{m}\sum_{j=1}^{m} \log(1 - S_\varphi(T_\theta(\mathbf{z}_j)))$$

Two-player zero-sum game: at equilibrium true sample = generated "fake" sample

Strong/Weak duality

$$\min_{T_\theta} \max_{S_\varphi} \frac{1}{n}\sum_{i=1}^{n} \log S_\varphi(\mathbf{x}_i) + \frac{1}{m}\sum_{j=1}^{m} \log(1 - S_\varphi(T_\theta(\mathbf{z}_j))) \qquad 0 \le S_\varphi \le 1$$

$\wr\wr$ $n, m \to \infty$ **arbitrary** T_θ, S_φ

$$\max_{S_\varphi} \min_{T_\theta} \frac{1}{n}\sum_{i=1}^{n} \log S_\varphi(\mathbf{x}_i) + \frac{1}{m}\sum_{j=1}^{m} \log(1 - S_\varphi(T_\theta(\mathbf{z}_j)))$$

Two-player zero-sum game: at equilibrium true sample = generated “fake” sample

sampling

explicitly ~~**evaluating**~~ $q_\theta(\mathbf{x})$

More GANs through IPM

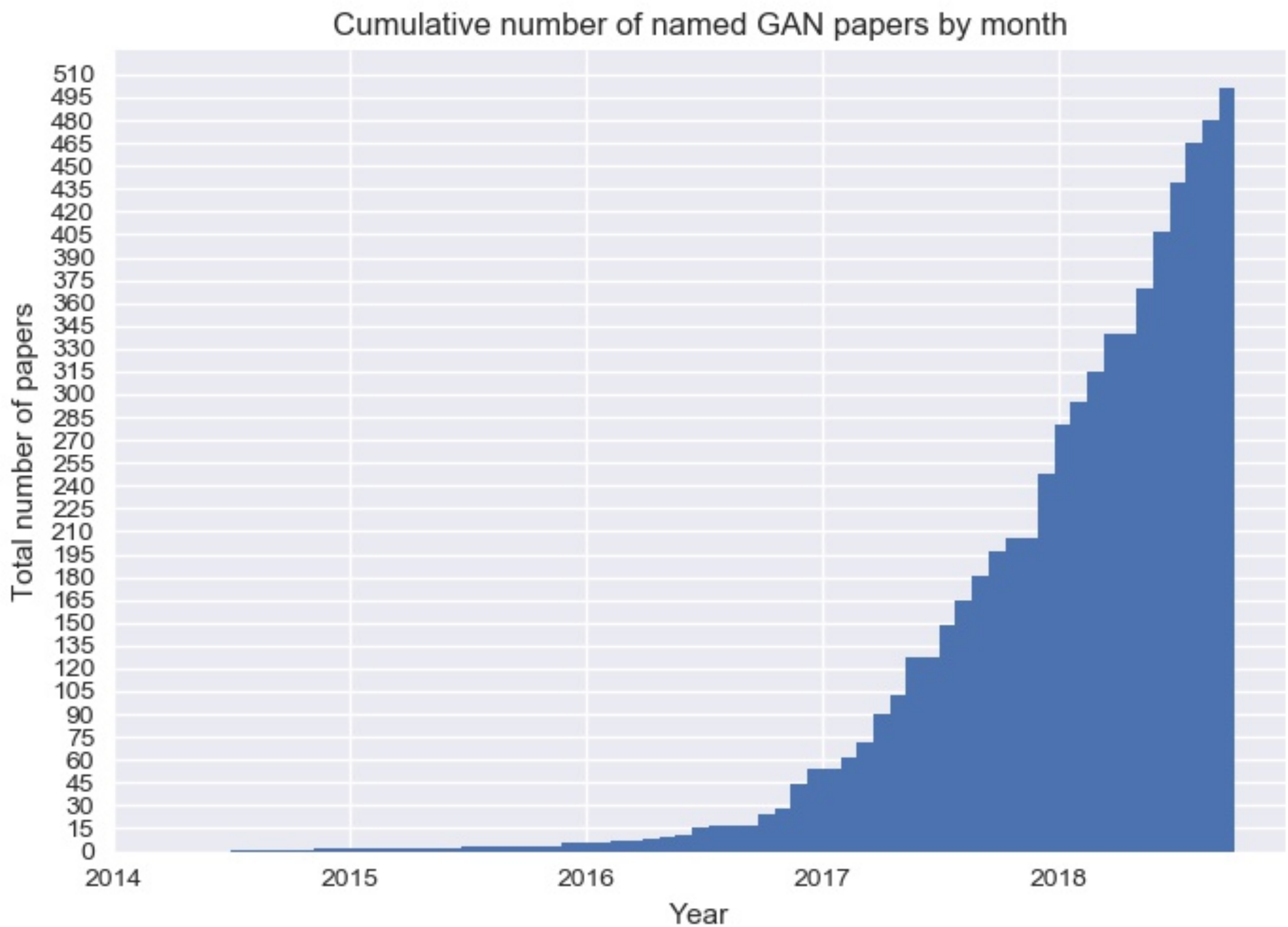
$$\min_{T_\theta} \max_{S_\varphi} \frac{1}{n}\sum_{i=1}^{n} S_\varphi(\mathbf{x}_i) - \frac{1}{m}\sum_{j=1}^{m} S_\varphi(T_\theta(\mathbf{z}_j))$$

- **S is the class of Lipschitz continuous functions: Wasserstein GAN**
- **S is the unit ball of some RKHS: MMD-GAN**
- **S is the class of indicator functions: TV-GAN**
- **S is the unit ball of some Sobolev space: Sobolev-GAN**
- **S is the class of differential functions: Stein-GAN**
- **……**
- **……**

Two-player zero-sum game: at equilibrium true sample = generated "fake" sample

sampling
explicitly **~~evaluating~~** $q_\theta(\mathbf{x})$

The GAN Zoo



https://github.com/hindupuravinash/the-gan-zoo