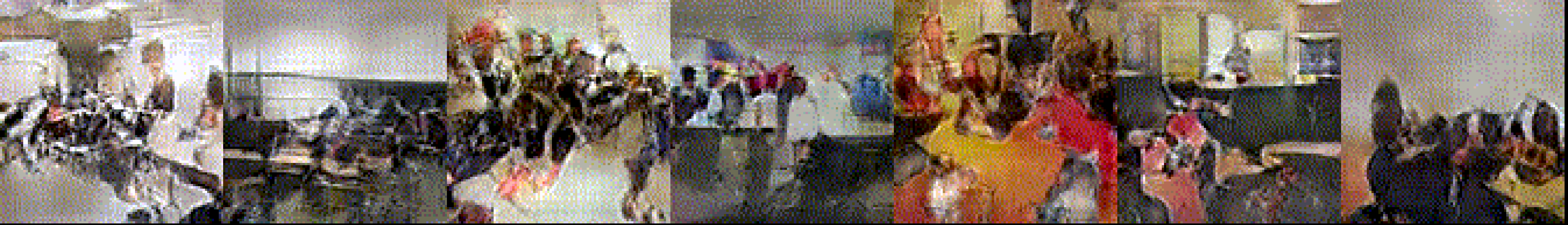


Probabilistic Deep Learning:

Unsupervised Learning and Representation Learning

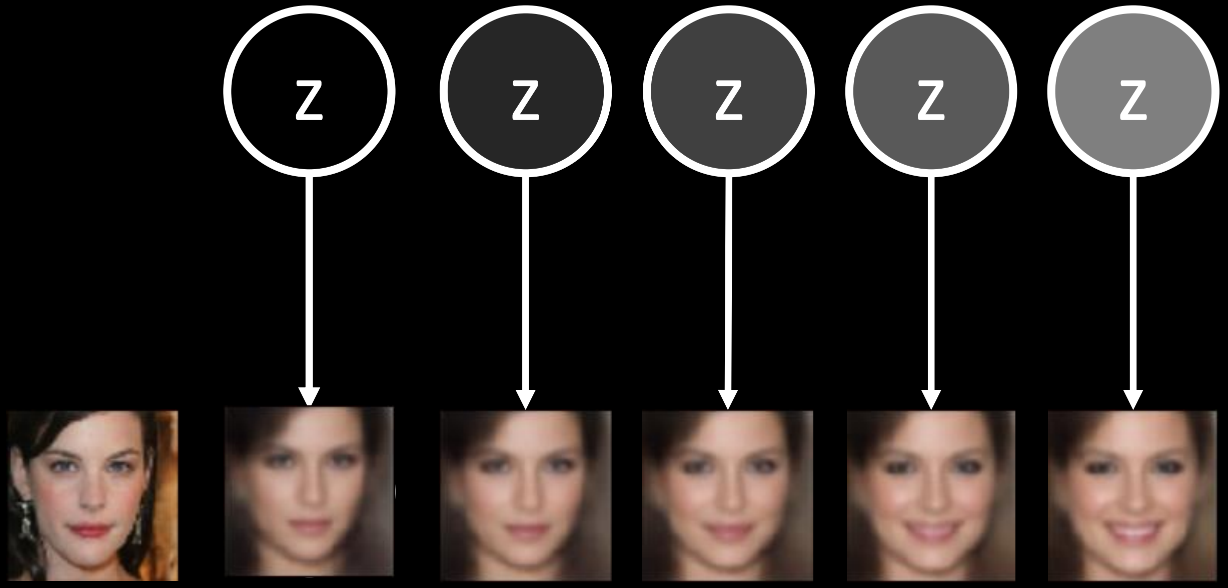
Sebastian Nowozin

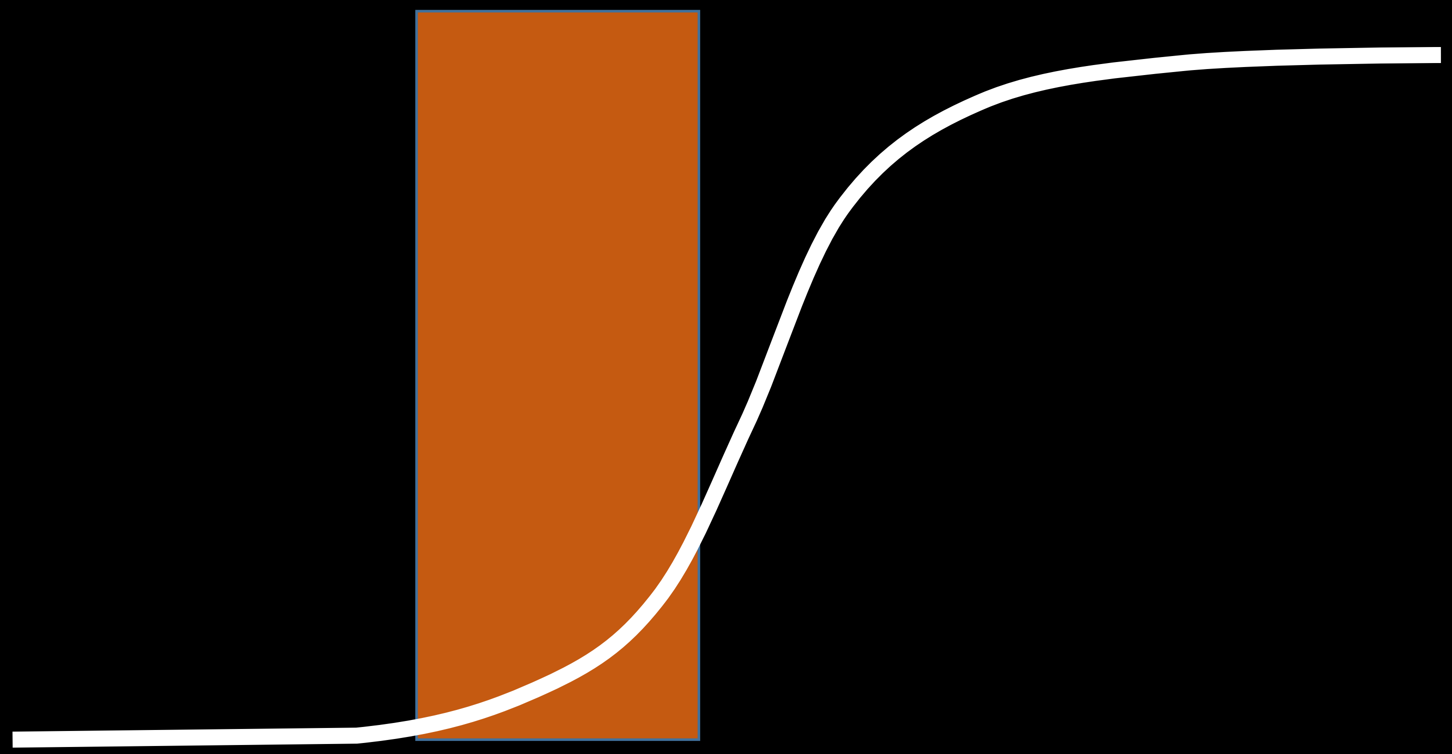
Microsoft Research

Unsupervised learning

Representation learning

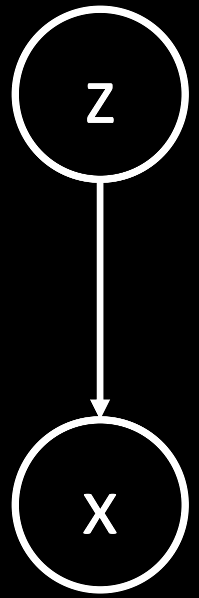
Representation Learning





Missing Pieces

- Controllable representations
- Learning from weak supervision
- Robust learning methods

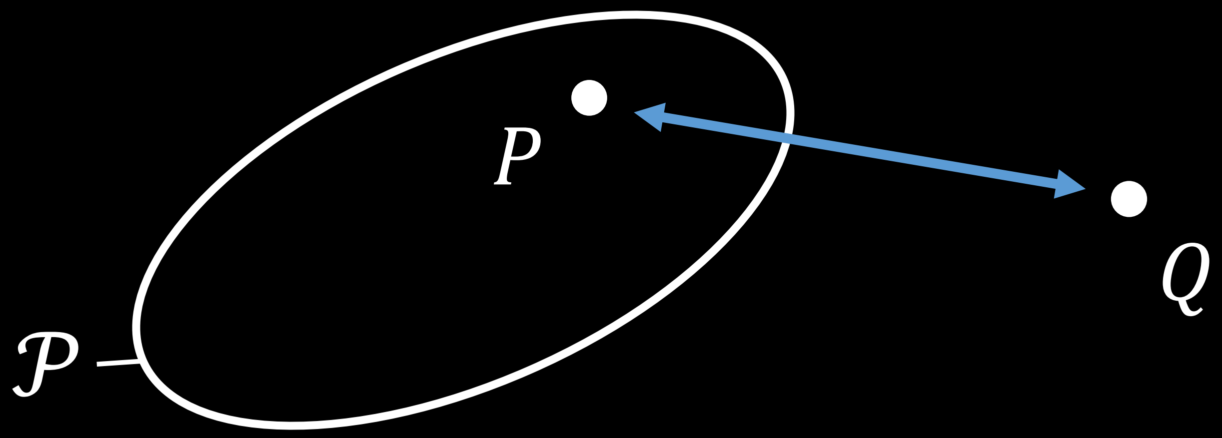


Talk Goals

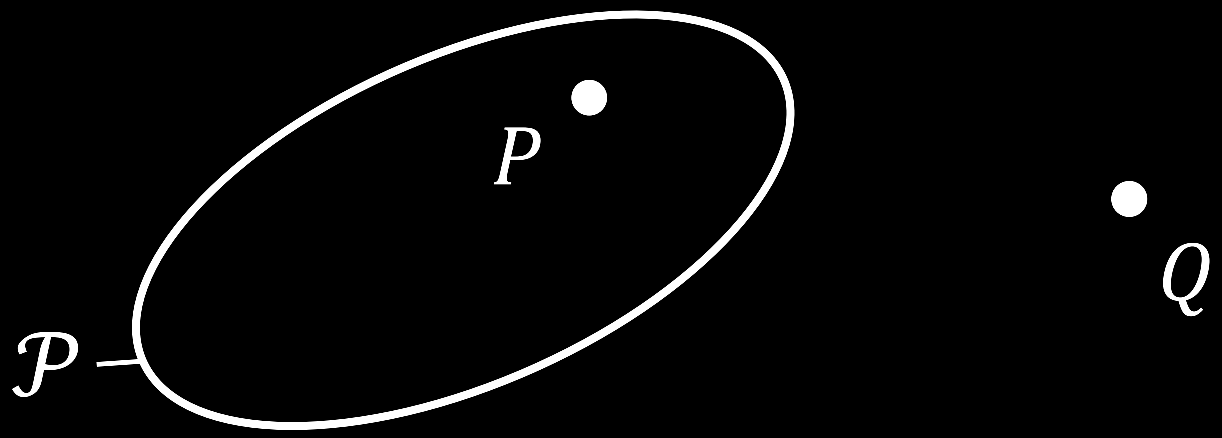
- Give overview of recent advances in unsupervised learning
- Highlight open research challenges

Density Estimation

Learning Probabilistic Models



Learning Probabilistic Models



Assumptions on *P*:

- tractable sampling
- tractable parameter gradient with respect to sample
- tractable likelihood function

Principles of Density Estimation

Integral Probability Metrics
[Müller, 1997]
[Sriperumbudur et al., 2010]

$$\gamma_{\mathcal{F}}(P,Q) = \sup_{f\in\mathcal{F}} \left| \int f \mathrm{d}P - \int f \mathrm{d}Q \right|$$

- Kernel MMD / DISCO
- Wasserstein GANs

Proper scoring rules
[Gneiting and Raftery, 2007]

$$S(P,Q) = \int S(P,x)\mathrm{d}Q(x)$$

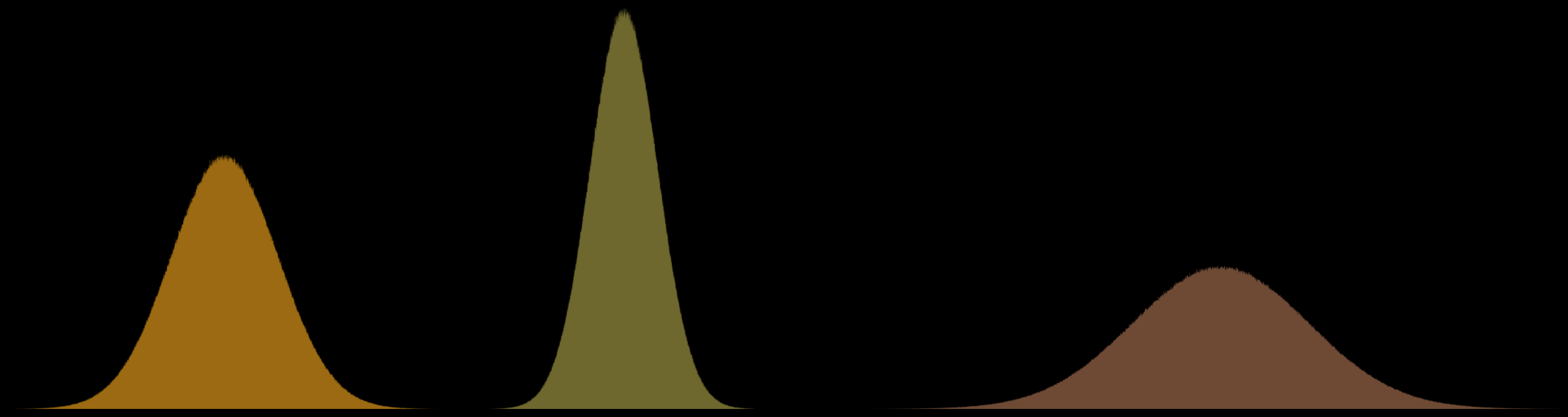
- Variational Autoencoders
- DISCO networks

f-divergences
[Ali and Silvey, 1966],
[Nguyen et al., 2010]

$$D_f(P \parallel Q) = \int q(x) f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x$$

- Generative adversarial networks
- f-GAN, b-GAN

Classic parametric models

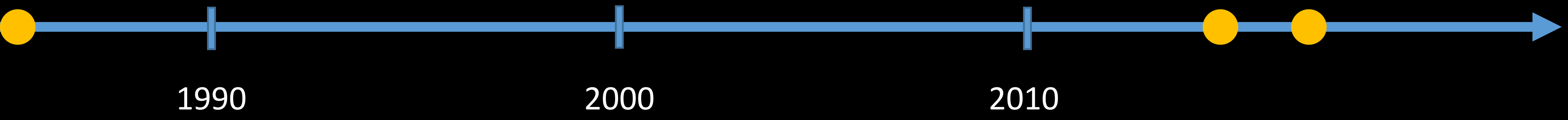


- Density function available
- Limited expressive power
- Mature field in statistics and learning theory

Implicit Model / Neural Sampler / Likelihood-free Model

- Highly expressive model class
- Density function not defined or intractable
- Lack of theory and learning algorithms
- Basis for generative adversarial networks (GANs)

Implicit Models

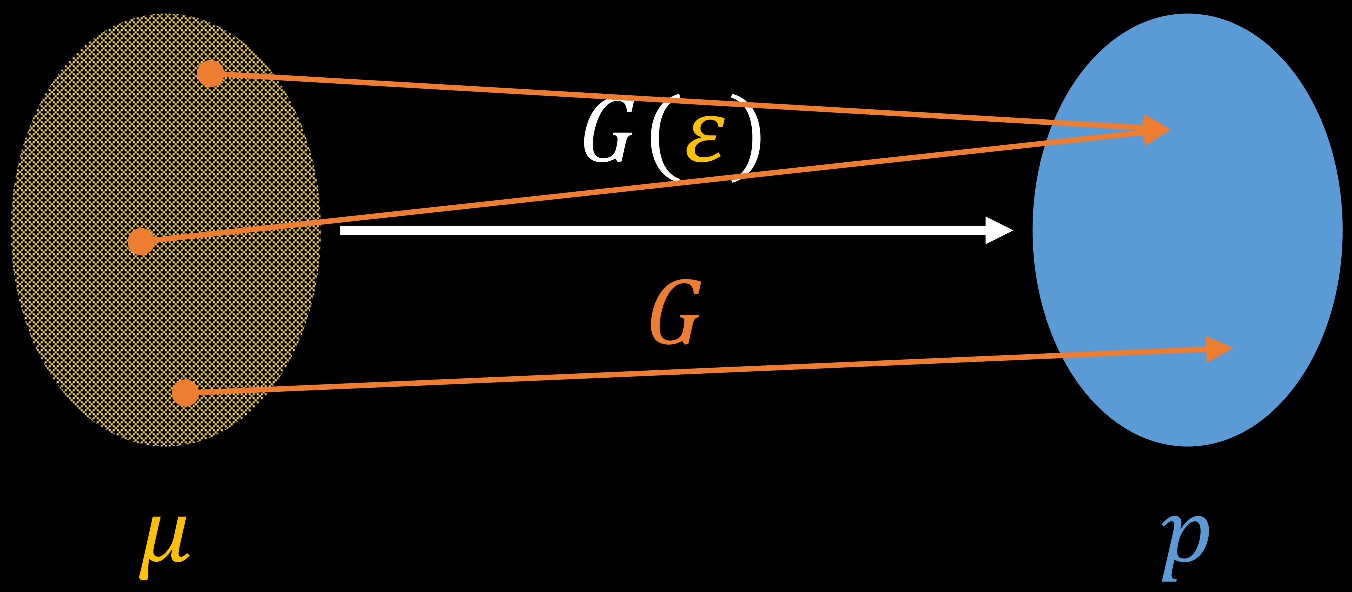


1. Diggle and Gratton (1984). Monte Carlo methods of inference for implicit statistical models. *JRSS B*
2. Goodfellow et al. (2014). Generative Adversarial Nets. NIPS
3. Mohamed and Lakshminarayanan (2016). Learning in Implicit Generative Models. *arXiv:1610.03483*

Implicit models as building blocks

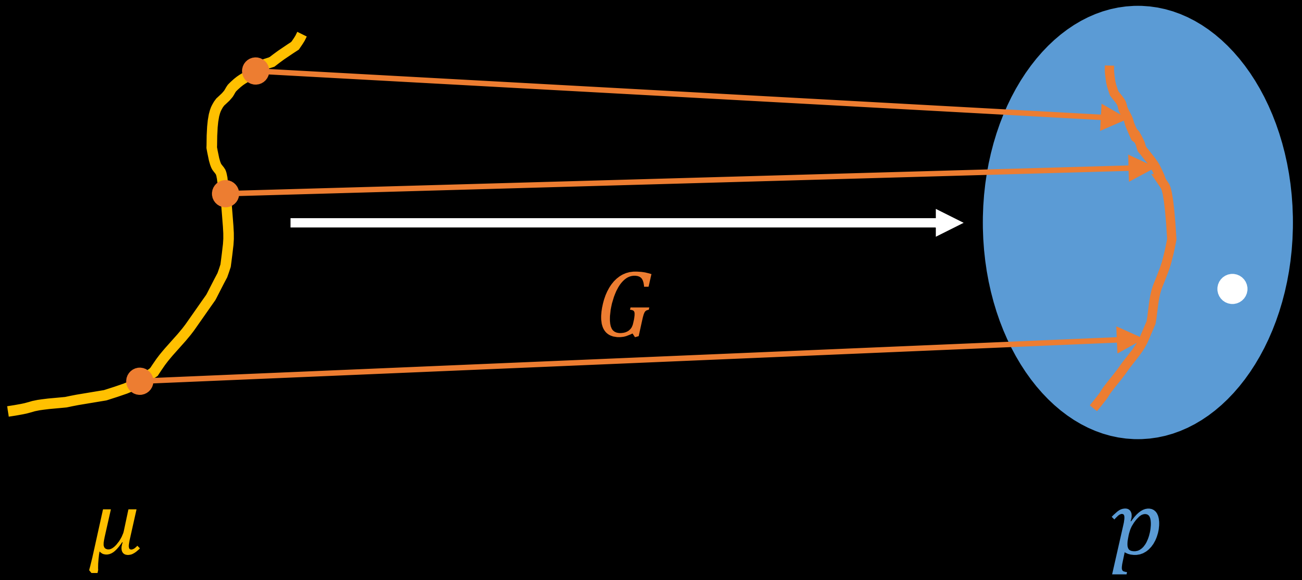
- For inference (as in AVB), or
- As model component, or
- As regularizer

Implicit Models: Problem 1



$$p(x) = \int_{z \in G^{-1}(x)} \mu(z)\, \mathrm{dz}$$

Implicit Models: Problem 2

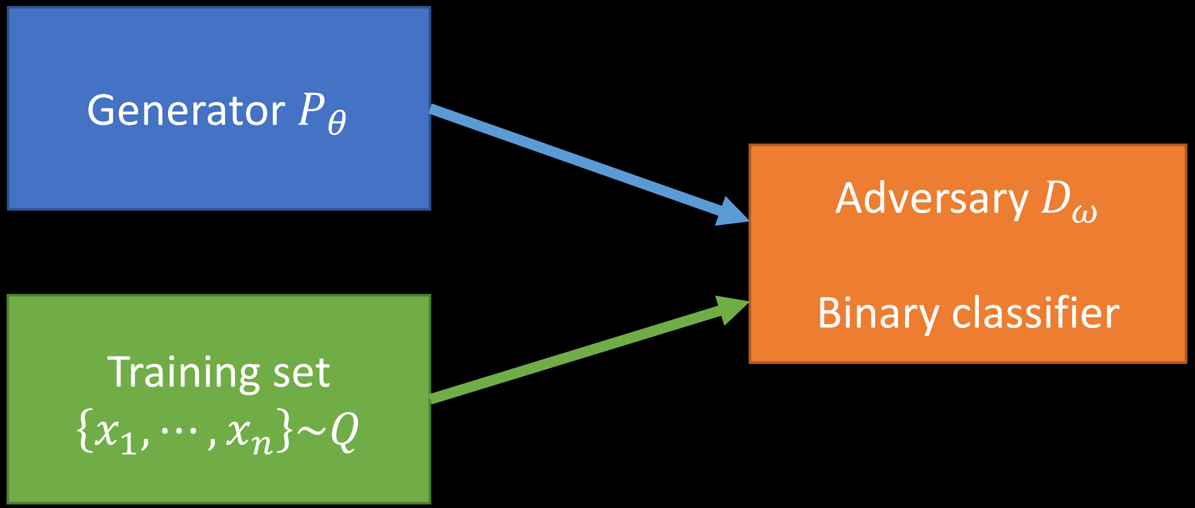


$p(x)$ not defined a.e.

Generative Adversarial Networks

GAN = Implicit Models +
Estimation procedure

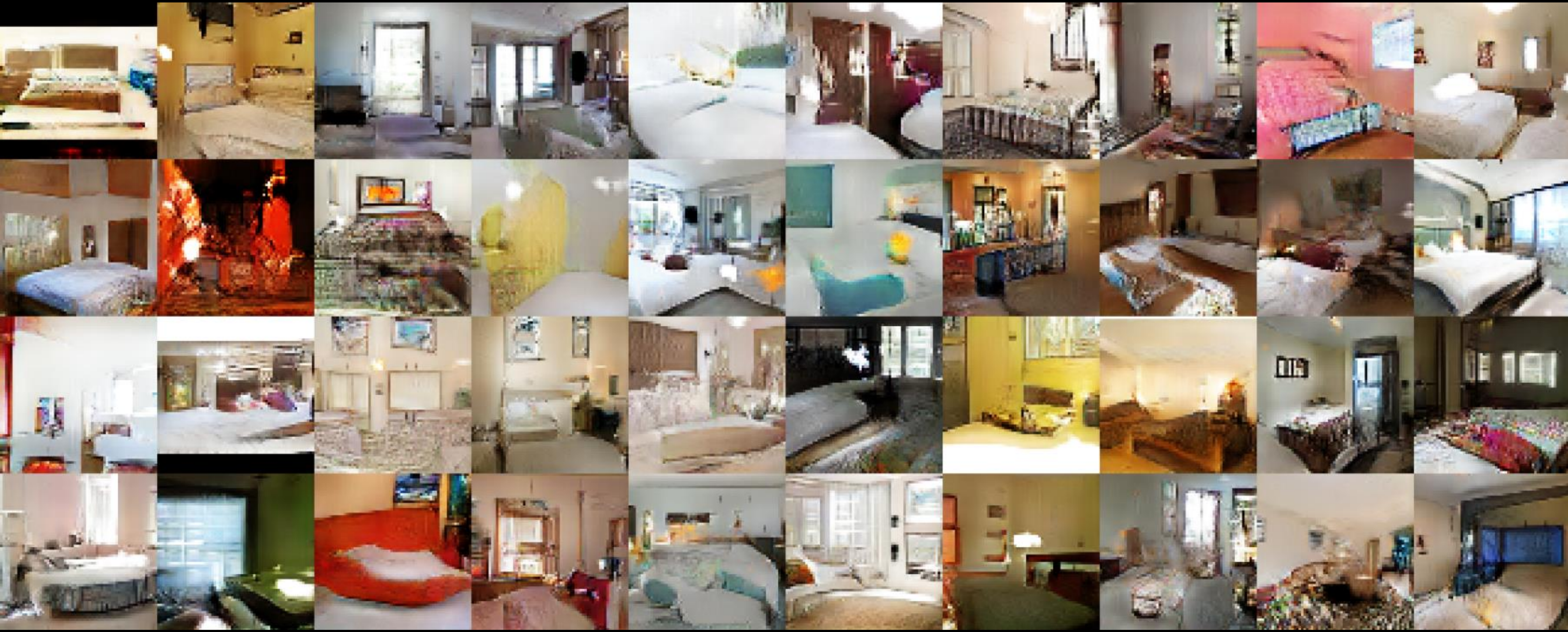
GAN Training Objective [Goodfellow et al., 2014]



- Generator tries to fool discriminator (i.e. generate realistic samples)
- Discriminator tries to distinguish fake from real samples
- Saddle-point problem

$$\min_\theta \max_\omega \mathbb{E}_{x \sim P_\theta}[\log D_\omega(x)] + \mathbb{E}_{x \sim Q}[\log(1 - D_\omega(x))]$$

Natural Images (Radford et al., 2015, arXiv:1511.06434)



Linear interpolation in latent space [Radford et al., 2015]

Estimating f-divergences from samples

[Nguyen, Wainwright, Jordan, 2010]

- Divergence between two distributions

$$D_f(P \parallel Q) = \int_{\mathcal{X}} q(x)\, f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x$$

f : generator function (convex & $f(1)=0$)

- Every convex function f has a *Fenchel conjugate* f^* so that

$$f(u) = \sup_{t \in \mathrm{dom}_{f^*}} \{tu - f^*(t)\}$$

"any convex f can be represented as point-wise max of *linear* functions"

Estimating f-divergences from samples (cont)

[Nguyen, Wainwright, Jordan, 2010]

$$D_f(P \parallel Q) = \int_{\mathcal{X}} q(x)\, f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x$$

$$= \int_{\mathcal{X}} q(x) \sup_{t \in \mathrm{dom}_{f^*}} \left\{ t \frac{p(x)}{q(x)} - f^*(t) \right\} \mathrm{d}x$$

$$\geq \sup_{T \in \mathcal{T}} \left(\int_{\mathcal{X}} p(x)\, T(x)\, \mathrm{d}x - \int_{\mathcal{X}} q(x)\, f^*\big(T(x)\big)\, \mathrm{d}x \right)$$

$$= \sup_{T \in \mathcal{T}} \Big(\underbrace{\mathbb{E}_{x \sim P}[T(x)]} - \underbrace{\mathbb{E}_{x \sim Q}[f^*(T(x))]} \Big)$$

Approximate using: samples from P samples from Q

f-GAN and GAN objectives

- GAN

$$\min_{\theta}\max_{\omega} \mathbb{E}_{x\sim P_\theta}[\log D_\omega(x)] + \mathbb{E}_{x\sim Q}[\log(1 - D_\omega(x))]$$

- f-GAN

$$\min_{\theta}\max_{\omega}\left(\mathbb{E}_{x\sim P_\theta}[T_\omega(x)] - \mathbb{E}_{x\sim Q}[f^*(T_\omega(x))]\right)$$

  - GAN discriminator-variational function correspondence: $\log D_\omega(x) = T_\omega(x)$
  - GAN minimizes the Jensen-Shannon divergence (which was also pointed out in Goodfellow et al., 2014)

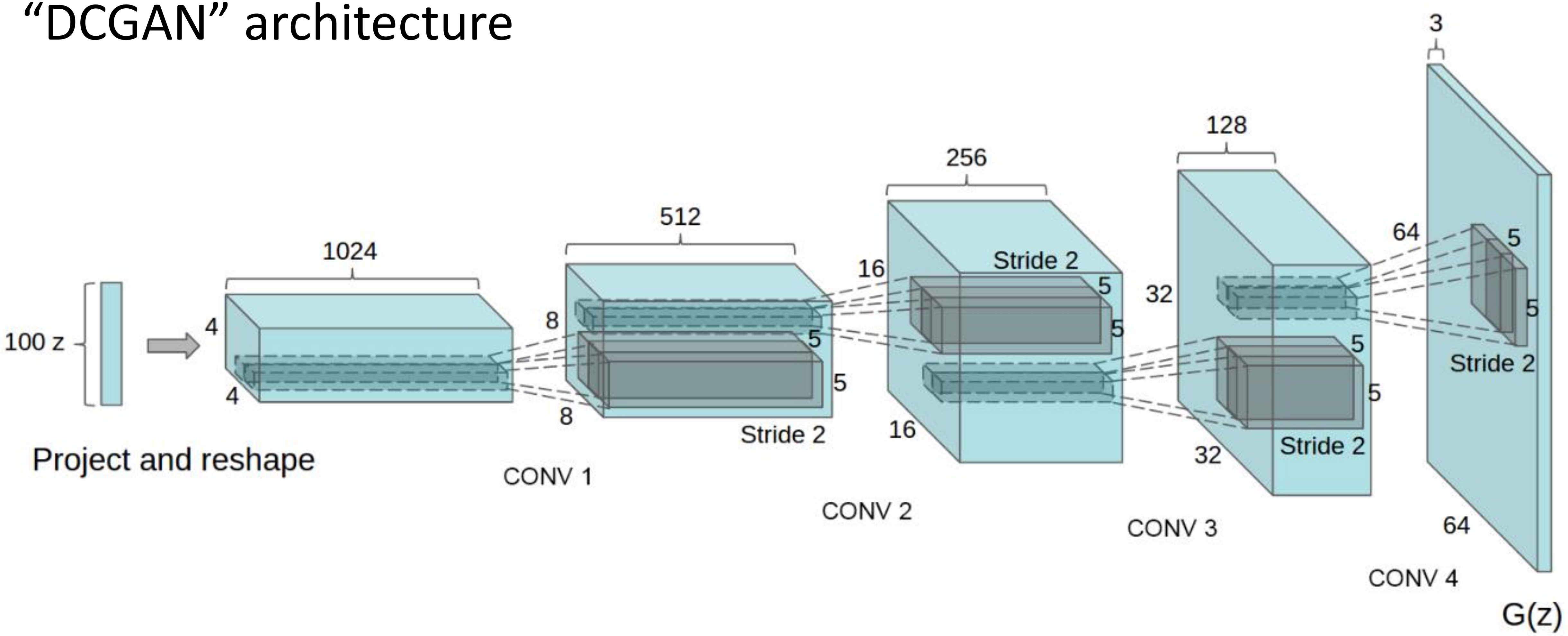
f-divergences

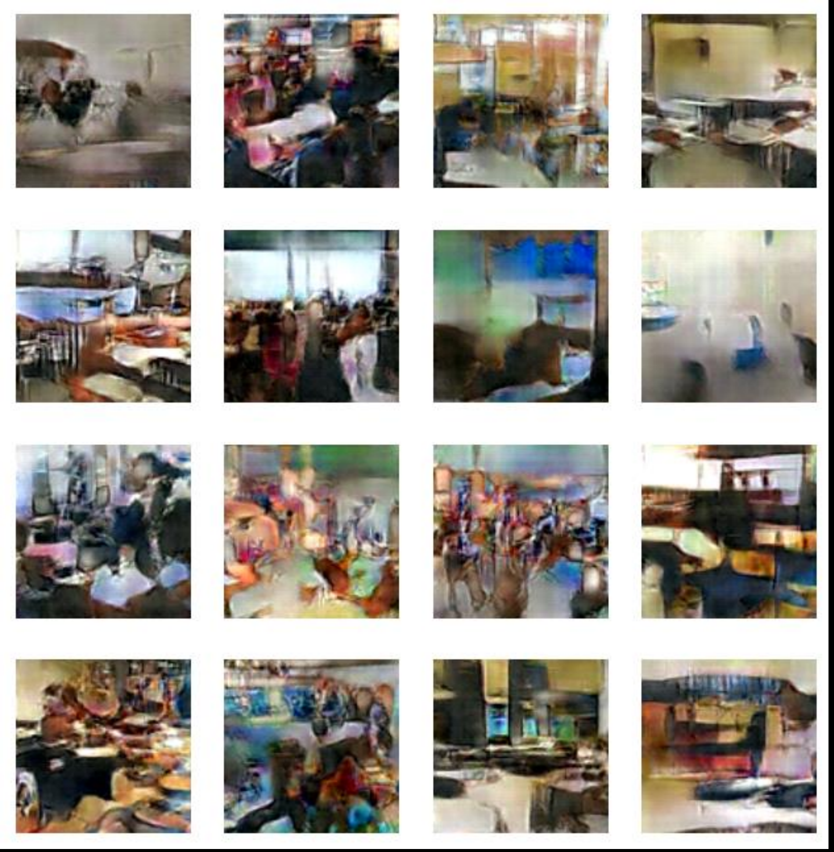
Name	$D_f(P\|Q)$	Generator $f(u)$
Total variation	$\frac{1}{2}\int \lvert p(x)-q(x)\rvert \,\mathrm{d}x$	$\frac{1}{2}\lvert u-1\rvert$
Kullback-Leibler	$\int p(x)\log\frac{p(x)}{q(x)}\,\mathrm{d}x$	$u\log u$
Reverse Kullback-Leibler	$\int q(x)\log\frac{q(x)}{p(x)}\,\mathrm{d}x$	$-\log u$
Pearson χ^2	$\int \frac{(q(x)-p(x))^2}{p(x)}\,\mathrm{d}x$	$(u-1)^2$
Neyman χ^2	$\int \frac{(p(x)-q(x))^2}{q(x)}\,\mathrm{d}x$	$\frac{(1-u)^2}{u}$
Squared Hellinger	$\int \left(\sqrt{p(x)}-\sqrt{q(x)}\right)^2\,\mathrm{d}x$	$(\sqrt{u}-1)^2$
Jeffrey	$\int (p(x)-q(x))\log\left(\frac{p(x)}{q(x)}\right)\,\mathrm{d}x$	$(u-1)\log u$
Jensen-Shannon	$\frac{1}{2}\int p(x)\log\frac{2p(x)}{p(x)+q(x)}+q(x)\log\frac{2q(x)}{p(x)+q(x)}\,\mathrm{d}x$	$-(u+1)\log\frac{1+u}{2}+u\log u$
Jensen-Shannon-weighted	$\int p(x)\pi\log\frac{p(x)}{\pi p(x)+(1-\pi)q(x)}+(1-\pi)q(x)\log\frac{q(x)}{\pi p(x)+(1-\pi)q(x)}\,\mathrm{d}x$	$\pi u\log u-(1-\pi+\pi u)\log(1-\pi+\pi u)$
GAN	$\int p(x)\log\frac{2p(x)}{p(x)+q(x)}+q(x)\log\frac{2q(x)}{p(x)+q(x)}\,\mathrm{d}x-\log(4)$	$u\log u-(u+1)\log(u+1)$
α-divergence ($\alpha\notin\{0,1\}$)	$\frac{1}{\alpha(\alpha-1)}\int\left(p(x)\left[\left(\frac{q(x)}{p(x)}\right)^{\alpha}-1\right]-\alpha(q(x)-p(x))\right)\,\mathrm{d}x$	$\frac{1}{\alpha(\alpha-1)}\left(u^{\alpha}-1-\alpha(u-1)\right)$

LSUN Natural Images

- [Yu et al., 2015] one of the largest databases of natural images
- 168k images of classrooms
- [Radford et al., 2015] architecture
  - Generator: deconvolutional network, ~3M parameters
  - Variational function: convnet, ~3M parameters
- Batch normalization, gradient clipping, Adam
- ~3 hours training time (Titan X), ~135 images/s

“DCGAN” architecture



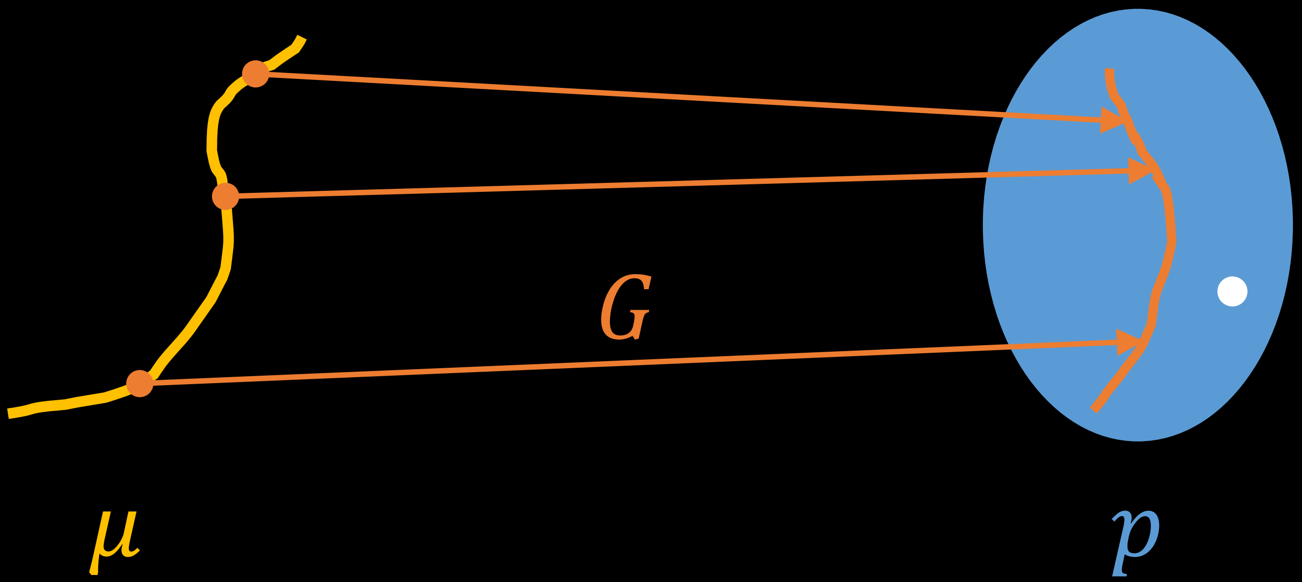


GAN (Jensen-Shannon)

Hellinger

Kullback-Leibler

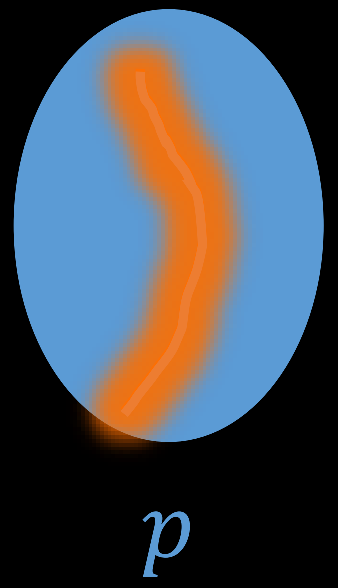
Implicit Models



$p(x)$ not defined a.e.

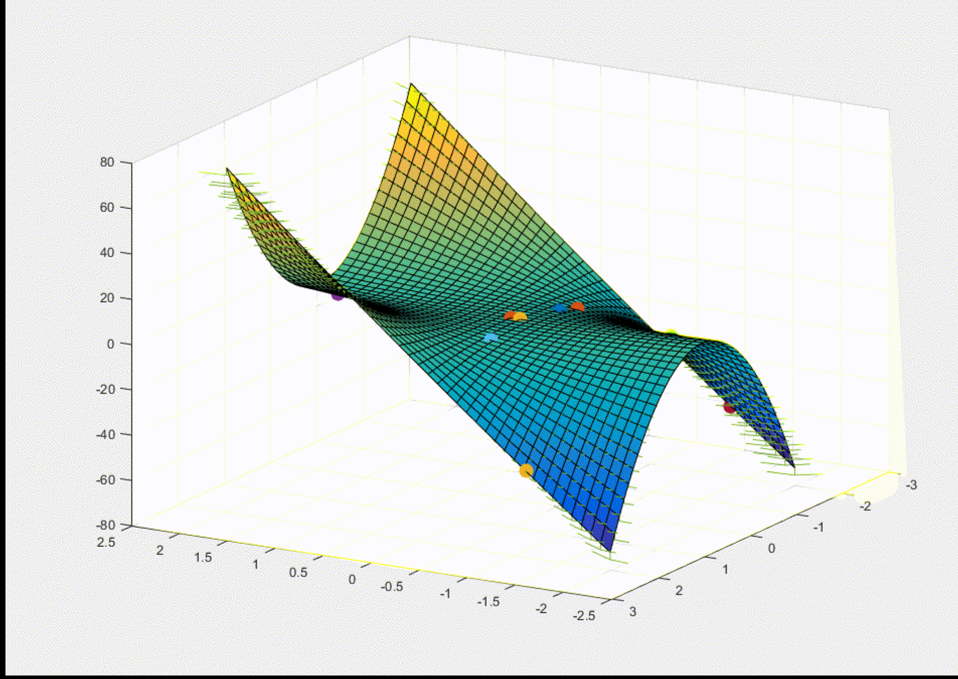
Implicit Models

- Use generalized f-divergence
$$D_{f,K}(P,Q) = D_f(K * P, K * Q)$$
- Implementation: add noise
  [Sønderby et al., 2016]
  [Arjovsky and Bottou, 2016]
- Implementation: analytic
  [Roth et al., 2017]
- Choice of kernel introduces local geometry



$p(x)$ not defined a.e.

Stability of GAN Training



- [Mescheder et al., "Numerics of GANs", arXiv:1705.10461, NIPS 2017]
- [Roth et al., "Stabilizing Training of Generative Adversarial Networks through Regularization", arXiv:1705.09367, NIPS 2017]

Principles of Density Estimation

Integral Probability Metrics
[Müller, 1997]
[Sriperumbudur et al., 2010]

$$\gamma_{\mathcal{F}}(P, Q) = \sup_{f \in \mathcal{F}} \left| \int f \mathrm{d}P - \int f \mathrm{d}Q \right|$$

Proper scoring rules
[Gneiting and Raftery, 2007]

$$S(P, Q) = \int S(P, x) \mathrm{d}Q(x)$$

f-divergences
[Ali and Silvey, 1966],
[Nguyen et al., 2010]

$$D_f(P \parallel Q) = \int q(x) f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x$$

- Kernel MMD / DISCO
- Wasserstein GANs

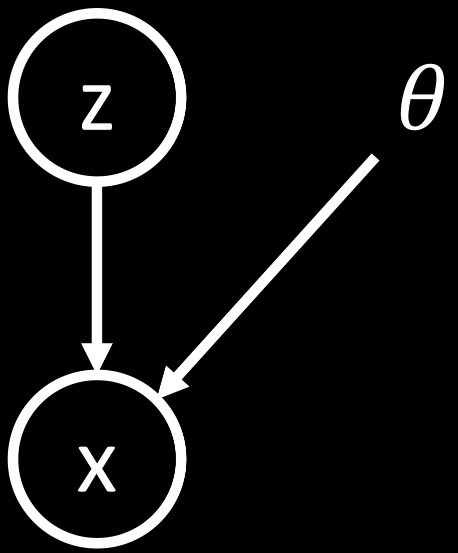
- Variational Autoencoders
- DISCO networks

- Generative adversarial networks
- f-GAN, b-GAN

Variational Autoencoders (VAE)

[Kingma and Welling, 2014], [Rezende et al., 2014]

VAE: Model



$$p(x|\theta) = \int \mathrm{p(x|z,\theta)}p(z)\mathrm{d}z$$

- $p(z)$ is a multivariate standard Normal
- $p(x|z,\theta)$ is a neural network outputting a simple distribution (e.g. diagonal Normal)

VAE: Maximum Likelihood Training

- Maximize the data log-likelihood, per-instance variational approximation

$$
\begin{aligned}
\log p(x|\theta) &= \log \int \mathrm{p(x|z,\theta)} p(z) \mathrm{dz} \\
&= \log \int \mathrm{p(x|z,\theta)} \frac{q(z)}{q(z)} p(z)\, \mathrm{dz} \\
&= \log \int \mathrm{p(x|z,\theta)} \frac{p(z)}{q(z)} q(z)\, \mathrm{dz} \\
&= \log \mathbb{E}_{z\sim q(z)} \left[\mathrm{p(x|z,\theta)} \frac{p(z)}{q(z)} \right] \\
&\geq \mathbb{E}_{z\sim q(z)} \left[\log \mathrm{p(x|z,\theta)} \frac{p(z)}{q(z)} \right] \\
&= \mathbb{E}_{z\sim q(z)} [\log \mathrm{p(x|z,\theta)}] - D_{\mathrm{KL}}(q(z) \parallel p(z))
\end{aligned}
$$

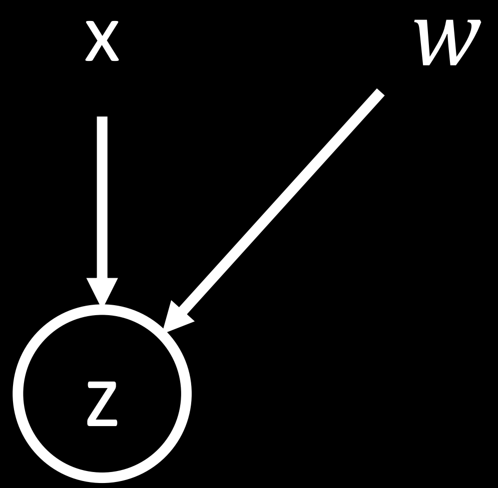
Inference networks

- Amortized inference [Stuhlmüller et al., NIPS 2013]
- Inference networks, recognition networks [Kingma and Welling, 2014]
- “Informed sampler” [Jampani et al., 2014]
- “Memory-based approach” [Kulkarni et al., 2015]

Inference networks

x_1 x_2 x_3

$q_1(z)$ $q_2(z)$ $q_3(z)$



VAE: Maximum Likelihood Training

- Maximize the data log-likelihood, inference network variational approximation

$$\log p(x|\theta) = \log \int \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) p(z) \mathrm{dz}$$

$$= \log \int \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) \frac{q(z|x,w)}{q(z|x,w)} p(z)\, \mathrm{dz}$$

$$= \log \int \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) \frac{p(z)}{q(z|x,w)} q(z|x,w)\, \mathrm{dz}$$

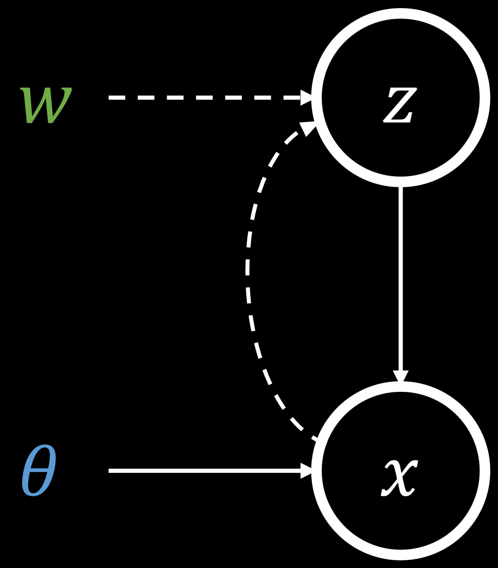
$$= \log \mathbb{E}_{z\sim q(z|x,w)} \left[\mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) \frac{p(z)}{q(z|x,w)} \right]$$

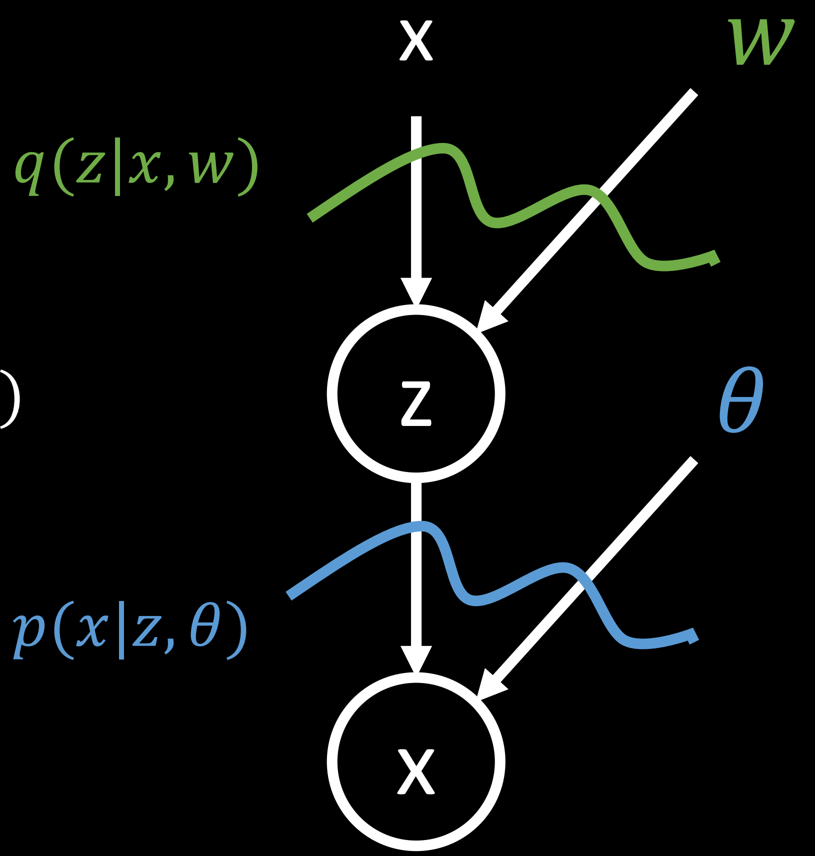
$$\geq \mathbb{E}_{z\sim q(z|x,w)} \left[\log \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) \frac{p(z)}{q(z|x,w)} \right]$$

$$= \mathbb{E}_{z\sim q(z|x,w)} [\log \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta)] - D_{\mathrm{KL}}(q(z|x,w) \parallel p(z))$$

Autoencoder viewpoint

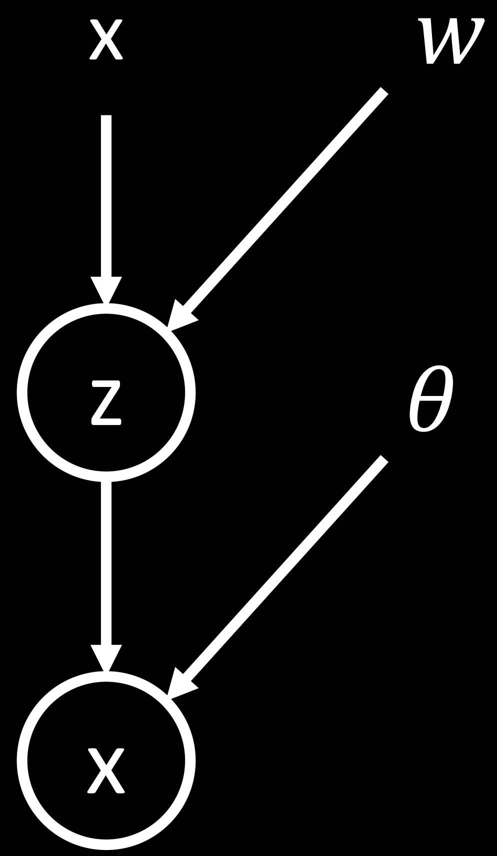
$$\max_{w,\theta} \; \mathbb{E}_{z\sim q(z|x,w)}[\log \mathrm{p}(\mathrm{x}|\mathrm{z},\theta)] - D_{\mathrm{KL}}(q(z|x,w) \parallel p(z))$$





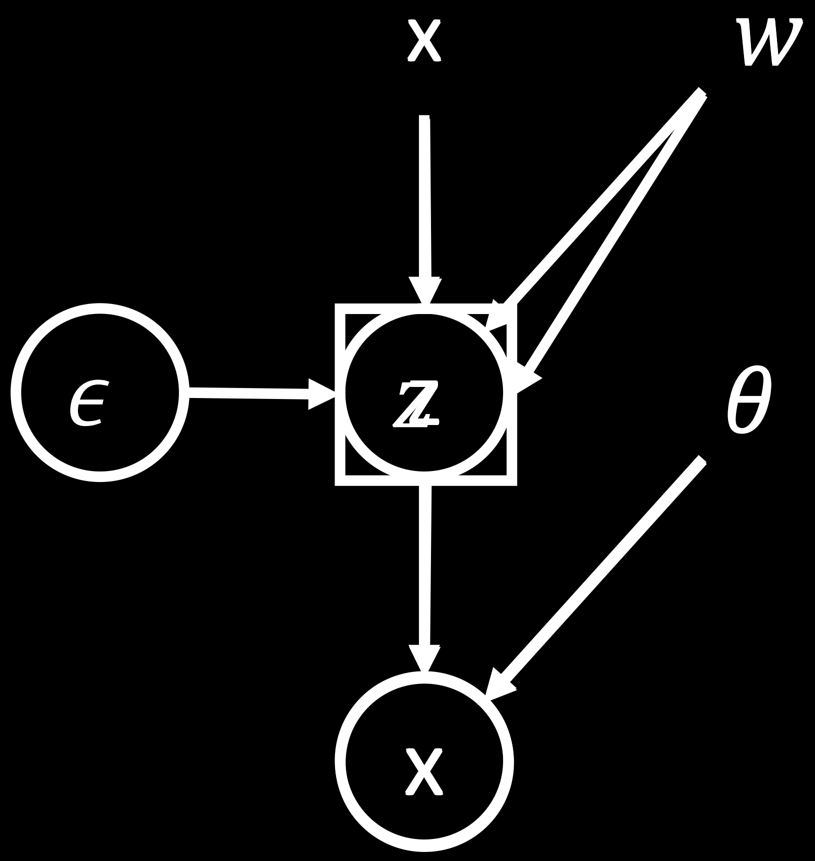
Reparametrization Trick

- [Rezende et al., 2014]
  [Kingma and Welling, 2014]

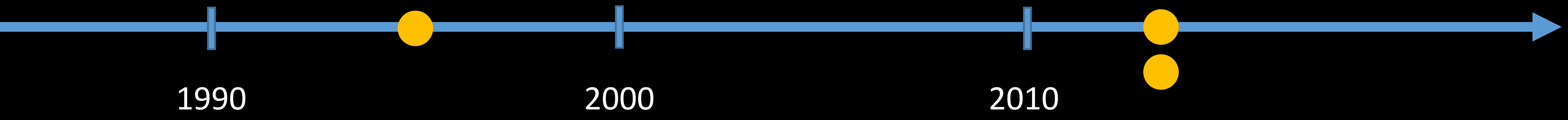


Reparametrization Trick

- [Rezende et al., 2014]
  [Kingma and Welling, 2014]
- Stochastic computation graphs
  [Schulman et al., 2015]

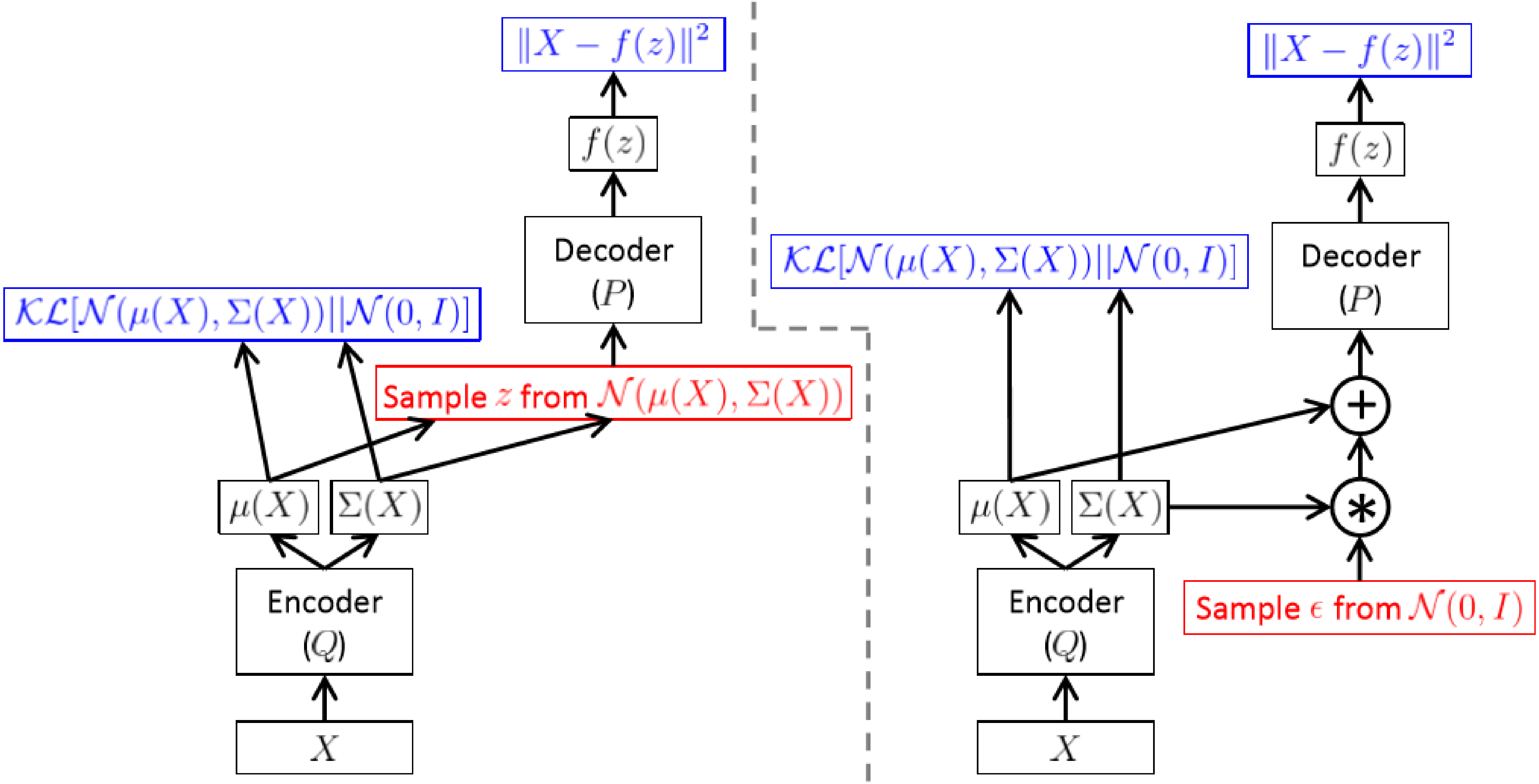


Variational Autoencoders

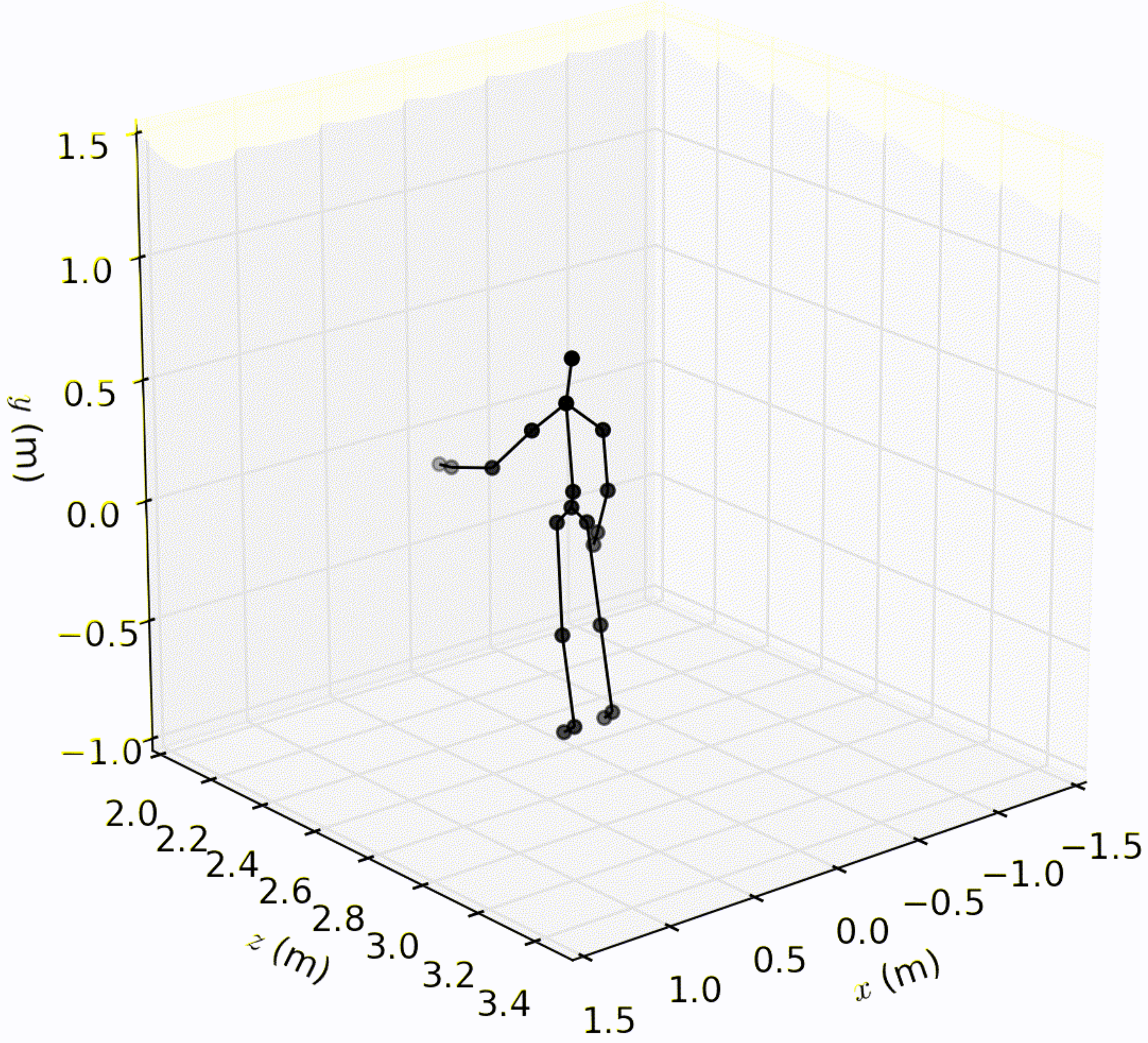


1. Dayan et al. (1995). The Helmholtz machine. *Neural Computation*
2. Kingma and Welling (2014). Auto-encoding Variational Bayes. NIPS
3. Rezende et al. (2014). Stochastic backpropagation and approximate inference in deep generative models. *ICML*

From highly-recommended tutorial: [Doersch, “Tutorial on Variational Autoencoders”, arXiv:1606.05908]



```
def encode(self, x):
    h = F.crelu(self.qlin0(x))
    h = F.crelu(self.qlin1(h))
    h = F.crelu(self.qlin2(h))
    h = F.crelu(self.qlin3(h))

    self.qmu = self.qlin_mu(h)
    self.qln_var = self.qlin_ln_var(h)

def decode(self, z):
    h = F.crelu(self.plin0(z))
    h = F.crelu(self.plin1(h))
    h = F.crelu(self.plin2(h))
    h = F.crelu(self.plin3(h))

    self.pmu = self.plin_mu(h)
    self.pln_var = self.plin_ln_var(h)

def __call__(self, x):
    # Compute q(z|x)
    self.encode(x)

    self.kl = gaussian_kl_divergence(self.qmu, self.qln_var)
    self.logp = 0
    for j in xrange(self.num_zsamples):
        # z ~ q(z|x)
        z = F.gaussian(self.qmu, self.qln_var)

        # Compute p(x|z)
        self.decode(z)

        # Compute objective
        self.logp += gaussian_logp(x, self.pmu, self.pln_var)

    self.logp /= self.num_zsamples
    self.obj = self.kl - self.logp

    return self.obj
```

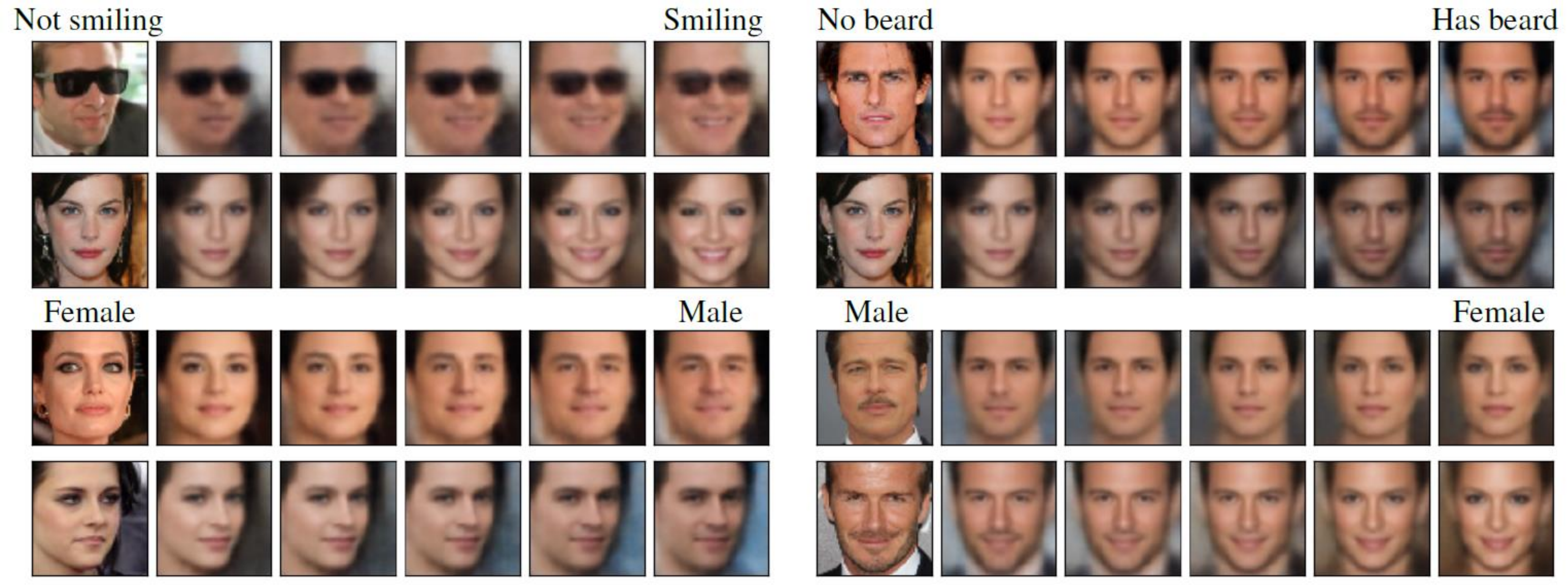


Problems in VAEs (as of 2017)

- Inadequate inference networks
  - Loose ELBO
  - Limits what the generative model can learn
- Parametric conditional likelihood assumptions
  - Limits the expressivity of the generative model
  - “Noise term has to explain too much”
- No control over latent representation that is learned

“Blurry images” in VAE models

from [Tulyakov, Fitzgibbon, Nowozin, ICCV 2017]



Improving Inference Networks

- State of the art in inference network design:
  - NICE [Dinh et al., 2015]
  - Hamiltonian variational inference (HVI) [Salimans et al., 2015]
  - Importance weighted autoencoder (IWAE) [Burda et al., 2016]
  - Normalizing flows [Rezende and Mohamed, 2016]
  - Auxiliary deep generative networks [Maaløe et al., 2016]
  - Inverse autoregressive flow (IAF) [Kingma et al., NIPS 2016]
  - Householder flows [Tomczak and Welling, 2017]
  - Adversarial variational Bayes (AVB) [Mescheder et al., 2017]
  - Deep and Hierarchical Implicit Models [Tran et al., 2017]
  - Variational Inference using Implicit Distributions [Huszár, 2017]
  - Adversarial Message Passing for Graphical Models [Karaletsos, 2016]

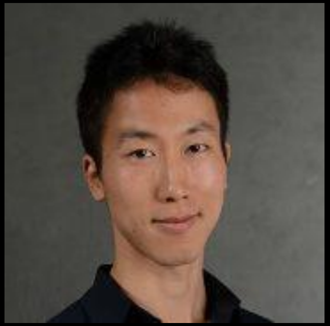
Problems in VAEs (as of 2017)

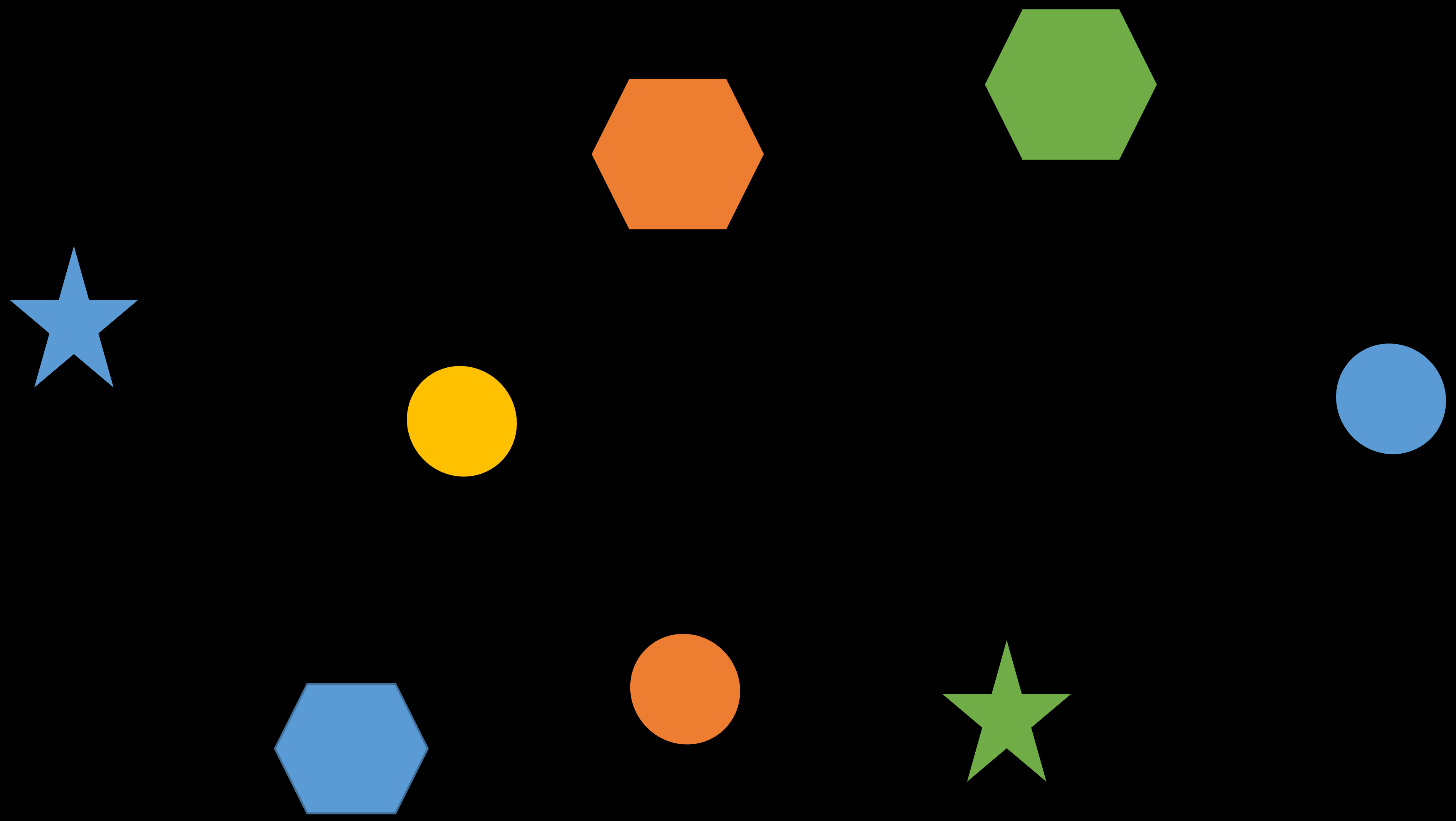
- Inadequate inference networks
  - Loose ELBO
  - Limits what the generative model can learn
- Parametric conditional likelihood assumptions
  - Limits the expressivity of the generative model
  - "Noise term has to explain too much"
- No control over latent representation that is learned

VAEs for Representation Learning

Diane Bouchacourt, Ryota Tomioka, Sebastian Nowozin

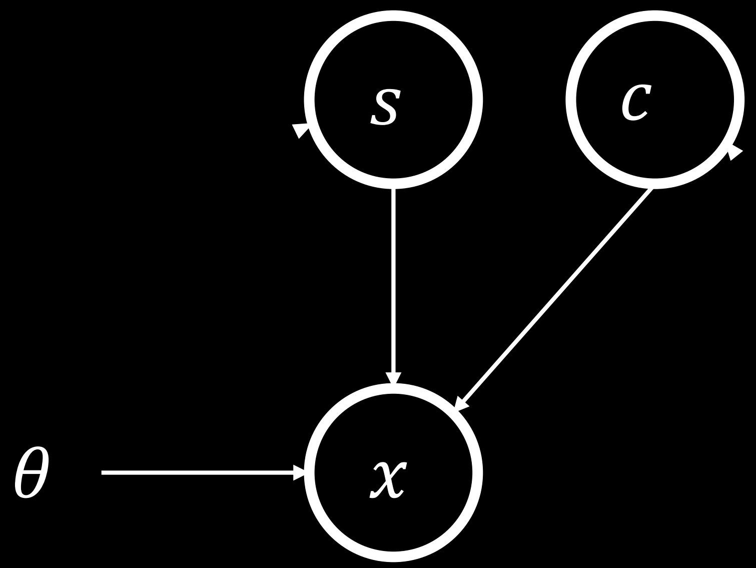
arXiv:1705.08841, NIPS 2017





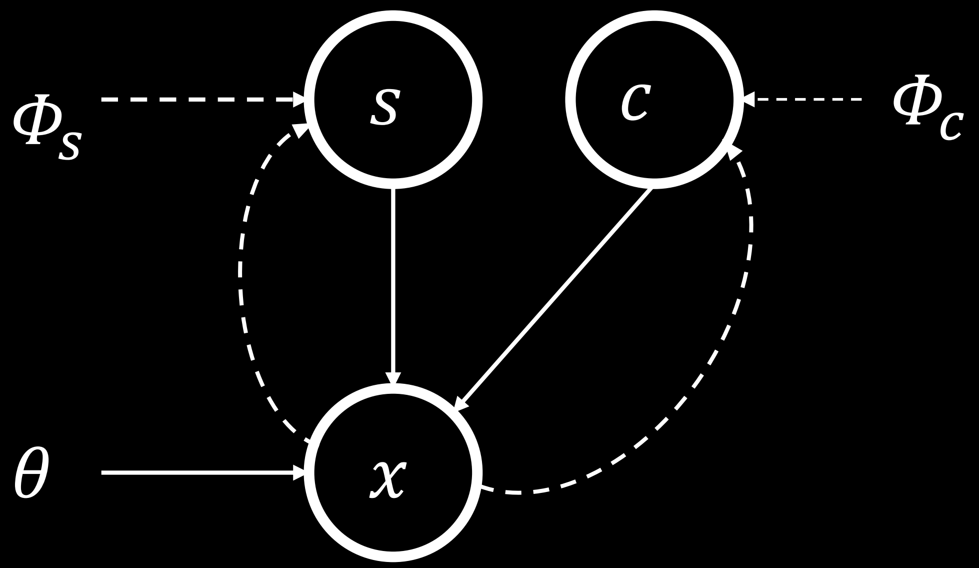
Two parts latent code

- Style s
- Content c



Two parts latent code

- Style s
- Content c



Related work

- Unsupervised
  [Chen et al., 2016; Wang and Gupta, 2016; Higgins et al., 2017]

    does not anchor specific meaning

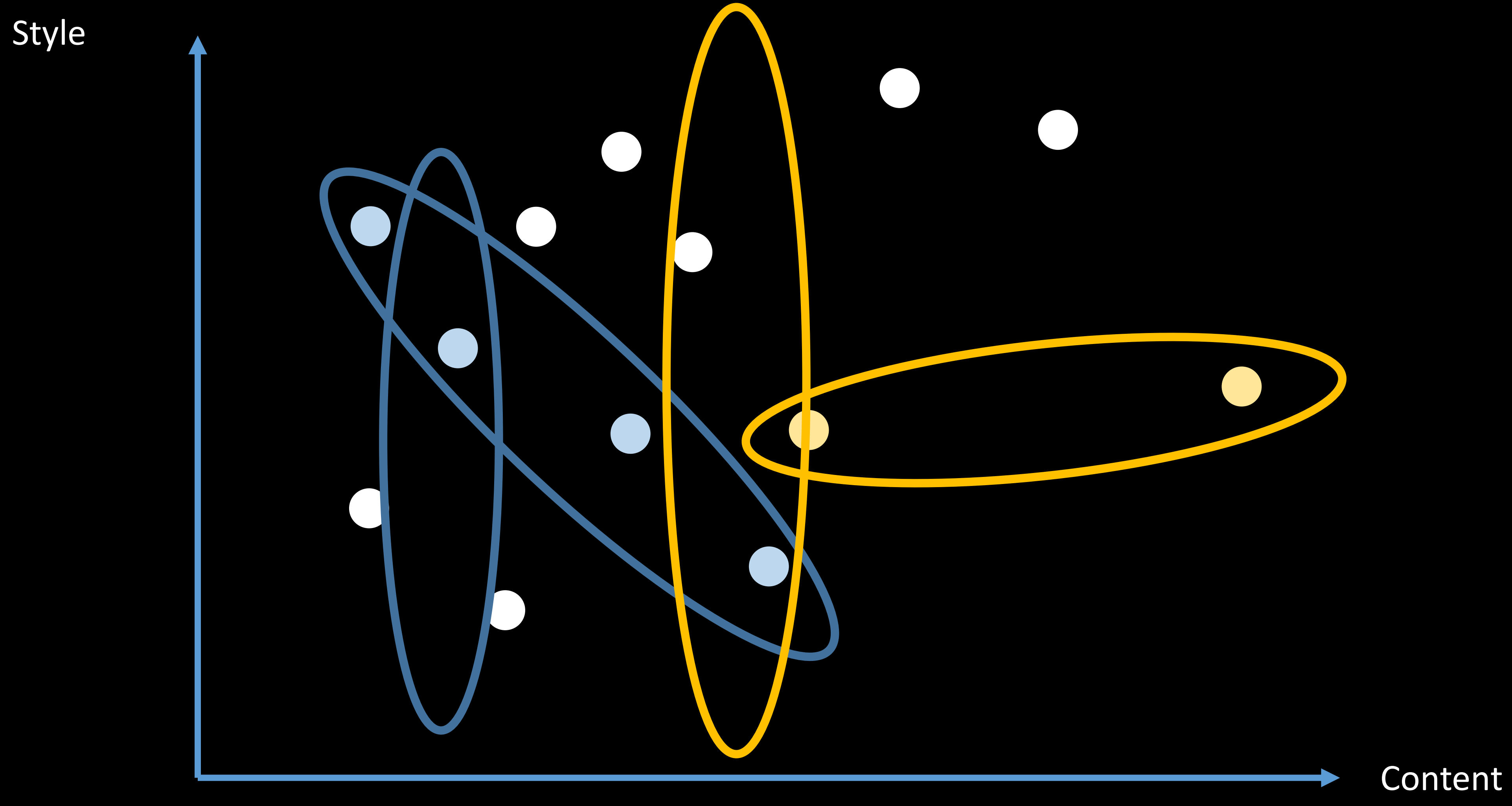
- Semi-supervised
  [Siddarth et al., 2017; Louizos et al., 2016; Chen et al. 2017]

    requires supervision

- Group supervision [Bouchacourt et al., arXiv:1705.08841]

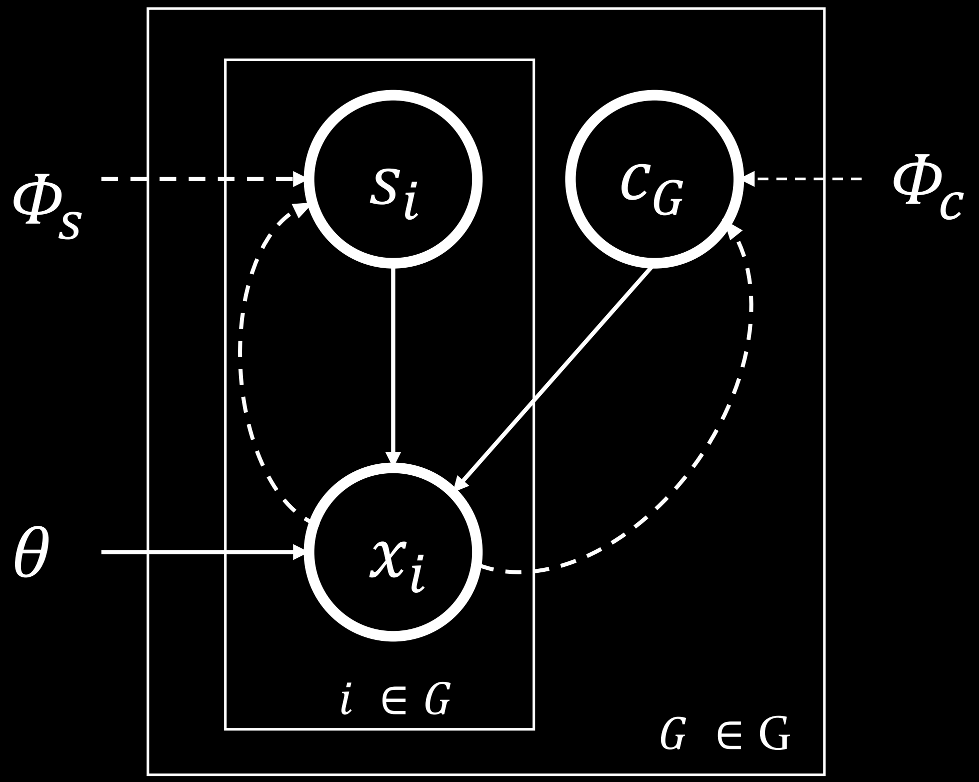
    inexpensive weak supervision

Group-level Supervision



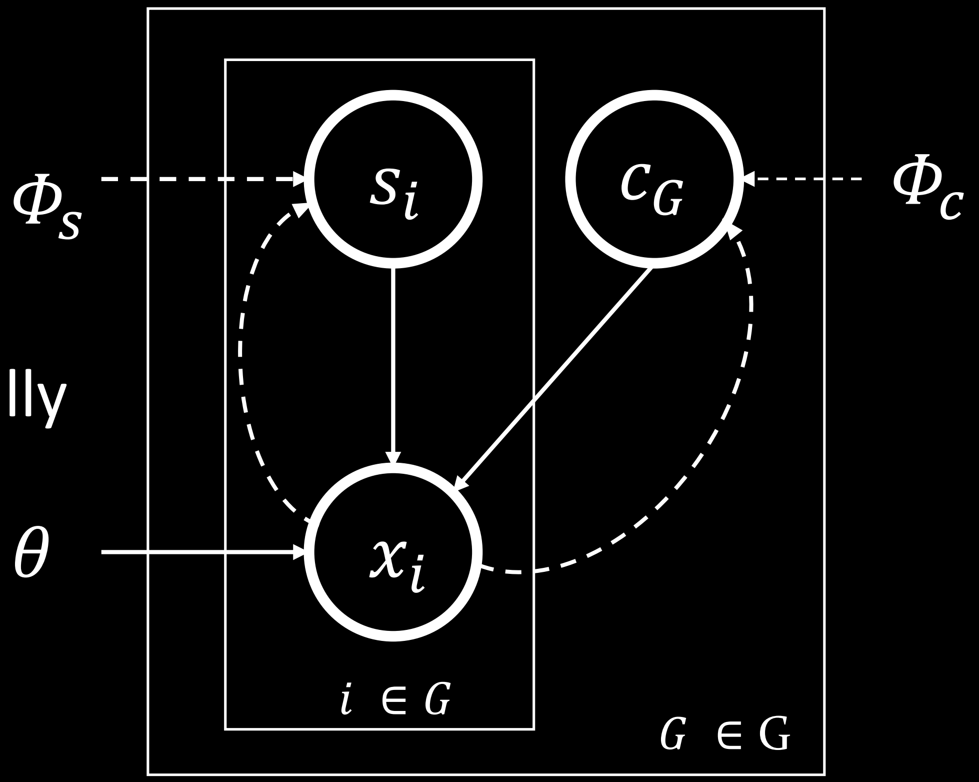
Two parts latent code

- Style s_i
- Content c_G



Multi-Level VAE

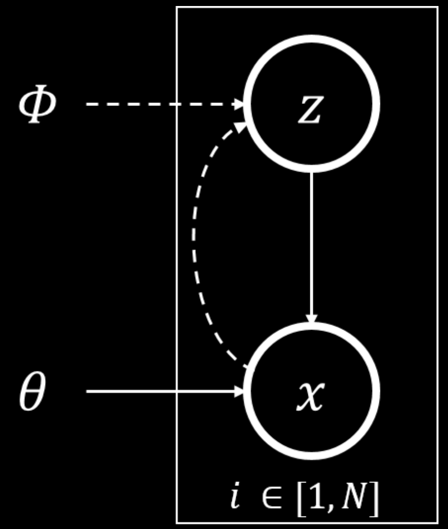
- Style s_i
- Content c_G
- Independent, identically distributed **groups**



VAE

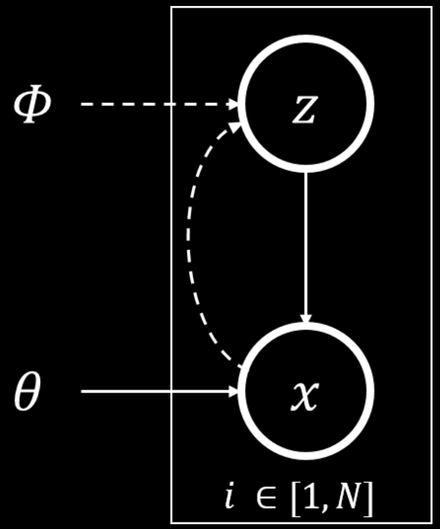
Independent, identically distributed samples

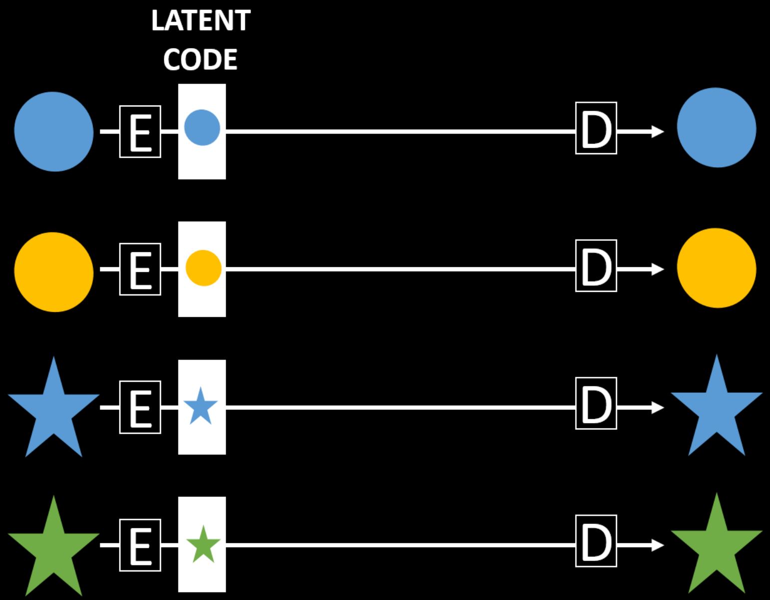
Amortised inference



VAE

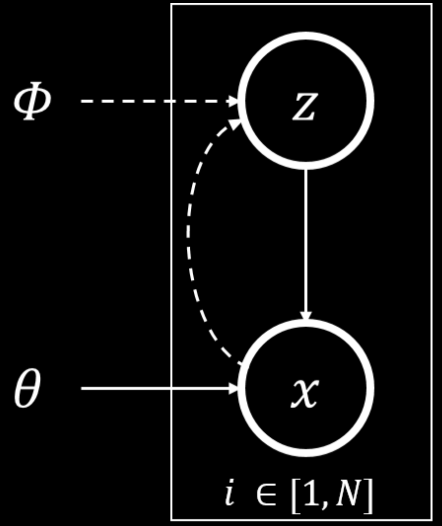
Independent, identically distributed samples
Amortised inference

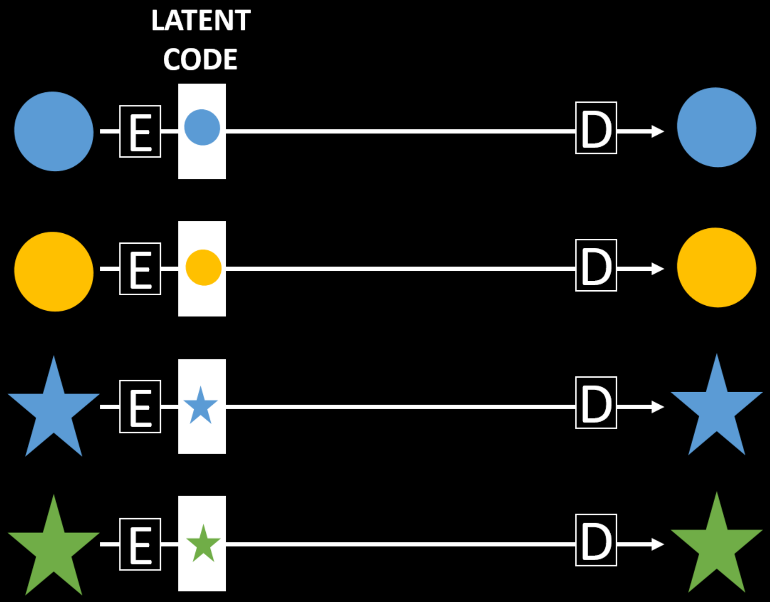




VAE

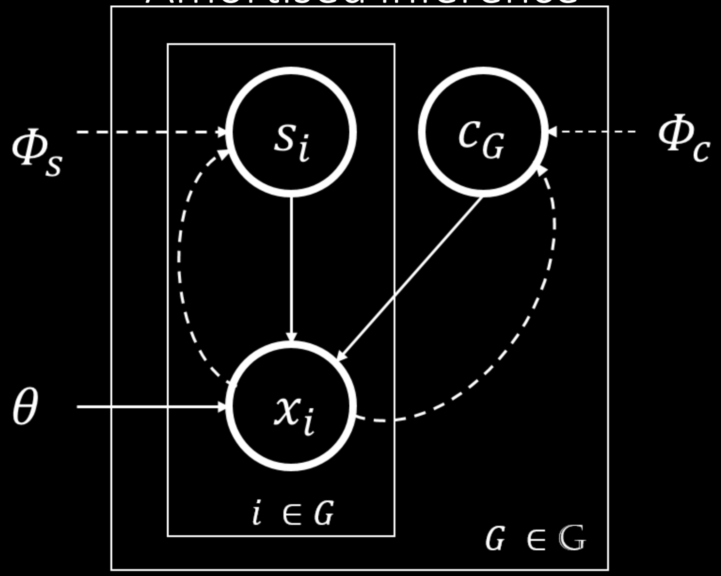
Independent, identically distributed samples
Amortised inference





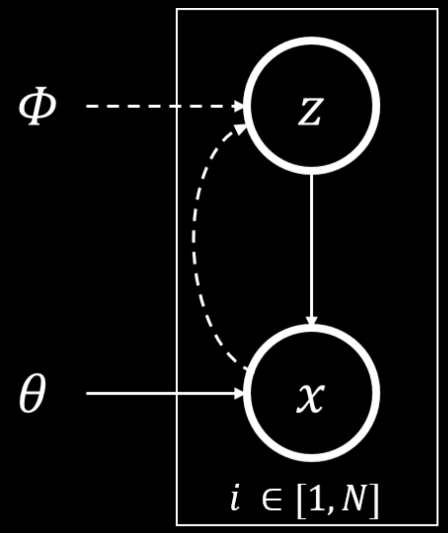
ML-VAE

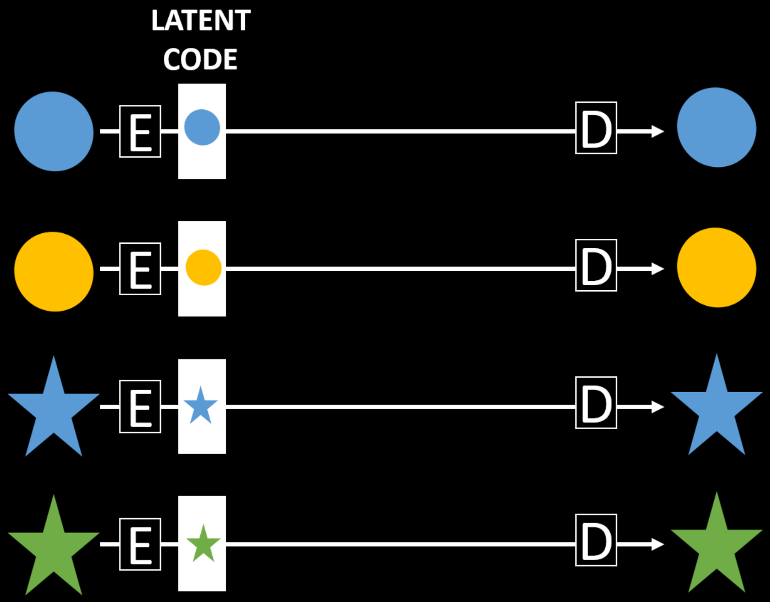
Independent, identically distributed **groups**
Amortised inference



VAE

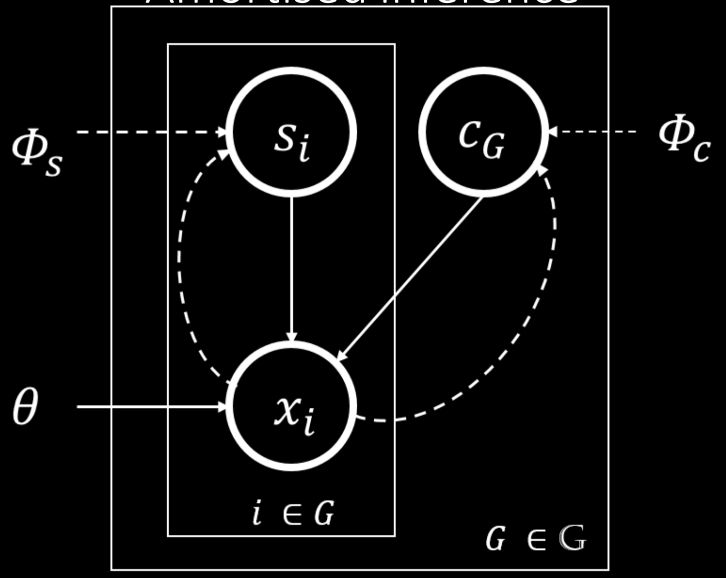
Independent, identically distributed samples
Amortised inference

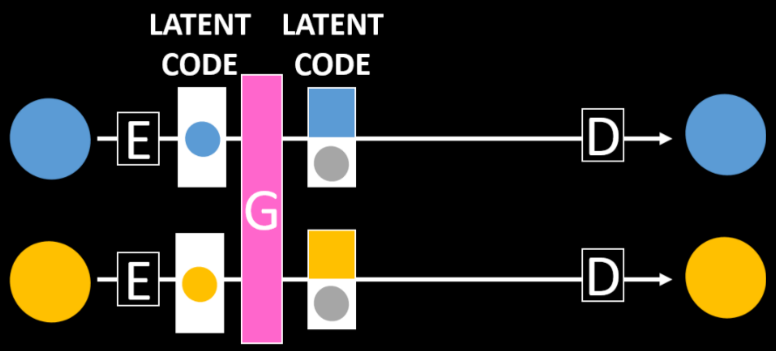




ML-VAE

Independent, identically distributed groups
Amortised inference



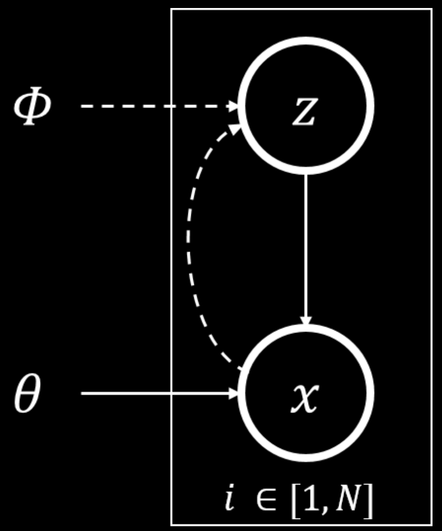


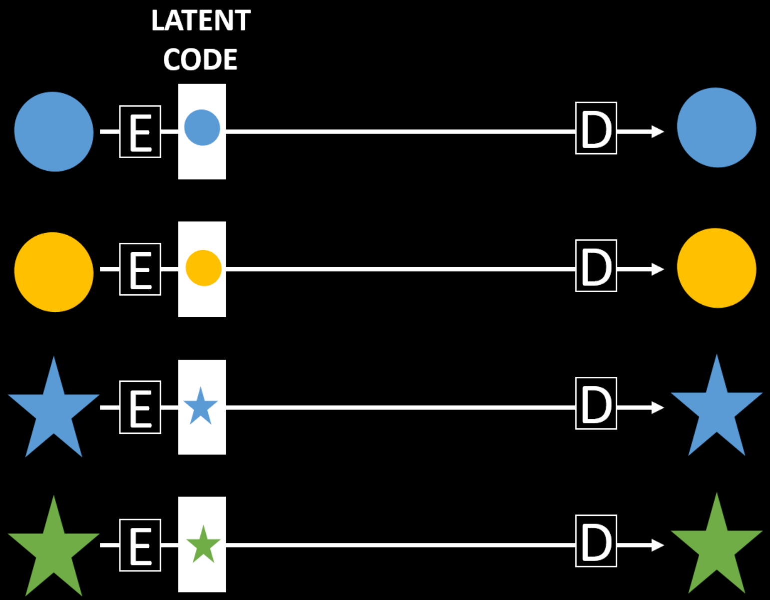
Grouping: build $Q(c_G|x_G;\Phi_c)$ from $Q(c_G|x_i;\Phi_c)$

VAE

Independent, identically distributed samples

Amortised inference

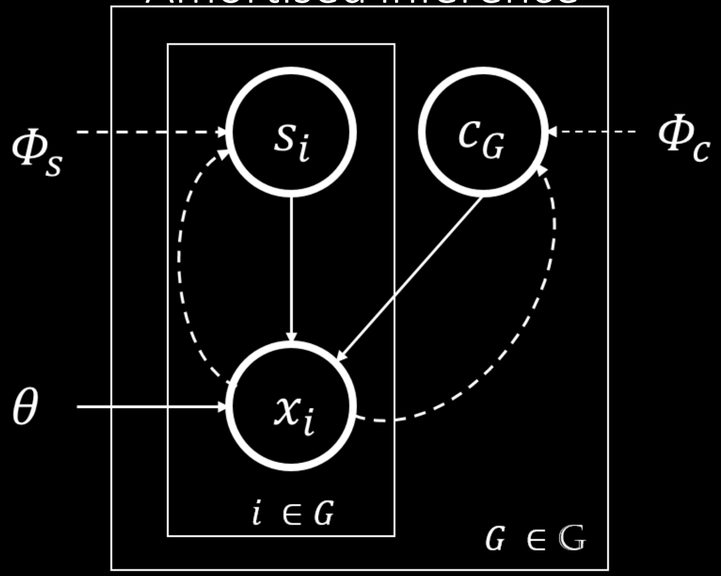


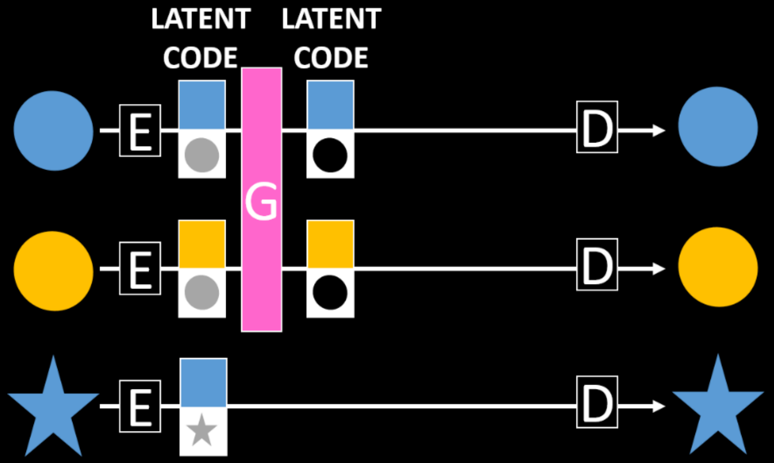


ML-VAE

Independent, identically distributed **groups**

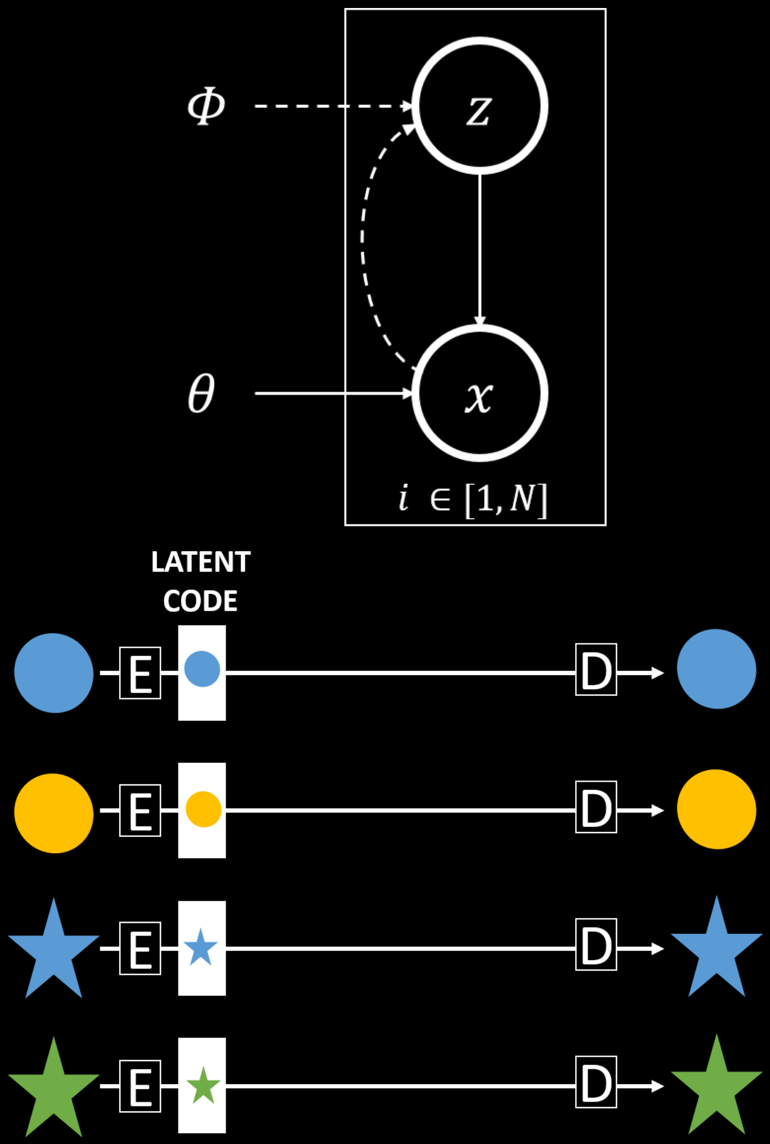
Amortised inference





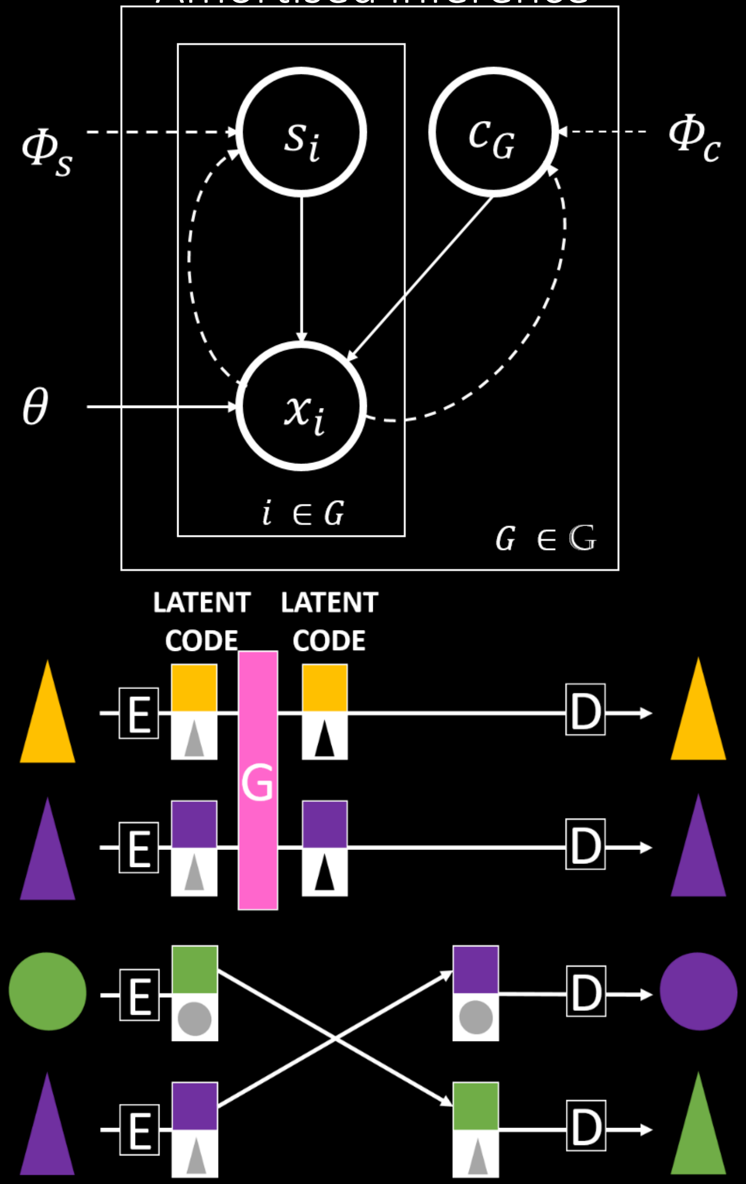
VAE

Independent, identically distributed samples
Amortised inference



ML-VAE

Independent, identically distributed groups
Amortised inference



Multi-Level VAE

- Maximise average Evidence Lower Bound

$$\frac{1}{|\mathrm{G}|}\log \mathrm{P}(x|\theta) = \frac{1}{|\mathrm{G}|}\sum_{G\in\mathrm{G}} \log \mathrm{P}(x_G|\theta) \geq \frac{1}{|\mathrm{G}|}\sum_{G\in\mathrm{G}} \mathrm{ELBO}_G$$

$$\mathrm{ELBO}_G = \underbrace{\sum_{i\in G} \mathbb{E}_{c_G\sim \mathrm{Q}(c_G|x_G;\Phi_c)}\left[\mathbb{E}_{s_i\sim \mathrm{Q}(s_i|x_i;\Phi_s)}\left[\log \mathrm{P}(x_i|z_i;\theta)\right]\right]}_{\text{Fit to the data}}$$

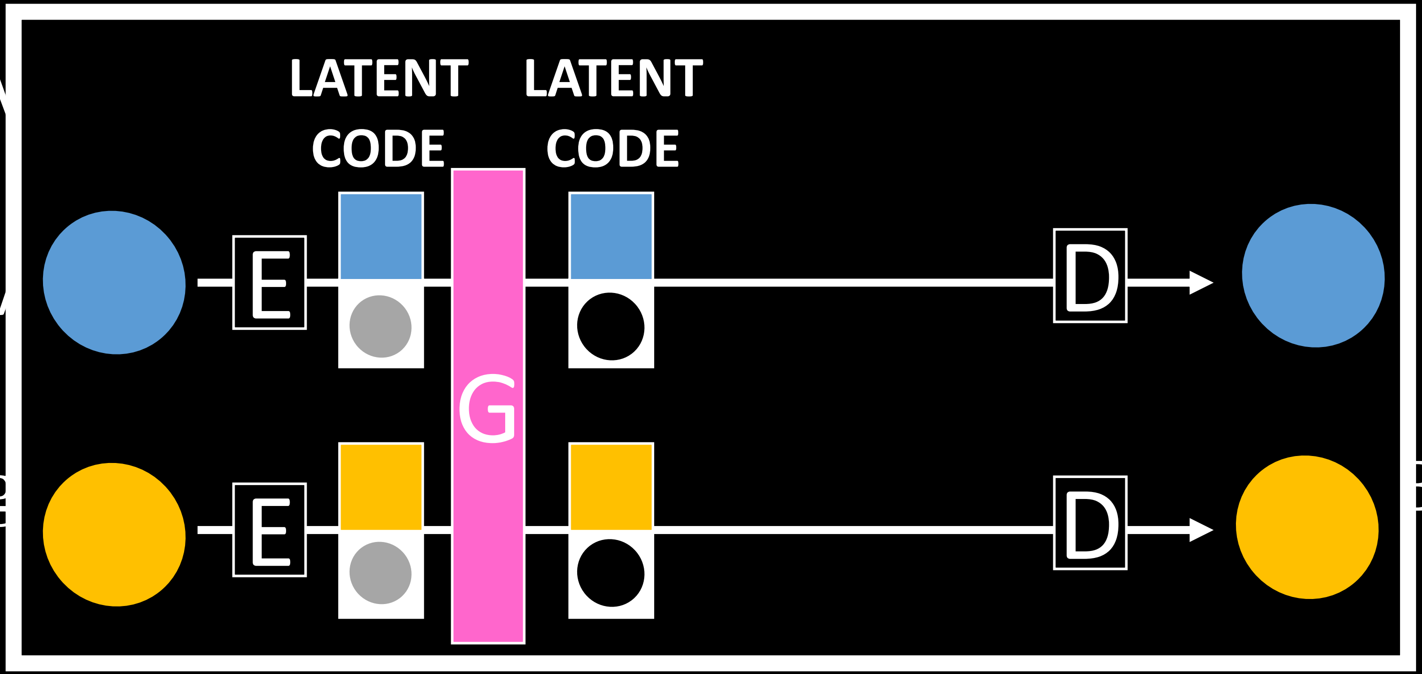
$$-\sum\nolimits_{i\in G} \underbrace{D_{\mathrm{KL}}(\mathrm{Q}(s_i|x_i;\Phi_s)||P(s_i)) - D_{\mathrm{KL}}(\mathrm{Q}(c_G|x_G;\Phi_c)||P(c_G))}_{\text{Regulariser}}$$

Multi-Lev

- Maximise av

$$\frac{1}{|G|}\log$$

$$\mathrm{BO}_G$$



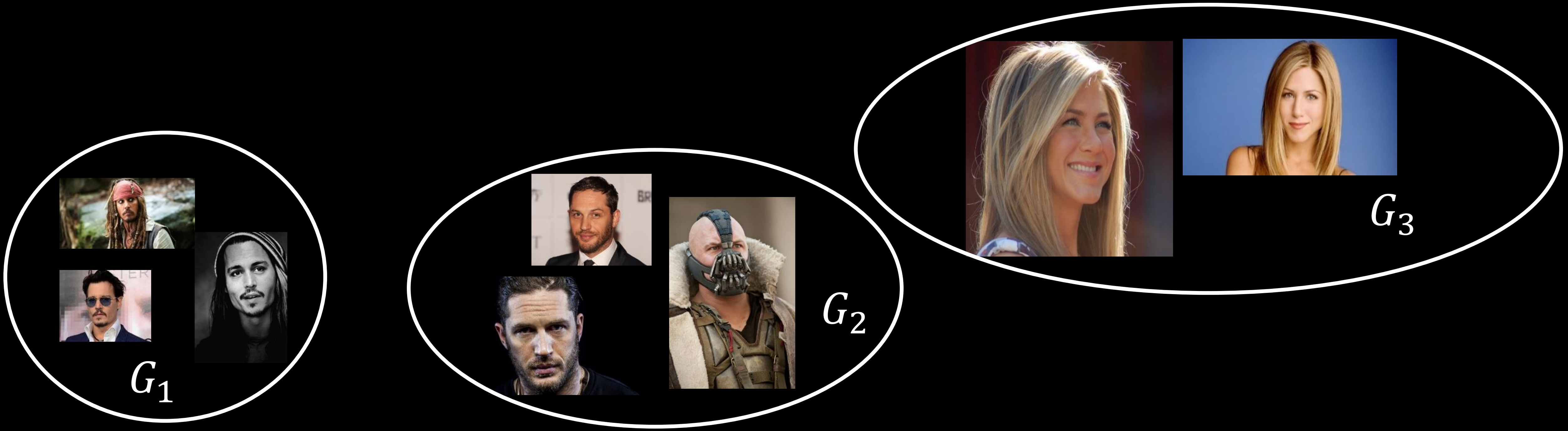
$$\mathrm{ELBO}_G = \underbrace{\sum_{i\in G} \mathbb{E}_{c_G \sim Q(c_G|x_G;\Phi_c)}\left[\mathbb{E}_{s_i \sim Q(s_i|x_i;\Phi_s)}\left[\log P(x_i|z_i;\theta)\right]\right]}_{\text{Fit to the data}}$$

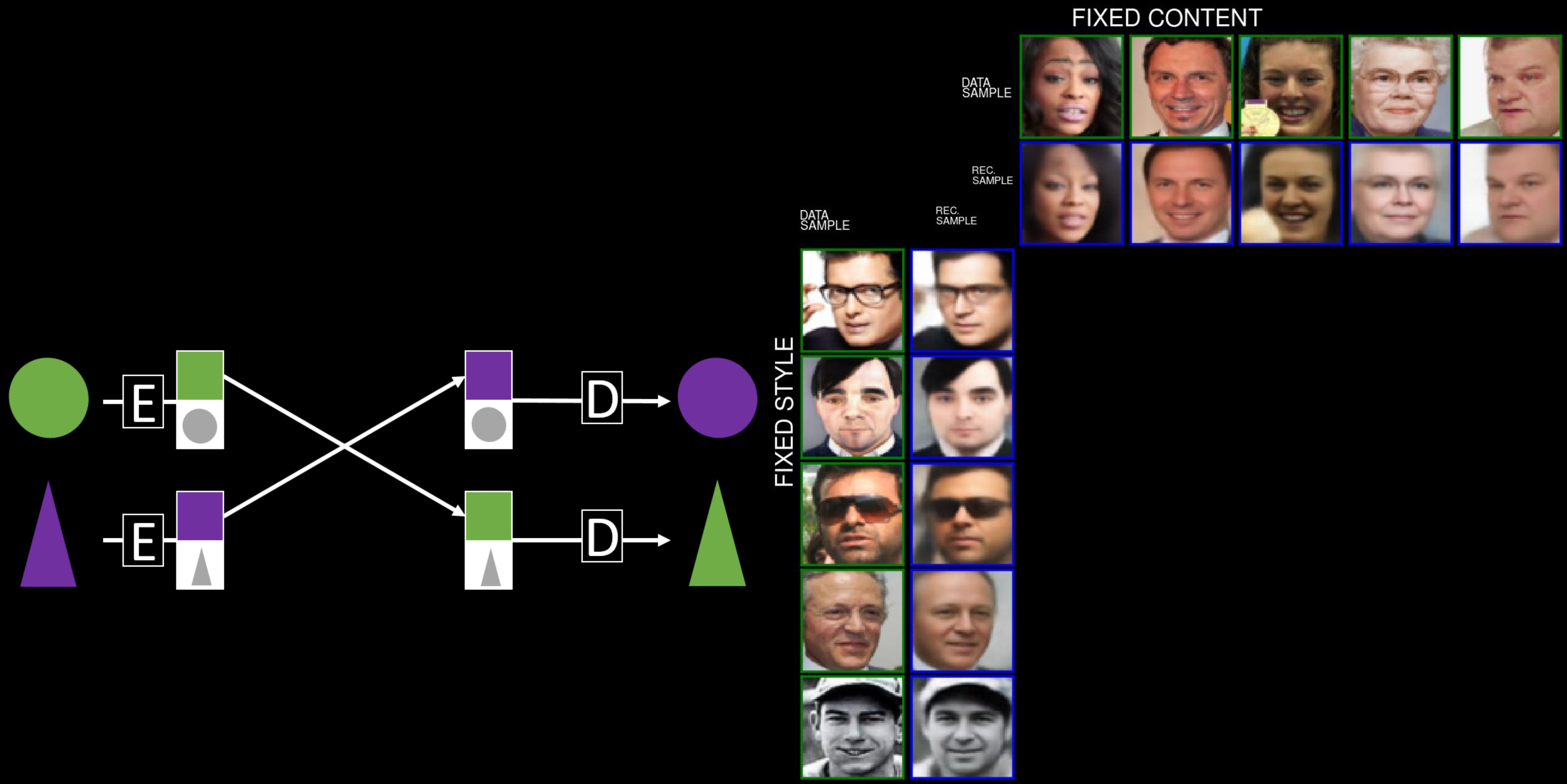
$$\underbrace{-\sum_{i\in G} D_{\mathrm{KL}}(Q(s_i|x_i;\Phi_s)||P(s_i)) - D_{\mathrm{KL}}(Q(c_G|x_G;\Phi_c)||P(c_G))}_{\text{Regulariser}}$$

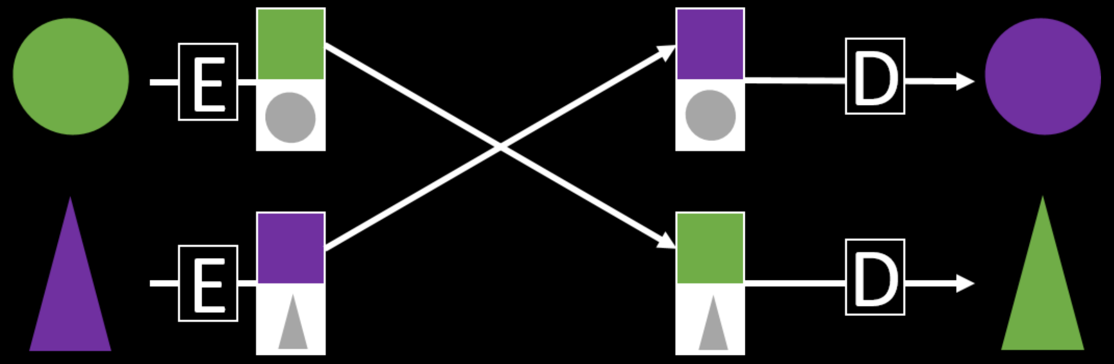
Experiments

MS-Celeb-1M dataset [MSR Asia]

- [Guo et al., 2016] celebrities face images
- Web queries per celebrity from popular search engines, with noise

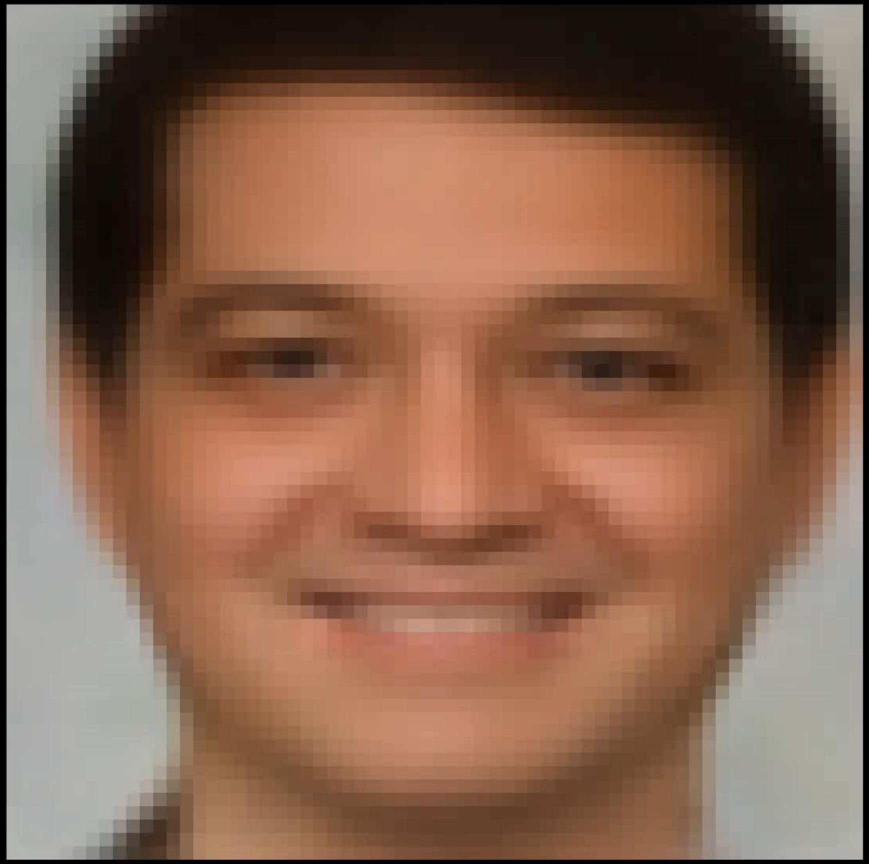



E
E
D
D
FIXED CONTENT
DATA SAMPLE
REC. SAMPLE
DATA SAMPLE
REC. SAMPLE
FIXED STYLE

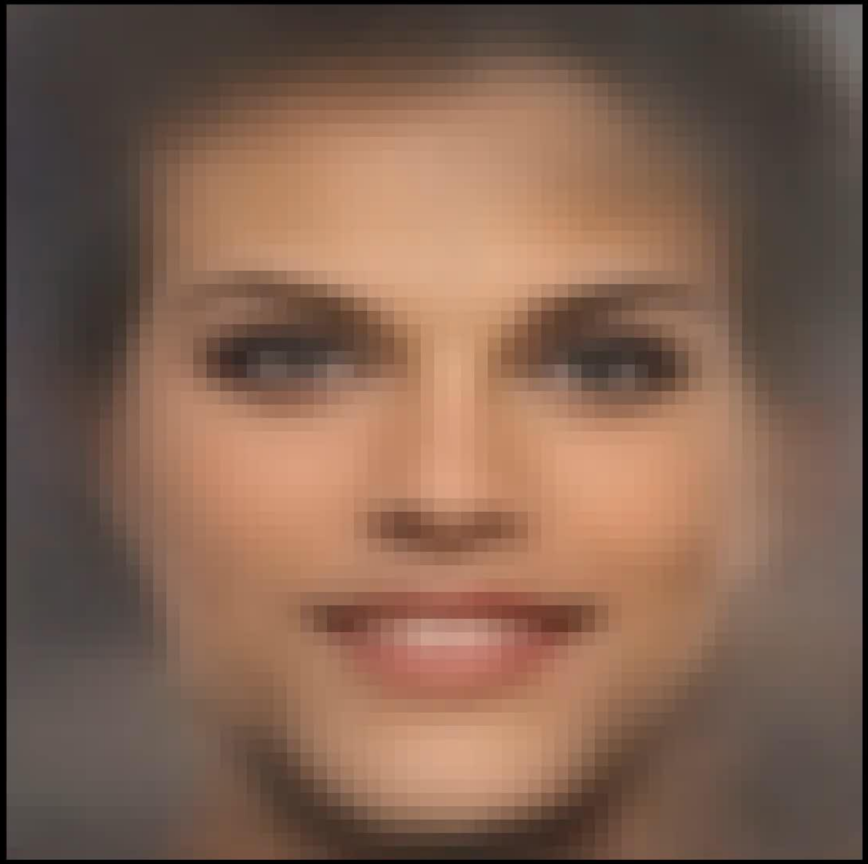


Control over the latent space

Same style, different ID



Same ID, different style



ML-VAE, summary

- Learns a useful disentangled representation
- Enables manipulation of the latent space
- Generalises to unseen groups

- Current work: text, controllable representations

Thanks!

Sebastian.Nowozin@microsoft.com

Additional Materials

GAN Archaeology

- Learning distributions by discriminative models
- Survey: [Mohamed and Lakshminarayanan, 2016]
- Partial history (in ML):
  [Tu, CVPR 2007]: generative model estimation via classification
  [Nguyen et al., 2010]: variational f-divergences
  [Sugiyama et al., 2012]: density ratio estimation
  [Gutmann and Hirayama, 2012], [Gutmann et al., 2014]

f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization

NIPS 2016

arXiv:1606.00709

Sebastian Nowozin, Botond Cseke, Ryota Tomioka

f-GAN Contributions

- Generalizes GAN objective to arbitrary f-divergences
- Simplifies the GAN algorithm
- Insights into choices of discriminator architectures

Estimating f-divergences from samples

[Nguyen, Wainwright, Jordan, 2010]

- Divergence between two distributions

$$D_f(P \parallel Q) = \int_{\mathcal{X}} q(x)\, f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x$$

f : generator function (convex & $f(1)=0$)

- Every convex function f has a *Fenchel conjugate* f^* so that

$$f(u) = \sup_{t \in \mathrm{dom}_{f^*}} \{tu - f^*(t)\}$$

"any convex f can be represented as point-wise max of *linear* functions"

Estimating f-divergences from samples (cont)

[Nguyen, Wainwright, Jordan, 2010]

$$D_f(P \parallel Q) = \int_{\mathcal{X}} q(x)\, f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x$$

$$= \int_{\mathcal{X}} q(x) \sup_{t \in \mathrm{dom}_{f^*}} \left\{ t \frac{p(x)}{q(x)} - f^*(t) \right\} \mathrm{d}x$$

$$\geq \sup_{T \in \mathcal{T}} \left(\int_{\mathcal{X}} q(x)\, T(x)\, \mathrm{d}x - \int_{\mathcal{X}} p(x)\, f^*\big(T(x)\big)\, \mathrm{d}x \right)$$

$$= \sup_{T \in \mathcal{T}} \Big(\underbrace{\mathbb{E}_{x \sim Q}[T(x)]}_{\text{samples from } Q} - \underbrace{\mathbb{E}_{x \sim P}[f^*(T(x))]}_{\text{samples from } P} \Big)$$

Approximate using: samples from Q samples from P

f-GAN and GAN objectives

- GAN

$$\min_{\theta} \max_{\omega} \mathbb{E}_{x\sim Q}[\log D_{\omega}(x)] + \mathbb{E}_{x\sim P_{\theta}}[\log(1 - D_{\omega}(x))]$$

- f-GAN

$$\min_{\theta} \max_{\omega} \left(\mathbb{E}_{x\sim Q}[T_{\omega}(x)] - \mathbb{E}_{x\sim P_{\theta}}[f^*(T_{\omega}(x))]\right)$$

  - GAN discriminator-variational function correspondence: $\log D_{\omega}(x) = T_{\omega}(x)$
  - GAN minimizes the Jensen-Shannon divergence (which was also pointed out in Goodfellow et al., 2014)

f-divergences

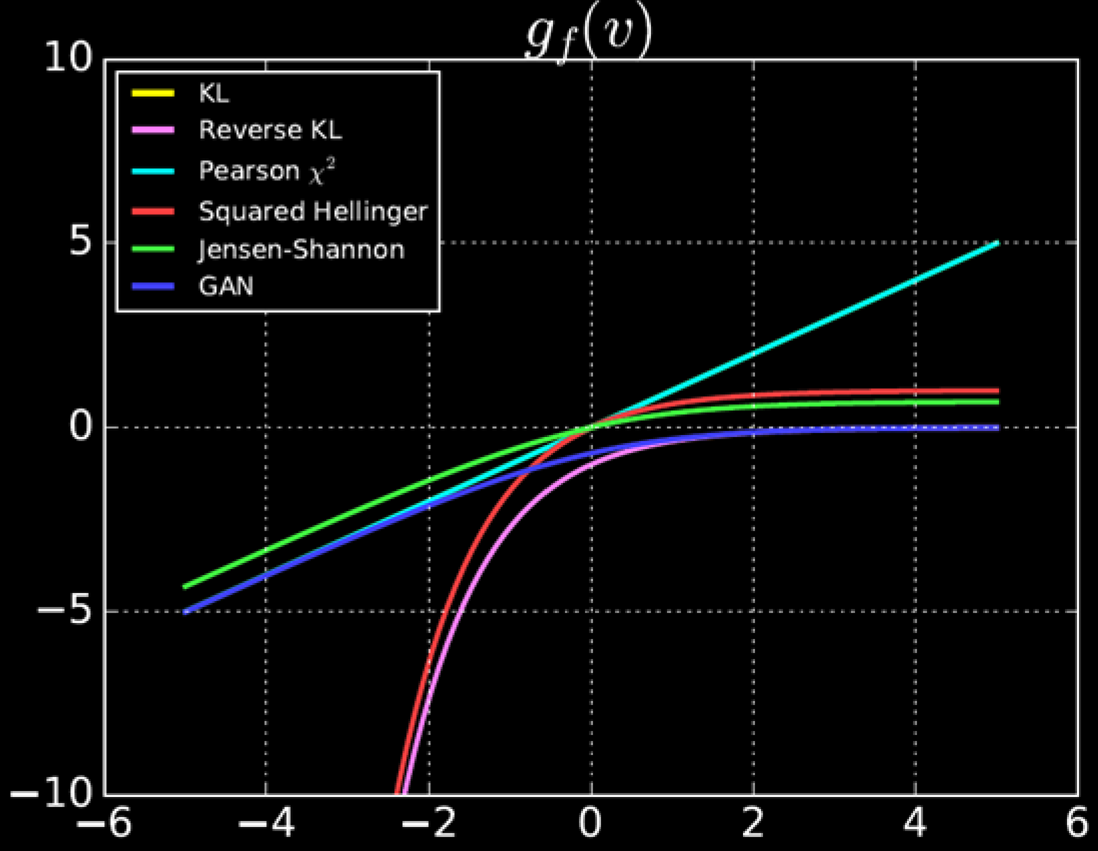
Name	$D_f(P\|Q)$	Generator $f(u)$
Total variation	$\frac{1}{2}\int \lvert p(x)-q(x)\rvert \,\mathrm{d}x$	$\frac{1}{2}\lvert u-1\rvert$
Kullback-Leibler	$\int p(x)\log\frac{p(x)}{q(x)}\,\mathrm{d}x$	$u\log u$
Reverse Kullback-Leibler	$\int q(x)\log\frac{q(x)}{p(x)}\,\mathrm{d}x$	$-\log u$
Pearson χ^2	$\int \frac{(q(x)-p(x))^2}{p(x)}\,\mathrm{d}x$	$(u-1)^2$
Neyman χ^2	$\int \frac{(p(x)-q(x))^2}{q(x)}\,\mathrm{d}x$	$\frac{(1-u)^2}{u}$
Squared Hellinger	$\int \left(\sqrt{p(x)}-\sqrt{q(x)}\right)^2\,\mathrm{d}x$	$(\sqrt{u}-1)^2$
Jeffrey	$\int (p(x)-q(x))\log\left(\frac{p(x)}{q(x)}\right)\,\mathrm{d}x$	$(u-1)\log u$
Jensen-Shannon	$\frac{1}{2}\int p(x)\log\frac{2p(x)}{p(x)+q(x)}+q(x)\log\frac{2q(x)}{p(x)+q(x)}\,\mathrm{d}x$	$-(u+1)\log\frac{1+u}{2}+u\log u$
Jensen-Shannon-weighted	$\int p(x)\pi\log\frac{p(x)}{\pi p(x)+(1-\pi)q(x)}+(1-\pi)q(x)\log\frac{q(x)}{\pi p(x)+(1-\pi)q(x)}\,\mathrm{d}x$	$\pi u\log u-(1-\pi+\pi u)\log(1-\pi+\pi u)$
GAN	$\int p(x)\log\frac{2p(x)}{p(x)+q(x)}+q(x)\log\frac{2q(x)}{p(x)+q(x)}\,\mathrm{d}x-\log(4)$	$u\log u-(u+1)\log(u+1)$
α-divergence ($\alpha\notin\{0,1\}$)	$\frac{1}{\alpha(\alpha-1)}\int\left(p(x)\left[\left(\frac{q(x)}{p(x)}\right)^{\alpha}-1\right]-\alpha(q(x)-p(x))\right)\mathrm{d}x$	$\frac{1}{\alpha(\alpha-1)}\left(u^{\alpha}-1-\alpha(u-1)\right)$

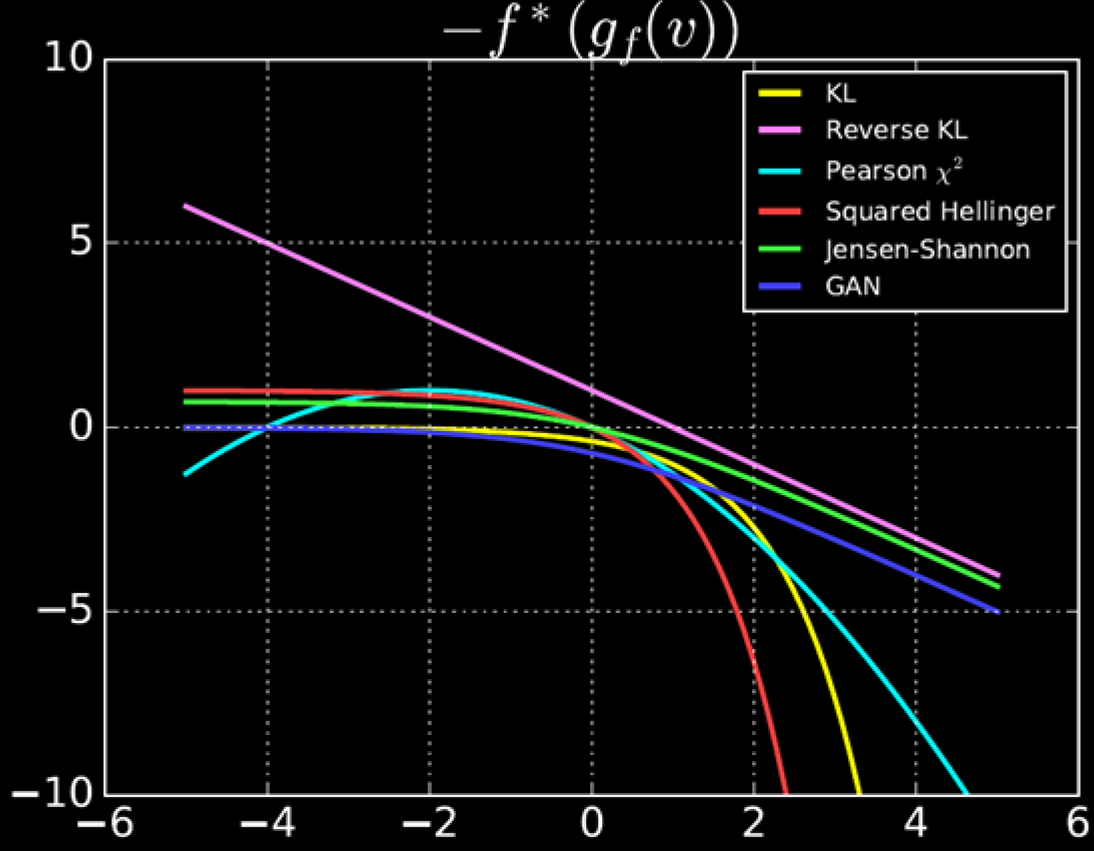
f-GAN

Name	Output activation g_f	dom_{f^*}	Conjugate $f^*(t)$	$f'(1)$
Total variation	$\frac{1}{2}\tanh(v)$	$-\frac{1}{2} \leq t \leq \frac{1}{2}$	t	0
Kullback-Leibler (KL)	v	$\mathbb{R}$	$\exp(t-1)$	1
Reverse KL	$-\exp(v)$	$\mathbb{R}_-$	$-1-\log(-t)$	-1
Pearson χ^2	v	$\mathbb{R}$	$\frac{1}{4}t^2+t$	0
Neyman χ^2	$1-\exp(v)$	$t<1$	$2-2\sqrt{1-t}$	0
Squared Hellinger	$1-\exp(v)$	$t<1$	$\frac{t}{1-t}$	0
Jeffrey	v	$\mathbb{R}$	$W(e^{1-t})+\frac{1}{W(e^{1-t})}+t-2$	0
Jensen-Shannon	$\log(2)-\log(1+\exp(-v))$	$t<\log(2)$	$-\log(2-\exp(t))$	0
Jensen-Shannon-weighted	$-\pi\log\pi-\log(1+\exp(-v))$	$t<-\pi\log\pi$	$(1-\pi)\log\frac{1-\pi}{1-\pi e^{t/\pi}}$	0
GAN	$-\log(1+\exp(-v))$	$\mathbb{R}_-$	$-\log(1-\exp(t))$	$-\log(2)$
α-div. ($\alpha<1, \alpha\neq 0$)	$\frac{1}{1-\alpha}-\log(1+\exp(-v))$	$t<\frac{1}{1-\alpha}$	$\frac{1}{\alpha}(t(\alpha-1)+1)^{\frac{\alpha}{\alpha-1}}-\frac{1}{\alpha}$	0
α-div. ($\alpha>1$)	v	$\mathbb{R}$	$\frac{1}{\alpha}(t(\alpha-1)+1)^{\frac{\alpha}{\alpha-1}}-\frac{1}{\alpha}$	0

Comparison of the objectives

$$\min_{\theta}\max_{\omega}\left(\mathbb{E}_{x\sim P}\left[g_f\left(V_\omega\left(x\right)\right)\right]+\mathbb{E}_{x\sim Q_\theta}\left[-f^*\left(g_f\left(V_\omega(x)\right)\right)\right]\right)$$





Implementing GANs

- “How to Train a GAN?”, Soumith Chintala
- Saddle-point problem versus two optimization problems
- Easy to make errors:
  - Optimization using “different generator objective” is broken: [Poole et al., 2016], highly recommended
- (More on the topic of GAN training later)

Outline

1. f-Divergences (GAN)
2. Proper Scoring Rules (VAE)
3. Integral Probability Metrics (DISCO, MMD, WGAN)
4. Current research areas

Proper Scoring Rules [Gneiting and Raftery, 2007]

- "Loss function for distributions":

$$S(P, Q) = \int S(P, x)\mathrm{d}Q(x)$$

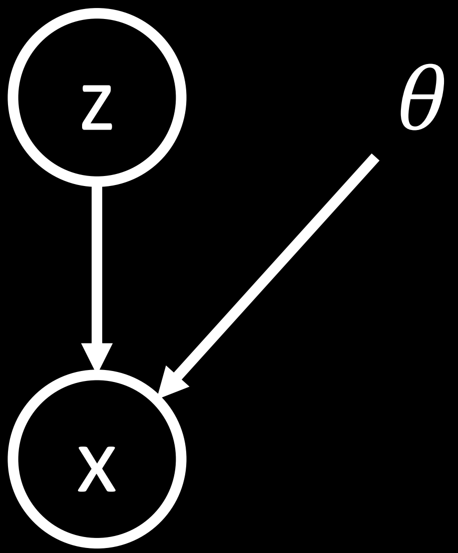
$$S(P, P) \le S(P, Q), \qquad \forall\, P, Q \in \mathcal{P}$$

- Discrete case: complete characterization (Savage representation)
- Continuous case, density function P
  - Log-likelihood [Good, 1952], $S(P, x) = \log P(x)$
  - Quadratic score [Bernado, 1979], $S(P, x) = 2\, P(x) - \|P\|_2^2$
  - Pseudospherical score [Good, 1971], $S(P, x) = P(x)^{\alpha-1} / \|P\|_\alpha^{\alpha-1}$

Variational Autoencoders (VAE)

[Kingma and Welling, 2014], [Rezende et al., 2015]

VAE: Model



$$p(x|\theta) = \int \mathrm{p(x|z,\theta)}p(z)\mathrm{d}z$$

- $p(z)$ is a multivariate standard Normal
- $p(x|z,\theta)$ is a neural network outputting a simple distribution (e.g. diagonal Normal)

VAE: Maximum Likelihood Training

- Maximize the data log-likelihood, per-instance variational approximation

$$
\begin{aligned}
\log p(x|\theta) &= \log \int \mathrm{p}(\mathrm{x}|\mathrm{z},\theta) p(z) \mathrm{d}z \\
&= \log \int \mathrm{p}(\mathrm{x}|\mathrm{z},\theta) \frac{q(z)}{q(z)} p(z)\, \mathrm{d}z \\
&= \log \int \mathrm{p}(\mathrm{x}|\mathrm{z},\theta) \frac{p(z)}{q(z)} q(z)\, \mathrm{d}z \\
&= \log \mathbb{E}_{z\sim q(z)} \left[\mathrm{p}(\mathrm{x}|\mathrm{z},\theta) \frac{p(z)}{q(z)} \right] \\
&\geq \mathbb{E}_{z\sim q(z)} \left[\log \mathrm{p}(\mathrm{x}|\mathrm{z},\theta) \frac{p(z)}{q(z)} \right] \\
&= \mathbb{E}_{z\sim q(z)} [\log \mathrm{p}(\mathrm{x}|\mathrm{z},\theta)] - D_{\mathrm{KL}}(q(z) \parallel p(z))
\end{aligned}
$$

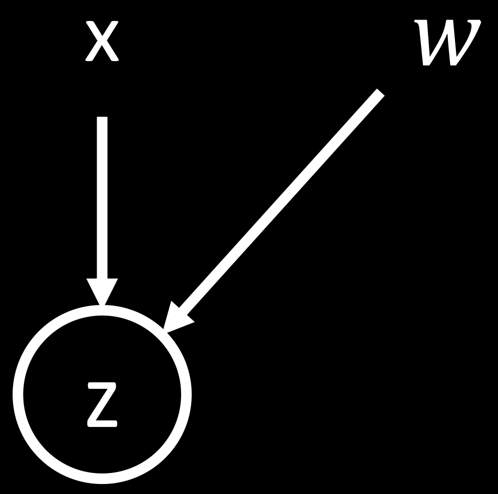
Inference networks

- Amortized inference [Stuhlmüller et al., NIPS 2013]
- Inference networks, recognition networks [Kingma and Welling, 2014]
- “Informed sampler” [Jampani et al., 2014]
- “Memory-based approach” [Kulkarni et al., 2015]

Inference networks

x_1 x_2 x_3

$q_1(z)$ $q_2(z)$ $q_3(z)$



VAE: Maximum Likelihood Training

- Maximize the data log-likelihood, inference network variational approximation

$$\log p(x|\theta) = \log \int \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) p(z) \mathrm{dz}$$

$$= \log \int \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) \frac{q(z|x, w)}{q(z|x, w)} p(z)\, \mathrm{dz}$$

$$= \log \int \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) \frac{p(z)}{q(z|x, w)} q(z|x, w)\, \mathrm{dz}$$

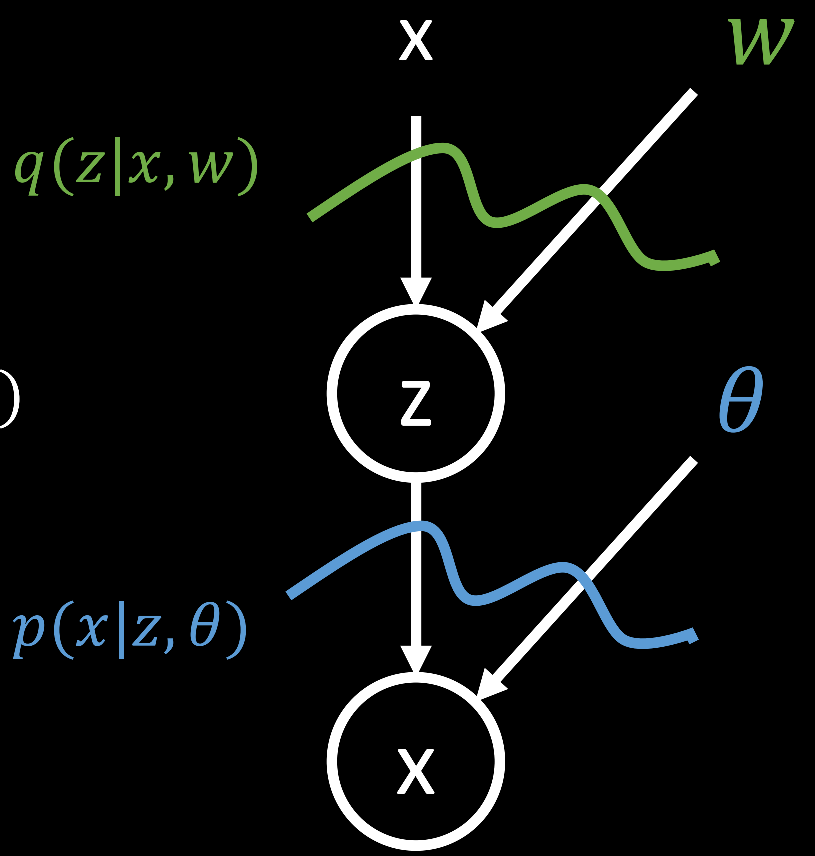
$$= \log \mathbb{E}_{z \sim q(z|x,w)} \left[\mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) \frac{p(z)}{q(z|x, w)} \right]$$

$$\geq \mathbb{E}_{z \sim q(z|x,w)} \left[\log \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) \frac{p(z)}{q(z|x, w)} \right]$$

$$= \mathbb{E}_{z \sim q(z|x,w)} [\log \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta)] - D_{\mathrm{KL}}(q(z|x, w) \parallel p(z))$$

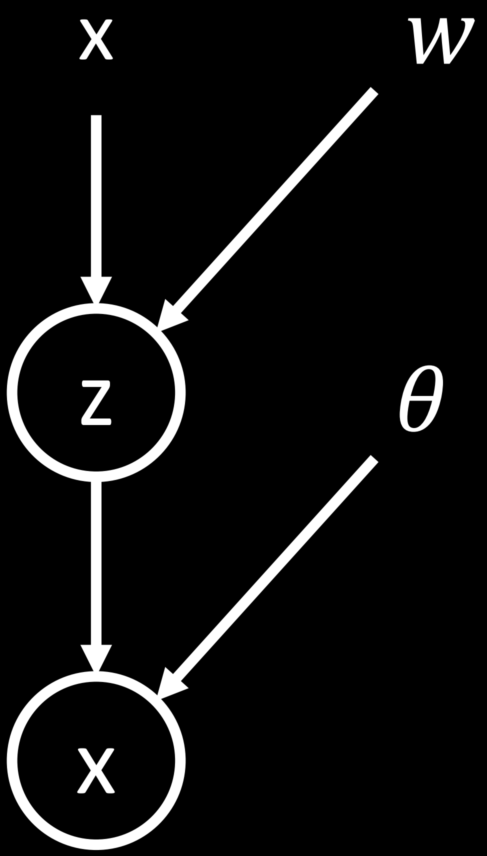
Autoencoder viewpoint

$$\max_{w,\theta} \; \mathbb{E}_{z\sim q(z|x,w)}[\log \mathrm{p}(\mathrm{x}|\mathrm{z},\theta)] - D_{\mathrm{KL}}(q(z|x,w) \parallel p(z))$$



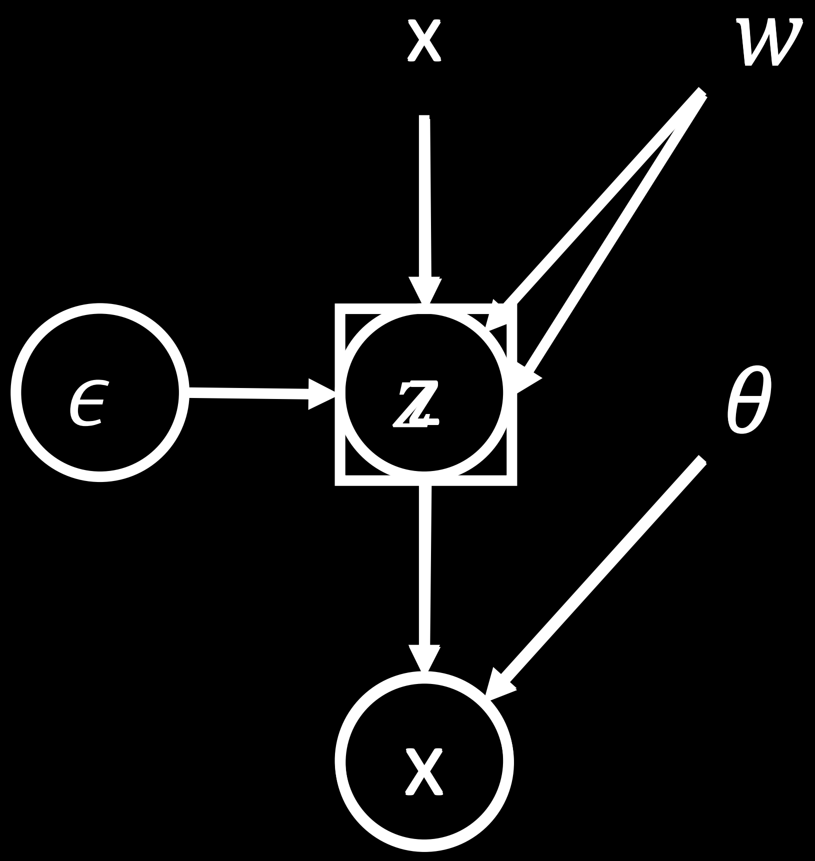
Reparametrization Trick

- [Rezende et al., 2014]
  [Kingma and Welling, 2014]

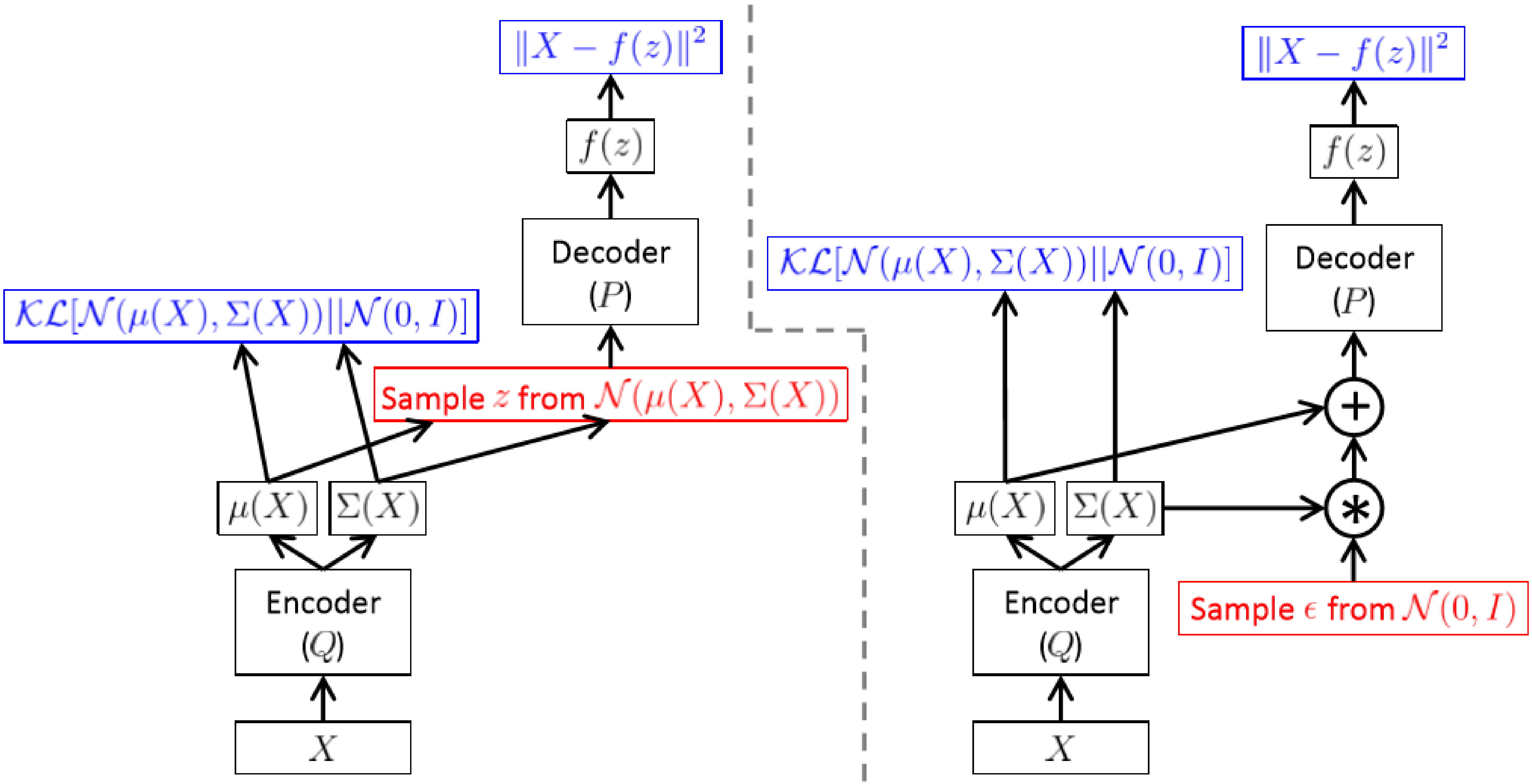


Reparametrization Trick

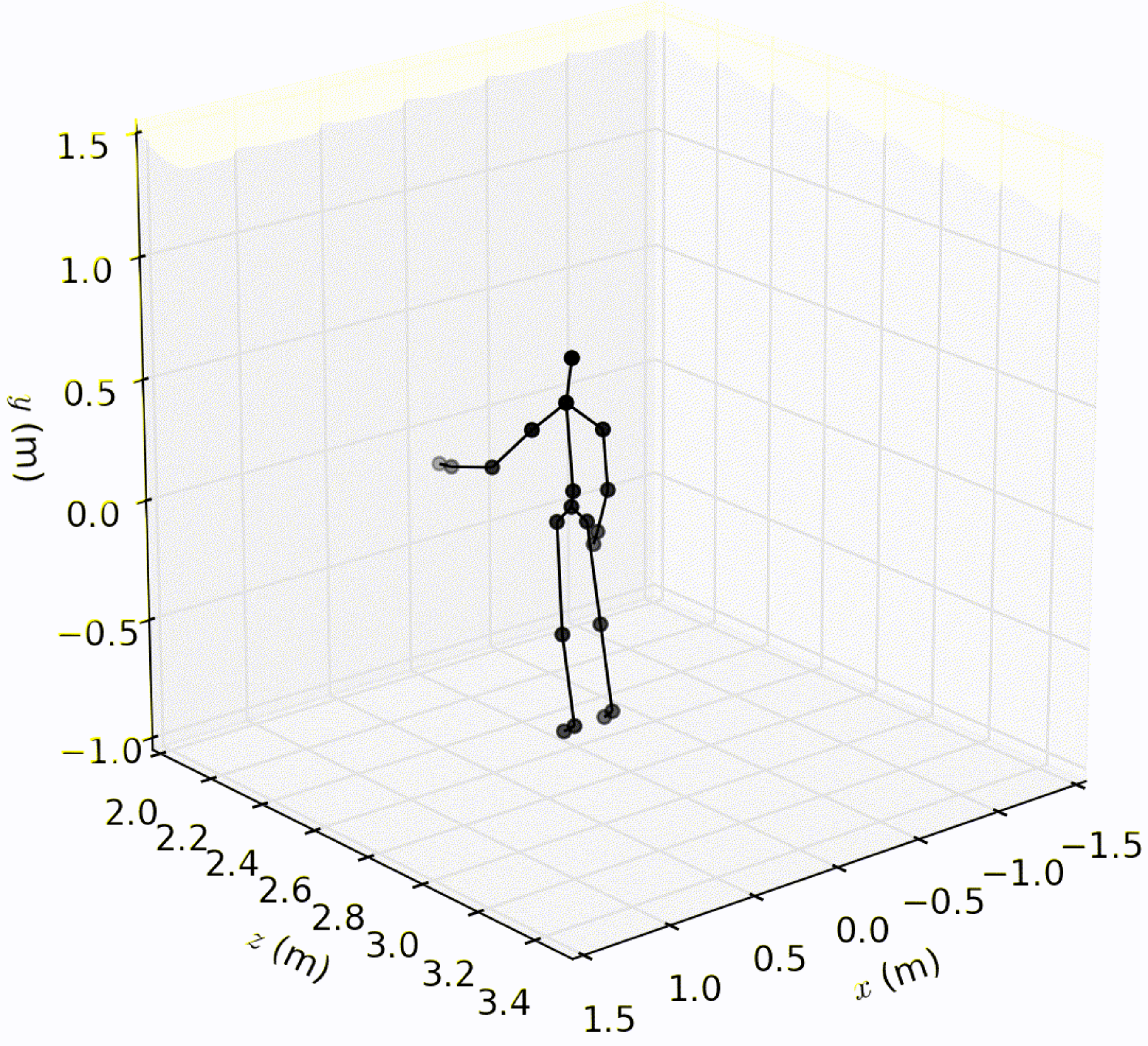
- [Rezende et al., 2014]
  [Kingma and Welling, 2014]
- Stochastic computation graphs
  [Schulman et al., 2015]



From highly-recommended tutorial: [Doersch, “Tutorial on Variational Autoencoders”, arXiv:1606.05908]



```
def encode(self, x):
    h = F.crelu(self.qlin0(x))
    h = F.crelu(self.qlin1(h))
    h = F.crelu(self.qlin2(h))
    h = F.crelu(self.qlin3(h))

    self.qmu = self.qlin_mu(h)
    self.qln_var = self.qlin_ln_var(h)

def decode(self, z):
    h = F.crelu(self.plin0(z))
    h = F.crelu(self.plin1(h))
    h = F.crelu(self.plin2(h))
    h = F.crelu(self.plin3(h))

    self.pmu = self.plin_mu(h)
    self.pln_var = self.plin_ln_var(h)

def __call__(self, x):
    # Compute q(z|x)
    self.encode(x)

    self.kl = gaussian_kl_divergence(self.qmu, self.qln_var)
    self.logp = 0
    for j in xrange(self.num_zsamples):
        # z ~ q(z|x)
        z = F.gaussian(self.qmu, self.qln_var)

        # Compute p(x|z)
        self.decode(z)

        # Compute objective
        self.logp += gaussian_logp(x, self.pmu, self.pln_var)

    self.logp /= self.num_zsamples
    self.obj = self.kl - self.logp

    return self.obj
```



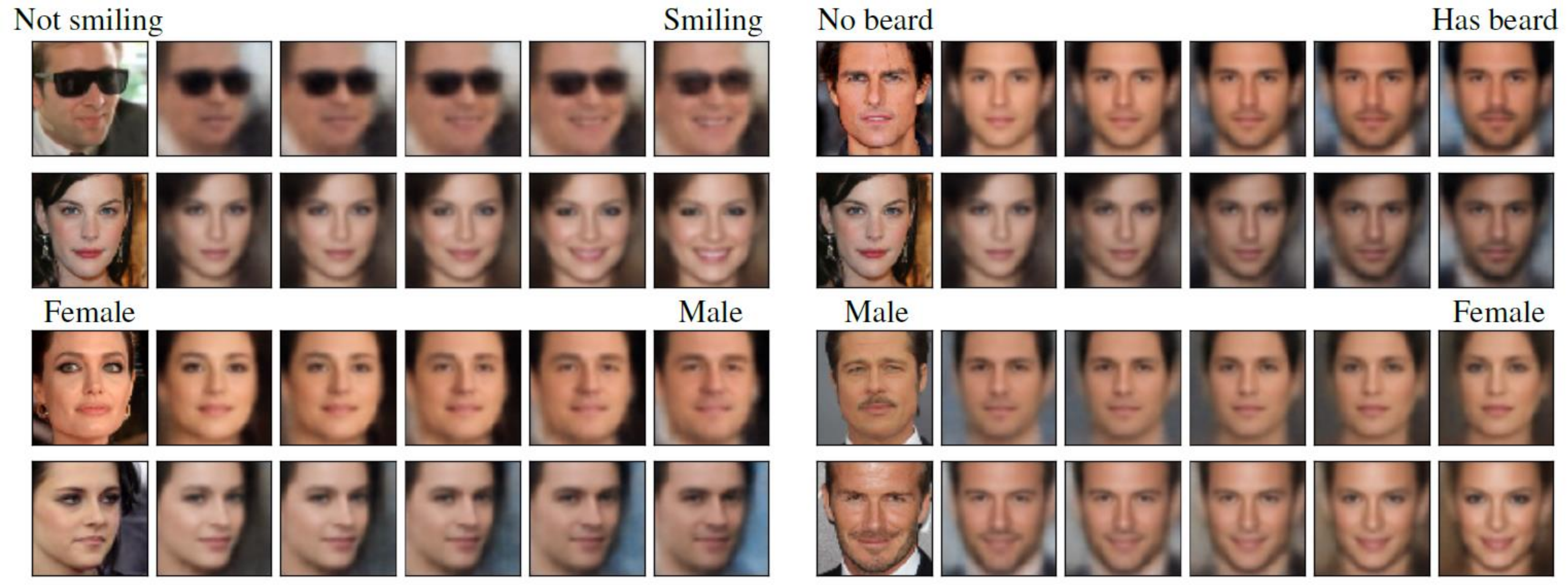
Motivation: Problems in VAEs (as of 2017)

- Inadequate inference networks
  - Loose ELBO
  - Limits what the generative model can learn

- Parametric conditional likelihood assumptions
  - Limits the expressivity of the generative model
  - "Noise term has to explain too much"

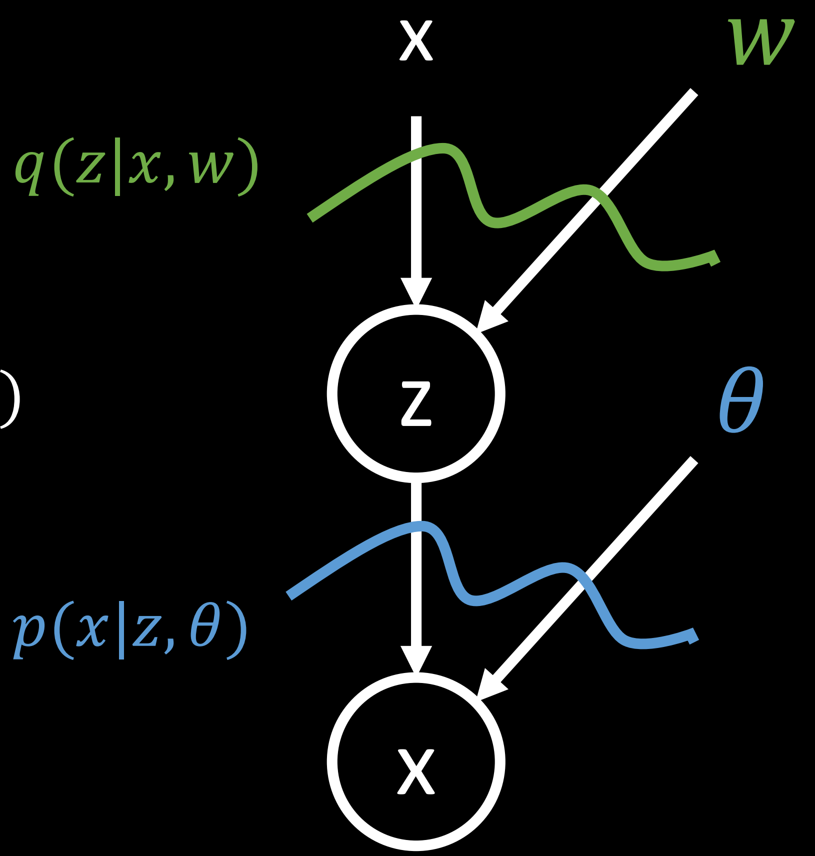
“Blurry images” in VAE models

from [Tulyakov, Fitzgibbon, Nowozin, ICCV 2017]



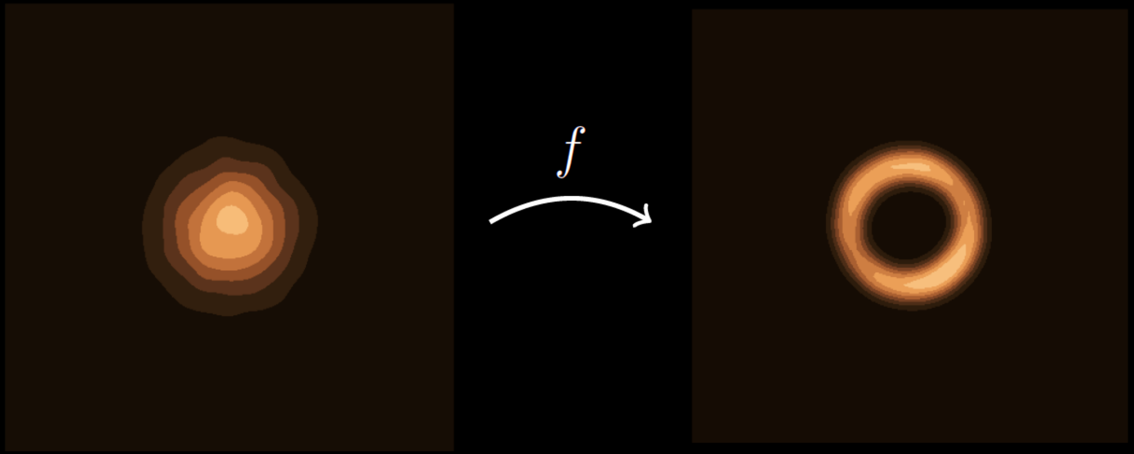
Autoencoder viewpoint

$$\max_{w,\theta} \; \mathbb{E}_{z\sim q(z|x,w)}[\log \mathrm{p}(\mathrm{x}|\mathrm{z},\theta)] - D_{\mathrm{KL}}(q(z|x,w) \parallel p(z))$$



Improving Inference Networks

- State of the art in inference network design:
  - NICE [Dinh et al., 2015]
  - Hamiltonian variational inference (HVI) [Salimans et al., 2015]
  - Importance weighted autoencoder (IWAE) [Burda et al., 2016]
  - Normalizing flows [Rezende and Mohamed, 2016]
  - Auxiliary deep generative networks [Maaløe et al., 2016]
  - Inverse autoregressive flow (IAF) [Kingma et al., NIPS 2016]
  - Householder flows [Tomczak and Welling, 2017]
  - Adversarial variational Bayes (AVB) [Mescheder et al., 2017]
  - Deep and Hierarchical Implicit Models [Tran et al., 2017]
  - Variational Inference using Implicit Distributions [Huszár, 2017]



Adversarial Variational Bayes:

Unifying Variational Autoencoders and Generative Adversarial Networks

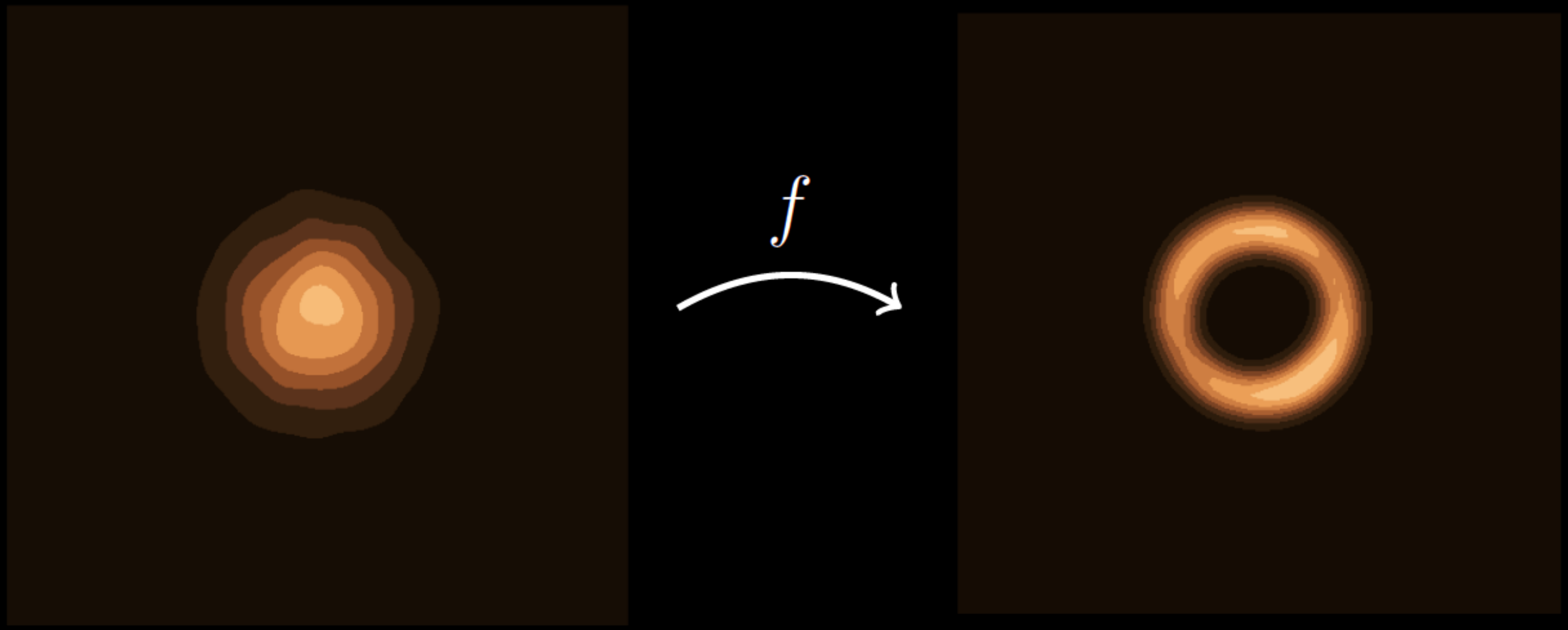
Lars Mescheder, Sebastian Nowozin, Andreas Geiger

ICML 2017 submission

arXiv:1701.04722

High-level idea: 1/2

- What do we currently require from $q(z|x, w)$?
  - Sampling: $z \sim q(z|x, w)$
  - ~~Log-density computation: $\log q(z|x, w)$~~



High-level idea: 2/2

$$\mathbb{E}_{z\sim q(z|x,w)}[\log \mathrm{p}(\mathrm{x}|\mathrm{z},\theta)] - D_{\mathrm{KL}}\big(q(z|x,w) \parallel p(z)\big)$$
$$= \mathbb{E}_{z\sim q(z|x,w)}[\log \mathrm{p}(\mathrm{x}|\mathrm{z},\theta) + \log p(z) - \log q(z|x,w)]$$

- Introduce a real-valued discriminator function $T(x,z)$ such that

$$T(x,z) \approx -\log p(z) + \log q(z|x,w)$$

Variational Approximation

$$\max_{T \epsilon \mathcal{T}} \mathbb{E}_{x \sim p_D}\left[\mathbb{E}_{z \sim q(z|x,w)}[\log \sigma(T(x,z))] + \mathbb{E}_{z \sim p(z)}[\log(1 - \sigma(T(x,z)))]\right]$$

Proposition:

For $q(z|x,w)$ fixed the optimal discriminator T^* is given by

$$T^*(x,z) = -\log p(z) + \log q(z|x,w)\,.$$

Rewrite

$$\mathbb{E}_{z \sim q(z|x,w)}[\log \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) + \log p(z) - \log q(z|x,w)]$$
$$= \mathbb{E}_{z \sim q(z|x,w)}\,[\log \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) - T^*(x,z)\,]$$

Reparametrization Trick

- Learning objective

$$\mathbb{E}_{z\sim q(z|x,w)}[\log \mathrm{p}(\mathrm{x}|\mathrm{z},\theta) - T^*(x,z)\,]$$

- Reparametrize [Kingma and Welling, 2013]

$$z\sim q(z|x,w) \Leftrightarrow \varepsilon\sim\mathcal{N}, z(x,w,\varepsilon)$$

- Reparametrized learning objective

$$\mathbb{E}_{\varepsilon}[\log \mathrm{p}(\mathrm{x}|\mathrm{z}(\mathrm{x},\mathrm{w},\varepsilon),\theta) - T^*(x,z(x,w,\varepsilon))\,]$$

Variational Approximation (Discriminator)

$$\max_{T \in \mathcal{T}} \mathbb{E}_{x \sim p_D}\left[\mathbb{E}_{z \sim q(z|x,w)}[\log \sigma(T(x,z))] + \mathbb{E}_{z \sim p(z)}[\log(1 - \sigma(T(x,z)))]\right]$$

Variational adversary

Data distribution (fixed)

Variational distribution (fixed)

Adversarial Variational Bayes

$$\max_{\theta, w} \mathbb{E}_{\varepsilon}[\log \mathrm{p}(\mathrm{x}|\mathrm{z}(\mathrm{x}, \mathrm{w}, \varepsilon), \theta) - T(x, z(x, w, \varepsilon), \psi)\ ]$$

$$\max_{\psi} \mathbb{E}_{x \sim p_D}\left[\mathbb{E}_{z \sim q(z|x,w)}[\log \sigma(T(x, z, \psi))] + \mathbb{E}_{z \sim p(z)}[\log(1 - \sigma(T(x, z, \psi)))]\right]$$

- Parameter-free expectation form → unbiased estimation
- GAN-type algorithm

Algorithm 1 Adversarial Variational Bayes (AVB)

$i = 0$

while not converged **do**

  Sample m examples $\{x^{(1)}, \dots, x^{(m)}\}$ from data distribution $p_{\mathcal{D}}(x)$.

  Sample m examples $\{z^{(1)}, \dots, z^{(m)}\}$ from prior distribution $p(z)$.

  Sample m noise examples $\{\epsilon^{(1)}, \dots, \epsilon^{(m)}\}$ from $\mathcal{N}(0, 1)$.

  Compute θ-gradient (eq. 3.9):

$$g_\theta = \nabla_\theta \frac{1}{m} \sum_{i=1}^{m} \log p_\theta \left(x^{(i)} \mid z_\phi \left(x^{(i)}, \epsilon^{(i)} \right) \right)$$

  Compute ϕ-gradient (eq. 3.9):

$$g_\phi = \nabla_\phi \frac{1}{m} \sum_{i=1}^{m} \Big[-T_\psi \left(x^{(i)}, z_\phi(x^{(i)}, \epsilon^{(i)}) \right) + \log p_\theta \left(x^{(i)} \mid z_\phi(x^{(i)}, \epsilon^{(i)}) \right) \Big].$$

  Compute ψ-gradient (eq. 3.3) :

$$g_\psi = \nabla_\psi \frac{1}{m} \sum_{i=1}^{m} \Big[\log \left(\sigma(T_\psi(x^{(i)}, z_\phi(x^{(i)}, \epsilon^{(i)}))) \right) + \log \left(1 - \sigma(T_\psi(x^{(i)}, z^{(i)})) \right) \Big].$$

  Perform SGD-updates for θ, ϕ and ψ:

$$\theta = \theta + h_i\, g_\theta, \quad \phi = \phi + h_i\, g_\phi, \quad \psi = \psi - h_i\, g_\psi.$$

  $i = i + 1$

end while

More Details in the Paper

- Connections to AAE/ALI and f-GAN
- Theory regarding approximation

Experiments

Binarized MNIST

- 28x28 binary images
- 50,000 training images
- 10,000 test images
- Train VAE model

- Report test ELBO
- Report Annealed importance sampling (AIS) estimates of test log-likelihood

Binarized MNIST density estimation

	ELBO	AIS	
AVB (8-dim)	$\approx -83.6 \pm 0.4$	-90.8 ± 1.0	
AVB + AC (8-dim)	$\approx -96.3 \pm 0.4$	-89.7 ± 1.0	
AVB + AC (32-dim)	$\approx -79.5 \pm 0.3$	-80.3 ± 0.6	
VAE (8-dim)	-98.0 ± 0.5	-91.0 ± 0.9	
VAE (32-dim)	-87.2 ± 0.3	-82.1 ± 0.6	
VAE + HF (T=2)	-79.5	–	(Tomczak & Welling, 2016)
VAE + NF (T=80)	-85.1	–	(Rezende & Mohamed, 2015)
VAE + NICE (T=80)	-88.3	–	(Dinh et al., 2014)
VAE + HVI (T=16)	-88.3	–	(Salimans et al., 2015)
convVAE + HVI (T=16)	-84.1	–	(Salimans et al., 2015)
VAE + VGP (2hl)	-81.3	–	(Tran et al., 2015)
DRAW + VGP	-79.9	–	(Tran et al., 2015)
VAE + IAF	-80.8	–	(Kingma et al., 2016)
Auxiliary VAE (L=2)	-83.0	–	(Maaløe et al., 2016)

Dataset samples

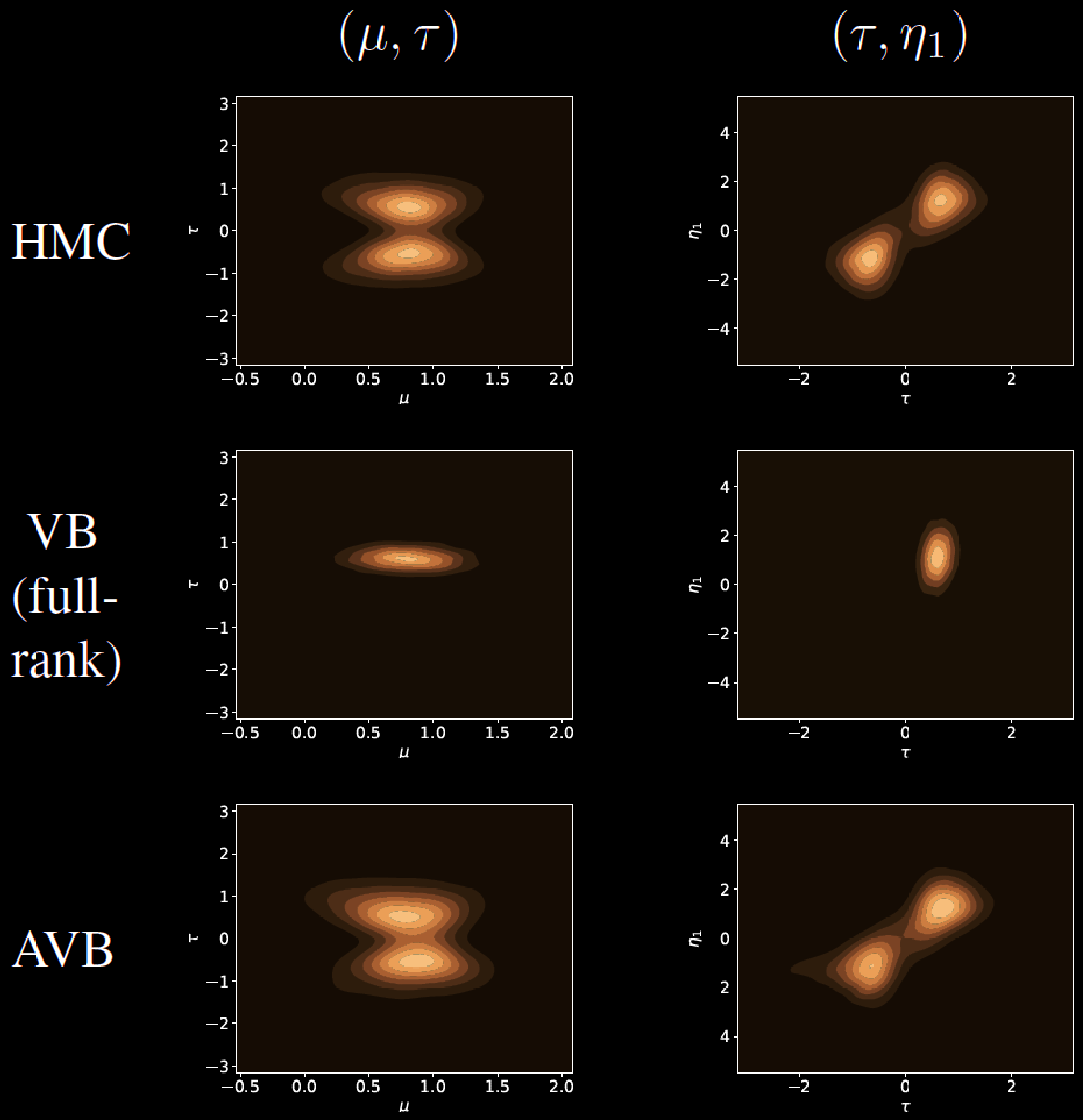
Model samples

CelebA face images

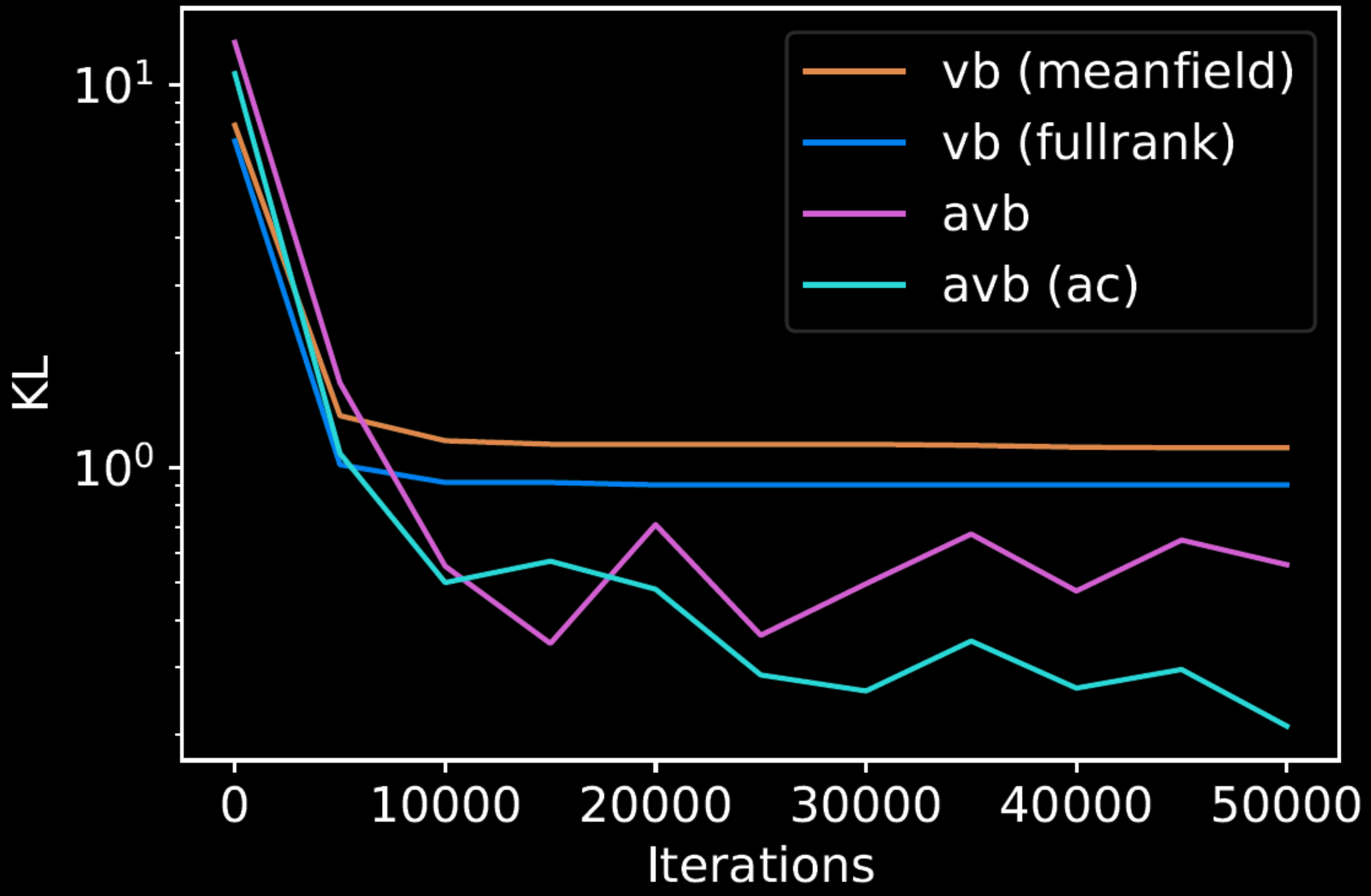
VB for Parameter Inference

- Stan [Stan Development Team, 2016]
- *Eight schools* model [Gelman et al., 2014]
- Eight parameters

- Ground truth posterior:
  MCMC with Hamiltonian Monte Carlo, 500k iterations (PyStan)
- Estimate KL divergence to true posterior
  - ITE package [Szabo, 2013]


(μ, τ)
(τ, η_1)
HMC
VB
(full-
rank)
AVB
μ
τ
η_1
−0.5 0.0 0.5 1.0 1.5 2.0
−3 −2 −1 0 1 2 3
−2 0 2
−4 −2 0 2 4

KL to true posterior



Conclusions

- AVB: likelihood-free variational families
- State-of-the-art performance in competitive VAE field
- Parameter Variational Bayes in large variational families

- Our TensorFlow implementation coming soon!
  - Third party implementation from Ben Poole: https://gist.github.com/poolio/b71eb943d6537d01f46e7b20e9225149

Outline

1. f-Divergences (GAN)
2. Proper Scoring Rules (VAE)
3. Integral Probability Metrics (DISCO, MMD, WGAN)
4. Current research areas

Kernel Two-Sample Tests

- [Gretton et al., “A Kernel Two-sample Test”, JMLR 2012]

  Maximum Mean Discrepancy (MMD)

$$\gamma_{\mathcal{F}}(P, Q) = \sup_{f \in \mathcal{F}} \left| \int f \mathrm{d}P - \int f \mathrm{d}Q \right|$$

Kernel Two-Sample Tests

- If $\mathcal{F}$ is a unit-ball in a reproducing kernel Hilbert space $\mathcal{H}$ we have

$$\gamma_{\mathcal{F}}(P,Q) = \sup_{f\in\mathcal{F}} \left| \int f \mathrm{d}P - \int f \mathrm{d}Q \right| = \|\mu_P - \mu_Q\|_{\mathcal{H}}$$

- Kernel mean embedding of a probability measure

$$\mu_P = \int k(x,\cdot) P(\mathrm{d}x)$$

- Estimator given sample $X = (x_1, \cdots, x_N)$ and $Y = (y_1, \cdots, y_M)$

$$\mathrm{MMD}^2(X,Y) = \frac{1}{N(N-1)} \sum_{n\neq n'} k(x_n, x_{n'}) + \frac{1}{M(M-1)} \sum_{m\neq m'} k(y_m, y_{m'}) - \frac{2}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} k(x_n, y_m)$$

Kernel MMD Training in Deep Learning

- Deep generative models
  [Li et al., 2015], [Dziugaite et al., 2015]
- Use for model criticism
  [Sutherland et al., ICLR 2017]

[Dziugaite et al., 2015]

- Neural MNIST/faces samples (RBF kernel)

DISCO Nets: DISsimilarity COefficients Networks

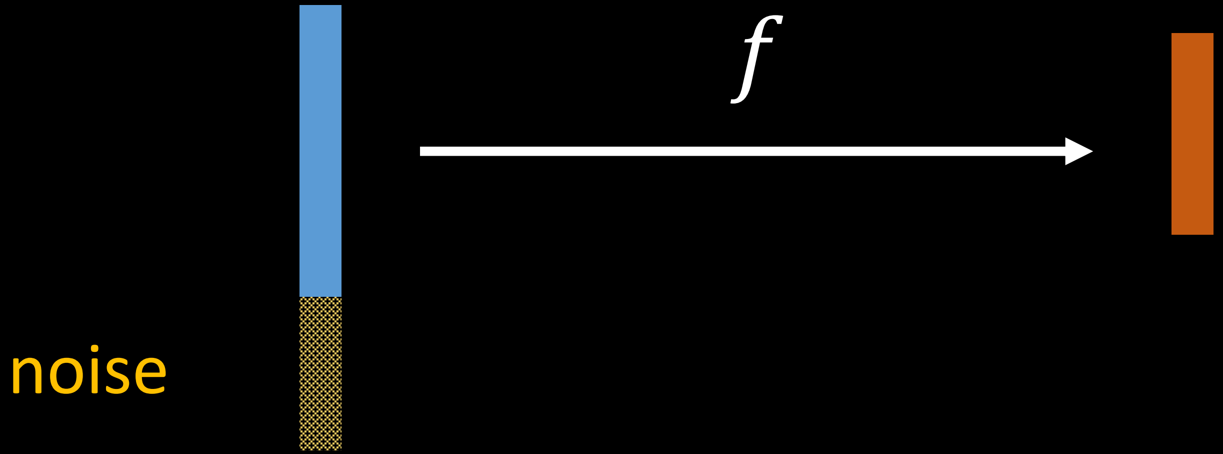
Diane Bouchacourt (Oxford)
Pawan Kumar (Oxford)
Sebastian Nowozin

NIPS 2016

arXiv:1606.02556

$$f\colon \mathcal{X} \to \mathcal{Y}$$

$$f\colon \mathcal{X} \to \mathcal{P}(\mathcal{Y})$$



$$\min_f \; \mathbb{E}_{(x,y)} \mathbb{E}_{\varepsilon} \left[\ell(y, f(x,\varepsilon)) - \frac{1}{2} \mathbb{E}_{\varepsilon',\varepsilon''} [\ell(f(x,\varepsilon'), f(x,\varepsilon''))] \right]$$

Minimize Expected loss minus Diversity

Constructing Divergence from Loss

- Loss $\Delta(y, y')$
- True joint distribution $\mathrm{T}(x, y)$
- Model distribution $P(y|x)$
- Expected loss (diversity coefficient)

$$\mathrm{DIV}(Q, P) = \mathbb{E}_{x \sim T(x)} \left[\mathbb{E}_{y \sim Q(y|x)} \left[\mathbb{E}_{y' \sim P(y|x)} [\Delta(y, y')] \right] \right]$$

- Dissimilarity coefficient [Rao, 1982]

$$\mathrm{DISC}(Q, P) = \mathrm{DIV}(Q, P) - \gamma \mathrm{DIV}(P, P)$$

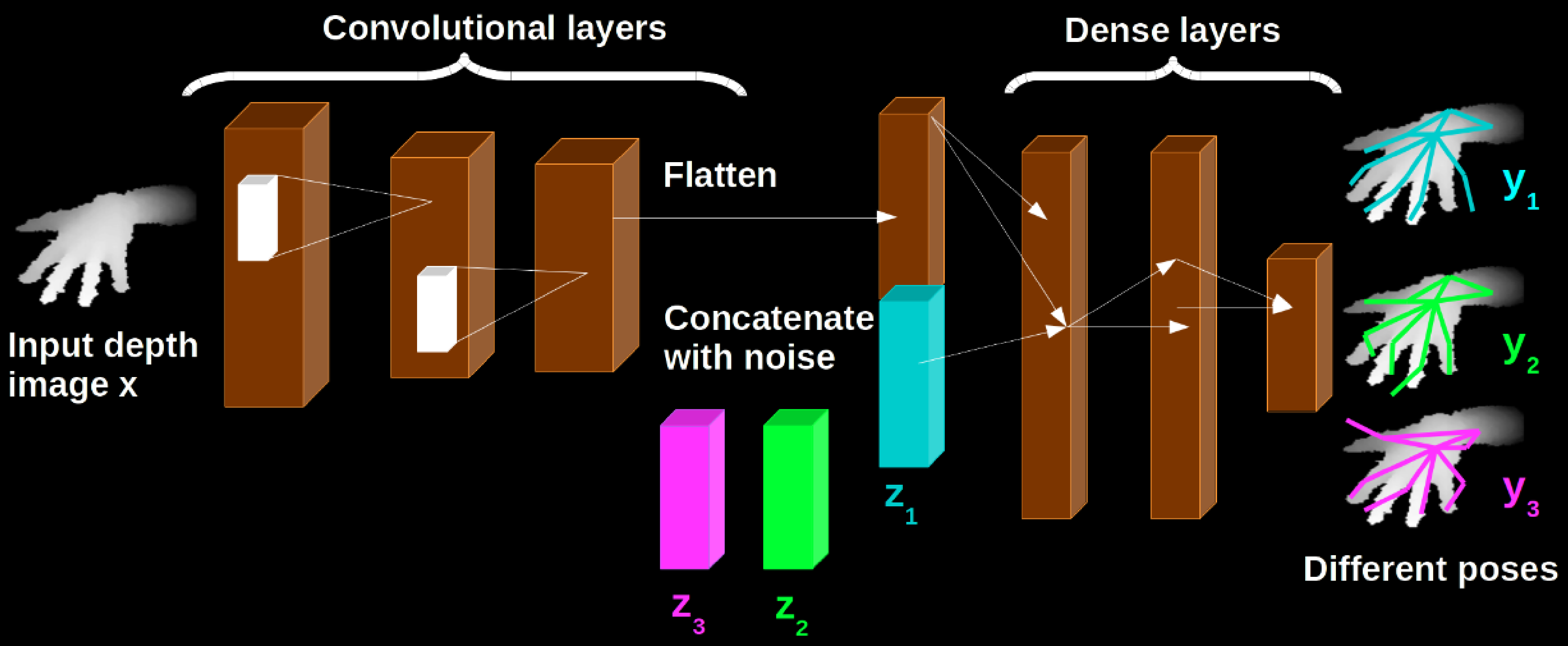
- $\gamma = \frac{1}{2}$

Relation to Scoring rules and Kernel MMD

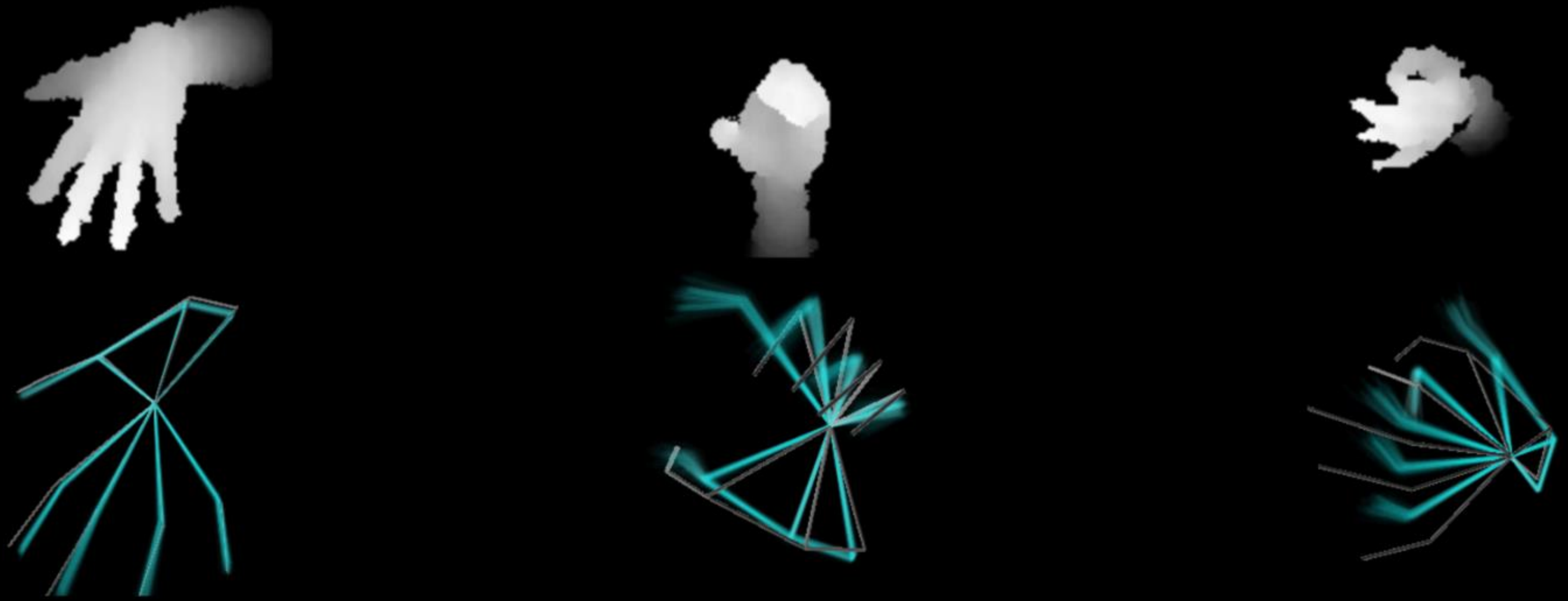
- Via [Gneiting and Raftery, 2007]:
  For $\Delta_\beta(y, y') = \|y - y'\|_2^\beta$, with $\beta \in (0,2)$ DISCO is a proper scoring rule.
- Via [Schölkopf, 2001]:
  For $k(y, y') = \|y - y'\|_2^\beta$, with $\beta \in (0,2)$, k is conditionally positive definite and DISCO is the kernel MMD objective with $k = \Delta$.

DISCO Nets

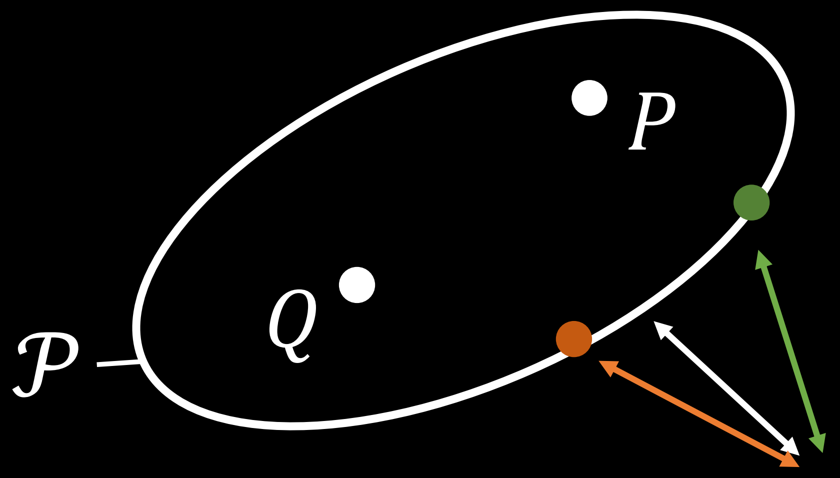
with Diane Bouchacourt, Pawan Kumar, NIPS 2016, arXiv:1606.02556



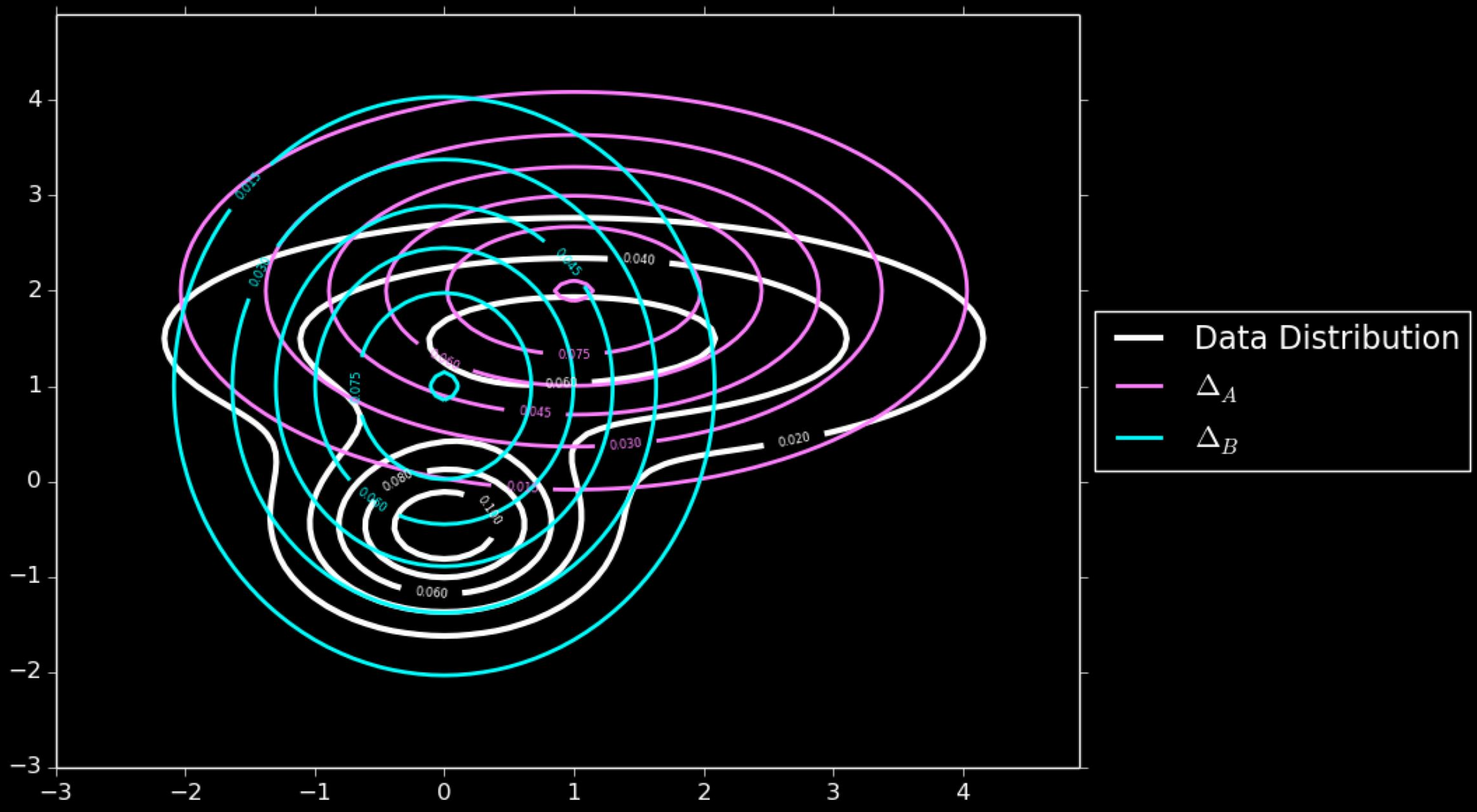
$$\mathrm{DISC}(Q,P) = \mathrm{DIV}(Q,P) - \gamma \mathrm{DIV}(P,P)$$



Bayesian Decision Theory



- “The well-calibrated Bayesian” [Dawid, 1982]
- “Loss-calibrated Bayesian” [Lacoste-Julien et al., 2011]
- [Pletscher, Nowozin, Rother, Kohli, 2011]
- [Fushiki, 2005]



Data Distribution
Δ_A
Δ_B
0.040
0.060
0.020
0.080
0.100
0.060
0.075
0.060
0.045
0.030
0.015
0.015
0.030
0.045
0.075
0.060
−3
−2
−1
0
1
2
3
4

Wasserstein Distance

$$\gamma_{\mathcal{F}}(P,Q) = \sup_{f\in\mathcal{F}} \left| \int f \mathrm{d}P - \int f \mathrm{d}Q \right|$$

- Wasserstein GAN [Arjovsky et al., 2017]
- $\mathcal{F} = \{f : \|f\|_L \leq 1\}$, with separable metric space (M, ρ)

$$\|f\|_L = \sup\{\frac{|f(x) - f(y)|}{\rho(x,y)} : x \neq y \text{ in } M\}$$

- [Sriperumbudur et al., JMLR 2010]

Wasserstein GAN, [Arjovsky et al., 2017]

- Kantorovich-Rubinstein duality

$$W(P,Q) = \max_{\|f\|_L \leq 1} \left(\mathbb{E}_{x\sim P}[f(x)] - \mathbb{E}_{x\sim Q}[f(x)] \right)$$

- How to set up rich function class uniformly respecting $\|f\|_L \leq 1$?
    - [Arjovsky et al., 2017]: weight clipping
      ("Weight clipping is a clearly terrible way to enforce a Lipschitz constraint")
    - [Gulrajani et al., 2017]: regularize gradient norm
      (DL frameworks such as TensorFlow easily support this.)

Outline

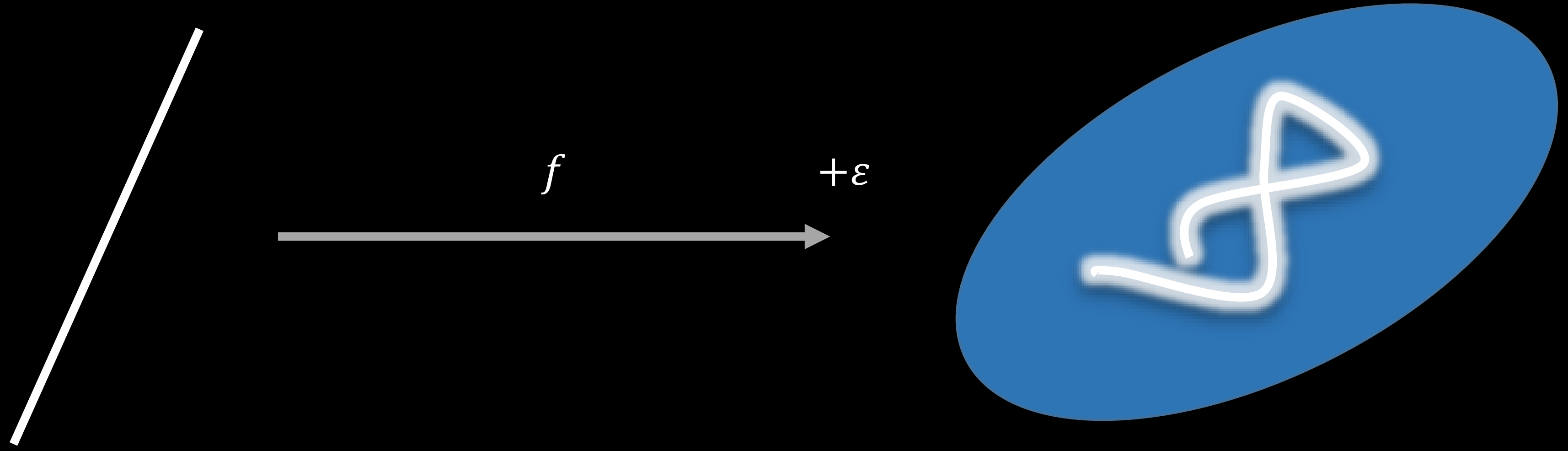
1. f-Divergences (GAN)
2. Proper Scoring Rules (VAE)
3. Integral Probability Metrics (DISCO, MMD, WGAN)
4. Current research areas

Current Research Areas

GANs as building blocks

- For inference (as in AVB), or
- As model component or regularizer

Dimensionality / Stability (IPM/GAN)



Adding Noise [Sønderby et al., 2016], [Arjovsky and Bottou, 2016]

Structuring the Latent Space

- Adding semantics through supervision
  [Louizos et al., ICLR 2016]
- Control of information/representation stored in latent factors
  [Chen et al., ICLR 2017], [Alemi et al., ICLR 2017], [Chalk et al., 2016], [Bouchacourt et al., 2017]

Interpolating in latent space


DATA
DATA SAMPLES
CONTENT
STYLE
REC.
REC. SAMPLES

Bayesian Deep Learning

- Bayesian neural networks make rapid progress
  - Stochastic Gradient Langevin Dynamics (SGLD) based algorithms
    [Li et al., AAAI 2016], [Springenberg et al., NIPS 2016], [Gan et al., ACL 2017], [Ahn et al., ICML 2012], [Welling and Teh, ICML 2011]
  - Stochastic Variational Bayes
    [Kingma et al, NIPS 2015], [Blundell et al., ICML 2015], [Hinton and Van Camp, 1993]
  - SGD as Variational Inference
    [Mandt et al., 2016], [Duvenaud et al., AISTATS 2016]
  - Dropout as Variational Inference
    [Gal and Ghahramani, 2015]
- The most powerful probabilistic deep learning models have no practical Bayesian version (yet)
- Reason 1: Likelihood not accessible
- Reason 2: Stability and variance issues
- Reason 3: Posterior/model size

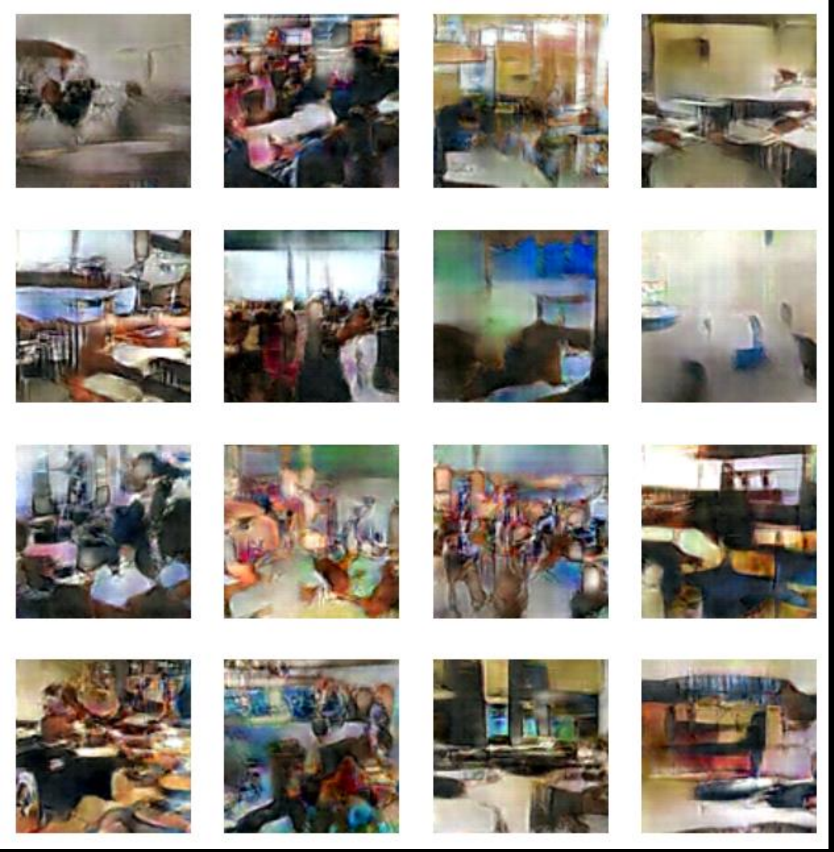
Stabilizing GAN Training

Thanks!

Additional Slides

LSUN Natural Images

- [Yu et al., 2015] one of the largest databases of natural images
- 168k images of classrooms
- [Radford et al., 2015] architecture
  - Generator: deconvolutional network, ~3M parameters
  - Variational function: convnet, ~3M parameters
- Batch normalization, gradient clipping, Adam
- ~24 hours training time (Titan X), ~200 images/s



GAN (Jensen-Shannon)

Hellinger

Kullback-Leibler

- Explanation for lack of differences in [Poole et al., arXiv:1612.02780]

Conclusion

- Generative model revival
- Powered by deep neural networks
- Key properties:
  - Training by backprop
  - Efficient at test time

Probabilistic Modeling

- Model of uncertainty is important in many applications
- Generative or discriminative
- Typical operations on model $\mathrm{P} \in \mathcal{P}$
  - *Sampling*: $x \sim P$
  - *Estimation*: given iid samples $\{x_1, \cdots, x_n\}$, find good $\mathrm{P} \in \mathcal{P}$
  - *Likelihood* evaluation: given x, evaluate likelihood $P(x)$
  - *Marginalization* and *conditioning*

Bayesian Decision Theory

- [Savage 1954]: every rational behaviour can be factorized into maintaining coherent beliefs and making optimal decisions under beliefs.
- *Likelihood-principle*:
  the only way to maintain coherent beliefs is due to Bayes rule
- Foundation of the *subjective Bayesian* school:
  choice of prior and utility

- Conditioned on assumed model

MNIST Setup

- Model of [Goodfellow et al., NIPS 2014]
- Generator, ~2.5M parameters

$$z \quad \rightarrow \mathrm{Linear}(100, 1200) \rightarrow \mathrm{BN} \rightarrow \mathrm{ReLU} \rightarrow \mathrm{Linear}(1200, 1200) \rightarrow \mathrm{BN} \rightarrow \mathrm{ReLU}$$
$$\rightarrow \mathrm{Linear}(1200, 784) \rightarrow \mathrm{Sigmoid}$$

- Variational function, ~250k parameters

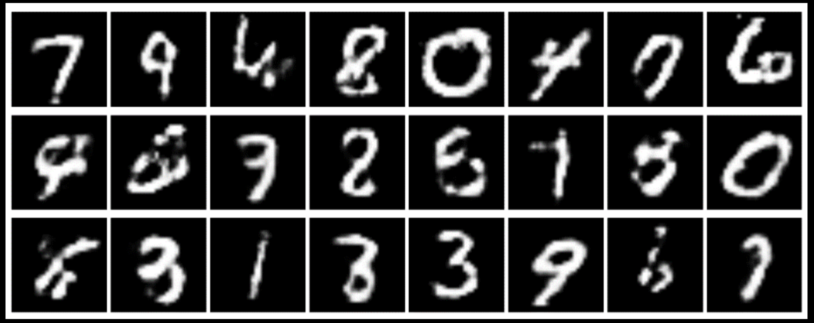
$$x \quad \rightarrow \mathrm{Linear}(784{,}240) \rightarrow \mathrm{ELU} \rightarrow \mathrm{Linear}(240{,}240) \rightarrow \mathrm{ELU} \rightarrow \mathrm{Linear}(240{,}1)$$

- Evaluation using KDE log-likelihoods
  - Known shortcomings, but popular in other works

MNIST Results

Training divergence	KDE $\langle LL \rangle$ (nats)	$\pm$ SEM
Kullback-Leibler	416	5.62
Reverse Kullback-Leibler	319	8.36
Pearson χ^2	429	5.53
Neyman χ^2	300	8.33
Squared Hellinger	-708	18.1
Jeffrey	-2101	29.9
Jensen-Shannon	367	8.19
GAN	305	8.97
Variational Autoencoder [18]	445	5.36
KDE MNIST train (60k)	502	5.99

Kullback-Leibler



Reverse Kullback-Leibler

Hellinger

NYU Hand dataset

- [Tompson et al., 2014], depth and hand pose annotations
- 72,757 training images, 8,252 testing images
- 14 joints
- Setup and architecture from [Oberweger et al., 2015]
- Minimum expected loss decisions
- Different loss functions

Quantitative results (NYU test)

Model	ProbLoss (mm)	MeJEE (mm)	MaJEE (mm)	FF (80mm)
$\text{BASE}_{\beta=1,\sigma=1}$	103.8±0.627	25.2±0.152	52.7±0.290	86.040
$\text{BASE}_{\beta=1,\sigma=5}$	99.3±0.620	25.5±0.151	52.9±0.289	85.773
$\text{BASE}_{\beta=1,\sigma=10}$	96.3±0.612	25.7±0.149	53.2±0.288	85.664
$\text{DISCO}_{\beta=1,\gamma=0.5}$	**83.8 ±0.503**	**20.9±0.124**	**45.1±0.246**	**94.438**

Model	ProbLoss (mm)	MeJEE (mm)	MaJEE (mm)	FF (80mm)
cGAN	442.7±0.513	109.8±0.128	201.4±0.320	0.000
$\text{cGAN}_{\text{init, fixed}}$	128.9±0.480	31.8±0.117	64.3±0.230	78.454
$\text{DISCO}_{\beta=1,\gamma=0.5}$	**83.8 ±0.503**	**20.9±0.124**	**45.1±0.246**	**94.438**

NIPS 2016 Paper Contributions

- Generalizes GAN objective to arbitrary f-divergences
- Simplifies the GAN algorithm
- Local convergence proof

Experiments

Synthetic 1D Univariate

Approximate a mixture of Gaussians by a Gaussian to

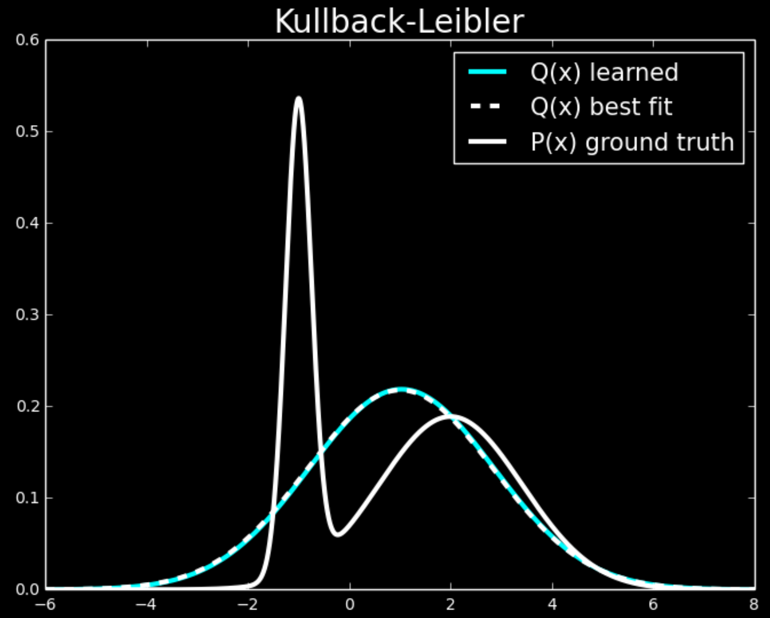
- Validate the approach
- Demonstrate the properties of different divergences [Minka, 2005]

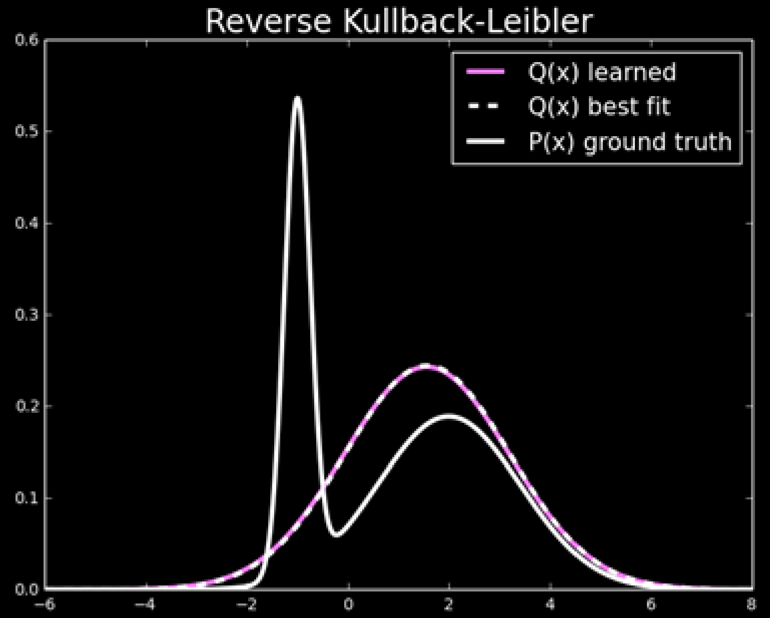
We compare the exact optimisation of the divergence with the GAN approach

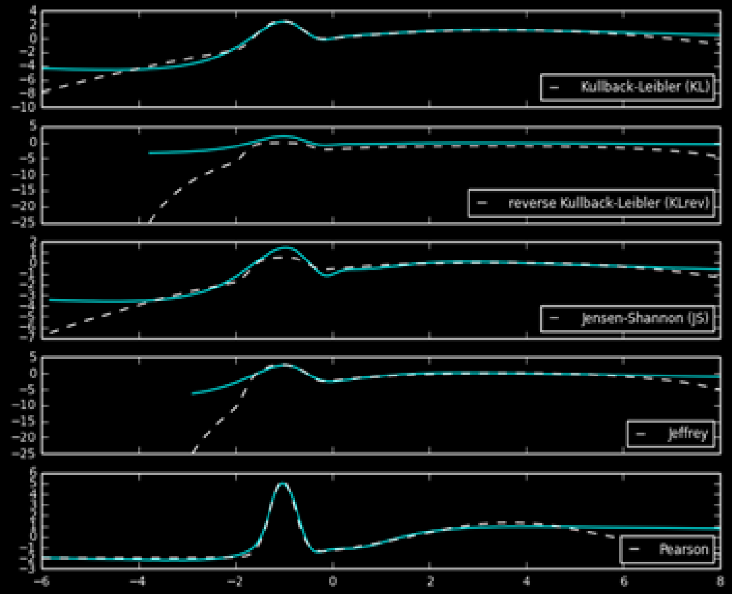
Setup

- Data: $P(x)$ is a mixture of Gaussians (any number of samples, not just a data set)
- Generator: $Q(x)$ is the distribution of $\mu + \sigma z$ where $z \sim N(0,1)$ (Gaussian)
- Discriminator: $T(x)$ is a two-layer NN with $tanh$ units

Synthetic 1D Univariate







	KL	KL-rev	JS	Jeffrey	Pearson
$D_f(P\|\|Q_{\theta^*})$	0.2831	0.2480	0.1280	0.5705	0.6457
$F(\hat{\omega}, \hat{\theta})$	0.2801	0.2415	0.1226	0.5151	0.6379
μ^*	1.0100	1.5782	1.3070	1.3218	0.5737
$\hat{\mu}$	1.0335	1.5624	1.2854	1.2295	0.6157
σ^*	1.8308	1.6319	1.7542	1.7034	1.9274
$\hat{\sigma}$	1.8236	1.6403	1.7659	1.8087	1.9031

f-GAN Future Work

- Applications to discriminative models
- Applications to Reinforcement Learning
  - Model-based RL, modelling $P(s_{t+1}, r_t | s_t, a_t)$
  - Policy-gradient methods, modelling $P(a_t | s_t)$
  - Promising method to handle large state and action spaces
- Applications to Variational Bayes: variational family of distributions
- Extension to discrete outputs (structured prediction)
  - Text
- Encoder/Decoder bidirectional models (e.g. BiCGAN)
- Factorized latent space (e.g. style/content separation), (e.g. InfoGAN)

Conclusions (DISCO)

- Learning probabilistic models under misspecification
- Starting point: task-specific loss function
- Theory from: proper scoring rules, kernel MMD
- Good empirical results on challenging application

Learning Probabilistic Models

Integral Probability Metrics
[Müller, 1997]
[Sriperumbudur et al., 2010]

$$\gamma_{\mathcal{F}}(P,Q) = \sup_{f\in\mathcal{F}} \left| \int f \mathrm{d}P - \int f \mathrm{d}Q \right|$$

- P: Expectation
- Q: Expectation
- Structure in $\mathcal{F}$
- Examples:
  - Energy statistic [Szekely, 1997]
  - Kernel MMD [Gretton et al., 2012], [Smola et al., 2007]
  - Wasserstein distance [Cuturi, 2013]
  - DISCO Nets [Bouchacourt et al., 2016]

Learning Probabilistic Models

[Nguyen et al., 2010], [Reid and Williamson, 2011], [Goodfellow et al., 2014]
Variational representation of divergences

- *P*: Expectation
- *Q*: Expectation

- *P*: Distribution
- *Q*: Expectation

- *P*: Distribution
- *Q*: Distribution

f-divergences

- Divergence between two distributions

$$D_f(P \parallel Q) = \int_{\mathcal{X}} q(x)\, f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x$$

- $f\colon\ \mathbb{R}_+ \to \mathbb{R}$ convex, lower-semicontinuous
- $f(1) = 0$.

Estimating f-divergences from samples

- [Nguyen, Wainwright, Jordan, Information Theory, 2010]
- Every convex function f has a convex *Fenchel conjugate* f^* so that

$$f(u) = \sup_{t \in \mathrm{dom}_{f^*}} \{tu - f^*(t)\}$$

Estimating f-divergences from samples (cont)

$$
\begin{aligned}
D_f(P \parallel Q) &= \int_{\mathcal{X}} q(x)\, f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x \\
&= \int_{\mathcal{X}} q(x) \sup_{t_x \in \mathrm{dom}_{f^*}} \left\{ t_x \frac{p(x)}{q(x)} - f^*(t_x) \right\} \mathrm{d}x \\
&\geq \sup_{T \in \mathcal{T}} \left(\int_{\mathcal{X}} p(x)\, T(x)\, \mathrm{d}x - \int_{\mathcal{X}} q(x)\, f^*\big(T(x)\big)\, \mathrm{d}x \right) \\
&= \sup_{T \in \mathcal{T}} \big(\mathbb{E}_{x \sim P}[T(x)] - \mathbb{E}_{x \sim Q}[f^*(T(x))] \big)
\end{aligned}
$$

VAE: Maximum Likelihood Training

- Maximize the data log-likelihood, per-instance variational approximation

$$
\begin{aligned}
\log p(x|\theta) &= \log \int \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) p(z) \mathrm{d}z \\
&= \log \int \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) \frac{q(z)}{q(z)} p(z)\, \mathrm{d}z \\
&= \log \int \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) \frac{p(z)}{q(z)} q(z)\, \mathrm{d}z \\
&= \log \mathbb{E}_{z\sim q(z)} \left[\mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) \frac{p(z)}{q(z)} \right] \\
&\geq \mathbb{E}_{z\sim q(z)} \left[\log \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) \frac{p(z)}{q(z)} \right] \\
&= \mathbb{E}_{z\sim q(z)} [\log \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta)] - D_{\mathrm{KL}}(q(z) \parallel p(z))
\end{aligned}
$$

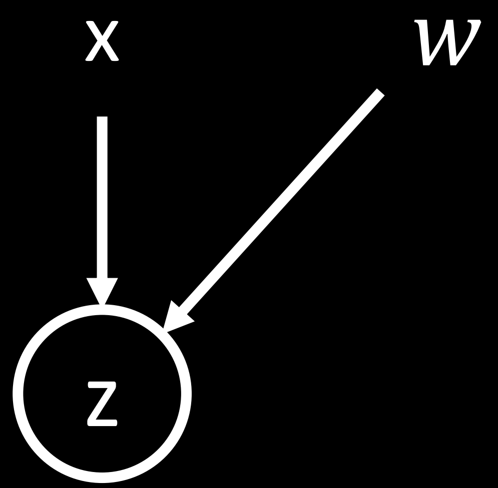
Inference networks

- Amortized inference [Stuhlmüller et al., NIPS 2013]
- Inference networks
- “Informed sampler” [Jampani et al., 2014]
- “Memory-based approach” [Kulkarni et al., 2015]

Inference networks

x_1 x_2 x_3

$q_1(z)$ $q_2(z)$ $q_3(z)$



VAE: Maximum Likelihood Training

- Maximize the data log-likelihood, inference network variational approximation

$$\log p(x|\theta) = \log \int \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) p(z) \mathrm{dz}$$

$$= \log \int \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) \frac{q(z|x,w)}{q(z|x,w)} p(z)\, \mathrm{dz}$$

$$= \log \int \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) \frac{p(z)}{q(z|x,w)} q(z|x,w)\, \mathrm{dz}$$

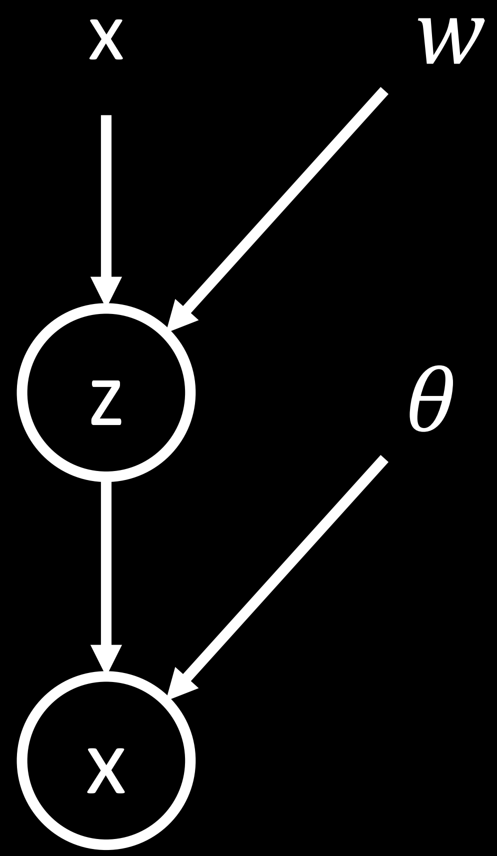
$$= \log \mathbb{E}_{z \sim q(z|x,w)} \left[\mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) \frac{p(z)}{q(z|x,w)} \right]$$

$$\geq \mathbb{E}_{z \sim q(z|x,w)} \left[\log \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta) \frac{p(z)}{q(z|x,w)} \right]$$

$$= \mathbb{E}_{z \sim q(z|x,w)} [\log \mathrm{p}(\mathrm{x}|\mathrm{z}, \theta)] - D_{\mathrm{KL}}(q(z|x,w) \parallel p(z))$$

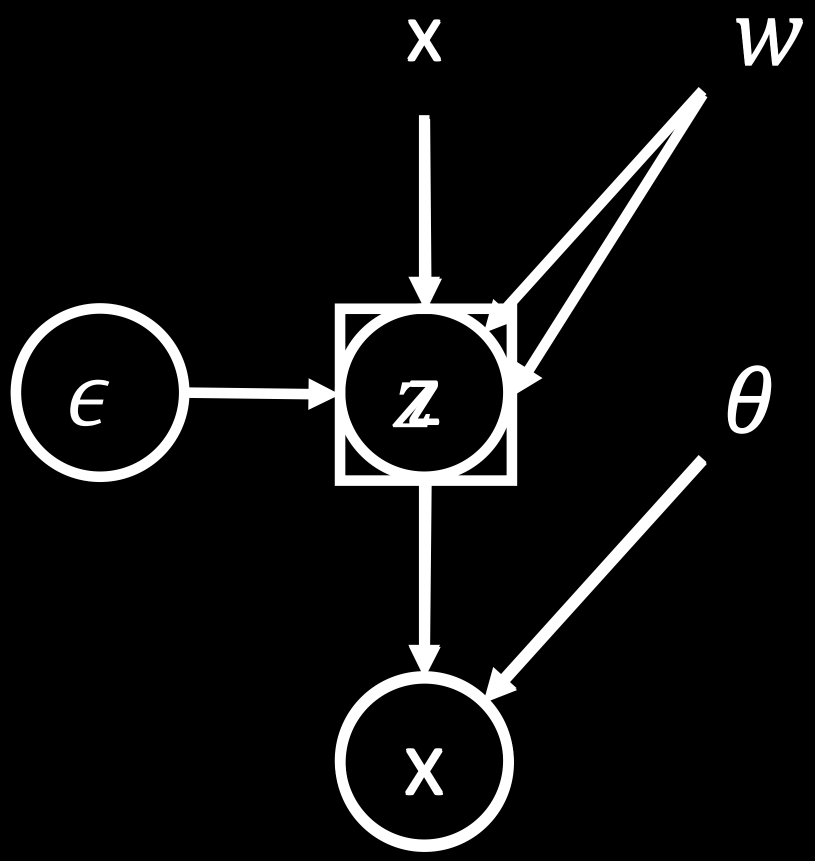
Reparametrization Trick

- [Rezende et al., 2014]
  [Kingma and Welling, 2014]



Reparametrization Trick

- [Rezende et al., 2014]
  [Kingma and Welling, 2014]
- Stochastic computation graphs
  [Schulman et al., 2015]



Derivatives

$$\nabla_w \mathbb{E}_{z\sim q(z|x,w)}[\log \mathrm{p}(\mathrm{x}|\mathrm{z},\theta) - T^*(x,z)\ ]$$

$$T^* = \underset{T\epsilon\mathcal{T}}{\operatorname{argmax}}\, \mathbb{E}_{x\sim p_D}\left[\mathbb{E}_{z\sim q(z|x,w)}[\log \sigma(T(x,z))] + \mathbb{E}_{z\sim p(z)}[\log(1-\sigma(T(x,z)))]\right]$$

Proposition:

For any $q(z|x,w)$ we have

$$\mathbb{E}_{z\sim q(z|x,w)}[\nabla_w T^*(x,z)] = 0.$$