

# Deep Generative Models

## Lecture 8

Roman Isachenko



AI Masters

Autumn, 2022

# Recap of previous lecture

Let our data  $\mathbf{y}$  comes from discrete distribution  $\Pi(\mathbf{y})$ .

## Discrete model

- ▶ Use **discrete** model (e.x.  $P(\mathbf{y}|\boldsymbol{\theta}) = \text{Cat}(\boldsymbol{\pi}(\boldsymbol{\theta}))$ ).
- ▶ Minimize any suitable divergence measure  $D(\Pi, P)$ .

## Continuous model

Use **continuous** model (e.x.  $p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}), \boldsymbol{\sigma}_{\boldsymbol{\theta}}^2(\mathbf{x}))$ ), but

- ▶ **discretize** model (make the model outputs discrete): transform  $p(\mathbf{x}|\boldsymbol{\theta})$  to  $P(\mathbf{y}|\boldsymbol{\theta})$ ;
- ▶ **dequantize** data (make the data continuous): transform  $\Pi(\mathbf{y})$  to  $\pi(\mathbf{x})$ .

## Model discretization through CDF

$$F(\mathbf{x}|\boldsymbol{\theta}) = \int_{-\infty}^{\mathbf{x}} p(\mathbf{x}'|\boldsymbol{\theta}) d\mathbf{x}'; \quad P(\mathbf{y}|\boldsymbol{\theta}) = F(\mathbf{y} + 0.5|\boldsymbol{\theta}) - F(\mathbf{y} - 0.5|\boldsymbol{\theta})$$

# Recap of previous lecture

## Uniform dequantization bound

Let dequantize discrete distribution  $\Pi(\mathbf{y})$  to continuous distribution  $\pi(\mathbf{x})$  in the following way:  $\mathbf{x} = \mathbf{y} + \mathbf{u}$ , where  $\mathbf{u} \sim U[0, 1]$ .

## Theorem

Fitting continuous model  $p(\mathbf{x}|\theta)$  on uniformly dequantized data is equivalent to maximization of a lower bound on log-likelihood for a discrete model:

$$P(\mathbf{y}|\theta) = \int_{U[0,1]} p(\mathbf{y} + \mathbf{u}|\theta) d\mathbf{u}$$

## Variational dequantization bound

Introduce variational dequantization noise distribution  $q(\mathbf{u}|\mathbf{x})$  and treat it as an approximate posterior.

$$\log P(\mathbf{x}|\theta) \geq \int q(\mathbf{u}|\mathbf{x}) \log \frac{p(\mathbf{x} + \mathbf{u}|\theta)}{q(\mathbf{u}|\mathbf{x})} d\mathbf{u} = \mathcal{L}(q, \theta).$$

# Recap of previous lecture

## Theorem

$$\frac{1}{n} \sum_{i=1}^n KL(q(\mathbf{z}|\mathbf{x}_i)||p(\mathbf{z})) = KL(q_{\text{agg}}(\mathbf{z})||p(\mathbf{z})) + \mathbb{I}_q[\mathbf{x}, \mathbf{z}].$$

## ELBO surgery

$$\frac{1}{n} \sum_{i=1}^n \mathcal{L}_i(q, \theta) = \underbrace{\frac{1}{n} \sum_{i=1}^n \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)} \log p(\mathbf{x}_i|\mathbf{z}, \theta)}_{\text{Reconstruction loss}} - \underbrace{\mathbb{I}_q[\mathbf{x}, \mathbf{z}]}_{\text{MI}} - \underbrace{KL(q_{\text{agg}}(\mathbf{z})||p(\mathbf{z}))}_{\text{Marginal KL}}$$

## Optimal prior

$$KL(q_{\text{agg}}(\mathbf{z})||p(\mathbf{z})) = 0 \quad \Leftrightarrow \quad p(\mathbf{z}) = q_{\text{agg}}(\mathbf{z}) = \frac{1}{n} \sum_{i=1}^n q(\mathbf{z}|\mathbf{x}_i).$$

The optimal prior distribution  $p(\mathbf{z})$  is aggregated posterior  $q(\mathbf{z})$ .

## Recap of previous lecture

- ▶ Standard Gaussian  $p(\mathbf{z}) = \mathcal{N}(0, I) \Rightarrow$  over-regularization;
- ▶  $p(\mathbf{z}) = q_{\text{agg}}(\mathbf{z}) = \frac{1}{n} \sum_{i=1}^n q(\mathbf{z}|\mathbf{x}_i) \Rightarrow$  overfitting and highly expensive.

## ELBO revisiting

$$\frac{1}{n} \sum_{i=1}^n \mathcal{L}_i(q, \theta) = \text{RL} - \text{MI} - \text{KL}(q_{\text{agg}}(\mathbf{z}) || p(\mathbf{z}|\lambda))$$

It is Forward KL with respect to  $p(\mathbf{z}|\lambda)$ .

## ELBO with flow-based VAE prior

$$\begin{aligned} \mathcal{L}(\phi, \theta) &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} [\log p(\mathbf{x}|\mathbf{z}, \theta) + \log p(\mathbf{z}|\lambda) - \log q(\mathbf{z}|\mathbf{x}, \phi)] \\ &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \phi)} \left[ \log p(\mathbf{x}|\mathbf{z}, \theta) + \underbrace{\left( \log p(f(\mathbf{z}, \lambda)) + \log |\det(\mathbf{J}_f)| \right)}_{\text{flow-based prior}} - \log q(\mathbf{z}|\mathbf{x}, \phi) \right] \end{aligned}$$

- ▶ RealNVP with coupling layers.
- ▶ Autoregressive flow (fast  $f(\mathbf{z}, \lambda)$ , slow  $g(\mathbf{z}^*, \lambda)$ ).

# Recall of previous lecture

## ELBO decomposition

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + KL(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})).$$

- ▶ E-step of EM-algorithm:  $KL(q(\mathbf{z}|\mathbf{x}, \phi) || p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})) = 0$ .  
(In this case the lower bound is tight  $\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta})$ ).
- ▶  $q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}_\phi^2(\mathbf{x}))$  is a unimodal distribution (not expressive enough).
- ▶ NF convert a simple distribution to a complex one. Let use NF in VAE posterior.

Apply a sequence of transformations to the random variable

$$\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}, \phi) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}_\phi^2(\mathbf{x})).$$

Let  $q(\mathbf{z}|\mathbf{x}, \phi)$  (VAE encoder) be a base distribution for a flow model.

# Outline

1. Flow-based VAE posterior vs flow-based VAE prior
2. Disentanglement learning
3. Likelihood-free learning
4. Generative adversarial networks

# Outline

1. Flow-based VAE posterior vs flow-based VAE prior
2. Disentanglement learning
3. Likelihood-free learning
4. Generative adversarial networks



## Flows in VAE posterior

- ▶ Encoder outputs base distribution  $q(\mathbf{z}|\mathbf{x}, \phi)$ .
- ▶ Flow model  $\mathbf{z}^* = f(\mathbf{z}, \lambda)$  transforms the base distribution  $q(\mathbf{z}|\mathbf{x}, \phi)$  to the distribution  $q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)$ .
- ▶ Distribution  $q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)$  is used as a variational distribution for ELBO maximization.
- ▶ Here  $\phi$  – encoder parameters,  $\lambda$  – flow parameters.

### Flow model in latent space

$$\log q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda) = \log q(\mathbf{z}|\mathbf{x}, \phi) + \log \left| \det \left( \frac{d\mathbf{z}}{d\mathbf{z}^*} \right) \right|$$

$$\mathbf{z}^* = f(\mathbf{z}, \lambda) = g^{-1}(\mathbf{z}^*, \lambda)$$

### ELBO with flow-based VAE posterior

$$\mathcal{L}(\phi, \theta, \lambda) = \mathbb{E}_{q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)} \log p(\mathbf{x}|\mathbf{z}^*, \theta) - KL(q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda) || p(\mathbf{z}^*)).$$

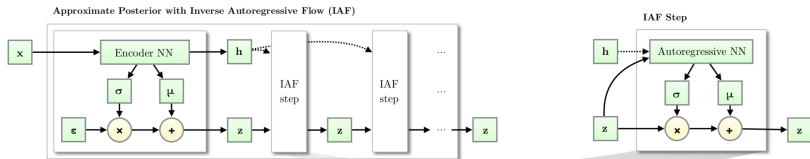
The second term in ELBO is reverse KL divergence with respect to  $q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)$ .

# Flow-based VAE posterior

## ELBO objective

$$\begin{aligned}\mathcal{L}(\phi, \theta, \lambda) &= \mathbb{E}_{q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)} [\log p(\mathbf{x}|\mathbf{z}^*, \theta) + \log p(\mathbf{z}^*) - \log q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)] = \\ &= \mathbb{E}_{q(\mathbf{z}^*|\mathbf{x}, \phi, \lambda)} \left[ \log p(\mathbf{x}|\mathbf{z}^*, \theta) + \log p(\mathbf{z}^*) - \right. \\ &\quad \left. - \left( \log q(g(\mathbf{z}^*, \lambda)|\mathbf{x}, \phi) + \log |\det(\mathbf{J}_g)| \right) \right].\end{aligned}$$

- ▶ RealNVP with coupling layers.
- ▶ Inverse autoregressive flow (slow  $f(\mathbf{z}, \lambda)$ , fast  $g(\mathbf{z}^*, \lambda)$ ).
- ▶ Is it OK to use AF for VAE posterior?



# Flows-based VAE prior vs posterior

## Theorem

VAE with the flow-based prior  $p(\mathbf{z}|\boldsymbol{\lambda})$  for latent code  $\mathbf{z}^*$  is equivalent to VAE with flow-based posterior  $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}, \boldsymbol{\lambda})$  for latent code  $\mathbf{z}$ .

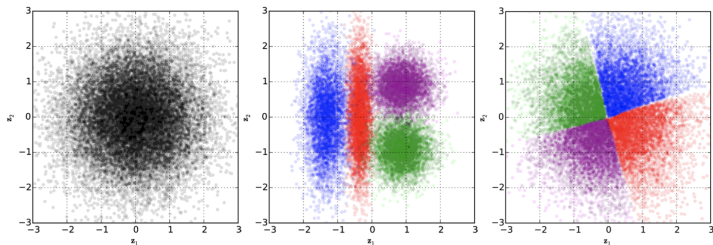
## Proof

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\lambda}) &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})} \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - \underbrace{KL(q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})||p(\mathbf{z}|\boldsymbol{\lambda}))}_{\text{flow-based prior}} \\ &= \mathbb{E}_{q(\mathbf{z}^*|\mathbf{x}, \boldsymbol{\phi}, \boldsymbol{\lambda})} \log p(\mathbf{x}|f(\mathbf{z}^*, \boldsymbol{\lambda}), \boldsymbol{\theta}) - \underbrace{KL(q(\mathbf{z}^*|\mathbf{x}, \boldsymbol{\phi}, \boldsymbol{\lambda})||p(\mathbf{z}^*))}_{\text{flow-based posterior}}\end{aligned}$$

(Here we use Flow KL duality theorem from Lecture 5 and LOTUS)

- ▶ IAF posterior decoder path:  $\mathbf{z} \sim p(\mathbf{z})$ ,  $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$ .
- ▶ AF prior decoder path:  $\mathbf{z}^* \sim p(\mathbf{z}^*)$ ,  $\mathbf{z} = f(\mathbf{z}^*, \boldsymbol{\lambda})$ ,  $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$ .

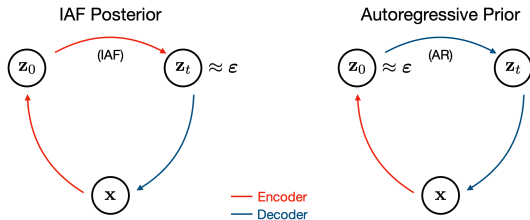
# Flows-based VAE prior vs posterior



(a) Prior distribution

(b) Posteriors in standard VAE

(c) Posteriors in VAE with IAF



Kingma D. P. et al. *Improving Variational Inference with Inverse Autoregressive Flow*, 2016

image credit: <https://courses.cs.washington.edu/courses/cse599i/20au>

# VAE limitations

- ▶ Poor generative distribution (decoder)

$$p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}), \boldsymbol{\sigma}_{\boldsymbol{\theta}}^2(\mathbf{z})) \quad \text{or} \quad = \text{Softmax}(\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{z})).$$

- ▶ Loose lower bound

$$\log p(\mathbf{x}|\boldsymbol{\theta}) - \mathcal{L}(q, \boldsymbol{\theta}) = (?).$$

- ▶ Poor prior distribution

$$p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I}).$$

- ▶ Poor variational posterior distribution (encoder)

$$q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \boldsymbol{\sigma}_{\boldsymbol{\phi}}^2(\mathbf{x})).$$

# Outline

1. Flow-based VAE posterior vs flow-based VAE prior
2. Disentanglement learning
3. Likelihood-free learning
4. Generative adversarial networks

# Disentangled representations

**Representation learning** is looking for an interpretable representation of the independent data generative factors.

## Disentanglement informal definition

Every single latent unit are sensitive to changes in a single generative factor, while being invariant to changes in other factors.

## ELBO objective

$$\mathcal{L}(q, \theta, \beta) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}, \theta) - \beta \cdot KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})).$$

What do we get at  $\beta = 1$ ?

## Constrained optimization

$$\max_{q, \theta} \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}, \theta), \quad \text{subject to } KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) < \epsilon.$$

**Note:** It leads to poorer reconstructions and a loss of high frequency details.

---

Higgins I. et al. *beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework*, 2017

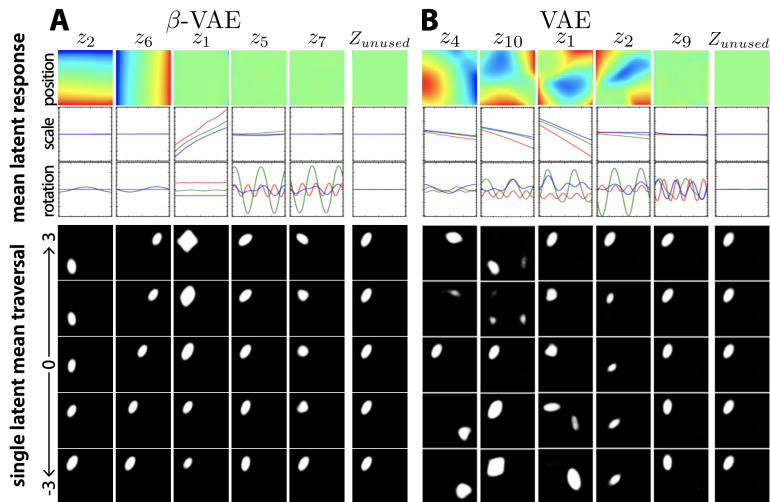
# $\beta$ -VAE samples



Higgins I. et al. *beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework*, 2017



# $\beta$ -VAE analysis



Higgins I. et al. *beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework*, 2017

# $\beta$ -VAE

## ELBO

$$\mathcal{L}(q, \theta, \beta) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}, \theta) - \beta \cdot KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})).$$

## ELBO surgery

$$\frac{1}{n} \sum_{i=1}^n \mathcal{L}_i(q, \theta, \beta) = \underbrace{\frac{1}{n} \sum_{i=1}^n \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)} \log p(\mathbf{x}_i|\mathbf{z}, \theta)}_{\text{Reconstruction loss}} - \underbrace{\beta \cdot \mathbb{I}_q[\mathbf{x}, \mathbf{z}]}_{\text{MI}} - \underbrace{\beta \cdot KL(q_{\text{agg}}(\mathbf{z})||p(\mathbf{z}))}_{\text{Marginal KL}}$$

## Minimization of MI

- ▶ It is not necessary and not desirable for disentanglement.
- ▶ It hurts reconstruction.

# DIP-VAE: disentangled posterior

## Disentangled aggregated variational posterior

$$q_{\text{agg}}(\mathbf{z}) = \frac{1}{n} \sum_{i=1}^n q(\mathbf{z}|\mathbf{x}) = \prod_{j=1}^d q_{\text{agg}}(z_j)$$

## DIP-VAE objective

$$\begin{aligned} \mathcal{L}_{\text{DIP}}(q, \theta) &= \frac{1}{n} \sum_{i=1}^n \mathcal{L}_i(q, \theta) - \lambda \cdot KL(q_{\text{agg}}(\mathbf{z}) || p(\mathbf{z})) = \\ &= \underbrace{\frac{1}{n} \sum_{i=1}^n [\mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)} \log p(\mathbf{x}_i|\mathbf{z}, \theta)]}_{\text{Reconstruction loss}} - \underbrace{\mathbb{I}_q[\mathbf{x}, \mathbf{z}]}_{\text{MI}} - \underbrace{(1 + \lambda) \cdot KL(q_{\text{agg}}(\mathbf{z}) || p(\mathbf{z}))}_{\text{Marginal KL}} \end{aligned}$$

Marginal KL term is intractable.  $\Rightarrow$  Let match the moments of  $q_{\text{agg}}(\mathbf{z})$  and  $p(\mathbf{z})$ :

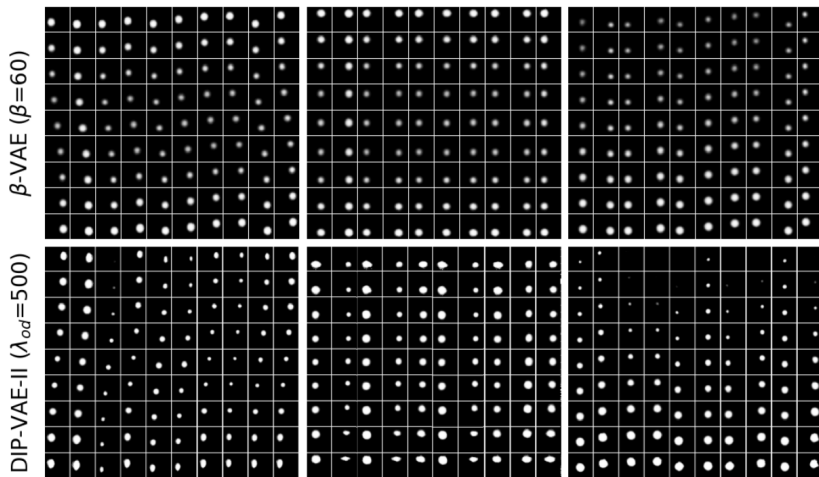
$$\text{cov}_{q_{\text{agg}}(\mathbf{z})}(\mathbf{z}) = \mathbb{E}_{q_{\text{agg}}(\mathbf{z})} \left[ (\mathbf{z} - \mathbb{E}_{q_{\text{agg}}(\mathbf{z})}(\mathbf{z}))(\mathbf{z} - \mathbb{E}_{q_{\text{agg}}(\mathbf{z})}(\mathbf{z}))^T \right].$$

---

Kumar A., Sattigeri P., Balakrishnan A. *Variational Inference of Disentangled Latent Concepts from Unlabeled Observations*, 2017

# DIP-VAE: analysis

Reconstructions become better.



---

Kumar A., Sattigeri P., Balakrishnan A. *Variational Inference of Disentangled Latent Concepts from Unlabeled Observations*, 2017

# Challenging disentanglement assumptions

## Theorem

Let  $\mathbf{z}$  has density  $p(\mathbf{z}) = \prod_{i=1}^d p(z_i)$ . Then, there exists an **infinite** family of bijective functions  $f : \text{supp}(\mathbf{z}) \rightarrow \text{supp}(\mathbf{z})$ :

- ▶  $\frac{\partial f_i(\mathbf{z})}{\partial z_j} \neq 0$  for all  $i$  and  $j$  ( $\mathbf{z}$  and  $f(\mathbf{z})$  are completely entangled);
- ▶  $P(\mathbf{z} \leq \mathbf{u}) = P(f(\mathbf{z}) \leq \mathbf{u})$  for all  $\mathbf{u} \in \text{supp}(\mathbf{z})$ .

Consider a generative model with disentangled representation  $\mathbf{z}$ .

- ▶  $\exists \hat{\mathbf{z}} = f(\mathbf{z})$  where  $\hat{\mathbf{z}}$  is completely entangled with respect to  $\mathbf{z}$ .
- ▶ The disentanglement method cannot distinguish between the two equivalent generative models:

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} = \int p(\mathbf{x}|\hat{\mathbf{z}})p(\hat{\mathbf{z}})d\hat{\mathbf{z}}.$$

Theorem claims that unsupervised disentanglement learning is impossible for arbitrary generative models with a factorized prior.

# Outline

1. Flow-based VAE posterior vs flow-based VAE prior
2. Disentanglement learning
3. Likelihood-free learning
4. Generative adversarial networks

# Likelihood based models

Is likelihood a good measure of model quality?

Poor likelihood  
Great samples

$$p_1(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \mathcal{N}(\mathbf{x}|\mathbf{x}_i, \epsilon \mathbf{I})$$

For small  $\epsilon$  this model will generate samples with great quality, but likelihood will be very poor.

Great likelihood  
Poor samples

$$p_2(\mathbf{x}) = 0.01p(\mathbf{x}) + 0.99p_{\text{noise}}(\mathbf{x})$$

$$\begin{aligned} \log [0.01p(\mathbf{x}) + 0.99p_{\text{noise}}(\mathbf{x})] &\geq \\ &\geq \log [0.01p(\mathbf{x})] = \log p(\mathbf{x}) - \log 100 \end{aligned}$$

Noisy irrelevant samples, but for high dimensions  $\log p(\mathbf{x})$  becomes proportional to  $m$ .

## Likelihood-free learning

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

### Where did we start

We would like to approximate true data distribution  $\pi(\mathbf{x})$ . Instead of searching true  $\pi(\mathbf{x})$  over all probability distributions, learn function approximation  $p(\mathbf{x}|\boldsymbol{\theta}) \approx \pi(\mathbf{x})$ .

Imagine we have two sets of samples

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

### Two sample test

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

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



# Likelihood-free learning

## Two sample test

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

## Desired behaviour

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

## GAN objective

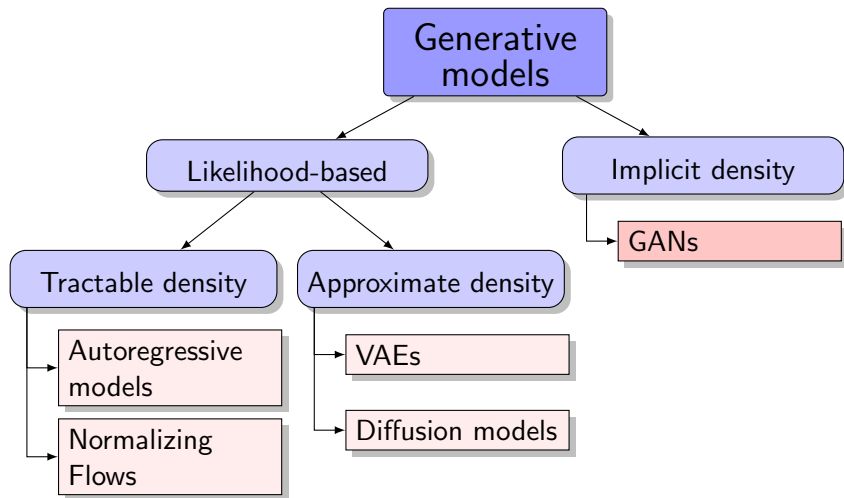
- ▶ **Generator:** generative model  $\mathbf{x} = G(\mathbf{z})$ , which makes generated sample more realistic.
- ▶ **Discriminator:** a classifier  $D(\mathbf{x}) \in [0, 1]$ , which distinguishes real samples from generated samples.

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

# Outline

1. Flow-based VAE posterior vs flow-based VAE prior
2. Disentanglement learning
3. Likelihood-free learning
4. Generative adversarial networks

# Generative models zoo



# Vanilla GAN optimality

## Theorem

The minimax game

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

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

## Proof (fixed $G$ )

$$\begin{aligned} V(G, D) &= \mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{x}|\theta)} \log(1 - D(\mathbf{x})) \\ &= \int \underbrace{[\pi(\mathbf{x}) \log D(\mathbf{x}) + p(\mathbf{x}|\theta) \log(1 - D(\mathbf{x}))]}_{y(D)} d\mathbf{x} \end{aligned}$$

$$\frac{dy(D)}{dD} = \frac{\pi(\mathbf{x})}{D(\mathbf{x})} - \frac{p(\mathbf{x}|\theta)}{1 - D(\mathbf{x})} = 0 \quad \Rightarrow \quad D^*(\mathbf{x}) = \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}) + p(\mathbf{x}|\theta)}$$

# Vanilla GAN optimality

Proof continued (fixed  $D = D^*$ )

$$\begin{aligned} V(G, D^*) &= \mathbb{E}_{\pi(\mathbf{x})} \log \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}) + p(\mathbf{x}|\theta)} + \mathbb{E}_{p(\mathbf{x}|\theta)} \log \frac{p(\mathbf{x}|\theta)}{\pi(\mathbf{x}) + p(\mathbf{x}|\theta)} \\ &= KL\left(\pi(\mathbf{x}) \parallel \frac{\pi(\mathbf{x}) + p(\mathbf{x}|\theta)}{2}\right) + KL\left(p(\mathbf{x}|\theta) \parallel \frac{\pi(\mathbf{x}) + p(\mathbf{x}|\theta)}{2}\right) - 2 \log 2 \\ &= 2JSD(\pi(\mathbf{x}) \parallel p(\mathbf{x}|\theta)) - 2 \log 2. \end{aligned}$$

Jensen-Shannon divergence (symmetric KL divergence)

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

Could be used as a distance measure!

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

# Vanilla GAN optimality

## Theorem

The minimax game

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

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

## Proof

for fixed  $G$ :

$$D^*(\mathbf{x}) = \frac{\pi(\mathbf{x})}{\pi(\mathbf{x}) + p(\mathbf{x}|\theta)}$$

for fixed  $D = D^*$ :

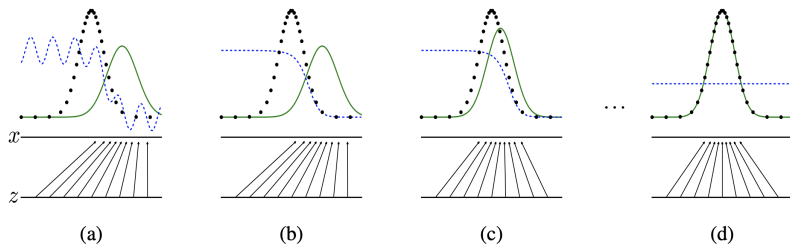
$$\min_G V(G, D^*) = \min_G [2JSD(\pi||p) - \log 4] = -\log 4, \quad \pi(\mathbf{x}) = p(\mathbf{x}|\theta).$$

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

# Vanilla GAN

## Objective

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



- ▶ Generator updates are made in parameter space.
- ▶ Discriminator is not optimal at every step.
- ▶ Generator and discriminator loss keeps oscillating during GAN training.

## Summary

- ▶ It is possible to use flows in VAE prior and posterior. This is almost the same.
- ▶ Disentanglement learning tries to make latent components more informative.
- ▶  $\beta$ -VAE makes the latent components more independent, but the reconstructions get poorer. DIP-VAE does not make the reconstructions worse using ELBO surgery theorem.
- ▶ Majority of disentanglement learning models use heuristic objective or regularizers to achieve the goal, but the task itself could not be solved without good inductive bias.
- ▶ Likelihood is not a perfect criteria to measure quality of generative model.
- ▶ Adversarial learning suggests to solve minimax problem to match the distributions.
- ▶ Vanilla GAN tries to optimize Jensen-Shannon divergence (in theory).