

# Deep Generative Models

## Lecture 6

Roman Isachenko



Spring, 2022

## Recap of previous lecture

### Change of variable theorem (CoV)

Let  $\mathbf{x}$  be a random variable with density function  $p(\mathbf{x})$  and  $f : \mathbb{R}^m \rightarrow \mathbb{R}^m$  is a differentiable, invertible function (diffeomorphism). If  $\mathbf{z} = f(\mathbf{x})$ ,  $\mathbf{x} = f^{-1}(\mathbf{z}) = g(\mathbf{z})$ , then

$$p(\mathbf{x}) = p(\mathbf{z}) |\det(\mathbf{J}_f)| = p(\mathbf{z}) \left| \det \left( \frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right) \right| = p(f(\mathbf{x})) \left| \det \left( \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right) \right|$$
$$p(\mathbf{z}) = p(\mathbf{x}) |\det(\mathbf{J}_g)| = p(\mathbf{x}) \left| \det \left( \frac{\partial \mathbf{x}}{\partial \mathbf{z}} \right) \right| = p(g(\mathbf{z})) \left| \det \left( \frac{\partial g(\mathbf{z})}{\partial \mathbf{z}} \right) \right|.$$

### Inverse function theorem

If function  $f$  is invertible and Jacobian is continuous and non-singular, then

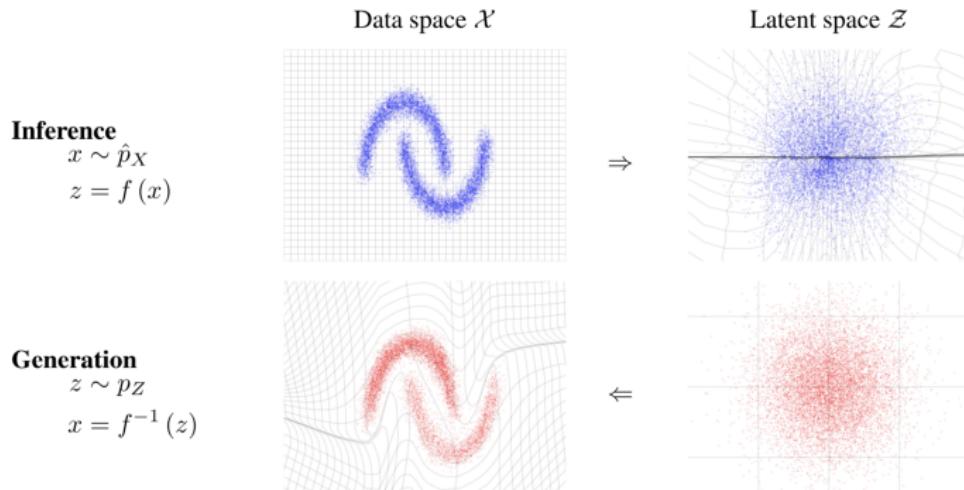
$$\mathbf{J}_f = \mathbf{J}_{g^{-1}} = \mathbf{J}_g^{-1}, \quad |\det(\mathbf{J}_f)| = \frac{1}{|\det(\mathbf{J}_g)|}$$

# Recap of previous lecture

## MLE problem

$$p(\mathbf{x}|\boldsymbol{\theta}) = p(\mathbf{z}) \left| \det \left( \frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right) \right| = p(f(\mathbf{x}, \boldsymbol{\theta})) \left| \det \left( \frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}} \right) \right|$$

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(f(\mathbf{x}, \boldsymbol{\theta})) + \log |\det(\mathbf{J}_f)| \rightarrow \max_{\boldsymbol{\theta}}$$



## Recap of previous lecture

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(f(\mathbf{x}, \boldsymbol{\theta})) + \log |\det(\mathbf{J}_f)| \rightarrow \max_{\boldsymbol{\theta}}$$

### Definition

Normalizing flow is a *differentiable, invertible* mapping from data  $\mathbf{x}$  to the noise  $\mathbf{z}$ .

- ▶ **Normalizing** means that the inverse flow takes samples from  $p(\mathbf{x})$  and normalizes them into samples from density  $p(\mathbf{z})$ .
- ▶ **Flow** refers to the trajectory followed by samples from  $p(\mathbf{z})$  as they are transformed by the sequence of transformations

$$\mathbf{z} = f_K \circ \cdots \circ f_1(\mathbf{x}); \quad \mathbf{x} = f_1^{-1} \circ \cdots \circ f_K^{-1}(\mathbf{z}) = g_1 \circ \cdots \circ g_K(\mathbf{z})$$

### Log likelihood

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(f_K \circ \cdots \circ f_1(\mathbf{x})) + \sum_{k=1}^K \log |\det(\mathbf{J}_{f_k})|,$$

where  $\mathbf{J}_{f_k} = \frac{\partial \mathbf{f}_k}{\partial \mathbf{f}_{k-1}}$ .

## Recap of previous lecture

### Forward KL for flow model

$$\log p(\mathbf{x}|\theta) = \log p(f(\mathbf{x}, \theta)) + \log |\det(\mathbf{J}_f)|$$

### Reverse KL for flow model

$$KL(p||\pi) = \mathbb{E}_{p(\mathbf{z})} [\log p(\mathbf{z}) - \log |\det(\mathbf{J}_g)| - \log \pi(g(\mathbf{z}, \theta))]$$

### Flow KL duality

$$\arg \min_{\theta} KL(\pi(\mathbf{x})||p(\mathbf{x}|\theta)) = \arg \min_{\theta} KL(p(\mathbf{z}|\theta)||p(\mathbf{z})).$$

- ▶  $p(\mathbf{z})$  is a base distribution;  $\pi(\mathbf{x})$  is a data distribution;
- ▶  $\mathbf{z} \sim p(\mathbf{z})$ ,  $\mathbf{x} = g(\mathbf{z}, \theta)$ ,  $\mathbf{x} \sim p(\mathbf{x}|\theta)$ ;
- ▶  $\mathbf{x} \sim \pi(\mathbf{x})$ ,  $\mathbf{z} = f(\mathbf{x}, \theta)$ ,  $\mathbf{z} \sim p(\mathbf{z}|\theta)$ ;

## Recap of previous lecture

### Flow log-likelihood

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \log p(f(\mathbf{x}, \boldsymbol{\theta})) + \log |\det(\mathbf{J}_f)|$$

The main challenge is a determinant of the Jacobian.

### Residual flows: planar/Sylvester

$$g(\mathbf{z}, \boldsymbol{\theta}) = \mathbf{z} + \mathbf{u} \sigma(\mathbf{w}^T \mathbf{z} + b); \quad g(\mathbf{z}, \boldsymbol{\theta}) = \mathbf{z} + \mathbf{A} \sigma(\mathbf{B}\mathbf{z} + \mathbf{b}).$$

Matrix determinant lemma for calculating the Jacobian.

### Linear flows

$$\mathbf{z} = f(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{W}\mathbf{x}, \quad \mathbf{W} \in \mathbb{R}^{m \times m}, \quad \boldsymbol{\theta} = \mathbf{W}, \quad \mathbf{J}_f = \mathbf{W}$$

Matrix decompositions (LU or QR helps to parametrize matrix  $\mathbf{W}$  and reduce the cost of computing the  $\det(\mathbf{J})$ ).

Rezende D. J., Mohamed S. Variational Inference with Normalizing Flows, 2015

Berg R. et al. Sylvester normalizing flows for variational inference, 2018

Kingma D. P., Dhariwal P. Glow: Generative Flow with Invertible 1x1 Convolutions, 2018

# Outline

1. Autoregressive flows
2. Inverse autoregressive flows
3. RealNVP: coupling layer
4. ELBO surgery

# Outline

1. Autoregressive flows
2. Inverse autoregressive flows
3. RealNVP: coupling layer
4. ELBO surgery

## Autoregressive flows

$$x_i = \tau(z_i, c(\mathbf{z}_{1:i-1})) \Leftrightarrow z_i = \tau^{-1}(x_i, c(\mathbf{z}_{1:i-1}))$$

- ▶  $\tau(\cdot, \cdot)$  – coupling law (invertible by first argument).
- ▶  $c(\cdot)$  – coupling function (do not need to be invertible, could be neural network).

### Coupling law $\tau(\cdot, \cdot)$

- ▶  $\tau(x, c) = x + c$  – additive;
- ▶  $\tau(x, c) = x \odot \exp c_1 + c_2$  – affine.

What is the Jacobian for the additive/affine coupling law?

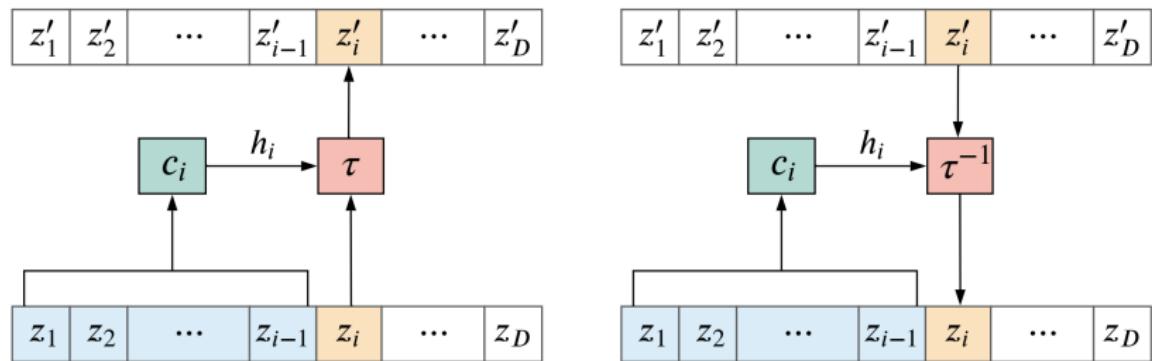
### Jacobian

$$\det \left( \frac{\partial \mathbf{x}}{\partial \mathbf{z}} \right) = \prod_{i=1}^m \frac{\partial x_i}{\partial z_i} = \prod_{i=1}^m \frac{\partial \tau(z_i, c(\mathbf{z}_{1:i-1}))}{\partial z_i}$$

# Autoregressive flows

## Forward and inverse transforms

$$x_i = \tau(z_i, c(\mathbf{z}_{1:i-1})) \Leftrightarrow z_i = \tau^{-1}(x_i, c(\mathbf{z}_{1:i-1}))$$



- ▶ Forward transform is **not sequential**.
- ▶ Inverse transform is **sequential**.

## Gaussian autoregressive model

Consider an autoregressive model

$$p(\mathbf{x}|\boldsymbol{\theta}) = \prod_{i=1}^m p(x_i|\mathbf{x}_{1:i-1}, \boldsymbol{\theta}), \quad p(x_i|\mathbf{x}_{1:i-1}, \boldsymbol{\theta}) = \mathcal{N}(\mu_i(\mathbf{x}_{1:i-1}), \sigma_i^2(\mathbf{x}_{1:i-1})).$$

Sampling: reparametrization trick

$$x_i = \sigma_i(\mathbf{x}_{1:i-1}) \cdot z_i + \mu_i(\mathbf{x}_{1:i-1}), \quad z_i \sim \mathcal{N}(0, 1).$$

Inverse transform

$$z_i = (x_i - \mu_i(\mathbf{x}_{1:i-1})) \cdot \frac{1}{\sigma_i(\mathbf{x}_{1:i-1})}.$$

We have got an autoregressive flow with base distribution  $\mathbf{z} = \mathcal{N}(0, 1)$  and affine coupling law  $\tau(x, \mu, \sigma) = x \odot \sigma + \mu$ .

## Gaussian autoregressive flow

$$\mathbf{x} = g(\mathbf{z}, \theta) \quad \Rightarrow \quad x_i = \sigma_i(\mathbf{x}_{1:i-1}) \cdot z_i + \mu_i(\mathbf{x}_{1:i-1}).$$

$$\mathbf{z} = f(\mathbf{x}, \theta) \quad \Rightarrow \quad z_i = (x_i - \mu_i(\mathbf{x}_{1:i-1})) \cdot \frac{1}{\sigma_i(\mathbf{x}_{1:i-1})}.$$

Generation function  $g(\mathbf{z}, \theta)$  is **sequential**. Inference function  $f(\mathbf{x}, \theta)$  is **not sequential**.

Forward KL for flow model

$$\log p(\mathbf{x}|\theta) = \log p(f(\mathbf{x}, \theta)) + \log \left| \det \left( \frac{\partial f(\mathbf{x}, \theta)}{\partial \mathbf{x}} \right) \right|$$

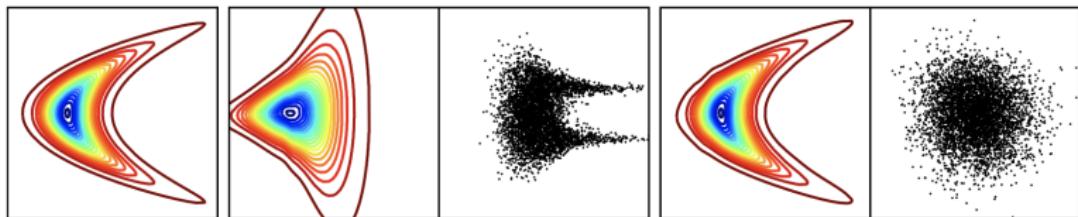
- ▶ We need to be able to compute  $f(\mathbf{x}, \theta)$  and its Jacobian.
- ▶ We need to be able to compute the density  $p(\mathbf{z})$ .
- ▶ We don't need to think about computing the function  $g(\mathbf{z}, \theta) = f^{-1}(\mathbf{z}, \theta)$  until we want to sample from the flow.

# Masked autoregressive flow (MAF)

## Gaussian autoregressive model

$$p(\mathbf{x}|\theta) = \prod_{i=1}^m p(x_i|\mathbf{x}_{1:i-1}, \theta) = \prod_{i=1}^m \mathcal{N}(x_i | \mu_i(\mathbf{x}_{1:i-1}), \sigma_i^2(\mathbf{x}_{1:i-1})).$$

We could use MADE for the conditionals. Samples from the base distribution could be an indicator of how good the flow was fitted.



(a) Target density

(b) MADE with Gaussian conditionals

(c) MAF with 5 layers

MAF is just a stacked MADE model with different ordering.

- ▶ Parallel density estimation.
- ▶ Sequential sampling.

# Outline

1. Autoregressive flows
2. Inverse autoregressive flows
3. RealNVP: coupling layer
4. ELBO surgery

## Inverse autoregressive flow (IAF)

Let's use the following reparametrization:  $\tilde{\sigma} = \frac{1}{\sigma}$ ;  $\tilde{\mu} = -\frac{\mu}{\sigma}$ .

### Gaussian autoregressive flow

$$x_i = \sigma_i(\mathbf{x}_{1:i-1}) \cdot z_i + \mu_i(\mathbf{x}_{1:i-1}) = (z_i - \tilde{\mu}_i(\mathbf{x}_{1:i-1})) \cdot \frac{1}{\tilde{\sigma}_i(\mathbf{x}_{1:i-1})}$$
$$z_i = (x_i - \mu_i(\mathbf{x}_{1:i-1})) \cdot \frac{1}{\sigma_i(\mathbf{x}_{1:i-1})} = \tilde{\sigma}_i(\mathbf{x}_{1:i-1}) \cdot x_i + \tilde{\mu}_i(\mathbf{x}_{1:i-1}).$$

Let's just swap  $\mathbf{z}$  and  $\mathbf{x}$ .

### Inverse autoregressive flow

$$\mathbf{x} = g(\mathbf{z}, \theta) \quad \Rightarrow \quad x_i = \tilde{\sigma}_i(\mathbf{z}_{1:i-1}) \cdot z_i + \tilde{\mu}_i(\mathbf{z}_{1:i-1})$$
$$\mathbf{z} = f(\mathbf{x}, \theta) \quad \Rightarrow \quad z_i = (x_i - \tilde{\mu}_i(\mathbf{z}_{1:i-1})) \cdot \frac{1}{\tilde{\sigma}_i(\mathbf{z}_{1:i-1})}.$$

# Inverse autoregressive flow (IAF)

Gaussian autoregressive flow:  $f(\mathbf{x}, \theta)$

$$x_i = \sigma_i(\mathbf{x}_{1:i-1}) \cdot z_i + \mu_i(\mathbf{x}_{1:i-1}).$$

Inverse transform:  $g(\mathbf{z}, \theta)$

$$z_i = (x_i - \mu_i(\mathbf{x}_{1:i-1})) \cdot \frac{1}{\sigma_i(\mathbf{x}_{1:i-1})};$$

$$z_i = \tilde{\sigma}_i(\mathbf{x}_{1:i-1}) \cdot x_i + \tilde{\mu}_i(\mathbf{x}_{1:i-1}).$$

Inverse autoregressive flow:  $f(\mathbf{x}, \theta)$

$$x_i = \tilde{\sigma}_i(\mathbf{z}_{1:i-1}) \cdot z_i + \tilde{\mu}_i(\mathbf{z}_{1:i-1}).$$

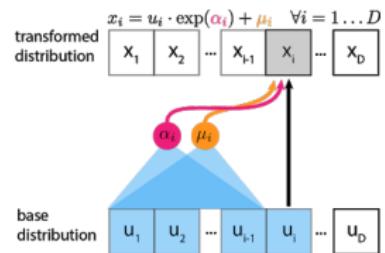
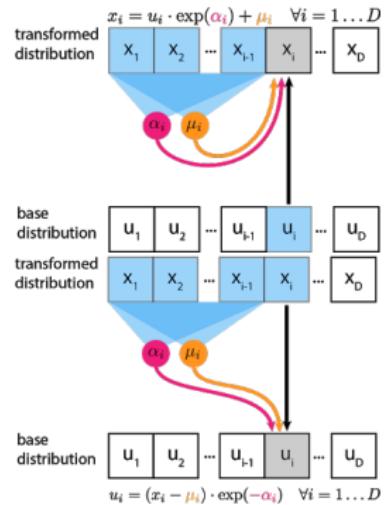


image credit: <https://blog.evjang.com/2018/01/nf2.html>

## Autoregressive flows

### Forward and inverse transforms in MAF

$$\mathbf{x} = g(\mathbf{z}, \theta) \quad \Rightarrow \quad x_i = \sigma_i(\mathbf{x}_{1:i-1}) \cdot z_i + \mu_i(\mathbf{x}_{1:i-1}).$$

$$\mathbf{z} = f(\mathbf{x}, \theta) \quad \Rightarrow \quad z_i = (x_i - \mu_i(\mathbf{x}_{1:i-1})) \cdot \frac{1}{\sigma_i(\mathbf{x}_{1:i-1})}.$$

- ▶ Sampling is sequential.
- ▶ Density estimation is parallel.

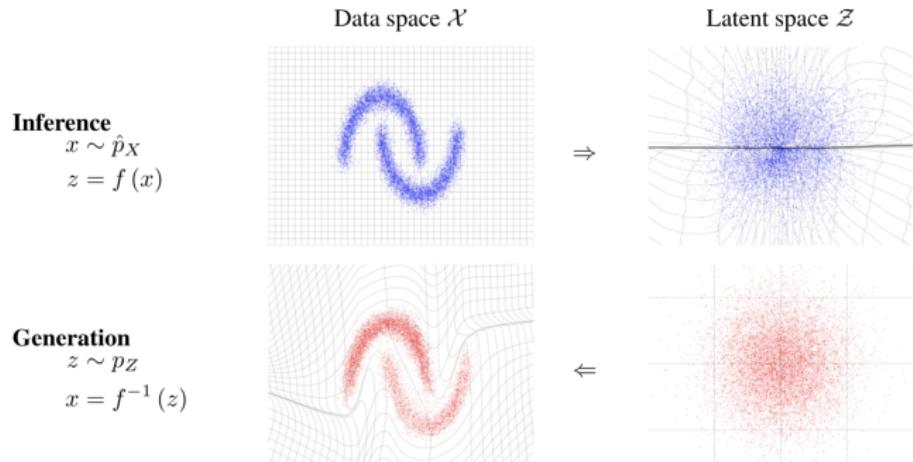
### Forward and inverse transforms in IAF

$$\mathbf{x} = g(\mathbf{z}, \theta) \quad \Rightarrow \quad x_i = \tilde{\sigma}_i(\mathbf{z}_{1:i-1}) \cdot z_i + \tilde{\mu}_i(\mathbf{z}_{1:i-1}).$$

$$\mathbf{z} = f(\mathbf{x}, \theta) \quad \Rightarrow \quad z_i = (x_i - \tilde{\mu}_i(\mathbf{z}_{1:i-1})) \cdot \frac{1}{\tilde{\sigma}_i(\mathbf{z}_{1:i-1})}.$$

- ▶ Sampling is parallel.
- ▶ Density estimation is sequential.

# Autoregressive flows



- ▶ MAF performs parallel inference that is useful for density estimation tasks (forward KL or MLE).
- ▶ IAF performs parallel generation that is useful for optimization of reverse KL.

# Outline

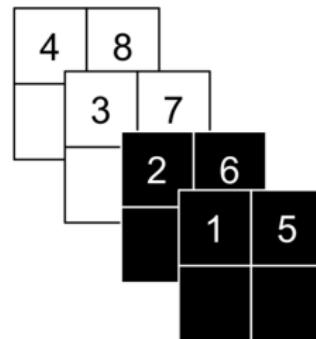
1. Autoregressive flows
2. Inverse autoregressive flows
3. RealNVP: coupling layer
4. ELBO surgery

# RealNVP

## Coupling layer

$$\begin{cases} \mathbf{z}_{1:d} = \mathbf{x}_{1:d}; \\ \mathbf{z}_{d:m} = \tau(\mathbf{x}_{d:m}, c(\mathbf{x}_{1:d})); \end{cases} \Leftrightarrow \begin{cases} \mathbf{x}_{1:d} = \mathbf{z}_{1:d}; \\ \mathbf{x}_{d:m} = \tau^{-1}(\mathbf{z}_{d:m}, c(\mathbf{z}_{1:d})). \end{cases}$$

## Image partitioning



Masked convolutions are used to define ordering.

# RealNVP

## Affine coupling law

$$\begin{cases} \mathbf{z}_{1:d} = \mathbf{x}_{1:d}; \\ \mathbf{z}_{d:m} = \mathbf{x}_{d:m} \odot \exp(c_1(\mathbf{x}_{1:d}, \theta)) + c_2(\mathbf{x}_{1:d}, \theta). \end{cases}$$

$$\begin{cases} \mathbf{x}_{1:d} = \mathbf{z}_{1:d}; \\ \mathbf{x}_{d:m} = (\mathbf{z}_{d:m} - c_2(\mathbf{z}_{1:d}, \theta)) \odot \exp(-c_1(\mathbf{z}_{1:d}, \theta)). \end{cases}$$

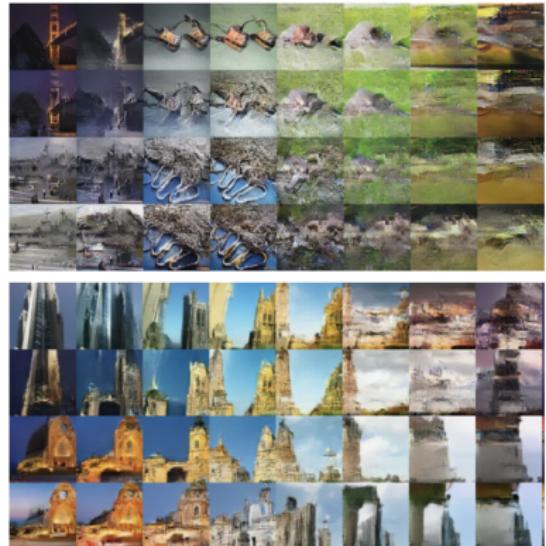
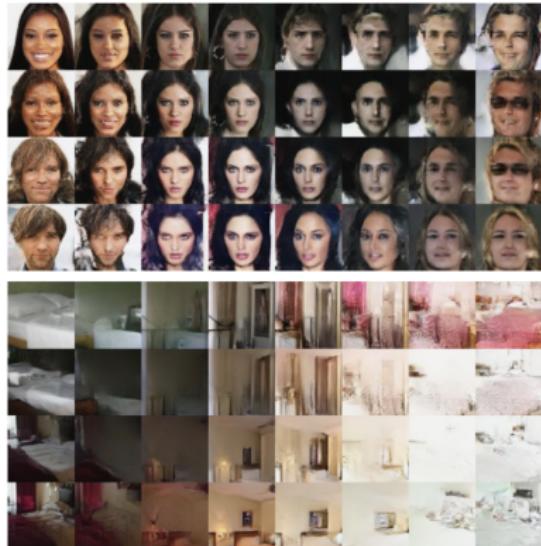
## Jacobian

$$\det\left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}}\right) = \det\begin{pmatrix} \mathbf{I}_d & 0_{d \times m-d} \\ \frac{\partial \mathbf{z}_{d:m}}{\partial \mathbf{x}_{1:d}} & \frac{\partial \mathbf{z}_{d:m}}{\partial \mathbf{x}_{d:m}} \end{pmatrix} = \prod_{i=1}^{m-d} \exp(c_1(\mathbf{x}_{1:d}, \theta)_i).$$

Non-Volume Preserving (the determinant of Jacobian  $\neq 0$ ).

# RealNVP

## Flow samples



## MAF vs IAF vs RealNVP

### MADE/MAF

$$\mathbf{x} = \sigma(\mathbf{z}) \odot \mathbf{z} + \boldsymbol{\mu}(\mathbf{x}).$$

Estimating the density  $p(\mathbf{x}|\theta)$  - 1 pass, sampling -  $m$  passes.

### IAF

$$\mathbf{x} = \tilde{\sigma}(\mathbf{z}) \odot \mathbf{z} + \tilde{\boldsymbol{\mu}}(\mathbf{z}).$$

Estimating the density  $p(\mathbf{x}|\theta)$  -  $m$  passes, sampling - 1 pass.

### RealINVP

$$\begin{cases} \mathbf{x}_{1:d} &= \mathbf{z}_{1:d}; \\ \mathbf{x}_{d:m} &= \mathbf{z}_{d:m} \odot \exp(c_1(\mathbf{z}_{1:d}, \theta)) + c_2(\mathbf{z}_{1:d}, \theta). \end{cases}$$

Estimating the density  $p(\mathbf{x}|\theta)$  - 1 pass, sampling - 1 pass.

# MAF vs IAF vs RealNVP

## RealNVP

$$\begin{cases} \mathbf{x}_{1:d} &= \mathbf{z}_{1:d}; \\ \mathbf{x}_{d:m} &= \mathbf{z}_{d:m} \odot \exp(c_1(\mathbf{z}_{1:d}, \theta)) + c_2(\mathbf{z}_{1:d}, \theta). \end{cases}$$

- ▶ Calculating the density  $p(\mathbf{x}|\theta)$  - 1 pass.
- ▶ Sampling - 1 pass.

RealNVP is a special case of MAF and IAF:

## MAF

$$\begin{cases} \mu_i = 0, \sigma_i = 1, i = 1, \dots, d; \\ \mu_i, \sigma_i - \text{functions of } \mathbf{x}_{1:d}, i = d + 1, \dots, m. \end{cases}$$

## IAF

$$\begin{cases} \tilde{\mu}_i = 0, \tilde{\sigma}_i = 1, i = 1, \dots, d; \\ \tilde{\mu}_i, \tilde{\sigma}_i - \text{functions of } \mathbf{z}_{1:d}, i = d + 1, \dots, m. \end{cases}$$

# Linear flows

## RealNVP

$$\begin{cases} \mathbf{z}_{1:d} = \mathbf{x}_{1:d}; \\ \mathbf{z}_{d:m} = \tau(\mathbf{x}_{d:m}, c(\mathbf{x}_{1:d})); \end{cases} \Leftrightarrow \begin{cases} \mathbf{x}_{1:d} = \mathbf{z}_{1:d}; \\ \mathbf{x}_{d:m} = \tau^{-1}(\mathbf{z}_{d:m}, c(\mathbf{z}_{1:d})). \end{cases}$$

- ▶ First step is a **split** operator which decouples a variable into 2 subparts:  $\mathbf{x}_1$  and  $\mathbf{x}_2$  (usually channel-wise).
- ▶ We should **permute** components between different layers.

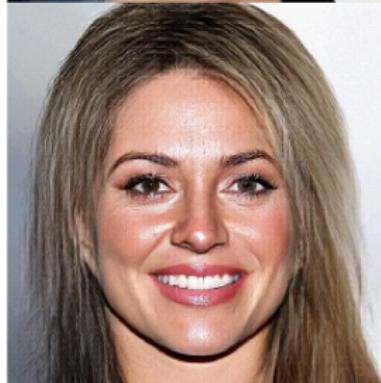
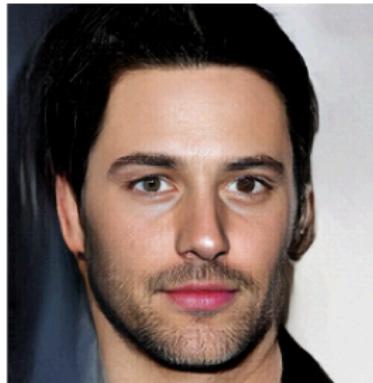
$$\mathbf{z} = \mathbf{W}\mathbf{x}, \quad \mathbf{W} \in \mathbb{R}^{m \times m}$$

In general, we need  $O(m^3)$  to invert matrix.

## Invertibility

- ▶ Diagonal matrix  $O(m)$ .
- ▶ Triangular matrix  $O(m^2)$ .
- ▶ It is impossible to parametrize all invertible matrices.

## Glow samples



# Outline

1. Autoregressive flows
2. Inverse autoregressive flows
3. RealNVP: coupling layer
4. ELBO surgery

# Likelihood-based models

## Exact likelihood evaluation

- ▶ Autoregressive models (WaveNet, PixelCNN, PixelCNN++);
- ▶ Flow models (RealNVP, IAF, Glow).

## Approximate likelihood evaluation

- ▶ Latent variable models (VAE).

What are the pros and cons of each of them?

## VAE limitations

- ▶ Poor generative distribution (decoder)

$$p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\mu_{\boldsymbol{\theta}}(\mathbf{z}), \sigma_{\boldsymbol{\theta}}^2(\mathbf{z})) \quad \text{or} \quad = \text{Softmax}(\pi_{\boldsymbol{\theta}}(\mathbf{z})).$$

- ▶ 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)

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

## ELBO interpretations

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

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

- ▶ Evidence minus posterior KL

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

- ▶ Average reconstruction loss with regularizer (prior KL)

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})} [\log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) + \log p(\mathbf{z}) - \log q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})] \\ &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})} \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi}) || p(\mathbf{z})).\end{aligned}$$

# ELBO surgery

$$\frac{1}{n} \sum_{i=1}^n \mathcal{L}_i(q, \theta) = \frac{1}{n} \sum_{i=1}^n [\mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)} \log p(\mathbf{x}_i|\mathbf{z}, \theta) - KL(q(\mathbf{z}|\mathbf{x}_i)||p(\mathbf{z}))].$$

## Theorem

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

- ▶  $q_{\text{agg}}(\mathbf{z}) = \frac{1}{n} \sum_{i=1}^n q(\mathbf{z}|\mathbf{x}_i)$  – **aggregated** posterior distribution.
- ▶  $\mathbb{I}_q[\mathbf{x}, \mathbf{z}]$  – mutual information between  $\mathbf{x}$  and  $\mathbf{z}$  under empirical data distribution and distribution  $q(\mathbf{z}|\mathbf{x})$ .
- ▶ First term pushes  $q_{\text{agg}}(\mathbf{z})$  towards the prior  $p(\mathbf{z})$ .
- ▶ Second term reduces the amount of information about  $\mathbf{x}$  stored in  $\mathbf{z}$ .

# ELBO surgery

## Theorem

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

## Proof

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n KL(q(\mathbf{z}|\mathbf{x}_i) || p(\mathbf{z})) &= \frac{1}{n} \sum_{i=1}^n \int q(\mathbf{z}|\mathbf{x}_i) \log \frac{q(\mathbf{z}|\mathbf{x}_i)}{p(\mathbf{z})} d\mathbf{z} = \\ &= \frac{1}{n} \sum_{i=1}^n \int q(\mathbf{z}|\mathbf{x}_i) \log \frac{q_{\text{agg}}(\mathbf{z})q(\mathbf{z}|\mathbf{x}_i)}{p(\mathbf{z})q_{\text{agg}}(\mathbf{z})} d\mathbf{z} = \int \frac{1}{n} \sum_{i=1}^n q(\mathbf{z}|\mathbf{x}_i) \log \frac{q_{\text{agg}}(\mathbf{z})}{p(\mathbf{z})} d\mathbf{z} + \\ &+ \frac{1}{n} \sum_{i=1}^n \int q(\mathbf{z}|\mathbf{x}_i) \log \frac{q(\mathbf{z}|\mathbf{x}_i)}{q_{\text{agg}}(\mathbf{z})} d\mathbf{z} = KL(q_{\text{agg}}(\mathbf{z}) || p(\mathbf{z})) + \frac{1}{n} \sum_{i=1}^n KL(q(\mathbf{z}|\mathbf{x}_i) || q_{\text{agg}}(\mathbf{z})) \end{aligned}$$

Without proof:

$$\mathbb{I}_q[\mathbf{x}, \mathbf{z}] = \frac{1}{n} \sum_{i=1}^n KL(q(\mathbf{z}|\mathbf{x}_i) || q_{\text{agg}}(\mathbf{z})) \in [0, \log n].$$

# ELBO surgery

## ELBO revisiting

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

Prior distribution  $p(\mathbf{z})$  is only in the last term.

## Optimal VAE 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  $p(\mathbf{z})$  is the aggregated posterior  $q_{\text{agg}}(\mathbf{z})$ .

## Summary

- ▶ Autoregressive flows use autoregressive transformation to make the Jacobian triangular.
- ▶ MAF/IAF is a special case of autoregressive flows.
- ▶ The RealNVP is an effective type of flow (special case of AR flows) that uses coupling layer.