# Deep Generative Models

## Lecture 4

Roman Isachenko

Ozon Masters

Spring, 2022

# Recap of previous lecture

### Variational lower Bound (ELBO)

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

$$\mathcal{L}(q, \boldsymbol{\theta}) = \int q(\mathbf{z}) \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z})} d\mathbf{z} = \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z})||p(\mathbf{z}))$$

### Log-likelihood decomposition

$$\log p(\mathbf{x}|\boldsymbol{\theta}) = \mathbb{E}_q \log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) - KL(q(\mathbf{z})||p(\mathbf{z})) + KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})).$$

- Instead of maximizing incomplete likelihood, maximize ELBO

$$\max_{\boldsymbol{\theta}} p(\mathbf{x}|\boldsymbol{\theta}) \quad \rightarrow \quad \max_{q, \boldsymbol{\theta}} \mathcal{L}(q, \boldsymbol{\theta})$$

- Maximization of ELBO by variational distribution $q$ is equivalent to minimization of KL

$$\max_q \mathcal{L}(q, \boldsymbol{\theta}) \equiv \min_q KL(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})).$$

# Recap of previous lecture

### EM-algorithm

- E-step

$$q^*(\mathbf{z}) = \arg\max_q \mathcal{L}(q, \boldsymbol{\theta}^*) = \arg\min_q KL(q(\mathbf{z})\|p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^*));$$

- M-step

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(q^*, \boldsymbol{\theta});$$

### Amortized variational inference

Restrict a family of all possible distributions $q(\mathbf{z})$ to a parametric class $q(\mathbf{z}|\mathbf{x}, \phi)$ conditioned on samples $\mathbf{x}$ with parameters $\phi$.

**Variational Bayes**

- E-step

$$\phi_k = \phi_{k-1} + \eta \nabla_\phi \mathcal{L}(\phi, \boldsymbol{\theta}_{k-1})|_{\phi=\phi_{k-1}}$$

- M-step

$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} + \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\phi_k, \boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{k-1}}$$

# Variational autoencoder (VAE)

### Final algorithm

- pick random sample $\mathbf{x}_i, i \sim U[1, n]$.
- compute the objective:
$$\epsilon^* \sim r(\epsilon); \quad \mathbf{z}^* = g(\mathbf{x}, \epsilon^*, \phi);$$

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

- compute a stochastic gradients w.r.t. $\phi$ and $\boldsymbol{\theta}$

$$\nabla_\phi \mathcal{L}(\phi, \boldsymbol{\theta}) \approx \nabla_\phi \log p(\mathbf{x}|g(\mathbf{x}, \epsilon^*, \phi), \boldsymbol{\theta}) - \nabla_\phi KL(q(\mathbf{z}|\mathbf{x}, \phi)||p(\mathbf{z}));$$
$$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\phi, \boldsymbol{\theta}) \approx \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}|\mathbf{z}^*, \boldsymbol{\theta}).$$
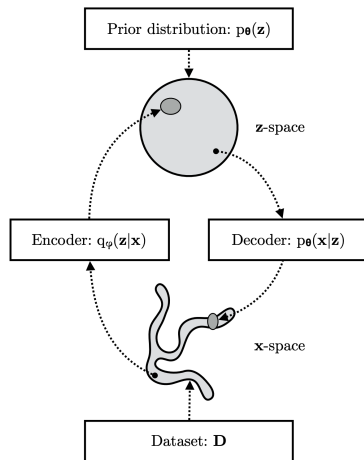
- update $\boldsymbol{\theta}, \phi$ according to the selected optimization method (SGD, Adam, RMSProp):

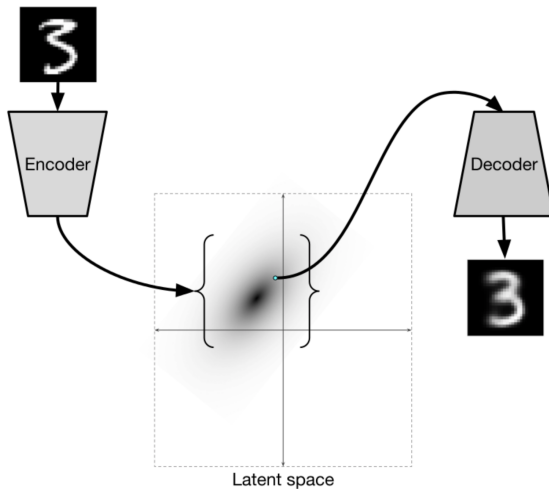$$\phi := \phi + \eta \nabla_\phi \mathcal{L}(\phi, \boldsymbol{\theta}),$$
$$\boldsymbol{\theta} := \boldsymbol{\theta} + \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\phi, \boldsymbol{\theta}).$$

# Variational autoencoder (VAE)

- ▶ VAE learns stochastic mapping between **x**-space, from complicated distribution $\pi(\mathbf{x})$, and a latent **z**-space, with simple distribution.

- ▶ The generative model learns a joint distribution $p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$, with a prior distribution $p(\mathbf{z})$, and a stochastic decoder $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$.

- ▶ The stochastic encoder $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})$ (inference model), approximates the true but intractable posterior $p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta})$ of the generative model.



Prior distribution: $p_{\boldsymbol{\theta}}(\mathbf{z})$

**z**-space

Encoder: $q_{\varphi}(\mathbf{z}|\mathbf{x})$     Decoder: $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$

**x**-space

Dataset: **D**

*Kingma D. P., Welling M. An introduction to variational autoencoders, 2019*

# Variational Autoencoder

# Variational autoencoder (VAE)

- Encoder $q(\mathbf{z}|\mathbf{x}, \phi) = \mathrm{NN}_e(\mathbf{x}, \phi)$ outputs $\boldsymbol{\mu}_\phi(\mathbf{x})$ and $\boldsymbol{\sigma}_\phi(\mathbf{x})$.
- Decoder $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = \mathrm{NN}_d(\mathbf{z}, \boldsymbol{\theta})$ outputs parameters of the sample distribution.
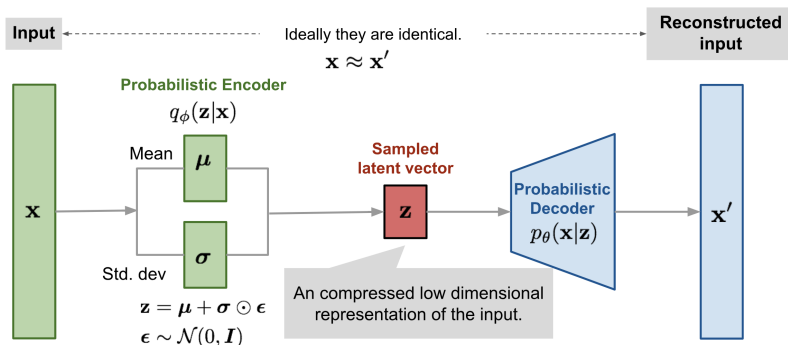


image credit:
https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

# Bayesian framework

### Posterior distribution

$$p(\boldsymbol{\theta}|\mathbf{X}) = \frac{p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{X})} = \frac{p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}$$

### Bayesian inference

$$p(\mathbf{x}|\mathbf{X}) = \int p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{X})d\boldsymbol{\theta}$$

### Maximum a posteriori (MAP) estimation

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}|\mathbf{X}) = \arg\max_{\boldsymbol{\theta}}\big(\log p(\mathbf{X}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})\big)$$

### MAP inference

$$p(\mathbf{x}|\mathbf{X}) = \int p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{X})d\boldsymbol{\theta} = \int p(\mathbf{x}|\boldsymbol{\theta})\delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*)d\boldsymbol{\theta} \approx p(\mathbf{x}|\boldsymbol{\theta}^*).$$

# VAE as Bayesian model

## Posterior distribution

$$p(\boldsymbol{\theta}|\mathbf{X}) = \frac{p(\mathbf{X}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{X})}$$

## ELBO

$$
\begin{aligned}
\log p(\boldsymbol{\theta}|\mathbf{X}) &= \log p(\mathbf{X}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) - \log p(\mathbf{X}) \\
&= \mathcal{L}(q, \boldsymbol{\theta}) + KL(q||p) + \log p(\boldsymbol{\theta}) - \log p(\mathbf{X}) \\
&\geq [\mathcal{L}(q, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta})] - \log p(\mathbf{X}).
\end{aligned}
$$

## EM-algorithm

- E-step

$$q(\mathbf{z}) = \arg\max_{q} \mathcal{L}(q, \boldsymbol{\theta}^*) = \arg\min_{q} KL(q||p) = p(\mathbf{z}|\mathbf{x}, \boldsymbol{\theta}^*);$$

- M-step

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \left[\mathcal{L}(q, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta})\right].$$

# VAE limitations

- Poor variational posterior distribution (inference model encoder)

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

- Poor prior distribution

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

- Poor probabilistic model (generative model, decoder)

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

- Loose lower bound

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

# Likelihood-based models so far...

### Autoregressive models

$$p(\mathbf{x}|\boldsymbol{\theta}) = \prod_{i=1}^{m} p(x_i|\mathbf{x}_{1:i-1}, \boldsymbol{\theta})$$

- ▶ tractable likelihood,
- ▶ no inferred latent factors.

### Latent variable models

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

- ▶ latent feature representation,
- ▶ intractable likelihood.

How to build model with latent variables and tractable likelihood?

# Normalizing flows prerequisites

### Change of variable theorem

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

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

$$p(\mathbf{z}) = p(\mathbf{x}) \left| \det \left( \frac{\partial \mathbf{x}}{\partial \mathbf{z}} \right) \right| = p(g(\mathbf{z})) \left| \det \left( \frac{\partial g(\mathbf{z})}{\partial \mathbf{z}} \right) \right|.$$
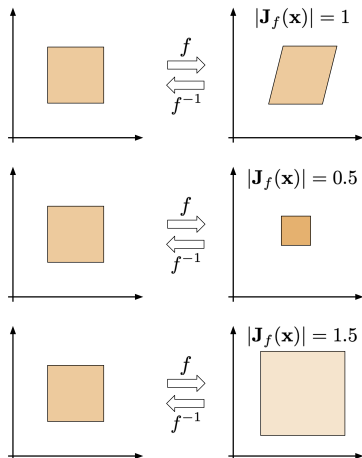
### Inverse function theorem

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

$$\left| \det \left( \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right) \right| = \left| \det \left( \frac{\partial g^{-1}(\mathbf{x})}{\partial \mathbf{x}} \right) \right| = \left| \det \left( \frac{\partial g(\mathbf{z})}{\partial \mathbf{z}} \right) \right|^{-1}$$

# Jacobian-determinant

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

- $\mathbf{x}$ and $\mathbf{z}$ have the same dimensionality ($\mathbb{R}^m$).
- $f(\mathbf{x}, \boldsymbol{\theta})$ could be parametric function.
- Determinant of Jacobian $\mathbf{J} = \frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}}$ shows how the volume changes under the transorfmation.
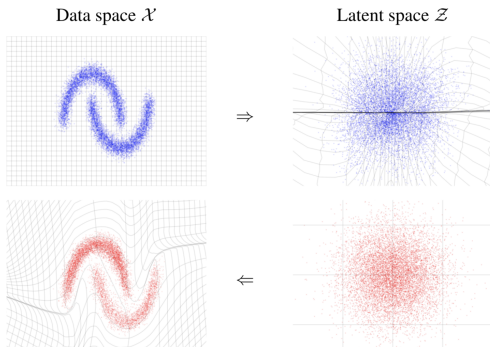


$|\mathbf{J}_f(\mathbf{x})| = 1$

$|\mathbf{J}_f(\mathbf{x})| = 0.5$

$|\mathbf{J}_f(\mathbf{x})| = 1.5$

*https://jmtomczak.github.io/blog/3/3_flows.html*

# Fitting flows

## MLE problem

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

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



Data space $\mathcal{X}$          Latent space $\mathcal{Z}$

**Inference**
$x \sim \hat{p}_X$
$z = f(x)$

$\Rightarrow$

**Generation**
$z \sim p_Z$
$x = f^{-1}(z)$

$\Leftarrow$

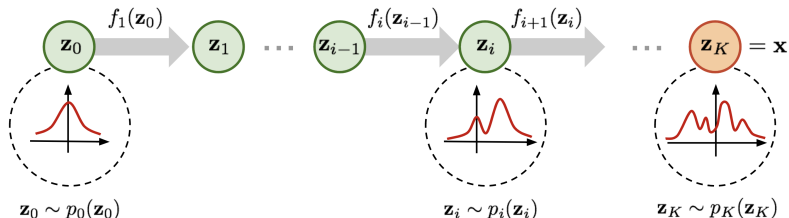*Dinh L., Sohl-Dickstein J., Bengio S. Density estimation using Real NVP, 2016*

# Composition of flows

### Theorem

Diffeomorphisms are **composable** (If $\{f_k\}_{k=1}^{K}$ satisfy conditions of the change of variable theorem, then $\mathbf{z} = f(\mathbf{x}) = f_K \circ \cdots \circ f_1(\mathbf{x})$ also satisfies it).

$$p(\mathbf{x}) = p(f(\mathbf{x})) \left| \det\left( \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right) \right| = p(f(\mathbf{x})) \left| \det\left( \frac{\partial f_K \circ \cdots \circ f_1(\mathbf{x})}{\partial \mathbf{x}} \right) \right| =$$

$$= p(f(\mathbf{x})) \left| \det\left( \frac{\partial \mathbf{f}_K}{\partial \mathbf{f}_{K-1}} \cdots \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}} \right) \right| = p(f(\mathbf{x})) \prod_{k=1}^{K} \left| \det\left( \frac{\partial \mathbf{f}_k}{\partial \mathbf{f}_{k-1}} \right) \right|$$



$$\mathbf{z}_0 \sim p_0(\mathbf{z}_0) \qquad\qquad \mathbf{z}_i \sim p_i(\mathbf{z}_i) \qquad\qquad \mathbf{z}_K \sim p_K(\mathbf{z}_K)$$

# Flows

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

### Definition
Normalizing flow is a *differentiable, invertible* mapping from data $\mathbf{x}$ to the noise $\mathbf{z}$.
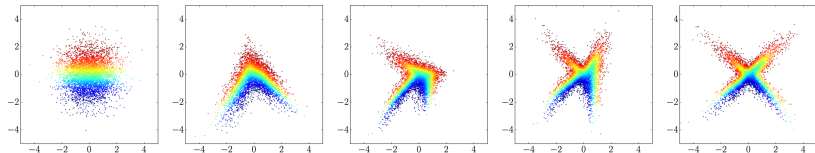
- **Normalizing** means that the inverse flow takes samples from $p(\mathbf{x})$ and normalizes them into samples from density $p(\mathbf{z})$.
- **Flow** refers to the trajectory followed by samples from $p(\mathbf{z})$ as they are transformed by the sequence of transformations

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

$$p(\mathbf{x}) = p(f_K \circ \cdots \circ f_1(\mathbf{x})) \left| \det \left( \frac{\partial f_K \circ \cdots \circ f_1(\mathbf{x})}{\partial \mathbf{x}} \right) \right| =$$

$$= p(f_K \circ \cdots \circ f_1(\mathbf{x})) \prod_{k=1}^{K} \left| \det \left( \frac{\partial \mathbf{f}_k}{\partial \mathbf{f}_{k-1}} \right) \right|.$$

# Flows

## Example of a 4-step flow



## Flow likelihood

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

What is the complexity of the determinant computation?

## What we want

- Efficient computation of Jacobian $\frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \mathbf{x}}$;
- Efficient sampling from the base distribution $p(\mathbf{z})$;
- Efficient inversion of $f(\mathbf{x}, \boldsymbol{\theta})$.

Papamakarios G. et al. Normalizing flows for probabilistic modeling and inference, 2019

# Forward KL vs Reverse KL

### Forward KL

$$KL(\pi||p) = \int \pi(\mathbf{x}) \log \frac{\pi(\mathbf{x})}{p(\mathbf{x}|\boldsymbol{\theta})} d\mathbf{x}$$
$$= -\mathbb{E}_{\pi(\mathbf{x})} \log p(\mathbf{x}|\boldsymbol{\theta}) + \text{const} \to \min_{\boldsymbol{\theta}}$$

Maximum likelihood estimation is equivalent to minimization of the Monte-Carlo estimation of forward KL.

### Forward KL for flow model

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

- We need to be able to compute $f(\mathbf{x}, \boldsymbol{\theta})$ 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 $g(\mathbf{z}, \boldsymbol{\theta}) = f^{-1}(\mathbf{z}, \boldsymbol{\theta})$ until we want to sample from the flow.

# Forward KL vs Reverse KL

### Reverse KL

$$KL(p||\pi) = \int p(\mathbf{x}|\boldsymbol{\theta}) \log \frac{p(\mathbf{x}|\boldsymbol{\theta})}{\pi(\mathbf{x})} d\mathbf{x}$$
$$= \mathbb{E}_{p(\mathbf{x}|\boldsymbol{\theta})} \left[ \log p(\mathbf{x}|\boldsymbol{\theta}) - \log \pi(\mathbf{x}) \right] \rightarrow \min_{\boldsymbol{\theta}}$$

### Reverse KL for flow model

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

$$KL(p||\pi) = \mathbb{E}_{p(\mathbf{z})} \left[ \log p(\mathbf{z}) - \log \left| \det \left( \frac{\partial g(\mathbf{z}, \boldsymbol{\theta})}{\partial \mathbf{z}} \right) \right| - \log \pi(g(\mathbf{z}, \boldsymbol{\theta})) \right]$$

- We need to be able to compute $g(\mathbf{z}, \boldsymbol{\theta})$ and its Jacobian.
- We need to be able to sample from the density $p(\mathbf{z})$ (do not need to evaluate it).
- We don't need to think about computing the function $f(\mathbf{x}, \boldsymbol{\theta})$.

# Flow KL duality

### Theorem
Fitting flow model $p(\mathbf{x}|\boldsymbol{\theta})$ to the target distribution $\pi(\mathbf{x})$ using forward KL (MLE) is equivalent to fitting the induced distribution $p(\mathbf{z}|\boldsymbol{\theta})$ to the base $p(\mathbf{z})$ using reverse KL:

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

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

$$\log p(\mathbf{z}|\boldsymbol{\theta}) = \log \pi(g(\mathbf{z}, \boldsymbol{\theta})) + \log \left| \det \left( \frac{\partial g(\mathbf{z}, \boldsymbol{\theta})}{\partial \mathbf{z}} \right) \right|;$$

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

Papamakarios G. et al. Normalizing flows for probabilistic modeling and inference, 2019

# Flow KL duality

### Theorem
Fitting flow model $p(\mathbf{x}|\boldsymbol{\theta})$ to the target distribution $\pi(\mathbf{x})$ using forward KL (MLE) is equivalent to fitting the induced distribution $p(\mathbf{z}|\boldsymbol{\theta})$ to the base $p(\mathbf{z})$ using reverse KL:

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

### Proof

$$KL\left(p(\mathbf{z}|\boldsymbol{\theta})||\pi(\mathbf{z})\right) = \mathbb{E}_{p(\mathbf{z}|\boldsymbol{\theta})}\big[\log p(\mathbf{z}|\boldsymbol{\theta}) - \log p(\mathbf{z})\big] =$$

$$= \mathbb{E}_{p(\mathbf{z}|\boldsymbol{\theta})}\left[\log \pi(g(\mathbf{z},\boldsymbol{\theta})) + \log\left|\det\left(\frac{\partial g(\mathbf{z},\boldsymbol{\theta})}{\partial \mathbf{z}}\right)\right| - \log p(\mathbf{z})\right] =$$

$$= \mathbb{E}_{\pi(\mathbf{x})}\left[\log \pi(\mathbf{x}) - \log\left|\det\left(\frac{\partial f(\mathbf{x},\boldsymbol{\theta})}{\partial \mathbf{x}}\right)\right| - \log p(f(\mathbf{x},\boldsymbol{\theta}))\right] =$$

$$= \mathbb{E}_{\pi(\mathbf{x})}\big[\log \pi(\mathbf{x}) - \log p(\mathbf{x}|\boldsymbol{\theta})\big] = KL(\pi(\mathbf{x})||p(\mathbf{x}|\boldsymbol{\theta})).$$

Papamakarios G., Pavlakou T., Murray I. *Masked Autoregressive Flow for Density Estimation*, 2017

# Summary

- The VAE model is an LVM with two neural network: stochastic encoder $q(\mathbf{z}|\mathbf{x}, \phi)$ and stochastic decoder $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$.

- VAE is not a "true" bayesian model since parameters $\boldsymbol{\theta}$ do not have a prior distribution.

- Standart VAE has several limitations that we will address later in the course.

- Flow models transform a simple base distribution to a complex one via a sequence of invertible transformations.

- Flow models have a tractable likelihood that is given by the change of variable theorem.

- Flows could be fitted using forward and reverse KL minimization. We will consider each of the scenarios later in the course.