

# Deep Generative Models

## Lecture 8

Roman Isachenko

 Ozon Masters

Spring, 2022

# 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  $q(\mathbf{z})$ .

---

Hoffman M. D., Johnson M. J. *ELBO surgery: yet another way to carve up the variational evidence lower bound*, 2016

# Recap of previous lecture

## 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  $q(\mathbf{z})$ .

## VampPrior

$$p(\mathbf{z}|\boldsymbol{\lambda}) = \frac{1}{K} \sum_{k=1}^K q(\mathbf{z}|\mathbf{u}_k),$$

where  $\boldsymbol{\lambda} = \{\mathbf{u}_1, \dots, \mathbf{u}_K\}$  are trainable pseudo-inputs.

## Flow-based VAE prior

$$\log p(\mathbf{z}|\boldsymbol{\lambda}) = \log p(\boldsymbol{\epsilon}) + \log \det \left| \frac{d\boldsymbol{\epsilon}}{d\mathbf{z}} \right| = \log p(\boldsymbol{\epsilon}) + \log \det \left| \frac{\partial f(\mathbf{z}, \boldsymbol{\lambda})}{\partial \mathbf{z}} \right|$$

# Recap of previous lecture

## Standart ELBO

$$p(\mathbf{x}|\boldsymbol{\theta}) \geq \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})} \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})} \rightarrow \max_{\boldsymbol{\phi}, \boldsymbol{\theta}}.$$

## Expressive flow-based VAE posterior

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

## ELBO with flow-based posterior

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

- ▶ Obtain samples  $\mathbf{z}$  from the encoder  $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\phi})$ .
- ▶ Apply flow model  $\mathbf{z}^* = f(\mathbf{z}, \boldsymbol{\lambda})$ .
- ▶ Compute likelihood for  $\mathbf{z}^*$  using the decoder, base distribution for  $\mathbf{z}^*$  and the Jacobian.

# Recap of previous lecture

## Expressive flow-based VAE posterior

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

## Expressive flow-based VAE prior

$$\log p(\mathbf{z}|\lambda) = \log p(\epsilon) + \log \det \left| \frac{d\epsilon}{d\mathbf{z}} \right|; \quad \mathbf{z} = g(\epsilon, \lambda) = f^{-1}(\epsilon, \lambda)$$

## Theorem

VAE with the flow-based prior for latent code  $\mathbf{z}$  is equivalent to VAE with flow-based posterior for latent code  $\mathbf{z}$ .

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

# Outline

# Dequantization

- ▶ Images are discrete data, pixels lie in the  $\{0, 255\}$  integer domain (the model is  $P(\mathbf{x}|\boldsymbol{\theta}) = \text{Categorical}(\boldsymbol{\pi}(\boldsymbol{\theta}))$ ).
- ▶ Flow is a continuous model (it works with continuous data  $\mathbf{x}$ ).

By fitting a continuous density model to discrete data, one can produce a degenerate solution with all probability mass on discrete values.

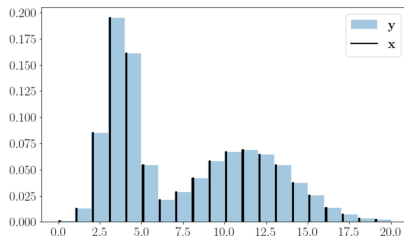
How to convert a discrete data distribution to a continuous one?

## Uniform dequantization

$$\mathbf{x} \sim \text{Categorical}(\boldsymbol{\pi})$$

$$\mathbf{u} \sim U[0, 1]$$

$$\mathbf{y} = \mathbf{x} + \mathbf{u} \sim \text{Continuous}$$



# Uniform dequantization

## Statement

Fitting continuous model  $p(\mathbf{y}|\boldsymbol{\theta})$  on uniformly dequantized data  $\mathbf{y} = \mathbf{x} + \mathbf{u}$ ,  $\mathbf{u} \sim U[0, 1]$  is equivalent to maximization of a lower bound on log-likelihood for a discrete model:

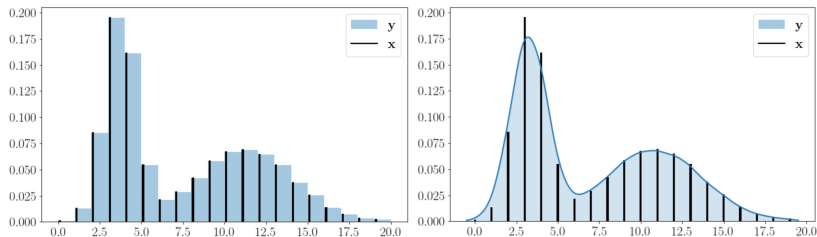
$$P(\mathbf{x}|\boldsymbol{\theta}) = \int_{U[0,1]} p(\mathbf{x} + \mathbf{u}|\boldsymbol{\theta}) d\mathbf{u}$$

## Proof

$$\begin{aligned}\mathbb{E}_{\pi} \log p(\mathbf{y}|\boldsymbol{\theta}) &= \int \pi(\mathbf{y}) \log p(\mathbf{y}|\boldsymbol{\theta}) d\mathbf{y} = \\ &= \sum \pi(\mathbf{x}) \int_{U[0,1]} \log p(\mathbf{x} + \mathbf{u}|\boldsymbol{\theta}) d\mathbf{u} \leq \\ &\leq \sum \pi(\mathbf{x}) \log \int_{U[0,1]} p(\mathbf{x} + \mathbf{u}|\boldsymbol{\theta}) d\mathbf{u} = \\ &= \sum \pi(\mathbf{x}) \log P(\mathbf{x}|\boldsymbol{\theta}) = \mathbb{E}_{\pi} \log P(\mathbf{x}|\boldsymbol{\theta}).\end{aligned}$$



# Variational dequantization



- ▶  $p(\mathbf{y}|\boldsymbol{\theta})$  assign uniform density to unit hypercubes  $\mathbf{x} + U[0, 1]$  (left fig).
- ▶ Neural network density models are smooth function approximators (right fig).
- ▶ Smooth dequantization is more natural.

How to perform the smooth dequantization?

## Variational dequantization

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

## Variational lower bound

$$\begin{aligned}\log P(\mathbf{x}|\boldsymbol{\theta}) &= \left[ \log \int q(\mathbf{u}|\mathbf{x}) \frac{p(\mathbf{x} + \mathbf{u}|\boldsymbol{\theta})}{q(\mathbf{u}|\mathbf{x})} d\mathbf{u} \right] \geq \\ &\geq \int q(\mathbf{u}|\mathbf{x}) \log \frac{p(\mathbf{x} + \mathbf{u}|\boldsymbol{\theta})}{q(\mathbf{u}|\mathbf{x})} d\mathbf{u} = \mathcal{L}(q, \boldsymbol{\theta}).\end{aligned}$$

## Uniform dequantization bound

$$\log P(\mathbf{x}|\boldsymbol{\theta}) = \log \int_{U[0,1]} p(\mathbf{x} + \mathbf{u}|\boldsymbol{\theta}) d\mathbf{u} \geq \int_{U[0,1]} \log p(\mathbf{x} + \mathbf{u}|\boldsymbol{\theta}) d\mathbf{u}.$$

Uniform dequantization is a special case of variational dequantization ( $q(\mathbf{u}|\mathbf{x}) = U[0, 1]$ ).

# Flow++

## Variational lower bound

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

Let  $\mathbf{u} = h(\epsilon, \phi)$  is a flow model with base distribution  $\epsilon \sim p(\epsilon) = \mathcal{N}(0, \mathbf{I})$ :

$$q(\mathbf{u}|\mathbf{x}) = p(h^{-1}(\mathbf{u}, \phi)) \cdot \left| \det \frac{\partial h^{-1}(\mathbf{u}, \phi)}{\partial \mathbf{u}} \right|.$$

## Flow-based variational dequantization

$$\log P(\mathbf{x}|\theta) \geq \mathcal{L}(\phi, \theta) = \int p(\epsilon) \log \left( \frac{p(\mathbf{x} + h(\epsilon, \phi)|\theta)}{p(\epsilon) \cdot \left| \det \frac{\partial h(\epsilon, \phi)}{\partial \epsilon} \right|^{-1}} \right) d\epsilon.$$

If  $p(\mathbf{x} + \mathbf{u}|\theta)$  is also a flow model, it is straightforward to calculate stochastic gradient of this ELBO.

---

*Ho J. et al. Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design, 2019*

## Flow-based variational dequantization

$$\log P(\mathbf{x}|\boldsymbol{\theta}) \geq \int p(\boldsymbol{\epsilon}) \log \left( \frac{p(\mathbf{x} + h(\boldsymbol{\epsilon}, \phi))}{p(\boldsymbol{\epsilon}) \cdot \left| \det \frac{\partial h(\boldsymbol{\epsilon}, \phi)}{\partial \boldsymbol{\epsilon}} \right|^{-1}} \right) d\boldsymbol{\epsilon}.$$

*Table 1. Unconditional image modeling results in bits/dim*

| Model family       | Model                                       | CIFAR10     | ImageNet 32x32 | ImageNet 64x64 |
|--------------------|---|-------------|----------------|----------------|
| Non-autoregressive | RealNVP (Dinh et al., 2016)                 | 3.49        | 4.28           | –              |
|                    | Glow (Kingma & Dhariwal, 2018)              | 3.35        | 4.09           | 3.81           |
|                    | IAF-VAE (Kingma et al., 2016)               | 3.11        | –              | –              |
|                    | <b>Flow++ (ours)</b>                        | <b>3.08</b> | <b>3.86</b>    | <b>3.69</b>    |
|                    |   |             |                |                |
| Autoregressive     | Multiscale PixelCNN (Reed et al., 2017)     | –           | 3.95           | 3.70           |
|                    | PixelCNN (van den Oord et al., 2016b)       | 3.14        | –              | –              |
|                    | PixelRNN (van den Oord et al., 2016b)       | 3.00        | 3.86           | 3.63           |
|                    | Gated PixelCNN (van den Oord et al., 2016c) | 3.03        | 3.83           | 3.57           |
|                    | PixelCNN++ (Salimans et al., 2017)          | 2.92        | –              | –              |
|                    | Image Transformer (Parmar et al., 2018)     | 2.90        | 3.77           | –              |
|                    | PixelSNAIL (Chen et al., 2017)              | 2.85        | 3.80           | 3.52           |

# 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.

## Generative process

- ▶  $\pi(\mathbf{x}|\mathbf{v}, \mathbf{w}) = \text{Sim}(\mathbf{v}, \mathbf{w})$  – true world simulator;
- ▶  $\mathbf{v}$  – conditionally independent factors:  $\pi(\mathbf{v}|\mathbf{x}) = \prod_{j=1}^d \pi(v_j|\mathbf{x})$ ;
- ▶  $\mathbf{w}$  – conditionally dependent factors.

## Unsupervised generative model

$$p(\mathbf{x}|\mathbf{z}, \theta) \approx \pi(\mathbf{x}|\mathbf{v}, \mathbf{w}).$$

The latent factors  $q(\mathbf{z}|\mathbf{x})$  capture the factors  $\mathbf{v}$  in a disentangled manner. The conditionally dependent factors  $\mathbf{w}$  remains entangled in a subset of  $\mathbf{z}$  that is not used for representing  $\mathbf{v}$ .

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

# $\beta$ -VAE

## 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

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

## Hypothesis

We are able to learn disentangled representations of the independent factors  $\mathbf{v}$  by setting a stronger constraint with  $\beta > 1$ .

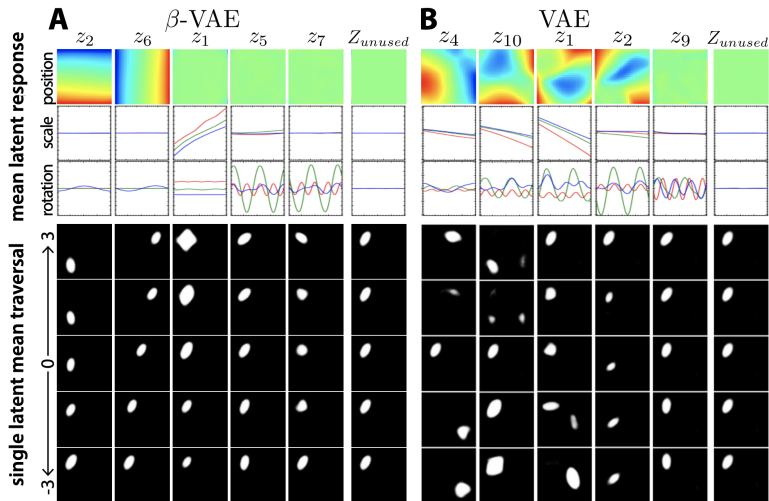
**Note:** It leads to poorer reconstructions and a loss of high frequency details.

---

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



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





## 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

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

## Minimization of MI

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

# DIP-VAE

## Disentangled aggregated variational posterior

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

## DIP-VAE Objective

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

# DIP-VAE

$$\mathcal{L}_{\text{DIP}}(q, \theta) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}_i(q, \theta) - \lambda \cdot \underbrace{KL(q(\mathbf{z}) || p(\mathbf{z}))}_{\text{intractable}}$$

Let match the moments of  $q(\mathbf{z})$  and  $p(\mathbf{z})$ :

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

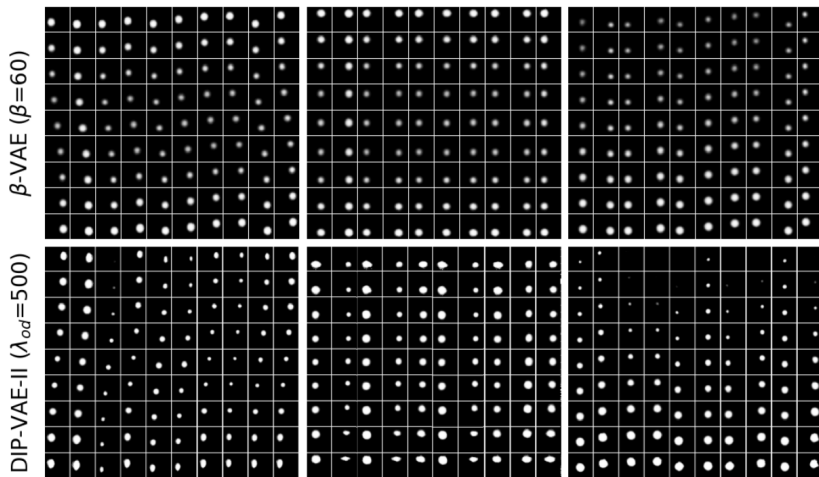
DIP-VAE regularizes  $\text{cov}_{q(\mathbf{z})}(\mathbf{z})$  to be close to the identity matrix.

## Objective

$$\max_{q, \theta} \left[ \frac{1}{n} \sum_{i=1}^n \mathcal{L}_i(q, \theta) - \lambda_1 \sum_{i \neq j} [\text{cov}_{q(\mathbf{z})}(\mathbf{z})]_{ij}^2 - \lambda_2 \sum_i \left( [\text{cov}_{q(\mathbf{z})}(\mathbf{z})]_{ii} - 1 \right)^2 \right]$$

# DIP-VAE

Reconstructions become better.



---

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

# Challenging Disentanglement Assumptions

## Theorem

Let  $\mathbf{z} \sim P$  with a 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  $\mathbf{u} \in \text{supp}(\mathbf{z})$ .

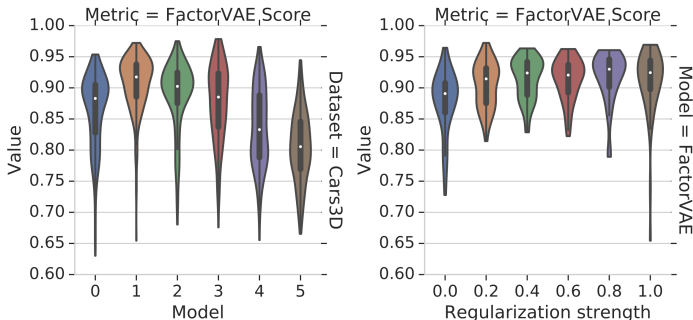
Consider a generative model with disentangled representation  $\mathbf{z}$ .

- ▶  $\exists \hat{\mathbf{z}} = f(\mathbf{z})$  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.

# Challenging Disentanglement Assumptions



Dataset = Noisy-dSprites

|                         |     |     |     |     |     |     |
|-------------------------|-----|-----|-----|-----|-----|-----|
| BetaVAE Score (A)       | 100 | 80  | 44  | 41  | 46  | 37  |
| FactorVAE Score (B)     | 80  | 100 | 49  | 52  | 25  | 38  |
| MIG (C)                 | 44  | 49  | 100 | 76  | 6   | 42  |
| DCI Disentanglement (D) | 41  | 52  | 76  | 100 | -8  | 38  |
| Modularity (E)          | 46  | 25  | 6   | -8  | 100 | 13  |
| SAP (F)                 | 37  | 38  | 42  | 38  | 13  | 100 |
|                         | (A) | (B) | (C) | (D) | (E) | (F) |

Locatello F. et al. *Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations*, 2018

# Summary

- ▶ Dequantization allows to fit discrete data using continuous model.
- ▶ Uniform dequantization is the simplest form of dequantization. Variational dequantization is a more natural type that was proposed in Flow++ model.
- ▶ 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.