Deep Generative Models for Signal Processing and Beyond

David Wipf

Microsoft Research

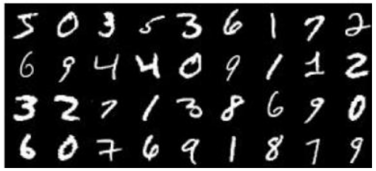
Note: Updated version of slides available at http://www.davidwipf.com/

Part I:
Introduction

Generative Models

- Given: Training data

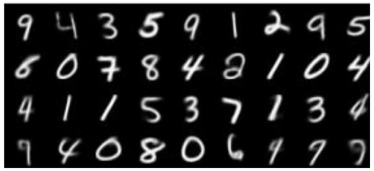
Example: MNIST digits



$$\left\{\mathbf{x}^{(i)}\right\}_{i=1}^{n}, \quad \mathbf{x}^{(i)} \sim p_{gt}\left(\mathbf{x}\right)$$

- Goal: Learn a parametric model capable of producing new samples

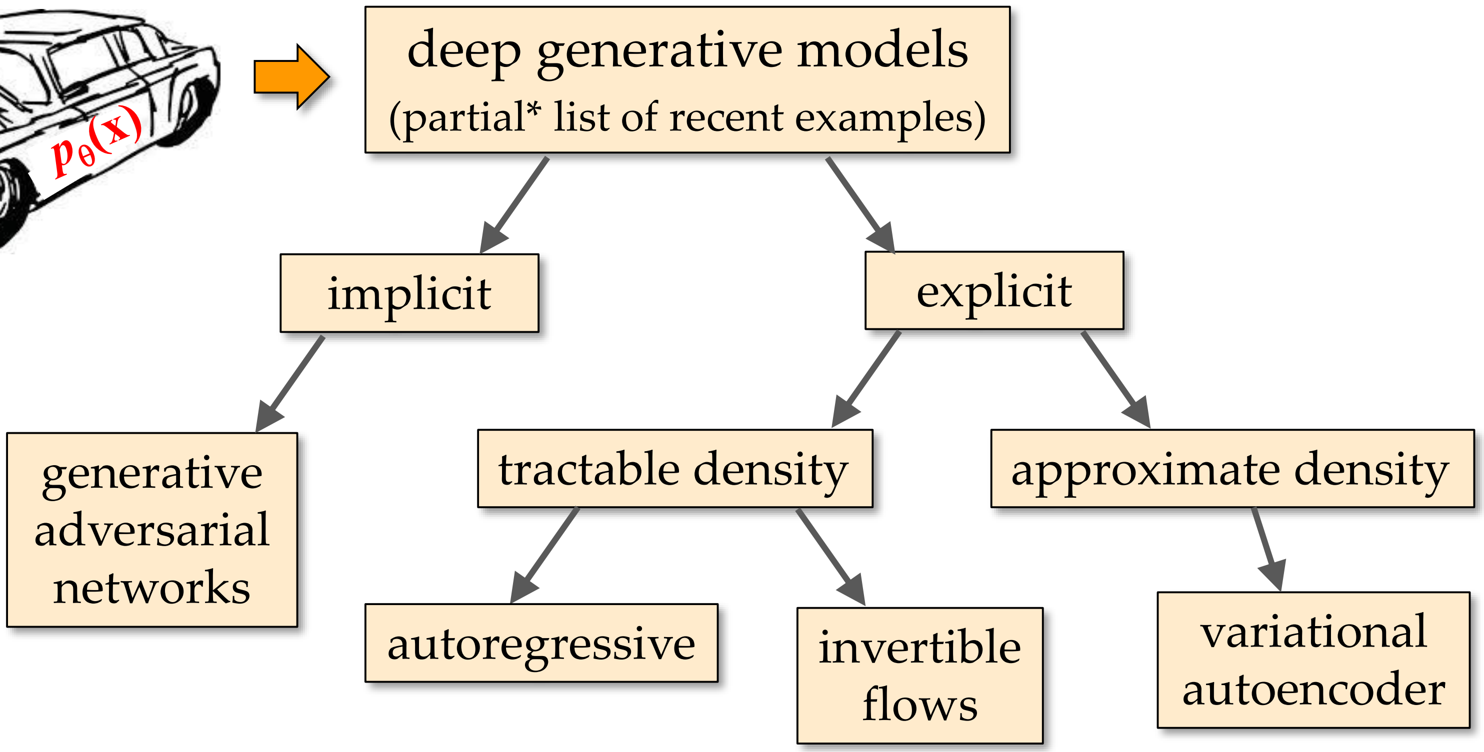
Similar to real data



$$\left\{\mathbf{x}^{(j)}\right\}_{j=1}^{m}, \quad \mathbf{x}^{(j)} \sim p_{\boldsymbol{\theta}}\left(\mathbf{x}\right) \approx p_{gt}\left(\mathbf{x}\right)$$

- Deep generative models use neural networks for implicitly or explicitly defining the density $p_{\boldsymbol{\theta}}\left(\mathbf{x}\right)$

Types of Deep Generative Models



*For additional examples, see tutorial [Goodfellow, 2016]

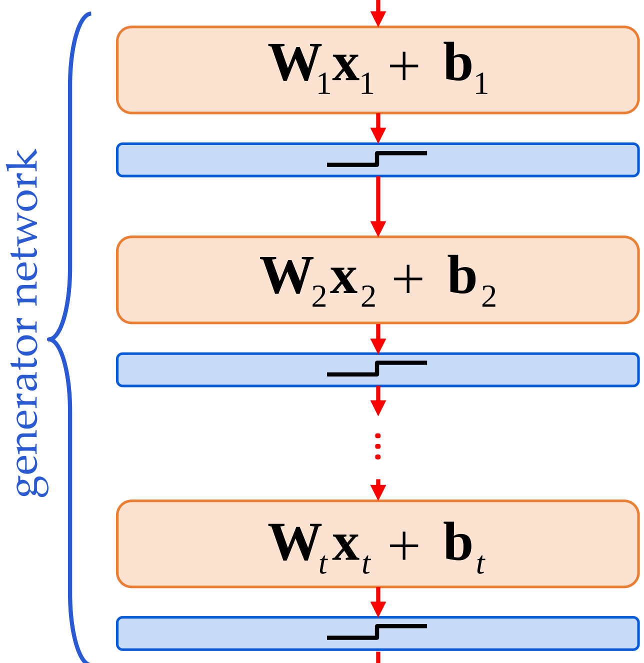
Implicit Deep Generative Models

- deep generative models (partial* list of recent examples)
  - implicit
    - generative adversarial networks
  - explicit
    - tractable density
      - autoregressive
      - invertible flows
    - approximate density
      - variational autoencoder

*For additional examples, see tutorial [Goodfellow, 2016]

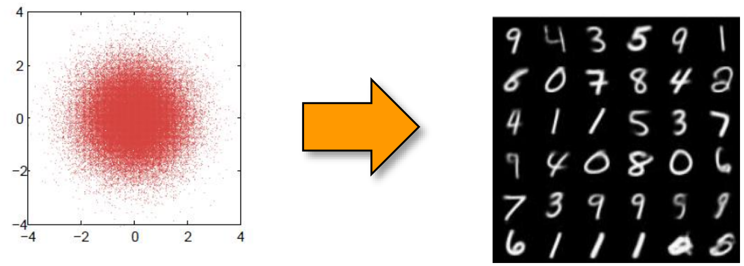
Implicit Deep Generative Modeling

$$\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$$ easy to sample



$$\mathbf{x} \sim \underbrace{p_{\boldsymbol{\theta}}(\mathbf{x})}_{\text{explicit form generally not feasible to compute}} \approx p_{gt}(\mathbf{x})$$

Example



Popular Example

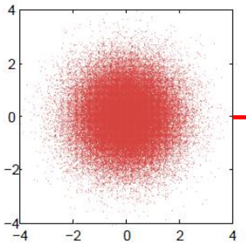
Generative Adversarial Networks:

- ❑ Based on game theory, Nash equilibrium
- ❑ 8679 citations [Goodfellow et al., 2014]

Generative Adversarial Networks (GANs)

noise samples

$\left\{\mathbf{z}^{(j)}\right\}_{j=1}^{m}, \ \mathbf{z}^{(j)} \sim N\left(\mathbf{z} \mid \mathbf{0}, \mathbf{I}\right)$



Generator Network

$G_{\boldsymbol{\theta}}$

fake samples

real samples

$\left\{\mathbf{x}^{(i)}\right\}_{i=1}^{n}, \mathbf{x}^{(i)} \in \mathbb{R}^{d} \sim p_{gt}\left(\mathbf{x}\right)$

Discriminator Network

real/fake

$D_{\boldsymbol{\varphi}} : \mathbb{R}^{d} \rightarrow (0,1)$

Binary classification:

$D_{\boldsymbol{\varphi}}\big(\underbrace{\mathbf{x}^{(i)}}_{\text{real}}\big) \approx 1, \qquad D_{\boldsymbol{\varphi}}\Big[\underbrace{G_{\boldsymbol{\theta}}\big(\mathbf{z}^{(j)}\big)}_{\text{fake}}\Big] \approx 0$

Basic GAN objective (cross-entropy-based):

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\varphi}} \ \mathrm{E}_{p_{gt}(\mathbf{x})}\left[\log D_{\boldsymbol{\varphi}}\left(\mathbf{x}\right)\right] + \mathrm{E}_{N(\mathbf{z}|\mathbf{0},\mathbf{I})}\left[\log\left(1 - D_{\boldsymbol{\varphi}}\left[G_{\boldsymbol{\theta}}\left(\mathbf{z}\right)\right]\right)\right]$$

expectations approximated with samples

[Goodfellow et al., 2014]

GAN Strengths

State-of-the-art GAN models generate highly realistic samples, e.g., StyleGAN [Karras et al, 2019]:

real

fake

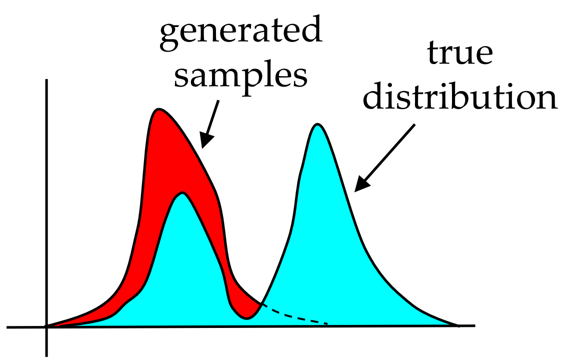
Examples from http://www.whichfaceisreal.com/

GAN Weaknesses

- Training involves potentially unstable minimax problem, iterations may diverge, be sensitive to tuning.

  [Lucic et al., 2018]

- Can be susceptible to mode collapse:



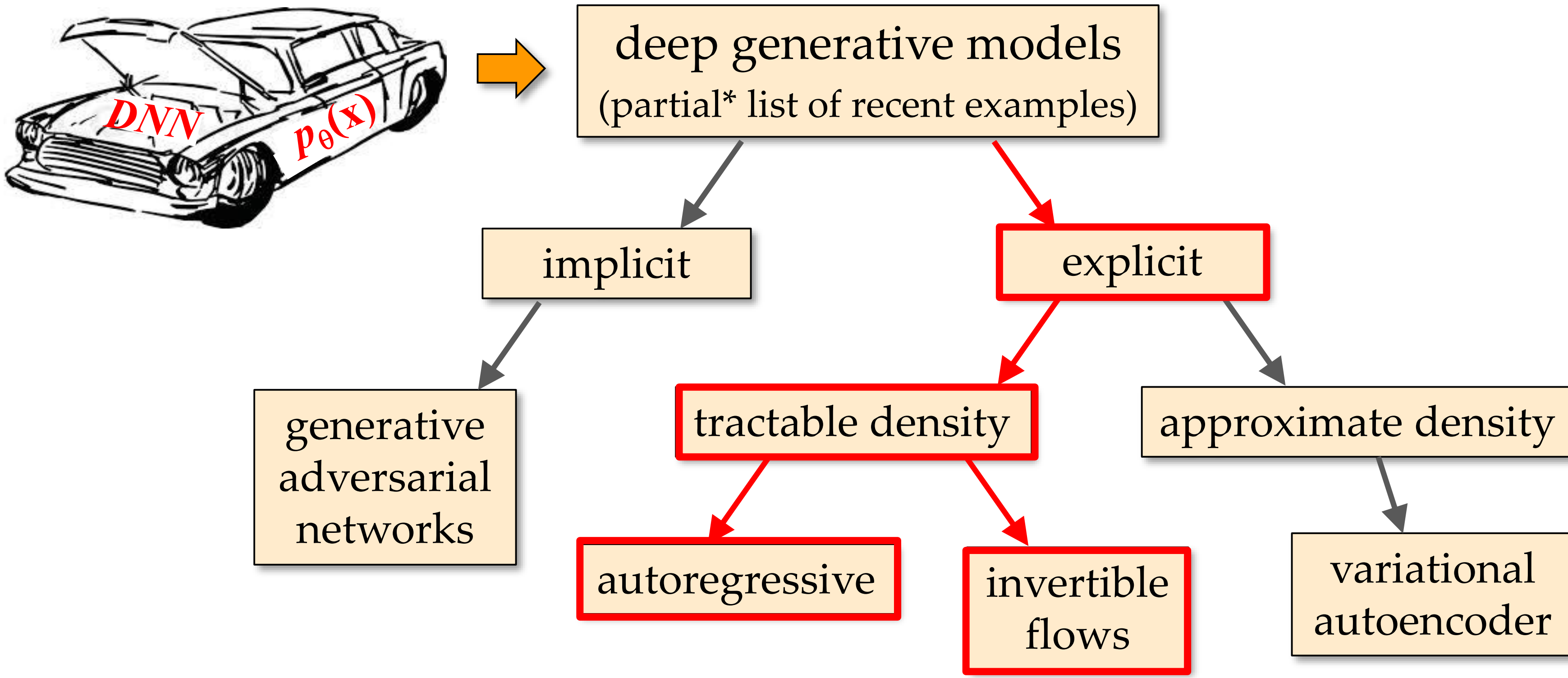
low sample diversity

[Arora and Zhang, 2017]

- No explicit density estimate $p_{\boldsymbol{\theta}}(\mathbf{x}) \approx p_{gt}(\mathbf{x})$, cannot infer the latent code that produced a sample:

$$\underbrace{p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x})}_{\text{cannot compute low-dimensional representation}} \;?$$

Explicit Deep Generative Modeling w/ a Tractable Density



*For additional examples, see tutorial [Goodfellow, 2016]

Explicit Deep Generative Modeling w/ a Tractable Density

- Density $p_{\boldsymbol{\theta}}(\mathbf{x})$ and gradients $\nabla p_{\boldsymbol{\theta}}(\mathbf{x})$ can be computed exactly

- Given training data $\left\{\mathbf{x}^{(i)}\right\}_{i=1}^{n}$, $\mathbf{x}^{(i)} \in \mathbb{R}^d$, can solve via SGD:

$$\boldsymbol{\theta}_* = \arg\min_{\boldsymbol{\theta}} -\sum_i \log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}\right)$$

maximum likelihood estimator

- **Key advantage:** Closed-form test data likelihood $p_{\boldsymbol{\theta}_*}\left(\mathbf{x}^{test}\right)$

- **Disadvantages:**
  - Generated samples arguably inferior to GANs
  - No dimensionality reduction, representation learning (mostly)

Examples

- **Autoregressive methods:**

  Apply chain rule to form:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \prod_{j=1}^{d} p_{\boldsymbol{\theta}}\left(x_j \mid x_1, \ldots, x_{j-1}\right) \quad \Bigg\} \text{ conditionals parameterized as RNN or CNN}$$

[Larochelle & Murray, 2011; van den Oord et al., 2016]

- **Invertible flows:**

  Assumptions:

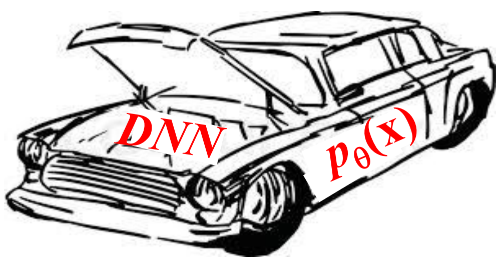
$$p(\mathbf{z}) = N(\mathbf{z} \mid \mathbf{0}, \mathbf{I}), \quad \dim(\mathbf{z}) = \dim(\mathbf{x}), \quad \mathbf{z} = f_{\boldsymbol{\theta}}(\mathbf{x}), \quad \mathbf{x} = f_{\boldsymbol{\theta}}^{-1}(\mathbf{z})$$

  Change of variables formula:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = N(\mathbf{z} \mid \mathbf{0}, \mathbf{I}) \left| \det\left(\frac{\partial f_{\boldsymbol{\theta}}(\mathbf{x})}{\partial \mathbf{x}^T} \right) \right| \quad \Bigg\} \text{ tractable determinant because of special DNN structure}$$

[Dinh et al., 2016; Kingma & Dhariwal, 2018]

Explicit Deep Generative Modeling Using a Density Approximation/Bound



- deep generative models (partial* list of recent examples)
  - implicit
    - generative adversarial networks
  - explicit
    - tractable form
      - autoregressive
      - invertible flows
    - approximate density
      - variational autoencoder

this talk (mostly)

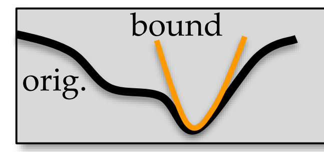
*For additional examples, see tutorial [Goodfellow, 2016]

Explicit Deep Generative Modeling Using a Density Approximation/Bound

- Often interested in densities of the form:

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\, \underbrace{p(\mathbf{z})}_{\text{low-dimensional latent factors}}\, d\mathbf{z}$$

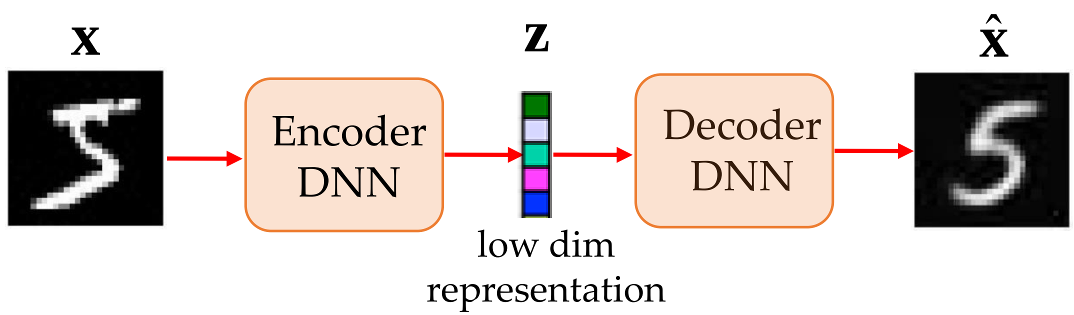
- Required integral is intractable … Optimize upper bound on $-\sum_i \log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}\right)$



- Popular example: The variational autoencoder (VAE)

$\underbrace{\text{[Kingma and Welling, 2014; Rezende et al., 2014]}}_{\text{6079 citations}}$

Upper bound based on autoencoder-like structure



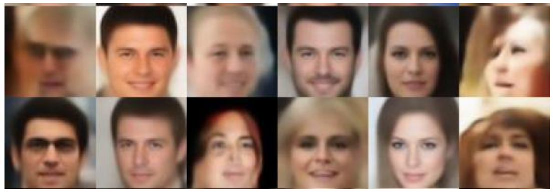
Variational Autoencoders
(details in Part II)

Advantages:

- Less prone to mode collapse than GANs, more stable training.
- Provides explicit estimate of latent distribution $p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{x})$; many applications in representation learning.
- Natural generalization of dimensionality reduction tools in common use for signal processing (Part III).

Disadvantages:

- Optimizes a bound on the data likelihood, not exact likelihood (but conditions for when bound is tight discussed in Part IV).
- Generated samples usually inferior to GANs ...

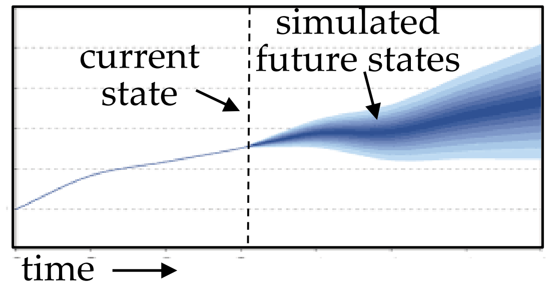


… although improvements possible (Part IV).

Representative Applications

Generative models in general:

- Model-based reinforcement learning:



[Finn et al., 2016]

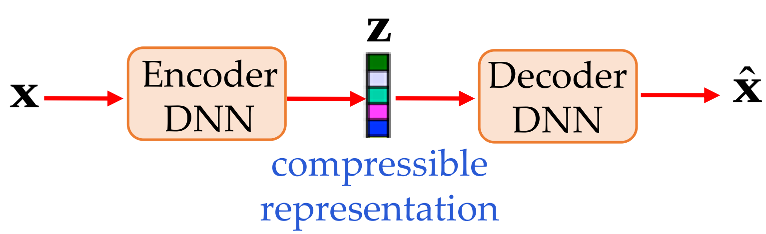
- Image-to-image translation:

[Isola et al., 2016]

- Many more, a generic unsupervised learning tool

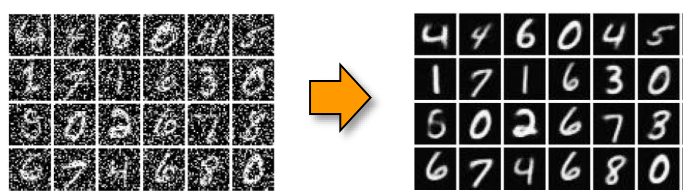
VAEs in particular:

- Compression:



[Ballé et al., 2018]

- Data cleaning, outlier removal:



[Dai et al., 2018]

Caveat

- Deep generative modeling is a rapidly changing field.
- Strengths and weaknesses of various methods frequently need recalibration in accordance with new developments.
- Also, important to differentiate:

1) General-purpose improvements in DNN architectures
2) Advances in specific generative modeling paradigms

Remainder of Tutorial

- Part II: Details of the variational autoencoder
- Part III: Connections with existing signal processing methods for finding low-dimensional structure in data
- Part IV: From signal reconstruction to generative modeling
- Part V: Practical usage issues and examples

Questions?

Part II:
Details of the Variational Autoencoder

Note: Updated version of slides available at http://www.davidwipf.com/

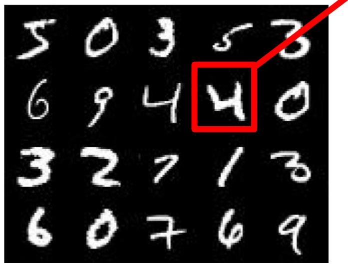
Latent Variable Model

Observed data: $\mathbf{X} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{n}, \quad \mathbf{x}^{(i)} \in \mathbb{R}^{d}, \ \forall i$

Assumed latent factors:

$$\mathbf{Z} = \left\{\mathbf{z}^{(i)}\right\}_{i=1}^{n}, \quad \underbrace{\mathbf{z}^{(i)}}_{\substack{\text{low-dimensional}\\ \text{representation of significant}\\ \text{factors of variation}}} \in \mathbb{R}^{\kappa}, \ \forall i, \quad \kappa \ll d$$

Example



28x28 = 784 dim MNIST digit

candidate latent factors:

digit type, stroke width, slant angle, etc.

$$\underbrace{\kappa < 20}_{\text{sufficient in practice}} \ll d = 784$$

Ground-truth generative process:

$$\mathbf{z}^{(i)} \sim \underbrace{p_{gt}(\mathbf{z})}_{\substack{\text{prior on}\\ \text{latent factors}\\ \text{of variation}}}, \qquad \mathbf{x}^{(i)} \sim \underbrace{p_{gt}\left(\mathbf{x} \mid \mathbf{z}^{(i)}\right)}_{\substack{\text{mapping to}\\ \text{observation space;}\\ \text{could be delta function}}}$$

$$\Rightarrow \quad \mathbf{x}^{(i)} \sim p_{gt}(\mathbf{x}) = \int p_{gt}(\mathbf{x} \mid \mathbf{z})\, p_{gt}(\mathbf{z})\, d\mathbf{z}$$

Parameterized Latent-Variable Model

Without loss of generality, assume: $\quad p_{gt}(\mathbf{z}) = N(\mathbf{z} \mid \mathbf{0}, \mathbf{I})$

Also assume parameterized family:

$$p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}), \quad \boldsymbol{\theta} \in \Omega$$

$$\text{s.t.} \quad p_{\boldsymbol{\theta}_*}(\mathbf{x} \mid \mathbf{z}) \approx p_{gt}(\mathbf{x} \mid \mathbf{z}), \quad \text{for some } \boldsymbol{\theta}_* \in \Omega$$

High-level goal:

Given $\quad \mathbf{X} = \left\{ \mathbf{x}^{(i)} \right\}_{i=1}^{n}, \quad \mathbf{x}^{(i)} \sim p_{gt}(\mathbf{x})$

Solve $\quad \underbrace{\min_{\boldsymbol{\theta}} \; -\sum_{i=1}^{n} \log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}\right)}_{\text{equivalent to maximum likelihood}} \equiv \min_{\boldsymbol{\theta}} \; -\sum_{i=1}^{n} \log \int p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)} \mid \mathbf{z}\right) N(\mathbf{z} \mid \mathbf{0}, \mathbf{I})\, d\mathbf{z}$

Key problem:

$$p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}\right) = \int p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)} \mid \mathbf{z}\right) N(\mathbf{z} \mid \mathbf{0}, \mathbf{I})\, d\mathbf{z}$$

$$p_{\boldsymbol{\theta}}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right) = p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)} \mid \mathbf{z}\right) N(\mathbf{z} \mid \mathbf{0}, \mathbf{I}) / p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}\right)$$

$\Rightarrow$ <u>intractable</u>

Naïve Approximation

Finite-sample approximation to intractable integral for each i:

$$\text{sample} \quad \mathbf{z}^{(i,j)} \sim N(\mathbf{z} \mid \mathbf{0}, \mathbf{I}), \quad j = 1, \ldots, m$$

$$\Rightarrow \quad \int p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)} \mid \mathbf{z}\right) N(\mathbf{z} \mid \mathbf{0}, \mathbf{I})\, dz \equiv \mathrm{E}_{N(\mathbf{z} \mid \mathbf{0}, \mathbf{I})}\left[p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)} \mid \mathbf{z}\right)\right] \approx \tfrac{1}{m} \sum_{j=1}^{m} p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)} \mid \mathbf{z}^{(i,j)}\right)$$

Revised tractable objective:

$$\min_{\boldsymbol{\theta}} \; -\sum_{i=1}^{n} \log\left[\tfrac{1}{m} \sum_{j=1}^{m} p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)} \mid \mathbf{z}^{(i,j)}\right)\right]$$

Lingering problem:

$$\text{for most} \quad \mathbf{z}^{(i,j)} \sim N(\mathbf{z} \mid \mathbf{0}, \mathbf{I}) \quad \Rightarrow \quad p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)} \mid \mathbf{z}^{(i,j)}\right) \approx 0$$

Need <u>huge</u> number of samples for reasonable approximation …

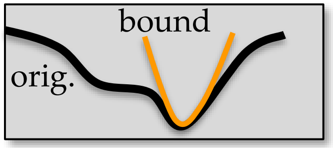
[Doersch, 2016]

A Useful Variational Bound

Define an approximate distribution as

$$\underbrace{q_{\varphi}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right)}_{\text{tractable}} \approx p_{\theta}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right) = p_{\theta}\left(\mathbf{x}^{(i)} \mid \mathbf{z}\right) N\left(\mathbf{z} \mid \mathbf{0},\mathbf{I}\right) / \underbrace{p_{\theta}\left(\mathbf{x}^{(i)}\right)}_{\text{intractable}}$$

Variational upper bound:



$$-\sum_i \log p_{\theta}\left(\mathbf{x}^{(i)}\right) \leq L(\theta,\varphi) \triangleq \sum_i \left\{ \underbrace{\mathrm{KL}\left[q_{\varphi}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right) \| \, p_{\theta}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right)\right]}_{\geq \; 0} - \log p_{\theta}\left(\mathbf{x}^{(i)}\right) \right\}$$

After standard manipulations …

$$L(\theta,\varphi) \equiv \sum_i \left\{ \underbrace{\mathrm{KL}\left[q_{\varphi}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right) \| \, N\left(\mathbf{z} \mid \mathbf{0},\mathbf{I}\right)\right] - \mathrm{E}_{q_{\varphi}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right)}\left[\log p_{\theta}\left(\mathbf{x}^{(i)} \mid \mathbf{z}\right)\right]} \right\}$$

Does <u>not</u> depend on intractable $p_{\theta}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right)$ or $p_{\theta}\left(\mathbf{x}^{(i)}\right)$

Basic VAE Energy Function Decomposition

$$L(\boldsymbol{\theta},\boldsymbol{\varphi}) = \boxed{\overset{\text{regularization factor}}{\sum_i \Big\{ \mathrm{KL}\left[q_{\boldsymbol{\varphi}}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right) \| N\left(\mathbf{z} \mid \mathbf{0},\mathbf{I}\right)\right]}} - \overset{\text{data-fit term}}{\mathrm{E}_{q_{\boldsymbol{\varphi}}\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)} \mid \mathbf{z}\right)\right]} \Big\}$$

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

Handling the Regularization Term

$\mathrm{KL}\left[q_{\varphi}\left(\mathbf{z}\,|\,\mathbf{x}^{(i)}\right) \| N\left(\mathbf{z}\,|\,\mathbf{0},\mathbf{I}\right)\right]$ is still intractable in general

Simplifying Gaussian approximate posterior assumption:

$$q_{\varphi}\left(\mathbf{z}\,|\,\mathbf{x}^{(i)}\right) = N\left(\mathbf{z}\,|\,\boldsymbol{\mu}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right],\ \boldsymbol{\Sigma}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]\right) \quad \Big\} \ \text{encoder distribution}$$

Encoder moments computed by deep networks:

$$\mathbf{x}^{(i)} \rightarrow \boxed{\boldsymbol{\varphi}} \rightarrow \boldsymbol{\mu}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right] \qquad \mathbf{x}^{(i)} \rightarrow \boxed{\boldsymbol{\varphi}} \rightarrow \boldsymbol{\Sigma}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]$$

KL term now satisfies:

$$2\,\mathrm{KL}\left[q_{\varphi}\left(\mathbf{z}\,|\,\mathbf{x}^{(i)}\right) \| N\left(\mathbf{z}\,|\,\mathbf{0},\mathbf{I}\right)\right] \equiv \left\|\boldsymbol{\mu}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]\right\|_2^2 + \mathrm{tr}\left(\boldsymbol{\Sigma}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]\right) - \log\left|\boldsymbol{\Sigma}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]\right|$$

Differentiable, suitable for minimization via SGD

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

Basic VAE Energy Function Decomposition

$$L(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \sum_i \Big\{ \underbrace{\mathrm{KL}\left[q_{\boldsymbol{\varphi}}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right) \| N\left(\mathbf{z} \mid \mathbf{0}, \mathbf{I}\right)\right]}_{\text{regularization factor}} \boxed{\underbrace{- \mathrm{E}_{q_{\boldsymbol{\varphi}}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right)}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)} \mid \mathbf{z}\right)\right]}_{\text{data-fit term}}} \Big\}$$

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

Handling the Data-Fit Term

$-\mathrm{E}_{q_{\boldsymbol{\varphi}}\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)} \mid \mathbf{z}\right)\right]$ is also generally intractable

For continuous data, typical assumption is

$$\left. p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)} \mid \mathbf{z}\right) = N\left(\mathbf{x}^{(i)} \mid \boldsymbol{\mu}_{\mathbf{x}}\left[\mathbf{z},\boldsymbol{\theta}\right],\ \boldsymbol{\Sigma}_{\mathbf{x}}\left[\mathbf{z},\boldsymbol{\theta}\right] \right) \right\} \text{decoder distribution}$$

Decoder moments computed by deep networks:

$$\mathbf{z} \to \boxed{\boldsymbol{\theta}} \to \boldsymbol{\mu}_{\mathbf{x}}\left[\mathbf{z},\boldsymbol{\theta}\right] \qquad \underbrace{\mathbf{z} \to \boxed{\boldsymbol{\theta}} \to \boldsymbol{\Sigma}_{\mathbf{x}}\left[\mathbf{z},\boldsymbol{\theta}\right]}$$

simplified version $\Rightarrow$ $\boldsymbol{\Sigma}_{\mathbf{x}}\left[\mathbf{z},\boldsymbol{\theta}\right] = \gamma \mathbf{I}, \quad \forall \mathbf{z}$

But …

$$-\mathrm{E}_{q_{\boldsymbol{\varphi}}\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)} \mid \mathbf{z}\right)\right] \equiv \mathrm{E}_{q_{\boldsymbol{\varphi}}\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}\left[\tfrac{1}{2\gamma}\left\|\mathbf{x}^{(i)} - \boldsymbol{\mu}_{\mathbf{x}}\left[\mathbf{z},\boldsymbol{\theta}\right]\right\|_2^2\right] \Rightarrow \text{still intractable}$$

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

Revisiting Finite-Sample Approximations

From before: $\mathbf{z}^{(i,j)} \sim N(\mathbf{z} \mid \mathbf{0}, \mathbf{I}), \quad j = 1, \ldots, m$

$$E_{N(\mathbf{z}|\mathbf{0},\mathbf{I})}\left[p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)} \mid \mathbf{z}\right)\right] \not\approx \frac{1}{m}\sum_{j=1}^{m} p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)} \mid \mathbf{z}^{(i,j)}\right) \Rightarrow$$

bad approximation unless m is huge

But what about the present situation: $\mathbf{z}^{(i,j)} \sim q_{\boldsymbol{\varphi}}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right), \quad j = 1, \ldots, m$

$$E_{q_{\boldsymbol{\varphi}}\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}\left[\frac{1}{2\gamma}\left\| \mathbf{x}^{(i)} - \boldsymbol{\mu}_{\mathbf{x}}\left[\mathbf{z}, \boldsymbol{\theta}\right]\right\|_2^2 \right] \overset{?}{\Leftrightarrow} \frac{1}{m}\sum_{j=1}^{m} \frac{1}{2\gamma}\left\| \mathbf{x}^{(i)} - \boldsymbol{\mu}_{\mathbf{x}}\left[\mathbf{z}^{(i,j)}, \boldsymbol{\theta}\right]\right\|_2^2$$

Unlike the prior $N(\mathbf{z} \mid \mathbf{0}, \mathbf{I})$, during training the encoder $q_{\boldsymbol{\varphi}}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right)$:

- Confines mass to narrow region of **z**-space
- *Excludes* regions that are unlikely to have produced $\mathbf{x}^{(i)}$

} much better for sampling

In practice, can use just <u>$m = 1$</u> sample at each training iteration:

$$\mathbf{z}^{(i)} \sim q_{\boldsymbol{\varphi}}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right), \quad E_{q_{\boldsymbol{\varphi}}\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}\left[\frac{1}{2\gamma}\left\| \mathbf{x}^{(i)} - \boldsymbol{\mu}_{\mathbf{x}}\left[\mathbf{z}, \boldsymbol{\theta}\right]\right\|_2^2 \right] \approx \underbrace{\frac{1}{2\gamma}\left\| \mathbf{x}^{(i)} - \boldsymbol{\mu}_{\mathbf{x}}\left[\mathbf{z}^{(i)}, \boldsymbol{\theta}\right]\right\|_2^2}_{\text{unbiased estimator}}$$

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

Reparameterization Trick

Data-term approximation:

$$\tfrac{1}{2\gamma}\left\|\mathbf{x}^{(i)}-\boldsymbol{\mu}_{\mathbf{x}}\left[\mathbf{z}^{(i)},\boldsymbol{\theta}\right]\right\|_2^2 \Rightarrow$$

easy to minimize over $\boldsymbol{\theta}$ via SGD

But what about sampling operator $\mathbf{z}^{(i)} \sim q_{\boldsymbol{\varphi}}\left(\mathbf{z}\,|\,\mathbf{x}^{(i)}\right)$?

Problem: Cannot directly propagate gradients w.r.t. $\boldsymbol{\varphi}$ through sampling operator …

Equivalent sampling procedures:

$$\mathbf{z}^{(i)} \sim q_{\boldsymbol{\varphi}}\left(\mathbf{z}\,|\,\mathbf{x}^{(i)}\right) \quad \Leftrightarrow \quad \begin{aligned} &\boldsymbol{\varepsilon}^{(i)} \sim N\left(\boldsymbol{\varepsilon}\,|\,\mathbf{0},\mathbf{I}\right) \\ &\mathbf{z}^{(i)} = \boldsymbol{\mu}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]+\boldsymbol{\Sigma}_{\mathbf{z}}^{1/2}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]\boldsymbol{\varepsilon}^{(i)} \end{aligned}$$

Revised approximation:

$$\left.\begin{aligned} \mathrm{E}_{q_{\boldsymbol{\varphi}}\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}\left[\tfrac{1}{2\gamma}\left\|\mathbf{x}^{(i)}-\boldsymbol{\mu}_{\mathbf{x}}\left[\mathbf{z},\boldsymbol{\theta}\right]\right\|_2^2\right] &= \mathrm{E}_{N(\boldsymbol{\varepsilon}|\mathbf{0},\mathbf{I})}\left[\tfrac{1}{2\gamma}\left\|\mathbf{x}^{(i)}-\boldsymbol{\mu}_{\mathbf{x}}\left(\boldsymbol{\mu}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]+\boldsymbol{\Sigma}_{\mathbf{z}}^{1/2}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]\boldsymbol{\varepsilon},\boldsymbol{\theta}\right)\right\|_2^2\right] \\ &\approx \tfrac{1}{2\gamma}\left\|\mathbf{x}^{(i)}-\boldsymbol{\mu}_{\mathbf{x}}\left(\boxed{\boldsymbol{\mu}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]+\boldsymbol{\Sigma}_{\mathbf{z}}^{1/2}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]\boldsymbol{\varepsilon}^{(i)}},\boldsymbol{\theta}\right)\right\|_2^2 \end{aligned}\right\} \text{SGD friendly}$$

differentiable sample from encoder

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

VAE Optimization Summary

Basic energy function:

$$L(\boldsymbol{\theta},\boldsymbol{\varphi}) \equiv \sum_i \left\{ \mathrm{KL}\left[q_{\boldsymbol{\varphi}}\left(\mathbf{z}\,|\,\mathbf{x}^{(i)}\right) \| N\left(\mathbf{z}\,|\,\mathbf{0},\mathbf{I}\right)\right] - \mathrm{E}_{q_{\boldsymbol{\varphi}}\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}\,|\,\mathbf{z}\right)\right] \right\}$$

$$\geq -\sum_i \log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}\right)$$

Solve:

$$\left.\begin{aligned} \boldsymbol{\theta}_*, \boldsymbol{\varphi}_* = \underset{\boldsymbol{\theta},\boldsymbol{\varphi}}{\operatorname{argmin}} \quad & L(\boldsymbol{\theta},\boldsymbol{\varphi}) \\ \text{s.t.} \quad & q_{\boldsymbol{\varphi}}\left(\mathbf{z}\,|\,\mathbf{x}^{(i)}\right) = N\left(\mathbf{z}\,|\,\boldsymbol{\mu}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right],\ \boldsymbol{\Sigma}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]\right) \\ & p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}\,|\,\mathbf{z}\right) = N\left(\mathbf{x}^{(i)}\,|\,\boldsymbol{\mu}_{\mathbf{x}}[\mathbf{z},\boldsymbol{\theta}],\ \boldsymbol{\Sigma}_{\mathbf{x}}[\mathbf{z},\boldsymbol{\theta}]\right) \end{aligned}\right\}$$

approximate via
reparameterization
trick + SGD

Generating New Samples

Simple hierarchical sampling:

$$\begin{aligned} \mathbf{z}^{(j)} &\sim N\left(\mathbf{z} \mid \mathbf{0}, \mathbf{I}\right), \quad j = 1, \ldots, m \\ \mathbf{x}^{(j)} &\sim p_{\boldsymbol{\theta}_*}\left(\mathbf{x} \mid \mathbf{z}^{(j)}\right), \quad j = 1, \ldots, m \end{aligned}$$

only decoder with
optimized parameters
is needed

Typical to ignore decoder noise variance, i.e.,

replace $\mathbf{x}^{(j)} \sim p_{\boldsymbol{\theta}_*}\left(\mathbf{x} \mid \mathbf{z}^{(j)}\right)$ with $\mathbf{x}^{(j)} = \boldsymbol{\mu}_x\left(\mathbf{z}^{(j)}, \boldsymbol{\theta}_*\right)$ ⇨ cleaner samples

Ideal scenario:

new samples $\left\{\mathbf{x}^{(j)}\right\}_{j=1}^{m}$ ⇔ training data $\left\{\mathbf{x}^{(i)}\right\}_{i=1}^{n} \sim p_{gt}\left(\mathbf{x}\right)$

similar in
distribution

MNIST Examples

Ground-truth samples

VAE-generated samples
with $\boldsymbol{\Sigma}_{\mathbf{x}}\left[\mathbf{z},\boldsymbol{\theta}\right]=\mathbf{I}, \quad \forall \mathbf{z}$

(better VAE options available; Part IV)

Computing Negative Log-Likelihood (NLL) Estimates

Can apply unbiased estimate of VAE bound:

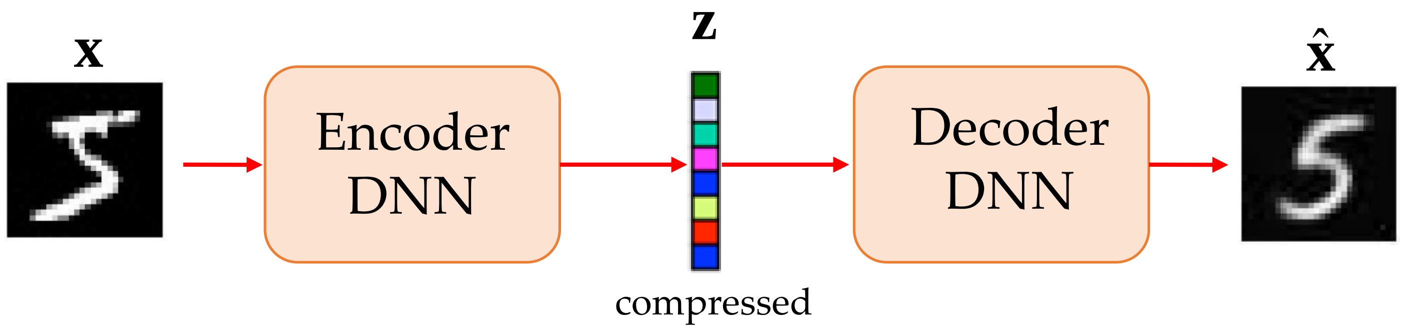
$$-\log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{test}\right) \le \underbrace{\mathrm{KL}\left[q_{\boldsymbol{\varphi}}\left(\mathbf{z} \mid \mathbf{x}^{test}\right) \| N\left(\mathbf{z} \mid \mathbf{0}, \mathbf{I}\right)\right]}_{\text{exact, closed-form}} - \underbrace{\mathrm{E}_{q_{\boldsymbol{\varphi}}\left(\mathbf{z} \mid \mathbf{x}^{test}\right)}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{test} \mid \mathbf{z}\right)\right]}_{\text{use unbiased estimate}}$$

$$\approx \frac{1}{m} \sum_{j=1}^{m} \frac{1}{2\gamma} \left\| \mathbf{x}^{test} - \boldsymbol{\mu}_{\mathbf{x}}\left[\mathbf{z}^{(j)}, \boldsymbol{\theta}\right] \right\|_2^2$$

$$\mathbf{z}^{(j)} \sim q_{\boldsymbol{\varphi}}\left(\mathbf{z} \mid \mathbf{x}^{test}\right)$$

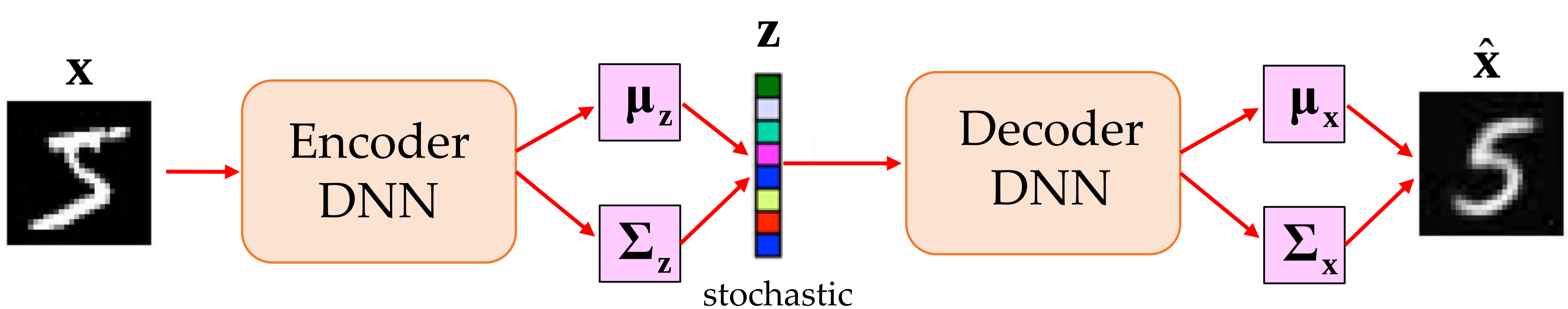
Comparison with an Autoencoder

Autoencoder (AE):



$\dim(\mathbf{z}) \ll \dim(\mathbf{x})$

VAE:



$\mathbf{z} \sim q_{\varphi}(\mathbf{z} \mid \mathbf{x}) = N(\boldsymbol{\mu}_{z}, \boldsymbol{\Sigma}_{z})$

$\hat{\mathbf{x}} \sim p_{\theta}(\mathbf{x} \mid \mathbf{z}) = N(\boldsymbol{\mu}_{x}, \boldsymbol{\Sigma}_{x})$

Questions?

Part III:
Connections with Existing Signal Processing Models for Finding Low-Dimensional Structure

Note: Updated version of slides available at http://www.davidwipf.com/

Outline

- Finding low-dimensional structure in high-dimensional data, possibly corrupted with outliers

- NP-hard decompositions into inlier and sparse outlier components

- Weaknesses of existing methods and useful VAE-based alternatives

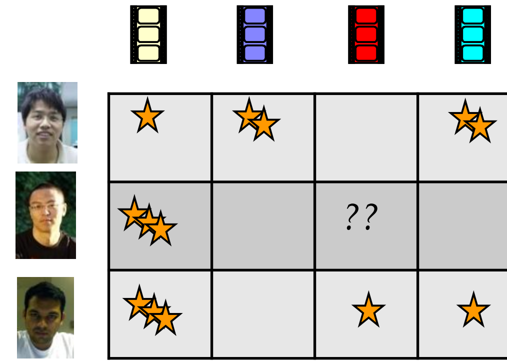
- Case Study: <u>Robust PCA</u> representative example
  - Connections with restricted class of VAE models
  - Advantages of the VAE in finding low-dimensional structure

Context

- Data is increasingly massive, high-dimensional

Images

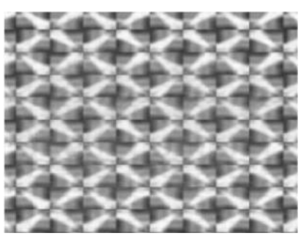
Videos

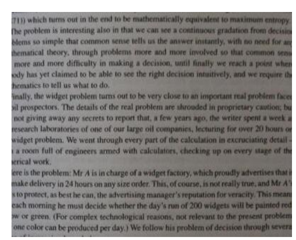


User data

- **Blessing of dimensionality:**

  Real data often concentrate on low-dimensional or degenerate structures in high-dimensional ambient space





local regularities, global symmetries, repetitive patterns, redundant sampling …

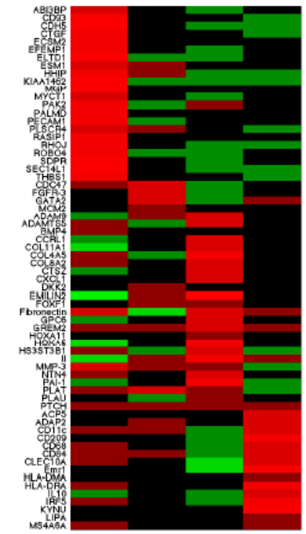
[John Wright and Yi Ma, 2014]

Robust Estimation

- But real-world data also frequently contain extraneous features, missing observations, or corruptions/outliers

face recognition
[Wright et al., 2009]

3D reconstruction
[Zhang et al., 2011]



gene expression
[Wang et al., 2012]

- Traditional methods (e.g., PCA, least squares regression) break down …

Replacements: Robust PCA, sparse representations/regression, and many others

[John Wright and Yi Ma, 2014]

Building Blocks for Robust Estimation

- Sparse representations:

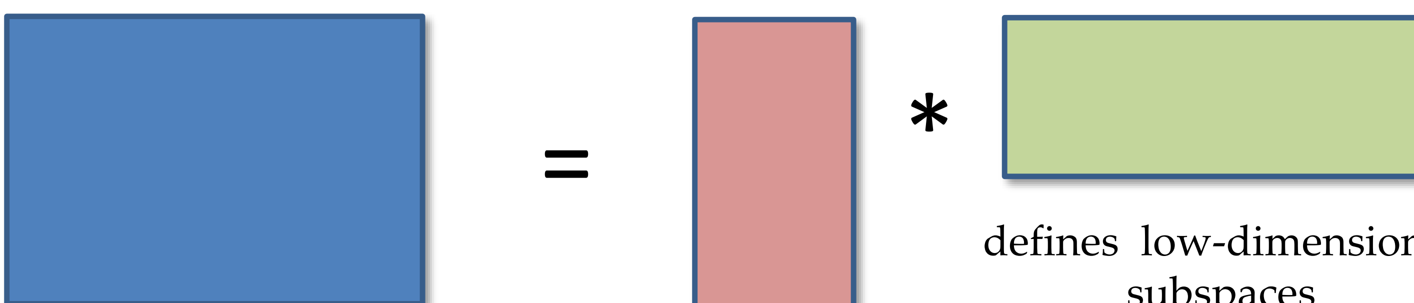
$$\mathbf{y} = \begin{bmatrix} -4 \\ -5 \\ 3 \end{bmatrix}, \qquad \Phi = \begin{bmatrix} 1 & 4 & 1 & 1 & 6 \\ -2 & 1 & -4 & 2 & -3 \\ 3 & 3 & 2 & -2 & 1 \end{bmatrix}$$

feasible solutions to $\mathbf{y} = \Phi\mathbf{x}$

$$\mathbf{u} = \begin{bmatrix} 4 \\ -1 \\ 3 \\ 5 \\ -2 \end{bmatrix} \qquad \mathbf{u}_0 = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \\ -1 \end{bmatrix}$$

non-sparse sparse

- Low-Rank matrices:



defines low-dimensional subspaces

High-Level Data Decomposition

Observed data: $\mathbf{X} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{n}, \quad \mathbf{x}^{(i)} \in \mathbb{R}^{d}, \ \forall i$

Basic building blocks can be combined in various ways to construct models of the form:

$$\mathbf{X} \quad = \quad \underbrace{\text{inlier component } (\mathbf{L})}_{\mathbf{L} = \left\{\mathbf{l}^{(i)}\right\}_{i=1}^{n} = \mathbf{UZ},\ \mathbf{l}^{(i)} = \mathbf{Uz}^{(i)}} \quad + \quad \underbrace{\text{outlier/noise component } (\mathbf{E})}_{\mathbf{E} = \left\{\mathbf{e}^{(i)}\right\}_{i=1}^{n}}$$

- low-dimensional latent structure, e.g., **U** defines a low-dim inlier subspace
- sometimes not fully observable, e.g., have measurement operator $\mathcal{A}(\mathbf{L})$

- sparse corruptions (possibly large)
- other errors or model mismatch

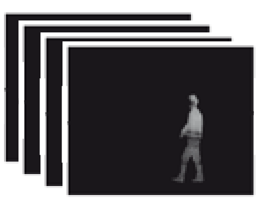
Background detection example:

observed video frames

=

1D subspace background component

+



sparse foreground component

Typical Objective for Signal Recovery

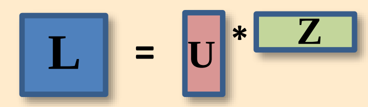
Challenging ill-posed inverse problem to recover low-dimensional representation $\mathbf{L}$:

$$\min_{\mathbf{L},\mathbf{E}} \; \left\| \mathbf{X} - \mathcal{A}(\mathbf{L}) - \mathbf{E} \right\|_2^2 \; + \; \lambda_1 g_1(\mathbf{L}) \; + \; \lambda_2 g_2(\mathbf{E})$$

$$\min_{\mathbf{L},\mathbf{E}} \; g_1(\mathbf{L}) \; + \; \lambda g_2(\mathbf{E}), \quad \text{s.t. } \mathbf{X} = \mathcal{A}(\mathbf{L}) + \mathbf{E}$$ (constrained version)

$\mathcal{A}$: linear meaurement operator

g_1 : favors low-dim representations



g_2 : favors sparsity

Example penalties:

$$g_1(\mathbf{L}) = \operatorname{rank}(\mathbf{L}) \equiv \left\| \sigma(\mathbf{L}) \right\|_0$$

\# nonzero singular values of $\mathbf{L}$

$$g_2(\mathbf{E}) = \left\| \mathbf{E} \right\|_0$$

\# nonzero elements in $\mathbf{E}$

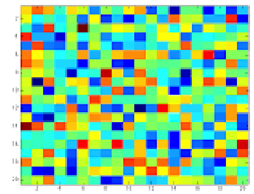
Note: Penalties are primarily used for limiting:

1) the intrinsic dimensionality of the inlier space
2) the cardinality of outliers

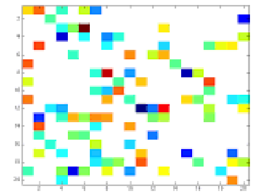
… not used for learning distribution within the inlier space

Special Cases

- Matrix recovery/completion:



low rank $\mathbf{L}$

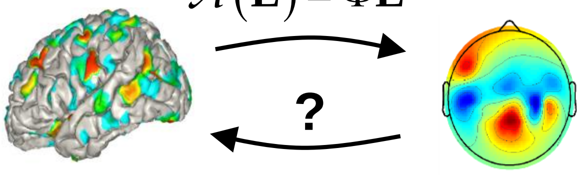


partial observations $\mathcal{A}(\mathbf{L})$

[Candès and Recht, 2008]

- Source localization:

$\mathcal{A}(\mathbf{L}) = \Phi\mathbf{L}$

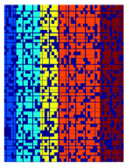


source space $\mathbf{L}$
(row sparse)

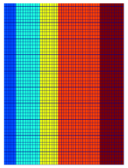
sensor space $\mathbf{X}$

[Baillet et al., 2001]

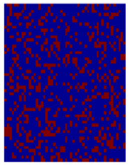
- Robust PCA:



$=$



$+$



observations $\mathbf{X}$ low rank $\mathbf{L}$ sparse $\mathbf{E}$

case study

[Chandrasekaran et al., 2011; Candès et al. 2011]

- **Many more …**

Weakness of Traditional Pipeline

Primary:

- Difficult nonconvex, NP-hard estimation process:

$$\min_{\mathbf{L},\mathbf{E}} \; g_1(\mathbf{L}) + \lambda g_2(\mathbf{E}), \quad \text{s.t. } \mathbf{X} = \mathcal{A}(\mathbf{L}) + \mathbf{E}$$

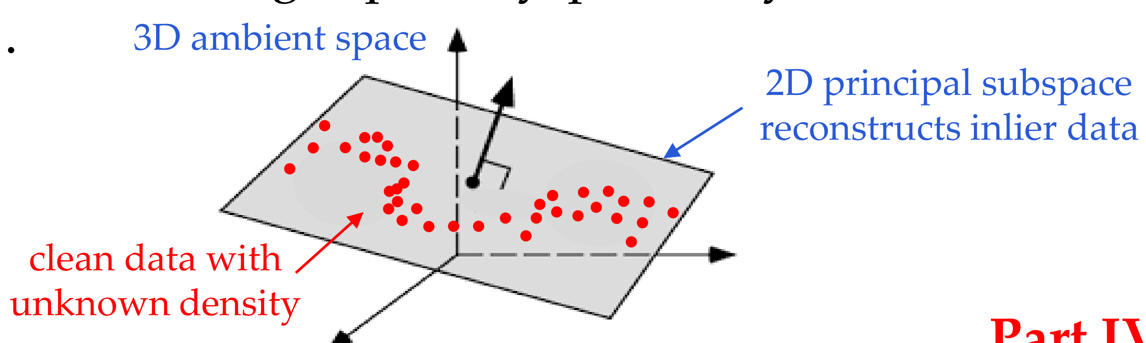
(and convex relaxations often fail …)

- Limited capacity inlier models, e.g., $\mathbf{L} = \mathbf{U} * \mathbf{Z}$

remainder of Part III

Secondary:

- Limited generative modeling capability, primarily used for data reconstruction …



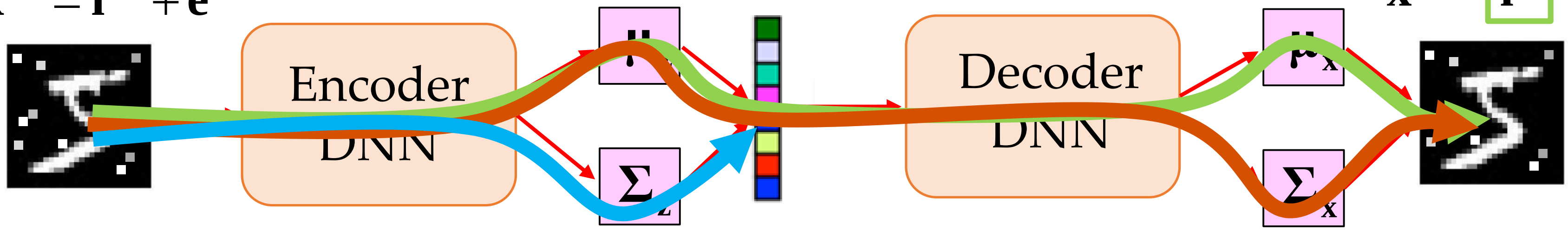
Part IV

How might the VAE model help?

easy to add observation model $\mathcal{A}$ here

Basic VAE architecture:

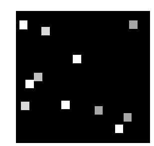
$\mathbf{x}^{(i)} = \mathbf{l}^{(i)} + \mathbf{e}^{(i)}$ $\qquad$ $\hat{\mathbf{x}}^{(i)} = \boxed{\hat{\mathbf{l}}^{(i)}} + \boxed{\hat{\mathbf{e}}^{(i)}}$



$\mathbf{z} \sim q_\varphi(\mathbf{z} \mid \mathbf{x}) = N(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$ $\qquad$ $\hat{\mathbf{x}} \sim p_\theta(\mathbf{x} \mid \mathbf{z}) = N(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$

High-Level Picture

Correspondences between VAE components and signal recovery:

- Deterministic path provides nonlinear inlier model: $\boldsymbol{\mu}_x\left(\boldsymbol{\mu}_z\left[\mathbf{x}^{(i)}, \boldsymbol{\varphi}\right], \boldsymbol{\theta}\right) \approx \mathbf{l}^{(i)} =$ 
- Decoder covariance path models sparse outliers: $\boldsymbol{\Sigma}_x\left(\boldsymbol{\mu}_z\left[\mathbf{x}^{(i)}, \boldsymbol{\varphi}\right], \boldsymbol{\theta}\right) \approx \left(\mathbf{e}^{(i)}\right)^2 =$ 
- VAE Encoder covariance:
  1) At global optima: determines inlier dimensionality
  2) Elsewhere: smooths bad local minima

[Dai et al., 2018]

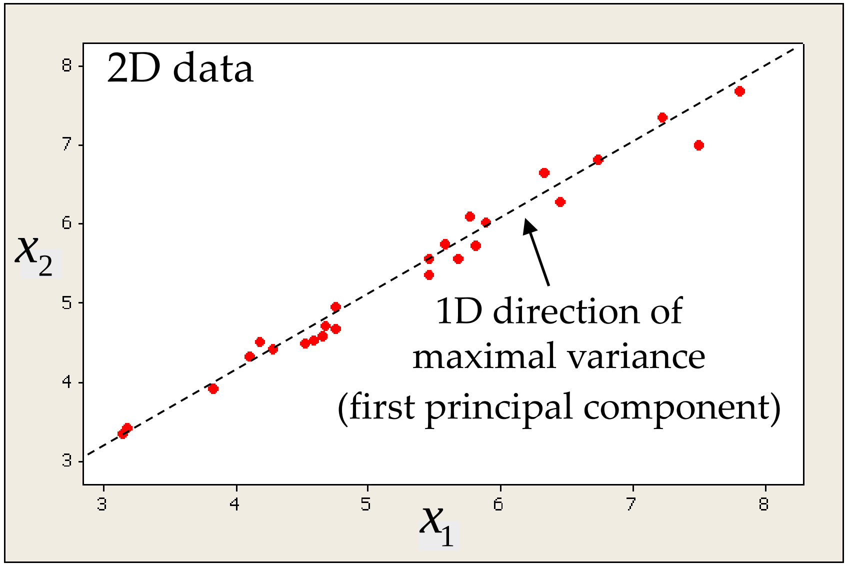
Case Study: Robust PCA

Why this is a good choice?

- Highly influential model e.g., 4358 citations to [Candès et al, 2011].
- Exactly follows inlier + outlier data decomposition (common to many signal processing applications …).
- Limited by
  - NP-hard estimation,
  - simple low-rank (bilinear inlier) model
- Correspondences with VAE can be explicitly quantified.
- Representative of connections between the VAE and other ill-posed inverse problems (e.g., compressive sensing, source localization, subspace clustering, matrix completion, …).

PCA Background

PCA finds directions of maximal variance:



Many different formulations given data $\mathbf{X} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{n} \in \mathbb{R}^{d\times n}$

Example (AE-like):

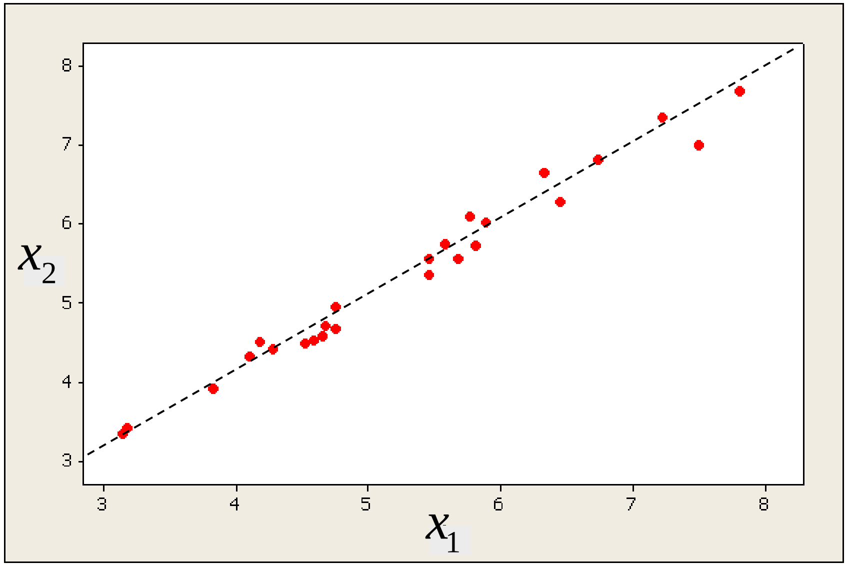
$$\underbrace{\mathbf{U}_*, \mathbf{V}_*}_{\text{defines } \kappa \text{ principal component directions}} = \arg\min_{\mathbf{U},\mathbf{V}} \sum_{i=1}^{n} \left\| \mathbf{x}^{(i)} - \underbrace{\mathbf{U}\mathbf{z}^{(i)}}_{\text{linear decoder}} \right\|_2^2, \quad \text{s.t. } \mathbf{z}^{(i)} = \underbrace{\mathbf{V}\mathbf{x}^{(i)}}_{\text{linear encoder}} \ \forall i, \quad \mathbf{U} \in \mathbb{R}^{d\times\kappa}, \ \mathbf{V} \in \mathbb{R}^{\kappa\times d}$$

Simple AE can compute principal components

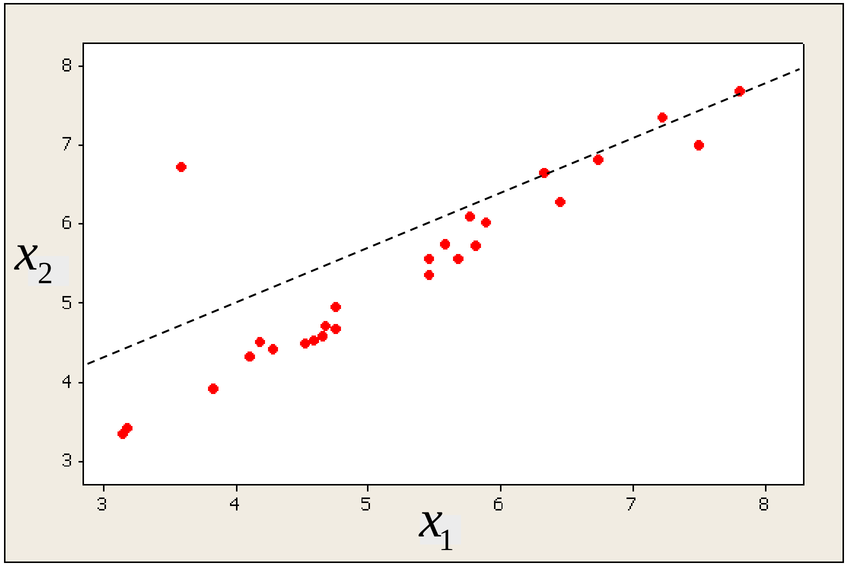
[Bourlard and Kamp, 1988]

PCA Sensitivity to Outliers

No Outliers



Single Outlier

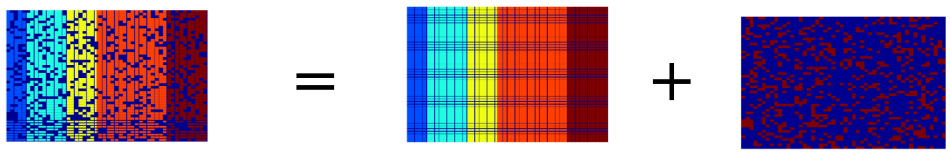


dashed line

first principal component direction

Robust PCA

[Chandrasekaran et al., 2011; Candès et al. 2011]



RPCA inverse problem:

$$\mathbf{L}_*, \mathbf{E}_* \quad = \quad \underbrace{\arg\min_{\mathbf{L},\mathbf{E}} \ \operatorname{rank}\left[\mathbf{L}\right] + \tfrac{1}{n}\left\|\mathbf{E}\right\|_0}_{\text{NP-hard}} \qquad \text{s.t.} \ \ \mathbf{X} = \mathbf{L} + \mathbf{E}$$

Convex relaxation:

$$\hat{\mathbf{L}}, \hat{\mathbf{E}} \quad = \quad \arg\min_{\mathbf{L},\mathbf{E}} \ \left\|\mathbf{L}\right\|_* + \tfrac{1}{\sqrt{n}}\left\|\mathbf{E}\right\|_1 \qquad \text{s.t.} \ \ \mathbf{X} = \mathbf{L} + \mathbf{E}$$

Theory: $\left\{\hat{\mathbf{L}}, \hat{\mathbf{E}}\right\} = \left\{\mathbf{L}_*, \mathbf{E}_*\right\}$ in very specialized conditions, but these rarely hold in practice …

What about the VAE?

Illustrative Degenerate Case

Original VAE objective:

$$L(\boldsymbol{\theta},\boldsymbol{\varphi}) = \sum_i \left\{ \mathrm{KL}\left[q_{\boldsymbol{\varphi}}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right) \| \, p(\mathbf{z}) \right] - \mathrm{E}_{q_{\boldsymbol{\varphi}}\left(\mathbf{z}\mid\mathbf{x}^{(i)}\right)}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)} \mid \mathbf{z}\right) \right] \right\}$$

Two assumptions:

1. Degenerate encoder covariance ⇨ $\boldsymbol{\Sigma}_{\mathbf{z}}[\mathbf{x},\boldsymbol{\varphi}] = \mathbf{0}$
2. High capacity decoder covariance ⇨ $\boldsymbol{\Sigma}_{\mathbf{x}}[\mathbf{z},\boldsymbol{\theta}]$ arbitrary

Can collapse VAE objective using *assumption 1*:

$$L(\boldsymbol{\theta},\boldsymbol{\varphi}) \equiv -\sum_i \log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)} \mid \mathbf{z} = \boldsymbol{\mu}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]\right)$$

Further simplification possible using *assumption 2*:

$$\min_{\boldsymbol{\Sigma}_{\mathbf{x}}} L(\boldsymbol{\theta},\boldsymbol{\varphi}) \equiv \sum_i \sum_j \log\left|e_j^{(i)}\right| \quad \text{s.t. } \mathbf{e}^{(i)} = \mathbf{x}^{(i)} - \boldsymbol{\mu}_{\mathbf{x}}\left[\boldsymbol{\mu}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right],\boldsymbol{\theta}\right], \ \forall \mathbf{x}^{(i)}$$

Equivalent to a deterministic autoencoder with Gaussian entropy loss …

VAE and Induced AE Side-by-Side

VAE model	Induced AE
$\boldsymbol{\Sigma}_{\mathbf{z}}[\mathbf{x},\boldsymbol{\varphi}]$ arbitrary	$\boldsymbol{\Sigma}_{\mathbf{z}}[\mathbf{x},\boldsymbol{\varphi}] = \mathbf{0}$
$\boldsymbol{\Sigma}_{\mathbf{x}}[\mathbf{z},\boldsymbol{\theta}]$ arbitrary	$\boldsymbol{\Sigma}_{\mathbf{x}}[\mathbf{z},\boldsymbol{\theta}]$ arbitrary

Induced AE is like a typical AE but with an outlier robust loss function:

$$L_{AE}(\boldsymbol{\theta},\boldsymbol{\varphi}) \triangleq \underbrace{\sum_i \sum_j \log\left|e_j^{(i)}\right|}_{\text{approximates } \ell_0 \text{ norm}} \quad \text{s.t. } \mathbf{e}^{(i)} = \mathbf{x}^{(i)} - \boldsymbol{\mu}_{\mathbf{x}}\left[\boldsymbol{\mu}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right],\boldsymbol{\theta}\right], \ \forall \mathbf{x}^{(i)}$$

$$\sum_i \sum_j \log\left|e_j^{(i)}\right| = \lim_{p\to 0} \sum_i \sum_j \tfrac{1}{p}\left(\left|e_j^{(i)}\right|^p - 1\right) \equiv \|\mathbf{E}\|_0$$

VAE is like a smoothed, regularized version of the induced AE:

$$\begin{aligned} L(\boldsymbol{\theta},\boldsymbol{\varphi}) &= \sum_i \left\{ \mathrm{KL}\left[q_{\boldsymbol{\varphi}}\left(\mathbf{z}\,|\,\mathbf{x}^{(i)}\right) \| \, p(\mathbf{z})\right] - \mathrm{E}_{q_{\boldsymbol{\varphi}}\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}\,|\,\mathbf{z}\right)\right] \right\} \\ &\geq \underbrace{\sum_i \mathrm{KL}\left[q_{\boldsymbol{\varphi}}\left(\mathbf{z}\,|\,\mathbf{x}^{(i)}\right) \| \, p(\mathbf{z})\right]}_{\text{regularization}} + \sum_i \underbrace{\mathrm{E}_{q_{\boldsymbol{\varphi}}\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}}_{\text{smoothing}}\left[\sum_j \log\left|\tilde{e}_j^{(i)}\right|\right] \end{aligned}$$

$$\text{s.t. } \tilde{\mathbf{e}}^{(i)} = \mathbf{x}^{(i)} - \boldsymbol{\mu}_{\mathbf{x}}[\mathbf{z},\boldsymbol{\theta}]$$

best possible w/
$\boldsymbol{\Sigma}_{\mathbf{x}}[\mathbf{z},\boldsymbol{\theta}]$ arbitrary

Both VAE and Induced AE have a distinct relationship with **Robust PCA** …

RPCA and the Induced AE

RPCA: $$\mathbf{L}_*, \mathbf{E}_* = \arg\min_{\mathbf{L},\mathbf{E}} \ \mathrm{rank}[\mathbf{L}] + \tfrac{1}{n}\|\mathbf{E}\|_0 \quad \text{s.t.} \quad \mathbf{X} = \mathbf{L} + \mathbf{E}$$

Assume *affine* decoder mean (arbitrary encoder mean):

$$\boldsymbol{\mu}_x[\mathbf{z},\boldsymbol{\theta}] = \mathbf{W}\mathbf{z} + \mathbf{b}, \quad \boldsymbol{\theta} = \{\mathbf{W},\mathbf{b}\}$$

$$\dim[\mathbf{z}] = \mathrm{rank}[\mathbf{L}_*]$$

Result: Induced AE shares the same combinatorial constellation of local and global minima of the constrained RPCA problem

$$\min_{\mathbf{L},\mathbf{E}} \ \|\mathbf{E}\|_0 \quad \text{s.t.} \quad \begin{array}{l} \mathbf{X} = \mathbf{L} + \mathbf{E} \\ \mathrm{rank}[\mathbf{L}] \leq \mathrm{rank}[\mathbf{L}_*] \end{array}$$

local minima a huge issue

Additional concern: If $\dim[\mathbf{z}] \neq \mathrm{rank}[\mathbf{L}_*]$, then even the global minimum need not be optimal …

superfluous dimensions can cause trouble

RPCA and the VAE

RPCA: $\quad \mathbf{L}_*, \mathbf{E}_* = \arg\min_{\mathbf{L},\mathbf{E}} \ \text{rank}[\mathbf{L}] + \frac{1}{n}\|\mathbf{E}\|_0 \quad \text{s.t. } \mathbf{X} = \mathbf{L} + \mathbf{E}$

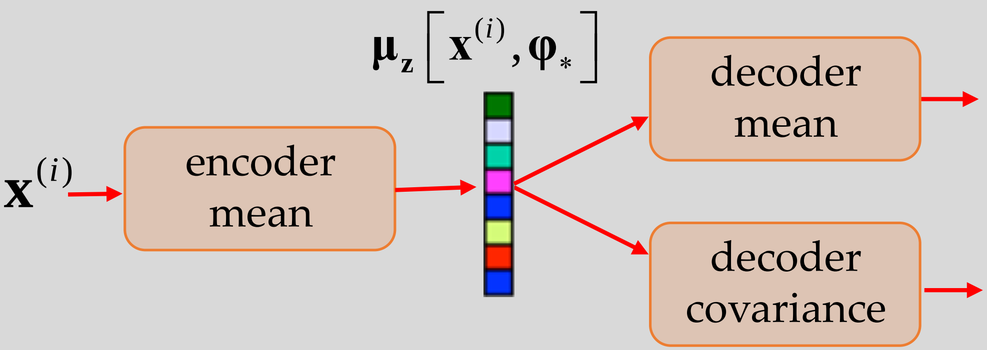
Assume *affine* decoder mean (arbitrary encoder mean):

$$\boldsymbol{\mu}_x[\mathbf{z}, \boldsymbol{\theta}] = \mathbf{W}\mathbf{z} + \mathbf{b}$$

$$\dim[\mathbf{z}] \boxed{\geq} \text{rank}[\mathbf{L}_*]$$

Theorem (Perfect Recovery):

Any VAE global optimum $\{\boldsymbol{\theta}_*, \boldsymbol{\varphi}_*\}$ is such that:



$$\boldsymbol{\mu}_x\left[\boldsymbol{\mu}_z\left[\mathbf{x}^{(i)}, \boldsymbol{\varphi}_*\right], \boldsymbol{\theta}_*\right] = \mathbf{l}_*^{(i)}$$

$$\boldsymbol{\Sigma}_x\left[\boldsymbol{\mu}_z\left[\mathbf{x}^{(i)}, \boldsymbol{\varphi}_*\right], \boldsymbol{\theta}_*\right] = \left(\mathbf{e}_*^{(i)}\right)^2$$

Matching global optima … **even after smoothing!**

… true even if $\dim[\mathbf{z}] > \text{rank}[\mathbf{L}_*]$

[Dai et al., 2018]

Two Underappreciated Distinctions

1) VAE can learn the optimal/minimal latent dimension of inlier model … unnecessary dimensions **can be automatically discarded**.

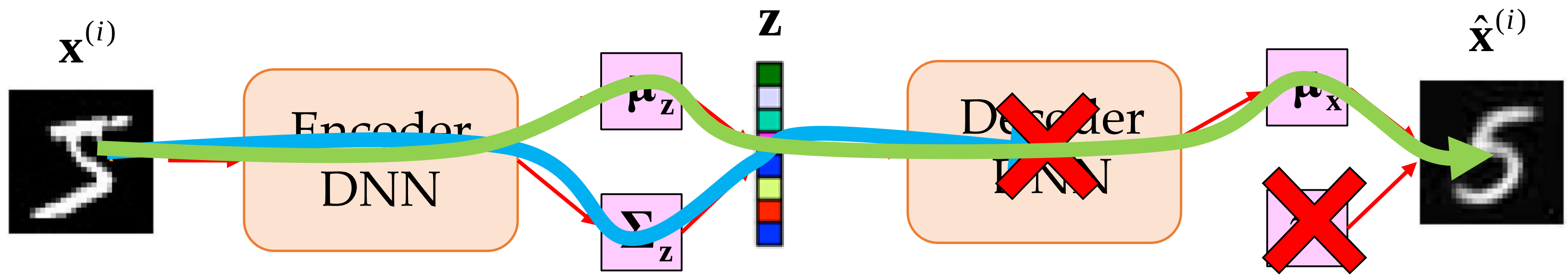
2) VAE smoothing/KL regularization impacts bad local minimum, does **not** change the global optimum.

Note: VAE capabilities motivated by Robust PCA example, but also translate to more complex inlier models

Discarding Unnecessary Latent Dimensions

Observed data: $\mathbf{X} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{n}, \quad \mathbf{x}^{(i)} = \mathbf{l}^{(i)} \in \mathcal{X}$ ⇒ arbitrary inlier manifold; for simplicity no outliers

Assumed VAE (arbitrary encoder/decoder networks):



$\mathbf{z} \sim q_\varphi(\mathbf{z} \mid \mathbf{x}) = N(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$ $\qquad$ $\hat{\mathbf{x}} \sim p_\theta(\mathbf{x} \mid \mathbf{z}) = N(\boldsymbol{\mu}_x, \gamma \mathbf{I})$

Theorem (Reconstruction Invariance):

Under some technical conditions, any VAE global optimum $\{\boldsymbol{\theta}_*, \boldsymbol{\varphi}_*\}$ is such that $\boxed{\gamma \to 0}$ and reconstructions are exact:

$$\boldsymbol{\mu}_x\left(\boldsymbol{\mu}_z\left[\mathbf{x}^{(i)}, \boldsymbol{\varphi}_*\right] + \boxed{\boldsymbol{\Sigma}_z^{1/2}\left[\mathbf{x}^{(i)}, \boldsymbol{\varphi}_*\right]\boldsymbol{\varepsilon}}, \boldsymbol{\theta}_*\right) = \boxed{\boldsymbol{\mu}_x\left(\boldsymbol{\mu}_z\left[\mathbf{x}^{(i)}, \boldsymbol{\varphi}_*\right], \boldsymbol{\theta}_*\right)} = \mathbf{x}^{(i)}, \quad \forall \boldsymbol{\varepsilon}, \forall i$$

Key Conclusion: At global minimum, encoder randomness will not impact perfect reconstructions ⇒ can be "pruned" with white noise

[Dai & Wipf, 2019]

Discarding Unnecessary Latent Dimensions Cont.

- Recall VAE KL term with Gaussian encoder satisfies

$$\mathrm{KL}\left[q_{\boldsymbol{\varphi}}\left(\mathbf{z}\,|\,\mathbf{x}^{(i)}\right)||\,N\left(\mathbf{z}\,|\,\mathbf{0},\mathbf{I}\right)\right] \propto \left\|\boldsymbol{\mu}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]\right\|_2^2 + \mathrm{tr}\left(\boldsymbol{\Sigma}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]\right) - \log\left|\boldsymbol{\Sigma}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]\right|$$

- With diagonal covariance (common choice), further decouples to

$$\mathrm{KL}\left[q_{\boldsymbol{\varphi}}\left(\mathbf{z}\,|\,\mathbf{x}^{(i)}\right)||\,N\left(\mathbf{z}\,|\,\mathbf{0},\mathbf{I}\right)\right] \propto \sum_{j=1}^{\kappa}\left\{\mu_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]_j^2 + \sigma_{\mathbf{z}}^2\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]_j - \log\left(\sigma_{\mathbf{z}}^2\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]_j\right)\right\}$$

- Reconstruction Invariance Theorem implies that certain dimensions will not influence VAE data term.

- Along these dimensions, KL term can be minimized independently:

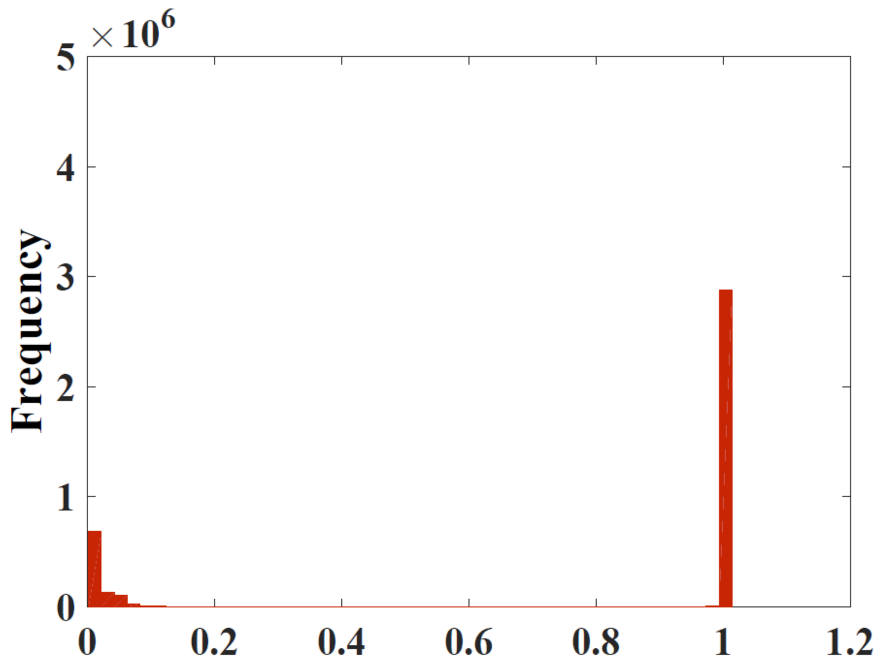
  Optimal moments for these unnecessary dimensions:

$$\mu_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]_j^2 \to 0, \qquad \sigma_{\mathbf{z}}^2\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]_j \to 1$$

- This non-informative white noise will be filtered out by the decoder.

Empirical Example

Histogram of $\sigma_{\mathbf{z}}^2\left[\mathbf{x}^{(i)}, \boldsymbol{\varphi}\right]_j$ values for VAE trained on MNIST data



for useful dimensions, encoder variance is near zero; facilitates good reconstructions

for unnecessary dimensions, encoder variance is near one; optimizes KL term

(Encoder noise will serve an important purpose in Part IV…)

Filtering Unnecessary Dimensions

$$\sigma^2_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]_j = 1.0$$

unnecessary dimension

Reconstructions as we change latent code along this dimension (other dimensions fixed)

Image Variance = $\boxed{0.000}$ **no changes**

$$\sigma^2_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]_j = 0.005 \approx 0$$

necessary dimension

Reconstructions as we change latent code along this dimension (other dimensions fixed)

Image Variance = $\boxed{27.20}$ **large changes**

Two Underappreciated Distinctions

1) VAE can learn the optimal/minimal latent dimension of inlier model … unnecessary dimensions **can be automatically discarded**.

2) VAE smoothing/KL regularization impacts bad local minimum, does **not** change the global optimum.

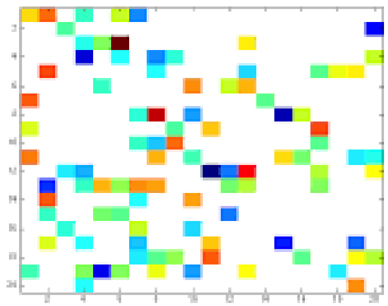
Note: VAE capabilities motivated by Robust PCA example, but also translate to more complex inlier models

Benefits of VAE smoothing

With induced AE (no smoothing), we enter a local minima at any outlier support pattern

$$\mathbf{E} = \left[\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(n)}\right] =$$

$$\mathbf{e}^{(i)} = \mathbf{x}^{(i)} - \boldsymbol{\mu}_{\mathbf{x}}\left[\boldsymbol{\mu}_{\mathbf{z}}\left[\mathbf{x}^{(i)}, \boldsymbol{\varphi}\right], \boldsymbol{\theta}\right]$$



zero-valued elements can never change

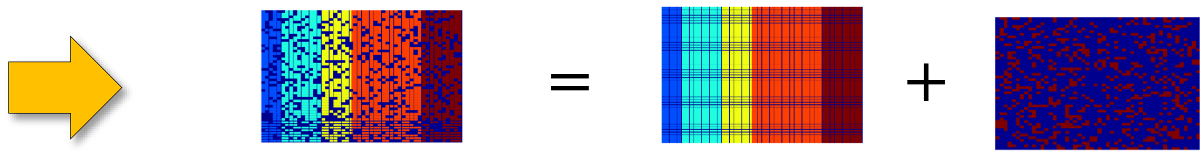
But for the VAE, every support pattern need **not** be a local minimum because of **selective smoothing** …

does not impact global minimum
(unlike convex relaxations …)

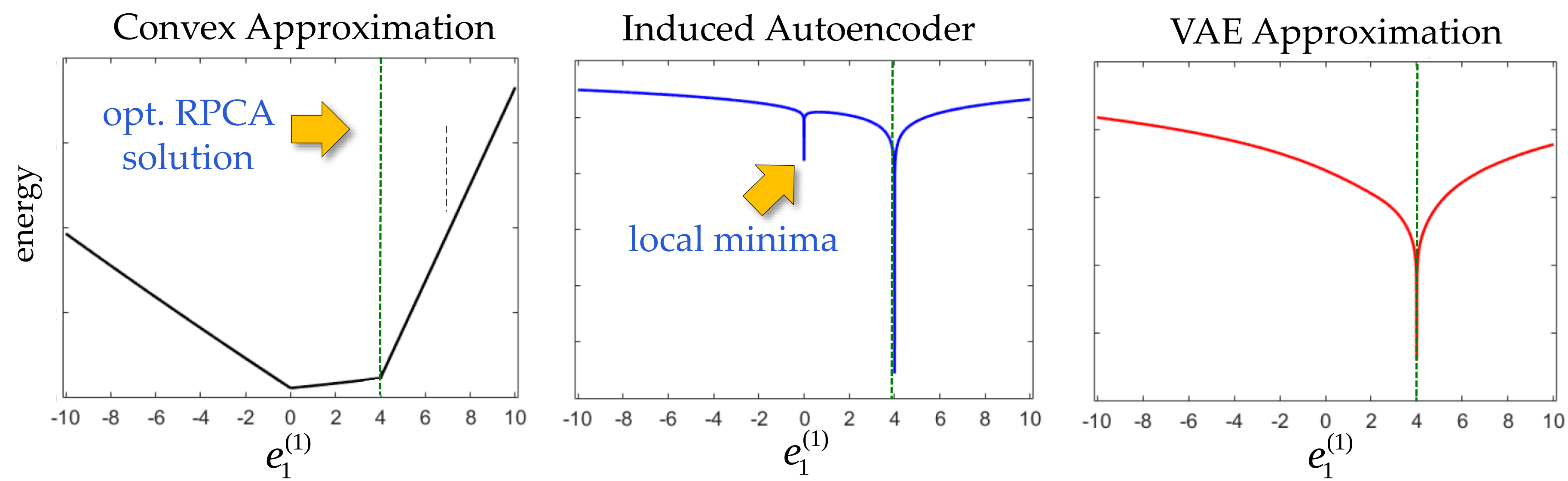
[Dai et al., 2018]

Illustration of Selective Smoothing Effects

Generated data



observations ($\mathbf{X}$) low rank ($\mathbf{L}$) sparse outliers ($\mathbf{E}$)



Representative 1D slice of energy functions while varying the coefficient $e_1^{(1)} = x_1^{(1)} - \mu_{\mathbf{x}}\left[\boldsymbol{\mu}_{\mathbf{z}}\left[\mathbf{x}^{(1)}, \boldsymbol{\varphi}\right], \boldsymbol{\theta}\right]_1$

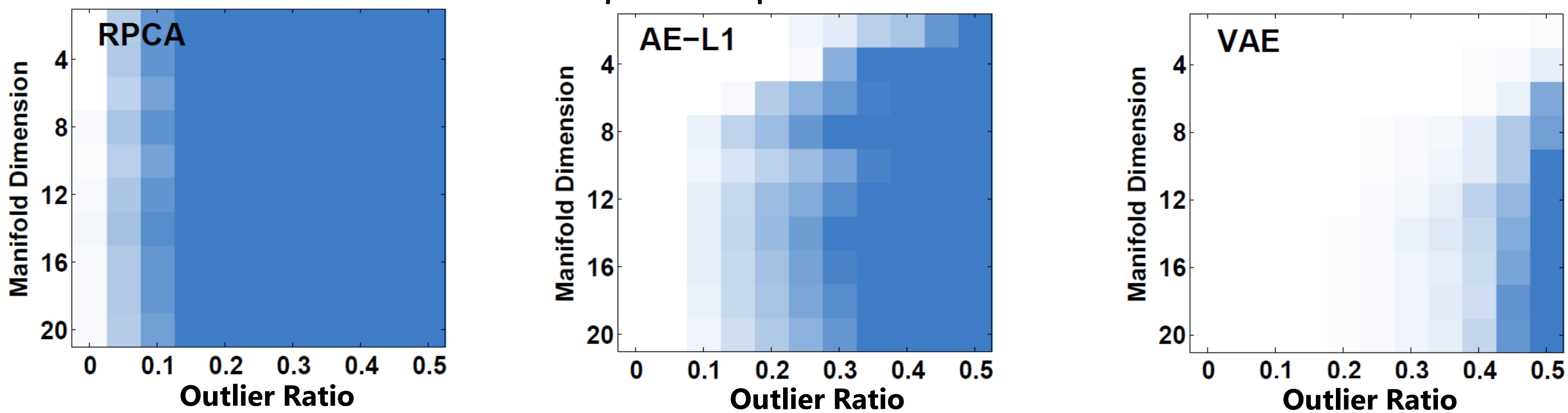
Non-Linear Manifold Recovery

$\mathbf{X}$ = low-dimentional manifold component + sparse outlier component

$$\boldsymbol{\mu}_{\mathrm{x}}\left[\boldsymbol{\mu}_{\mathrm{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}'\right],\boldsymbol{\theta}'\right], \quad \forall \mathbf{x}^{(i)}$$

convex relaxation

induced AE with extra ℓ_1 penalty on latent code

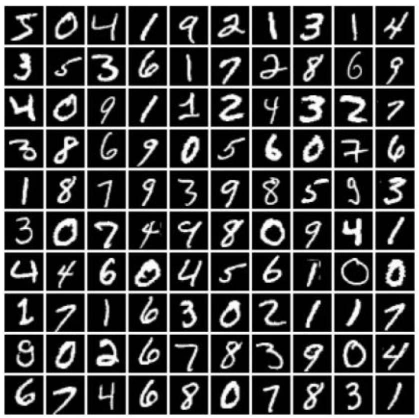


white = success (zero error), blue = failure (large error)

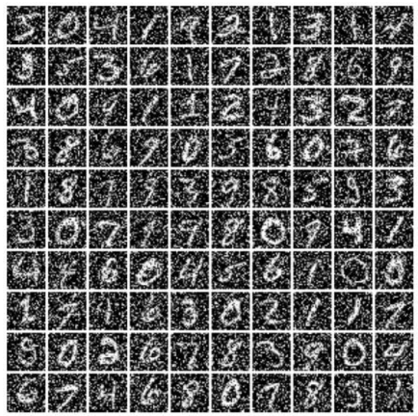
[Dai et al., 2018]

MNIST Example

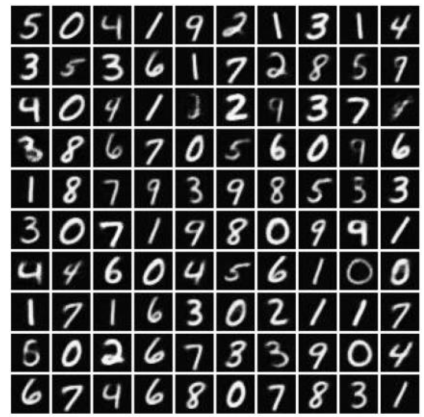
Original data



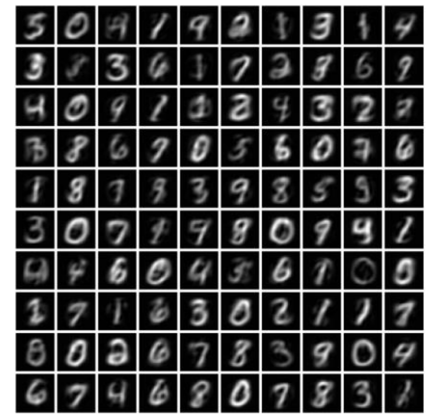
40% corrupted



VAE reconstructions



Convex RPCA reconstructions



[Wang et al., 2018]

A Lingering Issue …

A large training corpus $\mathbf{X} = \left\{ \mathbf{x}^{(i)} \right\}_{i=1}^{n}$ is required for learning complex manifolds with outliers

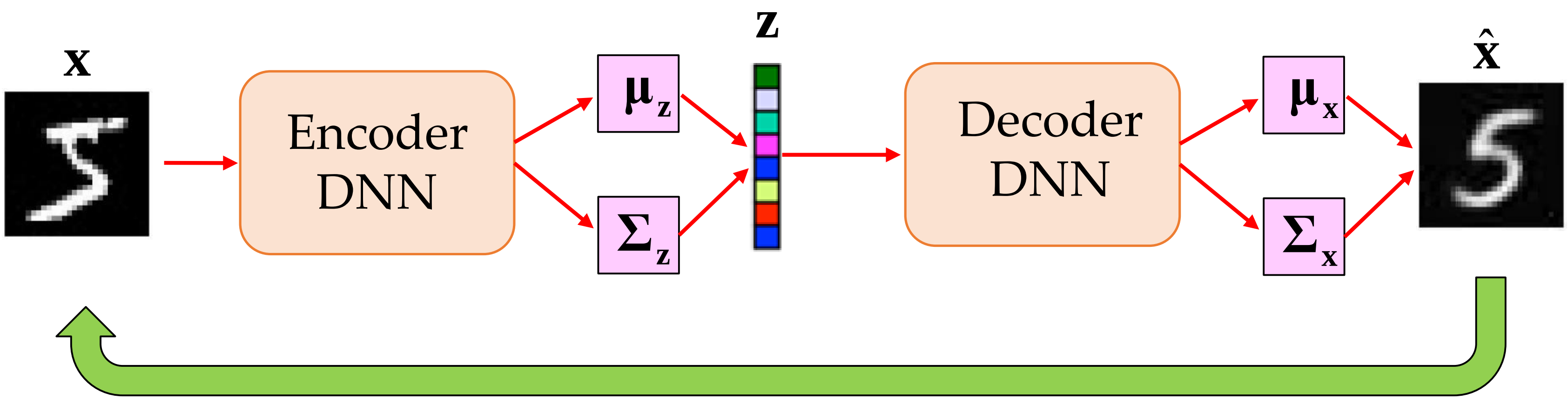
Solution:

Recycle dirty samples via specialized recurrent connections … automated data augmentation

[Wang et al., 2018]

Recycled/Recurrent VAE

Given a single input sample, bootstrap virtual samples via recurrent connection



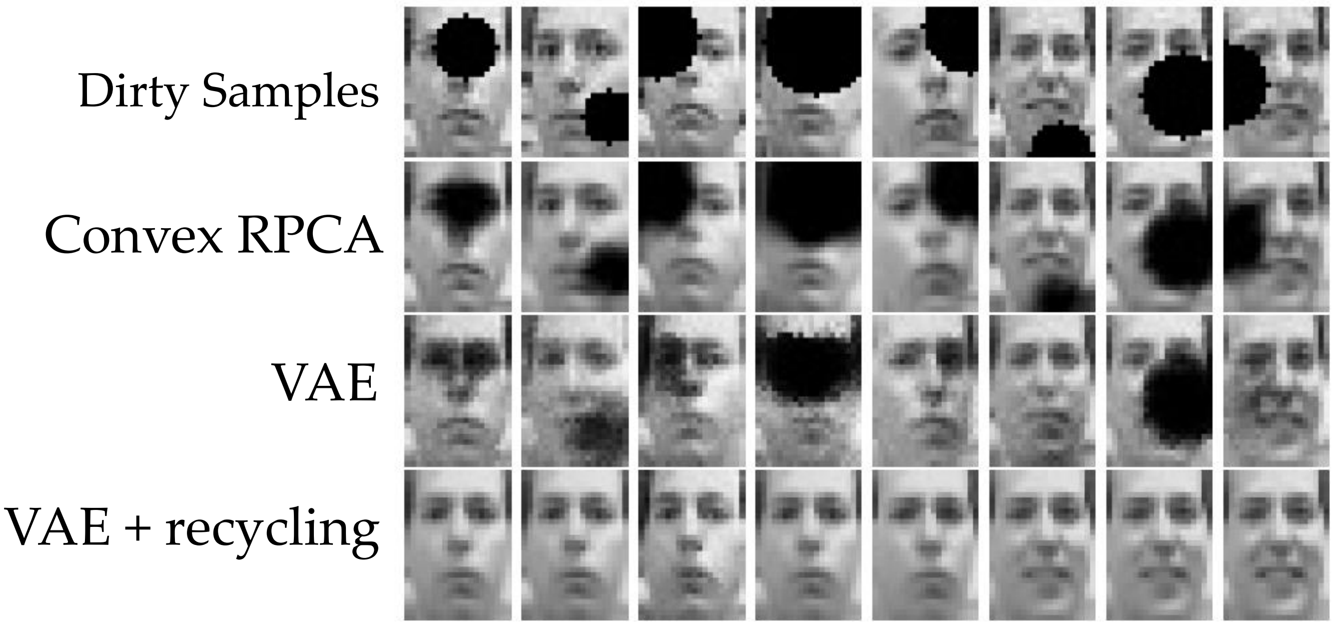
add simple feedback loop during training

Properties:

- No additional parameters required (simple SGD still works …)
- Partially detected outliers can be removed in multiple passes
- Close connection to iterative reweighting algorithms

[Wang et al., 2018]

Frey Face Data Recovery



[Wang et al., 2018]

Summary of Robust PCA Case Study

- The VAE with an affine decoder mean collapses to a robust PCA variant with attractive properties.
- In broader regimes, can be viewed as powerful nonlinear extension.
- Analysis reveals underappreciated effects of VAE regularization:
  1. Can learn optimal latent dimensionality
  2. Can selectively smooth away bad local minima while preserving good global solutions
  3. Can potentially be useful for deterministic data cleaning tasks unrelated to generative modeling per se.
  4. Extra recurrent connections/recycling, can serve as a useful fo rm of data augmentation.
- Representative of connections between the VAE and other ill-posed inverse problems.

Questions?

Part IV:
From Signal Reconstruction to Generative Modeling

Note: Updated version of slides available at http://www.davidwipf.com/

Recap

- VAE can extend/enhance capabilities of traditional algorithms for finding low-dimensional structure.

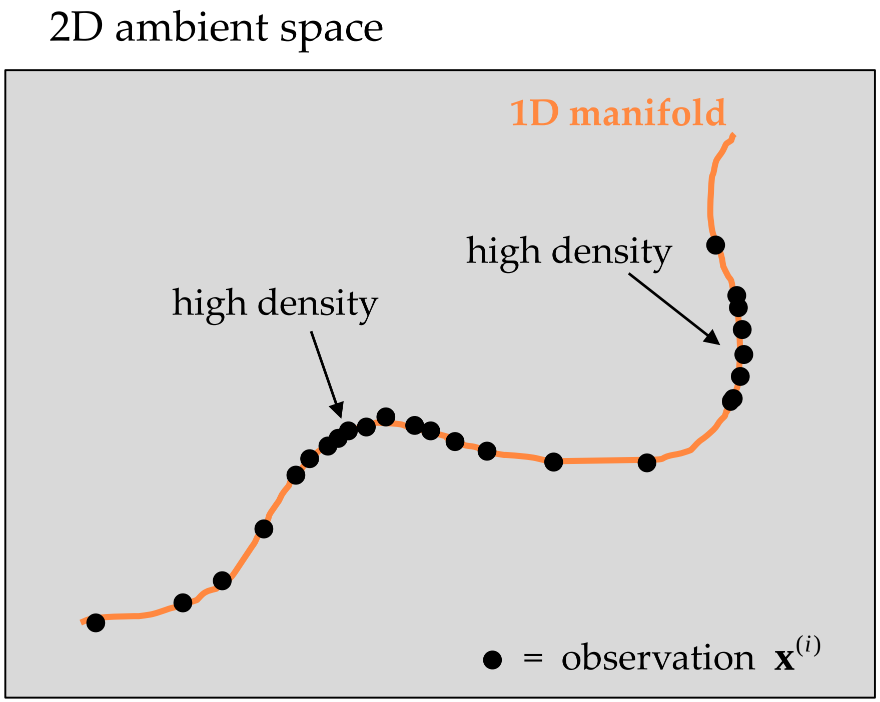
- Low-dimensional structure could be an arbitrary manifold.

- Can reconstruct data (possibly corrupted by outliers) by fitting a parsimonious inlier model.

- But this is not sufficient for a full generative model …

Note: Will mostly assume no outliers in Part IV
for simplicity … but the key concepts generalize.

Illustration



- Reconstructing data using parsimonious inlier model provides estimate of the 1D manifold (Part III).
- But it does **not** provide any information about the data distribution **within** the manifold.
- <u>**Key question**</u>**:** How can good reconstructions segue to a good generative model?

Revisiting Original VAE Bound

Variational upper bound (from Part II):

$$-\sum_i \log p_\theta\left(\mathbf{x}^{(i)}\right) \leq \sum_i \left\{ \mathrm{KL}\left[q_\varphi\left(\mathbf{z}|\mathbf{x}^{(i)}\right) \| \, p_\theta\left(\mathbf{z}|\mathbf{x}^{(i)}\right)\right] - \log p_\theta\left(\mathbf{x}^{(i)}\right) \right\}$$

$$\equiv \sum_i \left\{ \mathrm{KL}\left[q_\varphi\left(\mathbf{z}|\mathbf{x}^{(i)}\right) \| \, N\left(\mathbf{z}|\mathbf{0},\mathbf{I}\right)\right] - \mathrm{E}_{q_\varphi\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}\left[\log p_\theta\left(\mathbf{x}^{(i)}|\mathbf{z}\right)\right] \right\}$$

Equality iff:

$$\underbrace{q_\varphi\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}_{\text{encoder distribution}} = p_\theta\left(\mathbf{z}|\mathbf{x}^{(i)}\right) = \frac{p_\theta\left(\mathbf{x}^{(i)}|\mathbf{z}\right) N\left(\mathbf{z}|\mathbf{0},\mathbf{I}\right)}{\int \underbrace{p_\theta\left(\mathbf{x}^{(i)}|\mathbf{z}\right)}_{\text{decoder distribution}} N\left(\mathbf{z}|\mathbf{0},\mathbf{I}\right) dz}$$

Consequence: If encoder and decoder are sufficiently complex such that

Gaussian encoder/decoder

generally <u>not</u> Gaussian

$$\boxed{q_{\varphi_*}\left(\mathbf{z}|\mathbf{x}\right)} = \boxed{p_{\theta_*}\left(\mathbf{z}|\mathbf{x}\right)}$$

$$p_{\theta_*}\left(\mathbf{x}\right) = \int \boxed{p_{\theta_*}\left(\mathbf{x}|\mathbf{z}\right)} N\left(\mathbf{z}|\mathbf{0},\mathbf{I}\right) dz = p_{gt}\left(\mathbf{x}\right)$$

⇒ can estimate ground-truth distributions just by minimizing VAE cost

Problem: But typical VAEs for continuous data often involve <u>Gaussian</u> encoder and decoder distributions … no match with true latent posterior.

Impact of VAE Gaussian Assumptions

Assume for simplicity:

model

- Decoder covariance: $\boldsymbol{\Sigma}_{\mathbf{x}}\left[\mathbf{z},\boldsymbol{\theta}\right]=\gamma\mathbf{I},\quad\forall\mathbf{z}$

  single learnable parameter
  (common in practice if no outliers)

- Decoder mean, encoder mean/covariance all arbitrary functions

---

data

- Asymptotic regime

$$L(\boldsymbol{\theta},\boldsymbol{\varphi}) \equiv \sum_{i=1}^{n}\left\{ \mathrm{KL}\left[q_{\boldsymbol{\varphi}}\left(\mathbf{z}\,|\,\mathbf{x}^{(i)}\right)\|\,N\left(\mathbf{z}\,|\,\mathbf{0},\mathbf{I}\right)\right] - \mathrm{E}_{q_{\boldsymbol{\varphi}}\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}\,|\,\mathbf{z}\right)\right]\right\}$$

$$\xrightarrow{n\to\infty} \underbrace{\int\left\{\mathrm{KL}\left[q_{\boldsymbol{\varphi}}\left(\mathbf{z}\,|\,\mathbf{x}\right)\|\,N\left(\mathbf{z}\,|\,\mathbf{0},\mathbf{I}\right)\right] - \mathrm{E}_{q_{\boldsymbol{\varphi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{x}\,|\,\mathbf{z}\right)\right]\right\}\underbrace{\mu_{gt}\left(d\mathbf{x}\right)}_{\text{ground-truth measure}}}_{\vec{L}(\boldsymbol{\theta},\boldsymbol{\varphi})}$$

$\vec{L}(\boldsymbol{\theta},\boldsymbol{\varphi})$ ➡ asymptotic loss

- Potential low-dimensional structure in data (and no outliers):

$$\mu_{gt}\neq 0 \text{ on } r\text{-dimensional manifold } \mathcal{X} \quad ➡ \quad \Pr\left(\mathbf{x}\notin\mathcal{X}\right) = 0$$

(<u>Note</u>: If $r = \dim(\mathbf{x})$, then no manifold structure)

Impact of VAE Gaussian Assumptions Cont.

Notation: $\dim(\mathbf{x}) = d, \quad \dim(\mathbf{z}) = \kappa$

i.e., no manifold, VAE latent dim large enough, and density exists

Theorem (Exact Density Recovery):

Scenario: $r = d, \ \kappa \geq r, \ \text{ and } \ \mu_{gt}(d\mathbf{x}) = p_{gt}(\mathbf{x})\,d\mathbf{x}$

Then any optimum $\{\boldsymbol{\theta}_*, \boldsymbol{\varphi}_*\} \in \arg\min_{\boldsymbol{\theta},\boldsymbol{\varphi}} \vec{L}(\boldsymbol{\theta},\boldsymbol{\varphi})$ will be such that*

$$\mathrm{KL}\left[q_{\boldsymbol{\varphi}_*}(\mathbf{z}|\mathbf{x}) \| p_{\boldsymbol{\theta}_*}(\mathbf{z}|\mathbf{x})\right] = 0 \quad \text{and} \quad p_{\boldsymbol{\theta}_*}(\mathbf{x}) = \int p_{\boldsymbol{\theta}_*}(\mathbf{x}|\mathbf{z}) N(\mathbf{z}|\mathbf{0},\mathbf{I})\,d\mathbf{x} = p_{gt}(\mathbf{x})$$

[Dai & Wipf, 2019]

Positive:

- When there is no manifold, VAE global optimum exactly corresponds with recovery of ground-truth measure even with Gaussian assumptions.

Negative Corollary:

- When there is a manifold, i.e., $r < d$, cannot rule out globally optimal solutions that do not correspond with the ground-truth measure …

Conclusion: VAE needs modifications to correctly handle manifolds

*Some additional technical conditions apply

One Candidate Solution:
More Complex, Non-Gaussian Encoders

- A variety of non-Gaussian decoders have been proposed based on invertible flows (Part I) and related.

  [Burda et al., 2015; Kingma et al., 2016; Rezende & Mohamed, 2016; van den Berg et al., 2018]

- This can improve non-negative likelihood (NLL) scores on test data:

Test set performance on the CIFAR-10 data.

	$K = 0$	$K = 2$	$K = 5$	$K = 10$
$-\ln p(\mathbf{x})$	-293.7	-308.6	-317.9	-320.7

flow length

[Rezende & Mohamed, 2016]

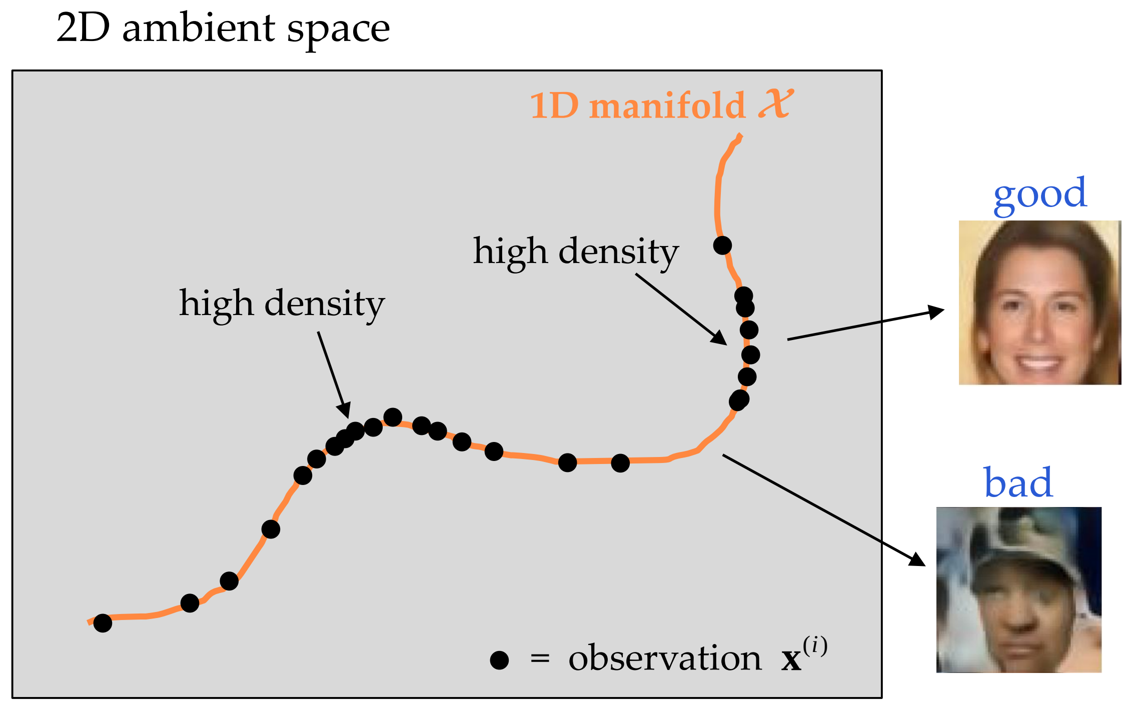
- Weaknesses:

  1) Does not solve the non-uniqueness issue with low-dim manifolds.
  2) Has not yet shown <u>quantitative</u> improvement generating new samples (… this is of course subject to change).

- Similar conclusions for non-Gaussian VAE latent priors

  [(Tomczak & Welling, 2018; Zhao et al., 2018)]

Potentially Misleading NLL Scores



Can have $-\sum_i \log p_{\boldsymbol{\theta}}\left(\mathbf{x}\right) \rightarrow -\infty$ (infinite density)
with just a uniform measure on $\mathcal{X}$ and $\Pr\left(\mathbf{x} \notin \mathcal{X}\right) = 0$

But samples drawn from the low-density manifold regions might be bad …

NLL scores need not correlate with sample quality

[Theis et al., 2016]

Helpful Alternative Viewpoint

- Fix: $\boldsymbol{\Sigma}_{\mathbf{z}}\left[\mathbf{x},\boldsymbol{\varphi}\right] = \mathbf{0}, \quad \boldsymbol{\Sigma}_{\mathbf{x}}\left[\mathbf{z},\boldsymbol{\theta}\right] = \mathbf{I}$

- VAE energy collapses to a simple deterministic, induced AE:

$$L_{\text{AE}}\left(\boldsymbol{\theta},\boldsymbol{\varphi}\right) \triangleq \sum_{i=1}^{n} \Big\| \mathbf{x}^{(i)} - \underbrace{\boldsymbol{\mu}_{\mathbf{x}}\left[\mathbf{z}^{(i)},\boldsymbol{\theta}\right]}_{\text{decoder}} \Big\|_2^2, \quad \text{s.t. } \mathbf{z}^{(i)} = \underbrace{\boldsymbol{\mu}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]}_{\text{encoder}}$$

- Compute: $\boldsymbol{\theta}_*,\boldsymbol{\varphi}_* = \arg\min_{\boldsymbol{\theta},\boldsymbol{\varphi}} L_{\text{AE}}\left(\boldsymbol{\theta},\boldsymbol{\varphi}\right)$

- Collect corresponding latent variables: $\left\{\mathbf{z}^{(i)}\right\}_{i=1}^{n}, \quad \mathbf{z}^{(i)} = \boldsymbol{\mu}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}_*\right]$

- **Hypothetical:** Suppose $\underbrace{L_{\text{AE}}\left(\boldsymbol{\theta}_*,\boldsymbol{\varphi}_*\right) \approx 0}$ and $\underbrace{\left\{\mathbf{z}^{(i)}\right\}_{i=1}^{n} \approx N\left(\mathbf{0},\mathbf{I}\right)}$

<u>Criteria A</u>: Good reconstruction of training data (like VAE from Part III)

<u>Criteria B</u>: Approximation to some known latent distribution

⇨ Can in principle apply an AE for generating new samples …

Illustration of AE Required Criteria

$$\{\mathbf{x}^{(i)}\}_{i=1}^{n} \sim p_{gt}(\mathbf{x}) \rightarrow \boxed{\text{Encoder DNN}} \rightarrow \underbrace{\{\hat{\mathbf{z}}^{(i)}\}_{i=1}^{n} \approx N(\mathbf{0},\mathbf{I})}_{\text{Criteria B}} \rightarrow \boxed{\text{Decoder DNN}} \rightarrow \underbrace{\{\hat{\mathbf{x}}^{(i)}\}_{i=1}^{n} \approx \{\mathbf{x}^{(i)}\}_{i=1}^{n}}_{\text{Criteria A}}$$

- Could generate <u>new</u> samples via:

$$N(\mathbf{0},\mathbf{I}) \rightarrow \boxed{\text{Decoder DNN}} \rightarrow \{\mathbf{x}^{(j)}\}_{j=1}^{m}$$

If Criteria A and B hold, should be similar to ground truth samples

- In practice, an AE can satisfy Criteria A, but will generally **not** satisfy Criteria B …

Practical workaround:

> Can penalize some measure of the distance between samples $\{\hat{\mathbf{z}}^{(i)}\}_{i=1}^{n}$ and $N(\mathbf{0},\mathbf{I})$

Generic Form of AE-Based Generative Model

Enhanced AE energy:

$$L_{\mathrm{AE}+}(\boldsymbol{\theta},\boldsymbol{\varphi}) \triangleq \sum_{i=1}^{n} \underbrace{\left\| \mathbf{x}^{(i)} - \boldsymbol{\mu}_{\mathbf{x}}\left(\mathbf{z}^{(i)},\boldsymbol{\theta}\right)\right\|_2^2}_{\text{data fit term}} + \underbrace{\lambda\Delta\left[\left\{\mathbf{z}^{(i)}\right\}_{i=1}^{n}, N(\mathbf{0},\mathbf{I})\right],}_{\substack{\text{penalty favors latent}\\ \text{samples ``similar'' to}\\ \text{standardized Gaussian}}} \quad \text{s.t. } \mathbf{z}^{(i)} = \boldsymbol{\mu}_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right], \quad \forall i$$

Candidate penalties based on Wasserstein distance measures $\Rightarrow$ Wasserstein AE (WAE)

Two main variants incorporate:

- Maximum mean discrepancy (MMD)
- Generative adversarial network (GAN)

$\}$ WAE-MMD, WAE-GAN

(Note: There also exists stochastic versions of the WAE encoder, but empirical results are not available)

[Tolstikhin et al., 2018]

WAE Results

quantitative measure of perceptual quality; lower is better

Algorithm	$\boxed{\textbf{FID}}$
VAE	63
WAE-MMD	55
WAE-GAN	42

significant improvement over the VAE

WAE-MMD generated samples:

[Tolstikhin et al., 2018]

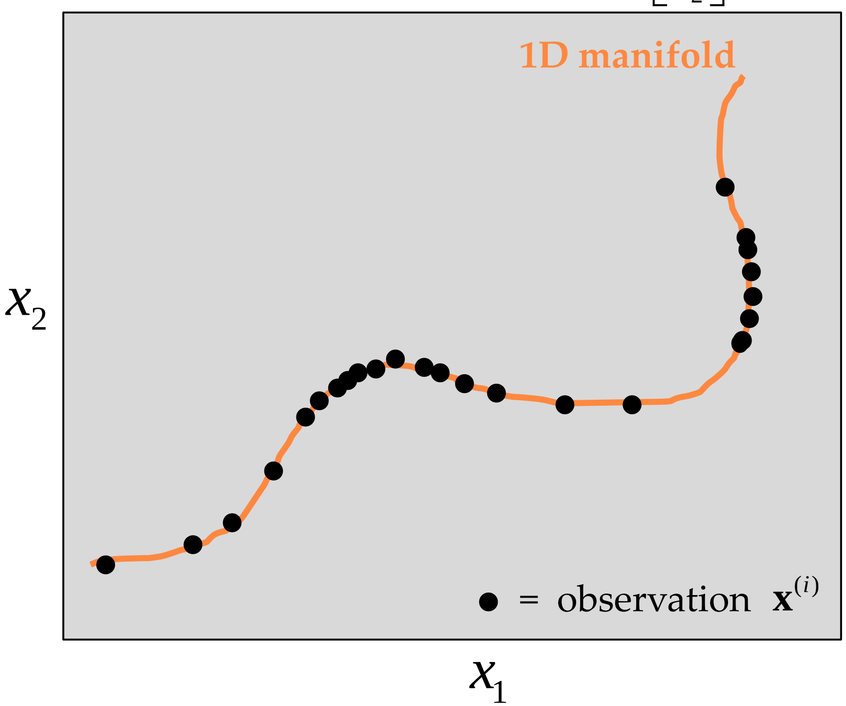
Potential WAE Limitations

- Must tune trade-off parameter λ
- If dim($\mathbf{z}$) is too small ⇨ large reconstruction error (fails Criteria A)
- If dim($\mathbf{z}$) is too high ⇨ large distribution mismatch (fails Criteria B)

The Problem of Excess Latent Dimensions

$\mathbf{z}^{(i)} = \boldsymbol{\mu}_{\mathbf{z}}\left[\mathbf{x}^{(i)}, \boldsymbol{\varphi}\right], \quad \forall i$ ➡ deterministic encoder mapping between **x** and **z** space

2D observation space, $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$



$\boldsymbol{\mu}_{\mathbf{z}}$

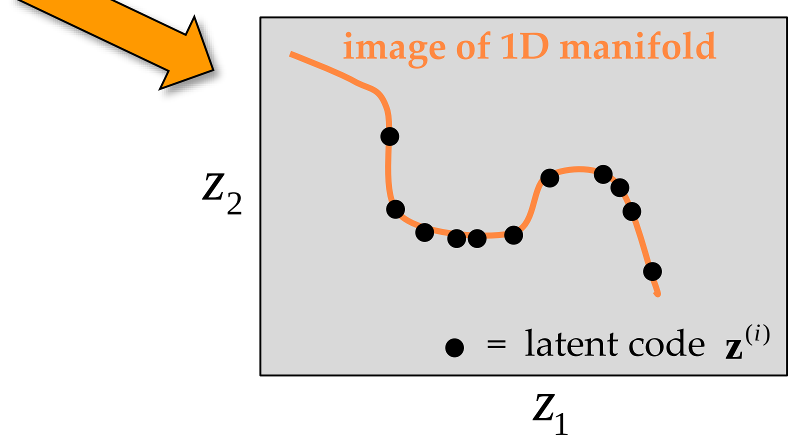
1D latent space, $\mathbf{z} = z$
(optimal)

● = latent code $z^{(i)}$

easy to transform to 1D Gaussian

$\boldsymbol{\mu}_{\mathbf{z}}$

2D latent space, $\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$
(suboptimal)



difficult to transform samples on 1D manifold to a 2D Gaussian

Returning to the VAE …

Critical Questions:

- How does the VAE behave w.r.t. Criteria A (perfect reconstructions) and B (latent space distribution match)?
- And can we use this information to make improvements?

Perfect VAE Reconstructions (Criteria A)

Recall from Part III:

Theorem (Reconstruction Invariance):

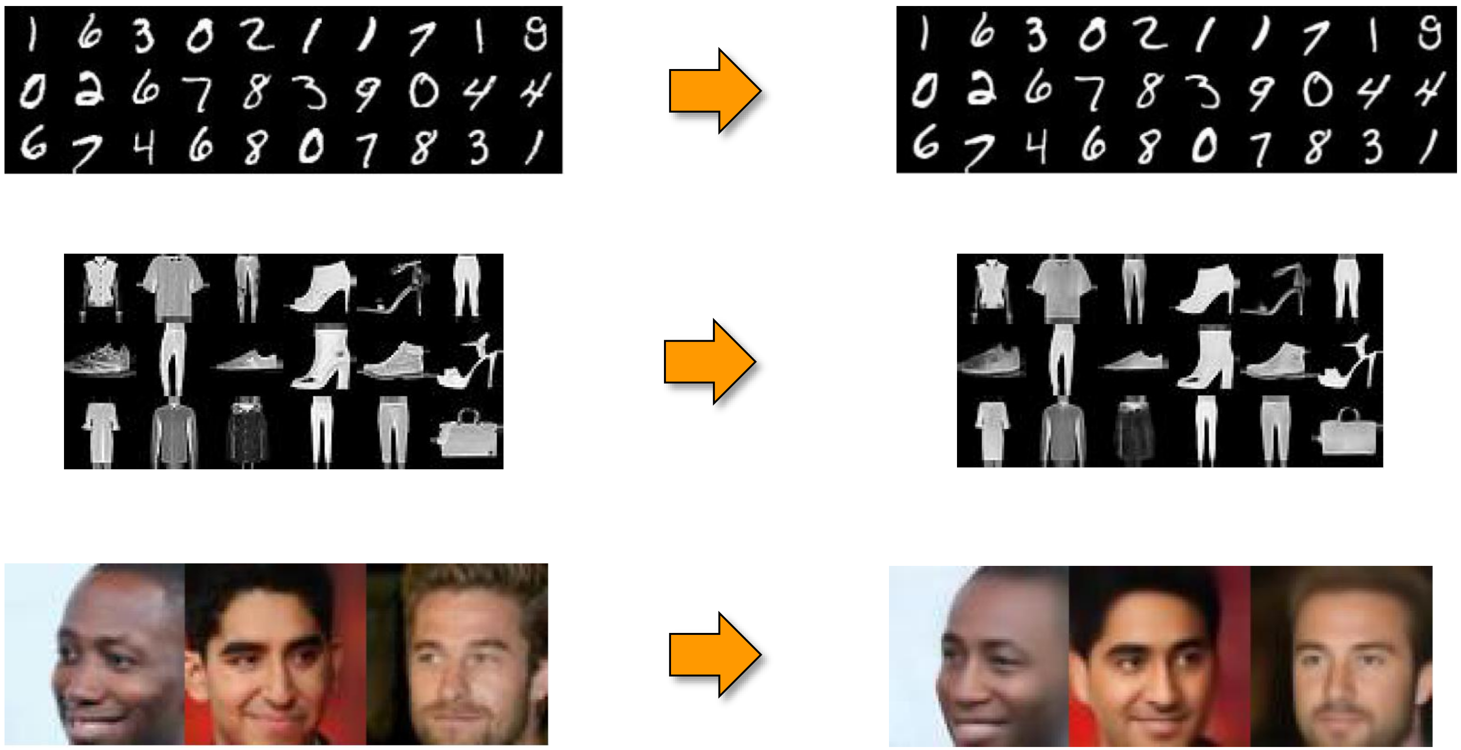
Under some technical conditions, any VAE global optimum $\{\boldsymbol{\theta}_*, \boldsymbol{\varphi}_*\}$ is such that $\gamma \rightarrow 0$ and reconstructions are exact:

$$\boldsymbol{\mu}_x\left(\boldsymbol{\mu}_z\left[\mathbf{x}^{(i)}, \boldsymbol{\varphi}_*\right] + \boldsymbol{\Sigma}_z^{1/2}\left[\mathbf{x}^{(i)}, \boldsymbol{\varphi}_*\right]\boldsymbol{\varepsilon}, \boldsymbol{\theta}_*\right) = \boldsymbol{\mu}_x\left(\boldsymbol{\mu}_z\left[\mathbf{x}^{(i)}, \boldsymbol{\varphi}_*\right], \boldsymbol{\theta}_*\right) = \mathbf{x}^{(i)}, \quad \forall \boldsymbol{\varepsilon}, \forall i$$

Key (rephrased) Conclusion: At global minimum, encoder randomness will not impact perfect reconstructions ⇨ VAE can satisfy Criteria A

Example Reconstructions

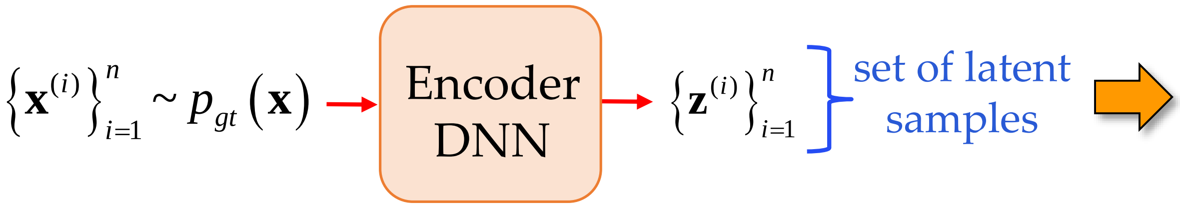
Ground Truth Samples VAE Reconstructions



So poor VAE performance may be related to Criteria B

Addressing the VAE Latent Space (Criteria B)

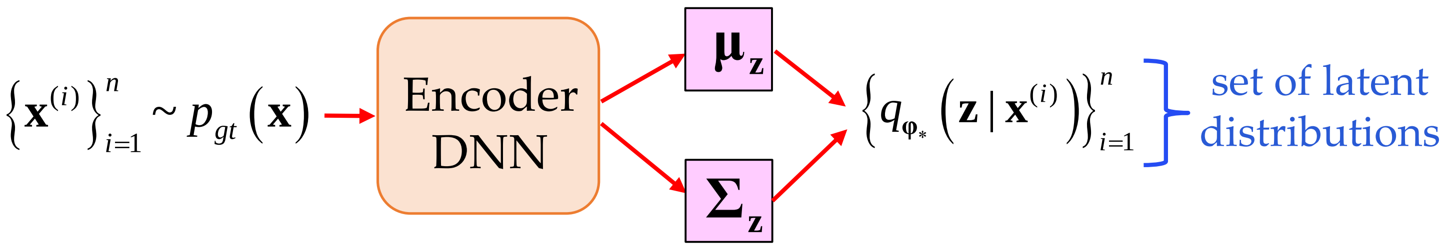
- Deterministic AE encoder:



Recall: Samples generally not close to $N(\mathbf{z}\,|\,\mathbf{0},\mathbf{I})$

fails Criteria B (without help…)

- Stochastic VAE encoder:



- Aggregated distribution of VAE latent space:

$$\underbrace{q_{\varphi_*}(\mathbf{z})}_{\text{aggregated posterior}} \triangleq \int q_{\varphi_*}(\mathbf{z}\,|\,\mathbf{x})\, p_{gt}(\mathbf{x})\, d\mathbf{x} \;\approx\; \tfrac{1}{n}\sum_{i=1}^{n} q_{\varphi_*}\left(\mathbf{z}\,|\,\mathbf{x}^{(i)}\right)$$

For generating good samples, should be close to $N(\mathbf{z}\,|\,\mathbf{0},\mathbf{I})$ } VAE version of Criteria B

Properties of VAE Aggregated Posterior

- When data lies on a manifold ($r < d$), at global minimum can have

$$q_{\varphi_*}(\mathbf{z}) \neq N(\mathbf{z} \mid \mathbf{0}, \mathbf{I})$$

➡ fails Criteria B

- But under reasonable assumptions, VAE aggregated posterior $q_{\varphi_*}(\mathbf{z})$ will satisfy conditions of Exact Density Recovery Theorem. (Note: samples from an AE generally will not)

- This means that a second VAE could be trained to learn $q_{\varphi_*}(\mathbf{z})$.

➡ implicitly addresses Criteria B

[Dai & Wipf, 2019]

Matching the VAE Aggregated Posterior

- From Exact Density Recovery Theorem, when $r = d$ we have

$$\mathrm{KL}\left[q_{\varphi_*}(\mathbf{z}|\mathbf{x}) \| p_{\theta_*}(\mathbf{z}|\mathbf{x})\right] = 0 \quad \text{and} \quad p_{\theta_*}(\mathbf{x}) = p_{gt}(\mathbf{x})$$

at any optimal solution, provided $\kappa \geq r$.

- This implies that:

$$q_{\varphi_*}(\mathbf{z}) \triangleq \int q_{\varphi_*}(\mathbf{z}|\mathbf{x})\, p_{gt}(\mathbf{x}) d\mathbf{x} = \int p_{\theta_*}(\mathbf{z}|\mathbf{x})\, p_{\theta_*}(\mathbf{x}) d\mathbf{x} = \int p_{\theta_*}(\mathbf{x}|\mathbf{z})\, N(\mathbf{z}|\mathbf{0},\mathbf{I}) d\mathbf{x} = N(\mathbf{z}|\mathbf{0},\mathbf{I})$$

perfect match!

- But when the data lie on a manifold (i.e., $r < d$), this no longer need be the case, i.e., $q_{\varphi_*}(\mathbf{z}) \neq N(\mathbf{z}|\mathbf{0},\mathbf{I})$

$$\boxed{\begin{matrix} \mathbf{z}^{(j)} \sim q_{\varphi_*}(\mathbf{z}) \\ \mathbf{x}^{(j)} \sim p_{\theta_*}\left(\mathbf{x}|\mathbf{z}^{(j)}\right) \end{matrix} \quad \neq \quad \begin{matrix} \mathbf{z}^{(j)} \sim N(\mathbf{z}|\mathbf{0},\mathbf{I}) \\ \mathbf{x}^{(j)} \sim p_{\theta_*}\left(\mathbf{x}|\mathbf{z}^{(j)}\right) \end{matrix}}$$

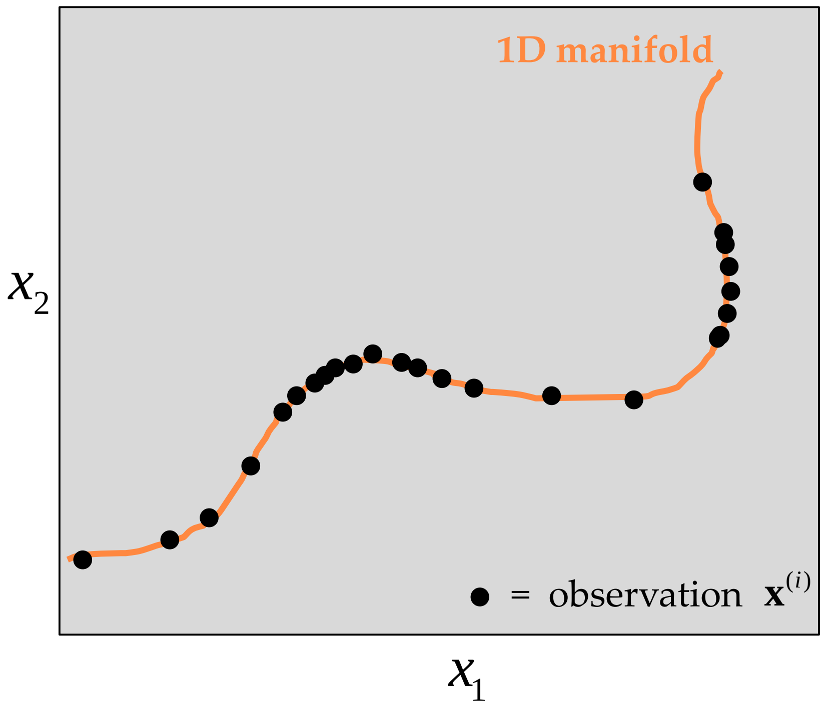
generates training data, but is intractable

?

- But intrinsic VAE properties suggest a practical solution …

2D Illustration

2D observation space, $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

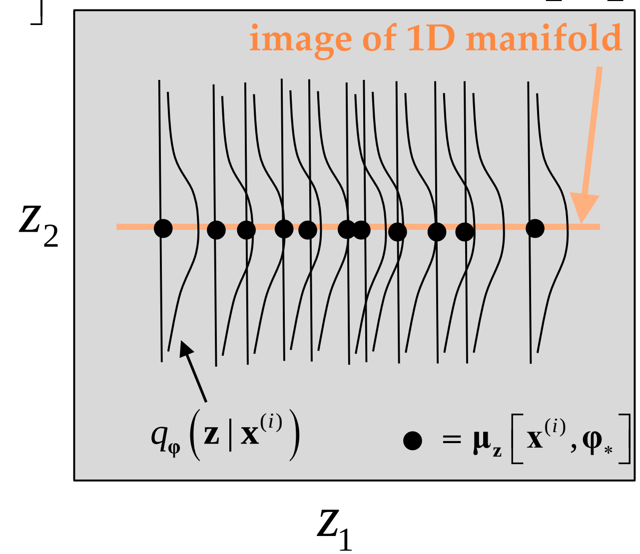


Optimal VAE encoder
mapping to 2D latent space

$$\boldsymbol{\mu}_{\mathbf{z}}\left[\mathbf{x}^{(i)}, \boldsymbol{\varphi}_*\right] \rightarrow \begin{bmatrix} \mu_z\left[\mathbf{x}^{(i)}, \boldsymbol{\varphi}_*\right]_1 \\ 0 \end{bmatrix}$$

$$\boldsymbol{\Sigma}_{\mathbf{z}}\left[\mathbf{x}^{(i)}, \boldsymbol{\varphi}_*\right] \rightarrow \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

2D VAE latent space, $\mathbf{z}$ $= \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$



$$\underbrace{q_{\boldsymbol{\varphi}_*}(\mathbf{z})}_{\text{aggregated posterior}} \triangleq \int q_{\boldsymbol{\varphi}_*}(\mathbf{z} \mid \mathbf{x})\, p_{gt}(\mathbf{x}) d\mathbf{x} \approx \tfrac{1}{n} \sum_{i=1}^{n} q_{\boldsymbol{\varphi}_*}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right)$$

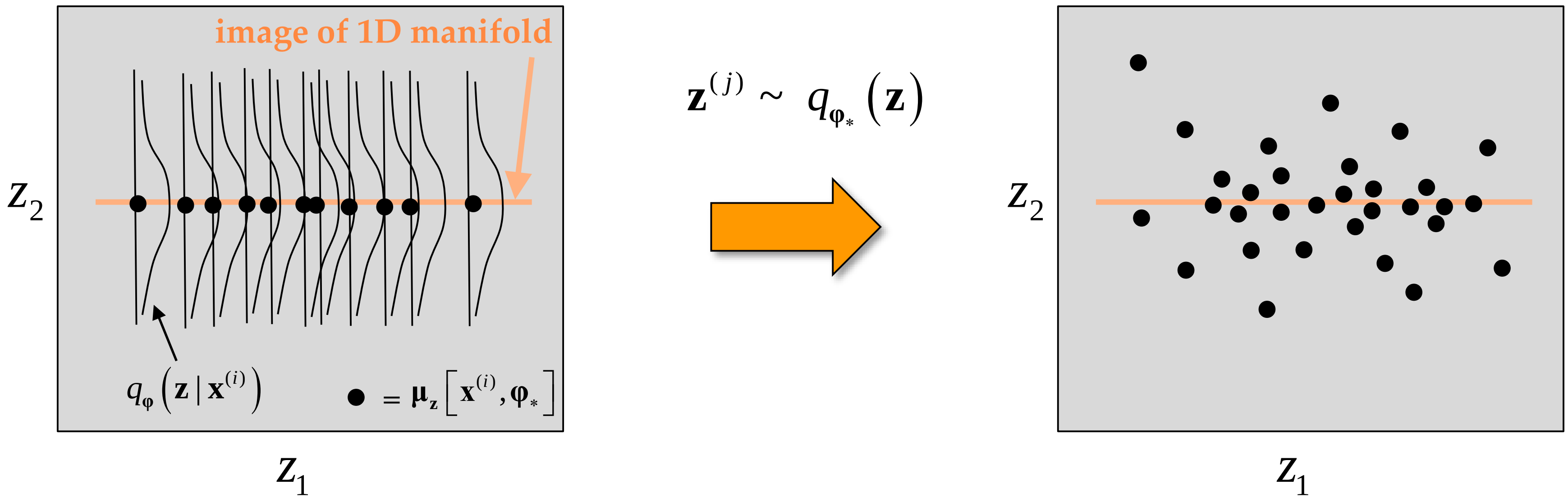
Aggregated posterior does not lie on a low-dim manifold as with deterministic AE

Comparing Aggregated Posterior Samples

$$q_{\varphi_*}(\mathbf{z}) \triangleq \int q_{\varphi_*}(\mathbf{z}\,|\,\mathbf{x})\, p_{gt}(\mathbf{x}) d\mathbf{x} \approx \tfrac{1}{n}\sum_{i=1}^{n} q_{\varphi_*}\left(\mathbf{z}\,|\,\mathbf{x}^{(i)}\right)$$

2D VAE latent space, $\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$

Aggregated posterior samples: $\left\{\mathbf{z}^{(j)}\right\}_{j=1}^{m}$



Remarks:

- Still no guarantee that the aggregated posterior will be close to $N(\mathbf{0},\mathbf{I})$
- But samples will **not** lie on a low-dimensional manifold
- The VAE decoder "fills out" unnecessary dimensions with random noise (Part III)
- This leads to a simple 2-stage VAE enhancement based on Exact Density Recovery Theorem from earlier …

Two-Stage VAE Strategy

$$\mathbf{X} = \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{n}, \quad \mathbf{x}^{(i)} \in \mathcal{X} \subset \mathbb{R}^{d}, \; r\text{-dim manifold}, \quad r < d$$

- Choose $\dim(\mathbf{z}) = \kappa$ sufficiently large, ensure $\kappa \geq r$ (do not need exact value)
- Solve via SGD: $\boldsymbol{\theta}_*, \boldsymbol{\varphi}_* = \arg\min_{\boldsymbol{\theta},\boldsymbol{\varphi}} L(\boldsymbol{\theta}, \boldsymbol{\varphi})$ → first-stage VAE
- Form aggregated posterior approximation: $q_{\boldsymbol{\varphi}_*}(\mathbf{z}) \approx \frac{1}{n}\sum_{i=1}^{n} q_{\boldsymbol{\varphi}_*}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right)$
- Samples from this approximation form new data set:

$$\left.\mathbf{Z} = \left\{\mathbf{z}^{(j)}\right\}_{j=1}^{m}, \quad \mathbf{z}^{(j)} \sim \frac{1}{n}\sum_{i=1}^{n} q_{\boldsymbol{\varphi}_*}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right)\right\}$$

latent codes associated with training data; no manifold structure

- This is regime where Exact Density Recovery Theorem applies
- Train a second VAE on data $\mathbf{Z}$ with latent code $\mathbf{u}$, and $\dim(\mathbf{u}) = \kappa$

$$\boldsymbol{\theta}_{2*}, \boldsymbol{\varphi}_{2*} = \arg\min_{\boldsymbol{\theta}_2,\boldsymbol{\varphi}_2} L(\boldsymbol{\theta}_2, \boldsymbol{\varphi}_2)$$

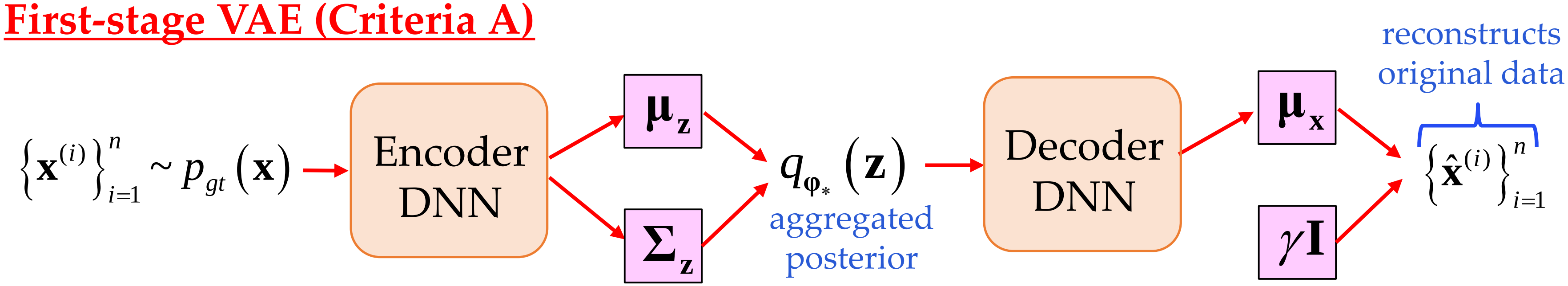
→ second-stage VAE (much smaller)

- By design, this VAE will (asymptotically) learn the exact aggregated posterior from the first-stage VAE

[Dai & Wipf, 2019]

Two-Stage VAE Visualization

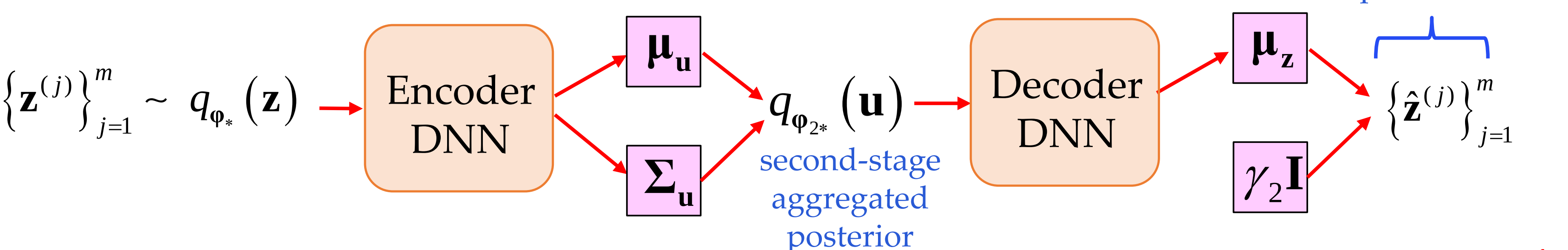
First-stage VAE (Criteria A)



$$q_{\varphi_*}(\mathbf{z}) \neq N(\mathbf{z}\,|\,\mathbf{0},\mathbf{I})$$

Second-stage VAE (Criteria B)

samples approximate first-stage aggregated posterior



$$q_{\varphi_{2*}}(\mathbf{u}) \approx \frac{1}{m}\sum_{j=1}^{m} q_{\varphi_{2*}}(\mathbf{u}\,|\,\mathbf{z}^{(j)}) \approx N(\mathbf{u}\,|\,\mathbf{0},\mathbf{I})$$

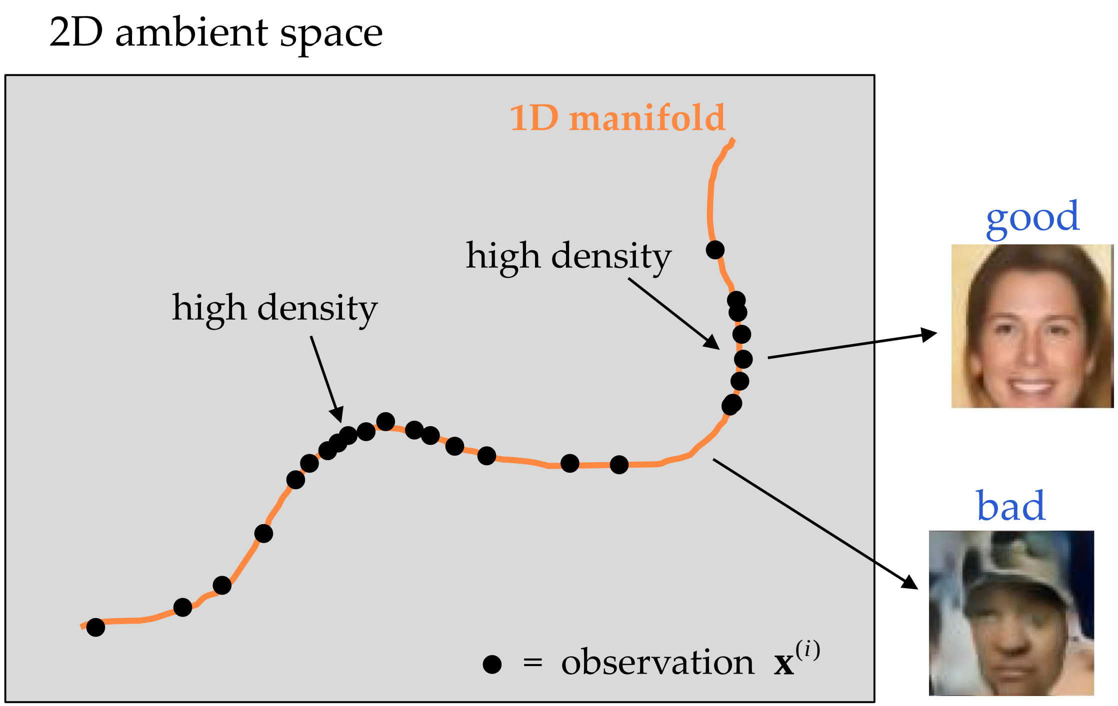
consequence of Exact Density Recovery Theorem

Generating new samples is now trivial:

$$\underbrace{\mathbf{u}^{new} \sim N(\mathbf{u}\,|\,\mathbf{0},\mathbf{I}),\;\; \mathbf{z}^{new} \sim p_{\theta_{2*}}(\mathbf{z}\,|\,\mathbf{u}^{new})}_{\mathbf{z}^{new} \sim q_{\varphi_*}(\mathbf{z})},\;\; \mathbf{x}^{new} \sim p_{\theta_*}(\mathbf{x}\,|\,\mathbf{z}^{new})$$

[Dai & Wipf, 2019]

Two-Stage VAE Intuition



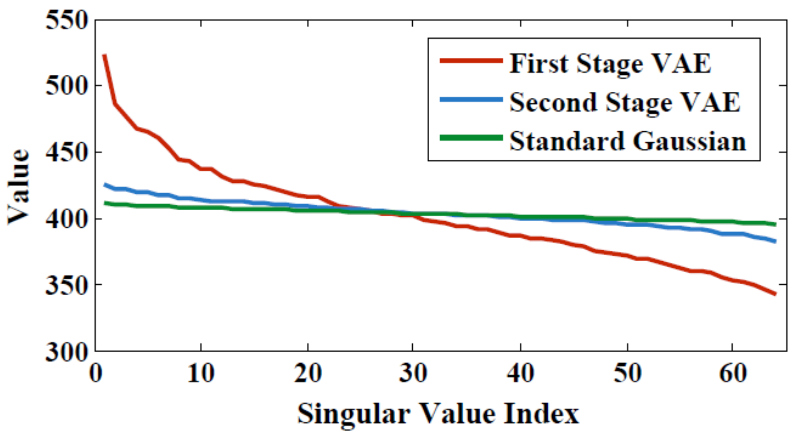
- First-stage VAE learns manifold model by efficiently reconstructing samples (analogous to Criteria A).
- Second-stage VAE learns distribution within the manifold (analogous to Criteria B).
- Note: Joint training does not work in this context.

Aggregated Posterior Comparisons

First-stage latent codes: $\mathbf{Z} = \left\{\mathbf{z}^{(j)}\right\}_{i=1}^{m}, \quad \mathbf{z}^{(j)} \sim q_{\hat{\varphi}}(\mathbf{z}) \approx \frac{1}{n}\sum_{i=1}^{n} q_{\hat{\varphi}}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right)$

Second-stage latent codes: $\mathbf{U} = \left\{\mathbf{u}^{(j)}\right\}_{i=1}^{m}, \quad \mathbf{u}^{(j)} \sim q_{\hat{\varphi}_2}(\mathbf{u}) \approx \frac{1}{m}\sum_{j=1}^{m} q_{\hat{\varphi}_2}\left(\mathbf{u} \mid \mathbf{z}^{(j)}\right)$

Singular Value Comparison



MMD from ideal $N(0,I)$

	First Stage	Second Stage
MNIST	2.85	0.43
Fashion	1.37	0.40
Cifar10	1.08	0.00
CelabA	7.42	0.29

low values; close to Gaussian

Two-Stage VAE Results

quantitative measure of perceptual quality; lower is better

Averaged FID Score Comparisons

Neutral testing conditions from [Lucic et al., 2018]

		MNIST	Fashion	CIFAR-10	CelebA
optimized, data-dependent settings	MM GAN	9.8 ± 0.9	29.6 ± 1.6	72.7 ± 3.6	65.6 ± 4.2
	NS GAN	6.8 ± 0.5	26.5 ± 1.6	58.5 ± 1.9	55.0 ± 3.3
	LSGAN	7.8 ± 0.6	30.7 ± 2.2	87.1 ± 47.5	53.9 ± 2.8
	WGAN	6.7 ± 0.4	21.5 ± 1.6	55.2 ± 2.3	41.3 ± 2.0
	WGAN GP	20.3 ± 5.0	24.5 ± 2.1	55.8 ± 0.9	30.3 ± 1.0
	DRAGAN	7.6 ± 0.4	27.7 ± 1.2	69.8 ± 2.0	42.3 ± 3.0
	BEGAN	13.1 ± 1.0	22.9 ± 0.9	71.4 ± 1.6	38.9 ± 0.9
default settings	Best GAN	~ 10	~ 32	~ 70	~ 49
	VAE (cross-entr.)	16.6 ± 0.4	43.6 ± 0.7	106.0 ± 1.0	53.3 ± 0.6
	VAE (fixed γ)	52.0 ± 0.6	84.6 ± 0.9	160.5 ± 1.1	55.9 ± 0.6
	VAE (learned γ)	54.5 ± 1.0	60.0 ± 1.1	76.7 ± 0.8	60.5 ± 0.6
	VAE + Flow	54.8 ± 2.8	62.1 ± 1.6	81.2 ± 2.0	65.7 ± 2.8
	WAE-MMD	115.0 ± 1.1	101.7 ± 0.8	80.9 ± 0.4	62.9 ± 0.8
	2-Stage VAE	12.6 ± 1.5	29.3 ± 1.0	72.9 ± 0.9	44.4 ± 0.7

similar to GANs, no tuning

WAE testing conditions from [Tolstikhin et al., 2018]

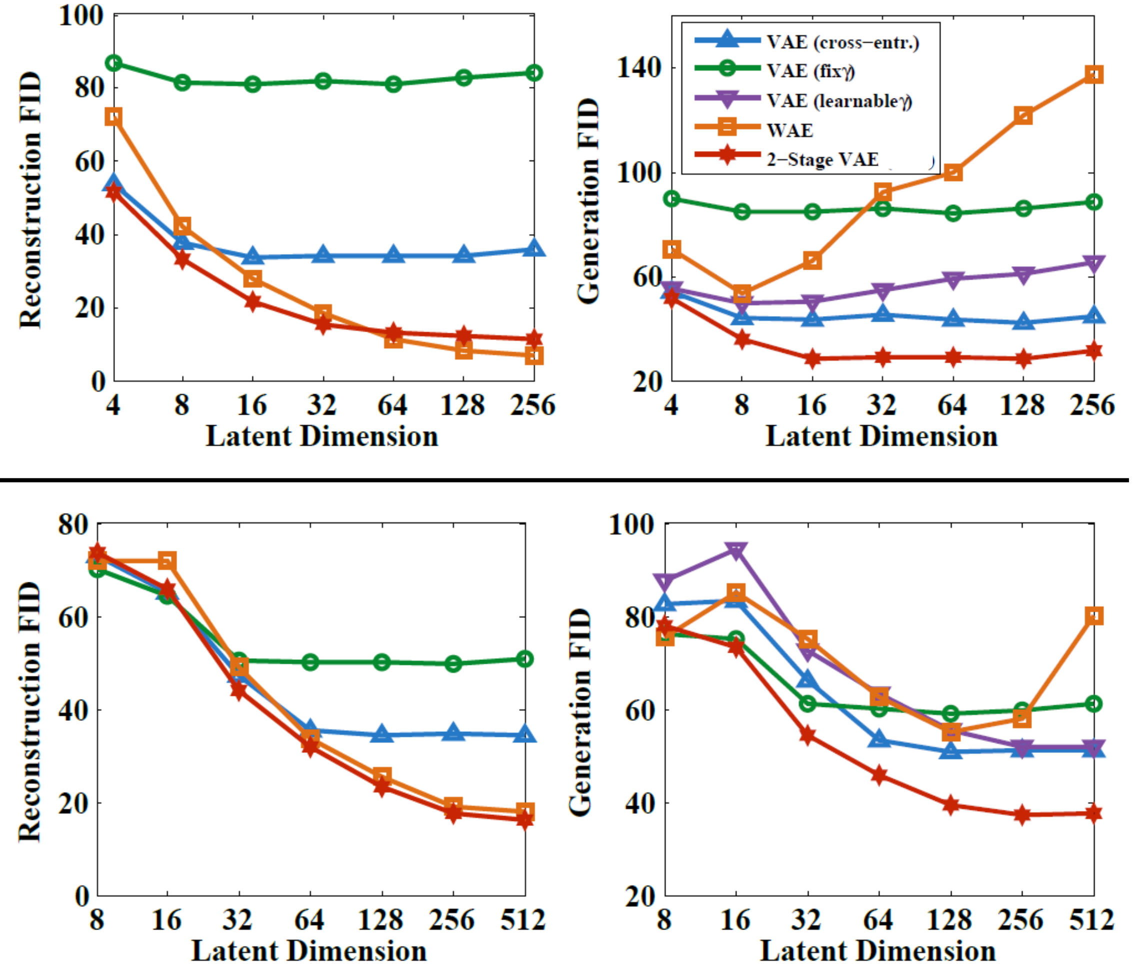
	VAE	WAE-MMD	WAE-GAN	2-Stage VAE
CelebA FID	63	55	42	34

improvement over WAE

Robustness to the Latent Space Dimension

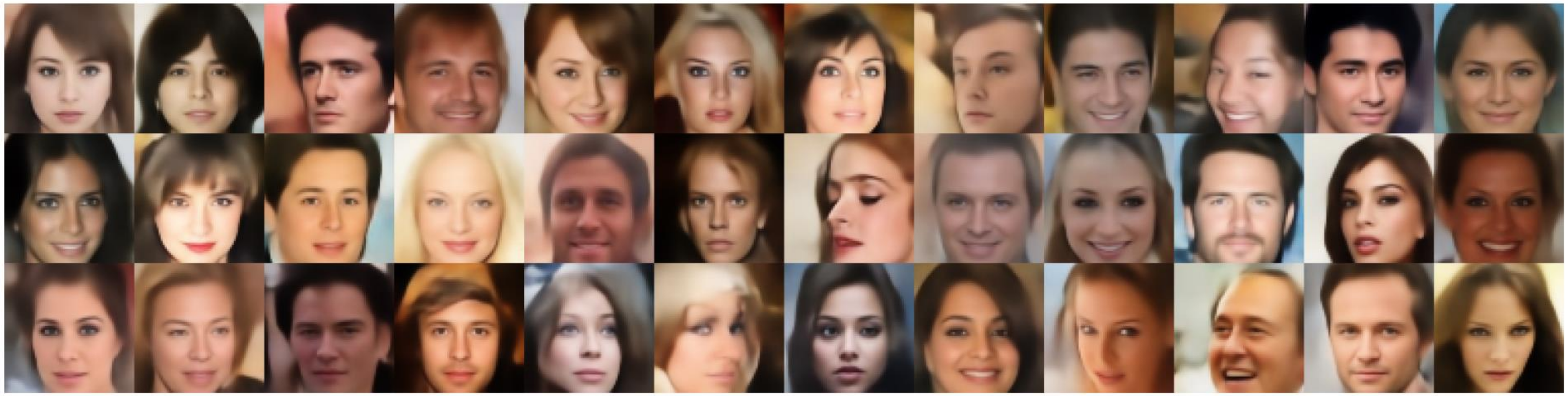
Fashion MNIST

CelebA



Comparison of Generated CelebA Samples

Two-Stage VAE

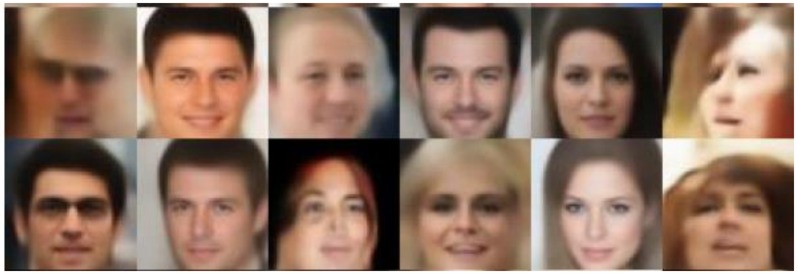


Single-Stage VAE (learned γ)

good \ clear reconstructions, poor new samples

Single-Stage VAE (fixed γ)

bad \ blurry reconstructions, poor new samples



WGAN-GP [Gulrajani et al., 2017]

https://github.com/LynnHo/WGAN-GP-DRAGAN-Celeba-Pytorch

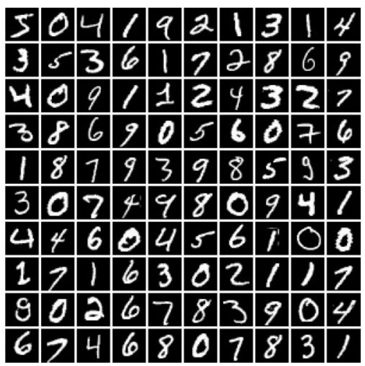
MNIST Example with Corruptions

Note: MNIST data is much simpler than CelebA, but with corruptions it is challenging to generate new clean samples

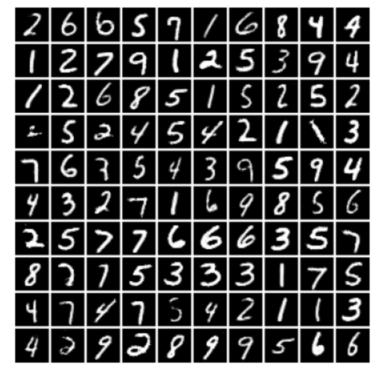
Originals

Enhanced VAE
$\boldsymbol{\Sigma}_{\mathbf{x}}\left[\mathbf{z},\boldsymbol{\theta}\right]$ arbitrary + recycling (Part III)

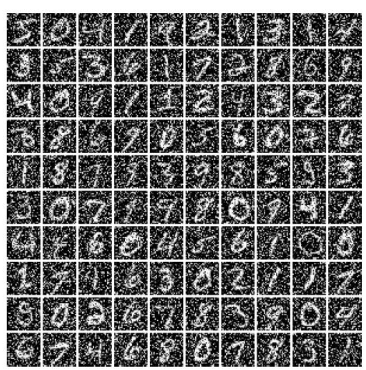
Noise-free



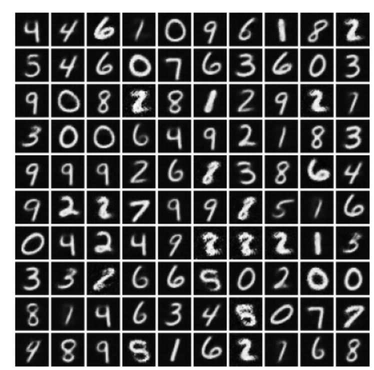
new samples, easy case



Corrupted



new samples, ignoring noise



clean generated samples, even though training data fully corrupted

Summary

- Standard VAE can reconstruct data lying on a low-dimensional manifold (first-stage VAE).

- But generated samples may not resemble training data.

- Fortunately, the distribution of the VAE latent codes can be successfully modeled and sampled from (second-stage VAE).

- Combined stages can produce more realistic samples, comparable to many GANs (w/ same neutral architecture).

- But two-stage model retains original VAE advantages (and additional complexity is minimal, second-stage can be small).

- Alternative VAE-inspired approaches like the WAE can also produce good results (but may be more sensitive to latent dimensions).

Questions?

Part V:
Practical Usage Issues and Examples

Note: Updated version of slides available at http://www.davidwipf.com/

Outline

- ❑ Cases of over- and under-regularization
- ❑ Identifiability of semantically-meaningful latent factors
- ❑ Practical applications via Conditional VAEs

Over-Regularized/Degenerate VAE Local Solutions

VAE Objective:

$$L(\boldsymbol{\theta},\boldsymbol{\varphi}) = \sum_i \Big\{ \underbrace{\mathrm{KL}\left[q_{\boldsymbol{\varphi}}\left(\mathbf{z}\,|\,\mathbf{x}^{(i)}\right) \| N\left(\mathbf{z}\,|\,\mathbf{0},\mathbf{I}\right)\right]}_{\text{KL term has trivial minimum, only requires parameters of last encoder layer}} - \underbrace{\mathrm{E}_{q_{\boldsymbol{\varphi}}\left(\mathbf{z}|\mathbf{x}^{(i)}\right)}\left[\log p_{\boldsymbol{\theta}}\left(\mathbf{x}^{(i)}\,|\,\mathbf{z}\right)\right]}_{\text{minimizing data term requires complex coordination of all parameters in encoder/decoder networks (hard)}} \Big\}$$

$$\sum_{j=1}^{\kappa}\left\{\mu_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]_j^2 + \sigma_{\mathbf{z}}^2\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]_j - \log\left(\sigma_{\mathbf{z}}^2\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]_j\right)\right\}$$

trivial minimum ➡

$$\mu_{\mathbf{z}}\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]_j^2 \to 0$$

$$\sigma_{\mathbf{z}}^2\left[\mathbf{x}^{(i)},\boldsymbol{\varphi}\right]_j \to 1$$

Potential for convergence to bad, overregularized (local) solutions

Candidate workarounds:

- KL warm-start [Bowman et al., 2015; Sønderby et al., 2016]
- Skip connections [Cai et al., 2017; Dieng et al., 2018]
- Ladder networks [Sønderby et al., 2016; Maaløe et al., 2019]

Under-Regularized VAE Global Optima

- In principle, VAE encoder can be arbitrarily complex; this just tightens the original upper bound

$$-\sum_i \log p_{\theta}\left(\mathbf{x}^{(i)}\right) \leq \sum_i \left\{ \underbrace{\mathrm{KL}\left[q_{\varphi}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right) \| p_{\theta}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right)\right]}_{\to 0 \quad \text{w/ complex encoder}} - \log p_{\theta}\left(\mathbf{x}^{(i)}\right) \right\}$$

- Likewise, decoder covariance can be arbitrarily complex to learn outlier locations (Part III).

- But the decoder mean network is more subtle …

Problem: While the VAE cost does penalize excessive dimensions of $\mathbf{z}$ (Part III), it cannot prevent overfitting from excessive depth.

Theorem

> Even with $\dim(\mathbf{z}) = 1$, VAE cost can be globally optimized by solution that just memorizes the training data if the decoder mean is too complex.

[Dai et al., 2018]

Outline

- ❑ Cases of over- and under-regularization
- ❑ Identifiability of semantically-meaningful latent factors
- ❑ Practical applications via Conditional VAEs

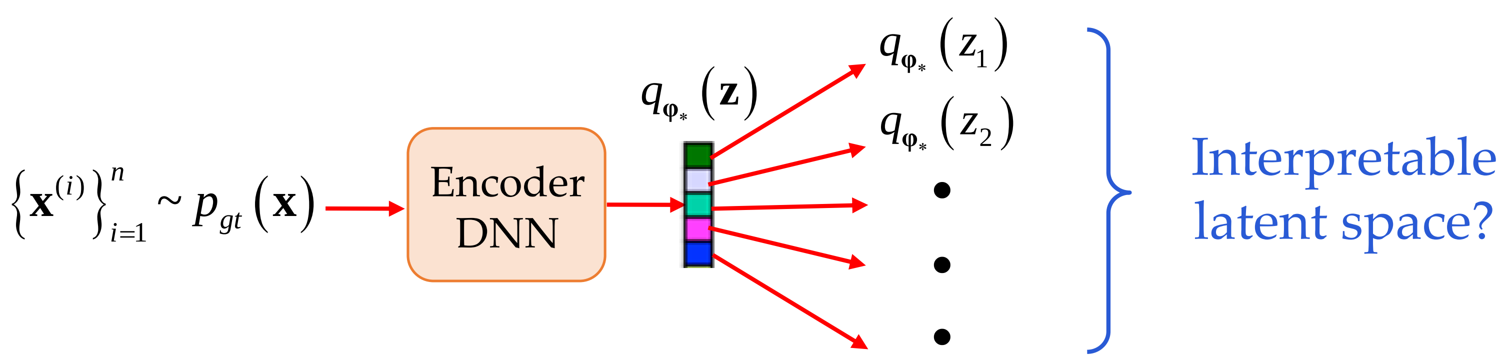
Interpretability of the VAE Latent Space

If VAE training is successful, then:

$$q_{\varphi_*}(\mathbf{z}) \triangleq \int q_{\varphi_*}(\mathbf{z} \mid \mathbf{x})\, p_{gt}(\mathbf{x})\, d\mathbf{x} \approx \tfrac{1}{n}\sum_{i=1}^{n} q_{\varphi_*}\left(\mathbf{z} \mid \mathbf{x}^{(i)}\right) \approx N(\mathbf{z} \mid \mathbf{0}, \mathbf{I})$$

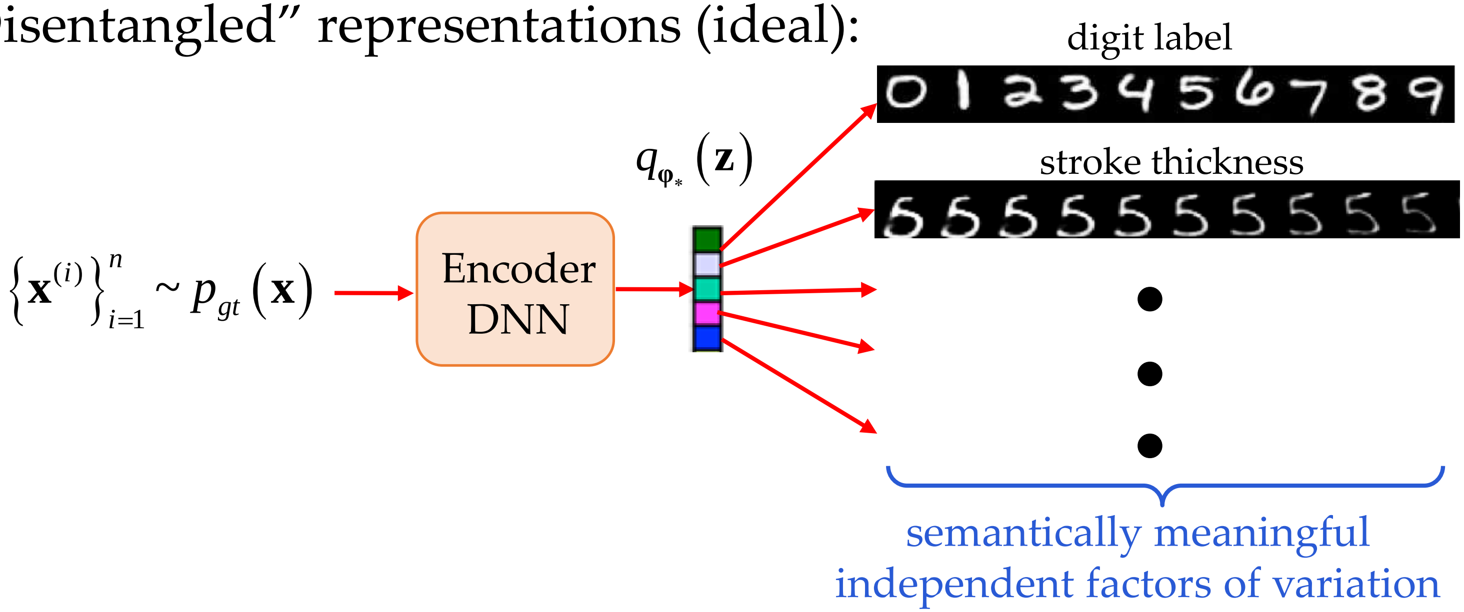
$$q_{\varphi_*}(\mathbf{z}) \approx \prod_{j=1}^{\kappa} q_{\varphi_*}(z_j)$$

Independent latent factors of variation:

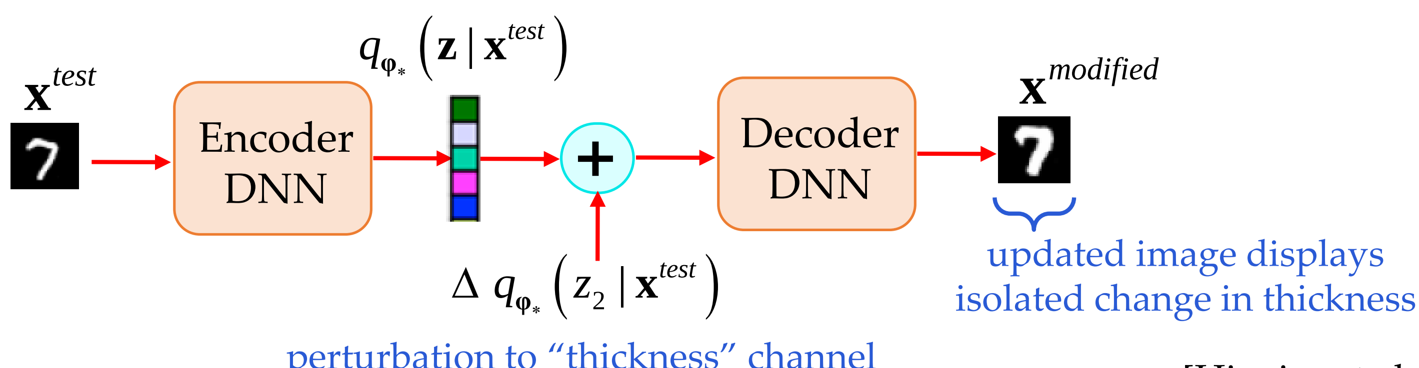


Independence vs. Semantic Meaning

"Disentangled" representations (ideal):



Useful for numerous computer vision, image processing applications, e.g., photo editing/manipulation:



[Higgins et al., 2017]

Identifiability Issues

Assume the ground-truth generative process:

$$\mathbf{z}^{(i)} \sim p_{gt}(\mathbf{z}), \quad \mathbf{x}^{(i)} = \underbrace{f_{gt}\left(\mathbf{z}^{(i)}\right)}_{\text{arbitrary deterministic decoder}}, \quad i = 1, \ldots, n$$

Also assume a disentangled latent density:

$$p_{gt}(\mathbf{z}) = \prod_j p_{gt}(z_j) \Rightarrow$$

semantically meaningful factors of variation

Identifiability Problem: Exact same samples can be generated using a simple transformed process …

$$\mathbf{z}^{(i)} \sim p_{gt}(\mathbf{z})$$

$$\tilde{\mathbf{z}}^{(i)} \triangleq D_2\left[\mathbf{R} \cdot D_1\left(\mathbf{z}^{(i)}\right)\right] \Rightarrow$$

transformation to arbitrary marginals

rotation

transformation to $N(\mathbf{z} \mid \mathbf{0}, \mathbf{I})$

new latent factorial distribution by design

$$\tilde{\mathbf{z}}^{(i)} \sim \tilde{p}(\tilde{\mathbf{z}}) = \prod_j \underbrace{\tilde{p}(\tilde{z}_j)}_{\text{mix of previous disentangled factors}}$$

$$\mathbf{x}^{(i)} = \underbrace{f_{gt}\left(D_1^{-1}\left[\mathbf{R}^T \cdot D_2^{-1}\left(\tilde{\mathbf{z}}^{(i)}\right)\right]\right)}_{\text{composite decoder inverts transformation}} = f_{gt}\left(\mathbf{z}^{(i)}\right)$$

Same samples as before, <u>no</u> unique disentangled representation

[Locatello et al., 2019; Dai & Wipf, 2019]

Trivial Discrete Example

$\mathbf{x}_{FY}$ $\mathbf{x}_{FO}$ $\mathbf{x}_{MY}$ $\mathbf{x}_{MO}$

Data set of 4 equiprobable images:

Two latent attributes: gender {female, male}, age {young,old}

Candidate 2D latent codes

	Disentangled Representation $z_1\ z_2$		Entangled Representation $\tilde{z}_1\ \tilde{z}_2$	
$\mathbf{x}_{FY}$	$\boxed{0\ 0}$	one attribute changes	$\boxed{0\ 0}$	both attributes change
$\mathbf{x}_{FO}$	0 1		1 1	
$\mathbf{x}_{MY}$	$\boxed{1\ 0}$		0 1	
$\mathbf{x}_{MO}$	1 1		$\boxed{1\ 0}$	

$p_{gt}(\mathbf{z}) = p_{gt}(z_1)\, p_{gt}(z_2)$ $\tilde{p}(\tilde{\mathbf{z}}) = \tilde{p}(\tilde{z}_1)\, \tilde{p}(\tilde{z}_2)$

both latent distributions factorize ➡ **not identifiable**

Workarounds

- Constraints on the ground-truth generative process, e.g.,

$$p_{gt}(\mathbf{z}) = \prod_{j=1}^{r} p_{gt}(z_j) \neq N(\mathbf{z} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$p_{gt}(\mathbf{x} \mid \mathbf{z}) = N(\mathbf{x} \mid \mathbf{W}\mathbf{z}, \gamma \mathbf{I})$$

linear ground-truth decoder

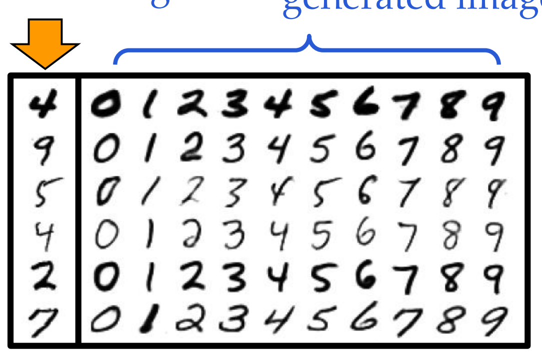
VAE model: Use linear decoder mean network and non-Gaussian (possibly parameterized) prior $p_{\boldsymbol{\theta}}(\mathbf{z})$

Leads to ICA-like model

identifiable up to permutation and scaling

[Hyvärinen et al., 2001]

- Apply some form of weak supervision or semi-supervised learning to resolve ambiguity, e.g.,



[Kingma et al., 2014]

Outline

- Cases of over- and under-regularization
- Identifiability of semantically-meaningful latent factors
- Practical applications via Conditional VAEs

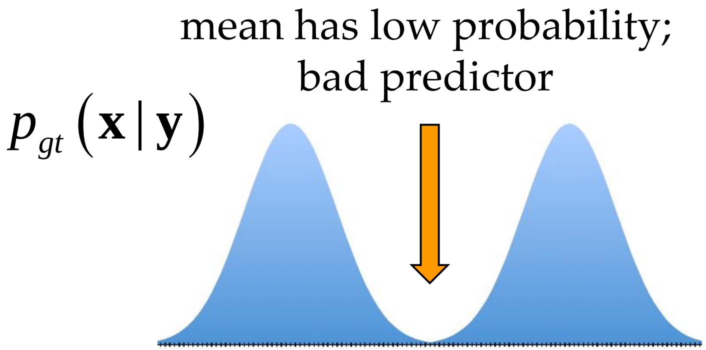
Conditional VAEs

- Often want a generative model for data conditioned on some variable of interest, e.g.,

$$p_{gt}(\mathbf{x}|\mathbf{y}) = \int p_{gt}(\mathbf{x}|\mathbf{z},\mathbf{y})\, \underbrace{p_{gt}(\mathbf{z})}_{\text{independent of } \mathbf{y}} d\mathbf{z}$$

- Basic VAE derivations go through as before, with extra conditioning variable $\mathbf{y}$.

- Many applications, e.g., structured output prediction:

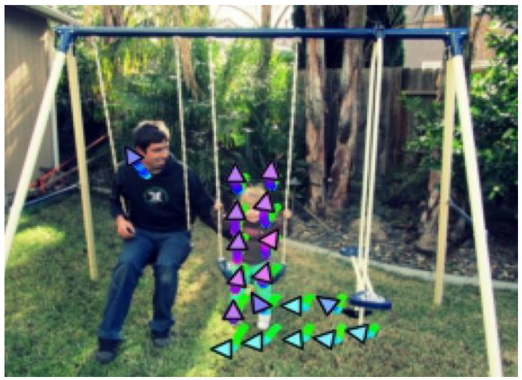


[Sohn et al., 2015]

Example Applications

- Forecasting possible motions from static images:

$$p_{gt}(\mathbf{x}|\mathbf{y}), \quad \mathbf{y} = \text{static image}, \quad \mathbf{x} = \text{dense motion trajectory}$$



[Walker et al., 2016]

- Image Captioning:

$$p_{gt}(\mathbf{x}|\mathbf{y}), \quad \mathbf{y} = \text{image}, \quad \mathbf{x} = \text{caption}$$

a woman sitting at a table with a cup of coffee
a person sitting at a table with a cup of coffee
a table with two plates of donuts and a cup of coffee
a woman sitting at a table with a plate of coffee
a man sitting at a table with a plate of food

[Wang et al., 2017]

Final Thoughts

- The VAE represents a natural extension of many existing signal processing, dimensionality reduction tools

- This is complementary to its role capability as a generative model

- Many diverse applications, algorithmic variants, extensions …

Questions?