Auto-Encoding Variational Bayes

Diederik P (Durk) Kingma, Max Welling
University of Amsterdam
Ph.D. Candidate, advised by Max

Durk Kingma

Max Welling

Problem class

- Directed graphical model:

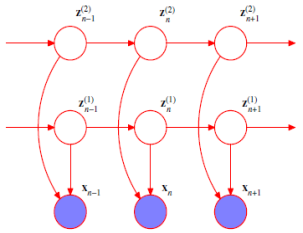
  x : observed variable

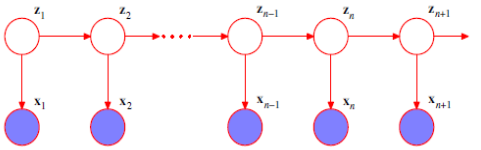
  z : latent variables (continuous)

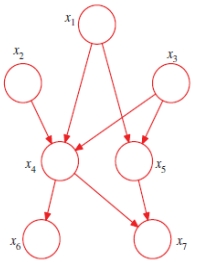
  θ : model parameters

  $p_{\theta}(\mathbf{x},\mathbf{z})$: joint PDF

  - Factorized, differentiable







- Goal: learning parameters **θ**
  - Requires inference over latent variables $p_{\theta}(\mathbf{z}|\mathbf{x})$

- Hard case: intractable posterior distribution $p_{\theta}(\mathbf{z}|\mathbf{x})$

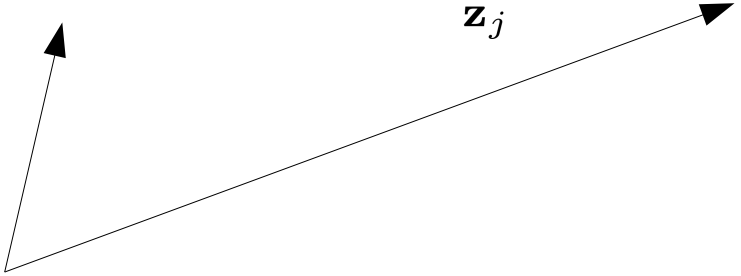
  e.g. neural nets as components

- We propose a method for fast approximate posterior inference per datapoint
  - After inference, learning params is easy

Gradients of log-posterior

- Gradients of log-posterior

$$
\begin{aligned}
p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) &= p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})/p_{\boldsymbol{\theta}}(\mathbf{x}) \\
p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) &\propto p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}) \\
\nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) &= \nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z}) \\
&= \sum_{\mathbf{x}_j} \nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{x}_j|pa(\mathbf{x}_j)) + \sum_{\mathbf{z}_j} \nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{z}_j|pa(\mathbf{z}_j))
\end{aligned}
$$



Conditionals can be given
by feedforward neural networks

Gradient-based Approximate Inference/Learning methods

- MCMC / Monte Carlo EM
  - often too slow / scaling issues
- Wake-Sleep
  - Improper
- Pure MAP / Maximization?
  - Heavlily overfits with high dimensional z
- Stochastic Variational Inference
  - Fit approximate posterior $q_\phi(z|x)$

Auto-Encoding Variational Bayes

Idea:

- **Fit an approximate the posterior**
  - $q_{\varphi}(z|x)$ with 'variational parameters' φ
  - one-shot approximate inference

- **Construct estimator of the variational lower bound** which we can optimize jointly w.r.t. $\boldsymbol{\varphi}$ jointly with $\boldsymbol{\theta}$

  -> Stochastic gradient ascent

Variational Lower Bound of the marg. lik.

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = KL(q_{\mathbf{z}|\mathbf{x}}||p_{\mathbf{z}|\mathbf{x}}) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x})$$

$$\begin{aligned} \text{where} \quad \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\right) \\ &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[f_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{z})\right] \end{aligned}$$

Monte Carlo estimator of the variational bound

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \left[f_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{z}) \right] \simeq \frac{1}{L} \sum_{l=1}^{L} f_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{z}^{(l)})$$

$$\text{where} \quad \mathbf{z}^{(l)} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \quad \text{(samples)}$$

Can we differentiate through the sampling process w.r.t. φ ?

Key reparameterization trick

Construct samples $z \sim q_\varphi(z|x)$ in two steps:

1. $\varepsilon \sim p(\varepsilon)$ *(random seed independent of φ)*
2. $z = g(\varphi, \varepsilon, x)$ (differentiable perturbation)

such that $z \sim q_\varphi(z|x)$ (the correct distribution)

Examples:

- if $q(z|x) \sim N(\mu(x), \sigma(x)^2)$
  $\varepsilon \sim N(0,I)$
  $z = \mu(x) + \sigma(x) * \varepsilon$
- (approximate) Inverse CDF
- Much more possibilities (see paper)

SGVB estimator

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = \int q_{\boldsymbol{\phi}}(\mathbf{z}) \left[\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}) \right] d\mathbf{z}$$

$$\simeq \frac{1}{L} \sum_{l=1}^{L} \left(\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}^{(l)}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}^{(l)}) \right)$$

$$\begin{aligned} \text{where} \quad & \boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon}) \quad \text{(samples from noise variable)} \\ & \mathbf{z}^{(l)} = g(\boldsymbol{\epsilon}^{(l)}, \boldsymbol{\phi}) \\ \text{(such that} \quad & \mathbf{z}^{(l)} \sim q_{\boldsymbol{\phi}}(\mathbf{z})) \end{aligned}$$

Really simple and appropriate for differentiation w.r.t. **φ** and **θ!**

Auto-Encoding Variational Bayes
Online algorithm

repeat

$$\mathbf{x} \leftarrow \text{random datapoint or minibatch}$$

$$\boldsymbol{\epsilon} \leftarrow \text{sample from } p(\boldsymbol{\epsilon})$$

$$g_{\boldsymbol{\theta}}, g_{\boldsymbol{\phi}} \leftarrow \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}, g(\boldsymbol{\epsilon}, \boldsymbol{\phi}))$$

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \cdot g_{\boldsymbol{\theta}}$$

$$\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \alpha \cdot g_{\boldsymbol{\phi}}$$

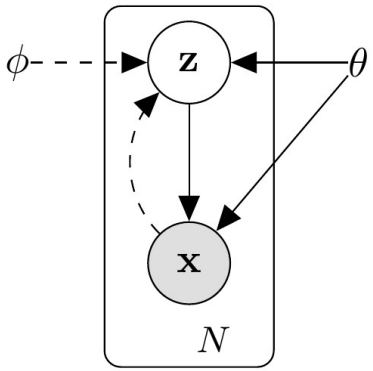
until convergence

Backprop
(Torch7 / Theano)

e.g. Adagrad

Scales to very large datasets!

Model used in experiments



$$p_{\boldsymbol{\theta}}(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$$

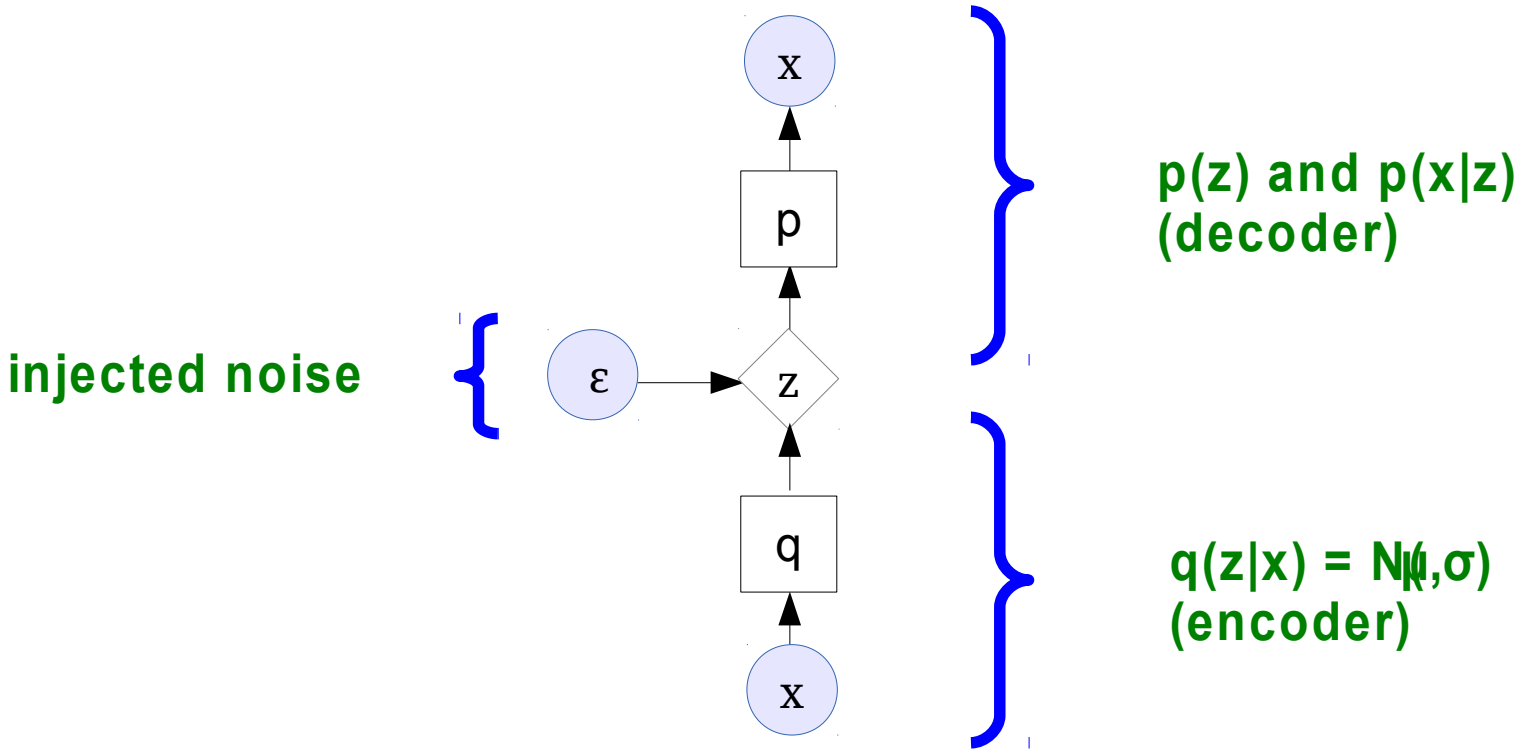
$$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\boldsymbol{\mu}(\mathbf{z}), \sigma(\mathbf{z})\mathbf{I})$$

$$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}), \sigma(\mathbf{x})\mathbf{I})$$

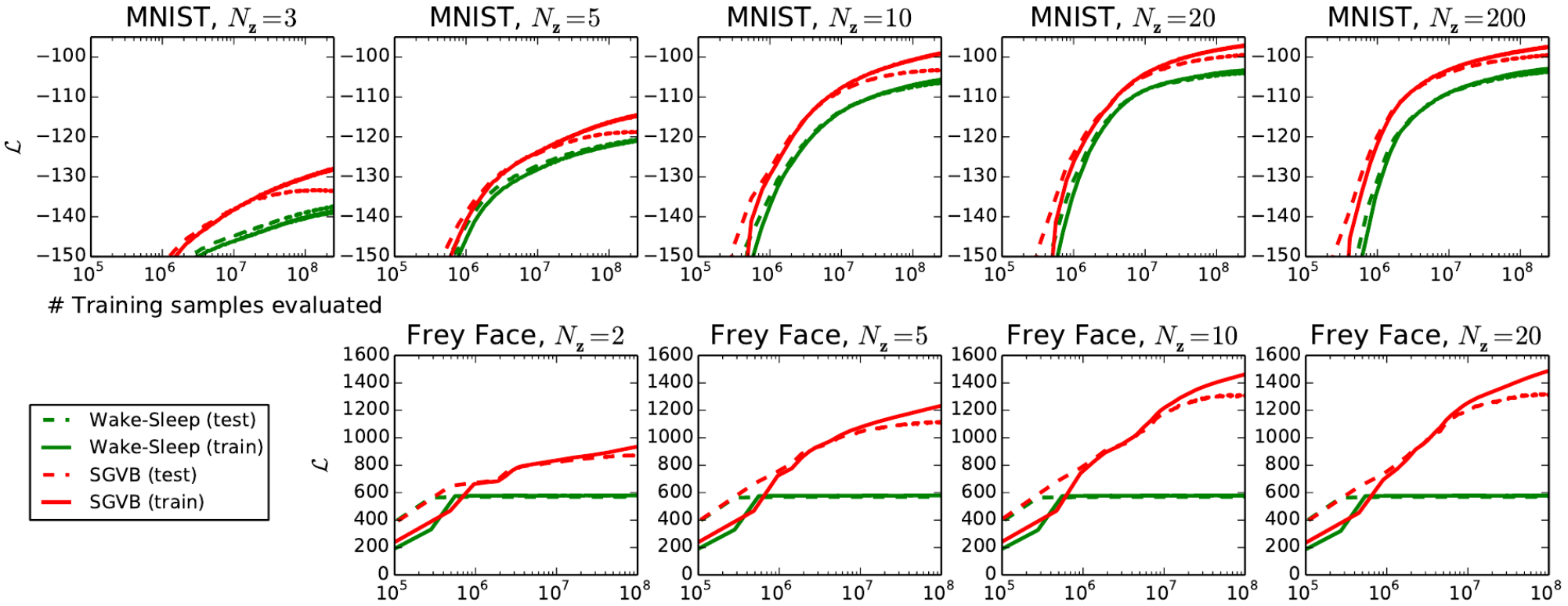
$$\widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) = \underbrace{\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}^{(l)})}_{\text{(noisy) negative reconstruction error}} + \underbrace{\log p_{\boldsymbol{\theta}}(\mathbf{z}^{(l)}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}^{(l)}|\mathbf{x})}_{\text{regularization terms}}$$

$$\text{where} \quad \mathbf{z}^{(l)} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$$

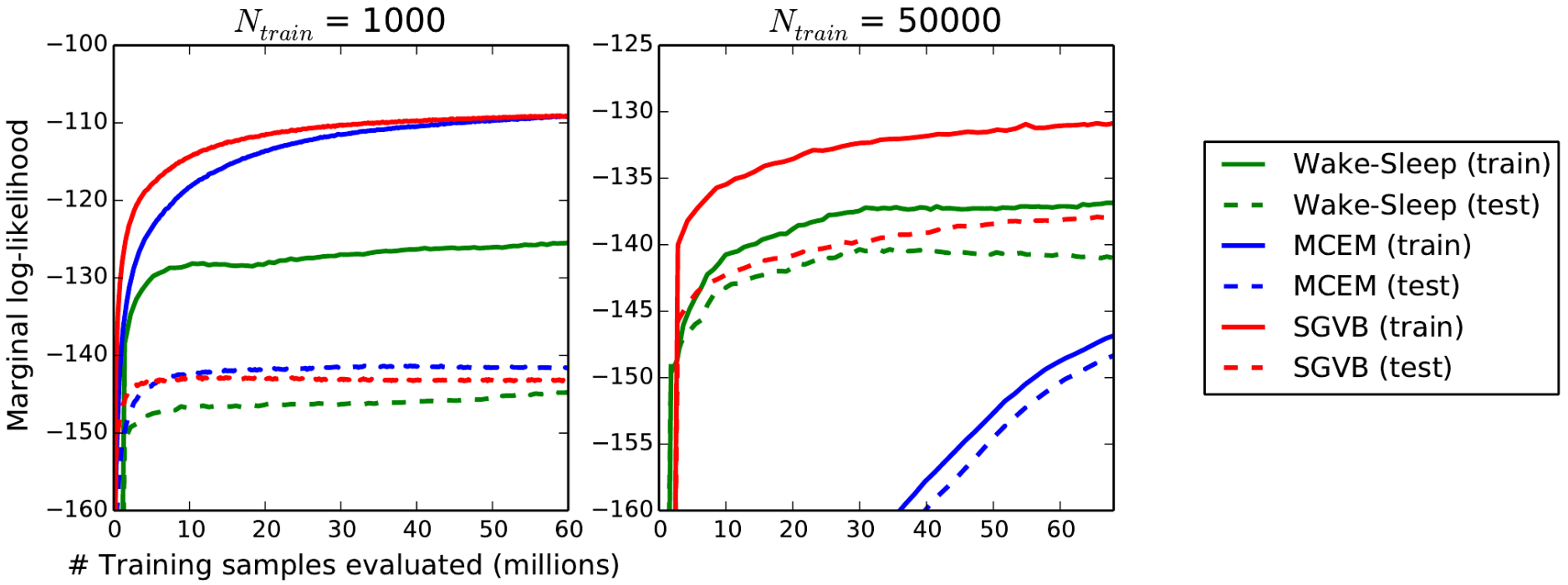
Variational auto-encoder



Results:
Marginal likelihood lower bound

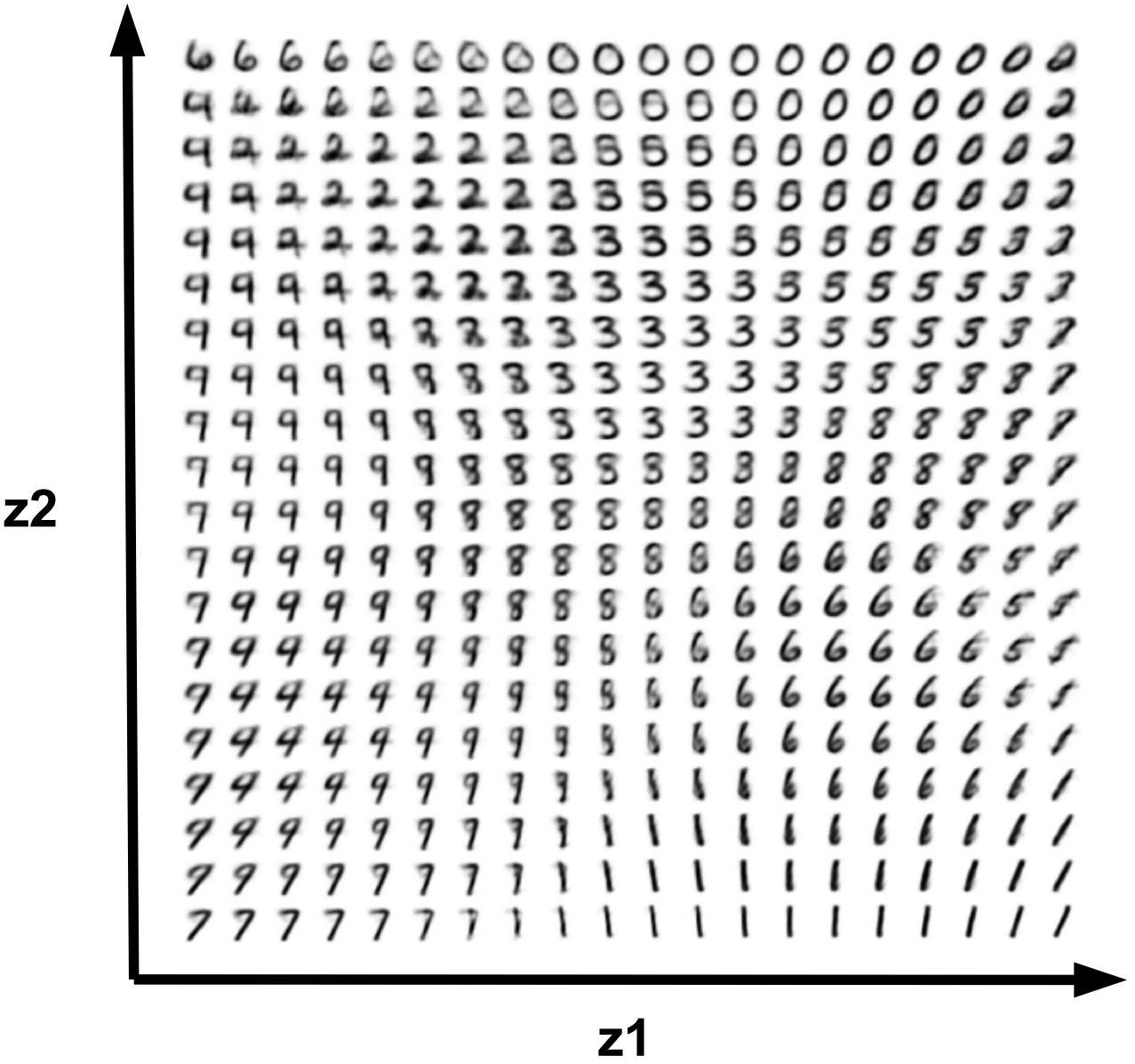


Results: Marginal log-likelihood

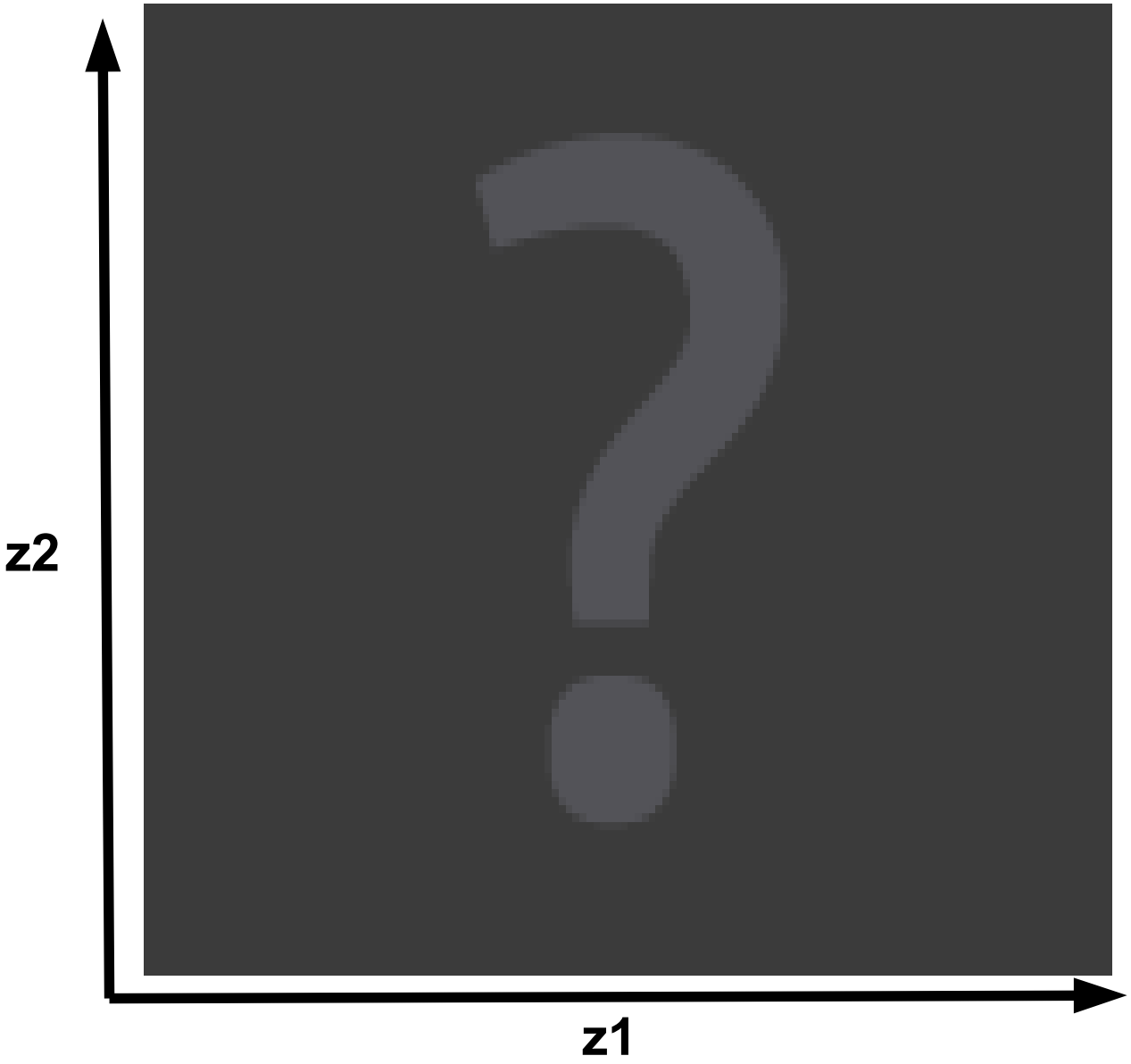


Monte Carlo EM does not scale well to large datasets

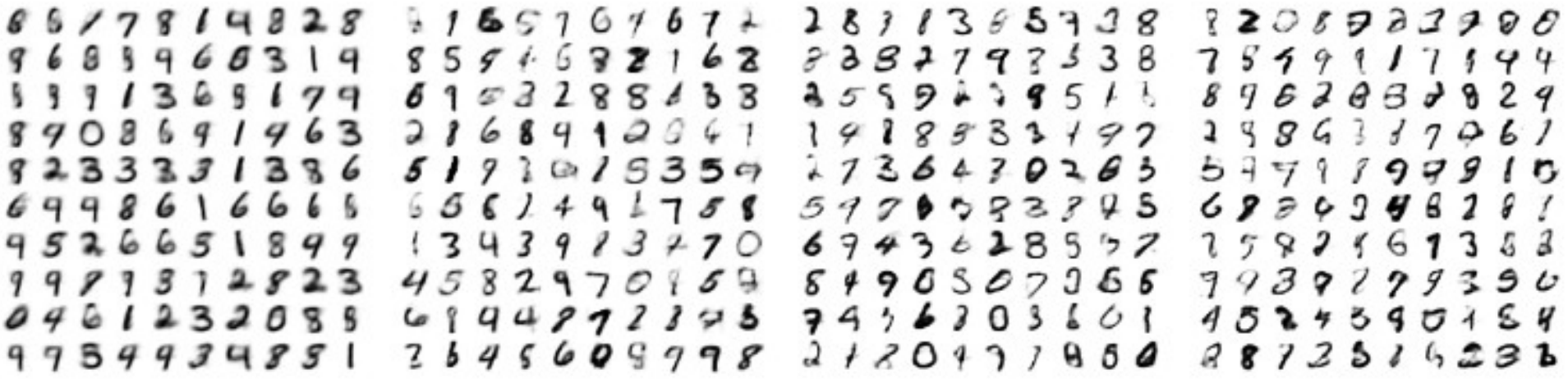
2D Latent space: MNIST



3D latent space: MNIST

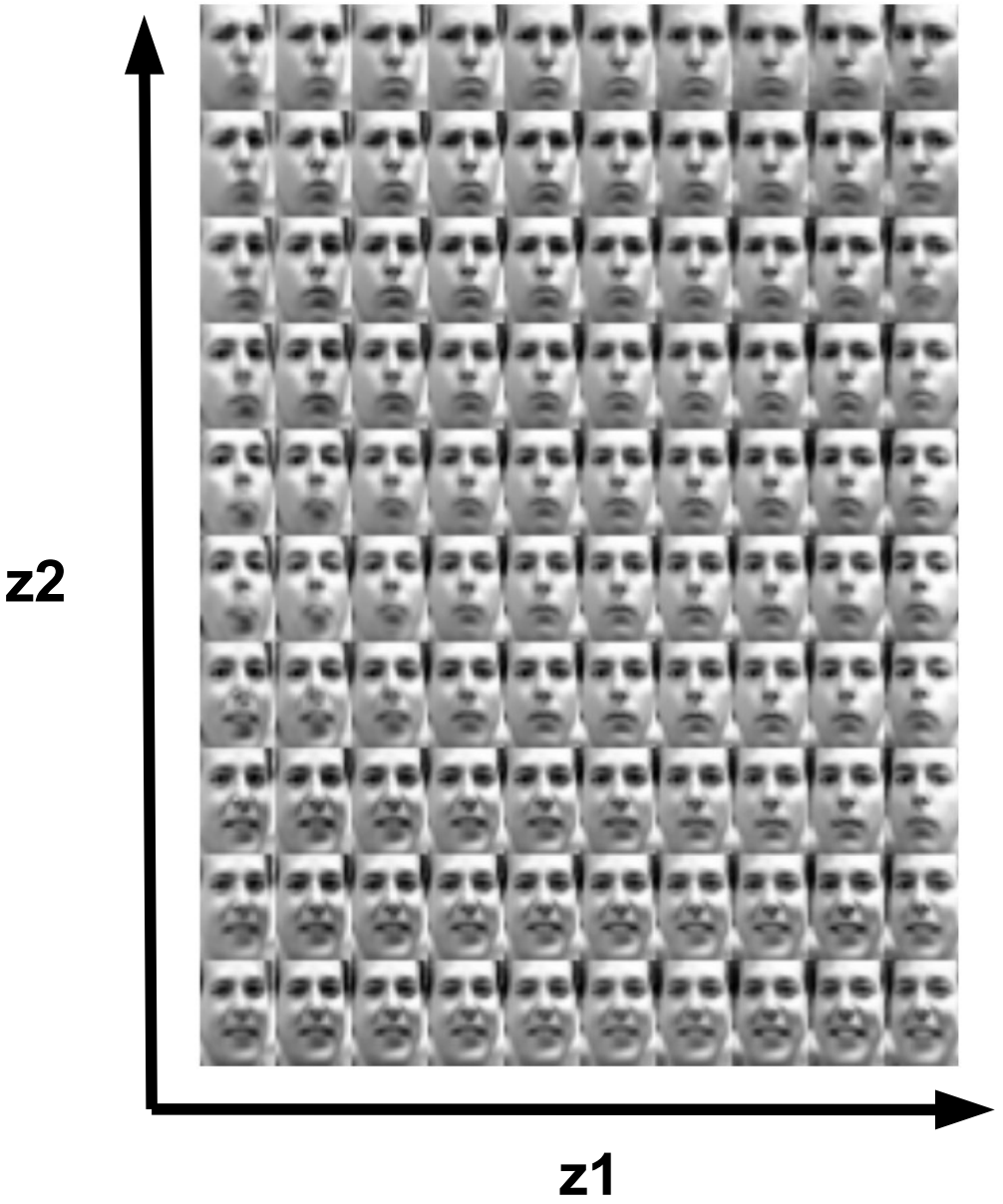


Samples from MNIST (simple ancestral sampling)



(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

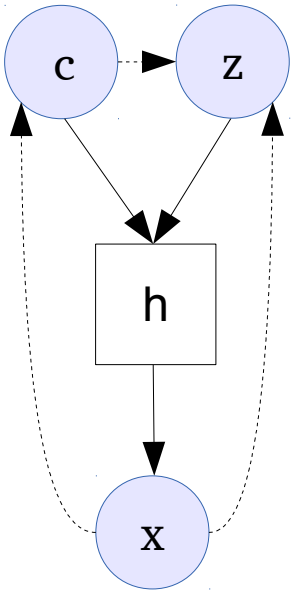
2D Latent space: Frey Face



Labeled Faces in the Wild
(random samples from generative model)

Semi-Supervised Learning and Analogy making

"Semi-Supervised Learning with Deep Generative Models
http://arxiv.org/abs/1406.5298



$p(x,y,z) = p(y)\, p(z)\, p(x|y,z)$

$q(y,z|x) = q(y|x)\, q(z|y,x)$

Table 1: Semi-supervised classification on MNIST with 100, 600 and 1000 labeled samples.

$N_{labeled}$	NN	SVM	CNN	TSVM	EmbNN	CAE	MTC	M1	M2
100	25.81	23.44	22.98	16.81	16.86	13.47	12.03	**11.82** (± 0.25)	11.97 (± 1.71)
600	11.44	8.85	7.68	6.16	5.97	6.3	5.13	5.72 (± 0.049)	**4.94** (± 0.13)
1000	10.7	7.77	6.45	5.38	5.73	4.77	3.64	4.24 (± 0.07)	**3.60** (± 0.56)

Analogy Making

1. Human writes digit
2. Model infers digit class and style
3. Model generates remaining classes with same style

Potential applications

- Representation learning
- Deep generative models of images, video, audio
- Optimal compression (bits-back coding)
- Broader applications of SGVB estimator:

    e.g. learning posterior of the global parameters

- Also see very recent paper:
  "Stochastic Back-propagation and Variational Inference in Deep Latent Gaussian Models"
  [*Danilo J. Rezende, Shakir Mohamed, Daan wierstra*, 2014]

Conclusion

- **Auto-Encoding Variational Bayes**
  - Applies to almost any directed model with continuous latent variables
  - Optimizes a lower bound of the marginal likelihood
  - Scales to very large datasets
  - Simple
  - Fast

Thanks!

https://github.com/y0ast/Variational-Autoencoder.git