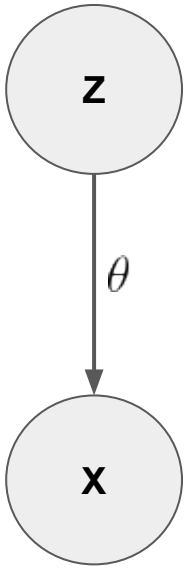
Auto-Encoding Variational Bayes

Diederik P. Kingma, Max Welling

Presenter: Kurtis David

Problem Background

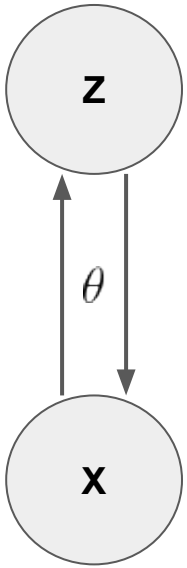


Latent Random Variable

Data

$$P_\theta(X|Z)$$

Bayes Theorem



$$P_\theta(Z|X) = \frac{P_\theta(X|Z)P_\theta(Z)}{P_\theta(X)}$$

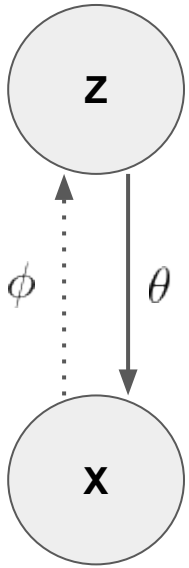
Intractability

$$P_\theta(X) = \int P_\theta(z) P_\theta(X|z) dz$$

Solution: **MCMC**

- Slow to converge
- Have to do this for each datapoint

Posterior Approximation



$$P_\theta(Z|X) \approx q_\phi(Z|X)$$

Maximizing Log Likelihood

$$\log\left(P_\theta(x^{(1)}, x^{(2)}, ..., x^{(n)})\right) = \sum_{i=1}^{n} \log\left(P_\theta(x^{(i)})\right)$$

$$\log\left(P_\theta(x^{(i)})\right) = D_{KL}(q_\phi(Z|x^{(i)})||P_\theta(Z|x^{(i)})) + \mathcal{L}(\theta, \phi|x^{(i)})$$

Variational Lower Bound

$$\mathcal{L}(\theta, \phi | x^{(i)}) = \mathbb{E}_{q_\phi(z|x^{(i)})} \left[\log(P_\theta(x^{(i)}, z)) - \log(q_\phi(z|x^{(i)})) \right]$$

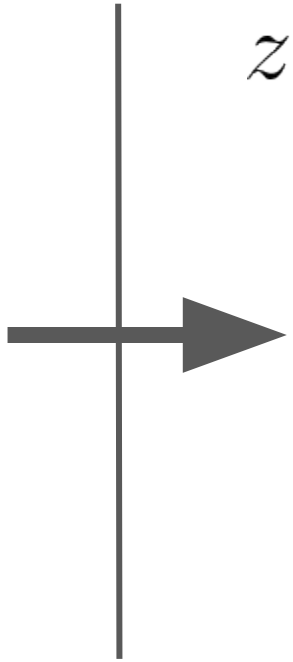
$f_{\theta,\phi}(z)$

$$\approx \frac{1}{L} \sum_{l=1}^{L} f_{\theta,\phi}(z^{(l)}) \qquad z^{(l)} \sim q_\phi(z|x^{(i)})$$

Reparameterization Trick

$$z^{(l)} \sim q_\phi(z|x^{(i)})$$

$$z^{(l)} \sim N(\mu, \sigma^2)$$



$$z^{(l)} = g_\phi(x^{(i)}, \epsilon^{(l)})$$

$$\epsilon^{(l)} \sim P(\epsilon)$$

$$z^{(l)} = \mu + \sigma\epsilon^{(l)}$$

$$\epsilon^{(l)} \sim N(0, 1)$$

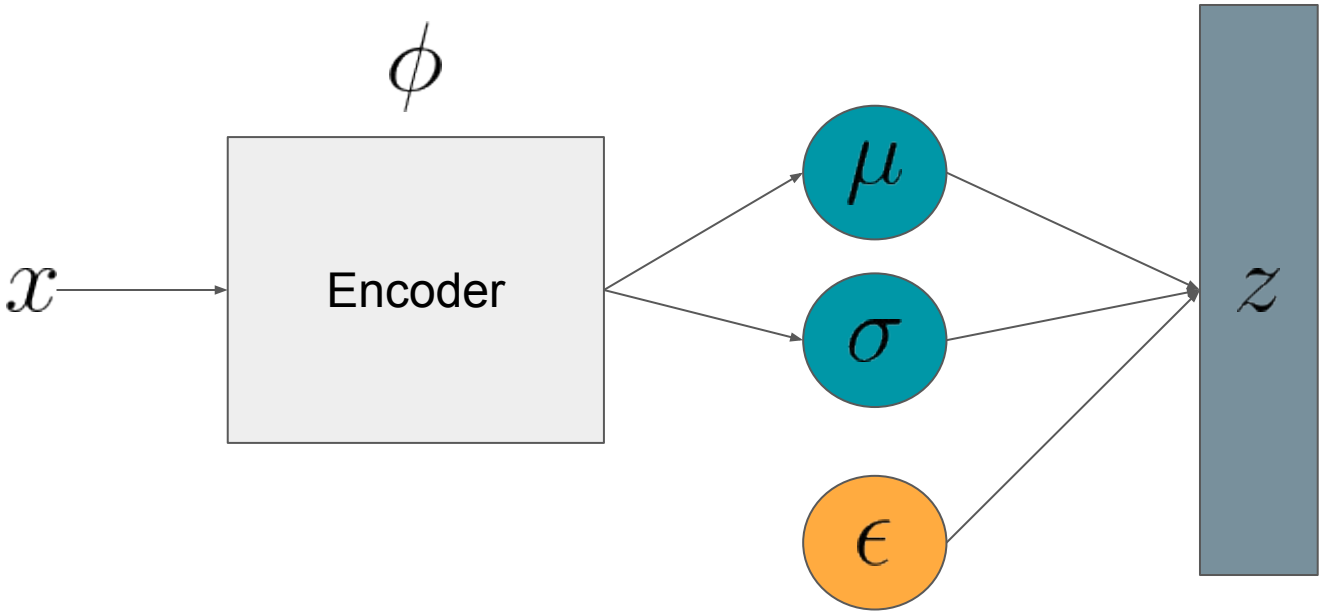
SGVB Estimator

$$\widetilde{\mathcal{L}}(\theta, \phi | x^{(i)}) = -D_{KL}\left(q_\phi(z|x^{(i)})||p_\theta(z)\right) + \frac{1}{L}\sum_{l=1}^{L}\log\left(P(x^{(i)}|z^{(l)})\right)$$

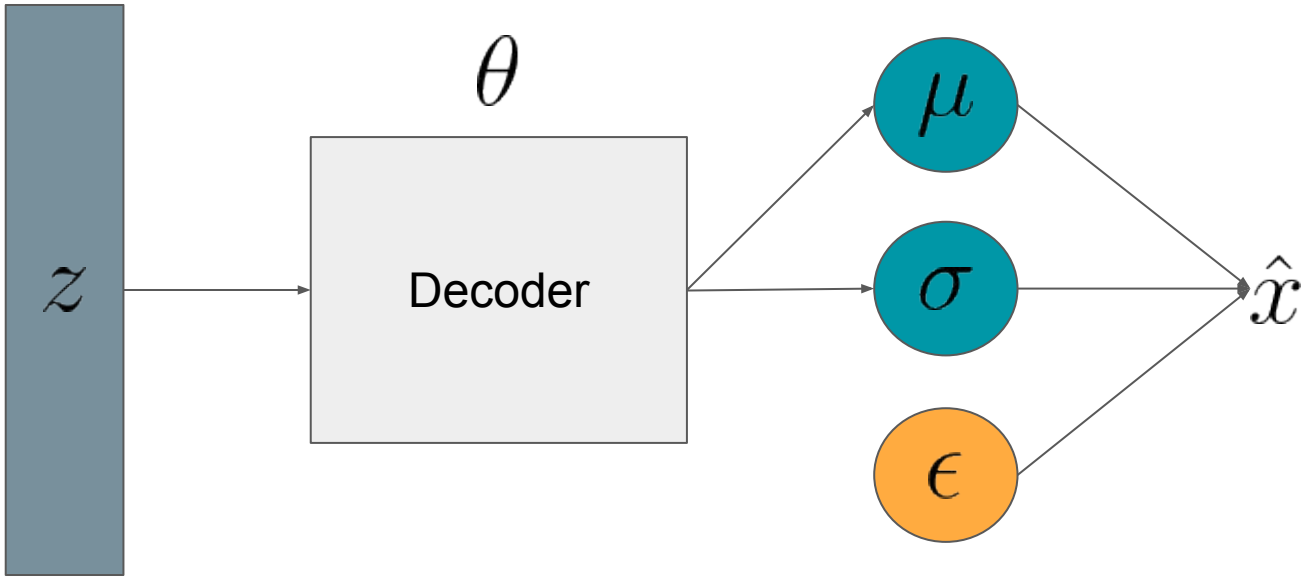
Regularization

Negative Reconstruction Error

Probabilistic Encoder

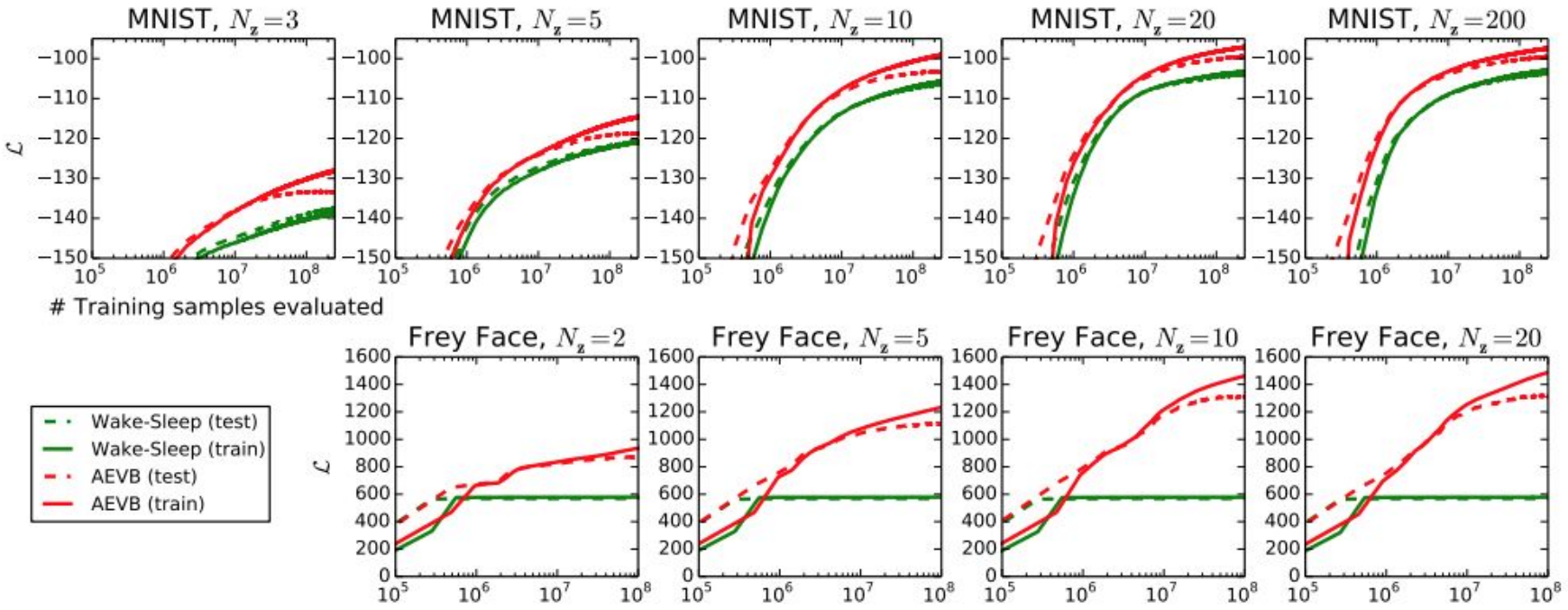


Probabilistic Decoder

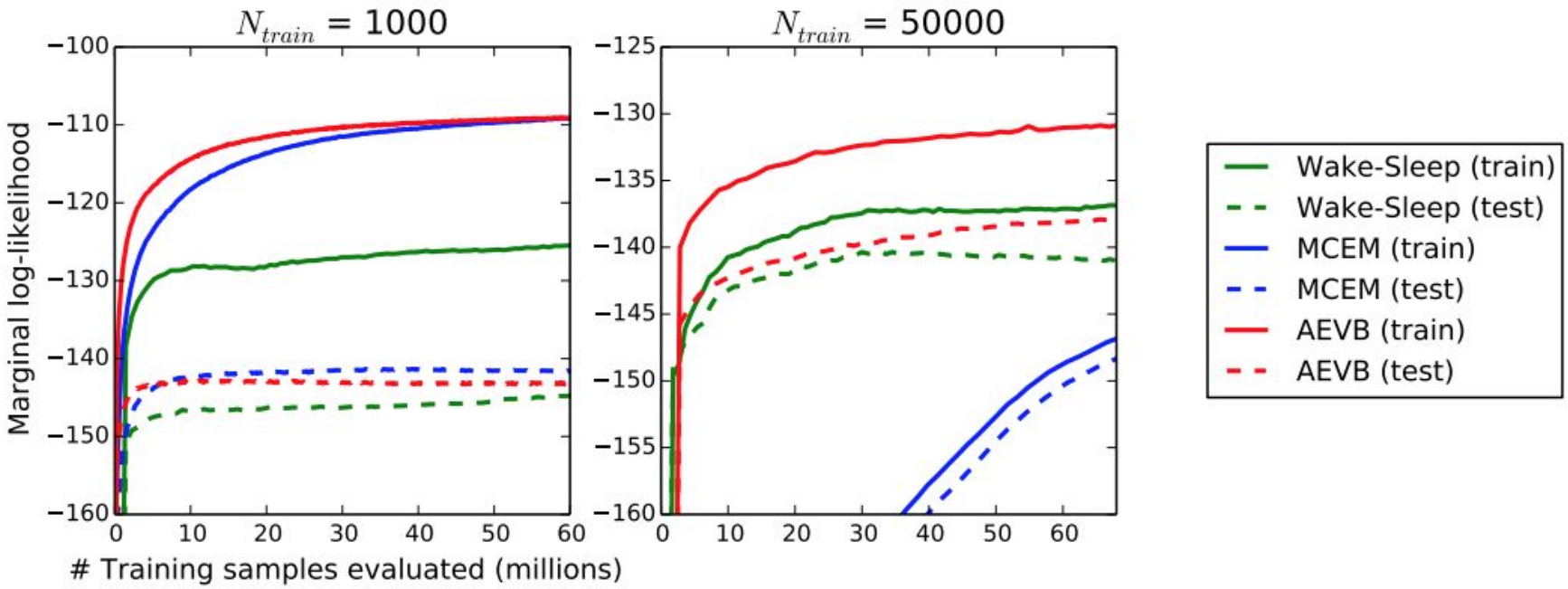


Pros

#1: Finds higher log likelihood



#2: More scalable to large datasets



#3: Straightforward algorithm to implement

Algorithm 1 Minibatch version of the Auto-Encoding VB (AEVB) algorithm. Either of the two SGVB estimators in section 2.3 can be used. We use settings $M = 100$ and $L = 1$ in experiments.

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Initialize parameters
repeat
  $\mathbf{X}^M \leftarrow$ Random minibatch of M datapoints (drawn from full dataset)
  $\boldsymbol{\epsilon} \leftarrow$ Random samples from noise distribution $p(\boldsymbol{\epsilon})$
  $\mathbf{g} \leftarrow \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \widetilde{\mathcal{L}}^M(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{X}^M, \boldsymbol{\epsilon})$ (Gradients of minibatch estimator (8))
  $\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Update parameters using gradients $\mathbf{g}$ (e.g. SGD or Adagrad [DHS10])
until convergence of parameters $(\boldsymbol{\theta}, \boldsymbol{\phi})$
return $\boldsymbol{\theta}, \boldsymbol{\phi}$

#4: Reparameterization works for many distributions

- Exponential
- Student’s t
- Uniform
- Gamma
- Chi-Squared