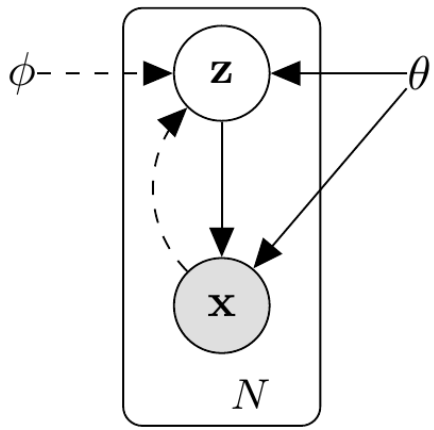
Auto-Encoding Variational Bayes

Diederik Kingma and Max Welling
ICLR 2014

Outline

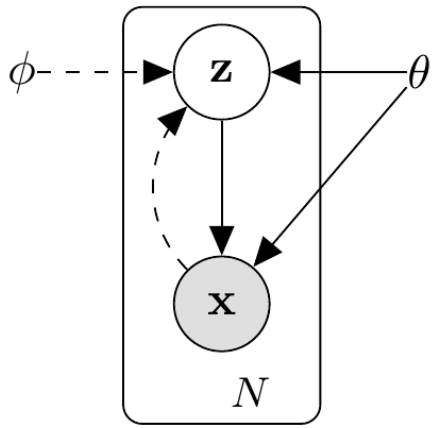
- Introduction to VAE
  - Posterior Approximation Problem
  - Variational Approximation and Evidence Lower Bound (ELBO)
  - Connection between VAE to the Auto-Encoder family
- How to train VAE, optimization of the ELBO
  - Reparameterization trick, how does it help to solve optimization problem?
  - Stochastic Gradient Variational Bayes (SGVB) two versions
  - Auto-Encoding VB (AEVB) algorithm
- Experiment results
- Summary

Posterior Approximation Problem



- Generative process
  - Observable variable (data) x is generated by some random process involving latent variable z
    - ★ step 1: $z \sim p_\theta(z)$
    - ★ step 2: $x \sim p_\theta(x|z)$
  - Latent variables z and generative parameters (or model) θ are unknown
  - Notes: The generative parameters of prior $p_\theta(z)$, the likelihood $p_\theta(x|z)$, and posterior $p_\theta(z|x)$ are different from one another, but here we group them into generative parameters θ

Posterior Approximation Problem

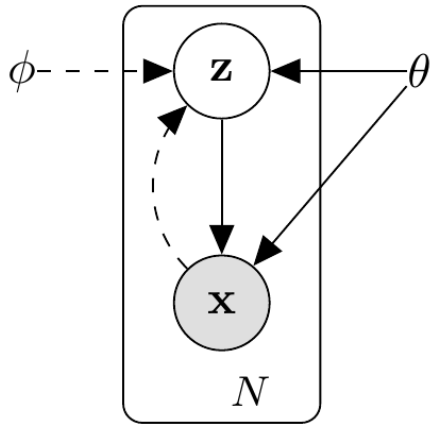


- Posterior approximation: we want to estimate posterior $p_\theta(z|x)$

$$p_\theta(z|x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)}$$

- Problems:
  - $p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz$ can be intractable
  - $p_\theta(z|x)$ can be intractable
  - These intractabilities typically happen when complex likelihood functions $p_\theta(x|z)$ neural networks are used

Variational Approximation



- Approach: find a variational approximation $q_\phi(z|x)$ close to intractable true posterior $p_\theta(z|x)$
  - $q_\phi(z|x)$ is conditional, instead of unconditional $q_\phi(z)$, like in previous discussed papers
    - ★ optimization techniques in this paper can be used for unconditional $q_\phi(z)$ as well
  - $q_\phi(z|x)$ belongs to a family of tractable distributions
  - $q_\phi(z|x)$ is not required to be factorial, like in mean-field VI
  - Notes: we will call ϕ variational parameters in contrast to generative parameters θ

Variational Approximation

- Formally, we set a family of distributions $q_\phi(z|x)$, and solve the optimization problem

$$\underset{\phi,\theta}{\operatorname{argmin}} \operatorname{KL}(q_\phi(z|x) \parallel p_\theta(z|x)) = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{q_\phi(z|x)}{p_\theta(z|x)}\right]$$

  - Notes: generative parameters θ are assumed to be unknown so we want to solve for them as well
- Problems
  - Since $p_\theta(z|x)$ is intractable, $\operatorname{KL}(q_\phi(z|x) \parallel p_\theta(z|x))$ is also intractable

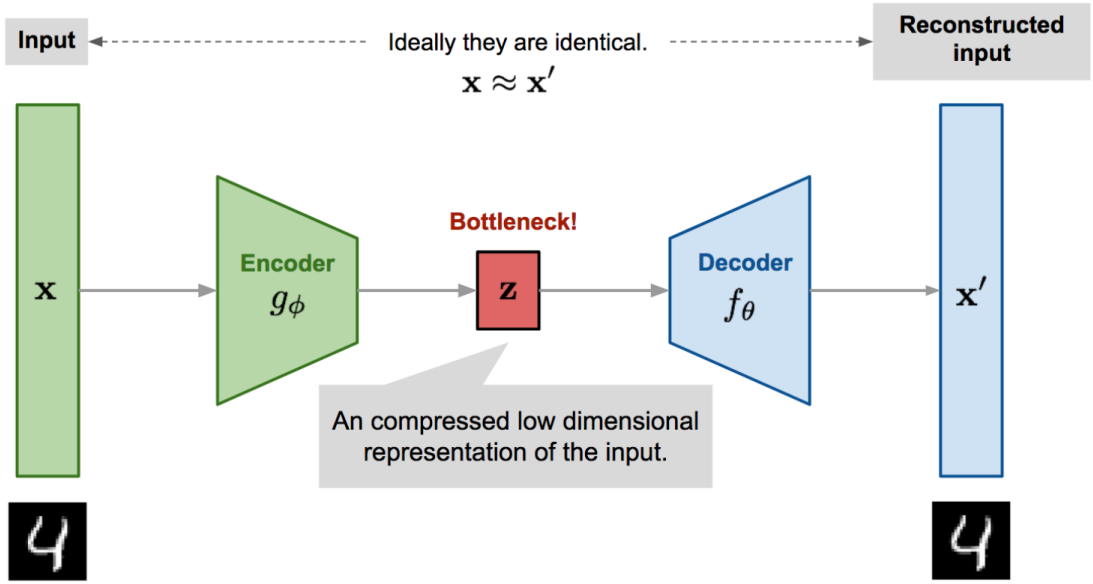
Evidence (or Variational) Lower Bound (ELBO)

- ELBO

$$
\begin{aligned}
&\mathrm{KL}(q_\phi(z|x) \,\|\, p_\theta(z|x)) \\
&\quad = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{q_\phi(z|x)}{p_\theta(z|x)}\right] \\
&\quad = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{q_\phi(z|x)p_\theta(x)}{p_\theta(z,x)}\right] \\
&\quad = \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{q_\phi(z|x)}{p_\theta(z,x)}\right] + \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x)\right] \\
&\quad = -\underbrace{\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(z,x) - \log q_\phi(z|x)\right]}_{\mathrm{ELBO}=\mathcal{L}(\phi,\theta;x)} + \underbrace{\log p_\theta(x)}_{\text{log likelihood}}
\end{aligned}
$$

- maximizing $\mathcal{L}(\phi, \theta; x)$ is equivalent to minimizing $\mathrm{KL}(q_\phi(z|x) \,\|\, p_\theta(z|x))$
- simultaneously, maximizing the lower bound of $\log p_\theta(x) \geq \mathcal{L}(\phi, \theta; x)$

Connection between $\mathcal{L}(\phi, \theta; x)$ and Auto-encoder



- Recall basic Auto-encoder model
  - It consists of two parts g_ϕ and f_θ
    - ★ g_ϕ learns to encode and compress data x into latent z and
    - ★ f_θ learns to decode and reconstruct latent z back x'
  - Learning objective: minimizing the reconstruction loss/error

$$\mathcal{L}_{\text{AE}}(\phi, \theta; x) = [x' - f_\theta(g_\phi(x))]^2$$

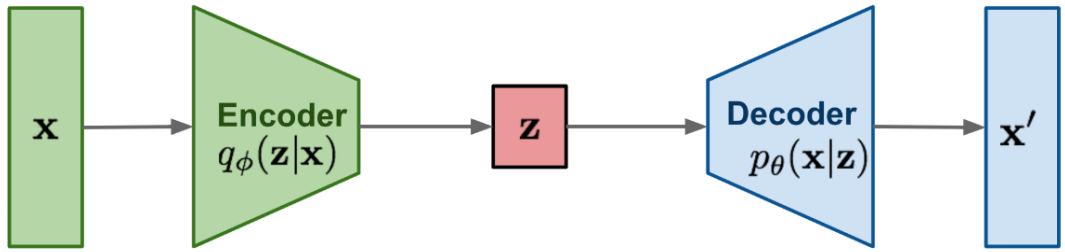
Connection between $\mathcal{L}(\phi, \theta; x)$ and Auto-encoder

- Rewritting $\mathcal{L}(\phi, \theta; x)$

$$
\begin{aligned}
\mathcal{L}(\phi, \theta; x) &= \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(z, x) - \log q_\phi(z|x) \right] \\
&= \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x|z) + log p_\theta(z) - \log q_\phi(z|x) \right] \\
&= \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x|z) \right] - \mathbb{E}_{q_\phi(z|x)} \left[\log \frac{q_\phi(z|x)}{p_\theta(z)} \right] \\
&= \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x|z) \right] - \mathbb{E}_{q_\phi(z|x)} \left[\log \frac{q_\phi(z|x)}{p_\theta(z)} \right] \\
&= \underbrace{\mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x|z) \right]}_{\text{expected reconstruction log likelihood}} - \underbrace{\mathrm{KL}(q_\phi(z|z), p_\theta(z))}_{\text{regularizer}}
\end{aligned}
$$

  - first term: learning objective to maximize the expected reconstruction log likelihood
  - second term: regularizer that makes $q_\phi(z|x)$ close to prior $p_\theta(z)$

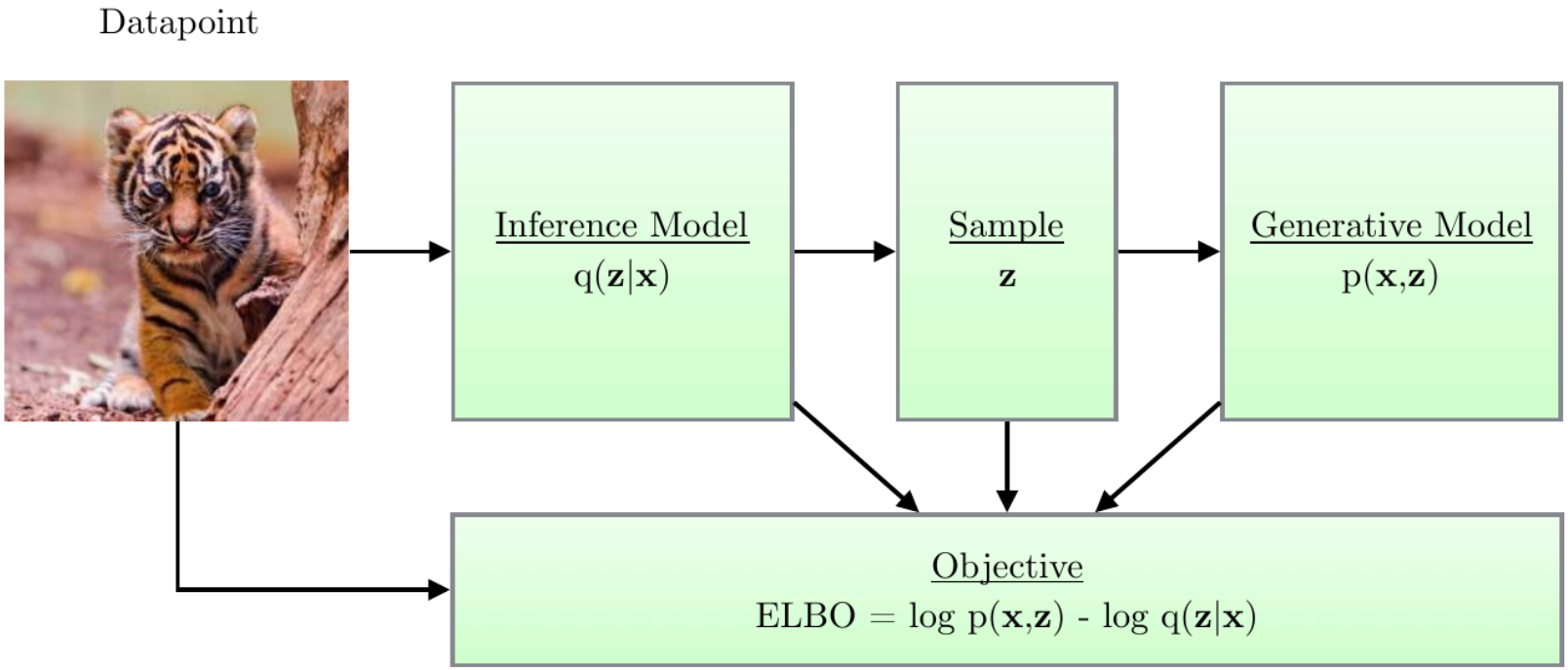
Connection between $\mathcal{L}(\phi, \theta; x)$ and Auto-encoder



- Variational Auto-encoder model, the intuition
  - It also consists of two parts: variational encoder $q_\phi(z|x)$ and generative decoder $p_\theta(x|z)$
    - ★ $q_\phi(z|x)$ learns to encode or compress data x into latent z
    - ★ $p_\theta(x|z)$ learns to decode or reconstruct latent z back to x'
    - ★ Notes: $z \sim q_\phi(z|x)$ is actually a random variable, in contrast to determistic z in basic Auto-encoder model
  - Learning objective: maximizing the expected reconstruction log likelihood with a regularizer that makes encoder $q_\phi(z|x)$ close to prior $p_\theta(z)$

$$\mathcal{L}_{\text{VAE}}(\phi, \theta; x) = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - \text{KL}(q_\phi(z|z), p_\theta(z))$$

Overview of Training Scheme in VAE



source: Kingma et al., 2019

Optimization of the $\mathcal{L}(\phi, \theta; x)$

- Optimization problem

$$\underset{\phi,\theta}{\operatorname{argmin}} \mathcal{L}(\phi, \theta; x) = \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(z, x) - \log q_\phi(z|x)\right]$$

- General approach
  - Compute the gradient $\mathcal{L}(\phi, \theta; x)$ w.r.t. variational parameters ϕ and generative parameters θ
  - Use backpropagation and stochastic gradient descent algorithm for training large networks and large dataset

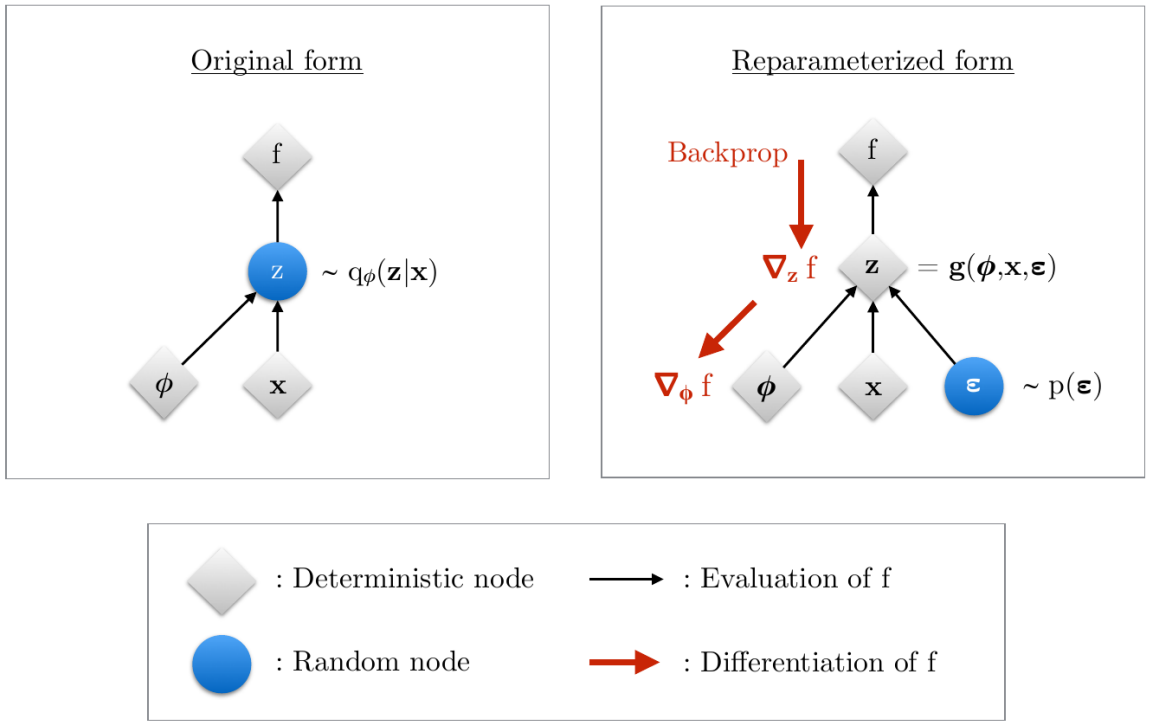
Optimization of the $\mathcal{L}(\phi, \theta; x)$

- The naive Monte Carlo gradient estimator

$$\nabla_\phi \mathbb{E}_{q_\phi(z|x)}[f(z)] = \mathbb{E}_{q_\phi(z|x)}[f(z) \nabla_{q_\phi(z|x)} \log q_\phi(z|x)]$$
$$\simeq \frac{1}{L} \sum_{l=1}^{L} f(z^l) \nabla_{q_\phi(z^l|x)} \log q_\phi(z^l|x)$$
$$\text{where } z^l \sim q_\phi(z|x)$$

  - However, this gradient estimator exhibits very high variance (Blei et al., 2012)
  - We donot use backpropagation in this case.

The reparameterization trick

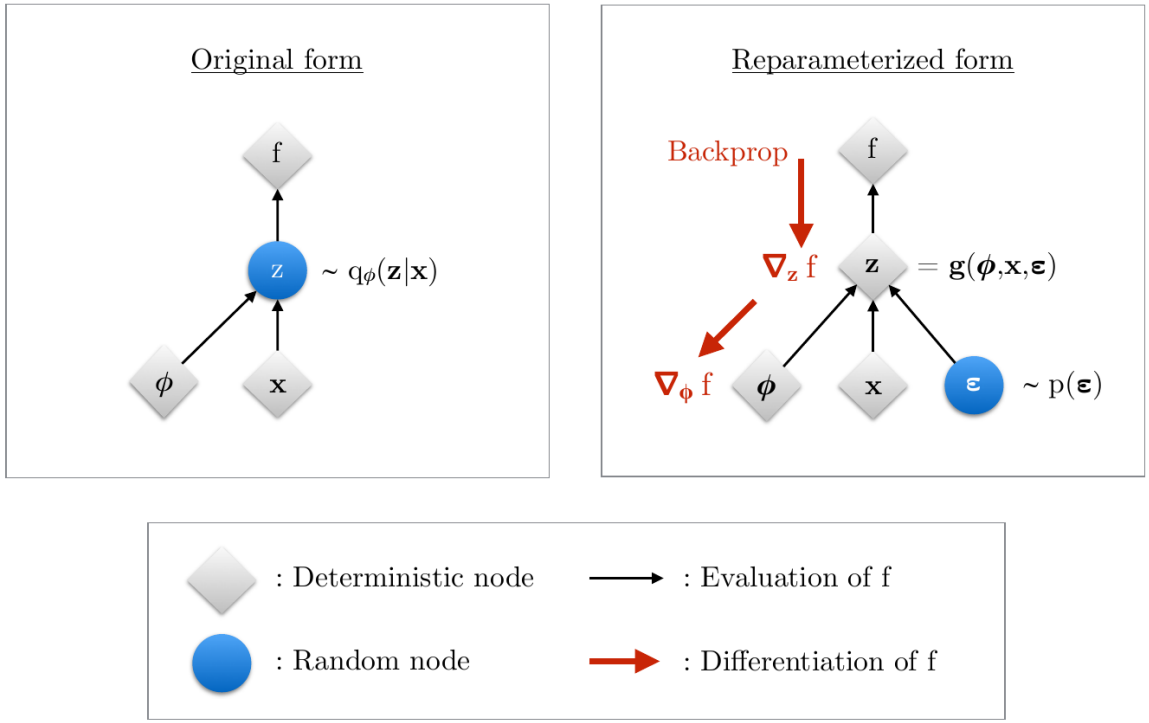


- Sampling $z \sim q_\phi(z|x)$ is a stochastic process, we cannot backpropagate gradient through z
- Reparameterize the r.v. z using a differentiable transformation $g_\phi(\epsilon, x)$ of an (auxiliary) noise variable ϵ

$$z = g_\phi(\epsilon, x) \quad \text{where} \quad \epsilon \sim p(\epsilon)$$

source: Kingma et al., 2019

The reparameterization trick



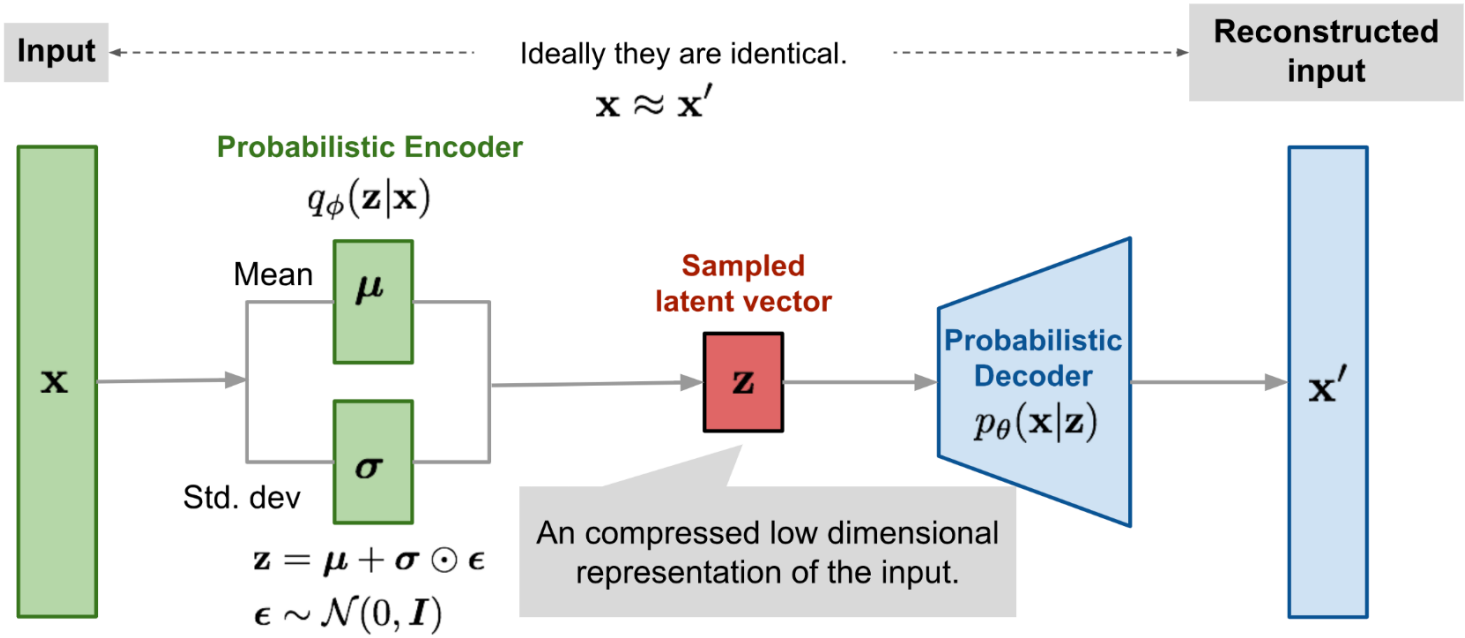
- An example, univariate Gaussian case
  - Let $z \sim p(z|x) = N(\mu, \sigma^2)$
  - Reparameterize $z = \mu + \sigma\epsilon$, where $\epsilon \sim N(0, 1)$

source: Kingma et al., 2019

The reparameterization trick

- Three basic approaches: how to choose such transformation $g_\phi(.)$ and auxiliary distribution $\epsilon \sim p(\epsilon)$
  1. Tractable inverse CDF?
     - ★ Let $\epsilon \sim U(0,1)$ and $g_\phi(\epsilon, x)$ be the inverse CDF of $q_\phi(z|x)$
     - ★ Examples: Exponential, Cauchy, Logistic, Gumbel, etc.
  2. "location-scale" family of distributions
     - ★ Let $\epsilon \sim N(0,1)$ and $g_\phi(.) = \text{location} + \text{scale} \cdot \epsilon$
     - ★ Examples: Gaussian, Laplace, Elliptical, etc.
  3. Composition
     - ★ It is often possible to express random variables as different transformation of auxiliary variables
     - ★ Examples: Log-Normal (exponentiation of normally distributed variable), Gamma (a sum over exponentially distributed variables), etc.

Recap: VAE with reparameterization trick, univariate Gaussian case



source: Lilianweng's blog post

The reparameterization trick

- The reparameterization is useful: the Monte Carlo estimate of the expectation $\mathbb{E}_{q_\phi(z|x)}[f(z)]$ is now differentiable w.r.t. ϕ.
  - We donot need the naive Monte Carlo gradient estimator

$$\nabla_\phi \mathbb{E}_{q_\phi(z|x)}[f(z)] \simeq \frac{1}{L}\sum_{l=1}^{L} f(z^l)\nabla_{q_\phi(z^l|x)} \log q_\phi(z^l|x)$$
$$\text{where } z^l \sim q_\phi(z|x)$$

  - Instead, more efficiently, we sample ϵ to estimate the expectation $\mathbb{E}_{q_\phi(z|x)}[f(z)]$ and apply backpropagation to learn parameters.

$$\mathbb{E}_{q_\phi(z|x)}[f(z)] \simeq \frac{1}{L}\sum_{l=1}^{L} f(g_\phi(x, \epsilon^l)) \quad \text{where} \quad \epsilon \sim p(\epsilon)$$

  - In practice, we may only need to sample once $L = 1$

The reparameterization trick

- Proof: we can estimate the expectation $\mathbb{E}_{q_\phi(z|x)}[f(z)]$ with Monte Carlo samples
  - Given the determistic mapping $z = g_\phi(\epsilon, x)$ we know that

$$q_\phi(z|x) \prod_i dz_i = p(\epsilon) \prod_i d\epsilon_i$$

  - Therefore,

$$\begin{aligned}
\mathbb{E}_{q_\phi(z|x)}[f(z)] &= \int q_\phi(z|x) f(z) dz \\
&= \int p(\epsilon) f(z) d\epsilon \\
&= \int p(\epsilon) f(g_\phi(\epsilon, x)) d\epsilon \\
&= \mathbb{E}_{p(\epsilon)}[f(g_\phi(\epsilon, x))] \\
&\simeq \frac{1}{L} \sum_{l=1}^{L} f(g_\phi(x, \epsilon^l)) \quad \text{where} \quad \epsilon \sim p(\epsilon)
\end{aligned}$$

The Stochastic Gradient Variational Bayes (SGVB) Estimator

- Optimization problem A

$$\underset{\phi,\theta}{\operatorname{argmax}}\, \mathcal{L}(\phi, \theta; x) = \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(z, x) - \log q_\phi(z|x)\right]$$

- Monte Carlo estimate of expectation

$$\mathbb{E}_{q_\phi(z|x)}[f(z)] \simeq \frac{1}{L} \sum_{l=1}^{L} f(g_\phi(x, \epsilon^l)) \quad \text{where} \quad \epsilon \sim p(\epsilon)$$

- SGVB estimator A: $\mathcal{L}(\phi, \theta; x) \simeq \tilde{\mathcal{L}}^A(\phi, \theta; x)$

$$\tilde{\mathcal{L}}^A(\phi, \theta; x) = \frac{1}{L} \sum_{l=1}^{L} \left[\log p_\theta(z^l, x) - \log q_\phi(z^l|x)\right]$$

$$\text{where} \quad z^l = g_\phi(\epsilon^l, x) \quad \text{and} \quad \epsilon^l \sim p(\epsilon)$$

The Stochastic Gradient Variational Bayes (SGVB) Estimator

- Optimization problem B

$$\underset{\phi,\theta}{\operatorname{argmax}}\, \mathcal{L}(\phi, \theta; x) = \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x|z)\right] - \mathrm{KL}(q_\phi(z|z), p_\theta(z))$$

- SGVB estimator B: $\mathcal{L}(\phi, \theta; x) \simeq \tilde{\mathcal{L}}^B(\phi, \theta; x)$

$$\tilde{\mathcal{L}}^B(\phi, \theta; x) = \frac{1}{L} \sum_{l=1}^{L} \left[\log q_\phi(x|z^l)\right] - \mathrm{KL}(q_\phi(z|z), p_\theta(z))$$

$$\text{where} \quad z^l = g_\phi(\epsilon^l, x) \quad \text{and} \quad \epsilon^l \sim p(\epsilon)$$

  - We use this version if $\mathrm{KL}(q_\phi(z|z), p_\theta(z))$ can be computed analytically
  - We only have to estimate $\mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x|z)\right]$ in this case

The SGVB Estimator for full dataset

- SGVB estimator for the full dataset X with N datapoints, based on dataset

$$\mathcal{L}(\phi, \theta; X) \simeq \tilde{\mathcal{L}}^M(\phi, \theta; X^M) = \frac{N}{M} \sum_{i=1}^{M} \mathcal{L}(\phi, \theta, x^i)$$

where $X^M = \{x^i\}_{i=1}^M$ is a randomly drawn minibatch of M datapoints from the full dataset X.

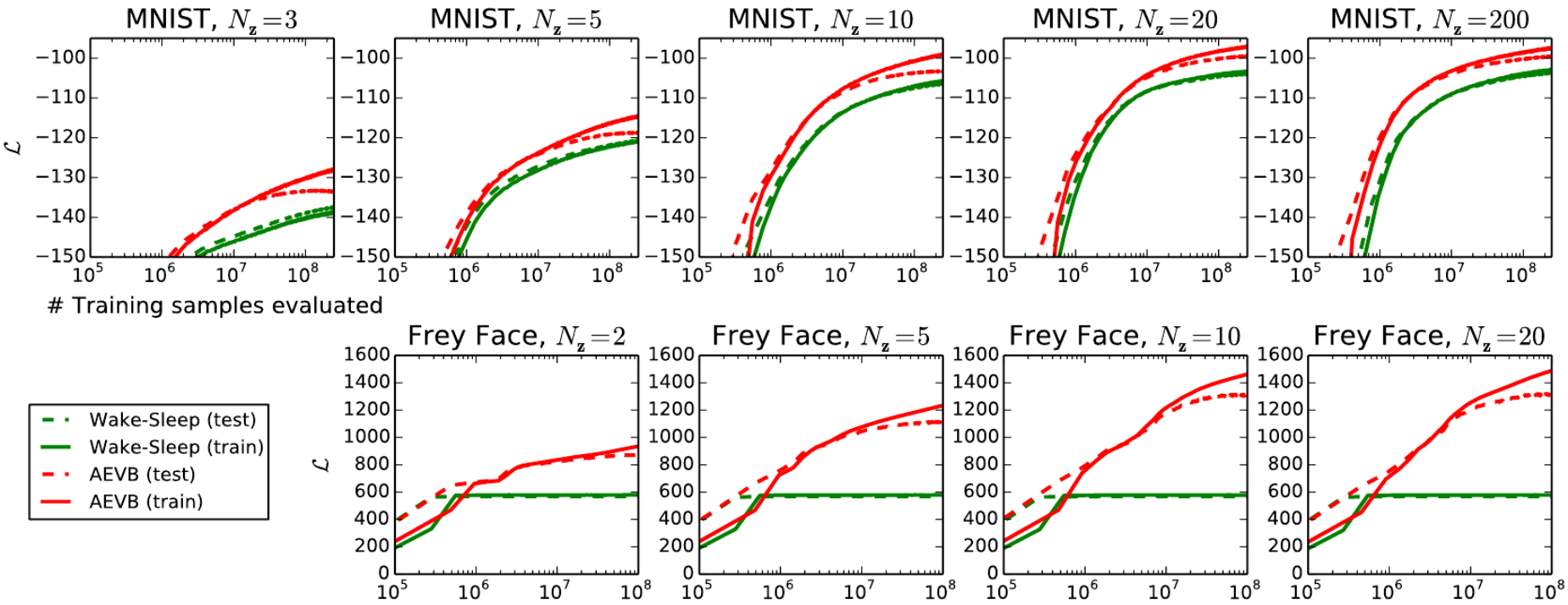
  - In the experiments, number of samples $L = 1$ if M is large enough, e.g., $M = 100$

The Auto-Encoding VB (AEVB) Algorithm

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.

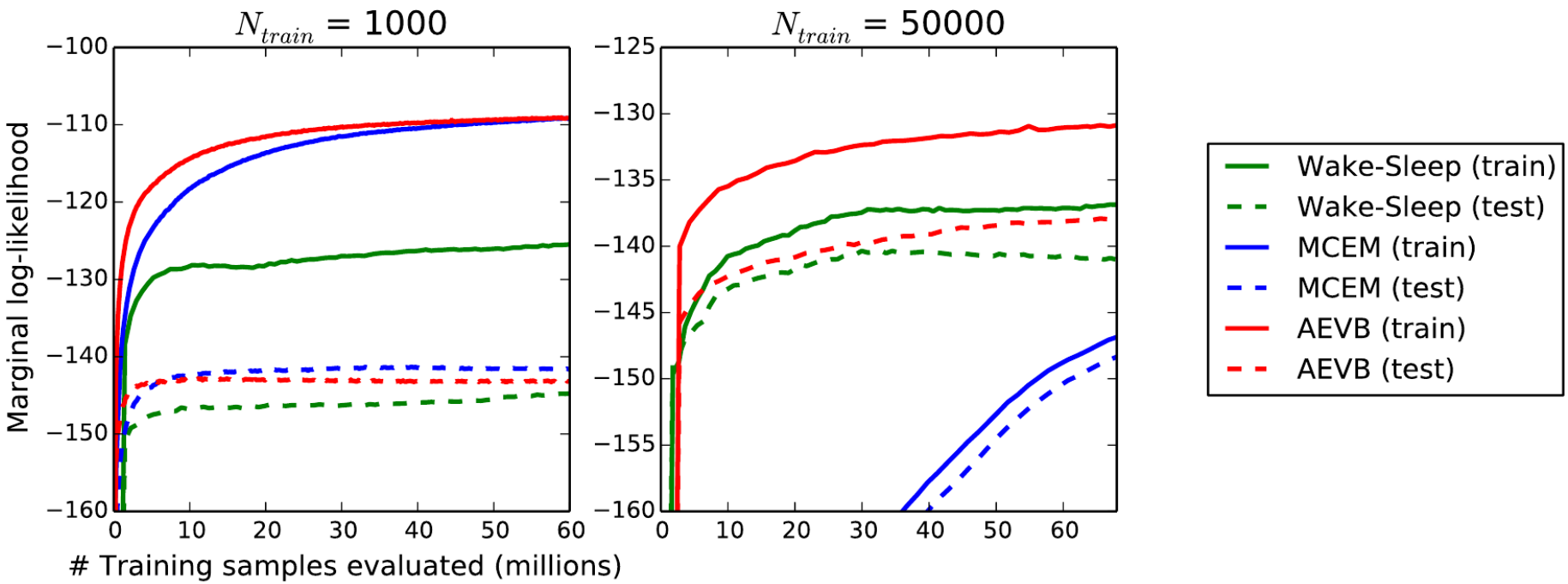
$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Initialize parameters
repeat
  $\mathbf{X}^M \leftarrow$ Random minibatch of M datapoints (drawn from full dataset)
  $\boldsymbol{\epsilon} \leftarrow$ Random samples from noise distribution $p(\boldsymbol{\epsilon})$
  $\mathbf{g} \leftarrow \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \widetilde{\mathcal{L}}^M(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{X}^M, \boldsymbol{\epsilon})$ (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 $\boldsymbol{\theta}, \boldsymbol{\phi}$

Experimental Results



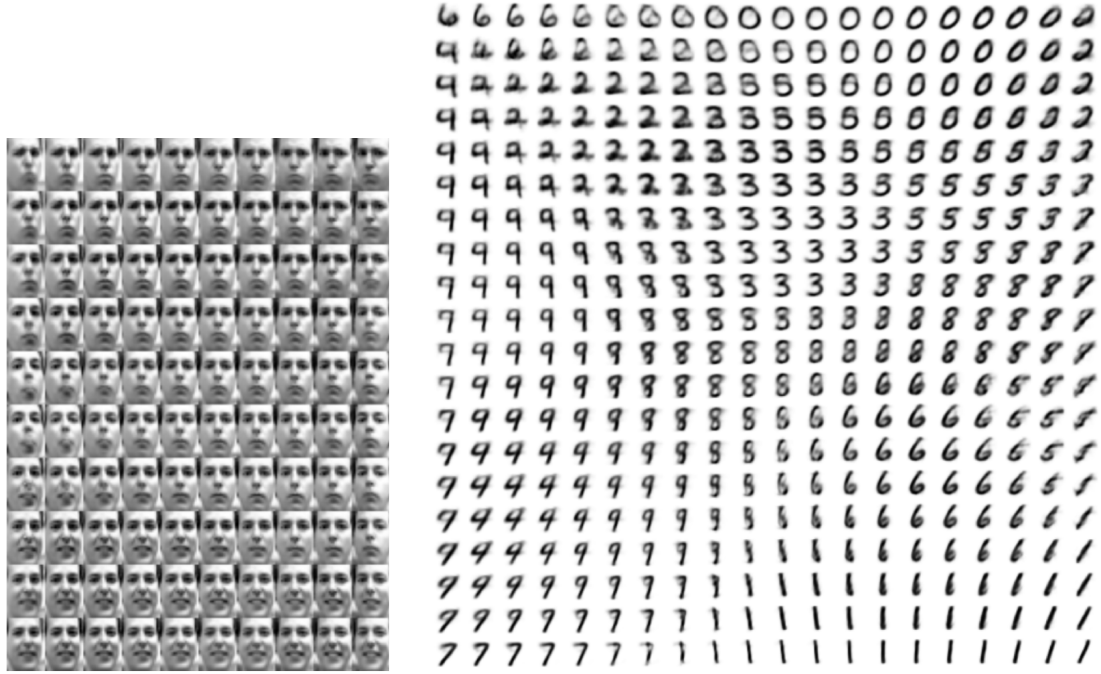
- Using MNIST and Frey Face dataset, comparison between AEVB and Wake-sleep in term of optimizing the lower bound
- AEVB converges faster and reaches better lower bound in all experiments
- More latent variables does not result in more overfitting, which is explained by the regularizing effect of the learning objective

Experimental Results



- Using MNIST, comparison between AEVB, Wake-sleep, and Monte Carlo EM in term of estimated marginal log-likelihood, for very low-dimensional latent space
- AEVB converges faster and reaches better marginal log-likelihood, even better with larger dataset
- Monte Carlo EM takes long time to learn on larger dataset

VAE as Generative Model



(a) Learned Frey Face manifold

(b) Learned MNIST manifold

- Since the prior of the latent space is Gaussian, linearly spaced coordinates on the unit square were transformed through the inverse CDF of the Gaussian to produce values of the latent variables z.
- For each z, we plot the corresponding x using $p_\theta(x|z)$

Summary

- VAE consists of two parts: variational encoder $q_\phi(z|x)$ and generative decoder $p_\theta(x|z)$
  - $q_\phi(z|x)$ learns to encode or compress data x into latent z
  - $p_\theta(x|z)$ learns to decode or reconstruct latent z back to x'
- VAE uses evidence lower bound (ELBO) as a learning objective

$$\mathcal{L}(\phi, \theta; x) = \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(z, x) - \log q_\phi(z|x)\right] \quad \text{(A)}$$
$$= \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x|z)\right] - \mathrm{KL}(q_\phi(z|z), p_\theta(z)) \quad \text{(B)}$$

  - Maximizing ELBO is equivalent to minimizing the $\mathrm{KL}(q_\phi(z|x) \parallel p_\theta(z|x))$, and maximizing lower bound of $\log p_\theta(x)$
  - We use form ELBO form B if we can compute $\mathrm{KL}(q_\phi(z|z), p_\theta(z))$

Summary

- Reparameterize trick
  - We reparameterize the r.v. z using a differentiable transformation $g_\phi(\epsilon, x)$ of an (auxiliary) noise variable ϵ

$$z = g_\phi(\epsilon, x) \quad \text{where} \quad \epsilon \sim p(\epsilon)$$

  - This allows us to construct a differentiable estimator of expectation

$$\mathbb{E}_{q_\phi(z|x)}[f(z)] \simeq \frac{1}{L} \sum_{l=1}^{L} f(g_\phi(x, \epsilon^l)) \quad \text{where} \quad \epsilon \sim p(\epsilon)$$

- SGVB estimator A

$$\tilde{\mathcal{L}}^A(\phi, \theta; x) = \frac{1}{L} \sum_{l=1}^{L} \Big[\log p_\theta(z^l, x) - \log q_\phi(z^l|x) \Big]$$

$$\text{where} \quad z^l = g_\phi(\epsilon^l, x) \quad \text{and} \quad \epsilon^l \sim p(\epsilon)$$

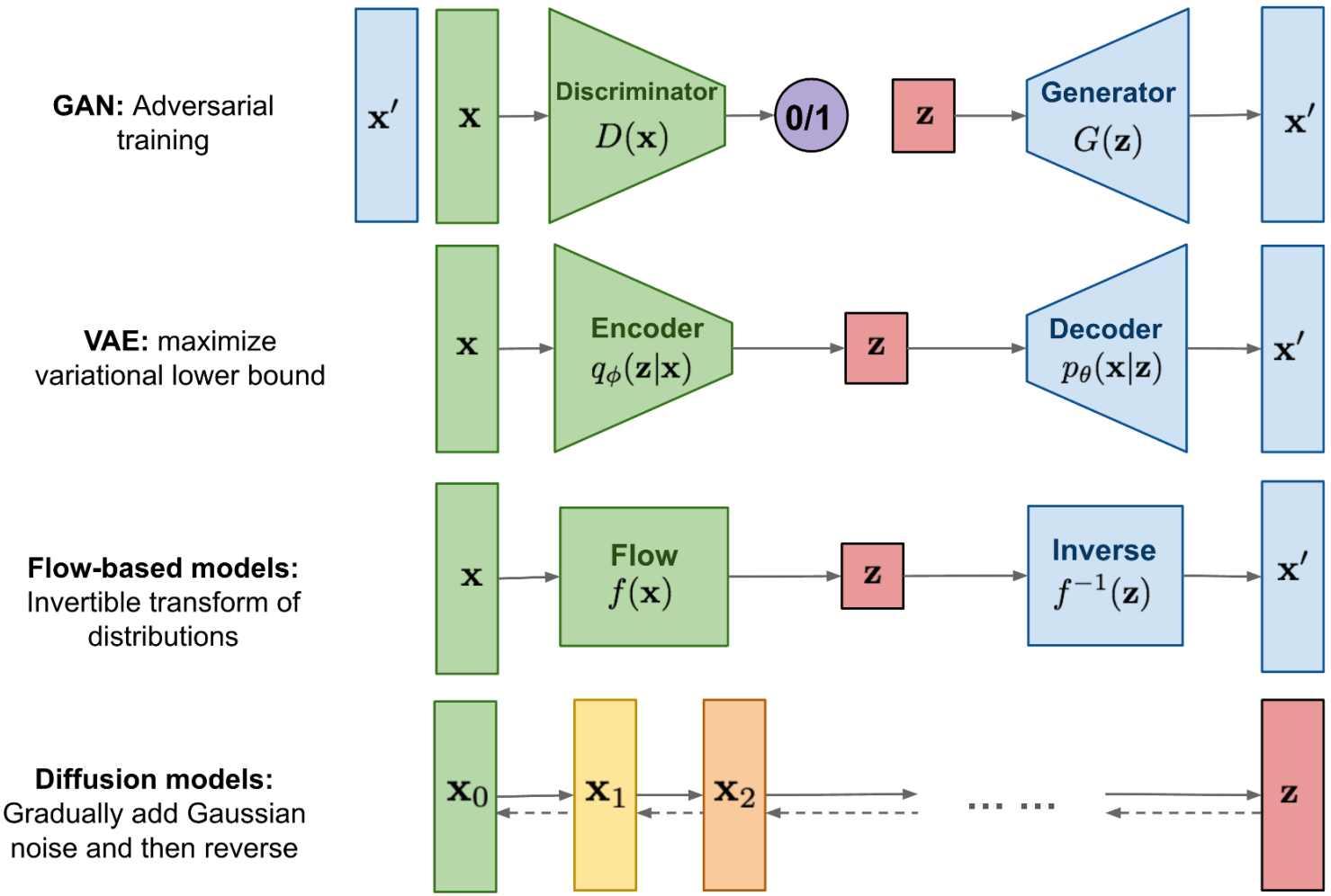
Summary

- AEVB Algorithm

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 $\boldsymbol{\theta}, \boldsymbol{\phi}$

Different Types of Models Similar to VAE



source: Lilianweng's blog post