# DSC291: Machine Learning with Few Labels

## Unsupervised Learning

**Zhiting Hu**

Lecture 17, May 13, 2024

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

# This Lecture

- Variational Inference (30mins)

- Presentation #1 (10mins):
  - Hung Nguyen, SPARF: Neural Radiance Fields from Sparse & Noisy Poses

- Presentation #2 (10mins):
  - Zhihan Chen, Efficient (Soft) Q-Learning for Text Generation with Limited Good Data

# Recap: EM and Variational Inference

- The EM algorithm:
  - E-step: $q^{t+1} = \arg\min_q F\left(q, \theta^t\right)$

    $$= p(\boldsymbol{z}|\boldsymbol{x}, \theta^t) = \frac{p(\boldsymbol{z}, \boldsymbol{x}|\theta^t)}{\sum_z p(\boldsymbol{z}, \boldsymbol{x}|\theta^t)}$$

  - M-step: $\theta^{t+1} = \arg\min_\theta F\left(q^{t+1}, \theta^t\right)$

$$\ell(\theta; \boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}|\theta)}{q(\boldsymbol{z}|\boldsymbol{x})}\right] + \mathrm{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big)$$

$$= -F(q, \theta) + \mathrm{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big)$$

# Recap: EM and Variational Inference

- The EM algorithm:
  - E-step: $q^{t+1} = \arg\min_q F\left(q, \theta^t\right)$

    $= p(\boldsymbol{z}|\boldsymbol{x}, \theta^t) = \dfrac{p(\boldsymbol{z}, \boldsymbol{x}|\theta^t)}{\sum_z p(\boldsymbol{z}, \boldsymbol{x}|\theta^t)}$

    **Intractable** when model $p(\boldsymbol{z}, \boldsymbol{x}|\theta)$ is complex
  - M-step: $\theta^{t+1} = \arg\min_\theta F\left(q^{t+1}, \theta^t\right)$

Approximate $p(\boldsymbol{z}|\boldsymbol{x}, \theta^t)$:

- find a **tractable** $q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{v}^*)$ that is closest to $p(\boldsymbol{z}|\boldsymbol{x}, \theta^t)$

$$q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{v}^*) = \min_{\boldsymbol{v}} \mathrm{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{v}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta^t)\big)$$

$$= \min_{\boldsymbol{v}} F(q(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{v}), \theta^t) + const.$$

$$\ell(\theta; \boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}|\theta)}{q(\boldsymbol{z}|\boldsymbol{x})}\right] + \mathrm{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big)$$

$$= -F(q, \theta) + \mathrm{KL}\big(q(\boldsymbol{z}|\boldsymbol{x}) \,||\, p(\boldsymbol{z}|\boldsymbol{x}, \theta)\big)$$

# Recap: EM and Variational Inference

- The EM algorithm:
  - E-step: $q^{t+1} = \arg\min_q F\left(q, \theta^t\right)$

    $= p(\mathbf{z}|\mathbf{x}, \theta^t) = \dfrac{p(\mathbf{z}, \mathbf{x}|\theta^t)}{\sum_z p(\mathbf{z}, \mathbf{x}|\theta^t)}$

  - M-step: $\theta^{t+1} = \arg\min_\theta F\left(q^{t+1}, \theta^t\right)$

**Intractable** when model $p(\mathbf{z}, \mathbf{x}|\theta)$ is complex

Approximate $p(\mathbf{z}|\mathbf{x}, \theta^t)$:
- find a **tractable** $q(\mathbf{z}|\mathbf{x}, \boldsymbol{v}^*)$ that is closest to $p(\mathbf{z}|\mathbf{x}, \theta^t)$

$$q(\mathbf{z}|\mathbf{x}, \boldsymbol{v}^*) = \min_{\boldsymbol{v}} \mathrm{KL}\big(q(\mathbf{z}|\mathbf{x}, \boldsymbol{v}) \,||\, p(\mathbf{z}|\mathbf{x}, \theta^t)\big)$$

$$= \min_{\boldsymbol{v}} F\big(q(\mathbf{z}|\mathbf{x}, \boldsymbol{v}), \theta^t\big) + const.$$

**Question:** what is the difference?

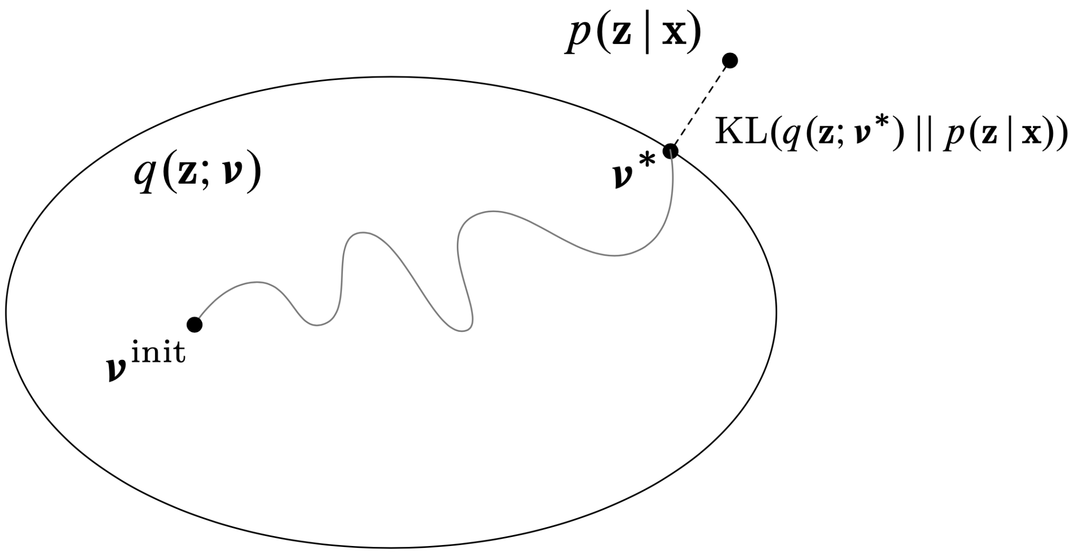
A: VI assume a simple form of $q$ to ensure tractability

# Recap: EM and Variational Inference

- The EM algorithm:
  - E-step: $q^{t+1} = \arg\min_q F\left(q, \theta^t\right)$

$$= p(\mathbf{z}|\mathbf{x}, \theta^t) = \frac{p(\mathbf{z}, \mathbf{x}|\theta^t)}{\sum_{\mathbf{z}} p(\mathbf{z}, \mathbf{x}|\theta^t)}$$

**Intractable** when model $p(\mathbf{z}, \mathbf{x}|\theta)$ is complex




Approximate $p(\mathbf{z}|\mathbf{x}, \theta^t)$:

- find a **tractable** $q(\mathbf{z}|\mathbf{x}, \boldsymbol{\nu}^*)$ that is closest to $p(\mathbf{z}|\mathbf{x}, \theta^t)$

$$q(\mathbf{z}|\mathbf{x}, \boldsymbol{\nu}^*) = \min_{\boldsymbol{\nu}} \mathrm{KL}\big(q(\mathbf{z}|\mathbf{x}, \boldsymbol{\nu})\;||\;p(\mathbf{z}|\mathbf{x}, \theta^t)\big)$$

$$= \min_{\boldsymbol{\nu}} F\left(q(\mathbf{z}|\mathbf{x}, \boldsymbol{\nu}), \theta^t\right) + const.$$

**Question:** what is the difference?

**A:** VI assume a simple form of $q$ to ensure tractability

# Recap: EM and Variational Inference

- The EM algorithm:
  - E-step: $q^{t+1} = \arg\min_q F\left(q, \theta^t\right)$

$$= p(\mathbf{z}|\mathbf{x}, \theta^t) = \frac{p(\mathbf{z}, \mathbf{x}|\theta^t)}{\sum_z p(\mathbf{z}, \mathbf{x}|\theta^t)}$$

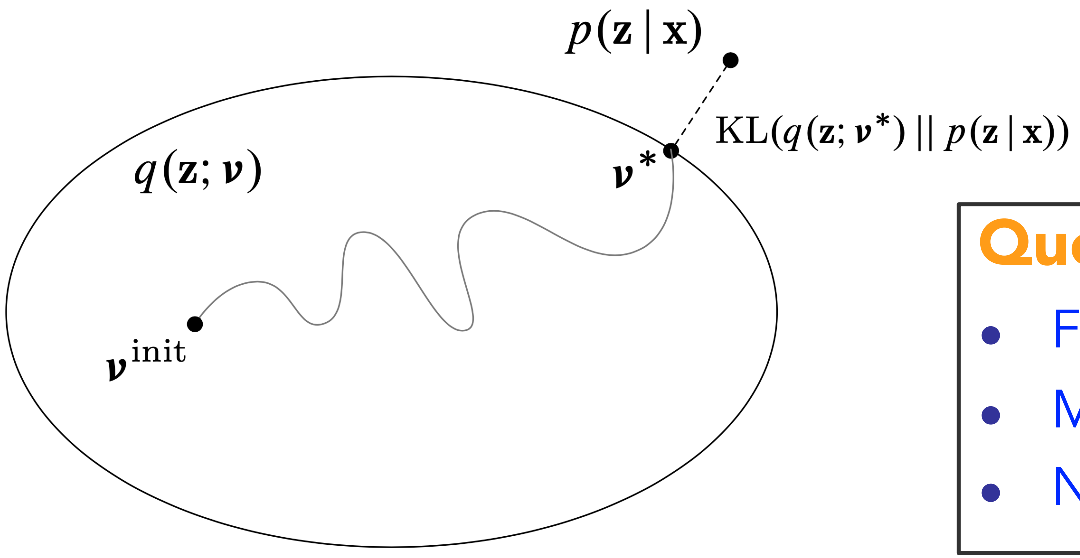
**Intractable** when model $p(\mathbf{z}, \mathbf{x}|\theta)$ is complex

Approximate $p(\mathbf{z}|\mathbf{x}, \theta^t)$:

- find a **tractable** $q(\mathbf{z}|\mathbf{x}, \mathbf{v}^*)$ that is closest to $p(\mathbf{z}|\mathbf{x}, \theta^t)$

$$q(\mathbf{z}|\mathbf{x}, \mathbf{v}^*) = \min_{\mathbf{v}} \mathrm{KL}\big(q(\mathbf{z}|\mathbf{x}, \mathbf{v}) \,||\, p(\mathbf{z}|\mathbf{x}, \theta^t)\big)$$

$$= \min_{\mathbf{v}} F(q(\mathbf{z}|\mathbf{x}, \mathbf{v}), \theta^t) + const.$$




**Question:** What forms of $q(\mathbf{z}|\mathbf{x}, \mathbf{v})$ shall we choose?

- Factorized distribution -> mean field VI
- Mixture of Gaussian distribution -> black-box VI
- Neural-based distribution -> Variational Autoencoders

# Example: Mean Field Variational Inference

- A popular family of variational approximations

- In this type of variational inference, we assume the variational distribution over the latent variables factorizes as

$$q(\mathbf{z}) = q(z_1, \ldots, z_m) = \prod_{j=1}^{m} q(z_j)$$

  - (where we omit variational parameters for ease of notation)
  - We refer to $q(z_j)$, the variational approximation for a single latent variable, as a "local variational approximation"

- In the above expression, the variational approximation $q(z_j)$ over each latent variable $z_j$ is independent

# Example: Mean Field Variational Inference

- Typically, this approximation does not contain the true posterior (because the latent variables are dependent).
    - E.g.: in the (Bayesian) mixture of Gaussians model, all of the cluster assignments $z_i$ for $i = 1, \ldots, n$ are dependent on each other and on the cluster locations $\mu_{1:K}$ given data.

# Example: Mean Field Variational Inference

**How do we optimize the ELBO in mean field variational inference?**

- Typically, we use coordinate ascent optimization.
- I.e. we optimize each latent variable's variational approximation $q(z_j)$ in turn while holding the others fixed.
  - At each iteration we get an updated "local" variational approximation.
  - And we iterate through each latent variable until convergence.

# Mean Field Variational Inference with Coordinate Ascent

Recap: Bayesian mixture of Gaussians

- Treat the mean $\mu_k$ as latent variables

$$\mu_k \sim \mathcal{N}(0, \tau^2) \text{ for } k = 1, \ldots, K$$

- For each data $i = 1, \ldots, n$

$$z_i \sim \mathrm{Cat}(\pi).$$

$$x_i \sim \mathcal{N}(\mu_{z_i}, \sigma^2).$$

- We have
    - observed variables $x_{1:n}$
    - latent variables $\mu_{1:k}$ and $z_{1:n}$
    - parameters $\{\tau^2, \sigma^2, \pi\}$

# Mean Field Variational Inference with Coordinate Ascent

Recap: Bayesian mixture of Gaussians

Assume mean-field $q(\mu_{1:K}, z_{1:n}) = \prod_k q(\mu_k) \prod_i q(z_i)$

- Initialize the global variational distributions $q(\mu_k)$ and parameters $\{\tau^2, \sigma^2, \pi\}$
- Repeat:
  - For each data example $i \in \{1,2,\ldots,D\}$
    - Update the local variational distribution $q(z_i)$
  - End for
  - Update the global variational distributions $q(\mu_k)$
  - Update the parameters $\{\tau^2, \sigma^2, \pi\}$
- Until ELBO converges

(Variational) E-step

(Variational) M-step

- What if we have millions of data examples? This could be very slow.

# Stochastic VI

Recap: Bayesian mixture of Gaussians

Assume mean-field $q(\mu_{1:K}, z_{1:n}) = \prod_k q(\mu_k) \prod_i q(z_i)$

- Initialize the global variational distributions $q(\mu_k)$ and parameters $\{\tau^2, \sigma^2, \pi\}$
- Repeat:
    - Sample a data example $i \in \{1,2, \dots, D\}$
    - Update the local variational distribution $q(z_i)$
    - Update the global variational distributions $q(\mu_k)$ with **natural gradient ascent**
    - Update the parameters $\{\tau^2, \sigma^2, \pi\}$
- Until ELBO converges

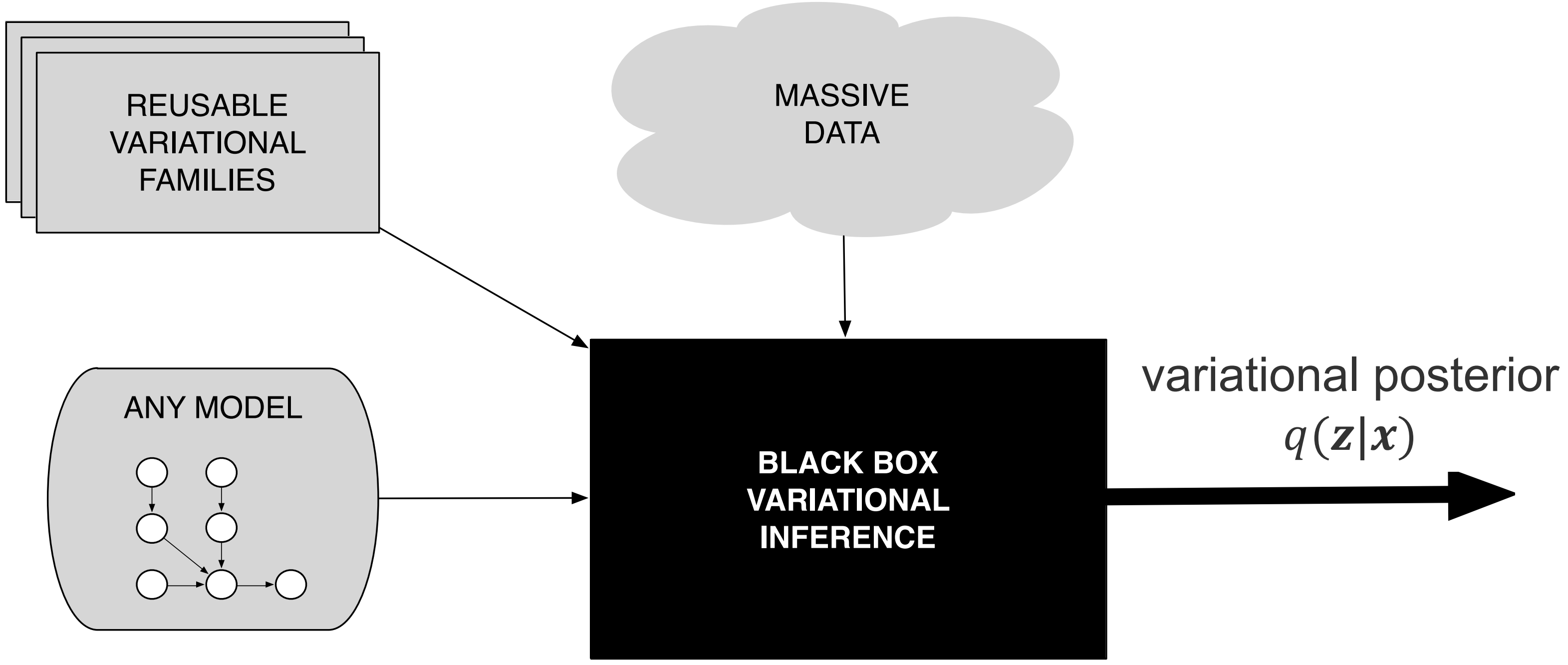
[Hoffman et al., Stochastic Variational Inference, 2013]

# Black-box Variational Inference

# Black-box Variational Inference (BBVI)

- We have derived variational inference specific for Bayesian Gaussian (mixture) models
- There are innumerable models
- Can we have a solution that does not entail model-specific work?

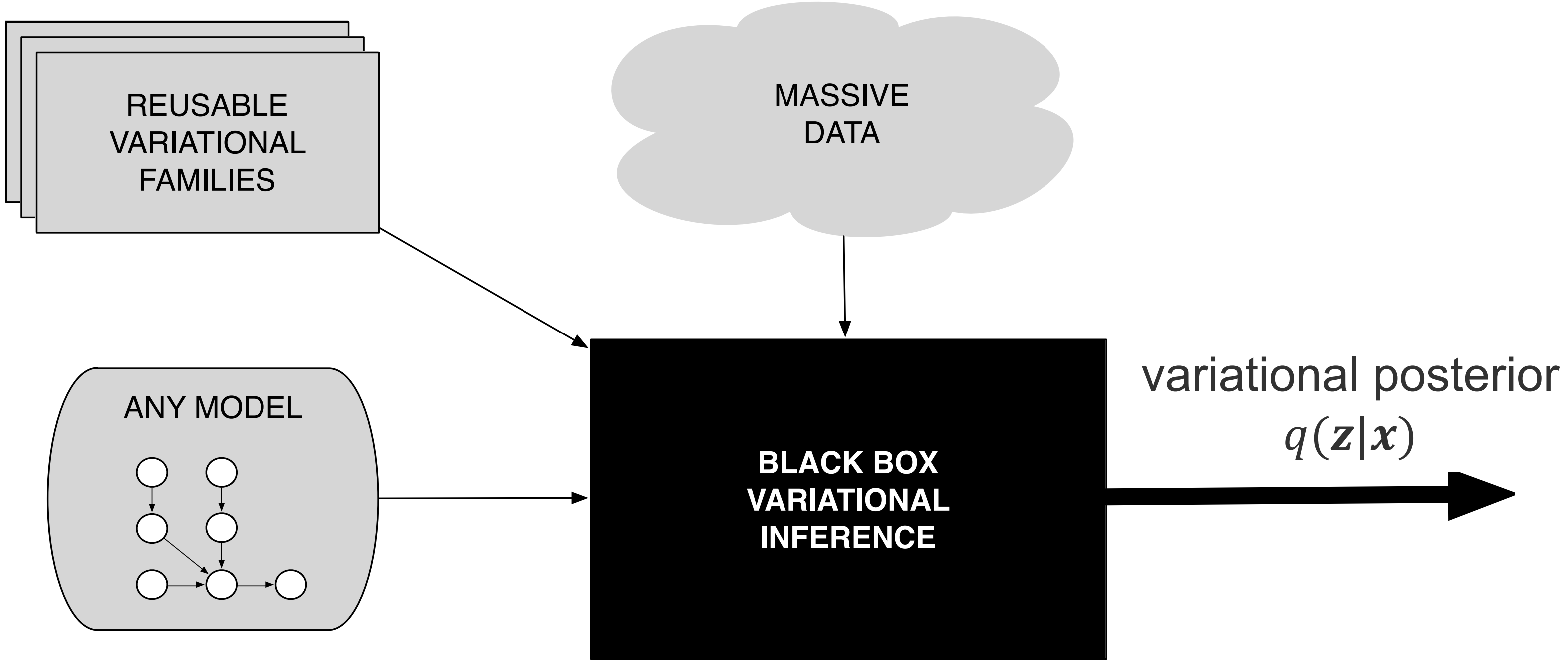
# Black-box Variational Inference (BBVI)




- Easily use variational inference with **any model**
- Perform inference with **massive data**
- **No mathematical work** beyond specifying the model

(Courtesy: Blei et al., 2018)

# Black-box Variational Inference (BBVI)




- Sample from $q(.)$
- Form noisy gradients (without model-specific computation)
- Use stochastic optimization

(Courtesy: Blei et al., 2018)

# Black-box Variational Inference (BBVI)

- Probabilistic model: $\boldsymbol{x}$ -- observed variables, $\boldsymbol{z}$ -- latent variables
- Variational distribution $q_\lambda(\boldsymbol{z}|\boldsymbol{x})$ with parameters $\lambda$, e.g.,
  - Gaussian mixture distribution:
    - "A Gaussian mixture model is a universal approximator of densities, in the sense that any smooth density can be approximated with any specific nonzero amount of error by a Gaussian mixture model with enough components." (Deep Learning book, pp.65)
  - Deep neural networks

- ELBO:

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log p(\boldsymbol{x}, \boldsymbol{z})] - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log q(\boldsymbol{z}|\lambda)]$$

- Want to compute the gradient w.r.t variational parameters $\lambda$

[Ranganath et al.,14]

# The General Problem: Computing Gradients of Expectations

- When the objective function $\mathcal{L}$ is defined as an expectation of a (differentiable) test function $f_\lambda(\boldsymbol{z})$ w.r.t. a probability distribution $q_\lambda(\boldsymbol{z})$

$$\mathcal{L} = \mathbb{E}_{q_{\boldsymbol{\lambda}}(\boldsymbol{z})}[f_\lambda(\boldsymbol{z})]$$

- Computing exact gradients w.r.t. the parameters $\lambda$ is often unfeasible
- Need stochastic gradient estimates
  - The score function estimator (a.k.a log-derivative trick, REINFORCE)
  - The reparameterization trick (a.k.a the pathwise gradient estimator)

# Computing Gradients of Expectations w/ score function

- Loss: $\mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z})]$

- Log-derivative trick: $\nabla_\lambda q_\lambda = q_\lambda \nabla_\lambda \log q_\lambda$
- Gradient w.r.t. $\lambda$:

$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z}) \underline{\nabla_\lambda \log q_\lambda(\boldsymbol{z})} + \nabla_\lambda f_\lambda(\boldsymbol{z})]$$

  - score function: the gradient of the log of a probability distribution
- Compute noisy unbiased gradients with Monte Carlo samples from $q_\lambda$

$$\nabla_\lambda \mathcal{L} \approx \frac{1}{S} \sum_{s=1}^{S} f_\lambda(\boldsymbol{z}_s) \nabla_\lambda \log q_\lambda(\boldsymbol{z}_s) + \nabla_\lambda f_\lambda(\boldsymbol{z}_s) \qquad \text{where } \boldsymbol{z}_s \sim q_\lambda(\boldsymbol{z})$$

- Pros: generally applicable to any distribution $q(z|\lambda)$
- Cons: empirically has high variance → slow convergence
  - To reduce variance: Rao-Blackwellization, control variates, importance sampling, ...

# Computing Gradients of Expectations w/ reparametrization trick

- Loss: $\mathcal{L} = \mathbb{E}_{q_\lambda(\boldsymbol{z})}[f_\lambda(\boldsymbol{z})]$

- Assume that we can express the distribution $q_{\boldsymbol{\lambda}}(\boldsymbol{z})$ with a transformation

$$\begin{aligned} \epsilon &\sim s(\epsilon) \\ z &= t(\epsilon, \lambda) \end{aligned} \quad \Leftrightarrow \quad z \sim q(z|\lambda)$$

  - E.g.,

$$\begin{aligned} \epsilon &\sim Normal(0, 1) \\ z &= \epsilon\sigma + \mu \end{aligned} \quad \Leftrightarrow \quad z \sim Normal(\mu, \sigma^2)$$

- Reparameterization gradient

$$\mathcal{L} = \mathbb{E}_{\boldsymbol{\epsilon} \sim s(\boldsymbol{\epsilon})}[f_\lambda(\boldsymbol{z}(\boldsymbol{\epsilon}, \lambda))]$$

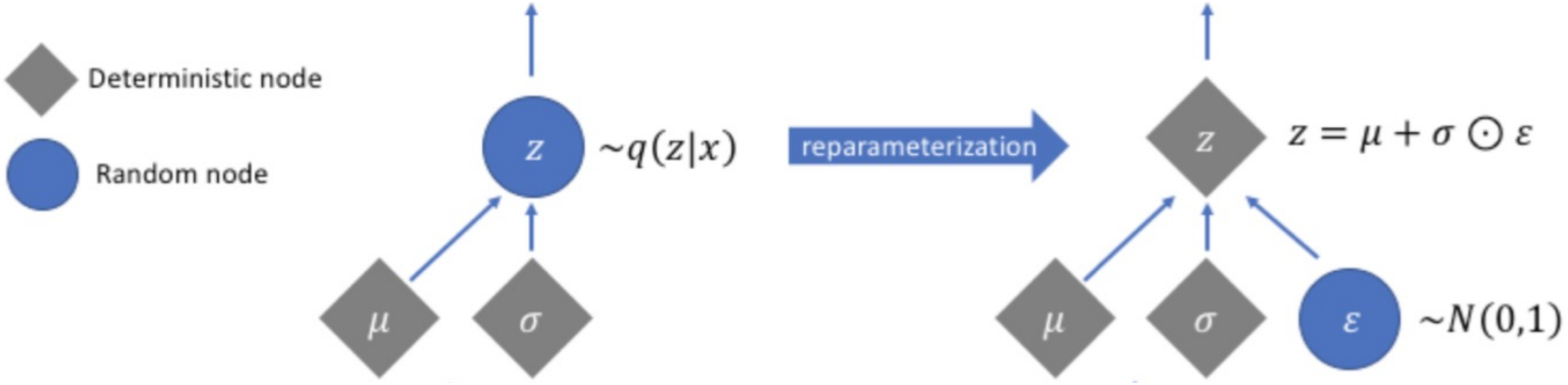
$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{\boldsymbol{\epsilon} \sim \boldsymbol{s}(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{z}} f_\lambda(\boldsymbol{z})\, \nabla_\lambda t(\epsilon, \lambda)]$$

- Pros: empirically, lower variance of the gradient estimate
- Cons: Not all distributions can be reparameterized

# Reparameterization trick

- Reparametrizing Gaussian distribution

$$\begin{array}{l} \epsilon \sim Normal(0,1) \\ z = \epsilon\sigma + \mu \end{array} \iff z \sim Normal(\mu, \sigma^2)$$




[Courtesy: Tansey, 2016]

# Reparameterization trick

- Reparametrizing Gaussian distribution

$$\begin{array}{l} \epsilon \sim Normal(0,1) \\ z = \epsilon\sigma + \mu \end{array} \Leftrightarrow \quad z \sim Normal(\mu, \sigma^2)$$

- Other reparameterizable distributions:
  - Tractable inverse CDF $F^{-1}$:
    - Exponential, Cauchy, Logistic, Rayleigh, Pareto, Weibull, Reciprocal, Gompertz, Gumbel, Erlang
  - Location-scale:
    - Laplace, Elliptical, Student's t, Logistic, Uniform, Triangular, Gaussian
  - Composition:
    - Log-Normal (exponentiated normal) Gamma (sum of exponentials) Dirichlet (sum of Gammas) Beta, Chi-Squared, F

$$\begin{array}{l} \epsilon \sim Uniform(\epsilon) \\ z = F^{-1}(\epsilon) \end{array} \Leftrightarrow \quad z \sim q(z)$$

[Courtesy: Tansey, 2016]

# Computing Gradients of Expectations: Summary

- Loss: $\mathcal{L} = \mathbb{E}_{q_{\boldsymbol{\lambda}}(\boldsymbol{z})}[f_\lambda(\boldsymbol{z})]$
- Score gradient

$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{q_{\boldsymbol{\lambda}}(\boldsymbol{z})}[f_\lambda(\boldsymbol{z})\nabla_\lambda \log q_\lambda(\boldsymbol{z}) + \nabla_\lambda f_\lambda(\boldsymbol{z})]$$

  - Pros: generally applicable to any distribution $q(z|\lambda)$
  - Cons: empirically has high variance → slow convergence

- Reparameterization gradient

$$\nabla_\lambda \mathcal{L} = \mathbb{E}_{\boldsymbol{\epsilon} \sim \boldsymbol{s}(\boldsymbol{\epsilon})}[\nabla_{\boldsymbol{z}} f_\lambda(\boldsymbol{z})\, \nabla_\lambda t(\epsilon, \lambda)]$$

  - Pros: empirically, lower variance of the gradient estimate
  - Cons: Not all distributions can be reparameterized

# Recall: Black-box Variational Inference (BBVI)

- Probabilistic model: $\boldsymbol{x}$ -- observed variables, $\boldsymbol{z}$ -- latent variables
- Variational distribution $q_\lambda(\boldsymbol{z}|\boldsymbol{x})$ with parameters $\lambda$, e.g.,
  - Gaussian mixture distribution:
    - "A Gaussian mixture model is a universal approximator of densities, in the sense that any smooth density can be approximated with any specific nonzero amount of error by a Gaussian mixture model with enough components." (Deep Learning book, pp.65)
  - Deep neural networks

$$\mathcal{L}(\lambda) \triangleq \mathrm{E}_{q_\lambda(z)}[\log p(x, z) - \log q(z)]$$

- ELBO:

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log p(\boldsymbol{x}, \boldsymbol{z})] - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log q(\boldsymbol{z}|\lambda)]$$

- Want to compute the gradient w.r.t variational parameters $\lambda$

[Ranganath et al.,14]

# BBVI with the score gradient

- ELBO:

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log p(\boldsymbol{x}, \boldsymbol{z})] - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log q(\boldsymbol{z}|\lambda)]$$

- Gradient w.r.t. $\lambda$ (using the log-derivative trick)

$$\nabla_\lambda \mathcal{L} = \mathrm{E}_q[\nabla_\lambda \log q(z|\lambda)(\log p(x, z) - \log q(z|\lambda))]$$

- Compute noisy unbiased gradients of the ELBO with Monte Carlo samples from the variational distribution

$$\nabla_\lambda \mathcal{L} \approx \frac{1}{S} \sum_{s=1}^{S} \nabla_\lambda \log q(z_s|\lambda)(\log p(x, z_s) - \log q(z_s|\lambda)),$$

$$\text{where } z_s \sim q(z|\lambda).$$

[Ranganath et al.,14]

# BBVI with the reparameterization gradient

- ELBO:

$$\mathcal{L}(\lambda) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log p(\boldsymbol{x}, \boldsymbol{z})] - \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{\lambda})}[\log q(\boldsymbol{z}|\lambda)]$$

- Gradient w.r.t. $\lambda$

$$\begin{aligned} &\epsilon \sim s(\epsilon) \\ &z = t(\epsilon, \lambda) \end{aligned} \quad \Leftrightarrow \quad z \sim q(z|\lambda)$$

$$\nabla_\lambda \mathcal{L} = \mathrm{E}_{\epsilon \sim s(\epsilon)}[\nabla_z[\log p(x, z) - \log q(z)] \nabla_\lambda t(\epsilon, \lambda)]$$

# Questions?