

# Advances in Approximate Inference

Yingzhen Li, Cheng Zhang  
Microsoft Research Cambridge

# What is the Number?



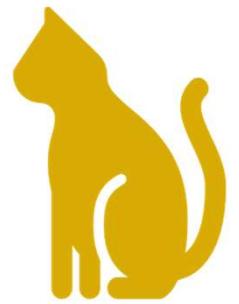
07? 09?

67? 69?

...



What is the Number?



09

# What is the Diagnosis?



Injury?  
Osteoarthritis?  
Neuropathic pain?  
....



# What is the Diagnosis?



Neuropathic pain  
(might have spine injury)



# Uncertainty is Important





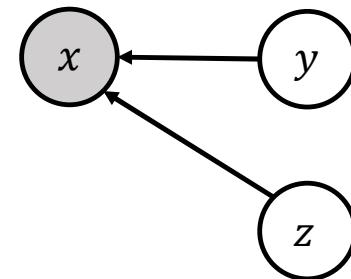
Bayesian ML / Probability Theory



Decision making under uncertainty

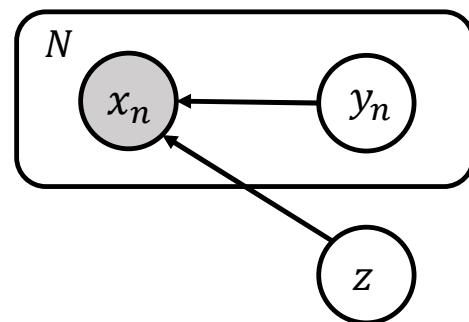
Image courtesy of Sebastian Nowozin  
Re-use of the image for any other purpose is not allowed

# Graphical Representation



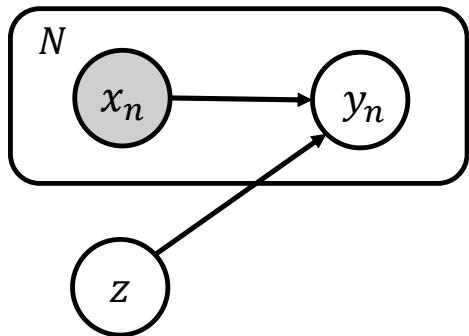
$$p(x, y, z) = p(y)p(z)p(x|y, z)$$

# Graphical Representation

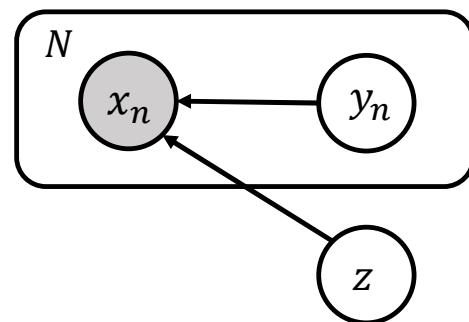


$$p(\mathbf{x}, \mathbf{y}, z) = p(z) \prod_n^N p(y_n) P(y_n | y_n, z)$$

# Graphical Representation

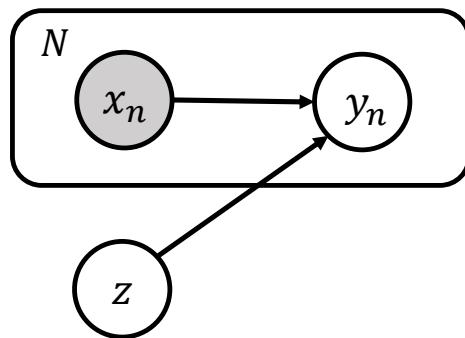
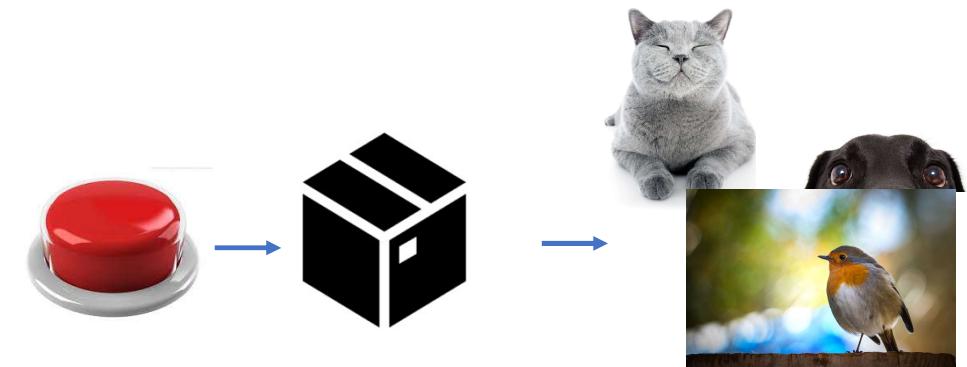
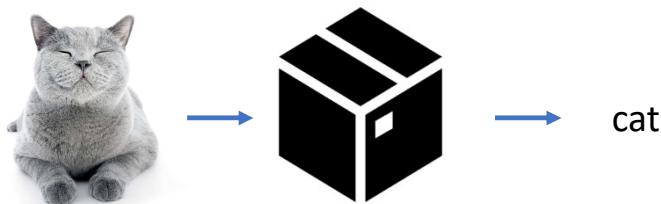


$$p(\mathbf{x}, \mathbf{y}, z) = p(z) \prod_{n=1}^N p(x_n) P(y_n | x_n, z)$$

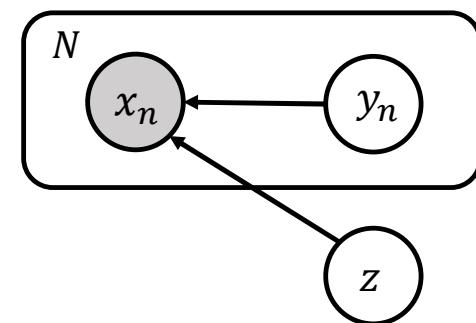


$$p(\mathbf{x}, \mathbf{y}, z) = p(z) \prod_{n=1}^N p(y_n) P(x_n | y_n, z)$$

# Discriminative Model vs Generative Model



$$p(x, y, z) = p(z) \prod_{n=1}^N p(x_n)P(y_n|x_n, z)$$

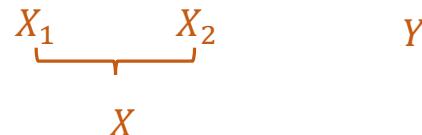


$$p(x, y, z) = p(z) \prod_{n=1}^N p(y_n)P(x_n|y_n, z)$$

# Discriminative Model Example

- Bayesian Logistic Regression

Name	A-level math score	# parents in STEM	Study STEM?
Alice	89	0	0 (No)
Bob	95	1	1 (Yes)
Ty	82	1	0 (No)
Emma	98	2	1 (Yes)
Anna	92	0	0 (No)
Mo	88	1	0 (No)
Li	95	0	1 (Yes)

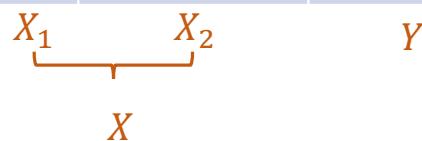


# Discriminative Model Example:

- Bayesian Logistic Regression

Name	A-level math score	# parents in STEM	Study STEM?
Alice	89	0	0 (No)
Bob	95	1	1 (Yes)
Ty	82	1	0 (No)
Emma	98	2	1 (Yes)
Anna	92	0	0 (No)
Mo	88	1	0 (No)
Li	95	0	1 (Yes)

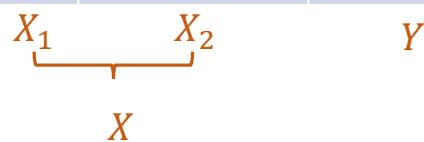
$$p(y = 1) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}$$



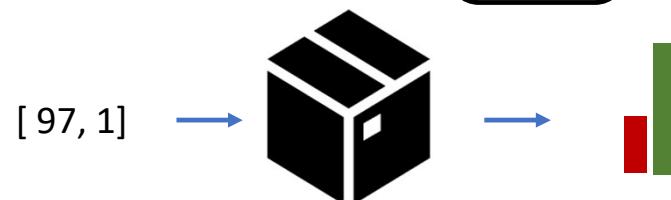
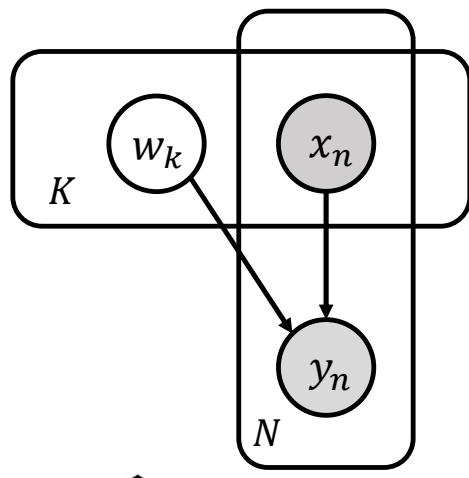
# Discriminative Model Example

- Bayesian Logistic Regression

Name	A-level math score	# parents in STEM	Study STEM?
Alice	89	0	0 (No)
Bob	95	1	1 (Yes)
Ty	82	1	0 (No)
Emma	98	2	1 (Yes)
Anna	92	0	0 (No)
Mo	88	1	0 (No)
Li	95	0	1 (Yes)

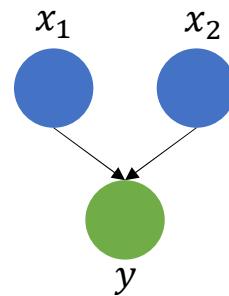


$$p(y = 1) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}$$

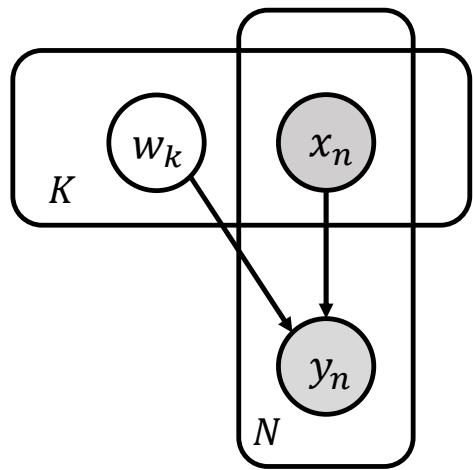


# Discriminative Model Example

Computation Graph

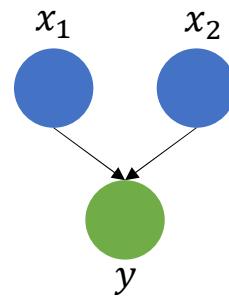


$$p(y = 1) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}$$

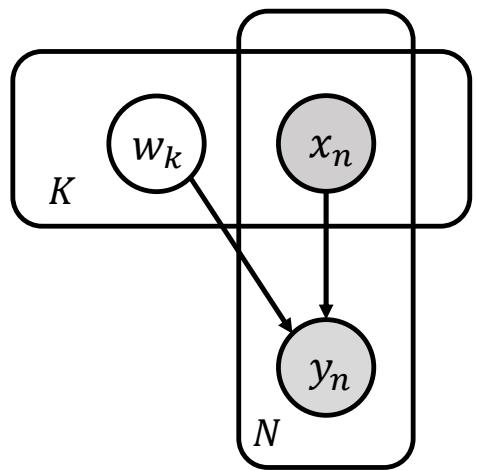


# Discriminative Model Example

Computation Graph

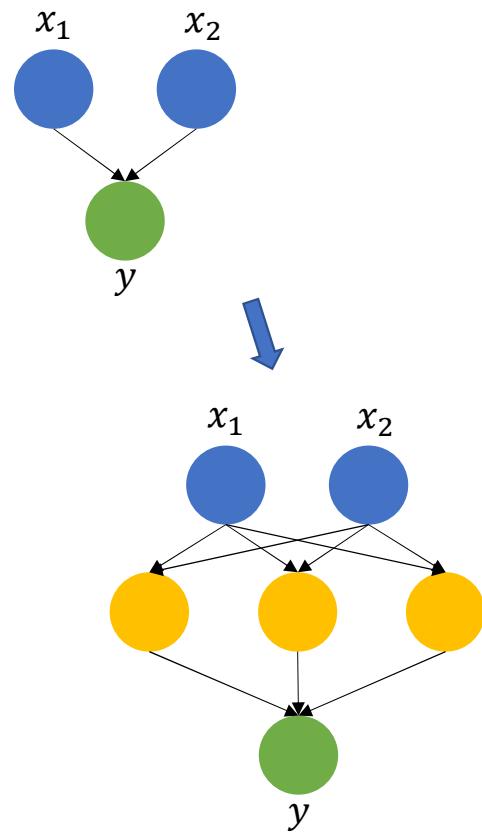


$$p(y=1) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}$$

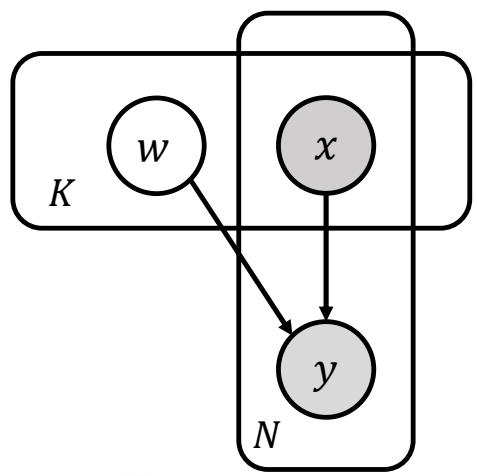


# Discriminative Model Example:

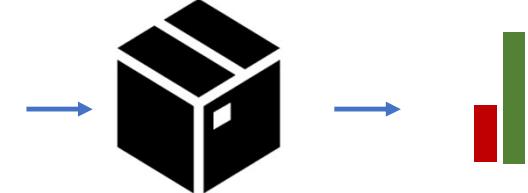
Computation Graph



$$p(y = 1) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}$$



[ 97, 1 ]

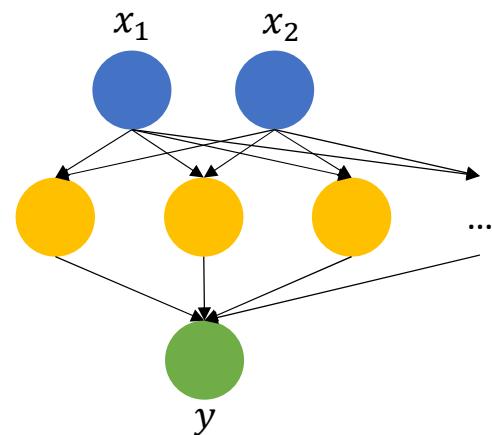


$W^T X$

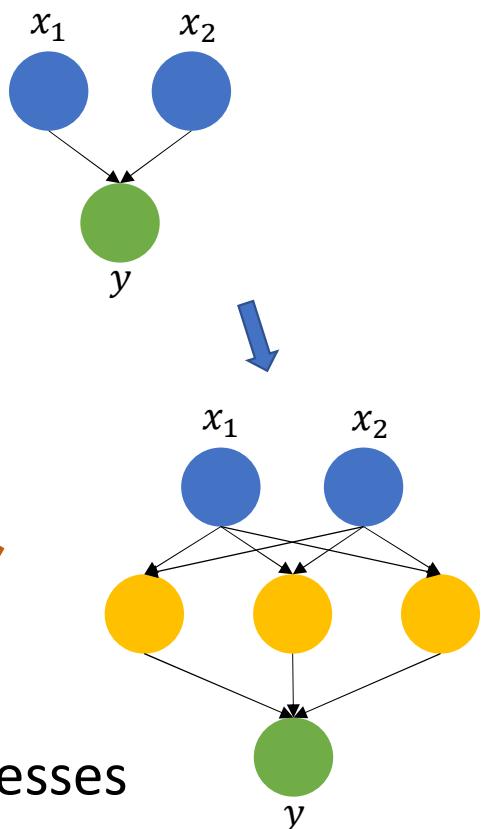
$W^T \Phi(X)$

# Discriminative Model Example

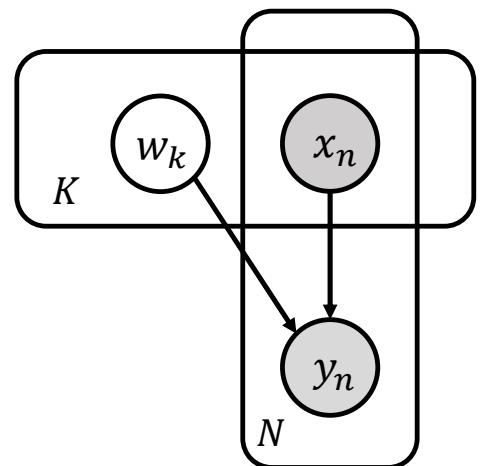
Computation Graph



Gaussian Processes



$$p(y = 1) = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2)}}$$



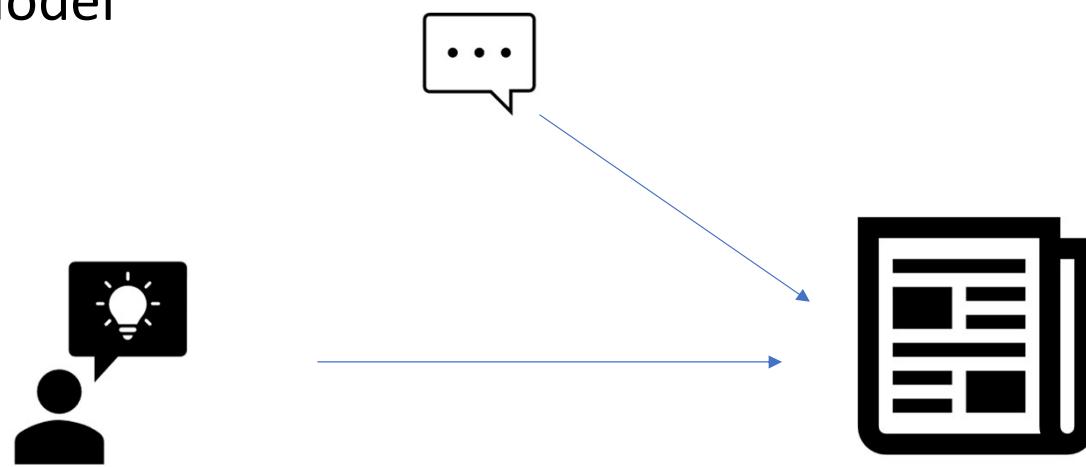
[ 97, 1 ]



Radford Neal's PhD thesis (1994)

# Generative Model Example

- Topic Model

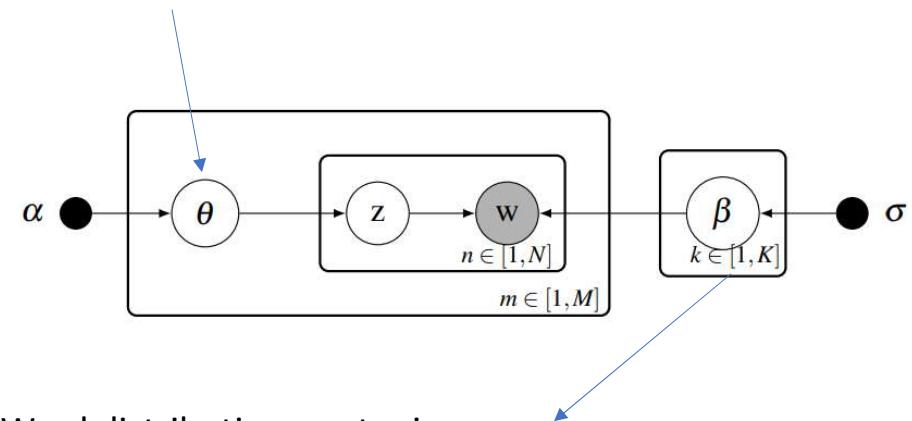


# Generative Model Example:

- Latent Dirichlet Allocation

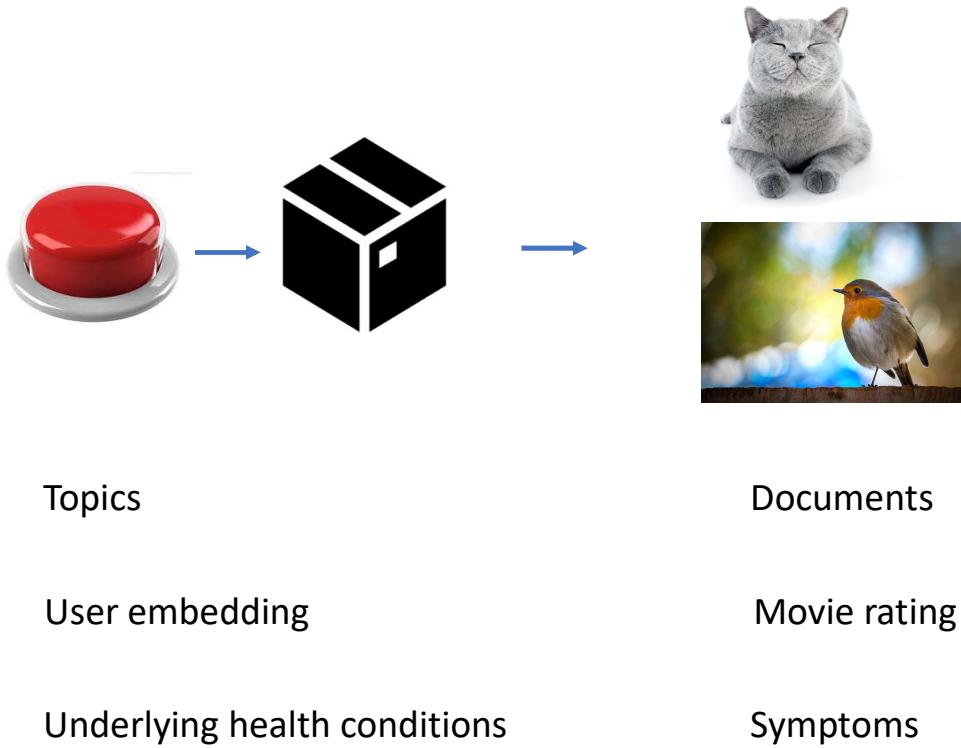
ID	topic	neural	distribution	...
1	15	2	19	...
2	1	13	21	...
3	0	16	1	...

Topic distribution per document:  
e.g. 30% “topic model”, 40% “natural language processing”,  
30% “interpretability”



Word distribution per topic:  
e.g. under “topic model”: “Dirichlet” 2%, “topic” 4%,  
“Categorical” 1.5%, .....

# Generative Model Example



# How to infer the unknowns?



# The Central Computation for Inference

- Inference: infer the **unknowns**
  - Unobserved/latent variables in the model
  - Quantities depending on the latent variables in the model

# The Central Computation for Inference

- Inference: infer the **unknowns**
  - Unobserved/latent variables in the model
  - Quantities depending on the latent variables in the model

$$\int F(\theta) \underbrace{\pi(\theta)}_{\text{probability measure}} d\theta$$

prob. density

Random variable (unobserved)

integrand function

(For discrete probability measures, integration becomes discrete sum.)

# Bayesian Inference

$$\pi(\theta) = p(\theta | \text{data})$$

$$P(\theta | \text{data}) = \frac{P(\theta)P(\text{data} | \theta)}{P(\text{data})}$$

- $P(\theta)$ : prior distribution
- $P(\text{data} | \theta)$ : likelihood of  $\theta$  given  $\text{data}$
- $P(\theta | \text{data})$ : posterior distribution of  $\theta$  given  $\text{data}$
- $P(\text{data})$ : marginal likelihood/model evidence

$$P(\text{data}) = \int P(\theta)P(\text{data} | \theta)$$



Image courtesy of Sebastian Nowozin

Re-use of the image for any other purpose is not allowed

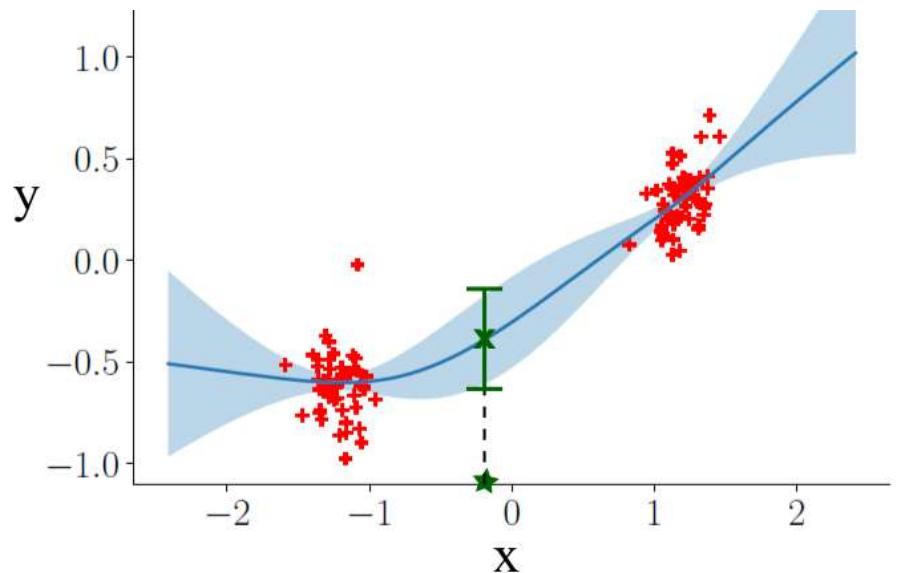
# Computation Challenge

- The central equation for inference:

$$\int F(\theta) \pi(\theta) d\theta$$

“What is the prediction distribution of the test output given a test input?”

$F(\theta) = p(y|x, \theta)$ ,  $\pi(\theta) = p(\theta | D)$ ,  
 $D$  = observed datapoints



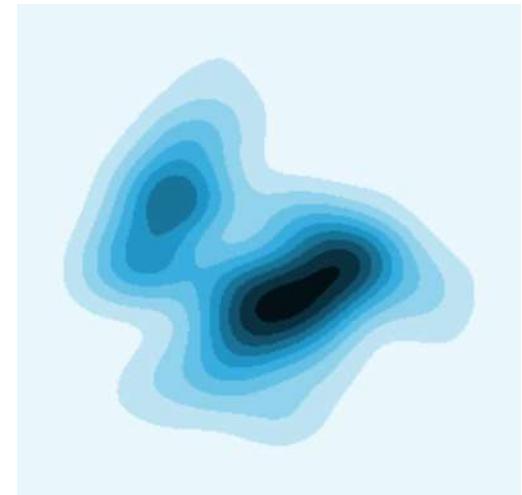
# Computation Challenge

- The central equation for inference:

$$\int F(\theta) \pi(\theta) d\theta$$

“What is the mean of this distribution?”

$F(\theta) = \theta$ ,  $\pi(\theta)$  can be complicated and high dimensional



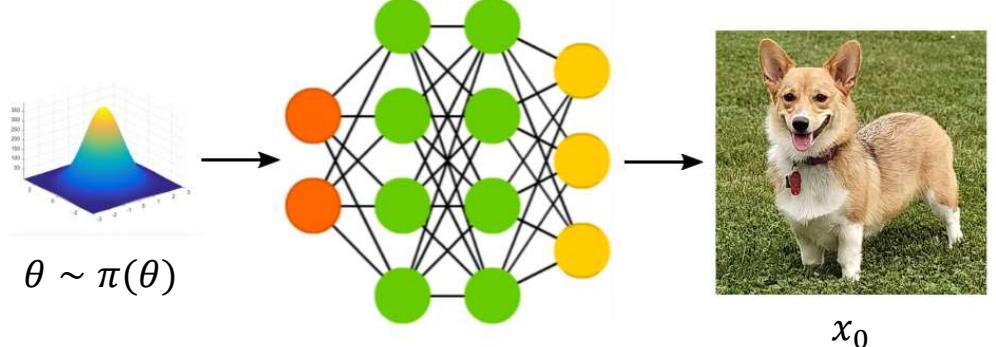
# Computation Challenge

- The central equation for inference:

$$\int F(\theta) \pi(\theta) d\theta$$

“What is the probability of generating this image?”

$$F(\theta) = \delta(NN(\theta) = x_0), \pi(\theta) = N(0, I)$$



# Computation Challenge

- The central equation for inference:

$$\int F(\theta) \pi(\theta) d\theta$$

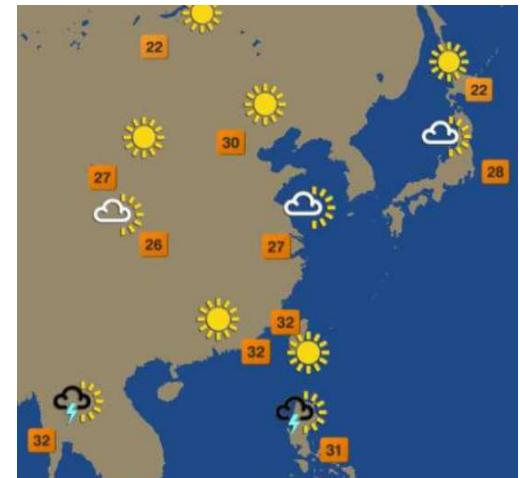
“What is the weather forecast for tomorrow?”

Answering this in a Bayesian way:

$\theta$ : forecasting simulator settings

$D$ : historical weather record

$$F(\theta) = \text{Simulator}(\theta), \pi(\theta) = p(\theta | D)$$



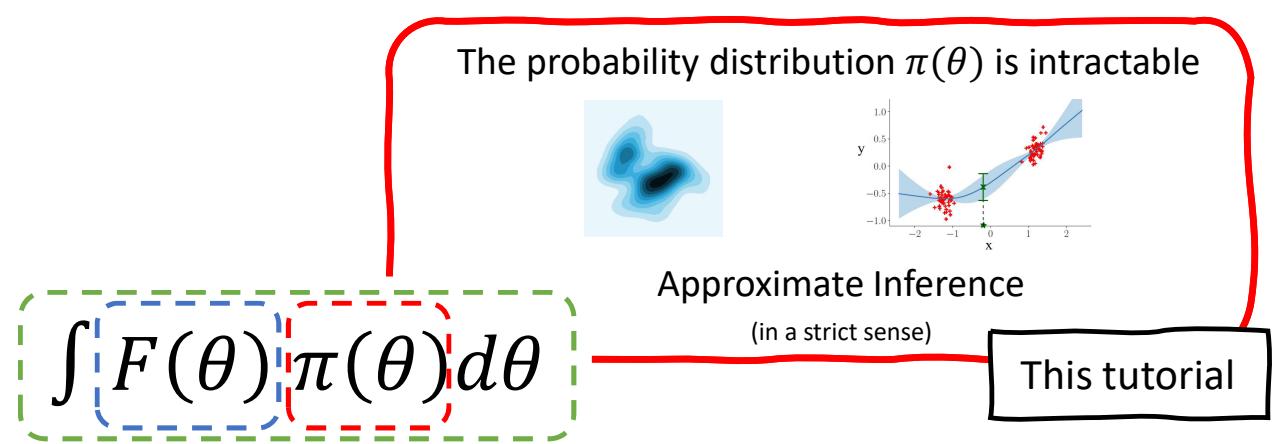
Nature laughs at the  
difficulties of integration.

--Pierre-Simon Laplace

Gordon and Sorkin. *The Armchair Science Reader*. New York 1959

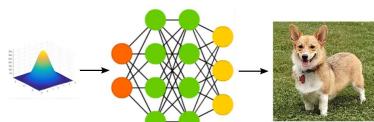


# Integration in Bayesian Computation



# Integration in Bayesian Computation

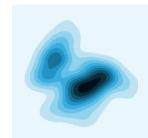
The integrand  $F(\theta)$  is intractable



Implicit Models

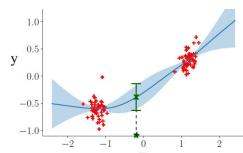
Bayesian Optimisation, Probabilistic Numerics

The probability distribution  $\pi(\theta)$  is intractable



Approximate Inference

(in a strict sense)

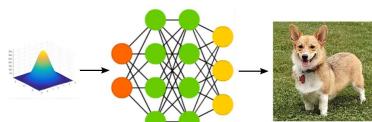


This tutorial

$$\int [F(\theta) \pi(\theta)] d\theta$$

# Integration in Bayesian Computation

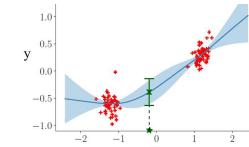
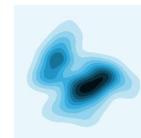
The integrand  $F(\theta)$  is intractable



Implicit Models

Bayesian Optimization, Probabilistic Numerics

The probability distribution  $\pi(\theta)$  is intractable



Approximate Inference

(in a strict sense)

This tutorial

$$\int [F(\theta) \pi(\theta)] d\theta$$

Both  $F(\theta)$  and  $\pi(\theta)$  are intractable



Approximate Bayesian Computation

# Approximate Inference

- Central task: approximate  $\pi(\theta)$

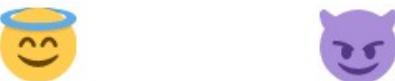


$$q(\theta) \approx \pi(\theta)$$

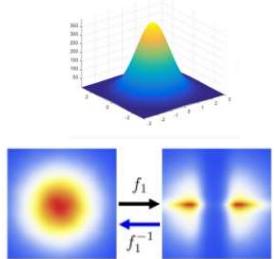
(Assumed  $\int F(\theta)q(\theta)d\theta$  can be computed or approximated efficiently.)

# Approximate Inference

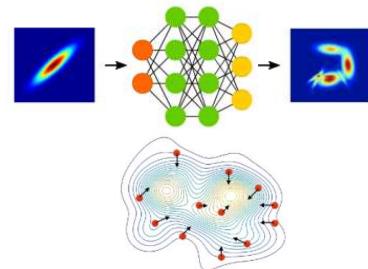
- Central task: approximate  $\pi(\theta)$


$$\underline{q(\theta)} \approx \pi(\theta)$$

Approximate distribution design



Explicit distributions



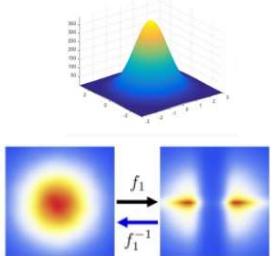
Implicit distributions

# Approximate Inference

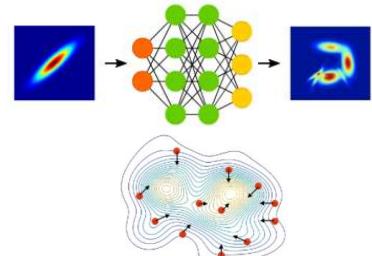
- Central task: approximate  $\pi(\theta)$

$$\underline{q(\theta)} \approx \underline{\pi(\theta)}$$

Approximate distribution design



Explicit distributions

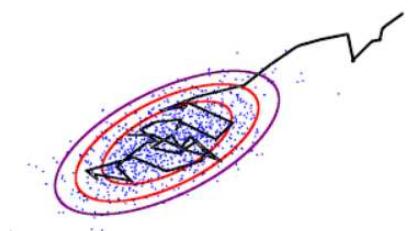


Implicit distributions

Algorithm for fitting  $q(\theta)$  to  $\pi(\theta)$

$$\min Loss(q(\theta), \pi(\theta))$$

Optimisation-based  
approaches



Sampling-based  
approaches

# Tutorial Outline



## Basics

Probabilistic modelling  
Approximate inference  
Variational inference



## Advances

Scalable variational inference  
Monte Carlo techniques  
Amortized inference  
 $q$  distribution design  
Optimization objective design



## Applications

Bayesian neural networks  
Partially observed VAEs  
Future challenges

# Bayesian Inference

$$P(\theta | D) = \frac{P(\theta)P(D | \theta)}{P(D)}$$

- $P(\theta)$ : prior
- $P(D | \theta)$ : likelihood
- $P(\theta | D)$ : posterior
- $P(D)$ : marginal



Image courtesy of Sebastian Nowozin

Re-use of the image for any other purpose is not allowed

# Variational Inference (VI)

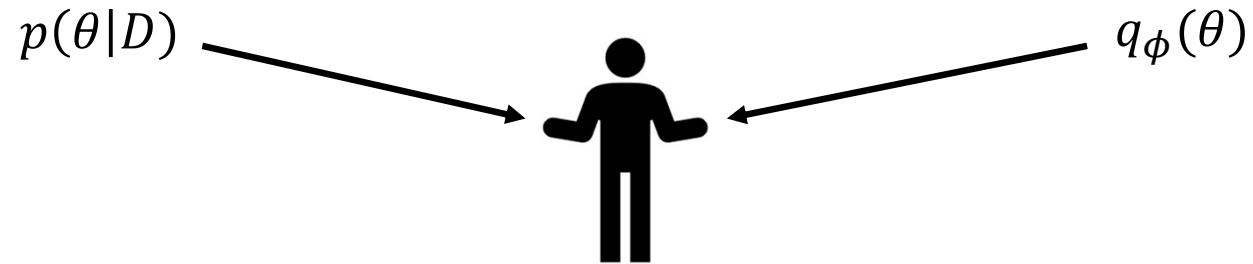
The posterior

$$p(\theta|D) = p(D|\theta)p(\theta)/p(D)$$

The variational distribution

$$q_{\phi}(\theta)$$

# Inference as Optimization



Kullback-Leibler (KL) divergence

# Kullback-Leibler Divergence

$$KL[q(\theta) || p(\theta)] = - \int q(\theta) \log \frac{p(\theta)}{q(\theta)} d\theta = E_{q(\theta)}[\log \frac{p(\theta)}{q(\theta)}]$$

- When  $p = q$ , KL is 0
- Otherwise,  $KL > 0$
- It measures how similar are these two distributions

# Let's Derive the Objective of VI

- Minimize  $KL[q(\theta)||p(\theta|D)]$

$$KL[q(\theta)||p(\theta|D)] = -E_{q(\theta)} \left[ \log \frac{p(\theta|D)}{q(\theta)} \right]$$

# Let's Derive the Objective of VI

- Minimize  $KL[q(\theta)||p(\theta|D)]$

$$KL[q(\theta)||p(\theta|D)] = -E_{q(\theta)} \left[ \log \frac{p(\theta|D)}{q(\theta)} \right]$$

$$= -E_{q(\theta)} \left[ \log \frac{p(\theta,D)}{p(D)q(\theta)} \right] = -E_{q(\theta)} \left[ \log \frac{p(\theta,D)}{q(\theta)} - \log p(D) \right]$$

# Let's Derive the Objective of VI

- Minimize  $KL[q(\theta)||p(\theta|D)]$

$$KL[q(\theta)||p(\theta|D)] = -E_{q(\theta)} \left[ \log \frac{p(\theta|D)}{q(\theta)} \right]$$

$$= -E_{q(\theta)} \left[ \log \frac{p(\theta,D)}{p(D)q(\theta)} \right] = -E_{q(\theta)} \left[ \log \frac{p(\theta,D)}{q(\theta)} - \log p(D) \right]$$

$$= \boxed{\log p(D)} - E_{q(\theta)} \left[ \log \frac{p(\theta,D)}{q(\theta)} \right]$$

Model Evidence

# Let's Derive the Objective of VI

Minimize  $KL[q(\theta)||p(\theta|D)]$

$$KL[q(\theta)||p(\theta|D)] = \log p(D) - E_{q(\theta)} \left[ \log \frac{p(\theta, D)}{q(\theta)} \right]$$

Maximize  $E_{q(\theta)} \left[ \log \frac{p(\theta, D)}{q(\theta)} \right]$

# Let's Derive the Objective of VI

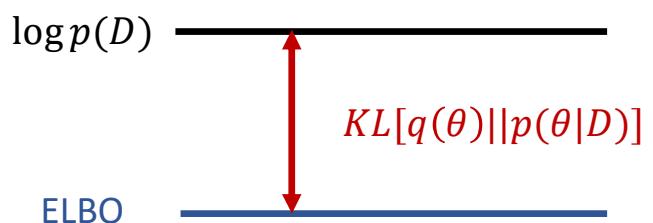
Minimize  $KL[q(\theta)||p(\theta|D)]$

$$KL[q(\theta)||p(\theta|D)] = \boxed{\log p(D)} - E_{q(\theta)} \left[ \log \frac{p(\theta, D)}{q(\theta)} \right]$$

Model Evidence

Maximize  $L = E_{q(\theta)} \left[ \log \frac{p(\theta, D)}{q(\theta)} \right]$

Evidence Lower Bound (ELBO)



"Model Evidence = ELBO + KL"

# Alternative Derivation

Let's start with the model evidence

$$\log p(D)$$

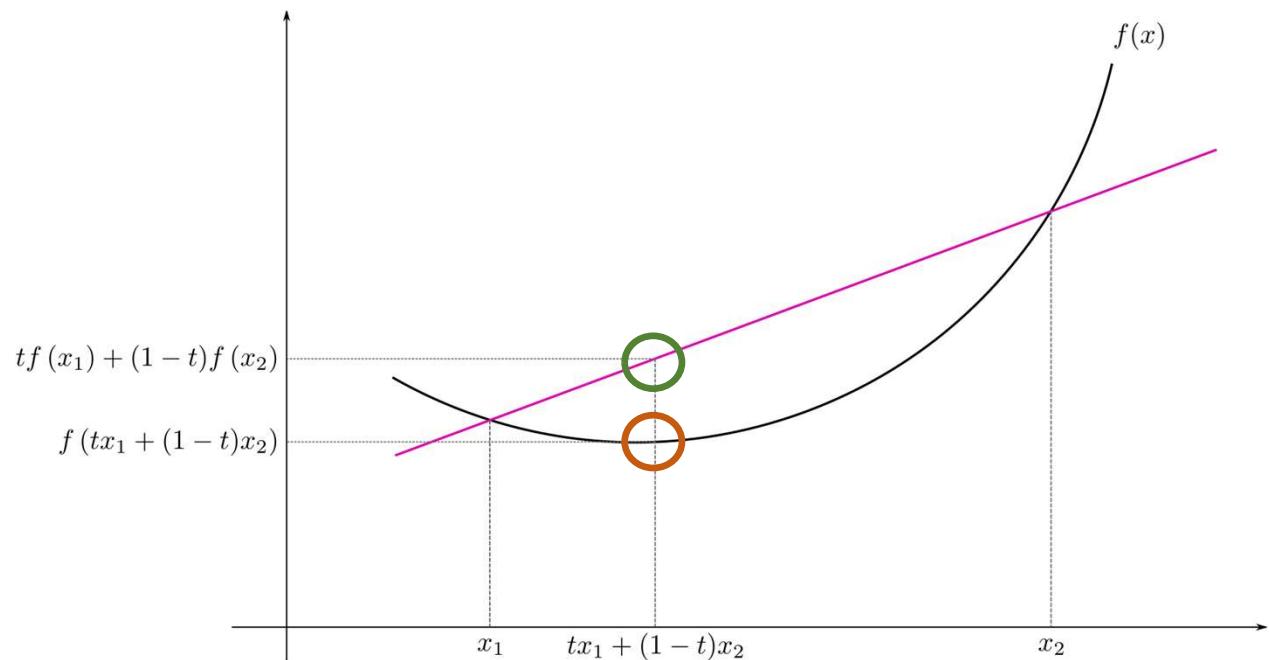
# Alternative Derivation

$$\log p(D) = \log \int p(\theta, D) \ d\theta$$

# Alternative Derivation

$$\begin{aligned}\log p(D) &= \log \int p(\theta, D) d\theta \\&= \log \int \frac{p(\theta, D) q(\theta)}{q(\theta)} d\theta \\&= \log E_{q(\theta)} \left[ \frac{p(\theta, D)}{q(\theta)} \right]\end{aligned}$$

# Jensen's Inequality



If  $f$  is a convex function, then  
 $f(E[X]) \leq E[ f(X) ]$

If  $f$  is a concave function, then  
 $f(E[X]) \geq E[ f(X) ]$

# Alternative Derivation

$$\begin{aligned}\log p(D) &= \log \int p(\theta, D) d\theta \\&= \log \int \frac{p(\theta, D) q(\theta)}{q(\theta)} d\theta \\&= \log E_{q(\theta)} \left[ \frac{p(\theta, D)}{q(\theta)} \right] \\&\stackrel{\text{Jensen's inequality}}{\geq} E_{q(\theta)} \left[ \log \frac{p(\theta, D)}{q(\theta)} \right]\end{aligned}$$

Log is a concave function, then  
 $f(E[X]) \geq E[f(X)]$

# Alternative Derivation

Model Evidence

$$\begin{aligned}\log p(D) &= \log \int p(\theta, D) d\theta \\&= \log \int \frac{p(\theta, D) q(\theta)}{q(\theta)} d\theta \\&= \log E_{q(\theta)} \left[ \frac{p(\theta, D)}{q(\theta)} \right] \\&\geq E_{q(\theta)} \left[ \log \frac{p(\theta, D)}{q(\theta)} \right]\end{aligned}$$

Evidence Lower Bound (ELBO)

# Variational Inference (VI)

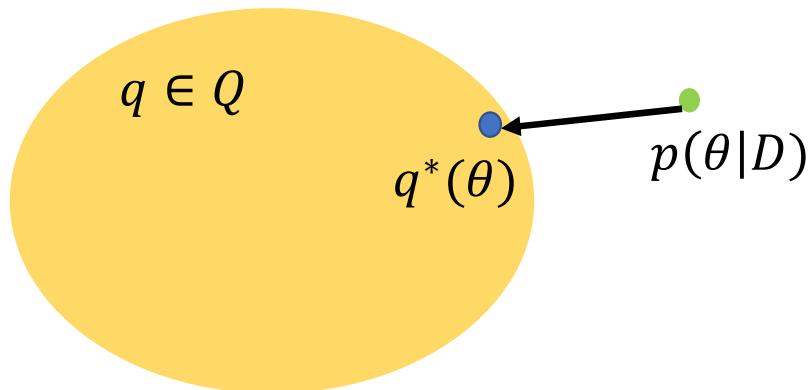
The posterior

$$p(\theta|D) = p(D|\theta)p(\theta)/p(D)$$

The variational distribution

$$q_\phi(\theta)$$

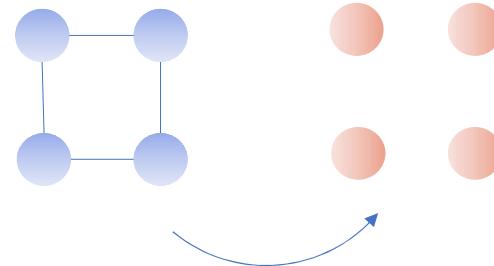
$$L = E_{q_\phi(\theta)} \left[ \log \frac{p(D, \theta)}{q_\phi(\theta)} \right] = \log p(D) - KL[q_\phi(\theta) || p(\theta)]$$



# Mean-field Variational Inference

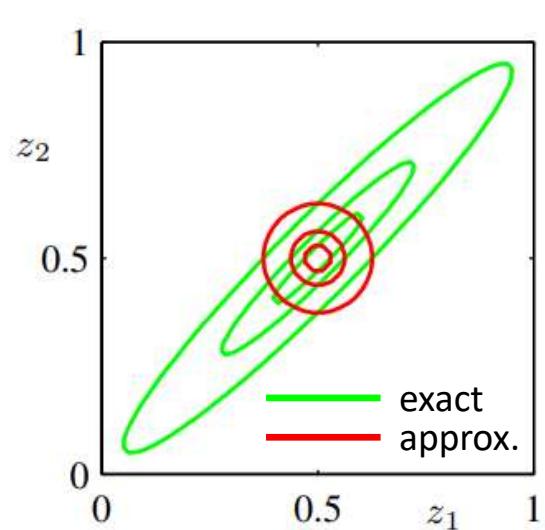
- A type of choices of the variational distribution
- The name originates in the mean field theory of physics
- The variational distribution factorizes

$$q_{\phi}(\boldsymbol{\theta}) = \prod_{i=1}^K q_{\phi_i}(\theta_i)$$



Opper and Saad, eds. Advanced mean field methods: Theory and practice. MIT press, 2001.

# A Gaussian Example



$$p(\mathbf{z}) = N(\mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1})$$

$$q(\mathbf{z}) = q(z_1)q(z_2)$$

# Mean-field Variational Inference

$$\text{ELBO} \quad L = E_{q(\theta)} \left[ \log \frac{p(\theta, D)}{q(\theta)} \right] \quad \xleftarrow{\hspace{1cm}} \quad \text{Fully Factorized Variational Distribution} \quad q(\boldsymbol{\theta}) = \prod_{i=1}^K q_{\phi_i}(\theta_i)$$

Jordan et al. An introduction to variational methods for graphical models. Machine learning 37.2 (1999): 183-233.

# Mean-field Variational Inference

ELBO                                  Fully Factorized Variational Distribution

$$L = E_{q(\theta)} \left[ \log \frac{p(\theta, D)}{q(\theta)} \right] \quad \longleftarrow \quad q(\boldsymbol{\theta}) = \prod_{i=1}^K q_{\phi_i}(\theta_i)$$
$$L = \int q(\theta_j) E_{q(\theta_{\neg j})} [\log p(\theta_j, D | \theta_{\neg j})] d\theta_j - \int q(\theta_j) \log p(\theta_j) d\theta_j + c_j$$

Jordan et al. An introduction to variational methods for graphical models. Machine learning 37.2 (1999): 183-233.

# Mean-field Variational Inference

ELBO    Fully Factorized Variational Distribution

$$L = E_{q(\theta)} \left[ \log \frac{p(\theta, D)}{q(\theta)} \right] \quad \longleftarrow \quad q(\boldsymbol{\theta}) = \prod_{i=1}^K q_{\phi_i}(\theta_i)$$

$$L = \int q(\theta_j) E_{q(\theta_{\neg j})} [\log p(\theta_j, D | \theta_{\neg j})] d\theta_j - \int q(\theta_j) \log p(\theta_j) d\theta_j + c_j$$

$$q^*(\theta_j) \propto \exp(E_{q(\theta_{\neg j})} [\log p(\theta_j, D | \theta_{\neg j})])$$

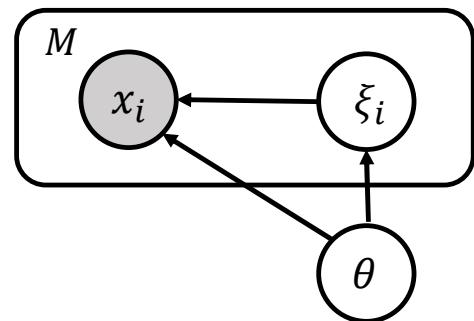
Jordan et al. An introduction to variational methods for graphical models. Machine learning 37.2 (1999): 183-233.



# Part II: Advances

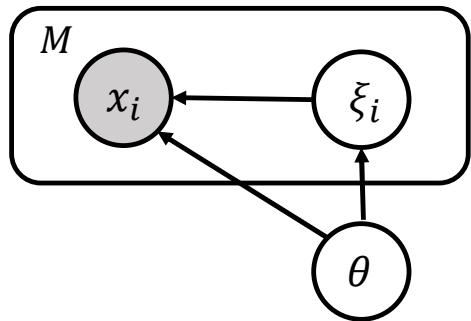
- Scalable variational inference
- Monte Carlo methods
- Amortized inference
- Approximate distribution design
- Optimization objective design

# Stochastic Variational Inference



$$p(\theta, \xi, x) = p(\theta) \prod_{i=1}^M p(\xi_i | \theta) p(x_i | \xi_i, \theta)$$

# Stochastic Variational Inference



$$p(\theta, \xi, x) = p(\theta) \prod_{i=1}^M p(\xi_i | \theta) p(x_i | \xi_i, \theta)$$

$$\begin{aligned} L &= E_q \left[ \log \frac{p(\theta, \xi, x)}{q(\theta, \xi)} \right] \\ &= E_q \left[ \log \frac{p(\theta) \prod_{i=1}^M p(\xi_i | \theta) p(x_i | \xi_i, \theta)}{q(\theta) \prod_{i=1}^M q(\xi_i)} \right] \\ &= E_q \left[ \log \frac{p(\theta)}{q(\theta)} \right] + \sum_{i=1}^M E_q \left[ \log \frac{p(\xi_i | \theta) p(x_i | \xi_i)}{q(\xi_i)} \right] \end{aligned}$$

- O(M) time to compute in each update iteration
- M can be extremely large
- Even one iteration might not be affordable

# Stochastic Variational Inference

$$p(\theta, \xi, x) = p(\theta) \prod_{i=1}^M p(\xi_i | \theta) p(x_i | \xi_i, \theta)$$

$$L = E_q \left[ \log \frac{p(\theta)}{q(\theta)} \right] + \sum_{i=1}^M E_q \left[ \log \frac{p(\xi_i | \theta) p(x_i | \xi_i)}{q(\xi_i)} \right]$$

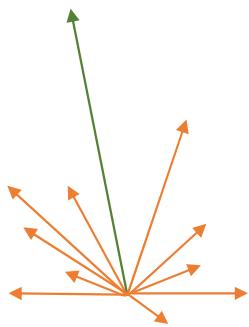
↓

stochastic approximation  
with  $S \ll M$

$$\hat{L} = E_q \left[ \log \frac{p(\theta)}{q(\theta)} \right] + \frac{M}{S} \sum_{i=1}^S E_q \left[ \log \frac{p(\xi_i | \theta) p(x_i | \xi_i)}{q(\xi_i)} \right]$$

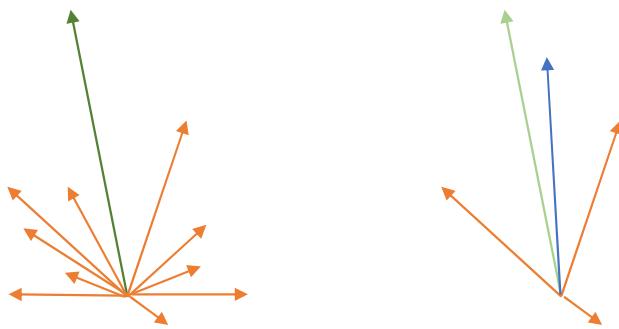
Computational complexity:  $O(M) \rightarrow O(S)$

# How Stochastic Gradient Works



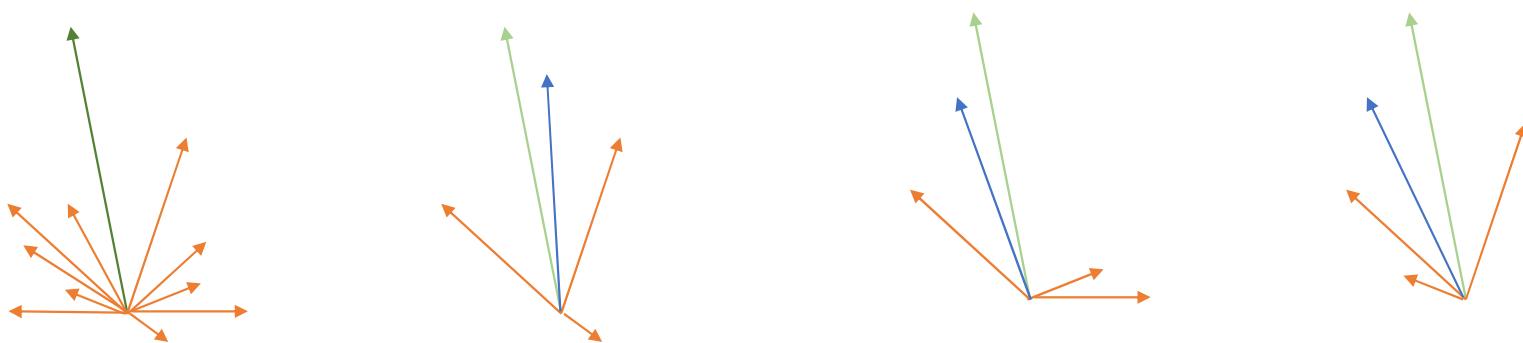
- $\nabla F(x_i)$  gradient of each single data point  $x_i$
- $E_x[\nabla F(x)]$  batch gradient considering all data points

# How Stochastic Gradient Works



- $\nabla F(x_i)$  gradient of each single data point  $x_i$
- $E_x[\nabla F(x)]$  batch gradient considering all  $M = 10$  data points
- $\frac{M}{S} \sum_{s=1}^S \nabla F(x_s)$  mini-batch gradient/stochastic gradient estimated using  $S=3$  data points

# How Stochastic Gradient Works



- $\nabla F(x_i)$  gradient of each single data point  $x_i$
- $E_x[\nabla F(x)]$  batch gradient considering all  $M = 10$  data points
- $\frac{M}{S} \sum_{s=1}^S \nabla F(x_s)$  mini-batch gradient/stochastic gradient estimated using  $S=3$  data points

Zhang. et.al. "Determinantal Point Processes for Mini-Batch Diversification." UAI, 2017

# Stochastic Variational Inference

$$\begin{aligned}
 L &= E_q \left[ \log \frac{p(\theta)}{q(\theta)} \right] + \sum_{i=1}^M E_q \left[ \log \frac{p(\xi_i|\theta)p(x_i|\xi_i)}{q(\xi_i)} \right] \xrightarrow{\text{gradient}} \nabla L = \nabla E_q \left[ \log \frac{p(\theta)}{q(\theta)} \right] + \sum_{i=1}^M E_q \left[ \nabla \log \frac{p(\xi_i|\theta)p(x_i|\xi_i)}{q(\xi_i)} \right] \\
 &\quad \downarrow \text{Stochastic approximation with } S \ll M \\
 \hat{L} &= E_q \left[ \log \frac{p(\theta)}{q(\theta)} \right] + \frac{M}{S} \sum_{i=1}^S E_q \left[ \log \frac{p(\xi_i|\theta)p(x_i|\xi_i)}{q(\xi_i)} \right] \xrightarrow{\text{gradient}} \nabla \hat{L} = \nabla E_q \left[ \log \frac{p(\theta)}{q(\theta)} \right] + \frac{M}{S} \sum_{i=1}^S E_q \left[ \nabla \log \frac{p(\xi_i|\theta)p(x_i|\xi_i)}{q(\xi_i)} \right]
 \end{aligned}$$

Stochastic Gradient

Nature laughs at the  
difficulties of integration.

--Pierre-Simon Laplace

Gordon and Sorkin. *The Armchair Science Reader*. New York 1959



# Monte Carlo Approximation

- To approximate:  $E_{p(x)}[f(x)]$
  - MC Approximation:
    1. Sample  $x_1, x_2, \dots, x_K \sim p(x)$
    2. Evaluate  $f(x_i)$  for each sample
    3. Compute  $E[f(x)] \approx \frac{1}{K} \sum_{i=1}^K f(x_i)$
- Unbiased Monte Carlo estimate

# Log-derivative Trick

$$\nabla_{\theta} \log p_{\theta}(x)$$

# Log-derivative Trick

$$\nabla_{\theta} \log p_{\theta}(x) = \frac{1}{p_{\theta}(x)} \nabla p_{\theta}(X)$$

# Log-derivative Trick

$$\nabla_{\theta} \log p_{\theta}(x) = \frac{1}{p_{\theta}(x)} \nabla p_{\theta}(x)$$

$$\nabla_{\theta} p_{\theta}(x) = p_{\theta}(x) \nabla_{\theta} \log p_{\theta}(x)$$

# REINFORCE Gradients

$$\text{ELBO } L = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right]$$

Gradient of the ELBO

$$\nabla_\phi L = \nabla_\phi E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] = \int \nabla_\phi \{ q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} \} d\theta$$

Glynn (1990). Likelihood ratio gradient estimation for stochastic systems. Communications of the ACM, 33(10), 75–84.

Williams (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8(3-4), 229–256.

Fu (2006). Gradient estimation. Handbooks in Operations Research and Management Science, 13, 575–616.

# REINFORCE Gradients

$$\text{ELBO } L = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right]$$

Gradient of the ELBO

$$\begin{aligned} \nabla_\phi L &= \nabla_\phi E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] = \int \nabla_\phi \{ q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} \} d\theta \\ &= \int \nabla_\phi q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} d\theta + \int q_\phi(\theta) \nabla_\phi \log \frac{p(\theta, D)}{q_\phi(\theta)} d\theta \end{aligned}$$

Glynn (1990). Likelihood ratio gradient estimation for stochastic systems. Communications of the ACM, 33(10), 75–84.

Williams (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8(3-4), 229–256.

Fu (2006). Gradient estimation. Handbooks in Operations Research and Management Science, 13, 575–616.

# REINFORCE Gradients

$$\text{ELBO } L = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right]$$

Gradient of the ELBO

$$\begin{aligned} \nabla_\phi L &= \nabla_\phi E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] = \int \nabla_\phi \{ q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} \} d\theta \\ &= \int \underline{\nabla_\phi q_\phi(\theta)} \log \frac{p(\theta, D)}{q_\phi(\theta)} d\theta + \int q_\phi(\theta) \nabla_\phi \log \frac{p(\theta, D)}{q_\phi(\theta)} d\theta \\ &= \int \underline{q_\phi(\theta) \nabla_\phi \log q_\phi(\theta)} \ \log \frac{p(\theta, D)}{q_\phi(\theta)} d\theta - \int \nabla_\phi q_\phi(\theta) d\theta \end{aligned}$$

Log-derivative trick

$$\nabla q_\phi(\theta) = q_\phi(\theta) \nabla_\phi \log q_\phi(\theta)$$

Glynn (1990). Likelihood ratio gradient estimation for stochastic systems. Communications of the ACM, 33(10), 75–84.

Williams (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8(3-4), 229–256.

Fu (2006). Gradient estimation. Handbooks in Operations Research and Management Science, 13, 575–616.

# REINFORCE Gradients

$$\text{ELBO } L = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right]$$

Gradient of the ELBO

$$\begin{aligned} \nabla_\phi L &= \nabla_\phi E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] = \int \nabla_\phi \{ q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} \} d\theta \\ &= \int \nabla_\phi q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} d\theta + \int q_\phi(\theta) \nabla_\phi \log \frac{p(\theta, D)}{q_\phi(\theta)} d\theta \\ &= \int q_\phi(\theta) \nabla_\phi \log q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} d\theta - \underline{\int \nabla_\phi q_\phi(\theta) d\theta} \\ &\quad = \nabla_\phi \int q_\phi(\theta) d\theta = \nabla_\phi 1 = 0 \end{aligned}$$

$$\begin{aligned} \nabla_\phi \log \frac{p(\theta, D)}{q_\phi(\theta)} \\ &= \frac{q_\phi(\theta)}{p(\theta, D)} \left( -\frac{p(\theta, D)}{q_\phi(\theta)^2} \right) \nabla q_\phi(\theta) \\ &= -\frac{\nabla_\phi q_\phi(\theta)}{q_\phi(\theta)} \end{aligned}$$

Glynn (1990). Likelihood ratio gradient estimation for stochastic systems. Communications of the ACM, 33(10), 75–84.

Williams (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8(3-4), 229–256.

Fu (2006). Gradient estimation. Handbooks in Operations Research and Management Science, 13, 575–616.

# REINFORCE Gradients

$$\text{ELBO } L = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right]$$

Gradient of the ELBO

$$\begin{aligned} \nabla_\phi L &= \nabla_\phi E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] = \int \nabla_\phi \{ q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} \} d\theta \\ &= \int \nabla_\phi q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} d\theta + \int q_\phi(\theta) \nabla_\phi \log \frac{p(\theta, D)}{q_\phi(\theta)} d\theta \\ &= \int q_\phi(\theta) \nabla_\phi \log q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} d\theta - \underline{\int \nabla_\phi q_\phi(\theta) d\theta} \\ &= E_{q_\phi(\theta)} \left[ \underbrace{\nabla_\phi \log q_\phi(\theta)}_{\text{Score function}} \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] \quad = \nabla_\phi \int q_\phi(\theta) d\theta = \nabla_\phi 1 = 0 \end{aligned}$$

Glynn (1990). Likelihood ratio gradient estimation for stochastic systems. Communications of the ACM, 33(10), 75–84.

Williams (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8(3-4), 229–256.

Fu (2006). Gradient estimation. Handbooks in Operations Research and Management Science, 13, 575–616.

# BBVI

$$\text{ELBO } L = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right]$$

- To approximate:  $E[f(x)]$
- MC Approximation:

1. Sample  $x_1, x_2, \dots, x_K \sim p(x)$
2. Evaluate  $f(x_i)$  for each sample
3. Compute  $E[f(x)] \approx \frac{1}{K} \sum_{i=1}^K f(x_i)$



Gradient of the ELBO

$$\nabla_\phi L = E_{q_\phi(\theta)} \left[ \nabla_\phi \log q_\phi(\theta) \log \frac{p(\theta, D)}{q_\phi(\theta)} \right]$$

Score function

Ranganath et al. Black box variational inference. AISTATS 2014

Glynn (1990). Likelihood ratio gradient estimation for stochastic systems. Communications of the ACM, 33(10), 75–84.

Williams (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8(3-4), 229–256.

Fu (2006). Gradient estimation. Handbooks in Operations Research and Management Science, 13, 575–616.

# BBVI

Gradient of the ELBO

$$\nabla_{\phi} L = E_{q_{\phi}(\theta)} \left[ \nabla_{\phi} \log q_{\phi}(\theta) \log \frac{p(\theta, D)}{q_{\phi}(\theta)} \right]$$

- To approximate:  $E[f(x)]$
- MC Approximation:

1. Sample  $x_1, x_2, \dots, x_K \sim p(x)$
2. Evaluate  $f(x_i)$  for each sample
3. Compute  $E[f(x)] \approx \frac{1}{K} \sum_{i=1}^K f(x_i)$



1. Sample  $\theta_1, \theta_2, \dots, \theta_K \sim q_{\phi}(\theta)$

# BBVI

Gradient of the ELBO

$$\nabla_{\phi} L = E_{q_{\phi}(\theta)} \left[ \nabla_{\phi} \log q_{\phi}(\theta) \log \frac{p(\theta, D)}{q_{\phi}(\theta)} \right]$$

- To approximate:  $E[f(x)]$
- MC Approximation:

1. Sample  $x_1, x_2, \dots, x_K \sim p(x)$
2. Evaluate  $f(x_i)$  for each sample
3. Compute  $E[f(x)] \approx \frac{1}{K} \sum_{i=1}^K f(x_i)$



1. Sample  $\theta_1, \theta_2, \dots, \theta_K \sim q_{\phi}(\theta)$
2. Evaluate  $\nabla_{\phi} \log q_{\phi}(\theta) \log \frac{p(\theta, D)}{q_{\phi}(\theta)}$  for each sample

# BBVI

Gradient of the ELBO

$$\nabla_{\phi} L = E_{q_{\phi}(\theta)} \left[ \nabla_{\phi} \log q_{\phi}(\theta) \log \frac{p(\theta, D)}{q_{\phi}(\theta)} \right]$$

- To approximate:  $E[f(x)]$
- MC Approximation:

1. Sample  $x_1, x_2, \dots, x_K \sim p(x)$
2. Evaluate  $f(x_i)$  for each sample
3. Compute  $E[f(x)] \approx \frac{1}{K} \sum_{i=1}^K f(x_i)$



1. Sample  $\theta_1, \theta_2, \dots, \theta_K \sim q_{\phi}(\theta)$
2. Evaluate  $\nabla_{\phi} \log q_{\phi}(\theta) \log \frac{p(\theta, D)}{q_{\phi}(\theta)}$  for each sample
3. The approximated gradient is:

$$\nabla_{\phi} \hat{L} = \frac{1}{K} \sum_{i=1}^K \nabla_{\phi} \log q_{\phi}(\theta_i) \log \frac{p(\theta_i, D)}{q_{\phi}(\theta_i)}$$

# Black-box Variational Inference (BBVI)



## A.2. Update of $\rho$

The partial derivative is given below:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \rho_{ji}} &= \sum_{m=1}^K \zeta_{im} \mathbb{E}_{\pi_\theta} [\nabla_{\mu_m} p(\pi_\theta(a_i)) + \nabla_{\mu_m} b_i] - 1 - \log \rho_{ji} \\ &\quad + \alpha_{j,i} (\frac{1}{N} \sum_{m=1}^K \zeta_{im}) \\ &\quad - \delta^{-1} \left( \sum_{m=1}^K \left( \left( \prod_{l=1}^J \sum_{k=1}^K \sum_{l'=1}^{J'} \rho_{jl} \zeta_{im} \exp(\frac{1}{N} \mu_{jl}) \right) \right. \right. \\ &\quad \left. \left. - \sum_{l=1}^J \left( \sum_{k=1}^K \sum_{l'=1}^{J'} \rho_{jl} \zeta_{im} \exp(\frac{1}{N} \mu_{jl}) \right) \right) \right). \end{aligned} \quad (18)$$

the inequality  $\log(x) \leq x^{-1} x + \log(1) - 1$ , where equality holds if and only if  $x = 1$ . Thus, set  $x = \delta' \zeta_{im}$ , and  $1 = \delta' \zeta_{im}^{-1}$ . The new bound becomes:

$$\begin{aligned} \mathcal{L}' &\geq \sum_{m=1}^K \sum_{l=1}^J \sum_{k=1}^K \left( \zeta_{im} \left( \sum_{l'=1}^{J'} \rho_{jl} \zeta_{im}^{-1} (\Psi(\lambda_{kl}) - \Psi(\sum \lambda_{kl})) \right) \right) \rho_{ji} - \delta \right) \\ &\quad + \zeta_{im} \left[ (\Psi(a_{j,i}) - \Psi(a_{j,i} + \alpha_{j,i})) \right. \\ &\quad \left. + \sum_{l=1}^J \left( \Psi(a_{j,l}) - \Psi(a_{j,l} + \alpha_{j,l}) \right) \right] - \zeta_{im} \log \zeta_{im} \\ &\quad + \sum_{j=1}^J \left( \mu_j' / \frac{1}{N} \sum_{m=1}^K \rho_{jm} \zeta_{im} \right) - (\delta' \zeta_{im}^{-1})' \delta' \zeta_{im} \\ &\quad - \log(\delta' \zeta_{im}^{-1}) + 1 \Big) = \mathcal{L}''. \end{aligned} \quad (21)$$

## A.3. Update of $\zeta$

$$\begin{aligned} \mathcal{L}' &= \sum_{j=1}^J \sum_{l=1}^J \sum_{k=1}^K \left( \zeta_{im} \left( \sum_{l'=1}^{J'} \rho_{jl} \zeta_{im}^{-1} (\Psi(\lambda_{kl}) - \Psi(\sum \lambda_{kl})) \right) \right) \rho_{ji} - \delta \\ &\quad - \zeta_{im} \left( \Psi(a_{j,i}) - \Psi(a_{j,i} + \alpha_{j,i}) \right) + \sum_{l=1}^J \left( \Psi(a_{j,l}) - \Psi(a_{j,l} + \alpha_{j,l}) \right) \\ &\quad - \zeta_{im} \log \zeta_{im} + \sum_{j=1}^J \left( \sum_{l=1}^J \sum_{k=1}^K \sum_{l'=1}^{J'} \rho_{jl} \zeta_{im} \exp(\frac{1}{N} \mu_{jl}) \right) \\ &\quad - \log \left( \sum_{j=1}^J \left( \sum_{l=1}^J \sum_{k=1}^K \sum_{l'=1}^{J'} \rho_{jl} \zeta_{im} \exp(\frac{1}{N} \mu_{jl}) \right) \right). \end{aligned} \quad (19)$$

where  $\Psi$  is the digamma function. For  $i \in \{1, \dots, T\}$ , we write:

$$b_i = \sum_{j=1}^J \left( \prod_{l=1}^J \left( \sum_{k=1}^K \sum_{l'=1}^{J'} \rho_{jl} \zeta_{im} \exp(\frac{1}{N} \mu_{jl}) \right) \right) \left( \sum_{l=1}^J \rho_{jl} \exp(\frac{1}{N} \mu_{jl}) \right).$$

which yields:

$$\sum_{j=1}^J \prod_{l=1}^J \sum_{k=1}^K \rho_{jl} \zeta_{im} \exp(\frac{1}{N} \mu_{jl}) = \sum_{j=1}^J b_i \cdot \zeta_{im}.$$

$\mathcal{L}'_{\zeta_{im}}$  can now be rewritten as:

$$\begin{aligned} \mathcal{L}'_{\zeta_{im}} &= \sum_{j=1}^J \left( \zeta_{im} \left( \sum_{l=1}^J \rho_{jl} \zeta_{im}^{-1} (\Psi(\lambda_{jl}) - \Psi(\sum \lambda_{jl})) \right) \right) \rho_{ji} - \delta \\ &\quad + \zeta_{im} \left( \Psi(a_{j,i}) - \Psi(a_{j,i} + \alpha_{j,i}) \right) + \sum_{l=1}^J \left( \Psi(a_{j,l}) - \Psi(a_{j,l} + \alpha_{j,l}) \right) \\ &\quad - \zeta_{im} \log \zeta_{im} + \left( \mu_j' / \frac{1}{N} \sum_{m=1}^K \rho_{jm} \zeta_{im} \right) - \log(\delta' \zeta_{im}). \end{aligned} \quad (20)$$

We follow the approach of [14] to derive the fixed point update. Suppose we have a previous value  $\zeta_{im}^{(t)}$ . Consider

the inequality  $\log(x) \leq x^{-1} x + \log(1) - 1$ , where equality holds if and only if  $x = 1$ . Thus, set  $x = \delta' \zeta_{im}^{(t)}$ , and  $1 = \delta' \zeta_{im}^{(t)-1}$ . The new bound becomes:

$$\begin{aligned} \mathcal{L}''_{\zeta_{im}} &= \sum_{j=1}^J \sum_{l=1}^J \left( \zeta_{im}^{(t)} \left( \sum_{l'=1}^{J'} \rho_{jl} \zeta_{im}^{(t)-1} (\Psi(\lambda_{jl}) - \Psi(\sum \lambda_{jl})) \right) \right) \rho_{ji} - \delta \\ &\quad + \zeta_{im}^{(t)} \left[ (\Psi(a_{j,i}) - \Psi(a_{j,i} + \alpha_{j,i})) \right. \\ &\quad \left. + \sum_{l=1}^J \left( \Psi(a_{j,l}) - \Psi(a_{j,l} + \alpha_{j,l}) \right) \right] - \zeta_{im}^{(t)} \log \zeta_{im} \\ &\quad + \sum_{j=1}^J \left( \mu_j' / \frac{1}{N} \sum_{m=1}^K \rho_{jm} \zeta_{im}^{(t)} \right) - (\delta' \zeta_{im}^{(t)-1})' \delta' \zeta_{im}^{(t)} \\ &\quad - \log(\delta' \zeta_{im}^{(t)-1}) + 1 \Big) = \mathcal{L}'''_{\zeta_{im}}. \end{aligned} \quad (21)$$

## Go beyond conjugate exponential family

We compute the derivative for the new bound:

$$\begin{aligned} \frac{\partial \mathcal{L}'''_{\zeta_{im}}}{\partial \zeta_{im}} &= \sum_{j=1}^J \left( \Psi(\lambda_{jl}) - \Psi(\sum \lambda_{jl}) \right) \rho_{ji} - \delta \\ &\quad + (\Psi(a_{j,i}) - \Psi(a_{j,i} + \alpha_{j,i})) \sum_{l=1}^J \left( \Psi(a_{j,l}) - \Psi(a_{j,l} + \alpha_{j,l}) \right) \\ &\quad - 1 - \log \zeta_{im} - \frac{1}{N} \mu_j' \rho_{ji} - (\delta' \zeta_{im}^{(t)-1})' \delta' \zeta_{im}^{(t)}. \end{aligned}$$

Finally, we set the derivative to zero to get the fixed point update:

$$\begin{aligned} \zeta_{im} &= \exp \left( \sum_{j=1}^J \rho_{ji} \sum_{l=1}^J \left( \Psi(\lambda_{jl}) - \Psi(\sum \lambda_{jl}) \right) \rho_{ji} - \delta \right) \\ &\quad + (\Psi(a_{j,i}) - \Psi(a_{j,i} + \alpha_{j,i})) \sum_{l=1}^J \left( \Psi(a_{j,l}) - \Psi(a_{j,l} + \alpha_{j,l}) \right) \\ &\quad - 1 - \frac{1}{N} \mu_j' \rho_{ji} - (\delta' \zeta_{im}^{(t)-1})' \delta' \zeta_{im}^{(t)} \\ &= \exp \left( \sum_{j=1}^J \rho_{ji} \frac{1}{N} \log \rho_{jm} \zeta_{im} \right) + \frac{1}{N} \log \pi_{ji} \\ &\quad + \frac{1}{N} \mu_j' \rho_{ji} - (\delta' \zeta_{im}^{(t)-1})' \delta' \zeta_{im}^{(t)}. \end{aligned} \quad (22)$$

## A.4. Update of $\mu$

Let  $\delta = \sum_{j=1}^J \left( \prod_{l=1}^J \sum_{k=1}^K \rho_{jl} \zeta_{im} \exp(\frac{1}{N} \mu_{jl}) \right)$ .

$$\begin{aligned} \frac{\partial \mathcal{L}'''_{\zeta_{im}}}{\partial \mu_{jl}} &= \frac{N}{J} \left( \delta j - \frac{1}{N} \sum_{m=1}^K \sum_{l=1}^J \rho_{jm} \zeta_{im} \right) \\ &\quad - \delta^{-1} \prod_{l=1}^J \sum_{k=1}^K \sum_{l'=1}^{J'} \rho_{jl} \zeta_{im} \exp(\frac{1}{N} \mu_{jl}) \\ &\quad + \frac{N}{J} \left( \frac{1}{N} \mu_j' \rho_{jm} \zeta_{im} - \frac{1}{N} \log(\delta' \zeta_{im}^{(t)}) \right) \end{aligned} \quad (24)$$

<sup>14</sup>To incorporate the constraint that  $\sum_{j=1}^J \zeta_{im} = 1$ ,  $\propto$  is used here instead of  $\equiv$ , since the normalizing factor is dropped in the above result.

# Reparameterization Trick

Express  $q_\phi(\theta)$  as  $\epsilon \sim r(\epsilon)$ ,  $\theta = g(\epsilon, \phi)$

Kingma and Welling. Auto-encoding variational bayes. ICLR 2014

Salimans and Knowles. Fixed-Form Variational Posterior Approximation through Stochastic Linear Regression. Bayesian Analysis 2013

# Reparameterization Trick

Express  $q_\phi(\theta)$  as  $\epsilon \sim r(\epsilon)$ ,  $\theta = g(\epsilon, \phi)$

$$\begin{aligned}\theta &\sim N(\mu, \sigma^2) \\ \epsilon &\sim N(0, 1), \theta = \mu + \sigma\epsilon\end{aligned}$$

Kingma and Welling. Auto-encoding variational bayes. ICLR 2014

Salimans and Knowles. Fixed-Form Variational Posterior Approximation through Stochastic Linear Regression. Bayesian Analysis 2013

# Reparameterization Trick

Express  $q_\phi(\theta)$  as  $\epsilon \sim r(\epsilon)$ ,  $\theta = g(\epsilon, \phi)$

$$\begin{aligned}\theta &\sim N(\mu, \sigma^2) \\ \epsilon &\sim N(0, 1), \theta = \mu + \sigma\epsilon\end{aligned}$$

$$\text{ELBO } L = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] = E_{r(\epsilon)} \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right]$$

Kingma and Welling. Auto-encoding variational bayes. ICLR 2014

Salimans and Knowles. Fixed-Form Variational Posterior Approximation through Stochastic Linear Regression. Bayesian Analysis 2013

# Reparameterization Trick

Express  $q_\phi(\theta)$  as  $\epsilon \sim r(\epsilon)$ ,  $\theta = g(\epsilon, \phi)$

$$\begin{aligned}\theta &\sim N(\mu, \sigma^2) \\ \epsilon &\sim N(0, 1), \theta = \mu + \sigma\epsilon\end{aligned}$$

ELBO  $L = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] = E_{r(\epsilon)} \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right]$

Gradient  $\nabla_\phi L = \nabla_\phi E_{r(\epsilon)} \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right]$

Kingma and Welling. Auto-encoding variational bayes. ICLR 2014

Salimans and Knowles. Fixed-Form Variational Posterior Approximation through Stochastic Linear Regression. Bayesian Analysis 2013

# Reparameterization Trick

Express  $q_\phi(\theta)$  as  $\epsilon \sim r(\epsilon)$ ,  $\theta = g(\epsilon, \phi)$

$$\begin{aligned}\theta &\sim N(\mu, \sigma^2) \\ \epsilon &\sim N(0, 1), \theta = \mu + \sigma\epsilon\end{aligned}$$

ELBO  $L = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] = E_{r(\epsilon)} \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right]$

Gradient  $\nabla_\phi L = \nabla_\phi E_{r(\epsilon)} \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right] = E_{r(\epsilon)} \left[ \nabla_\phi \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right]$

Kingma and Welling. Auto-encoding variational bayes. ICLR 2014

Salimans and Knowles. Fixed-Form Variational Posterior Approximation through Stochastic Linear Regression. Bayesian Analysis 2013

# Reparameterization Trick

Express  $q_\phi(\theta)$  as  $\epsilon \sim r(\epsilon)$ ,  $\theta = g(\epsilon, \phi)$

$$\begin{aligned}\theta &\sim N(\mu, \sigma^2) \\ \epsilon &\sim N(0, 1), \theta = \mu + \sigma\epsilon\end{aligned}$$

$$\text{ELBO } L = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] = E_{r(\epsilon)} \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right]$$

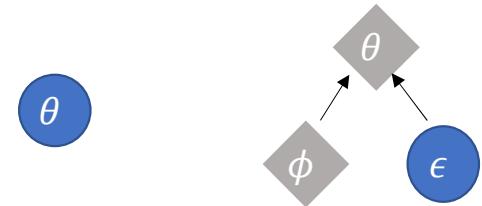
$$\text{Gradient } \nabla_\phi L = \nabla_\phi E_{r(\epsilon)} \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right] = E_{r(\epsilon)} \left[ \nabla_\phi \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right]$$

$$\nabla_\phi \hat{L} = \frac{1}{K} \sum_{k=1}^K \nabla_\phi \log \frac{p(g(\epsilon_k, \phi), D)}{q_\phi(g(\epsilon_k, \phi))}, \epsilon_k \sim r(\epsilon)$$



# Reparameterization Trick

Express  $q_\phi(\theta)$  as  $\epsilon \sim r(\epsilon)$ ,  $\theta = g(\epsilon, \phi)$



$$\theta \sim N(\mu, \sigma^2)$$

$$\epsilon \sim N(0, 1), \theta = \mu + \sigma\epsilon$$

$$\text{ELBO } L = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] = E_{r(\epsilon)} \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right]$$

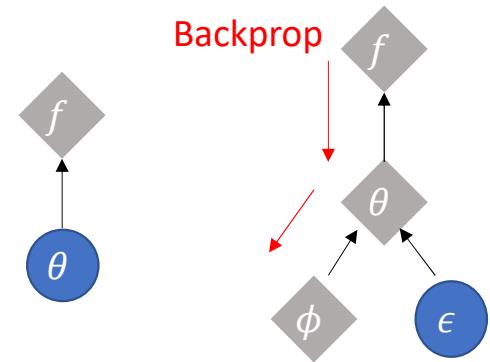
$$\text{Gradient } \nabla_\phi L = \nabla_\phi E_{r(\epsilon)} \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right] = E_{r(\epsilon)} \left[ \nabla_\phi \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right]$$

$$\nabla_\phi \hat{L} = \frac{1}{K} \sum_{k=1}^K \nabla_\phi \log \frac{p(g(\epsilon_k, \phi), D)}{q_\phi(g(\epsilon_k, \phi))}, \epsilon_k \sim r(\epsilon)$$



# Reparameterization Trick

Express  $q_\phi(\theta)$  as  $\epsilon \sim r(\epsilon)$ ,  $\theta = g(\epsilon, \phi)$



$$\theta \sim N(\mu, \sigma^2)$$

$$\epsilon \sim N(0, 1), \theta = \mu + \sigma\epsilon$$

$$\text{ELBO } L = E_{q_\phi(\theta)} \left[ \log \frac{p(\theta, D)}{q_\phi(\theta)} \right] = E_{r(\epsilon)} \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right]$$

$$\text{Gradient } \nabla_\phi L = \nabla_\phi E_{r(\epsilon)} \left[ \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right] = E_{r(\epsilon)} \left[ \nabla_\phi \log \frac{p(g(\epsilon, \phi), D)}{q_\phi(g(\epsilon, \phi))} \right]$$

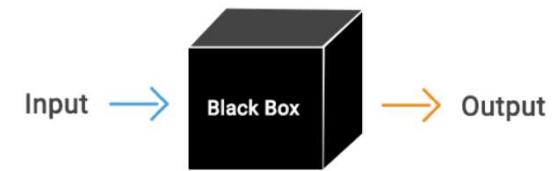
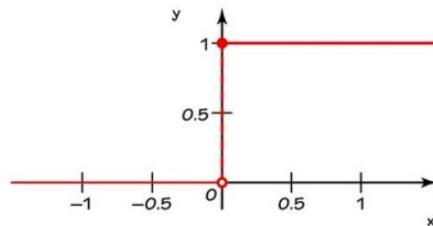
$$\nabla_\phi \hat{L} = \frac{1}{K} \sum_{k=1}^K \nabla_\phi \log \frac{p(g(\epsilon_k, \phi), D)}{q_\phi(g(\epsilon_k, \phi))}, \epsilon_k \sim r(\epsilon)$$



# Variance Reduction Techniques in MCVI

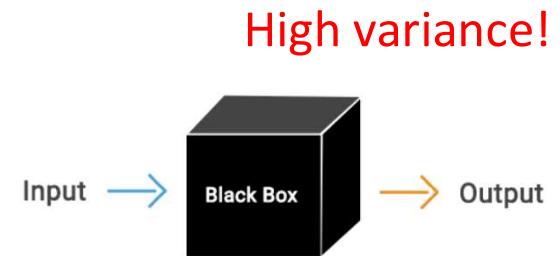
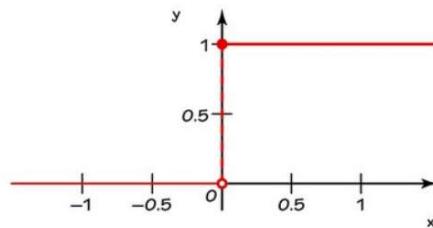
- When non-differentiable, falls back to REINFORCE gradient

High variance!



# Variance Reduction Techniques in MCVI

- When non-differentiable, falls back to REINFORCE gradient

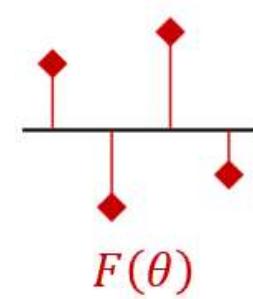


- Solutions to high variance REINFORCE gradients:
  - Low variance unbiased estimators with control variates
  - Biased estimators to enable reparam. trick (potentially low variance)

# Variance Reduction Techniques in MCVI

- Control variate method:
  - Assume we want to estimate with MC simulation

$$E_{q(\theta)}[F(\theta)] \approx \frac{1}{K} \sum_k^K F(\theta_k), \quad \theta_k \sim q(\theta)$$



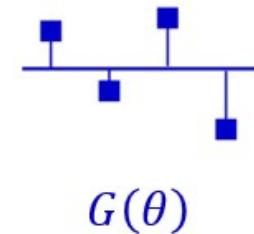
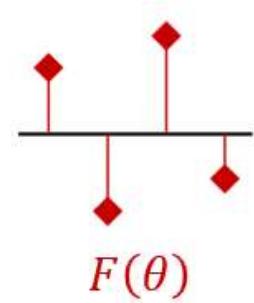
# Variance Reduction Techniques in MCVI

- Control variate method:

- Assume we want to estimate with MC simulation

$$E_{q(\theta)}[F(\theta)] \approx \frac{1}{K} \sum_k^K F(\theta_k), \quad \theta_k \sim q(\theta)$$

- Control variate: define a control function  $G(\theta)$  satisfying:
    - $V_{q(\theta)}[G(\theta)] < \infty$
    - Known or fast computable  $E_{q(\theta)}[G(\theta)]$



# Variance Reduction Techniques in MCVI

- Control variate method:
  - Then define the new MC estimator

$$E_{q(\theta)}[F(\theta)] \approx \frac{1}{K} \sum_k^K \hat{F}(\theta_k), \quad \theta_k \sim q(\theta),$$

$$\hat{F}(\theta) = F(\theta) - G(\theta) + E_{q(\theta)}[G(\theta)]$$

$$V_{q(\theta)}[\hat{F}(\theta)] = V_{q(\theta)}[F(\theta)] + \underline{V_{q(\theta)}[G(\theta)] - 2 \text{Cov}_{q(\theta)}[F(\theta), G(\theta)]}$$

$< 0$  if  $F$  and  $G$  are strongly and positively correlated

# Variance Reduction Techniques in MCVI

- Application to REINFORCE gradient:

$$\bullet \quad F(\theta) = \log \frac{p(D, \theta)}{q_\phi(\theta)} \nabla_\phi \log q_\phi(\theta)$$

$\underbrace{\phantom{\log \frac{p(D, \theta)}{q_\phi(\theta)} \nabla_\phi \log q_\phi(\theta)}}_{:= f(\theta)}$

- Define  $G(\theta) = g(\theta) \nabla_\phi \log q_\phi(\theta)$

# Variance Reduction Techniques in MCVI

- Application to REINFORCE gradient:

$$\begin{aligned} \bullet \quad F(\theta) &= \log \frac{p(D, \theta)}{q_\phi(\theta)} \nabla_\phi \log q_\phi(\theta) & \bullet \quad \text{Define } G(\theta) &= g(\theta) \nabla_\phi \log q_\phi(\theta) \\ &\stackrel{\text{:= } f(\theta)}{\underline{}} & & \\ \bullet \quad \text{Variance reduced gradient: } \hat{F}(\theta) &= \frac{(f(\theta) - g(\theta)) \nabla_\phi \log q_\phi(\theta) + E_{q_\phi(\theta)}[G(\theta)]}{\stackrel{\text{:= } \Delta(\theta)}{\underline{}}} \end{aligned}$$

# Variance Reduction Techniques in MCVI

- Application to REINFORCE gradient:

- $F(\theta) = \log \frac{p(D, \theta)}{q_\phi(\theta)} \nabla_\phi \log q_\phi(\theta)$   
 $\underline{\hspace{10em}} := f(\theta)$
- Define  $G(\theta) = g(\theta) \nabla_\phi \log q_\phi(\theta)$
- Variance reduced gradient:  $\hat{F}(\theta) = \underline{(f(\theta) - g(\theta)) \nabla_\phi \log q_\phi(\theta)} + E_{q_\phi(\theta)}[G(\theta)]$   
 $\underline{\hspace{10em}} := \Delta(\theta)$

- “Baseline” approach:

$$g(\theta) = b$$

$$\Rightarrow E_{q(\theta)}[G(\theta)] = b E_{q(\theta)}[\nabla_\phi \log q(\theta)] = b \nabla_\phi \int q_\phi(\theta) d\theta = b \nabla_\phi 1 = 0$$

(log-derivative trick)

$$\Rightarrow \hat{F}(\theta) = \Delta(\theta) \nabla_\phi \log q_\phi(\theta)$$

Ranganath et al. Black Box Variational Inference. AISTATS 2014

Mnih and Gregor. Neural Variational Inference and Learning in Belief Networks. ICML 2014

# Variance Reduction Techniques in MCVI

- Application to REINFORCE gradient:

- $F(\theta) = \log \frac{p(D, \theta)}{q_\phi(\theta)} \nabla_\phi \log q_\phi(\theta)$
- Define  $G(\theta) = g(\theta) \nabla_\phi \log q_\phi(\theta)$
- $\underline{\hspace{10em}} := f(\theta)$
- Variance reduced gradient:  $\hat{F}(\theta) = \underline{(f(\theta) - g(\theta)) \nabla_\phi \log q_\phi(\theta)} + E_{q_\phi(\theta)}[G(\theta)]$
- $\underline{\hspace{10em}} := \Delta(\theta)$

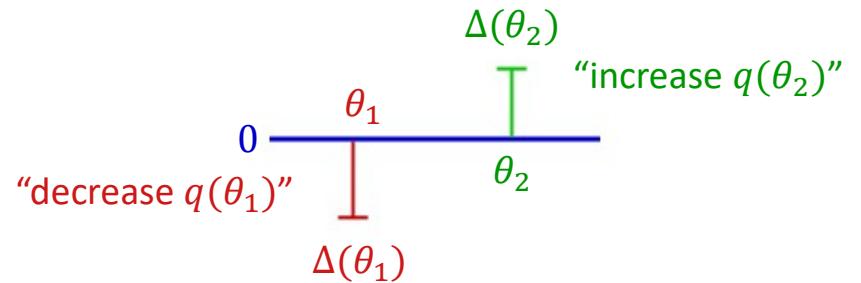
- “Baseline” approach:

$$g(\theta) = b$$

$$\Rightarrow E_{q(\theta)}[G(\theta)] = \underline{b E_{q(\theta)}[\nabla_\phi \log q(\theta)]} = 0$$

log-derivative trick

$$\Rightarrow \hat{F}(\theta) = \Delta(\theta) \nabla_\phi \log q_\phi(\theta)$$



$b$  fitted by minimising either  $V_{q(\theta)}[\hat{F}(\theta)]$  or  $E_{q(\theta)}[\Delta(\theta)^2]$

Ranganath et al. Black Box Variational Inference. AISTATS 2014

Mnih and Gregor. Neural Variational Inference and Learning in Belief Networks. ICML 2014

# Variance Reduction Techniques in MCVI

- Application to REINFORCE gradient:

- $F(\theta) = \log \frac{p(D, \theta)}{q_\phi(\theta)} \nabla_\phi \log q_\phi(\theta)$   
 $\qquad\qquad\qquad \underline{\qquad\qquad\qquad} := f(\theta)$
- Define  $G(\theta) = g(\theta) \nabla_\phi \log q_\phi(\theta)$
- Variance reduced gradient:  $\hat{F}(\theta) = \underline{(f(\theta) - g(\theta)) \nabla_\phi \log q_\phi(\theta)} + E_{q_\phi(\theta)}[G(\theta)]$   
 $\qquad\qquad\qquad \underline{\qquad\qquad\qquad} := \Delta(\theta)$

- “Taylor expansion” approach (e.g. 1<sup>st</sup> order):

$$g(\theta) = f(\theta_0) + \nabla_{\theta_0} f(\theta_0)(\theta - \theta_0)$$

# Variance Reduction Techniques in MCVI

- Application to REINFORCE gradient:

- $F(\theta) = \log \frac{p(D, \theta)}{q_\phi(\theta)} \nabla_\phi \log q_\phi(\theta)$   
 $\underline{\hspace{10em}} := f(\theta)$
- Define  $G(\theta) = g(\theta) \nabla_\phi \log q_\phi(\theta)$
- Variance reduced gradient:  $\hat{F}(\theta) = \underline{(f(\theta) - g(\theta)) \nabla_\phi \log q_\phi(\theta)} + E_{q_\phi(\theta)}[G(\theta)]$   
 $\underline{\hspace{10em}} := \Delta(\theta)$

- “Taylor expansion” approach (e.g. 1<sup>st</sup> order):

$$g(\theta) = f(\theta_0) + \nabla_{\theta_0} f(\theta_0)(\theta - \theta_0) \Rightarrow E_{q(\theta)}[G(\theta)] = \underline{(f(\theta_0) - \nabla_{\theta_0} f(\theta_0)\theta_0)E_{q(\theta)}[\nabla_\phi \log q(\theta)]} + \underline{\nabla_{\theta_0} f(\theta_0)E_{q(\theta)}[\theta \nabla_\phi \log q(\theta)]} = \nabla_{\theta_0} f(\theta_0) \nabla_\phi E_{q(\theta)}[\theta] \text{ (log-derivative trick)}$$

# Variance Reduction Techniques in MCVI

- Application to REINFORCE gradient:

- $F(\theta) = \log \frac{p(D, \theta)}{q_\phi(\theta)} \nabla_\phi \log q_\phi(\theta)$
- Define  $G(\theta) = g(\theta) \nabla_\phi \log q_\phi(\theta)$
- $\underline{\underline{:= f(\theta)}}$
- Variance reduced gradient:  $\hat{F}(\theta) = \underline{(f(\theta) - g(\theta)) \nabla_\phi \log q_\phi(\theta)} + E_{q_\phi(\theta)}[G(\theta)]$
- $\underline{\underline{:= \Delta(\theta)}}$

- “Taylor expansion” approach (e.g. 1<sup>st</sup> order):

$$g(\theta) = f(\theta_0) + \nabla_{\theta_0} f(\theta_0)(\theta - \theta_0) \Rightarrow E_{q(\theta)}[G(\theta)] = \underline{(f(\theta_0) - \nabla_{\theta_0} f(\theta_0)\theta_0)E_{q(\theta)}[\nabla_\phi \log q(\theta)]}$$

$$\underline{\underline{+ \nabla_{\theta_0} f(\theta_0)E_{q(\theta)}[\theta \nabla_\phi \log q(\theta)]}}$$

$$= \nabla_{\theta_0} f(\theta_0) \nabla_\phi E_{q(\theta)}[\theta] \text{ (log-derivative trick)}$$

$$\Rightarrow \hat{F}(\theta) = \Delta(\theta) \nabla_\phi \log q_\phi(\theta) + \underline{\underline{\nabla_{\theta_0} f(\theta_0) \nabla_\phi E_{q(\theta)}[\theta]}}$$

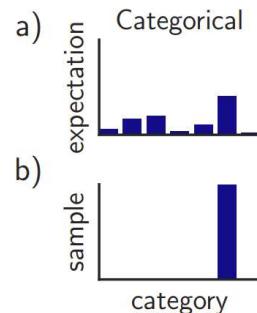
$$:= \nabla_\phi f(\theta_0) \text{ if } \theta_0 = E_{q(\theta)}[\theta]$$

Paisley et al. Variational Bayesian Inference with Stochastic Search. ICML 2012

Gu et al. MuProp: Unbiased Backpropagation for Stochastic Neural Networks. ICLR 2016

# Variance Reduction Techniques in MCVI

- Gumbel-Softmax trick
  - Biased gradient estimator
  - Empirically found to have smaller variance

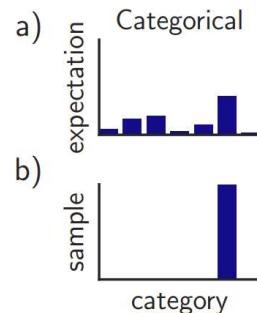


Categorical distribution:

$$p(y = k) = \pi_k, \sum_k \pi_k = 1$$

# Variance Reduction Techniques in MCVI

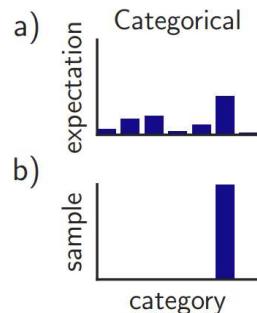
- Gumbel-Softmax trick
  - Biased gradient estimator
  - Empirically found to have smaller variance



Gumbel trick to sample  $y$ :  
$$y = \arg \max [g_k + \log \pi_k],$$
$$g_k \sim \text{Gumbel}(0, 1)$$

# Variance Reduction Techniques in MCVI

- Gumbel-Softmax trick
  - Biased gradient estimator
  - Empirically found to have smaller variance



Gumbel trick to sample  $y$ :

$$y = \arg \max [g_k + \log \pi_k],$$

replace with softmax

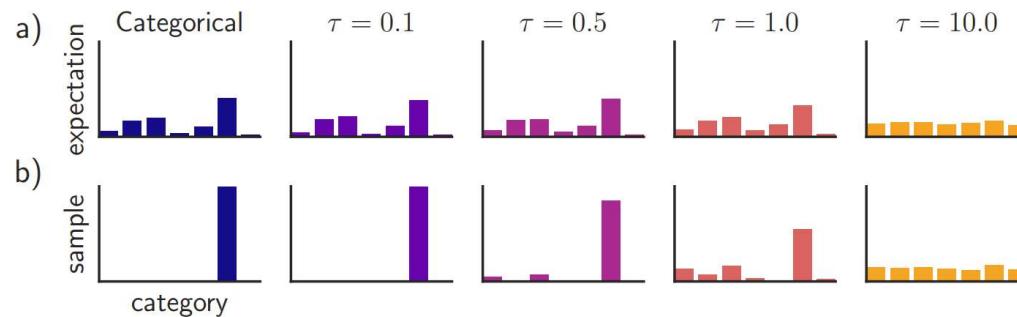
$$g_k \sim \text{Gumbel}(0, 1)$$

Jang et al. Categorical Reparameterization with Gumbel-Softmax. ICLR 2017

Maddison et al. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. ICLR 2017

# Variance Reduction Techniques in MCVI

- Gumbel-Softmax trick
  - Biased gradient estimator
  - Empirically found to have smaller variance



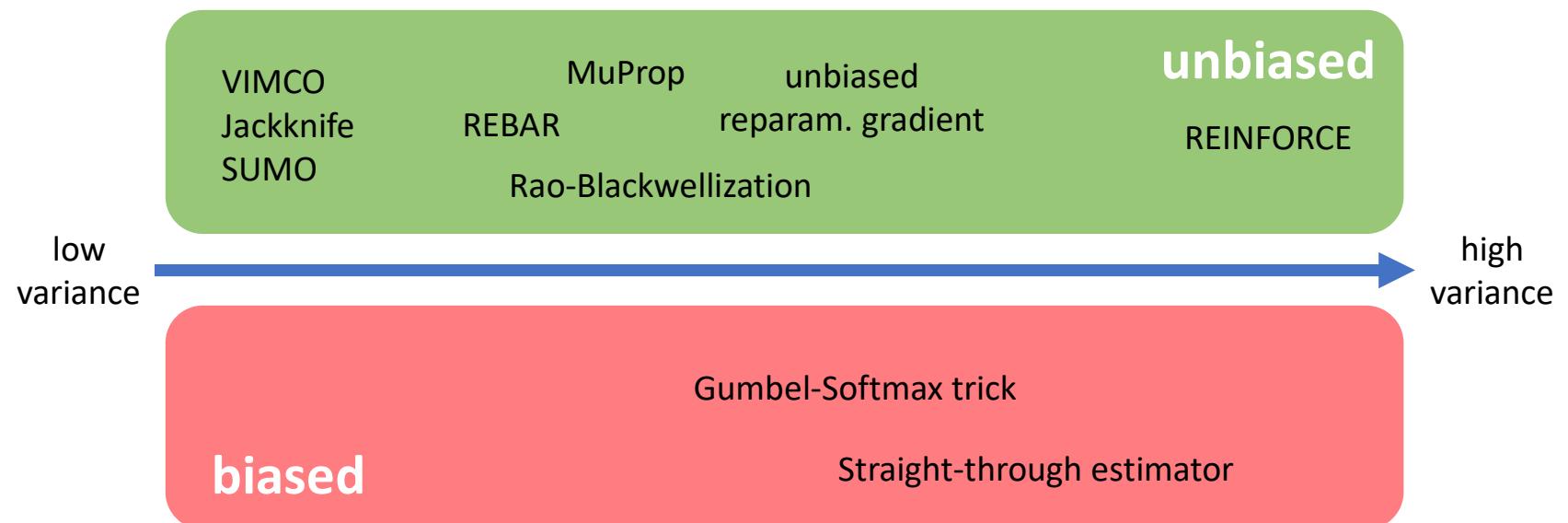
Concrete distribution/Gumbel-Softmax trick:  
sample the “soft vector” (instead of one-hot encoding of  $y$ )

$$[y_1, \dots, y_K] = \text{softmax}\left(\left[\frac{(g_1 + \log \pi_1)}{\tau}, \dots, \frac{(g_K + \log \pi_K)}{\tau}\right]\right)$$

Jang et al. Categorical Reparameterization with Gumbel-Softmax. ICLR 2017

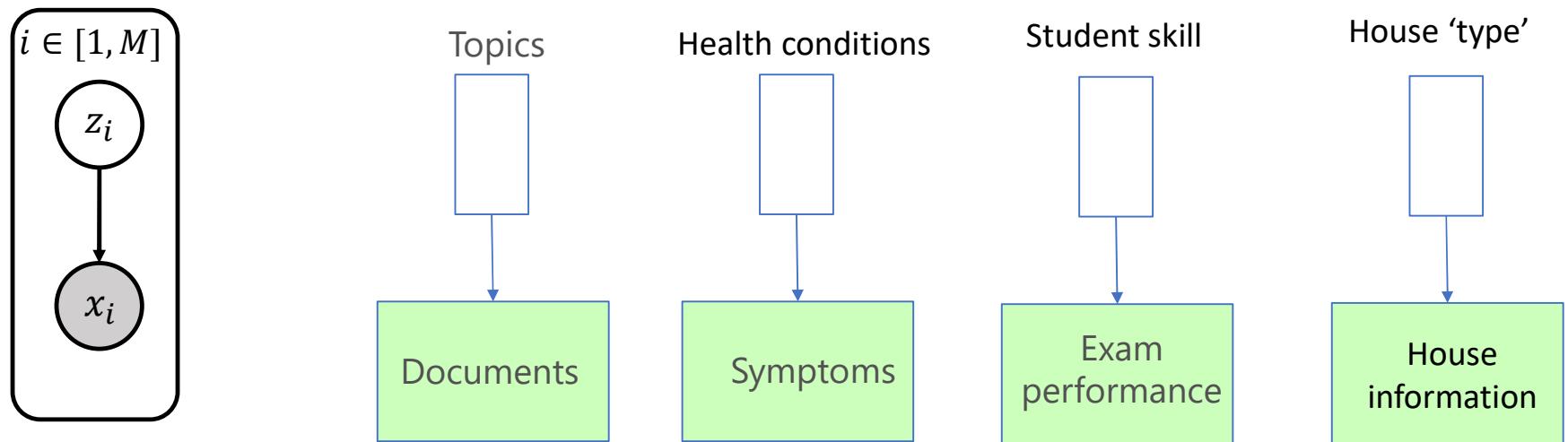
Maddison et al. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. ICLR 2017

# Variance Reduction Techniques in MCVI

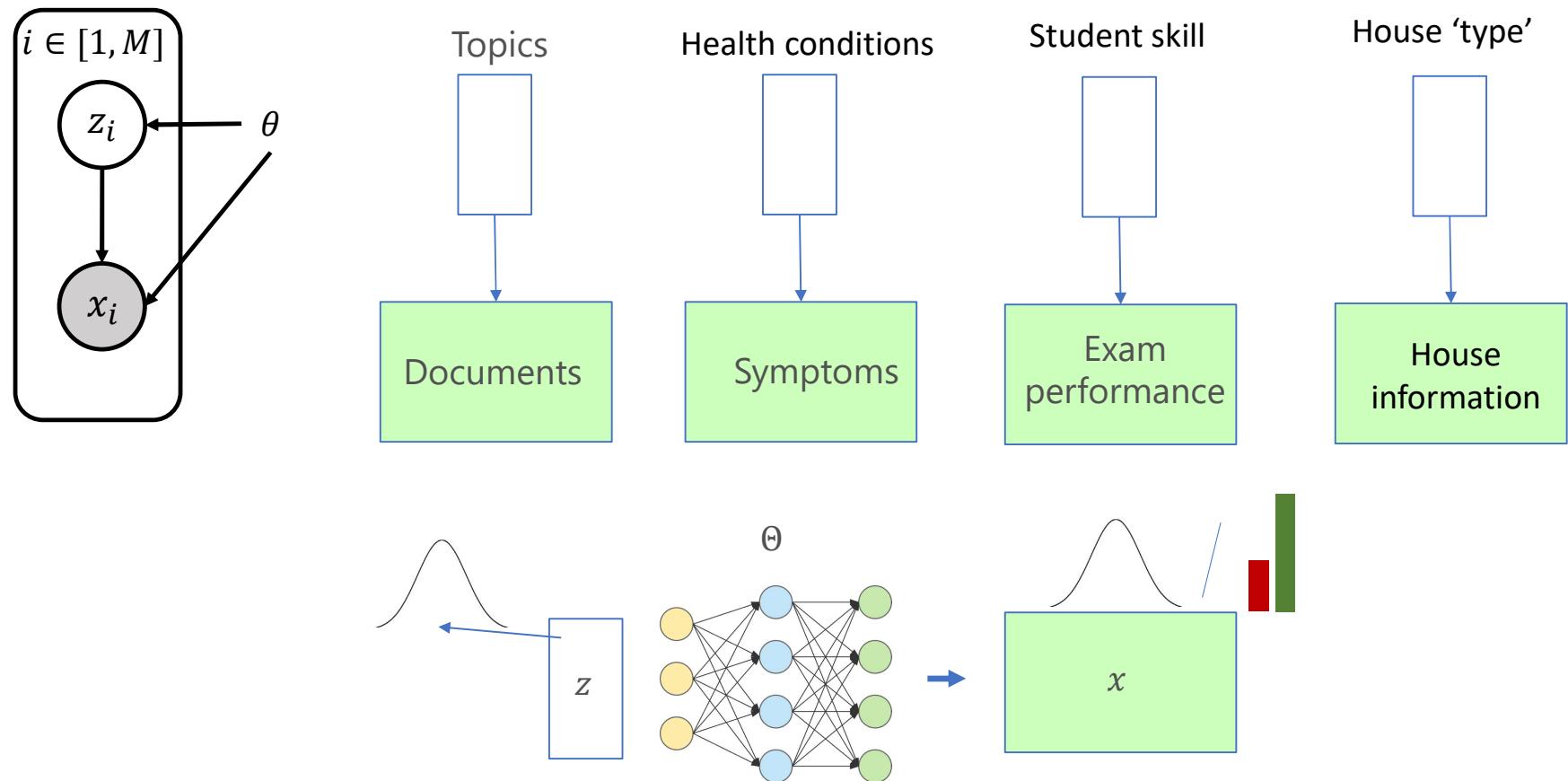


For an incomplete list of variance reduced gradient estimators, see [http://yingzhenli.net/home/en/?page\\_id=1262](http://yingzhenli.net/home/en/?page_id=1262)

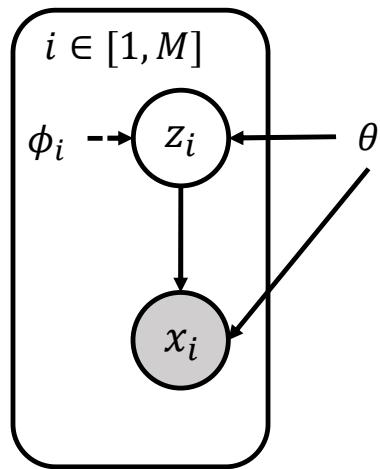
# Latent Variable Model



# Deep Latent Variable Model



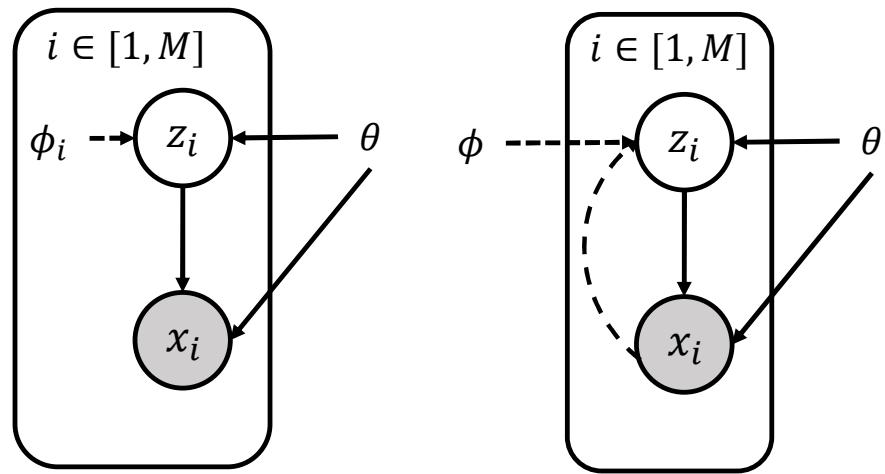
# Amortized Inference



$\phi$  parameter for variational distribution

$\theta$  decoder parameter

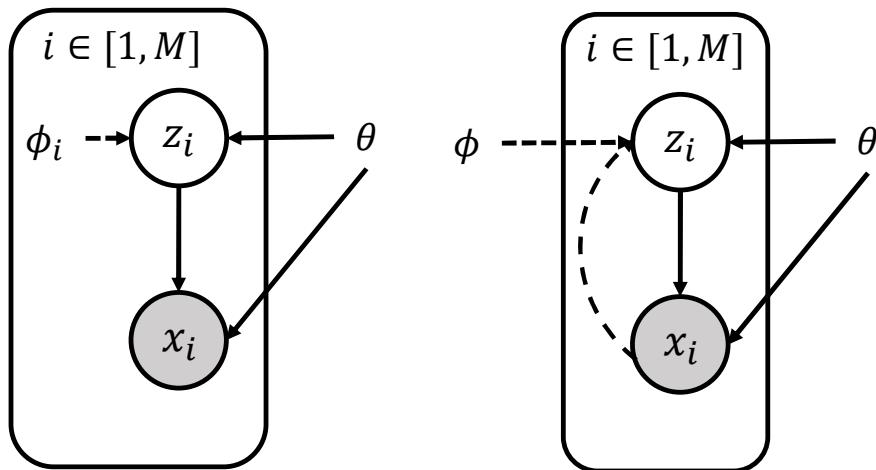
# Amortized Inference



$\phi$  parameter for variational distribution

$\theta$  decoder parameter

# Amortized Inference

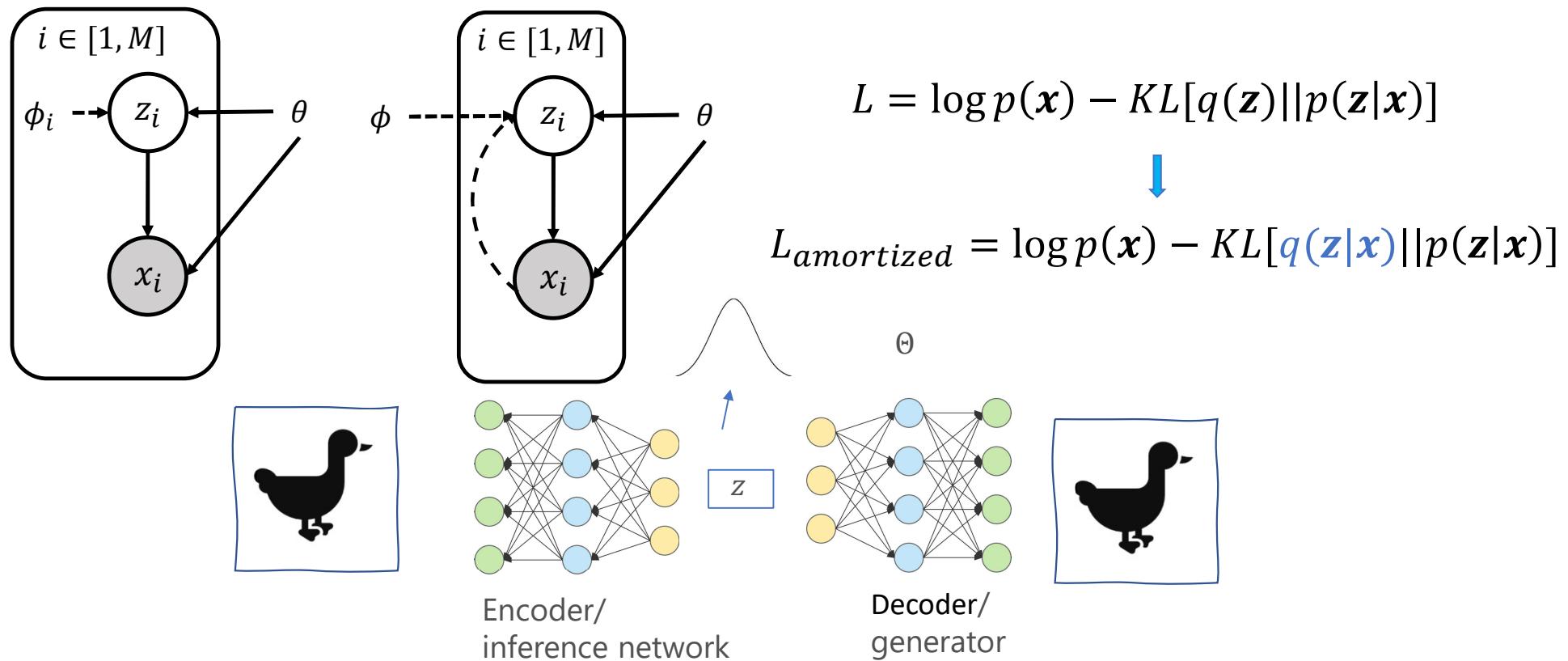


$$L = \log p(\mathbf{x}) - KL[q(\mathbf{z}) || p(\mathbf{z}|\mathbf{x})]$$

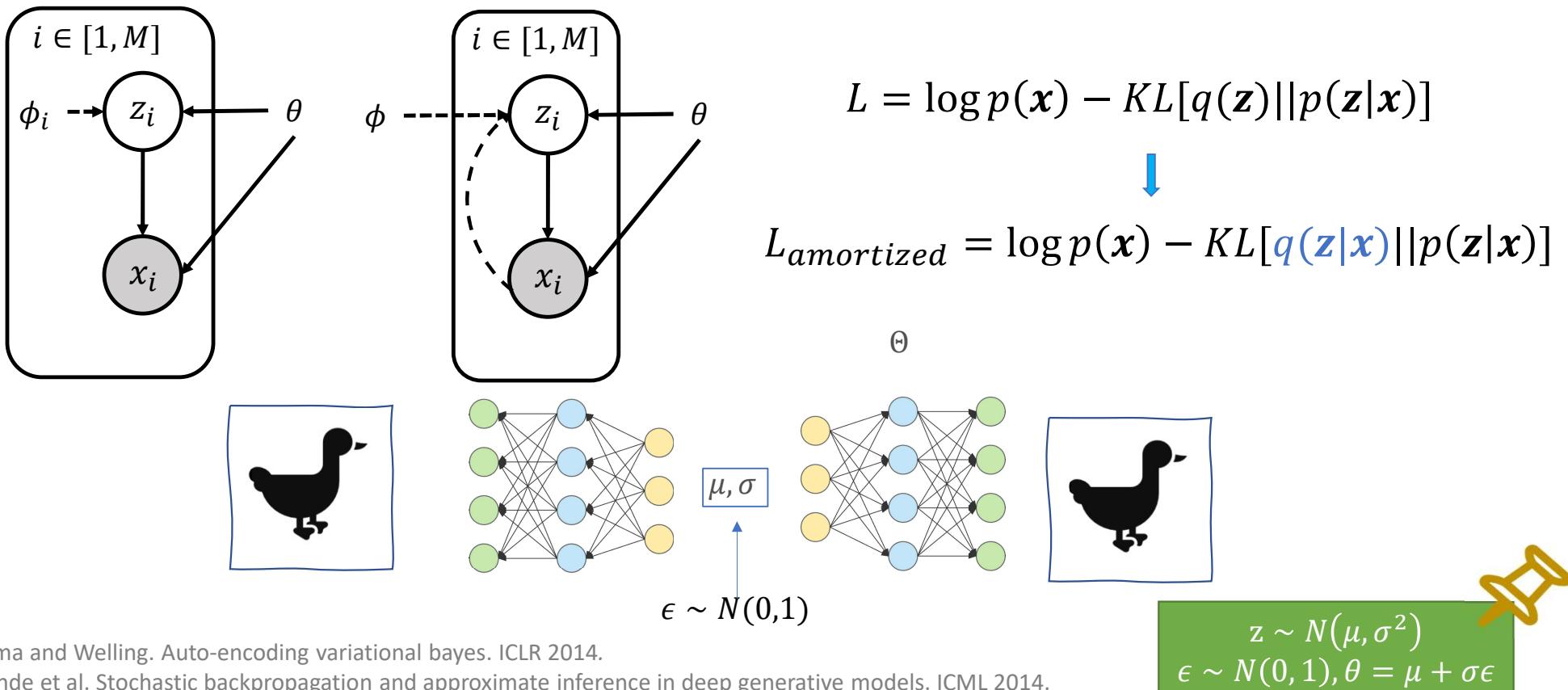


$$L_{\text{amortized}} = \log p(\mathbf{x}) - KL[q(\mathbf{z}|\mathbf{x}) || p(\mathbf{z}|\mathbf{x})]$$

# Variational Auto-Encoders (VAE)



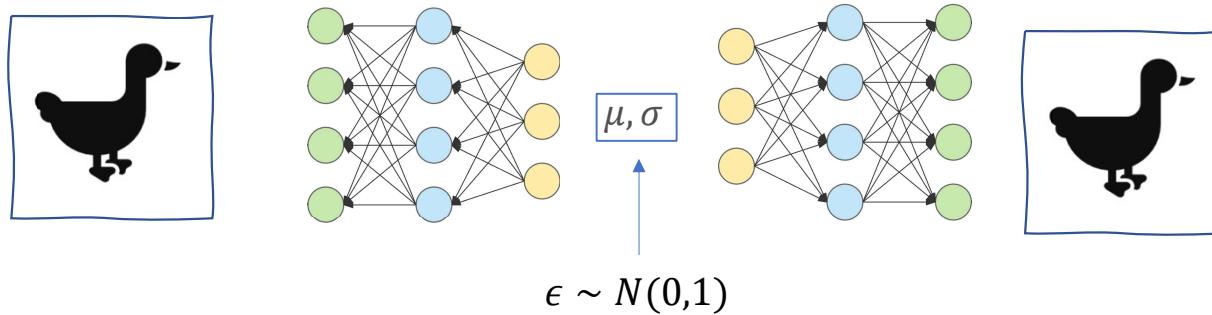
# Variational Auto-Encoders (VAE)



Kingma and Welling. Auto-encoding variational bayes. ICLR 2014.

Rezende et al. Stochastic backpropagation and approximate inference in deep generative models. ICML 2014.

# Variational Auto-Encoders (VAE)



$$\begin{aligned} L_{amortized} &= \log p(\mathbf{x}) - KL[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})] \\ &= E_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - KL[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})] \end{aligned}$$

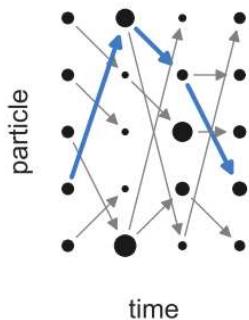
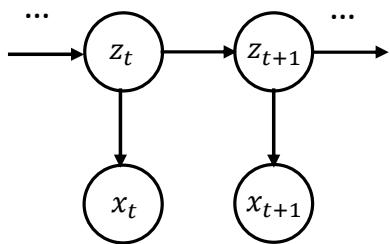
Kingma and Welling. Auto-encoding variational bayes. ICLR 2014.

Rezende et al. Stochastic backpropagation and approximate inference in deep generative models. ICML 2014.

How to apply amortization to  
other inference methods?

# Amortized Inference: Further Examples

- Amortized SMC



Find the optimal proposal distribution for each sequence  $\{x_{1:T}\}$  & each time step

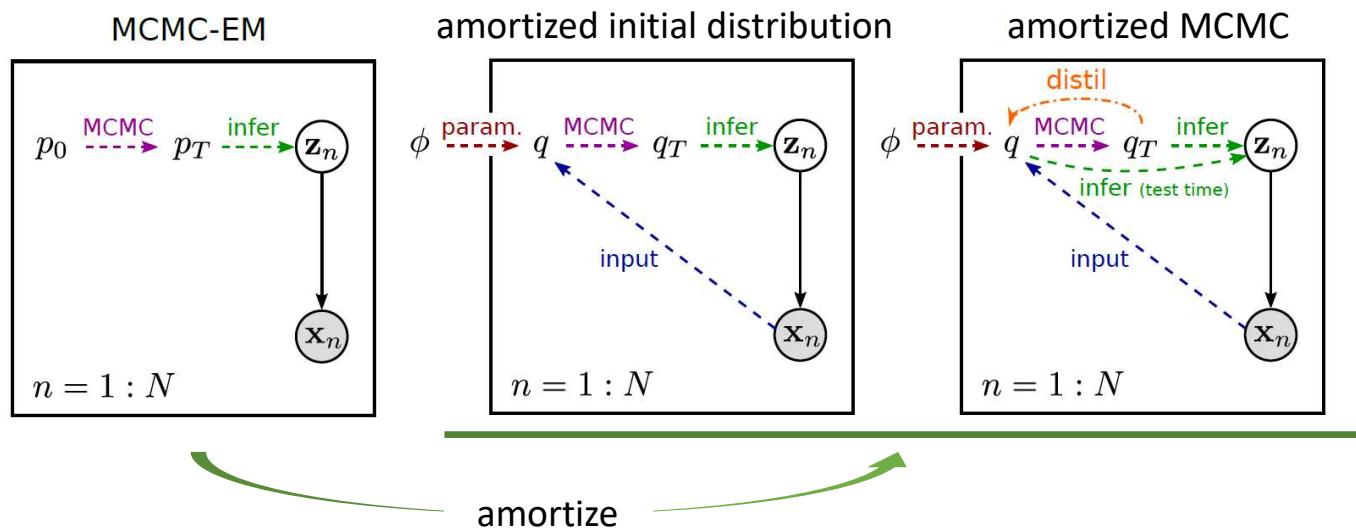
amortize

Explicitly parameterise & optimise  $(x_{1:T}, z_{1:t}) \rightarrow$  proposal dist. for  $z_t$

Naesseth et al. Variational Sequential Monte Carlo. AISTATS 2018  
Maddison et al. Filtering Variational Objectives. NeurIPS 2017  
Le et al. Auto-encoding Sequential Monte Carlo. ICLR 2018

# Amortized Inference: Further Examples

- Amortized MCMC



Hoffman. Learning Deep Latent Gaussian Models with Markov Chain Monte Carlo. ICML 2017

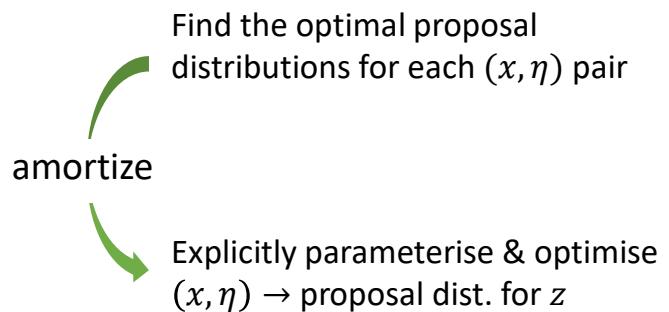
Li et al. Approximate Inference with Amortised MCMC. ICML 2017 Workshop on Implicit Models

# Amortized Inference: Further Examples

- Amortized Monte Carlo integration

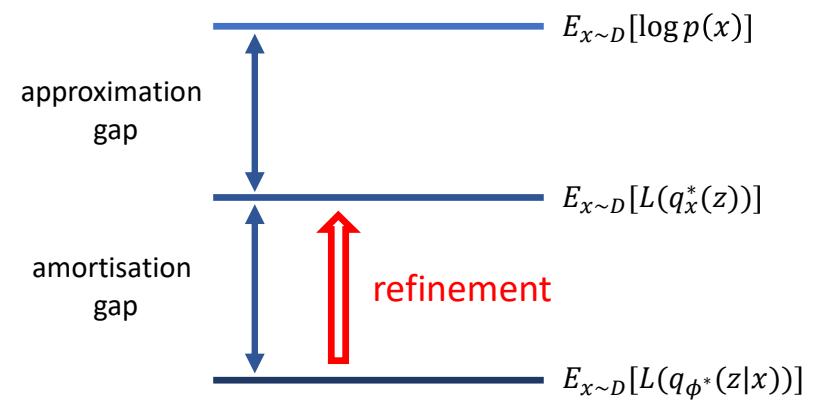
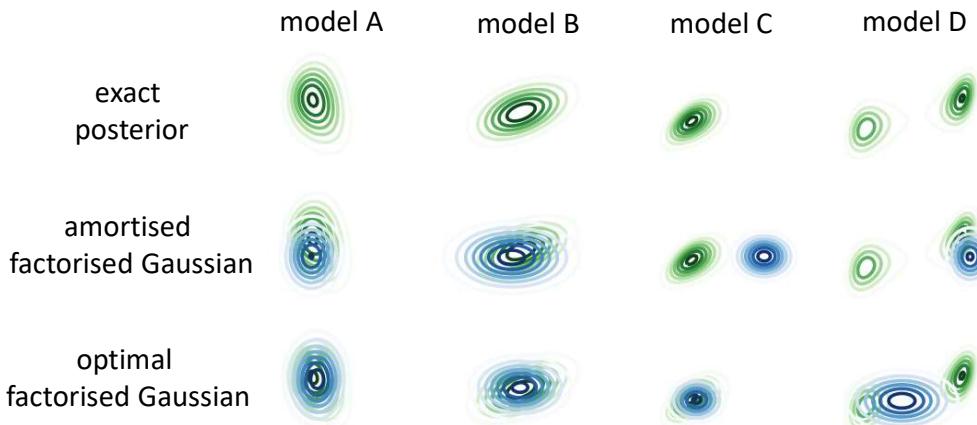
Goal: estimate with  
importance sampling

$$E_{p(z|x)}[F_\eta(z)]$$



# Amortized Inference: Limitations

- Amortised approximate posteriors in practice are sub-optimal

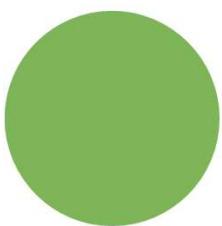


- The “refinement” idea:
  - Initialise  $q(z|x) = N(z; \mu, \sigma^2)$  with the amortised solution  $\mu \leftarrow \mu_\phi(x), \sigma \leftarrow \sigma_\phi(x)$
  - Then run  $T$  more VI gradient steps to update  $\mu, \sigma$

Cremer et al. Inference Suboptimality in Variational Autoencoders. ICML 2018

Marino et al. Iterative Amortized Inference. ICML 2018

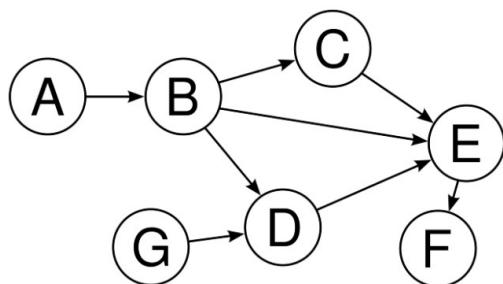
Kim et al. Semi-Amortized Variational Autoencoders. ICML 2018



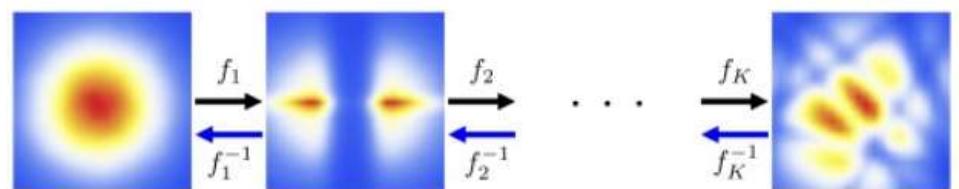
# Part II: Advances

- Scalable variational inference
- Monte Carlo methods
- Amortized inference
- **Approximate distribution design**
- **Optimization objective design**

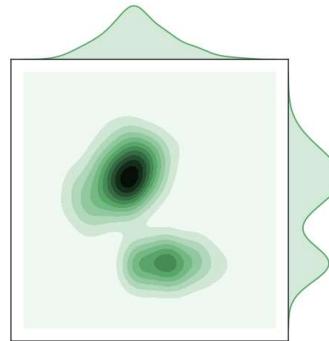
# Designing $q$ Distributions



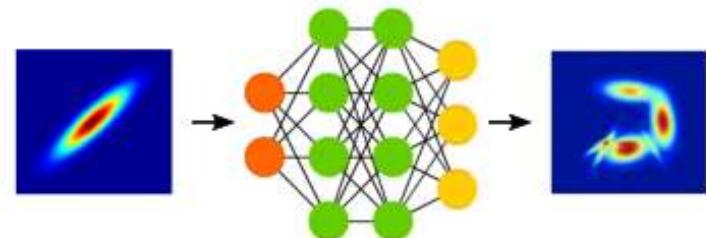
Structured approximations



Normalizing flows



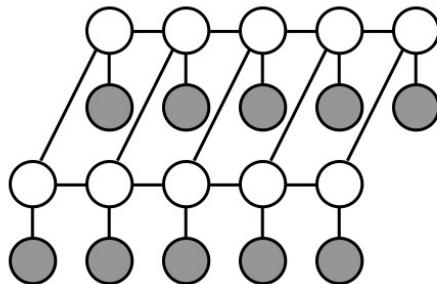
Auxiliary variables & mixture distributions



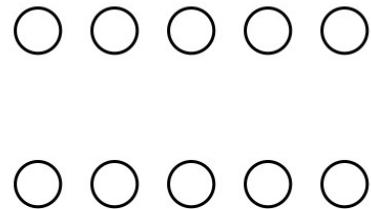
Implicit approximate posteriors

# Structured Approximations

- introduce dependencies between random variables for  $q$ :



Hidden Markov Model



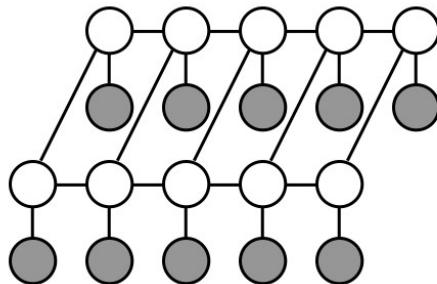
Mean-field approximation

Exact posterior  $p(z | x)$   
 $z_i \not\perp z_j | x$

$$q(z) = \prod_i q(z_i)$$

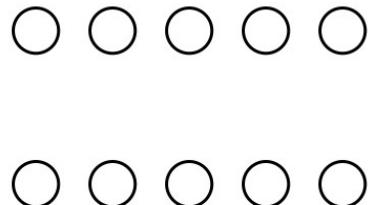
# Structured Approximations

- introduce dependencies between random variables for  $q$ :



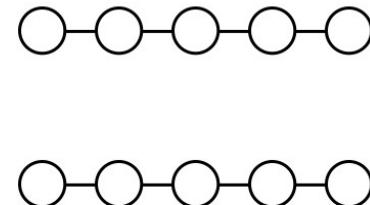
Hidden Markov Model

Exact posterior  $p(z \mid x)$   
 $z_i \not\perp z_j \mid x$



Mean-field approximation

$$q(z) = \prod_i q(z_i)$$



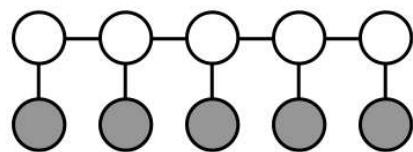
Structured approximation

$$\begin{aligned} q(z) &= \prod_s q(z_s) \\ q(z_s) &= q(\{z_i\}_{i \in s}) \end{aligned}$$

Main design question:  
the grouping and  
conditional dependency  
structure

# Structured Approximations

- Auto-regressive distributions (as a specific dependency structure)



Hidden Markov Model

Exact posterior  $p(z | x)$   
 $z_i \not\perp z_j | x$



Structured approximation

$$q(z) = \prod_s q(z_s)$$
$$q(z_s) = q(\{z_i\}_{i \in s})$$



Auto-regressive approximation

$$q(z) = \prod_i q(z_i | z_{<i})$$
$$q(z_1 | z_{<1}) = q(z_1)$$

Main design question:  
the ordering of the  
latent variables

# Normalizing Flows

- Change-of-variable formula:
  - $x$  is a random variable with probability density function (PDF)  $p_X(x)$
  - $y = f(x)$  is an invertible mapping

# Normalizing Flows

- Change-of-variable formula:
  - $x$  is a random variable with probability density function (PDF)  $p_X(x)$
  - $y = f(x)$  is an invertible mapping
  - The probability mass is preserved, and the PDF for  $y = f(x)$  satisfies

$$\frac{p_Y(y)dy}{\text{prob. mass of region around } y} = \frac{p_X(x)dx}{\text{prob. mass of region around } x}$$

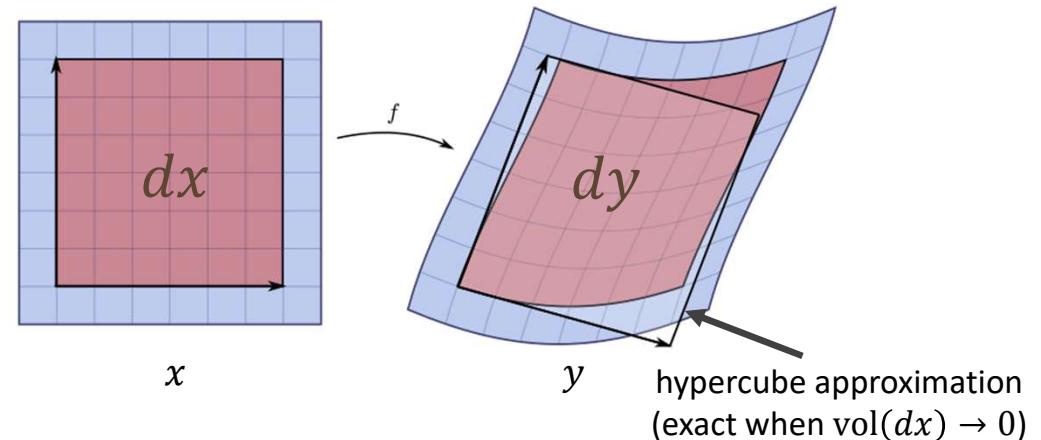
# Normalizing Flows

- Change-of-variable formula:
  - $x$  is a random variable with probability density function (PDF)  $p_X(x)$
  - $y = f(x)$  is an invertible mapping
  - The probability mass is preserved, and the PDF for  $y = f(x)$  satisfies

$$\frac{p_Y(y)dy}{\text{prob. mass of region around } y} = \frac{p_X(x)dx}{\text{prob. mass of region around } x}$$

$$p_Y(y) = p_X(x) \left| \det \left( \frac{dx}{dy} \right) \right|$$

$$p_X(x) = p_Y(y) \left| \det \left( \frac{dy}{dx} \right) \right|$$



# Normalizing Flows

- Variational inference with Normalizing flow

- Assume  $q_0(z_0) = N(z_0; 0, I)$
- Define  $z = f_\phi(z_0)$  where  $f_\phi(\cdot)$  is an invertible mapping parameterized by  $\phi$

$$q(z) = q_0(z_0) \left| \det \left( \frac{dz}{dz_0} \right) \right|^{-1} \quad \text{with } z_0 = f_\phi^{-1}(z)$$

(change of variable:  $q(z)dz = q_0(z_0)dz_0$ )

# Normalizing Flows

- Variational inference with Normalizing flow

- Assume  $q_0(z_0) = N(z_0; 0, I)$
- Define  $z = f_\phi(z_0)$  where  $f_\phi(\cdot)$  is an invertible mapping parameterized by  $\phi$

$$q(z) = q_0(z_0) \left| \det \left( \frac{dz}{dz_0} \right) \right|^{-1} \quad \text{with } z_0 = f_\phi^{-1}(z)$$

- Fit  $q(z)$  to  $p(x | z)$  with VI:

$$L(q(z)) = E_{q(z)}[\log p(x | z) + \log p(z) - \log q(z)]$$

# Normalizing Flows

- Variational inference with Normalizing flow

- Assume  $q_0(z_0) = N(z_0; 0, I)$
- Define  $z = f_\phi(z_0)$  where  $f_\phi(\cdot)$  is an invertible mapping parameterized by  $\phi$

$$q(z) = q_0(z_0) \left| \det \left( \frac{dz}{dz_0} \right) \right|^{-1} \quad \text{with } z_0 = f_\phi^{-1}(z)$$

- Fit  $q(z)$  to  $p(x | z)$  with VI:

$$\begin{aligned} L(q(z)) &= E_{q(z)}[\log p(x | z) + \log p(z) - \log q(z)] && \text{by def. of } q(z) \\ &= E_{q(z)} \left[ \log p(x, z) - \log q_0(z_0 = f_\phi^{-1}(z)) \left| \det \left( \frac{dz}{dz_0} \right) \right|^{-1} \right] \end{aligned}$$

# Normalizing Flows

- Variational inference with Normalizing flow

- Assume  $q_0(z_0) = N(z_0; 0, I)$
- Define  $z = f_\phi(z_0)$  where  $f_\phi(\cdot)$  is an invertible mapping parameterized by  $\phi$

$$q(z) = q_0(z_0) \left| \det \left( \frac{dz}{dz_0} \right) \right|^{-1} \quad \text{with } z_0 = f_\phi^{-1}(z)$$

- Fit  $q(z)$  to  $p(x | z)$  with VI:

$$\begin{aligned} L(q(z)) &= E_{q(z)}[\log p(x | z) + \log p(z) - \log q(z)] && \text{by def. of } q(z) \\ &= E_{q(z)}\left[\log p(x, z) - \log q_0(z_0 = f_\phi^{-1}(z)) \left| \det \left( \frac{dz}{dz_0} \right) \right|^{-1}\right] \\ &= E_{q_0(z_0)}\left[\log p(x, f_\phi(z_0)) - \log q_0(z_0) + \log \left| \det \left( \frac{df_\phi}{dz_0} \right) \right|\right] \end{aligned}$$

reparam. trick:  
 $z \sim q(z) \Leftrightarrow z_0 \sim q_0(z_0), z = f_\phi(z_0)$

# Normalizing Flows

- Variational inference with Normalizing flow

- Assume  $q_0(z_0) = N(z_0; 0, I)$
- Define  $z = f_\phi(z_0)$  where  $f_\phi(\cdot)$  is an invertible mapping parameterized by  $\phi$

$$q(z) = q_0(z_0) \left| \det \left( \frac{dz}{dz_0} \right) \right|^{-1} \quad \text{with } z_0 = f_\phi^{-1}(z)$$

- Fit  $q(z)$  to  $p(x | z)$  with VI:

$$L(q(z)) = E_{q(z)}[\log p(x | z) + \log p(z) - \log q(z)] \quad \text{by def. of } q(z)$$

$$= E_{q(z)} \left[ \log p(x, z) - \log q_0(z_0 = f_\phi^{-1}(z)) \left| \det \left( \frac{dz}{dz_0} \right) \right|^{-1} \right]$$

$$= E_{q_0(z_0)} \left[ \log p(x, f_\phi(z_0)) - \log q_0(z_0) + \log \left| \det \left( \frac{df_\phi}{dz_0} \right) \right| \right]$$

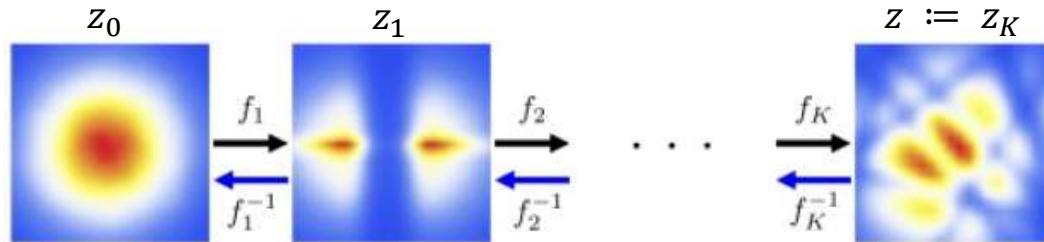
reparam. trick:

$$z \sim q(z) \Leftrightarrow z_0 \sim q_0(z_0), z = f_\phi(z_0)$$

- Computing ELBO requires  $\log \left| \det \left( \frac{df_\phi}{dz_0} \right) \right|$

# Normalizing Flows

- Variational inference with Normalizing flow
  - Idea: define  $f_\phi$  such that  $\log |\det\left(\frac{df_\phi}{dz_0}\right)|$  is easy to compute!
  - Chain simple invertible mappings together to make a flexible mapping



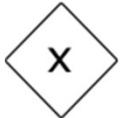
$$f_\phi = f_K \circ f_{K-1} \circ \cdots \circ f_1, f_k(\cdot) := f_{\phi_k}(\cdot), \phi = \{\phi_k\}_{k=1}^K$$

- For each simple mapping, hopefully the Jacobian log-determinant is easy to compute

$$\Rightarrow \log |\det\left(\frac{df_\phi}{dz_0}\right)| = \sum_{k=1}^K \log |\det\left(\frac{dz_k}{dz_{k-1}}\right)|$$

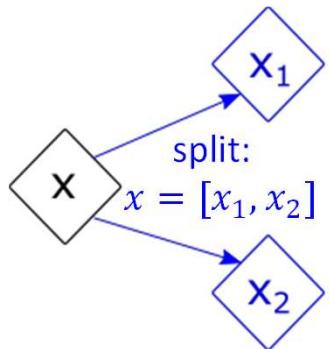
# Normalizing Flows

- Goal: construct  $f_k$  to enable fast compute of  $\log |\det(\frac{dz_k}{dz_{k-1}})|$ 
  - Example (RealNVP):  $y := f_{\phi_k}(x)$  computed as follows



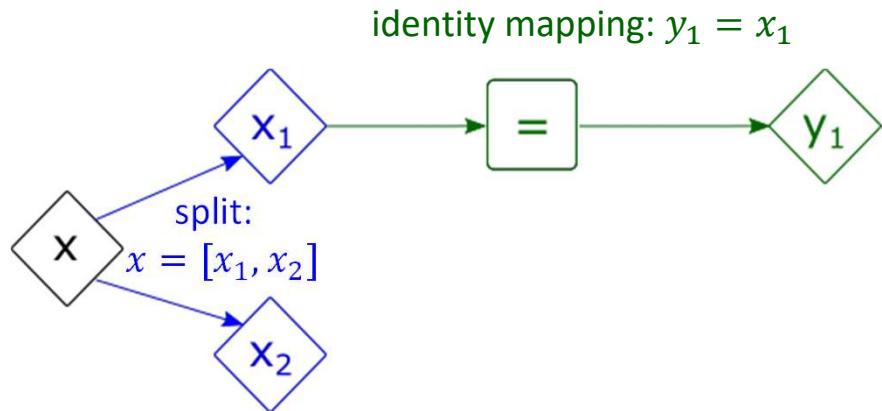
# Normalizing Flows

- Goal: construct  $f_k$  to enable fast compute of  $\log |\det(\frac{dz_k}{dz_{k-1}})|$ 
  - Example (RealNVP):  $y := f_{\phi_k}(x)$  computed as follows



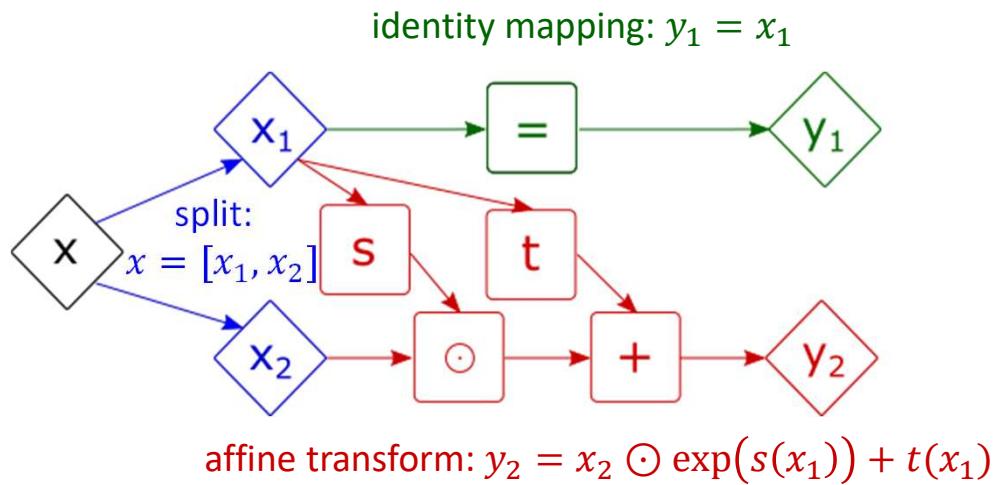
# Normalizing Flows

- Goal: construct  $f_k$  to enable fast compute of  $\log |\det(\frac{dz_k}{dz_{k-1}})|$ 
  - Example (RealNVP):  $y \coloneqq f_{\phi_k}(x)$  computed as follows



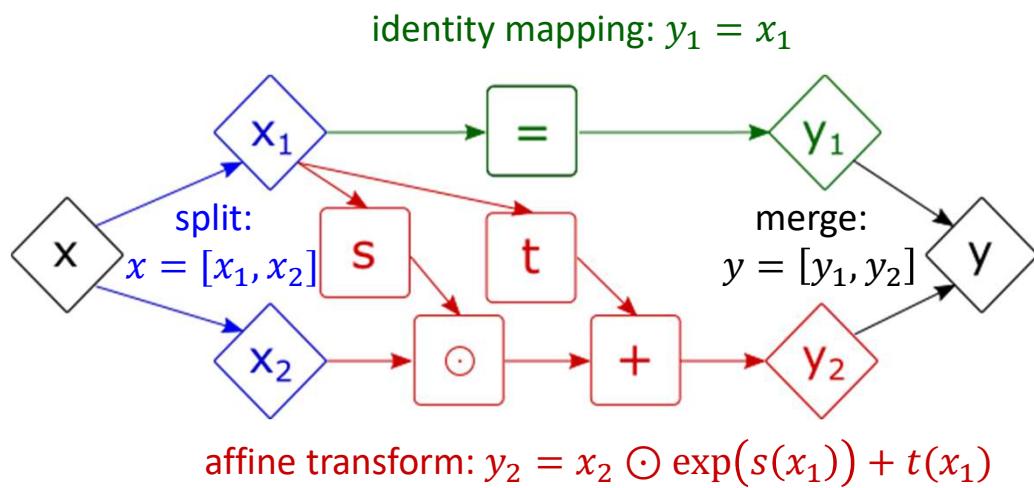
# Normalizing Flows

- Goal: construct  $f_k$  to enable fast compute of  $\log |\det(\frac{dz_k}{dz_{k-1}})|$ 
  - Example (RealNVP):  $y \coloneqq f_{\phi_k}(x)$  computed as follows



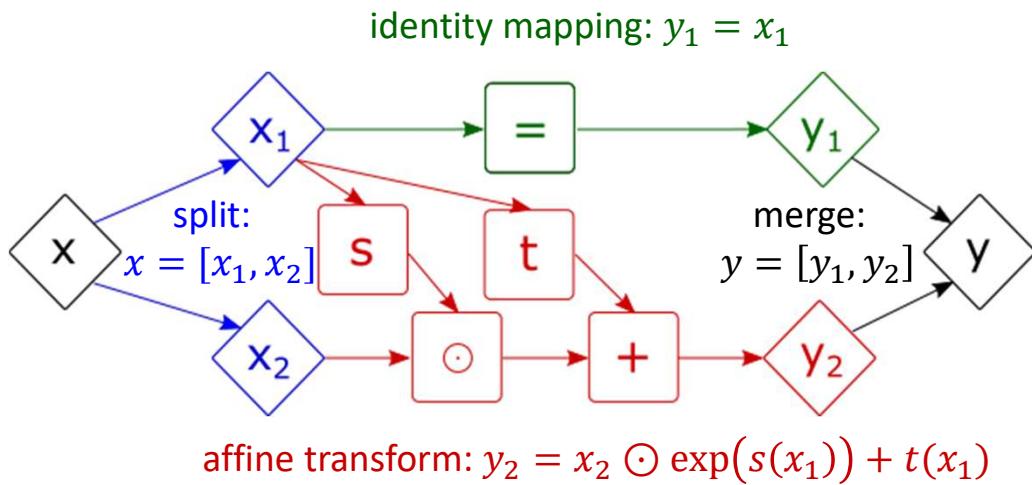
# Normalizing Flows

- Goal: construct  $f_k$  to enable fast compute of  $\log |\det(\frac{dz_k}{dz_{k-1}})|$ 
  - Example (RealNVP):  $y := f_{\phi_k}(x)$  computed as follows



# Normalizing Flows

- Goal: construct  $f_k$  to enable fast compute of  $\log |\det(\frac{dz_k}{dz_{k-1}})|$ 
  - Example (RealNVP):  $y \coloneqq f_{\phi_k}(x)$  computed as follows



Jacobian:

$$\frac{df_{\phi_k}}{dx} = \begin{pmatrix} I & 0 \\ dy_2/dx_1 & \text{diag}(\exp(s(x_1))) \end{pmatrix}$$

Log-determinant of Jacobian:

$$\Rightarrow \log \left| \det \left( \frac{df_{\phi}}{dx} \right) \right| = \sum_i s(x_1)_i$$

# Auxiliary Variables & Mixture Distributions

- Construct  $q(\theta)$  as a (hierarchical) mixture distribution

$$q(\theta) = \int q(\theta | a) q(a) da$$

- $a$  is the auxiliary variable used to enrich the approximate posterior

# Auxiliary Variables & Mixture Distributions

- Construct  $q(\theta)$  as a (hierarchical) mixture distribution

$$q(\theta) = \int q(\theta | a) q(a) da$$

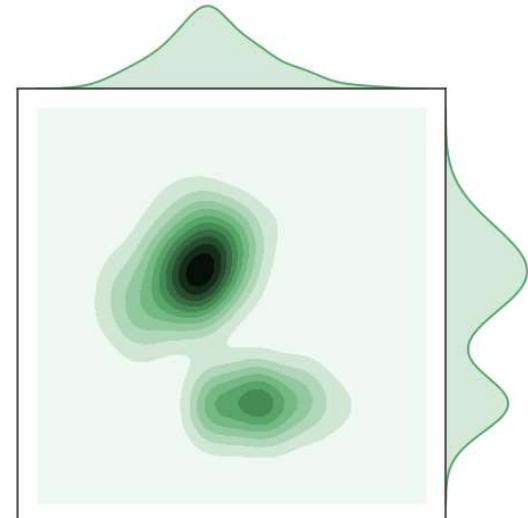
- $a$  is the auxiliary variable used to enrich the approximate posterior

- Example: Mixture of Gaussians

$$a \sim q(a) = \text{Categorical}(\pi_1, \dots, \pi_K)$$

$$\theta \sim q(\theta | a) = N(\theta; m_a, \Sigma_a)$$

Can be very flexible with many components!



# Auxiliary Variables & Mixture Distributions

- Construct  $q(\theta)$  as a (hierarchical) mixture distribution

$$q(\theta) = \int q(\theta | a) q(a) da$$

- $a$  is the auxiliary variable used to enrich the approximate posterior
- Now the variational lower-bound becomes intractable:

$$L(\phi) = \underbrace{E_{q(\theta)}[\log p(D, \theta)]}_{\text{Estimated by Monte Carlo: } a_k \sim q(a), \theta_k \sim q(\theta | a_k)} - \underbrace{E_{q(\theta)}[\log q(\theta)]}_{\text{Intractable density } q(\theta) = \int q(\theta | a)q(a) da}$$

Estimated by Monte Carlo:  
 $a_k \sim q(a), \theta_k \sim q(\theta | a_k)$

Intractable density  
 $q(\theta) = \int q(\theta | a)q(a) da$

# Auxiliary Variables & Mixture Distributions

- Solution: introducing an auxiliary variational lower-bound  $L(\phi, r)$  with an auxiliary distribution  $r(a|\theta)$ :

The diagram illustrates the components of the variational lower-bound. At the top is  $\log p(D)$ . A blue double-headed vertical arrow connects it to  $KL(q(\theta) \parallel p(\theta|D))$ . Below this is  $L(\phi) = E_{q(\theta)}[\log p(D, \theta)] - E_{q(\theta)}[\log q(\theta)]$ , indicated by a blue horizontal line with a red arrow pointing up to the first double-headed arrow. Below  $L(\phi)$  is  $L(\phi, r) = E_{q(\theta, a)}[\log p(D|\theta)] - KL[q(\theta, a)||p(\theta)r(a|\theta)]$ , indicated by a red horizontal line with a red arrow pointing down to the second double-headed arrow.

$$\log p(D)$$
$$KL(q(\theta) \parallel p(\theta|D))$$
$$L(\phi) = E_{q(\theta)}[\log p(D, \theta)] - E_{q(\theta)}[\log q(\theta)]$$
$$L(\phi, r) = E_{q(\theta, a)}[\log p(D|\theta)] - KL[q(\theta, a)||p(\theta)r(a|\theta)]$$

Agakov and Barber. An Auxiliary Variational Method. ICONIP 2004

Salimans et al. Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. ICML 2015

Ranganath et al. Hierarchical Variational Models. ICML 2016

# Auxiliary Variables & Mixture Distributions

- Solution: introducing an auxiliary variational lower-bound  $L(\phi, r)$  with an auxiliary distribution  $r(a|\theta)$ :

The diagram shows three horizontal lines representing different terms:

- The top line is labeled  $\log p(D)$ .
- The middle line is labeled  $L(\phi) = E_{q(\theta)}[\log p(D, \theta)] - E_{q(\theta)}[\log q(\theta)]$ .
- The bottom line is labeled  $L(\phi, r) = E_{q(\theta, a)}[\log p(D|\theta)] - KL[q(\theta, a)||p(\theta)r(a|\theta)]$ .

Vertical double-headed arrows connect the top line to the middle line, and the middle line to the bottom line. A red arrow points from the bottom line up to the middle line.

Labels for the arrows:

- Between  $\log p(D)$  and  $L(\phi)$ :  $KL(q(\theta)|| p(\theta|D))$
- Between  $L(\phi)$  and  $L(\phi, r)$ :  $E_{q(\theta)}[KL[q(a|\theta)||r(a|\theta)]]$

- Optimize  $r(a|\theta)$  to close the gap!
- $L(\phi, r)$  estimated by Monte Carlo:  $a_k \sim q(a), \theta_k \sim q(\theta | a_k)$

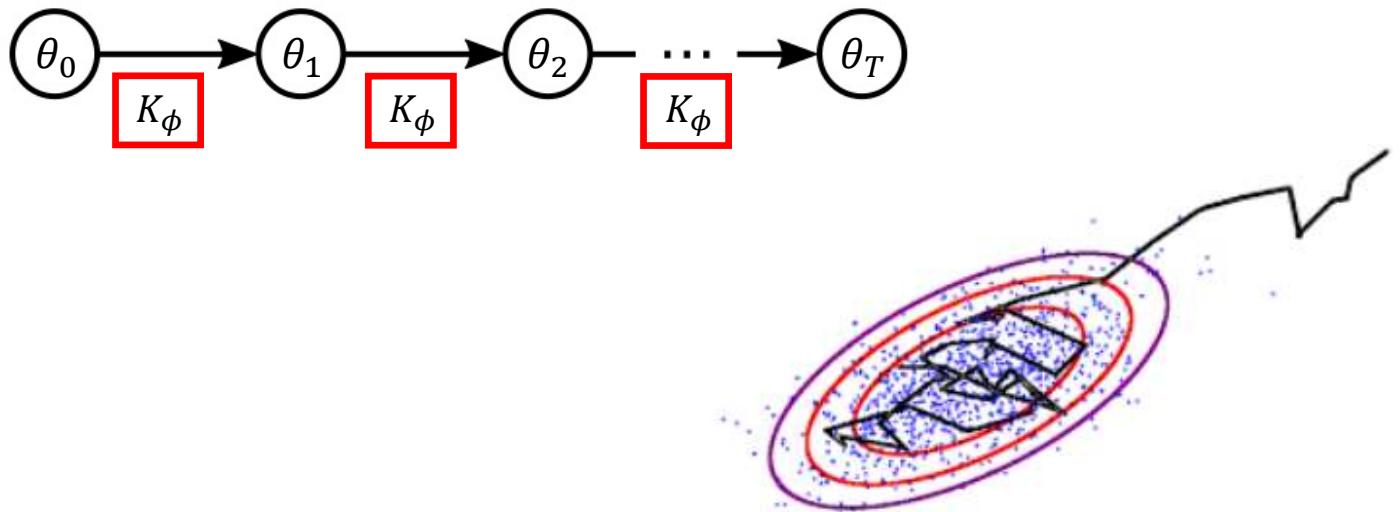
Agakov and Barber. An Auxiliary Variational Method. ICONIP 2004

Salimans et al. Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. ICML 2015

Ranganath et al. Hierarchical Variational Models. ICML 2016

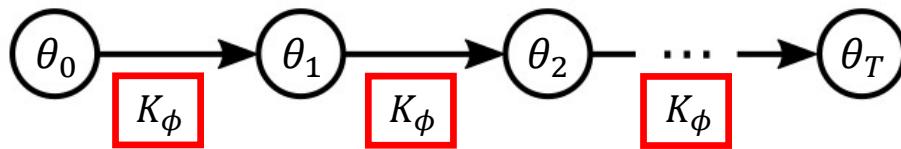
# Auxiliary Variables & Mixture Distributions

- Hierarchical mixture distributions for  $q(\theta, a)$ 
  - VI-MCMC hybrid: build  $q(\theta)$  with a Markov Chain:



# Auxiliary Variables & Mixture Distributions

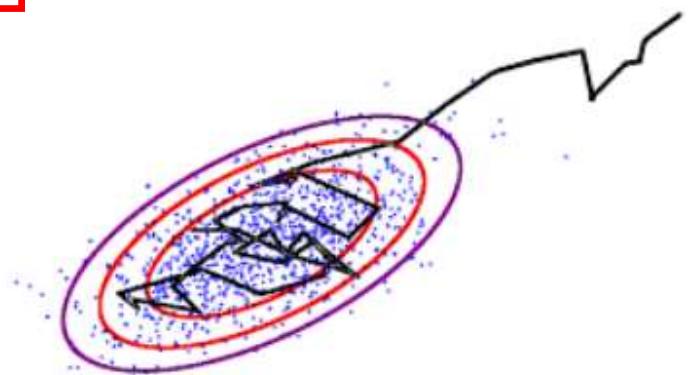
- Hierarchical mixture distributions for  $q(\theta, a)$ 
  - VI-MCMC hybrid: build  $q(\theta)$  with a Markov Chain:



learn the transition kernel with VI:

$$\theta := \theta^T, a = \{\theta^{0:T-1}\}$$

$$q(\theta^T) = \int q_0(\theta^0) \prod_{t=1}^T K_\phi(\theta^t | \theta^{t-1}) d\theta^{0:T-1}$$



Salimans et al. Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. ICML 2015

Huang et al. Improving Explorability in Variational Inference with Annealed Variational Objectives. NeurIPS 2018

# Implicit Approximate Posteriors

- Two quantities computed in (approximate) Bayesian inference:

approximate Bayesian predictive	approximate posterior moments
$p(y^* x^*, D) \approx E_{q(\theta)}[p(y^* x^*, \theta)]$	$E_{q(\theta)}[F(\theta)]$

# Implicit Approximate Posteriors

- Two quantities computed in (approximate) Bayesian inference:

approximate Bayesian predictive

$$p(y^*|x^*, D) \approx E_{q(\theta)}[p(y^*|x^*, \theta)]$$

$$\approx \frac{1}{K} \sum_k^K p(y^*|x^*, \theta_k), \quad \theta_k \sim q(\theta)$$

approximate posterior moments

$$E_{q(\theta)}[F(\theta)]$$

$$\approx \frac{1}{K} \sum_k^K F(\theta_k), \quad \theta_k \sim q(\theta)$$

Computed with Monte Carlo estimates

Only require fast sampling from  $q$ !  
(no need for analytic form of the  $q$  distribution)

Mohamed and Lakshminarayanan. Learning in Implicit Generative Models. arXiv 2016

Li and Liu. Wild Variational Inference. AABI 2016

Huszár. Variational Inference using Implicit Distributions. arXiv 2017

# Implicit Approximate Posteriors

- Two quantities computed in (approximate) Bayesian inference:

approximate Bayesian predictive

$$p(y^*|x^*, D) \approx E_{q(\theta)}[p(y^*|x^*, \theta)]$$

$$\approx \frac{1}{K} \sum_k^K p(y^*|x^*, \theta_k), \quad \theta_k \sim q(\theta)$$

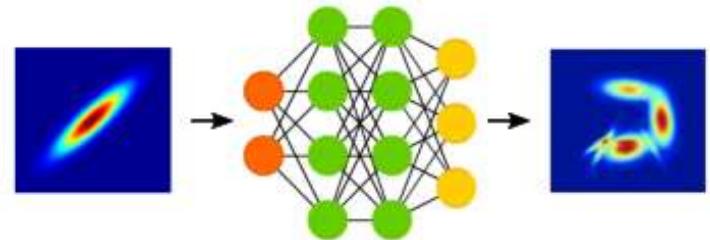
approximate posterior moments

$$E_{q(\theta)}[F(\theta)]$$

$$\approx \frac{1}{K} \sum_k^K F(\theta_k), \quad \theta_k \sim q(\theta)$$

Computed with Monte Carlo estimates

Only require fast sampling from  $q$ !  
(no need for analytic form of the  $q$  distribution)



implicit distributions

Mohamed and Lakshminarayanan. Learning in Implicit Generative Models. arXiv 2016

Li and Liu. Wild Variational Inference. AABI 2016

Huszár. Variational Inference using Implicit Distributions. arXiv 2017

# Implicit Approximate Posteriors

$$L(\phi) = \underbrace{E_{q(\theta)}[\log p(D|\theta)]}_{\text{estimated by Monte Carlo}} - \underbrace{E_{q(\theta)}[\log \frac{p(\theta)}{q(\theta)}]}_{\text{intractable}} \\ (q \text{ density unknown})$$

Mescheder et al. Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks. ICML 2017

Tran et al. Hierarchical Implicit Models and Likelihood-Free Variational Inference. NeurIPS 2017

**Li and Turner. Gradient Estimators for Implicit Models. ICLR 2018**

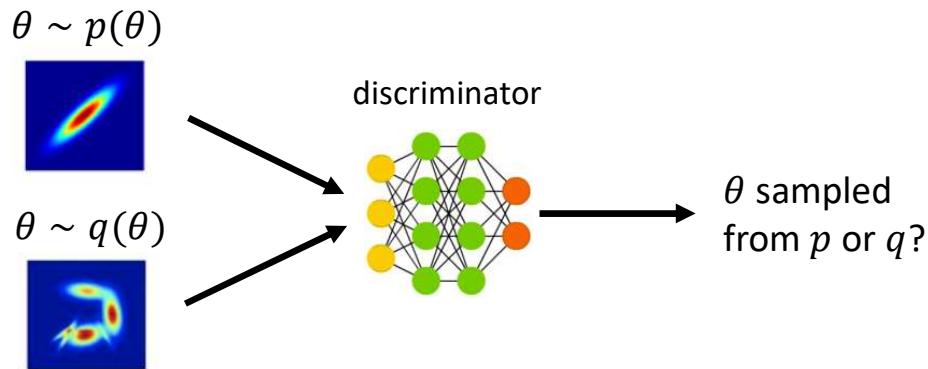
Yin and Zhou. Semi-Implicit Variational Inference. ICML 2018

# Implicit Approximate Posteriors

$$L(\phi) = \underline{E_{q(\theta)}[\log p(D|\theta)]} - E_{q(\theta)}[\log \frac{p(\theta)}{q(\theta)}]$$

estimated by Monte Carlo

Approximated by using a discriminator (AVB):



Mescheder et al. Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks. ICML 2017

Tran et al. Hierarchical Implicit Models and Likelihood-Free Variational Inference. NeurIPS 2017

**Li and Turner.** Gradient Estimators for Implicit Models. ICLR 2018

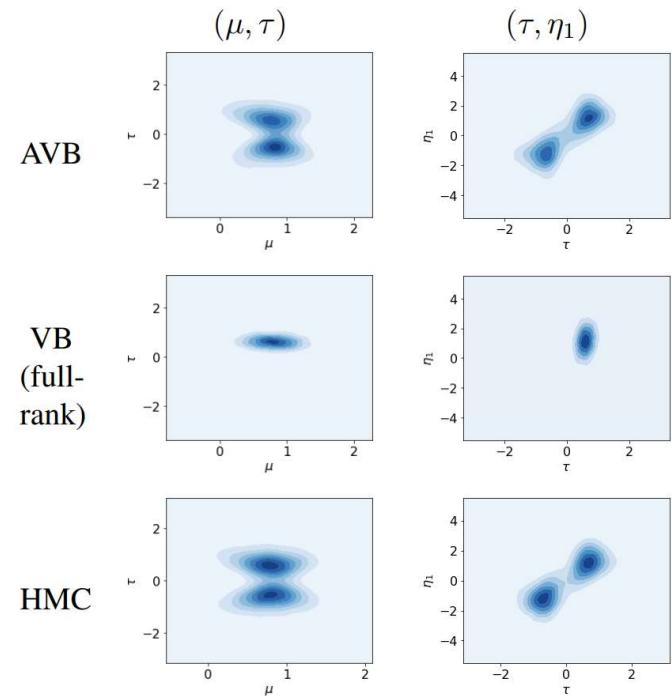
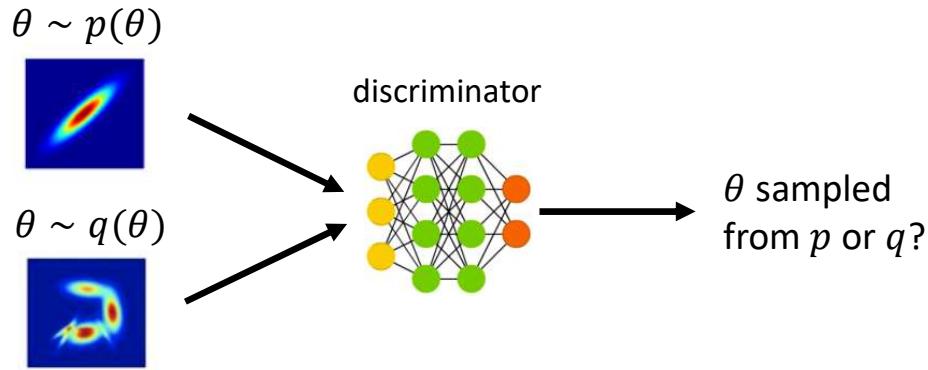
Yin and Zhou. Semi-Implicit Variational Inference. ICML 2018

# Implicit Approximate Posteriors

$$L(\phi) = E_{q(\theta)}[\log p(D|\theta)] - E_{q(\theta)}[\log \frac{p(\theta)}{q(\theta)}]$$

estimated by Monte Carlo

Approximated by using a discriminator (AVB):



Mescheder et al. Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks. ICML 2017

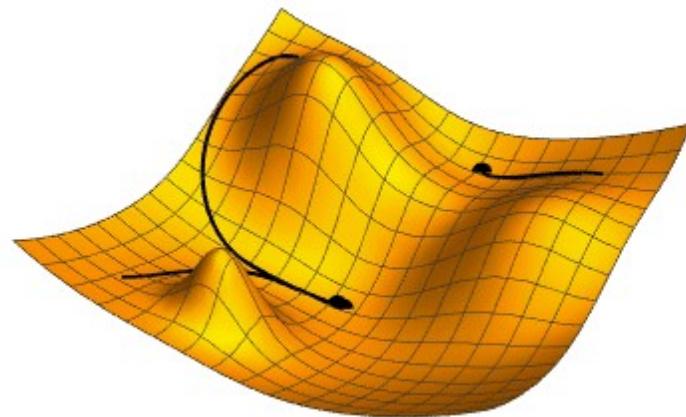
Tran et al. Hierarchical Implicit Models and Likelihood-Free Variational Inference. NeurIPS 2017

Li and Turner. Gradient Estimators for Implicit Models. ICLR 2018

Yin and Zhou. Semi-Implicit Variational Inference. ICML 2018

# Objective Functions

For fitting the approximate posterior



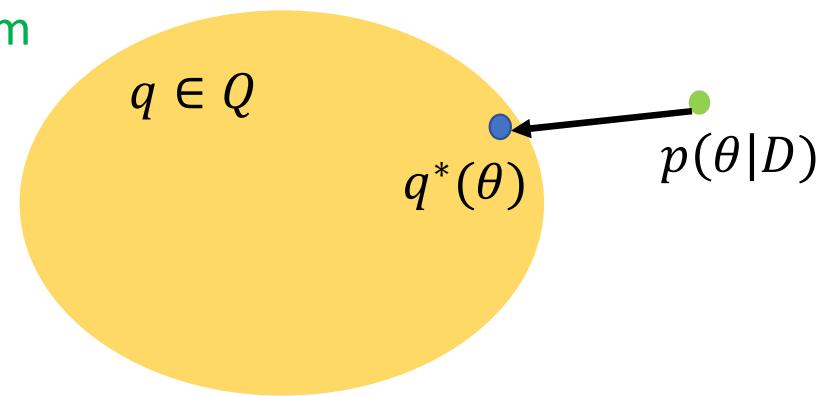
# VI Ingredients

$$L = E_{\theta \sim q_\phi} \left[ \log \frac{p(D, \theta)}{q_\phi(\theta)} \right] = \log p(D) - KL[q_\phi || p]$$

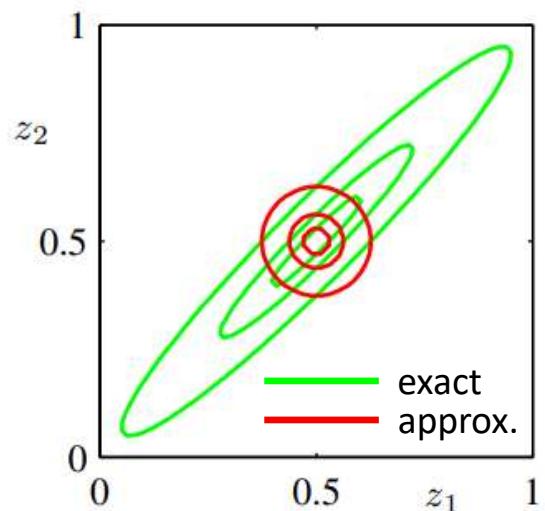
p: your model design

q: your choice of variational distribution, e.g. mean field, flow based

KL: defines the algorithm



# Does It Work?



VI underestimate the uncertainty

# (Rényi) $\alpha$ -Divergence

$$\alpha > 0, \alpha \neq 1$$

$$D_\alpha[p||q] = \frac{1}{\alpha - 1} \log \int p(\theta)^\alpha q^{1-\alpha} d\theta$$

$$\alpha = 1$$

$$D_1[p||q] = \lim_{\alpha \rightarrow 1} D_\alpha(p||q) = KL(p||q)$$

# VI with $\alpha$ -Divergence

ELBO

$$L = E_{\theta \sim q_\phi} \left[ \log \frac{p(D, \theta)}{q_\phi(\theta)} \right] = \log p(D) - \boxed{KL[q_\phi || p]}$$

Variational Rényi bound:

$$L_\alpha = \frac{1}{1-\alpha} E_{\theta \sim q_\phi} \left[ \left( \log \frac{p(D, \theta)}{q_\phi(\theta)} \right)^{1-\alpha} \right] = \log p(D) - \boxed{D_\alpha[q_\phi || p]}$$

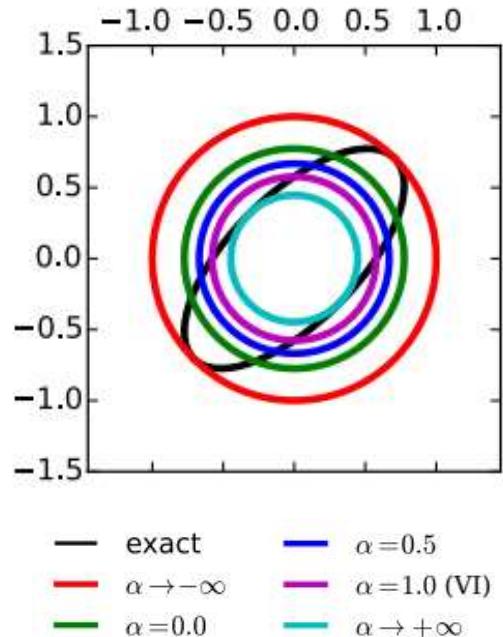
$$\lim_{\alpha \rightarrow 1} L_\alpha = L$$

Li and Turner. Rényi Divergence Variational Inference. NeurIPS 2016

Dieng et al. Variational Inference via  $\chi$ -Upper Bound Minimization. NeurIPS 2017

Minka, Tom. Divergence measures and message passing. Technical report, Microsoft Research, 2005.

# Does It Work?

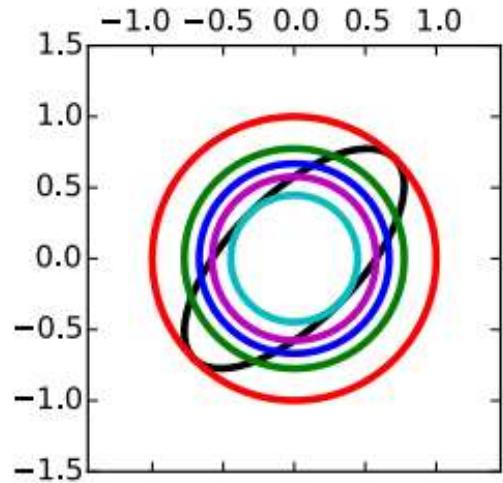


Li and Turner. Rényi Divergence Variational Inference. NeurIPS 2016

Dieng et al. Variational Inference via  $\chi$ -Upper Bound Minimization. NeurIPS 2017

Minka, Tom. Divergence measures and message passing. Technical report, Microsoft Research, 2005.

# Does It Work?



— exact  
—  $\alpha \rightarrow -\infty$   
—  $\alpha = 0.0$   
—  $\alpha = 0.5$   
—  $\alpha = 1.0$  (VI)  
—  $\alpha \rightarrow +\infty$

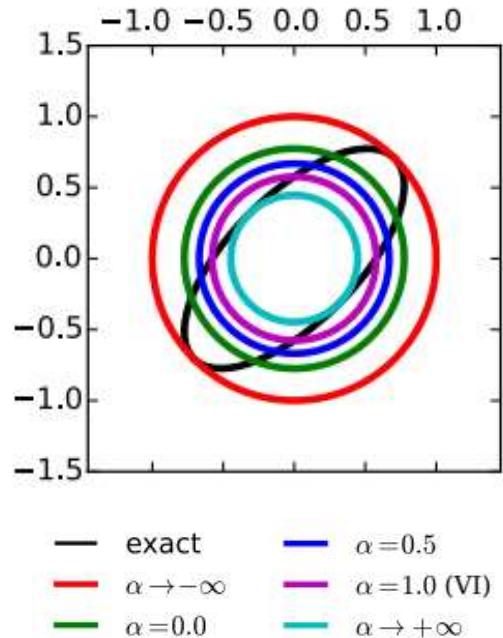
How to choose alpha?

Li and Turner. Rényi Divergence Variational Inference. NeurIPS 2016

Dieng et al. Variational Inference via  $\chi$ -Upper Bound Minimization. NeurIPS 2017

Minka, Tom. Divergence measures and message passing. Technical report, Microsoft Research, 2005.

# Does It Work?



How to choose alpha?

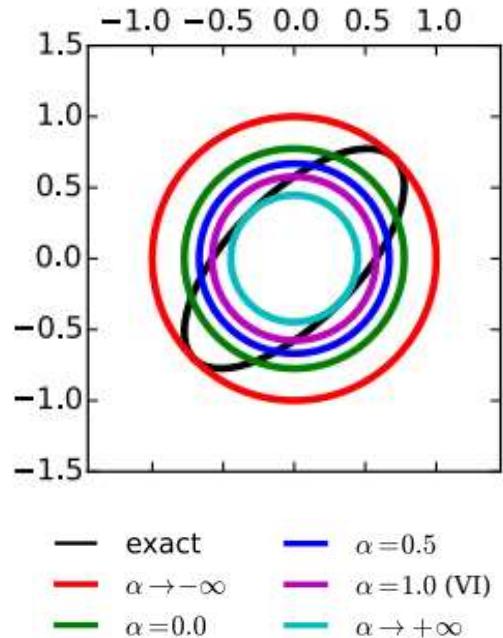
$$L_\alpha = \frac{1}{1-\alpha} E_{\theta \sim q_\phi} \left[ \left( \log \frac{p(D, \theta)}{q_\phi(\theta)} \right)^{1-\alpha} \right]$$

Li and Turner. Rényi Divergence Variational Inference. NeurIPS 2016

Dieng et al. Variational Inference via  $\chi$ -Upper Bound Minimization. NeurIPS 2017

Minka, Tom. Divergence measures and message passing. Technical report, Microsoft Research, 2005.

# Does It Work?



How to choose alpha?

$$L_\alpha = \frac{1}{1-\alpha} E_{\theta \sim q_\phi} \left[ \left( \log \frac{p(D, \theta)}{q_\phi(\theta)} \right)^{1-\alpha} \right]$$

Too small or too big alpha leads to extremely big variances

Li and Turner. Rényi Divergence Variational Inference. NeurIPS 2016

Dieng et al. Variational Inference via  $\chi$ -Upper Bound Minimization. NeurIPS 2017

Minka, Tom. Divergence measures and message passing. Technical report, Microsoft Research, 2005.

# Revisiting Perturbation Theory for VI

$$\log p(x) = \log \left( E_{z \sim q_\lambda} \left[ \frac{p(x, z)}{q_\lambda(z)} \right] \right) = \underbrace{\log \left( E_{z \sim q_\lambda} [e^{-\beta V(x, z)}] \right)}_{V(x, z) \equiv \log q_\lambda(z) - \log p(x, z)} |_{\beta=1}$$

# Revisiting Perturbation Theory for VI

$$\log p(x) = \log \left( E_{z \sim q_\lambda} \left[ \frac{p(x, z)}{q_\lambda(z)} \right] \right) = \log \left( E_{z \sim q_\lambda} [e^{-\beta V(x, z)}] \right) |_{\beta=1}$$

Taylor expansion around  $\beta = 1$  :

$$\begin{aligned} \log p(x) \approx & \boxed{E_{q_\lambda}[-V]} + \frac{1}{2} \left[ (V - E_{q_\lambda}[-V])^2 \right] - \frac{1}{3!} \left[ (V - E_{q_\lambda}[-V])^3 \right] \\ & + \frac{1}{4!} \left[ (V - E_{q_\lambda}[-V])^4 \right] - \dots \end{aligned}$$

# Revisiting Perturbation Theory for VI

$$\log p(x) = \log \left( E_{z \sim q_\lambda} \left[ \frac{p(x, z)}{q_\lambda(z)} \right] \right) = \log \left( E_{z \sim q_\lambda} [e^{-\beta V(x, z)}] \right) |_{\beta=1}$$

Taylor expansion around  $\beta = 1$  :

$$\begin{aligned} \log p(x) \approx & \boxed{E_{q_\lambda}[-V]} + \frac{1}{2} \left[ (V - E_{q_\lambda}[-V])^2 \right] - \frac{1}{3!} \left[ (V - E_{q_\lambda}[-V])^3 \right] \\ & + \frac{1}{4!} \left[ (V - E_{q_\lambda}[-V])^4 \right] - \dots \end{aligned}$$

$$E_{q_\lambda}[-V(x, z)] = E_{q_\lambda}[\log p(x, z) - \log q_\lambda(z)]$$

# Revisiting Perturbation Theory for VI

$$\log p(x) = \log \left( E_{z \sim q_\lambda} \left[ \frac{p(x, z)}{q_\lambda(z)} \right] \right) = \log \left( E_{z \sim q_\lambda} [e^{-\beta V(x, z)}] \right) |_{\beta=1}$$

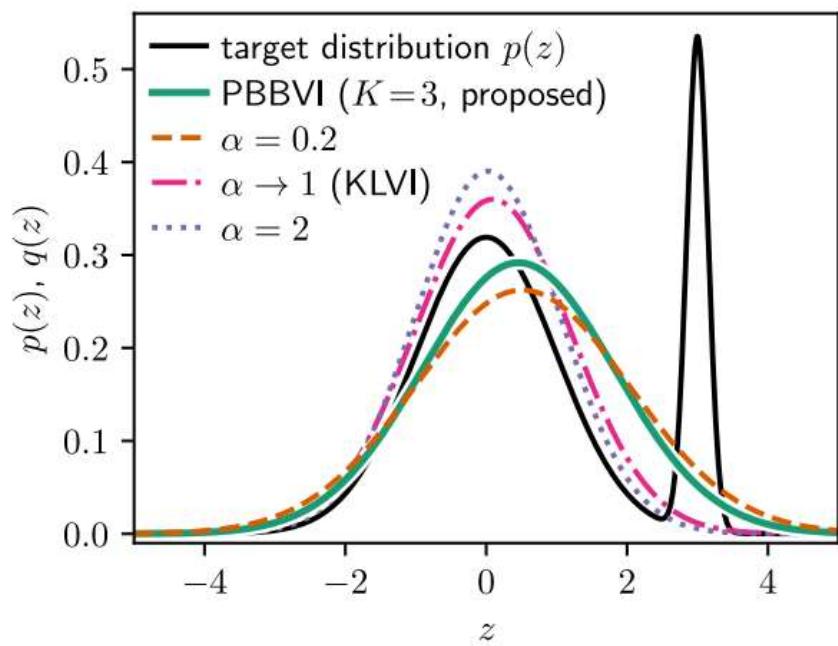
Taylor expansion around  $\beta = 1$  :

$$\begin{aligned} \log p(x) \approx & \boxed{E_{q_\lambda}[-V]} + \frac{1}{2} \left[ (V - E_{q_\lambda}[-V])^2 \right] - \frac{1}{3!} \left[ (V - E_{q_\lambda}[-V])^3 \right] \\ & + \frac{1}{4!} \left[ (V - E_{q_\lambda}[-V])^4 \right] - \dots \end{aligned}$$

$$E_{q_\lambda}[-V(x, z)] = E_{q_\lambda}[\log p(x, z) - \log q_\lambda(z)]$$

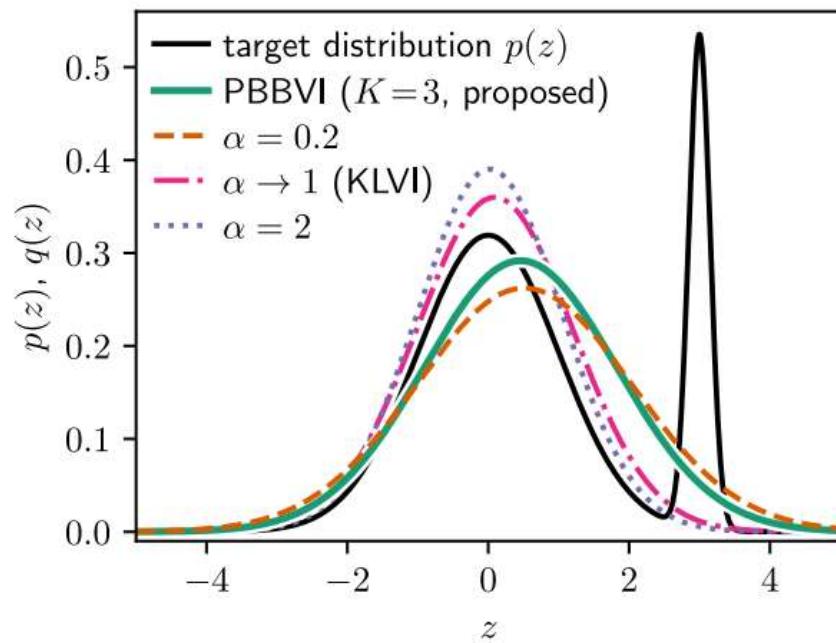
Truncation at any odd number term provides a bound.

# Behaviour of PBBVI



- Better uncertainty estimation than KLVI
- Better bias-variance trade-off comparing to  $\alpha$ -VI

# Behaviour of PBBVI



- Better uncertainty estimation than KLVI
- Better bias-variance trade-off comparing to  $\alpha$ -VI

Where do we truncate?  
Is it flexible enough?

# F-Divergence

$$D_f[p||q_\phi] = E_{\theta \sim q_\phi} [f\left(\frac{p(\theta)}{q_\phi(\theta)}\right) - f(1)]$$

# F-Divergence

$$D_f[p||q_\phi] = E_{\theta \sim q_\phi} [f\left(\frac{p(\theta)}{q_\phi(\theta)}\right) - f(1)]$$

$$f(t) = -\log t \quad \longrightarrow \quad KL(q||p)$$

$$f(t) = t \log t \quad \longrightarrow \quad KL(p||q)$$

$$f(t) = \frac{t^\alpha}{\alpha(\alpha - 1)} \quad \longrightarrow \quad D_\alpha(p||q)$$

Wang et.al. Variational Inference with Tail-adaptive f-Divergence. NeurIPS 2018  
Wan et.al. f-Divergence Variational Inference. NeurIPS 2020

# Integral Probability Metric (IPM)

- Using a test function to describe difference:

$$D[q(z), p(z|x)] = \sup_{f \in F} |E_{q(z)}[f(z)] - E_{p(z|x)}[f(z)]|$$

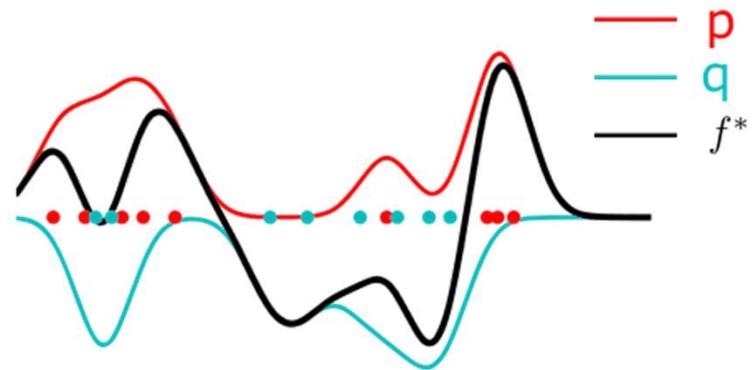


Figure adapted, source: Dougal Sutherland

Gorham and Mackey. Measuring Sample Quality with Stein's Method. NeurIPS 2015

Ranganath et al. Operator Variational Inference. NeurIPS 2016

Liu and Wang. Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm. NeurIPS 2016

# Integral Probability Metric (IPM)

- Using a test function to describe difference:

$$D[q(z), p(z|x)] = \sup_{f \in F} |E_{q(z)}[f(z)] - E_{p(z|x)}[f(z)]|$$

- Stein discrepancy: only requires  $z \sim q(z)$  and  $\nabla_z \log p(z|x) = \nabla_z \log p(z, x)$

$$S[q(z), p(z|x)] = \sup_{f \in F_q} |E_{q(z)}[\nabla_z \log p(z, x)^\top f(z) + \nabla_z^\top f(z)]|$$

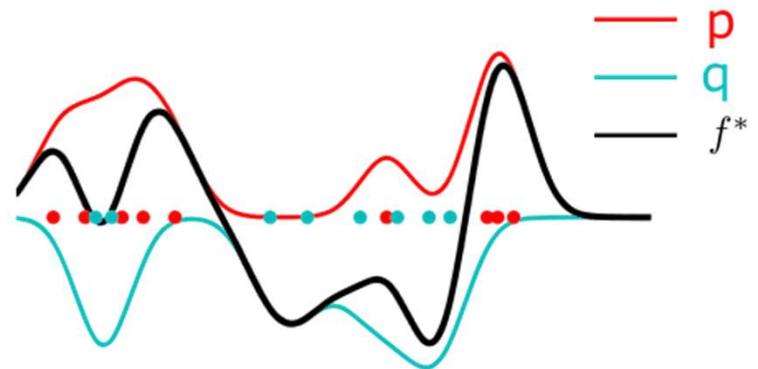


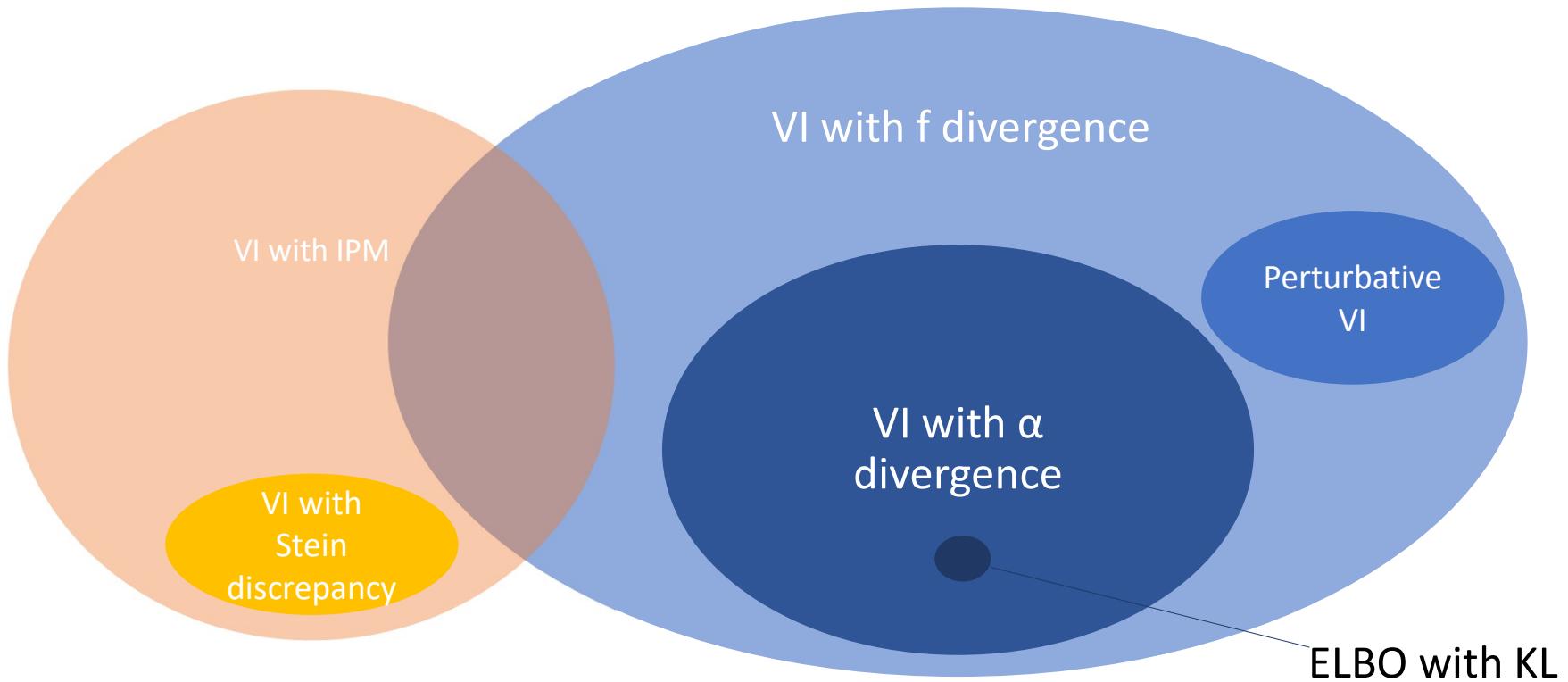
Figure adapted, source: Dougal Sutherland

Gorham and Mackey. Measuring Sample Quality with Stein's Method. NeurIPS 2015

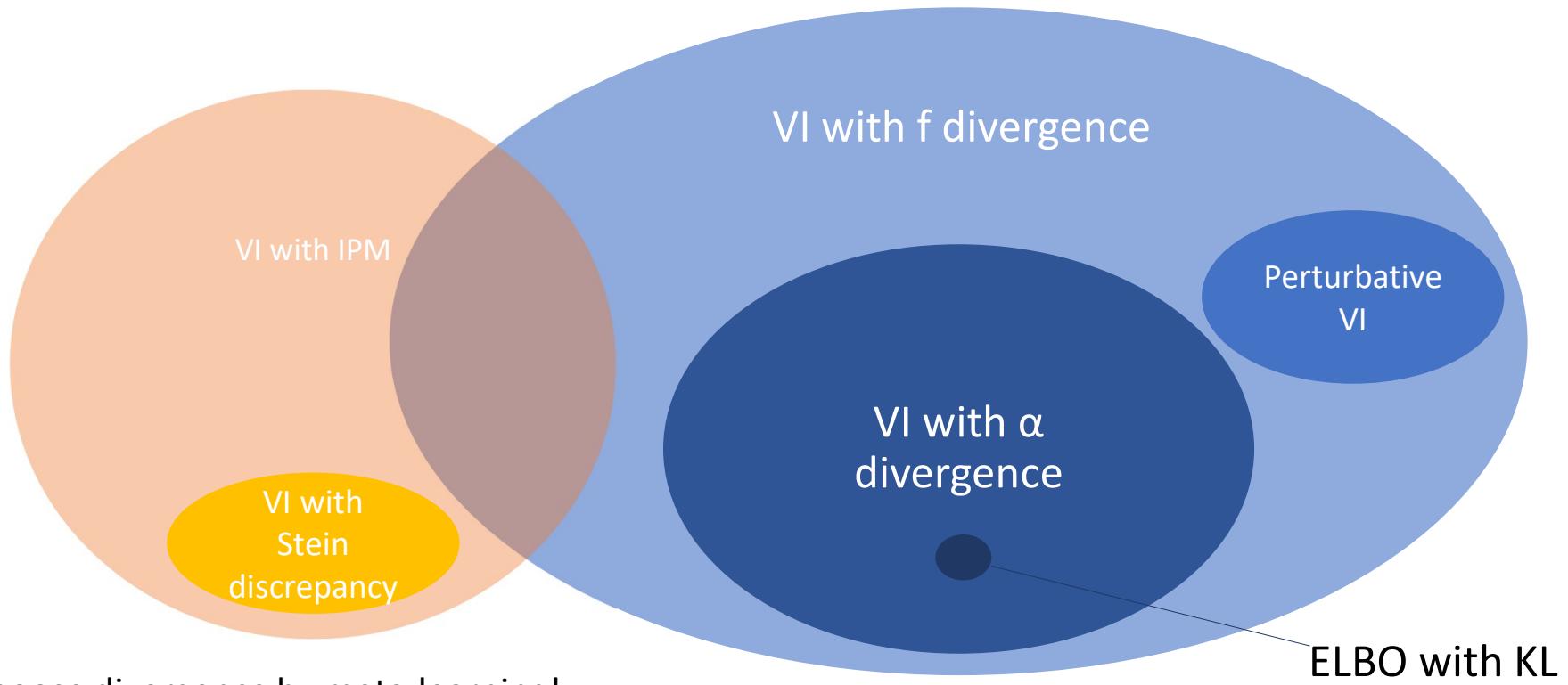
Ranganath et al. Operator Variational Inference. NeurIPS 2016

Liu and Wang. Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm. NeurIPS 2016

# Looking Back



# How to Choose the Inference Algorithm?



Choose divergence by meta-learning!

Zhang et al. Meta-Learning for Variational Inference. AABI 2019

# Improved Monte Carlo Bounds

- Importance weighted auto-encoder (IWAE) bound:

$$L_K(\phi) = E_{z_1, \dots, z_K \sim q(z)} \left[ \log \frac{1}{K} \sum_{k=1}^K \frac{p(x, z_k)}{q(z_k)} \right]$$

Importance sampling estimate of  $p(x)$

Burda et al. Importance Weighted Auto-encoders. ICLR 2016  
Naesseth et al. Variational Sequential Monte Carlo. AISTATS 2018  
Maddison et al. Filtering Variational Objectives. NeurIPS 2017  
Le et al. Auto-encoding Sequential Monte Carlo. ICLR 2018  
Masrani et al. The Thermodynamic Variational Objective. NeurIPS 2019

# Improved Monte Carlo Bounds

- Importance weighted auto-encoder (IWAE) bound:

$$L_K(\phi) = E_{z_1, \dots, z_K \sim q(z)} \left[ \log \frac{1}{K} \sum_{k=1}^K \frac{p(x, z_k)}{q(z_k)} \right]$$

Importance sampling estimate of  $p(x)$

$\log p(x)$  

$L(\phi)$    
 $(K = 1)$

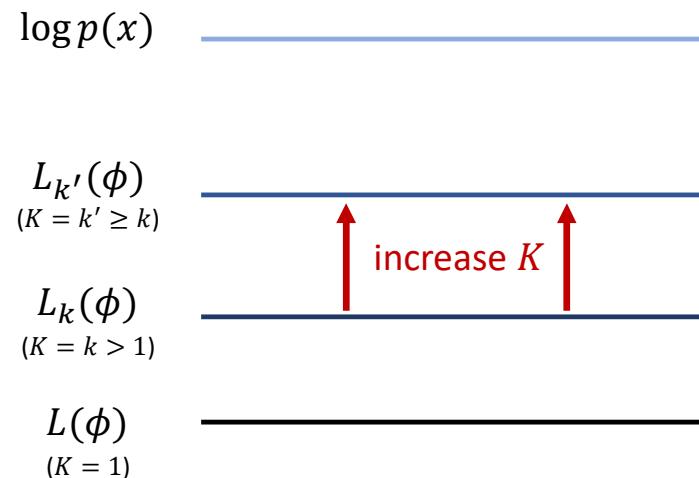
Burda et al. Importance Weighted Auto-encoders. ICLR 2016  
Naesseth et al. Variational Sequential Monte Carlo. AISTATS 2018  
Maddison et al. Filtering Variational Objectives. NeurIPS 2017  
Le et al. Auto-encoding Sequential Monte Carlo. ICLR 2018  
Masrani et al. The Thermodynamic Variational Objective. NeurIPS 2019

# Improved Monte Carlo Bounds

- Importance weighted auto-encoder (IWAE) bound:

$$L_K(\phi) = E_{z_1, \dots, z_K \sim q(z)} \left[ \log \frac{1}{K} \sum_{k=1}^K \frac{p(x, z_k)}{q(z_k)} \right]$$

Importance sampling estimate of  $p(x)$



Burda et al. Importance Weighted Auto-encoders. ICLR 2016

Naesseth et al. Variational Sequential Monte Carlo. AISTATS 2018

Maddison et al. Filtering Variational Objectives. NeurIPS 2017

Le et al. Auto-encoding Sequential Monte Carlo. ICLR 2018

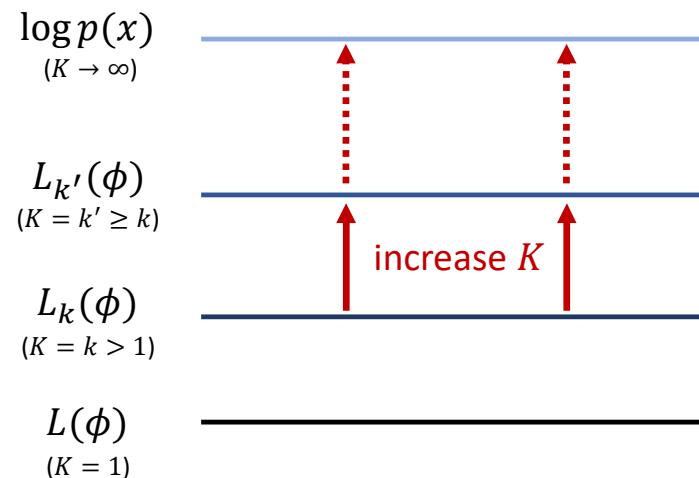
Masrani et al. The Thermodynamic Variational Objective. NeurIPS 2019

# Improved Monte Carlo Bounds

- Importance weighted auto-encoder (IWAE) bound:

$$L_K(\phi) = E_{z_1, \dots, z_K \sim q(z)} \left[ \log \frac{1}{K} \sum_{k=1}^K \frac{p(x, z_k)}{q(z_k)} \right]$$

Importance sampling estimate of  $p(x)$



Burda et al. Importance Weighted Auto-encoders. ICLR 2016

Naesseth et al. Variational Sequential Monte Carlo. AISTATS 2018

Maddison et al. Filtering Variational Objectives. NeurIPS 2017

Le et al. Auto-encoding Sequential Monte Carlo. ICLR 2018

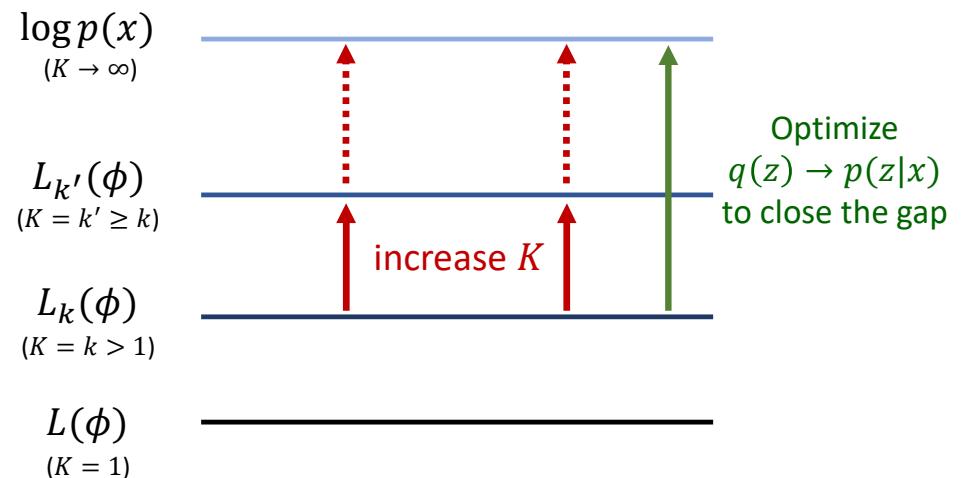
Masrani et al. The Thermodynamic Variational Objective. NeurIPS 2019

# Improved Monte Carlo Bounds

- Importance weighted auto-encoder (IWAE) bound:

$$L_K(\phi) = E_{z_1, \dots, z_K \sim q(z)} \left[ \log \frac{1}{K} \sum_{k=1}^K \frac{p(x, z_k)}{q(z_k)} \right]$$

Importance sampling estimate of  $p(x)$



Burda et al. Importance Weighted Auto-encoders. ICLR 2016

Naesseth et al. Variational Sequential Monte Carlo. AISTATS 2018

Maddison et al. Filtering Variational Objectives. NeurIPS 2017

Le et al. Auto-encoding Sequential Monte Carlo. ICLR 2018

Masrani et al. The Thermodynamic Variational Objective. NeurIPS 2019

# Improved Monte Carlo Bounds

- Constructing lower-bounds from an estimator  $R$  of the marginal:

$$E_{q(h)}[R(h, x)] = p(x) \Rightarrow \underline{E_{q(h)}[\log R(h, x)] \leq \log p(x)}$$

Jensen's inequality

# Improved Monte Carlo Bounds

- Constructing lower-bounds from an estimator  $R$  of the marginal:

$$E_{q(h)}[R(h, x)] = p(x) \Rightarrow \underline{E_{q(h)}[\log R(h, x)] \leq \log p(x)}$$

- Variational lower-bound:  $h = z$ ,  $R(z, x) = \frac{p(x, z)}{q(z)}$  Jensen's inequality
- IWAE bound:  $h = (z_1, \dots, z_K)$ ,  $R(h, x) = \frac{1}{K} \sum_{k=1}^K \frac{p(x, z_k)}{q(z_k)}$

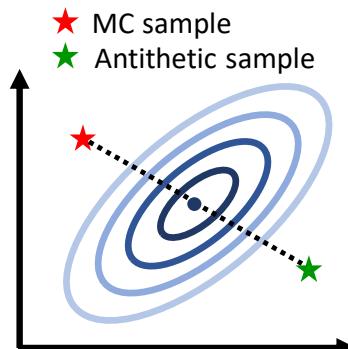
# Improved Monte Carlo Bounds

- Constructing lower-bounds from an estimator  $R$  of the marginal:

$$E_{q(h)}[R(h, x)] = p(x) \Rightarrow \underline{E_{q(h)}[\log R(h, x)] \leq \log p(x)}$$

- Variational lower-bound:  $h = z$ ,  $R(z, x) = \frac{p(x, z)}{q(z)}$  Jensen's inequality
- IWAE bound:  $h = (z_1, \dots, z_K)$ ,  $R(h, x) = \frac{1}{K} \sum_{k=1}^K \frac{p(x, z_k)}{q(z_k)}$
- Fit  $q$  using existing Monte Carlo estimators of  $p(x)$ 
  - Example: antithetic sampling with Gaussian  $q(z)$ :

$$R(z, x) = \frac{p(x, z) + p(x, T(z))}{2q(z)}, \quad T(z) = \mu_q - (z - \mu_q)$$



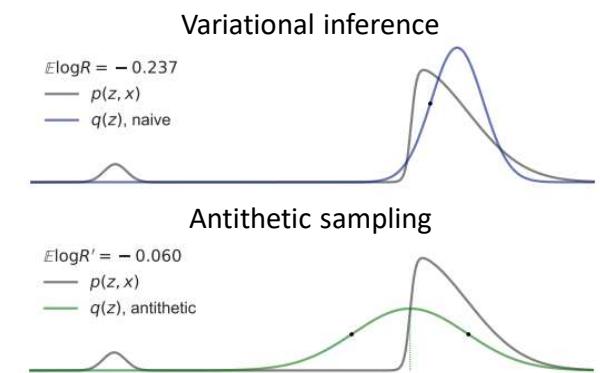
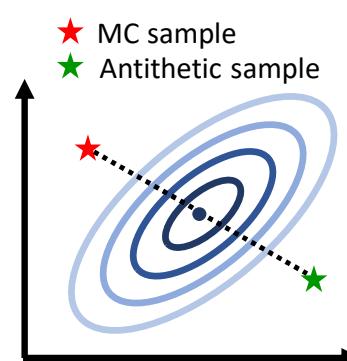
# Improved Monte Carlo Bounds

- Constructing lower-bounds from an estimator  $R$  of the marginal:

$$E_{q(h)}[R(h, x)] = p(x) \Rightarrow \underline{E_{q(h)}[\log R(h, x)] \leq \log p(x)}$$

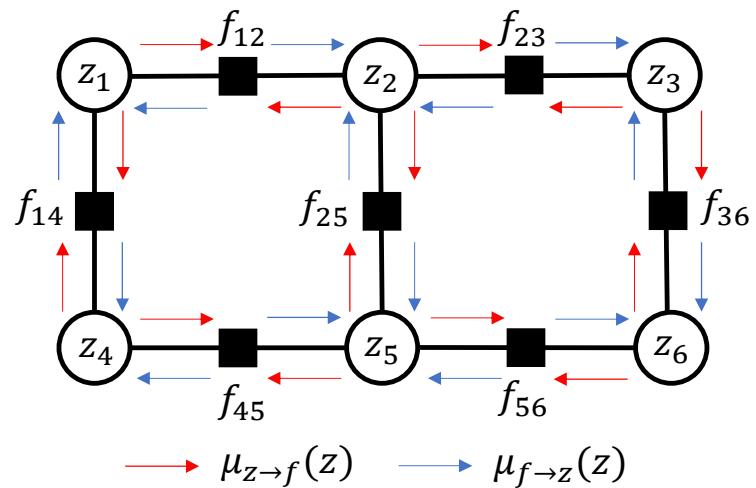
- Variational lower-bound:  $h = z$ ,  $R(z, x) = \frac{p(x, z)}{q(z)}$
- IWAE bound:  $h = (z_1, \dots, z_K)$ ,  $R(h, x) = \frac{1}{K} \sum_{k=1}^K \frac{p(x, z_k)}{q(z_k)}$
- Fit  $q$  using existing Monte Carlo estimators of  $p(x)$ 
  - Example: antithetic sampling with Gaussian  $q(z)$ :

$$R(z, x) = \frac{p(x, z) + p(x, T(z))}{2q(z)}, \quad T(z) = \mu_q - (z - \mu_q)$$



# Free-energy as an Objective

- Bethe free-energy & message passing:



- Both  $q$  and the inference algorithm are defined by the *factor graph*
- Optimal  $q$  achieved at the fixed point of the *Bethe free energy*

Wainwright and Jordan. Graphical Models, Exponential Families, and Variational Inference. 2008.

Li and Turner. A Unifying Approximate Inference Framework from Variational Free Energy Relaxation. AABI 2016

# Landscape of Advances

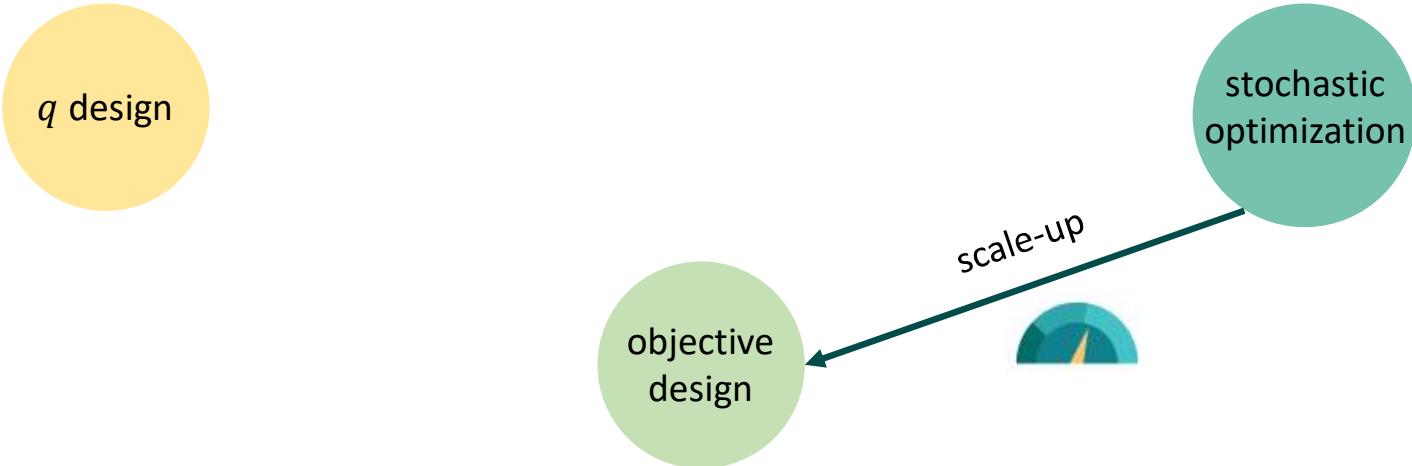


e.g. mean-field:  $q(\theta) = \prod_i q(\theta_i)$

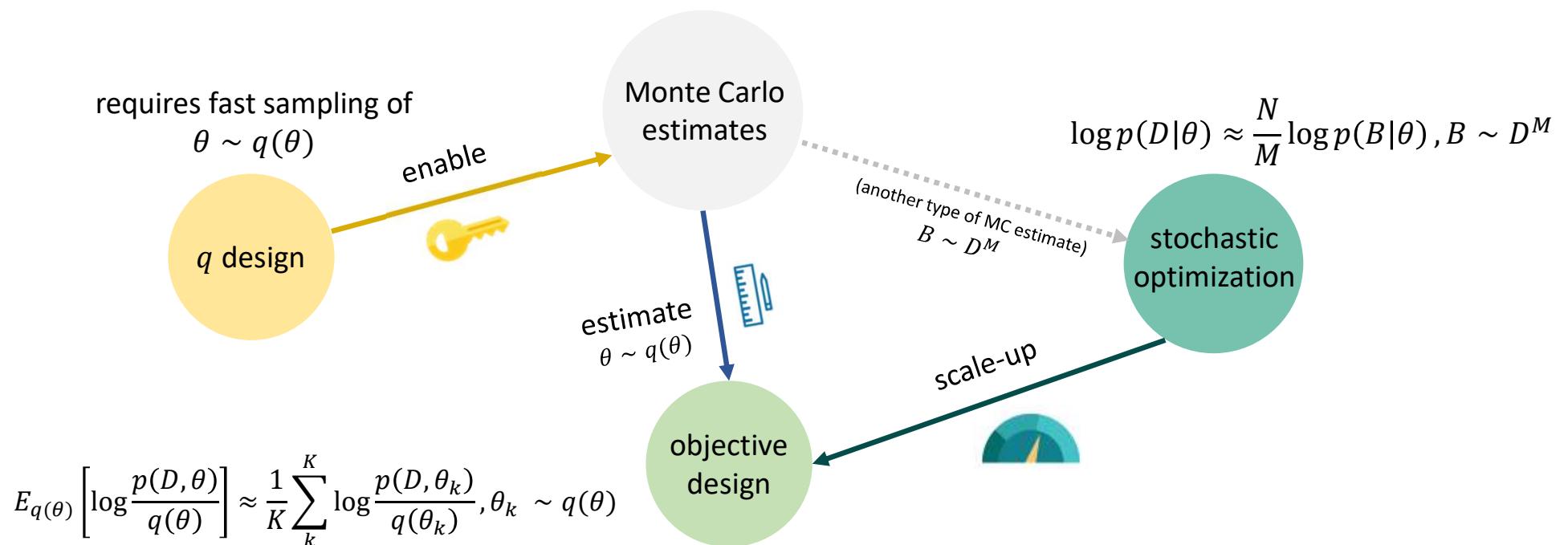


variational lower-bound:  
$$L(\phi) = E_{q(\theta)}[\log p(D|\theta)] - KL[q(\theta)||p(\theta)]$$

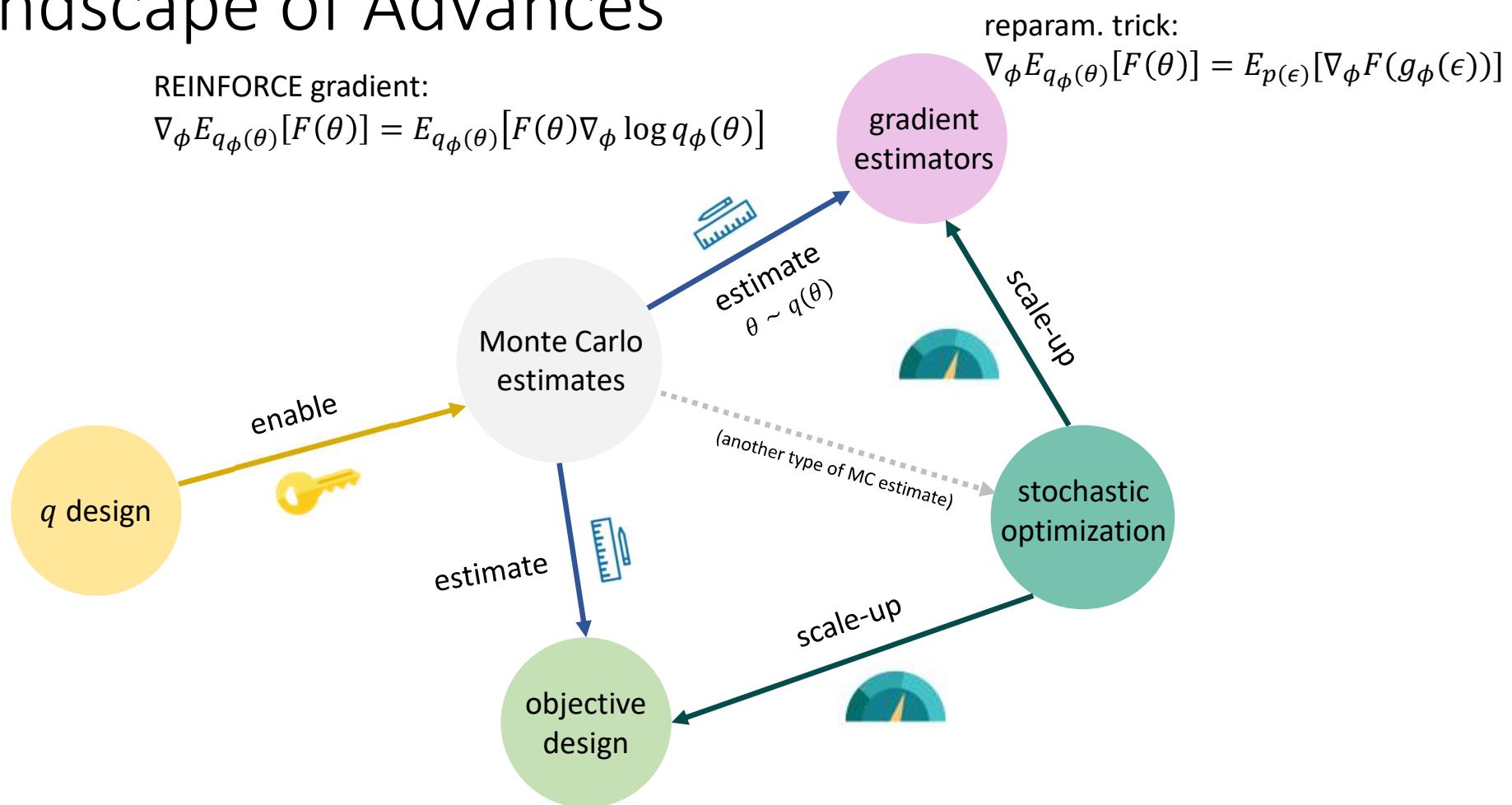
# Landscape of Advances



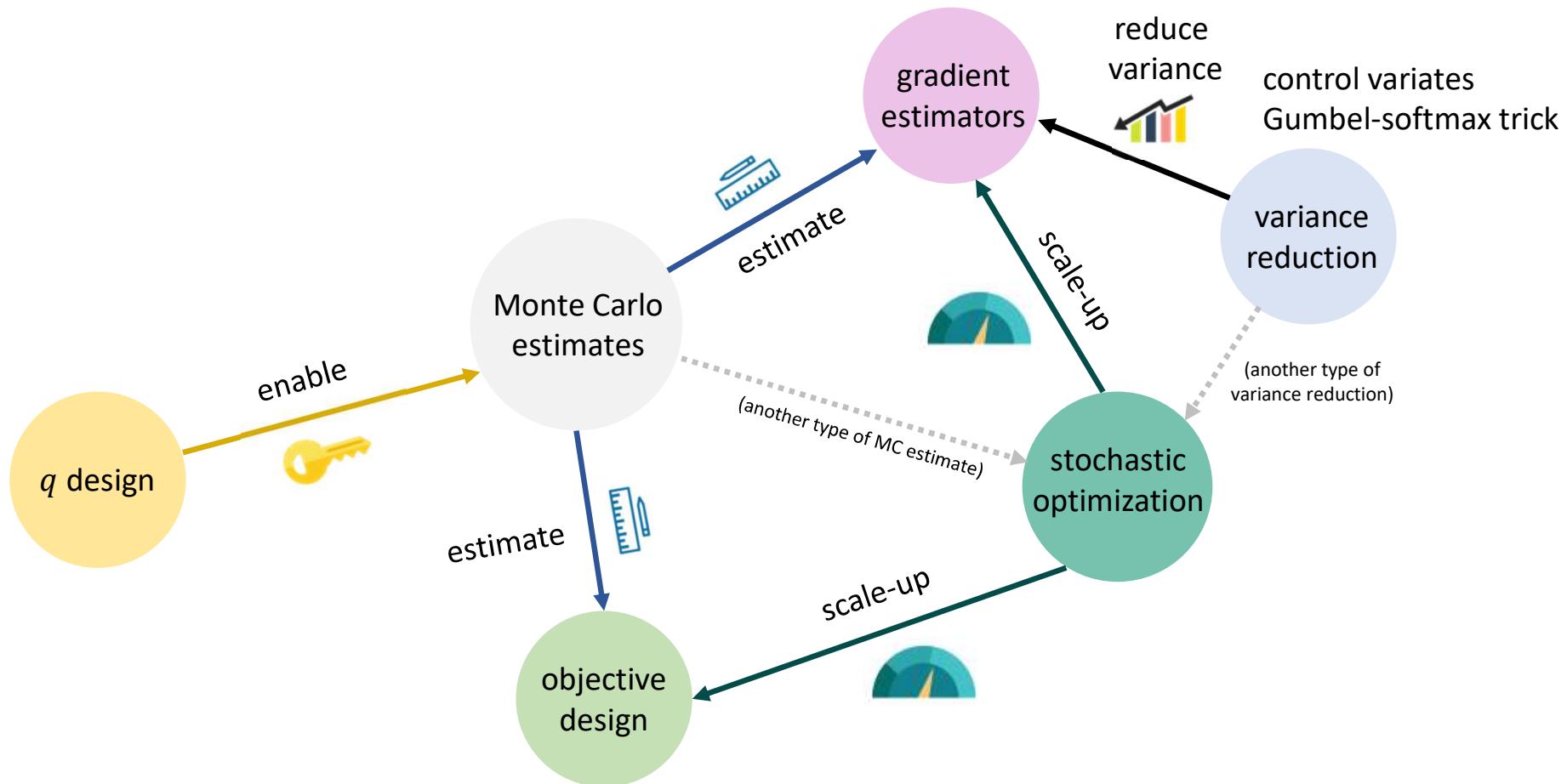
# Landscape of Advances



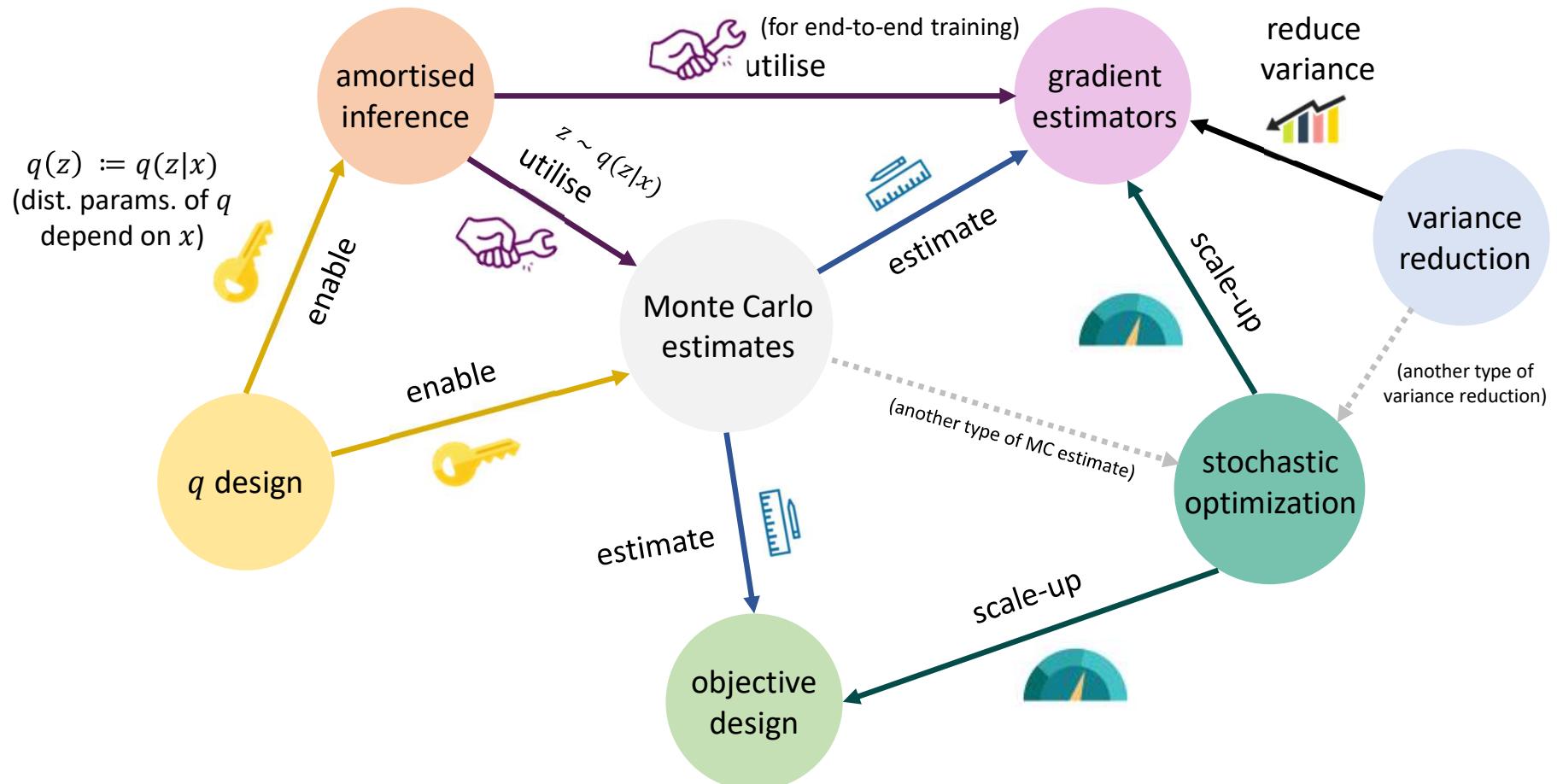
# Landscape of Advances



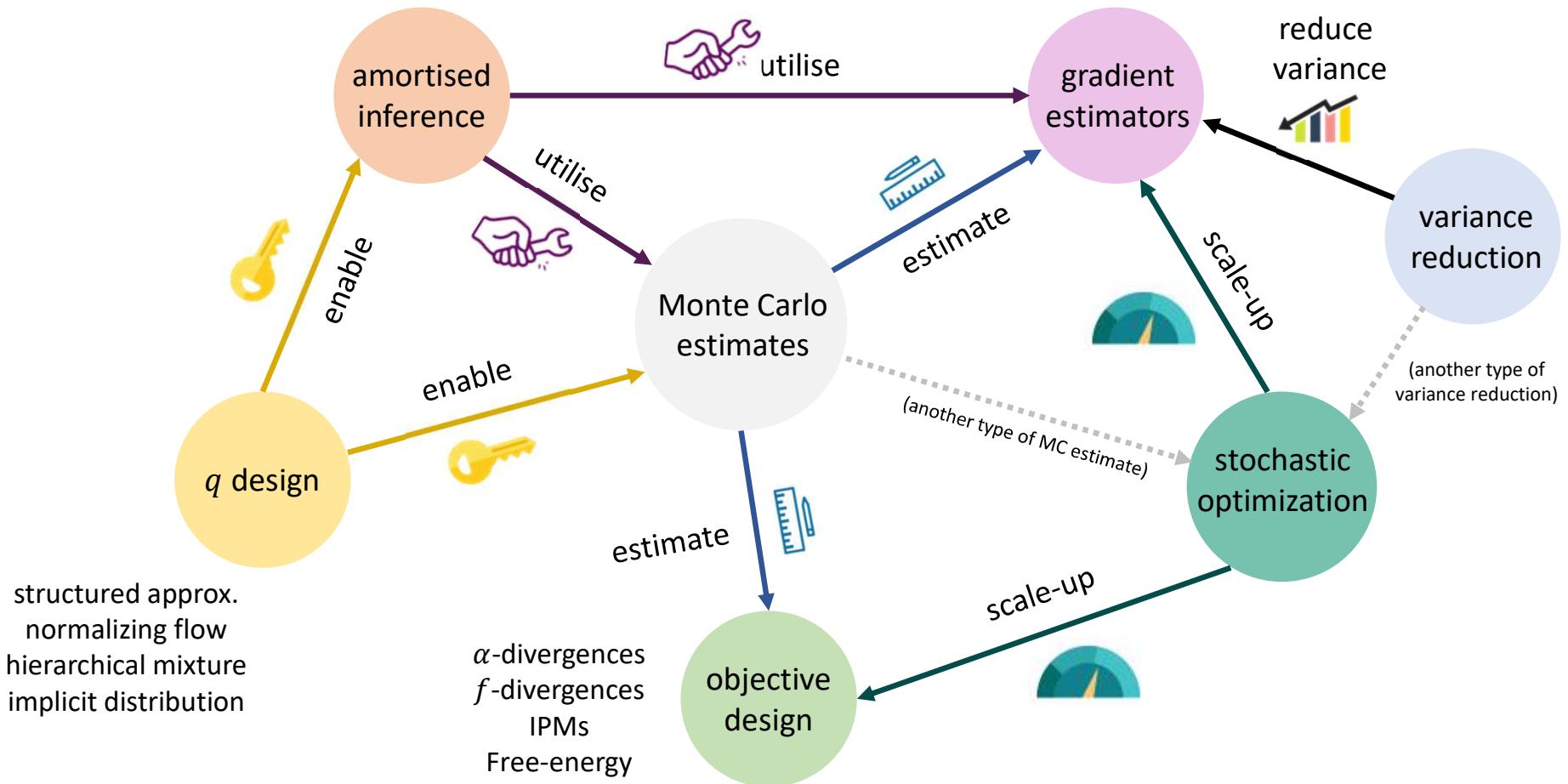
# Landscape of Advances



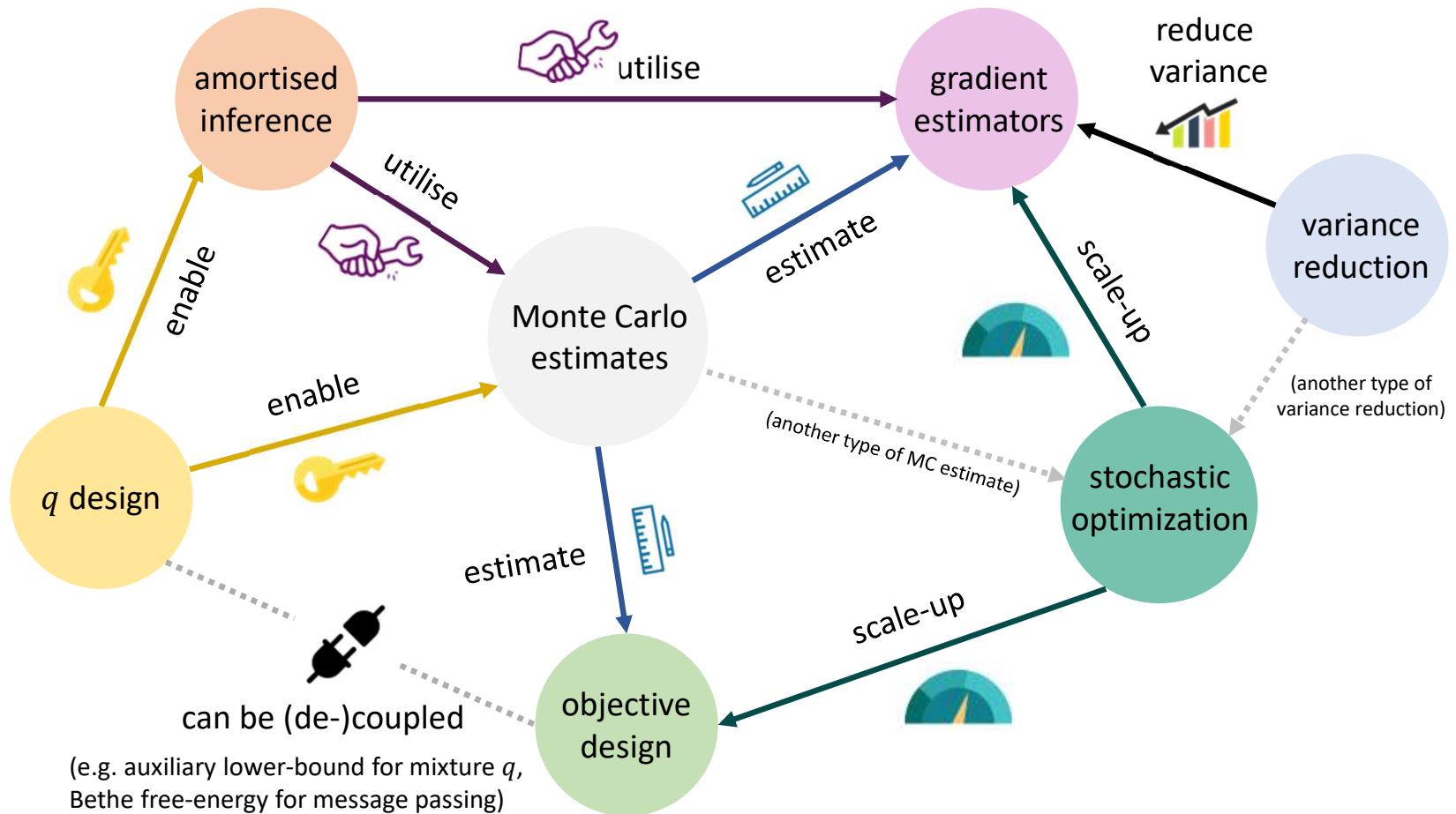
# Landscape of Advances



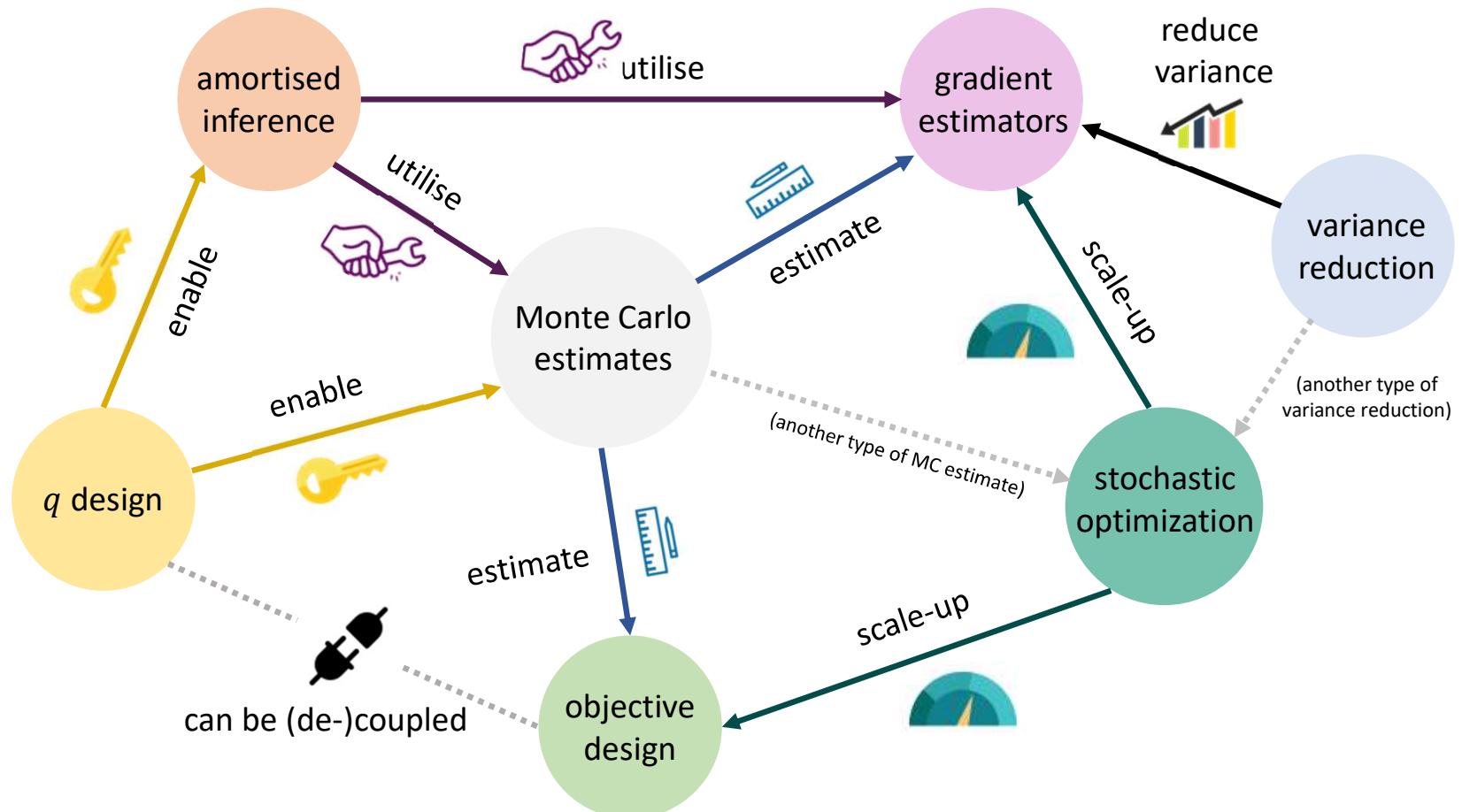
# Landscape of Advances

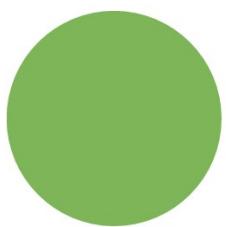


# Landscape of Advances



# Landscape of Advances





# Part III: Applications

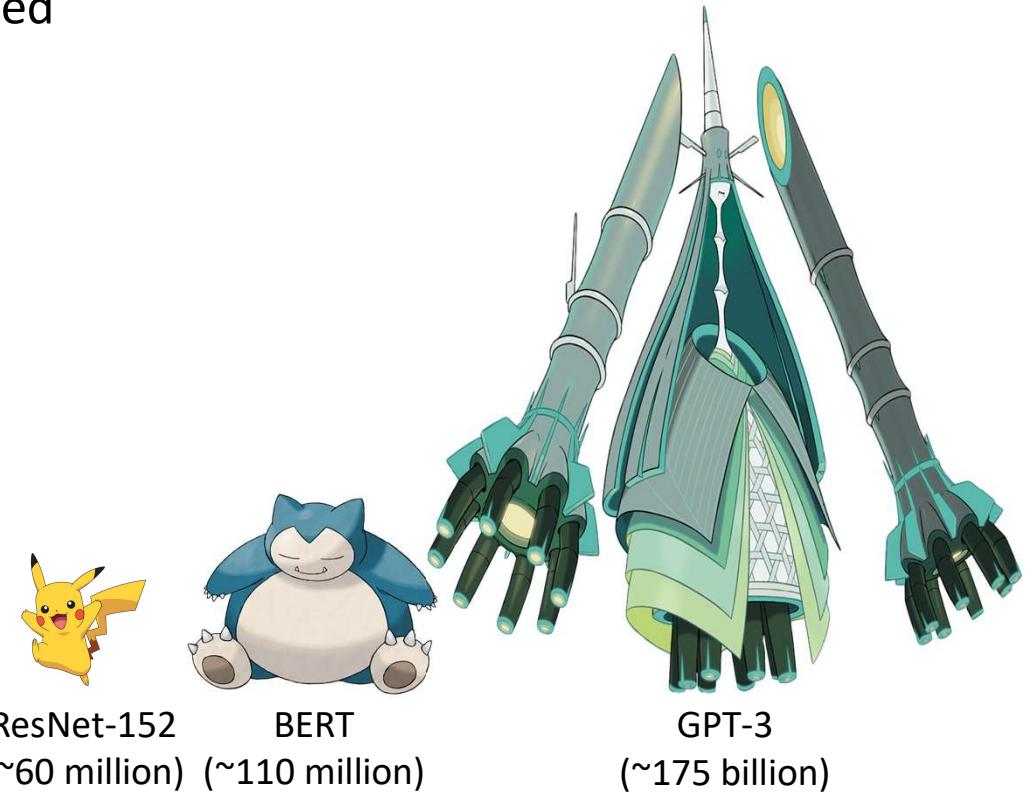
- Bayesian neural networks
- Generative models for decision making
- Future directions

# Why Estimating Uncertainty in DL?

- Models are often over-parameterised
  - E.g. BERT, GPT-3 in NLP
  - E.g. ResNet-152 for vision tasks

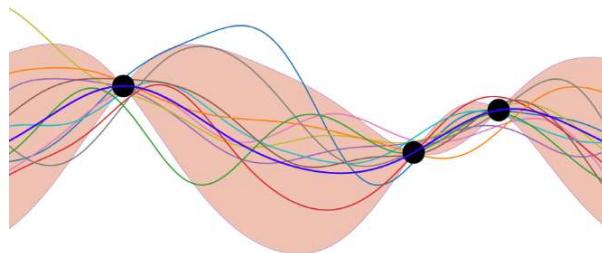
# Why Estimating Uncertainty in DL?

- Models are often over-parameterised
  - E.g. BERT, GPT-3 in NLP
  - E.g. ResNet-152 for vision tasks



# Why Estimating Uncertainty in DL?

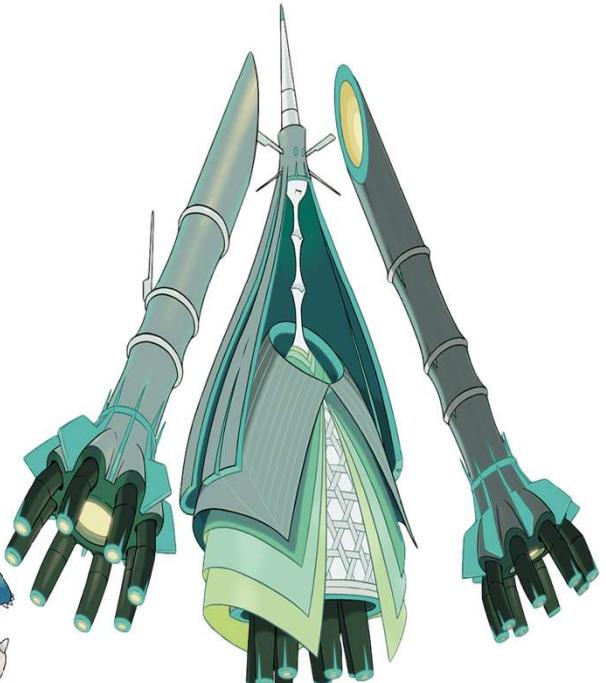
- Models are often over-parameterised
  - E.g. BERT, GPT-3 in NLP
  - E.g. ResNet-152 for vision tasks
- Multiple parameter settings can fit the same data
  - They might provide different predictions on test data



ResNet-152  
(~60 million)



BERT



GPT-3  
(~175 billion)

# Why Estimating Uncertainty in DL?

- Critical tasks need uncertainty estimates to assist decision making
  - Inform end users when uncertain, for safe decision making



Healthcare AI

# Why Estimating Uncertainty in DL?

- Critical tasks need uncertainty estimates to assist decision making
  - Inform end users when uncertain, for safe decision making



Healthcare AI

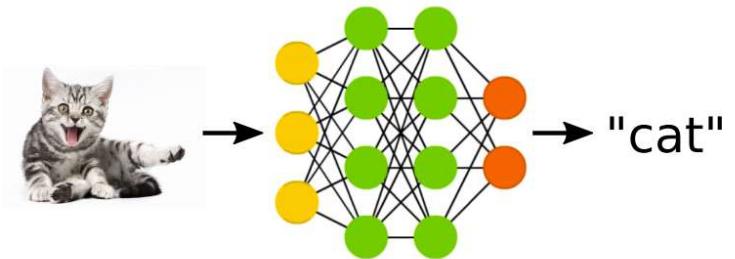


Autonomous driving

# Bayesian Neural Network (BNN) 101

Classifying different types of animals:

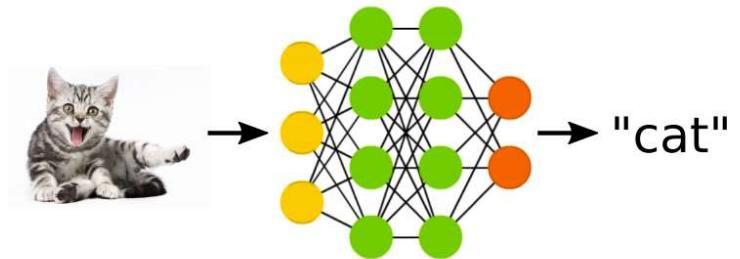
- $x$ : input image;  $y$ : output label
- Build a neural network with parameters  $\theta$ :  
$$p(y|x, \theta) = \text{softmax}(f_\theta(x))$$



# Bayesian Neural Network (BNN) 101

Classifying different types of animals:

- $x$ : input image;  $y$ : output label
- Build a neural network with parameters  $\theta$ :  
$$p(y|x, \theta) = \text{softmax}(f_\theta(x))$$



A typical neural network (with non-linearity  $g(\cdot)$ ):

$$f_\theta(x) = W^L g(W^{L-1} g(\dots g(W^1 x + b^1) + b^{L-1}) + b^L,$$

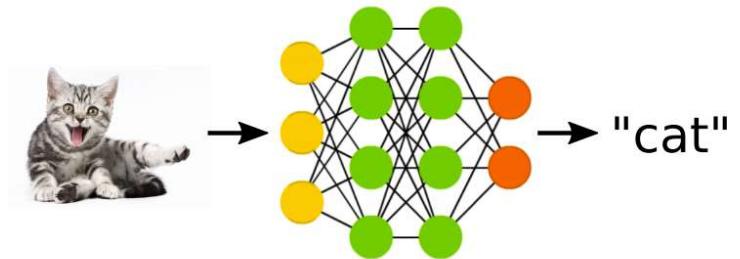
$$h^l = g(W^l h^{l-1} + b^l), h^1 = g(W^1 x + b^1).$$

Neural network parameters:  $\theta = \{W^l, b^l\}_{l=1}^L$

# Bayesian Neural Network (BNN) 101

Classifying different types of animals:

- $x$ : input image;  $y$ : output label
- Build a neural network with parameters  $\theta$ :  
 $p(y|x, \theta) = \text{softmax}(f_\theta(x))$



Typical deep learning solution:

- Optimize  $\theta$  to obtain a point estimates (MLE):

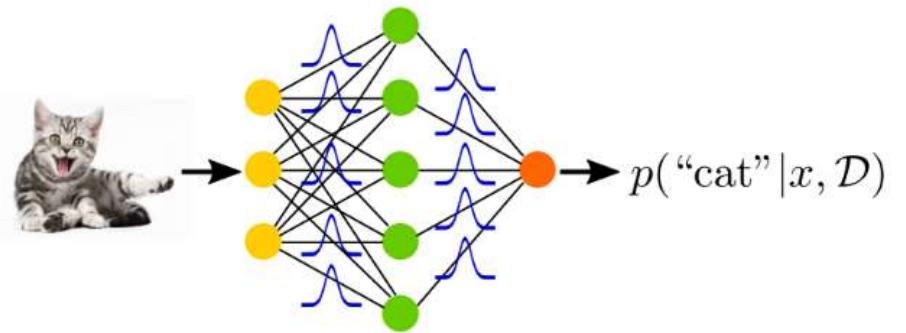
$$\begin{aligned}\theta^* &= \text{argmax } \log p(D | \theta), \\ \log p(D | \theta) &= \sum_{n=1}^N \log p(y_n | x_n, \theta), D = \{(x_n, y_n)\}_{n=1}^N\end{aligned}$$

- Prediction: using  $p(y^* | x^*, \theta^*)$

# Bayesian Neural Network (BNN) 101

Classifying different types of animals:

- $x$ : input image;  $y$ : output label
- Build a neural network with parameters  $\theta$ :  
 $p(y|x, \theta) = \text{softmax}(f_\theta(x))$



**Bayesian solution:**

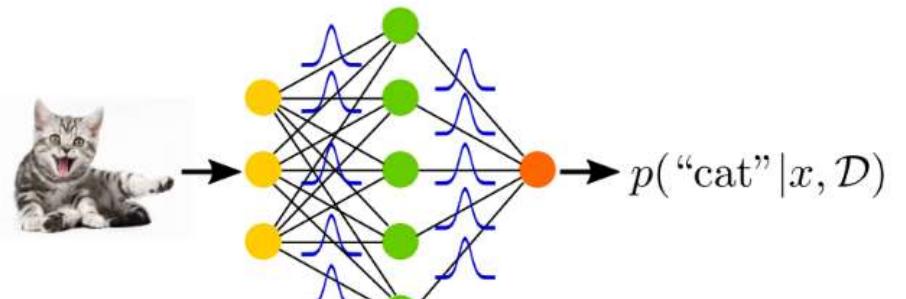
- Put a prior  $p(\theta)$  on network parameters  $\theta$ , e.g. Gaussian prior  
$$p(\theta) = N(\theta; 0, \sigma^2 I)$$
- Compute the posterior distribution  $p(\theta | D)$ :  
$$p(\theta | D) \propto p(D | \theta) p(\theta)$$
- Bayesian predictive inference:

$$p(y^* | x^*, D) = E_{p(\theta | D)}[p(y^* | x^*, \theta)]$$

# Bayesian Neural Network (BNN) 101

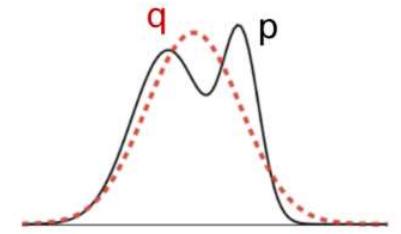
Classifying different types of animals:

- $x$ : input image;  $y$ : output label
- Build a neural network with parameters  $\theta$ :  
 $p(y|x, \theta) = \text{softmax}(f_\theta(x))$



**Approximate (Bayesian) inference solution:**

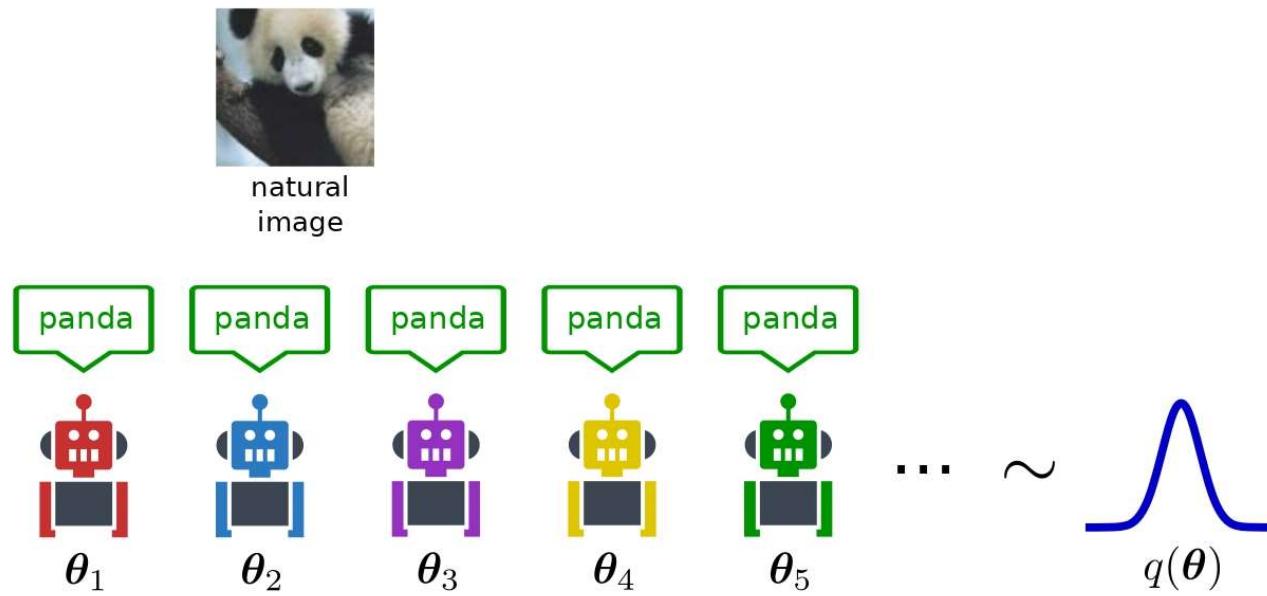
- Exact posterior intractable, use approximate posterior:  
$$q(\theta) \approx p(\theta | D)$$
- Approximate Bayesian predictive inference:  
$$p(y^* | x^*, D) \approx E_{q(\theta)}[p(y^* | x^*, \theta)]$$



- Monte Carlo approximation:

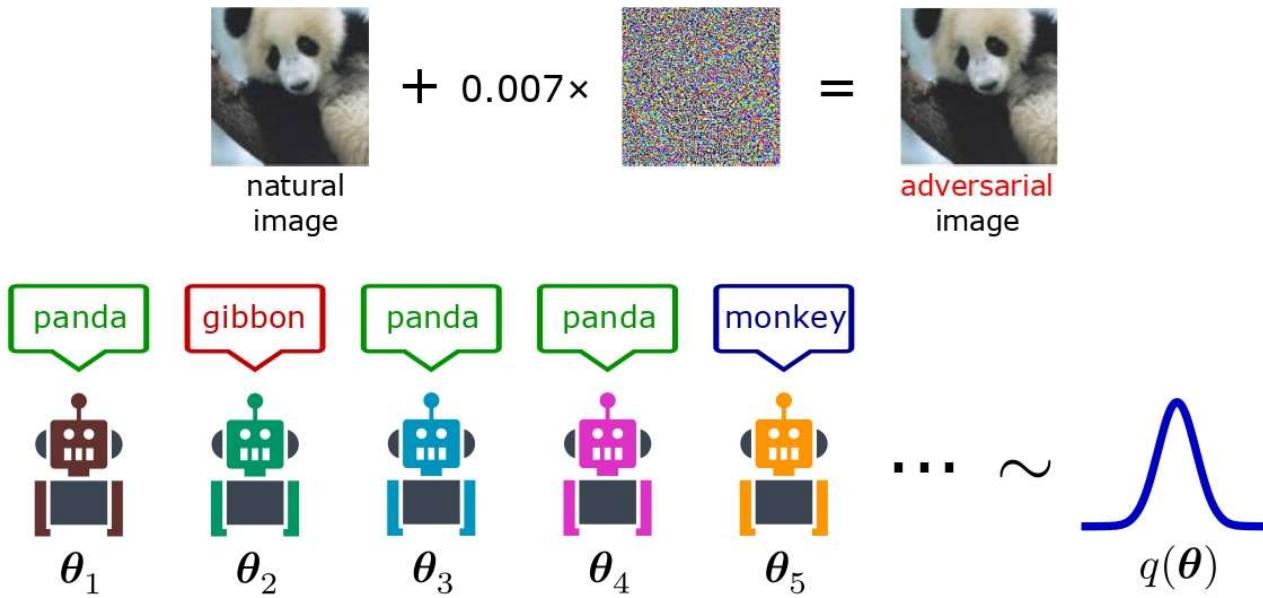
$$p(y^* | x^*, D) \approx \frac{1}{K} \sum_{k=1}^K p(y^* | x^*, \theta_k), \quad \theta_k \sim q(\theta)$$

# Bayesian Neural Network (BNN) 101



Prediction on in-distribution data:  
ensemble over networks, using weights sampled from  $q(\theta)$

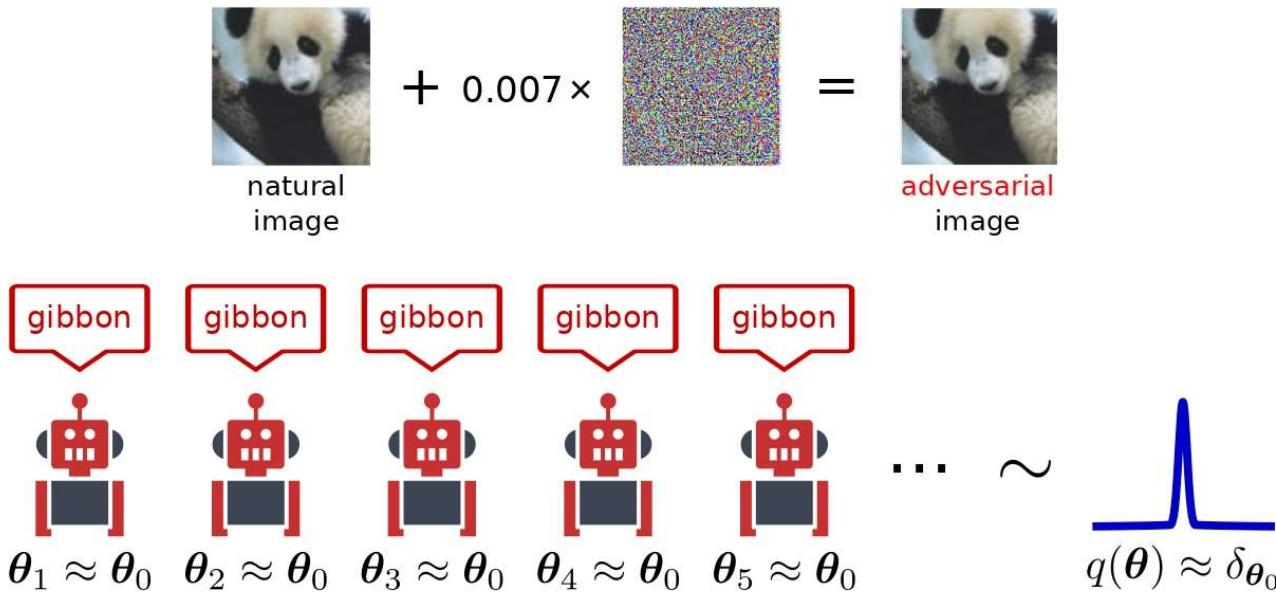
# Bayesian Neural Network (BNN) 101



Prediction on OOD/noisy/adversarial data:

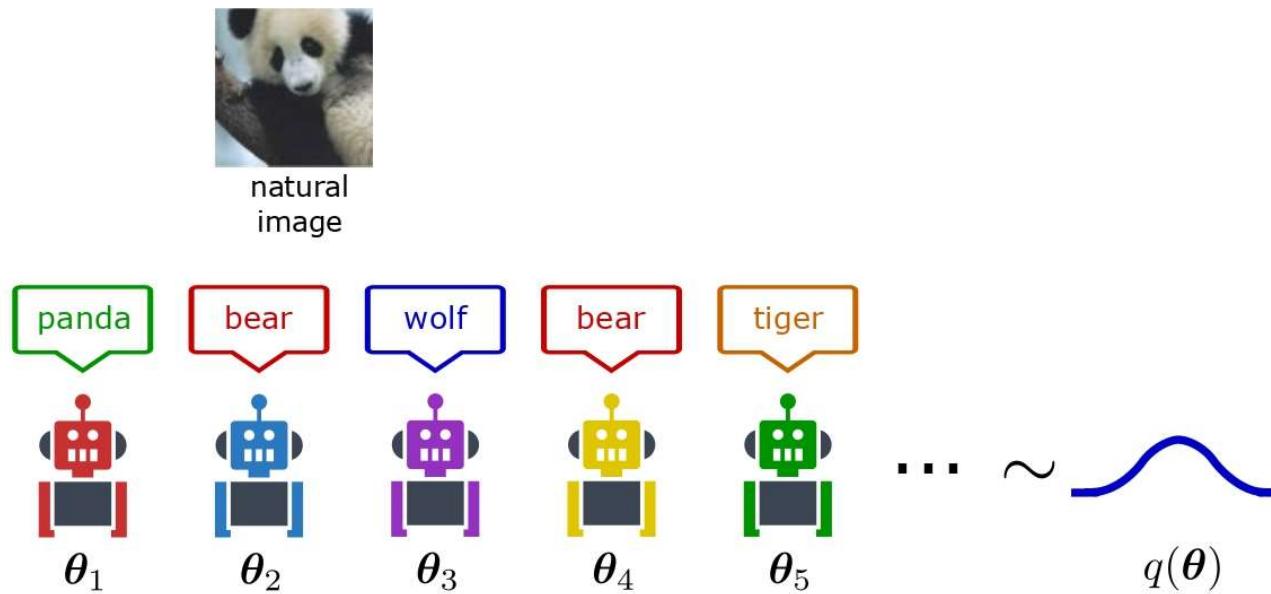
Disagreement (i.e. uncertainty) exists over networks sampled from  $q(\theta)$

# Bayesian Neural Network (BNN) 101



Prediction on OOD/noisy/adversarial data **when  $q(\theta)$  is over-confident:**  
Return **confidently wrong answers** (close to point estimate)

# Bayesian Neural Network (BNN) 101

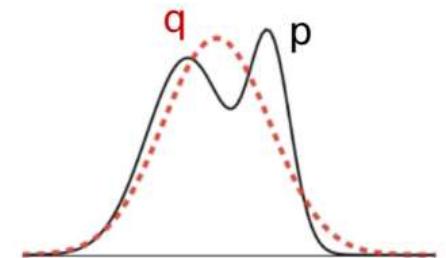


Prediction on in-distribution data **when  $q(\theta)$  is under-confident:**  
**Low accuracy in prediction tasks (less desirable)**

# Approximate Inference in BNNs

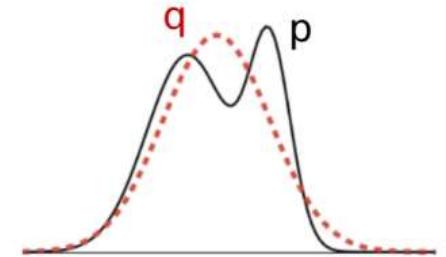
- Key steps of approximate inference in BNNs

1. Construct the  $q(\theta) \approx p(\theta | D)$  distribution
  - Simple distributions: e.g. Mean-field Gaussian
  - Structured approximations, e.g. low-rank Gaussians
  - Others (non-Gaussian)



# Approximate Inference in BNNs

- Key steps of approximate inference in BNNs
  1. Construct the  $q(\theta) \approx p(\theta | D)$  distribution
    - Simple distributions: e.g. Mean-field Gaussian
    - Structured approximations, e.g. low-rank Gaussians
    - Others (non-Gaussian)
  2. Fit the  $q(\theta)$  distribution
    - E.g. with variational inference

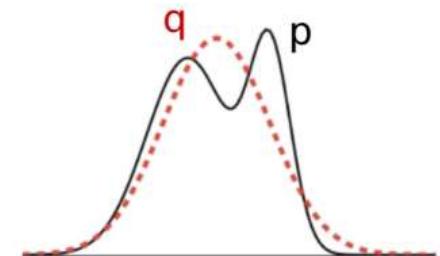


# Approximate Inference in BNNs

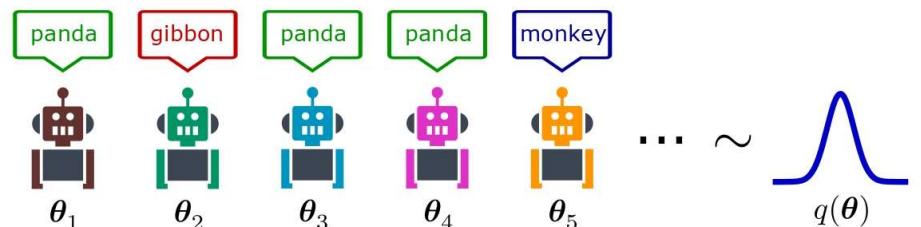
- Key steps of approximate inference in BNNs

1. Construct the  $q(\theta) \approx p(\theta | D)$  distribution
  - Simple distributions: e.g. Mean-field Gaussian
  - Structured approximations, e.g. low-rank Gaussians
  - Others (non-Gaussian)
2. Fit the  $q(\theta)$  distribution
  - E.g. with variational inference

3. Compute prediction with Monte Carlo approximations



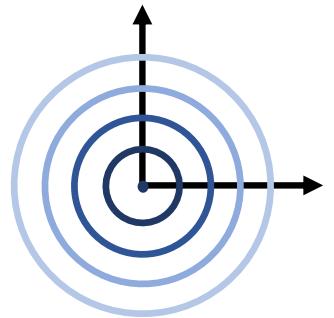
$$\text{natural image} + 0.007 \times \text{noise} = \text{adversarial image}$$



# Approximate Inference in BNNs

- Step 1: construct the  $q(\theta) \approx p(\theta | D)$  distribution
  - Example: Mean-field Gaussian distribution:

$$q(\theta) = \prod_{l=1}^L q(W^l) q(b^l)$$
$$q(W_l) = \prod_{ij} q(W_{ij}^l), \quad q(W_{ij}^l) = N(W_{ij}^l; M_{ij}^l, V_{ij}^l)$$
$$q(b^l) = \prod_i q(b_i^l), \quad q(b_i^l) = N(b_i^l; m_i^l, v_i^l)$$



- Variational parameters:  $\phi = \{M_{ij}^l, \log V_{ij}^l, m_i^l, \log v_i^l\}_{l=1}^L$

# Approximate Inference in BNNs

- Step 2: fit the  $q(\theta)$  distribution:
  - Variational inference:  $\phi^* = \operatorname{argmax} L(\phi)$ 
$$L(\phi) = E_{q(\theta)}[\log p(D | \theta)] - KL[q(\theta) || p(\theta)]$$

# Approximate Inference in BNNs

- Step 2: fit the  $q(\theta)$  distribution:

- Variational inference:  $\phi^* = \operatorname{argmax} L(\phi)$

$$L(\phi) = E_{q(\theta)}[\log p(D | \theta)] - KL[q(\theta) || p(\theta)]$$

- First scalable technique: **Stochastic optimization**

- i.i.d. assumption of data:  $\log p(D | \theta) = \sum_{n=1}^N \log p(y_n | x_n, \theta)$

- Enable mini-batch training with  $\{(x_m, y_m)\} \sim D^M$ :

$$L(\phi) \approx \frac{N}{M} \sum_{m=1}^M E_{q(\theta)}[\log p(y_m | x_m, \theta)] - KL[q(\theta) || p(\theta)]$$

# Approximate Inference in BNNs

- Step 2: fit the  $q(\theta)$  distribution:

- Variational inference:  $\phi^* = \operatorname{argmax} L(\phi)$

$$L(\phi) = E_{q(\theta)}[\log p(D | \theta)] - KL[q(\theta) || p(\theta)]$$

- First scalable technique: **Stochastic optimization**

- i.i.d. assumption of data:  $\log p(D | \theta) = \sum_{n=1}^N \log p(y_n | x_n, \theta)$

- Enable mini-batch training with  $\{(x_m, y_m)\} \sim D^M$ :

$$L(\phi) \approx \frac{N}{M} \sum_{m=1}^M E_{q(\theta)}[\log p(y_m | x_m, \theta)] - KL[q(\theta) || p(\theta)]$$

rewighting to ensure calibrated  
posterior concentration

# Approximate Inference in BNNs

- Step 2: fit the  $q(\theta)$  distribution:
  - 2nd scalable technique: **Monte Carlo sampling**
    - $E_{q(\theta)}[\log p(y | x, \theta)]$  intractable even with Gaussian  $q(\theta)$
    - **Solution: Monte Carlo estimate:**

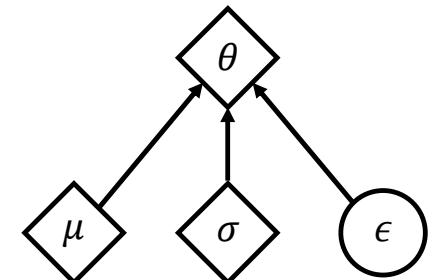
$$E_{q(\theta)}[\log p(y | x, \theta)] \approx \frac{1}{K} \sum_k^K \log p(y | x, \theta_k), \quad \theta_k \sim q(\theta)$$

# Approximate Inference in BNNs

- Step 2: fit the  $q(\theta)$  distribution:
  - 2nd scalable technique: **Monte Carlo sampling**
    - $E_{q(\theta)}[\log p(y | x, \theta)]$  intractable even with Gaussian  $q(\theta)$
    - **Solution: Monte Carlo estimate:**

$$E_{q(\theta)}[\log p(y | x, \theta)] \approx \frac{1}{K} \sum_k^K \log p(y | x, \theta_k), \quad \theta_k \sim q(\theta)$$

- **Reparameterization trick** to sample mean-field Gaussians:  
 $\theta_k \sim q(\theta) \Leftrightarrow \theta_k = m_\theta + \sigma_\theta \epsilon_k, \quad \epsilon_k \sim N(0, I)$



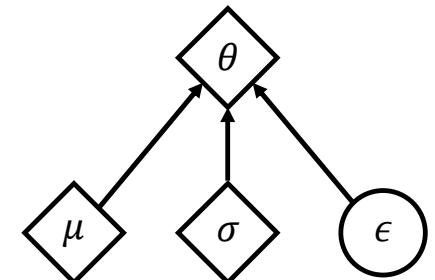
# Approximate Inference in BNNs

- Step 2: fit the  $q(\theta)$  distribution:
  - 2nd scalable technique: **Monte Carlo sampling**
    - $E_{q(\theta)}[\log p(y | x, \theta)]$  intractable even with Gaussian  $q(\theta)$
    - **Solution: Monte Carlo estimate:**

$$E_{q(\theta)}[\log p(y | x, \theta)] \approx \frac{1}{K} \sum_k^K \log p(y | x, \theta_k), \quad \theta_k \sim q(\theta)$$

- **Reparameterization trick** to sample mean-field Gaussians:  
 $\theta_k \sim q(\theta) \Leftrightarrow \theta_k = m_\theta + \sigma_\theta \epsilon_k, \quad \epsilon_k \sim N(0, I)$

$$\Rightarrow E_{q(\theta)} [\log p(y | x, \theta)] \approx \frac{1}{K} \sum_k^K \log p(y | x, m_\theta + \sigma_\theta \epsilon_k), \quad \epsilon_k \sim N(0, I)$$



# Approximate Inference in BNNs

- Combining both steps:

$$L(\phi) \approx \frac{N}{M} \sum_{m=1}^M \frac{1}{K} \sum_{k=1}^K \log p(y_m | x_m, \theta_k) - \underline{KL[q(\theta) || p(\theta)]}, \theta_k \sim q(\theta)$$

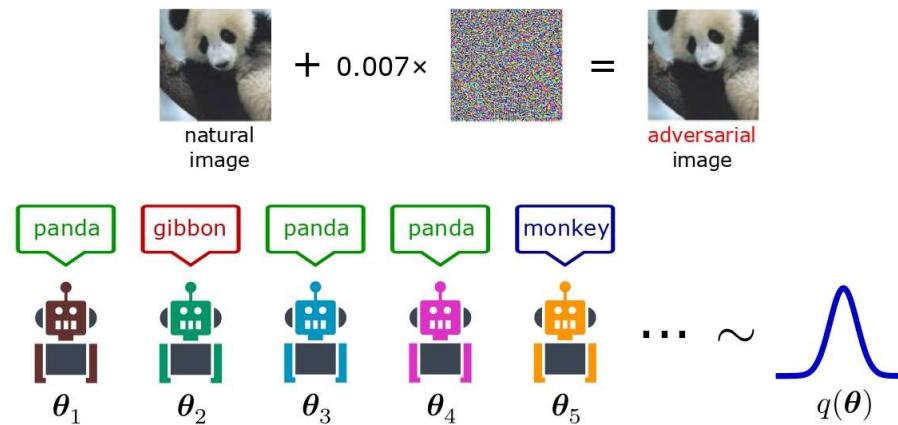
analytic between two Gaussians  
(if not, can also be estimated with Monte Carlo)

# Approximate Inference in BNNs

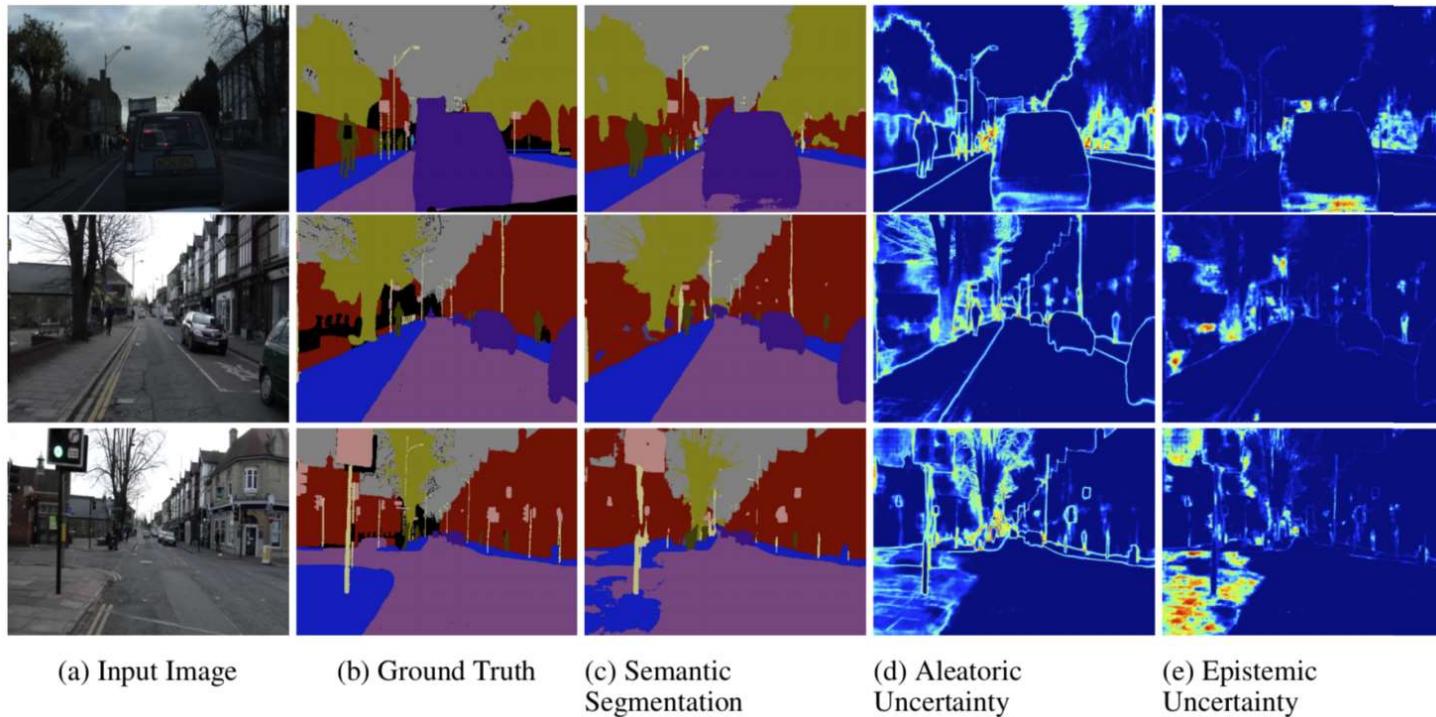
- Step 3: compute prediction with Monte Carlo approximations:

$$p(y^* | x^*, D) \approx \frac{1}{K} \sum_{k=1}^K p(y^* | x^*, \theta_k), \quad \underline{\theta_k \sim q(\theta)}$$

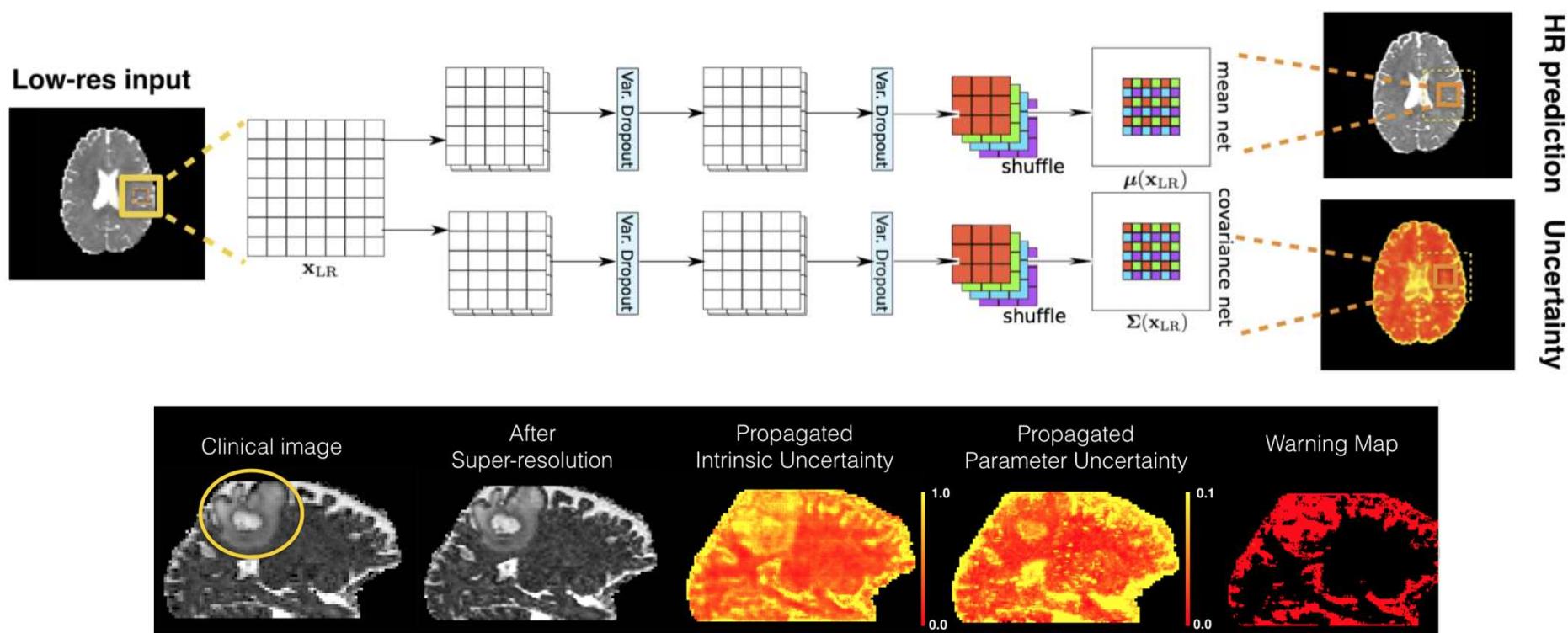
Mean-field Gaussian case:  
 $\theta_k = m_\theta + \sigma_\theta \epsilon_k, \epsilon_k \sim N(0, I)$



# Applications of BNNs: Image Segmentation

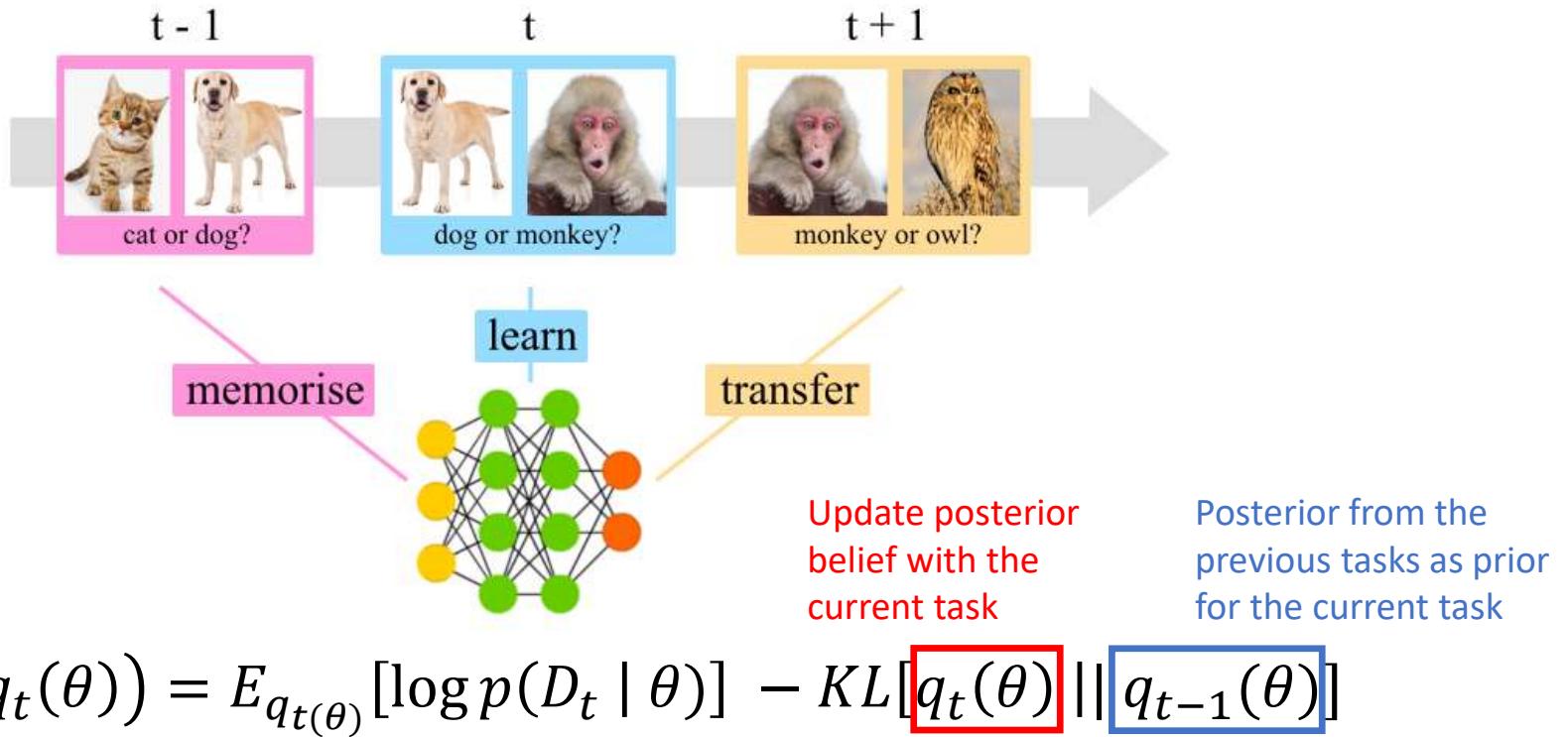


# Applications of BNNs: Super Resolution



Tanno et al. Uncertainty Quantification in Deep Learning for Safer Neuroimage Enhancement. Neuroimage 2020

# Applications of BNNs: Continual Learning



Nguyen et al. Variational Continual Learning. ICLR 2018

Pan et al. Continual Deep Learning by Functional Regularisation of Memorable Past. NeurIPS 2020

# Recent Progress in BNNs: Inference

$$\text{SGD: } \theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} \tilde{U}(\theta_t)$$

$$\text{SGLD: } \theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} \tilde{U}(\theta_t) + \sqrt{2\eta}\epsilon, \quad \epsilon \sim N(0, I)$$

Stochastic gradient MCMC

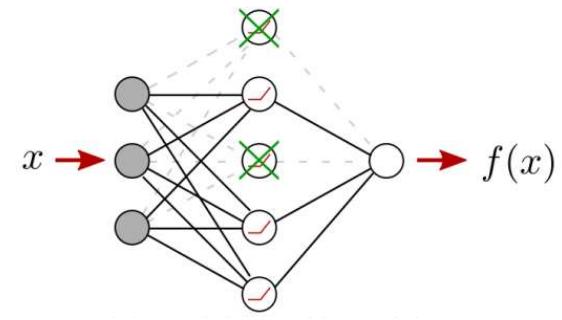
Li et al. Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks. AAAI 2016  
Zhang et al. Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning. ICLR 2020

# Recent Progress in BNNs: Inference

$$\text{SGD: } \theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} \tilde{U}(\theta_t)$$

$$\text{SGLD: } \theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} \tilde{U}(\theta_t) + \sqrt{2\eta}\epsilon, \quad \epsilon \sim N(0, I)$$

Stochastic gradient MCMC



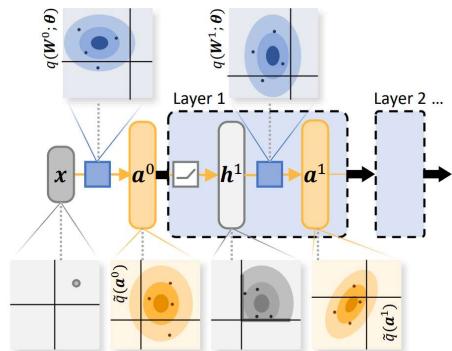
Monte Carlo dropout

# Recent Progress in BNNs: Inference

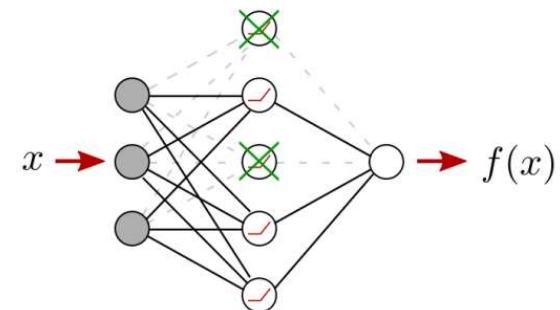
$$\text{SGD: } \theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} \tilde{U}(\theta_t)$$

$$\text{SGLD: } \theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} \tilde{U}(\theta_t) + \sqrt{2\eta}\epsilon, \quad \epsilon \sim N(0, I)$$

Stochastic gradient MCMC



Deterministic approximations



Monte Carlo dropout

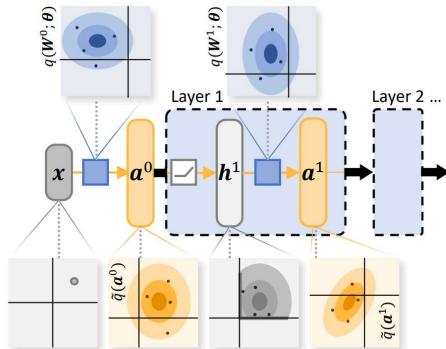
Hernandez-Lobato and Adams. Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. ICML 2015  
Wu et al. Deterministic Variational Inference for Robust Bayesian Neural Networks. ICLR 2019

# Recent Progress in BNNs: Inference

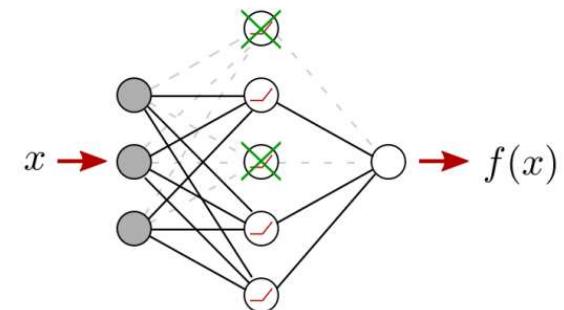
$$\text{SGD: } \theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} \tilde{U}(\theta_t)$$

$$\text{SGLD: } \theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} \tilde{U}(\theta_t) + \sqrt{2\eta}\epsilon, \quad \epsilon \sim N(0, I)$$

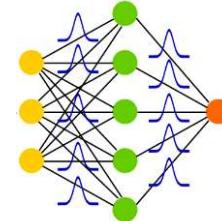
Stochastic gradient MCMC



Deterministic approximations



Monte Carlo dropout

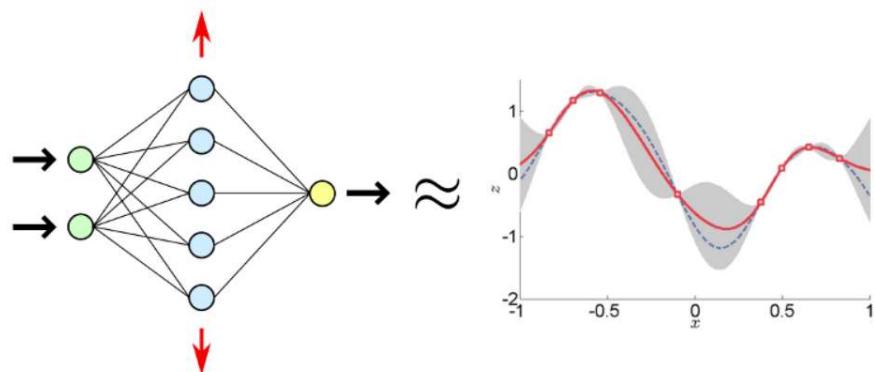


Function space approximate inference

Ma et al. Variational Implicit Processes. ICML 2019

Sun et al. Functional Variational Bayesian Neural Networks. ICLR 2019

# Recent Progress in BNNs: Theory



Connections to GPs:

- BNN with very wide hidden layers  
≈ Gaussian process
- Width limit convergence: in both prior (Neal's result) and posterior

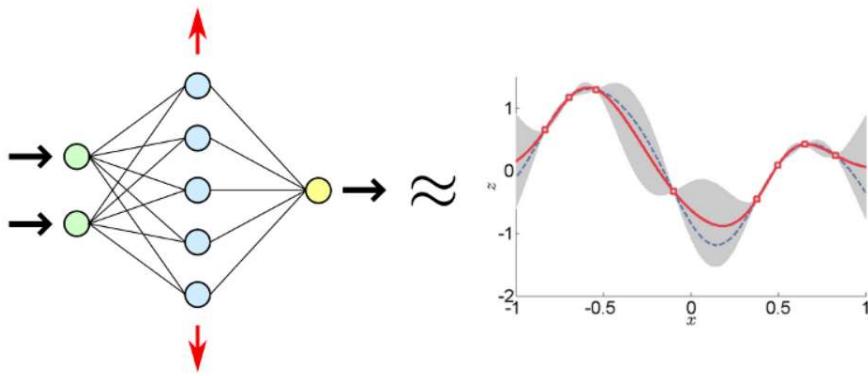
Neal. Bayesian Learning for Neural Networks. PhD Thesis, 1996

Matthews et al. Gaussian Process Behaviour in Wide Deep Neural Networks. ICLR 2018

Lee et al. Deep Neural Networks as Gaussian Processes. ICLR 2018

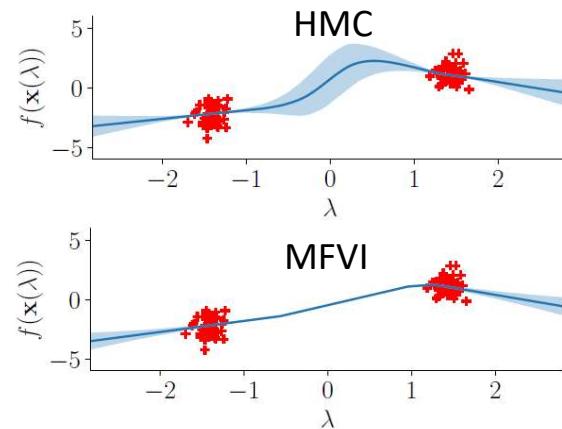
Hron et al. Exact posterior distributions of wide Bayesian neural networks. 2020

# Recent Progress in BNNs: Theory



Connections to GPs:

- BNN with very wide hidden layers  
≈ Gaussian process
- Width limit convergence: in both prior (Neal's result) and posterior



Approx. vs exact inference:

- Theoretical limitation of MFVI in shallow BNNs with ReLU activations
- Empirically deep BNNs with MVFI still fails in certain cases

Foong et al. On the Expressiveness of Approximate Inference in Bayesian Neural Networks. NeurIPS 2020

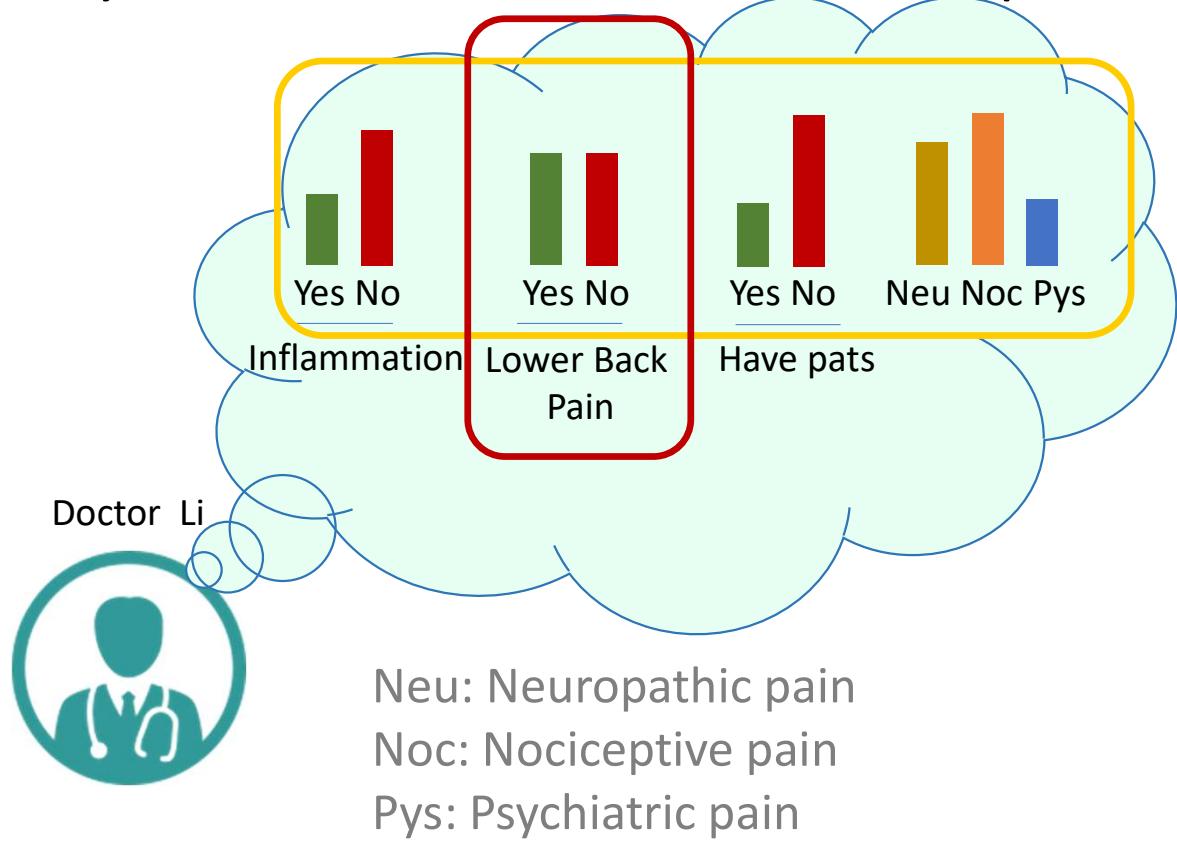
Farquhar et al. Liberty or Depth: Deep Bayesian Neural Nets Do Not Need Complex Weight Posterior Approximations. NeurIPS 2020

# Dynamic Information Acquisition

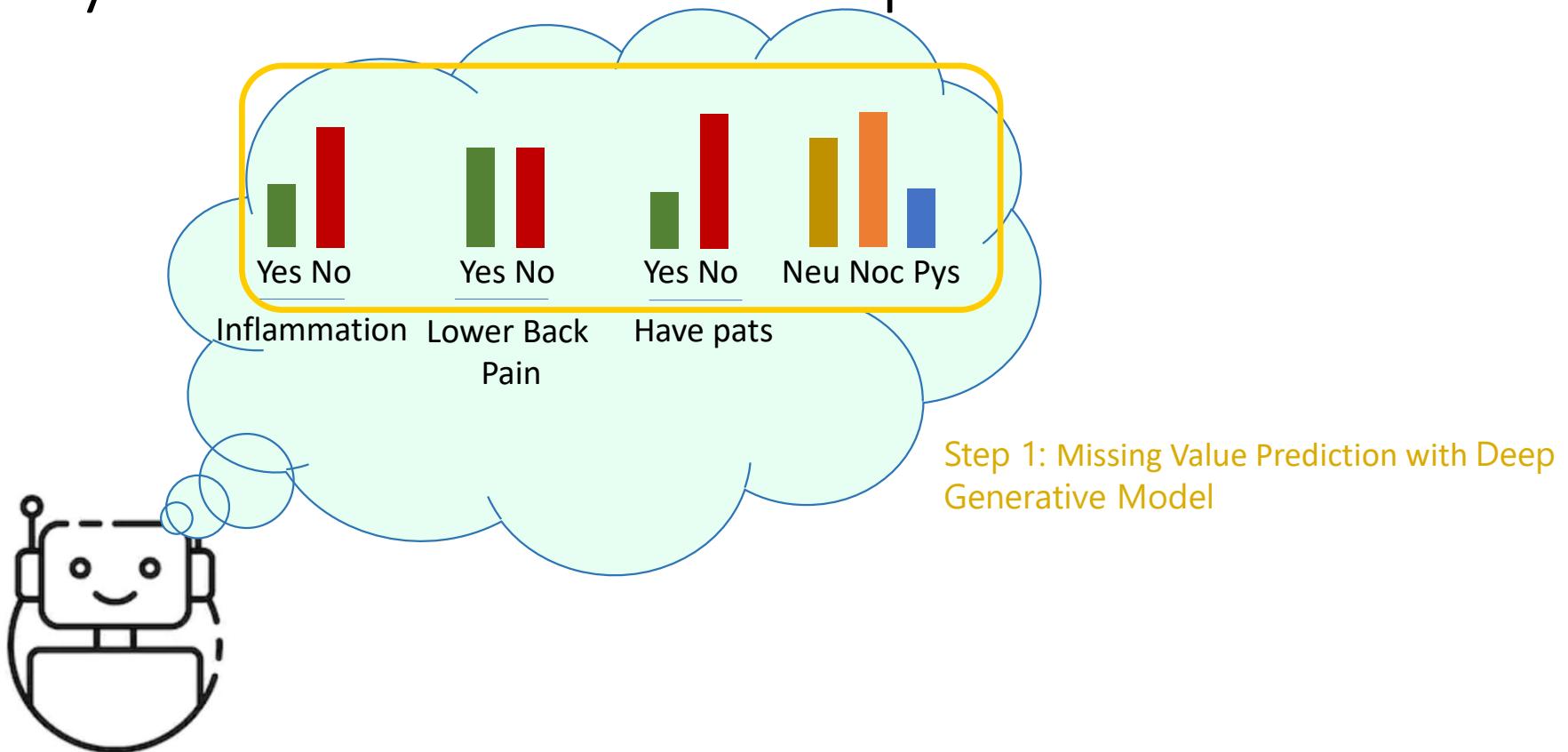
Doctor Li



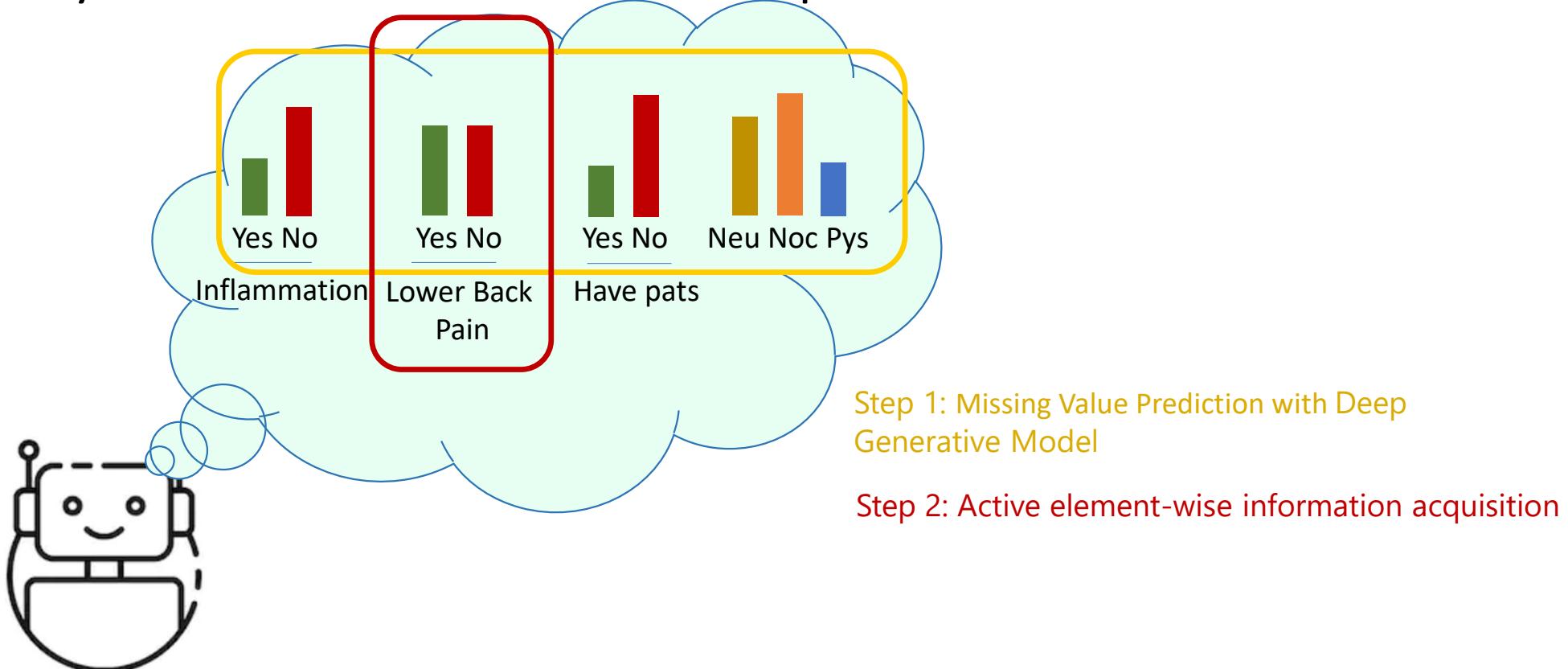
# Dynamic Information Acquisition



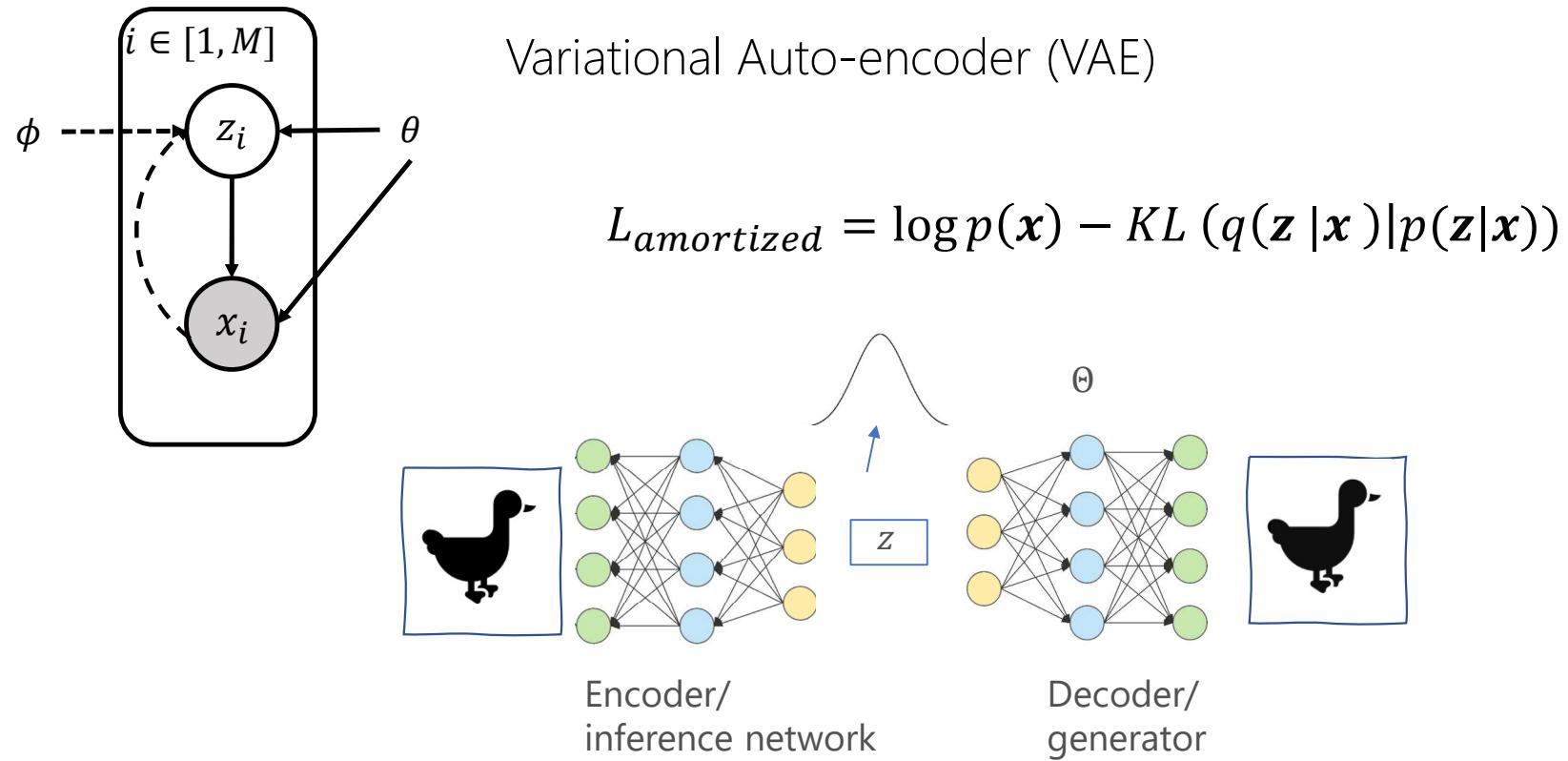
# Dynamic Information Acquisition



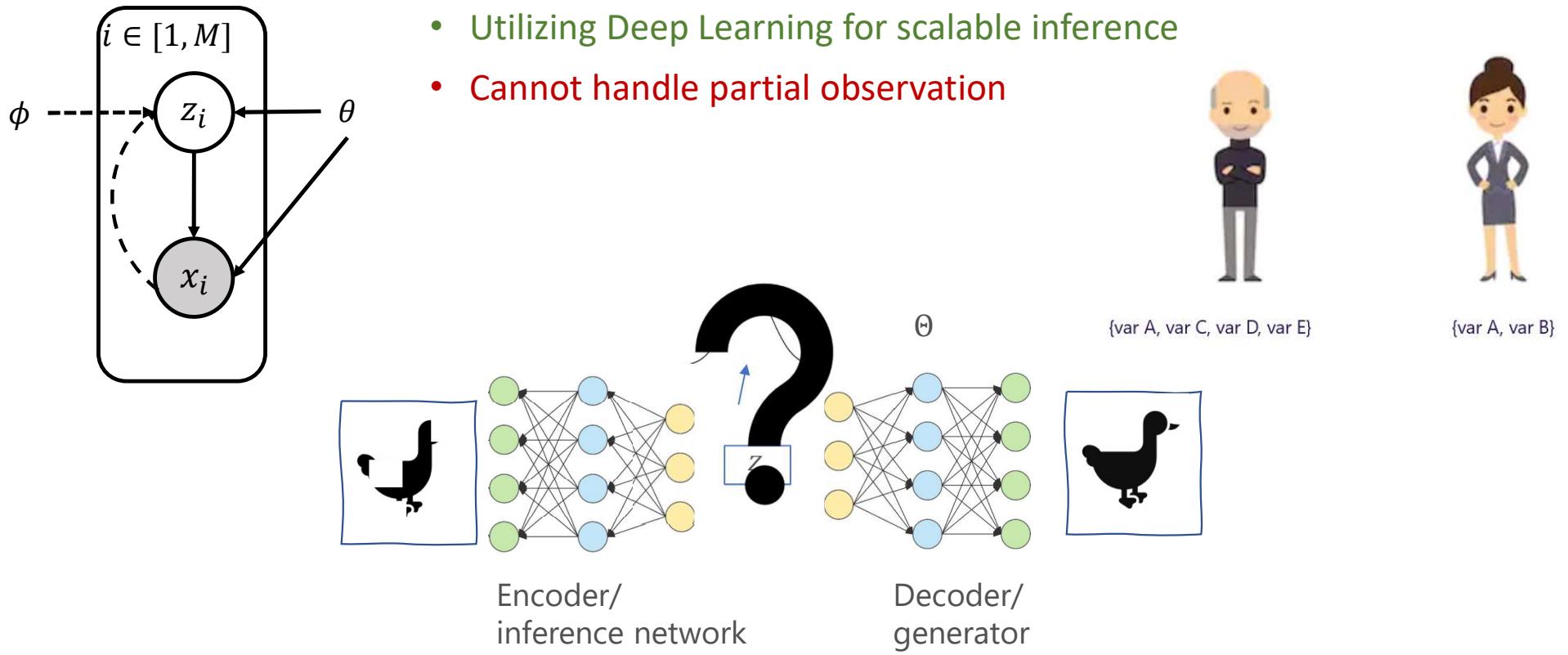
# Dynamic Information Acquisition



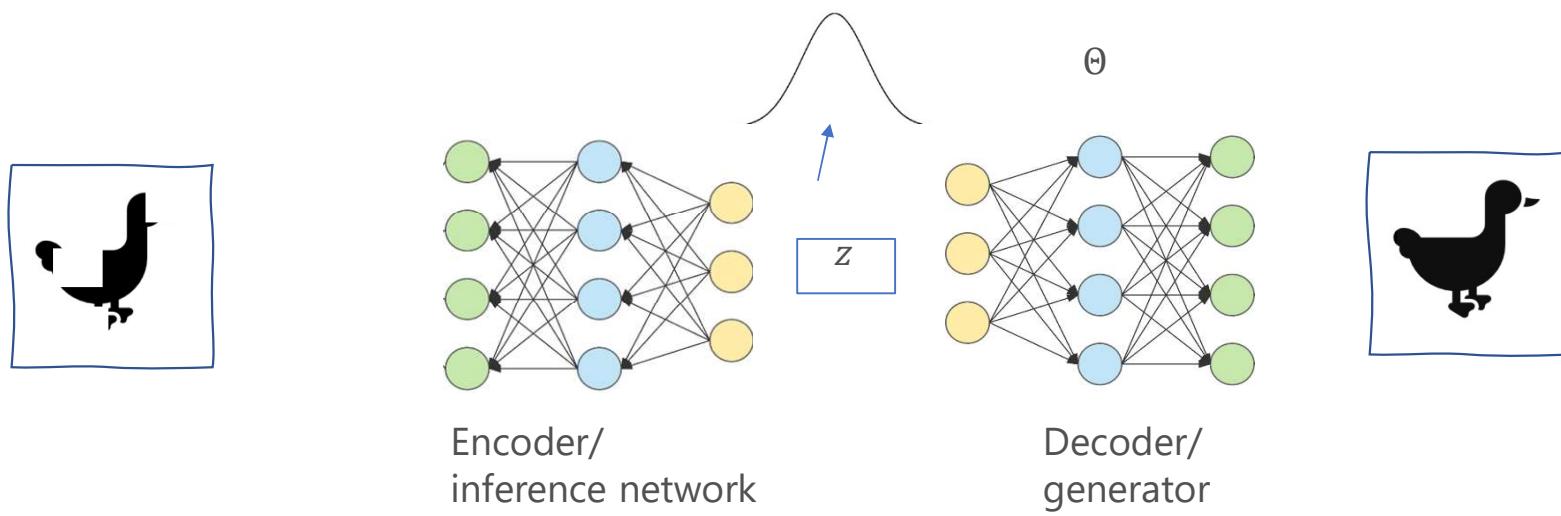
# A Deep Generative Model



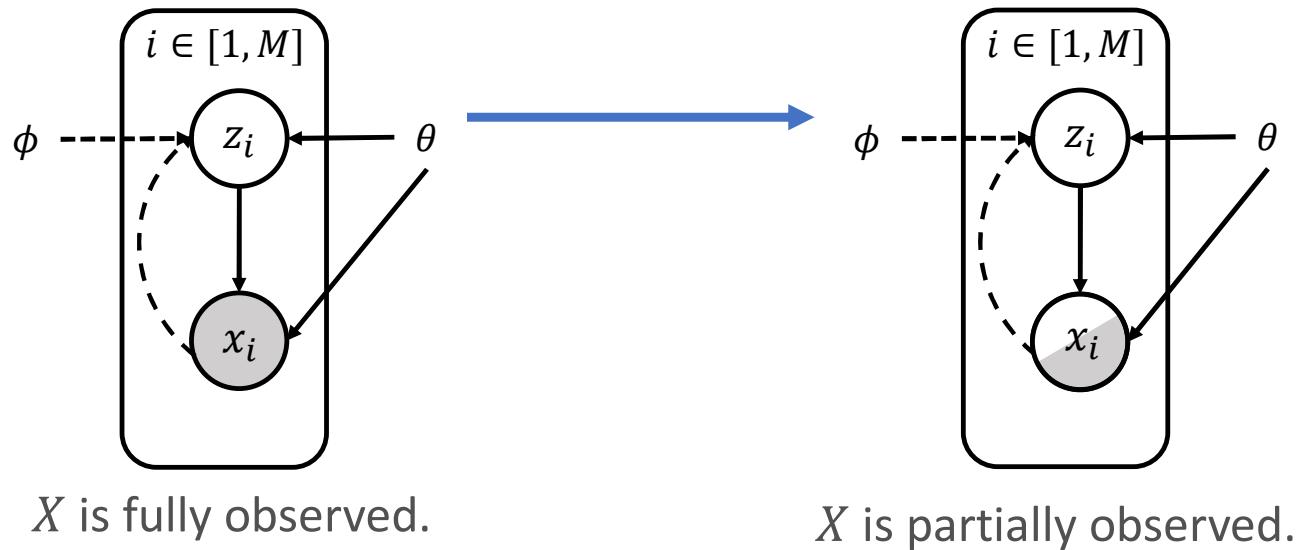
# Challenges



# VAE to Partial VAE

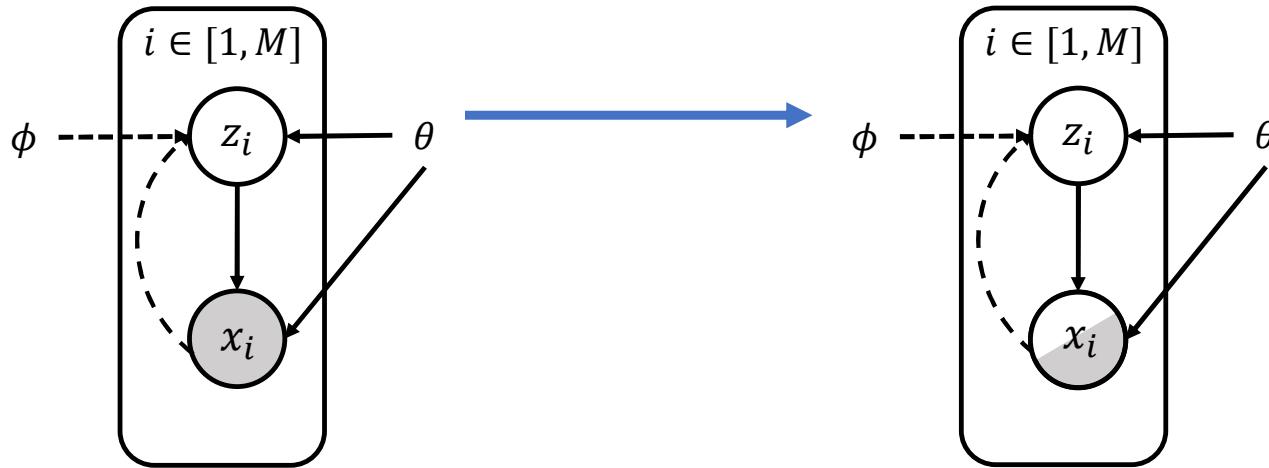


# VAE to Partial VAE



We aim to infer the missing values  $X_U$  from the observed values  $X_O$

# VAE to Partial VAE



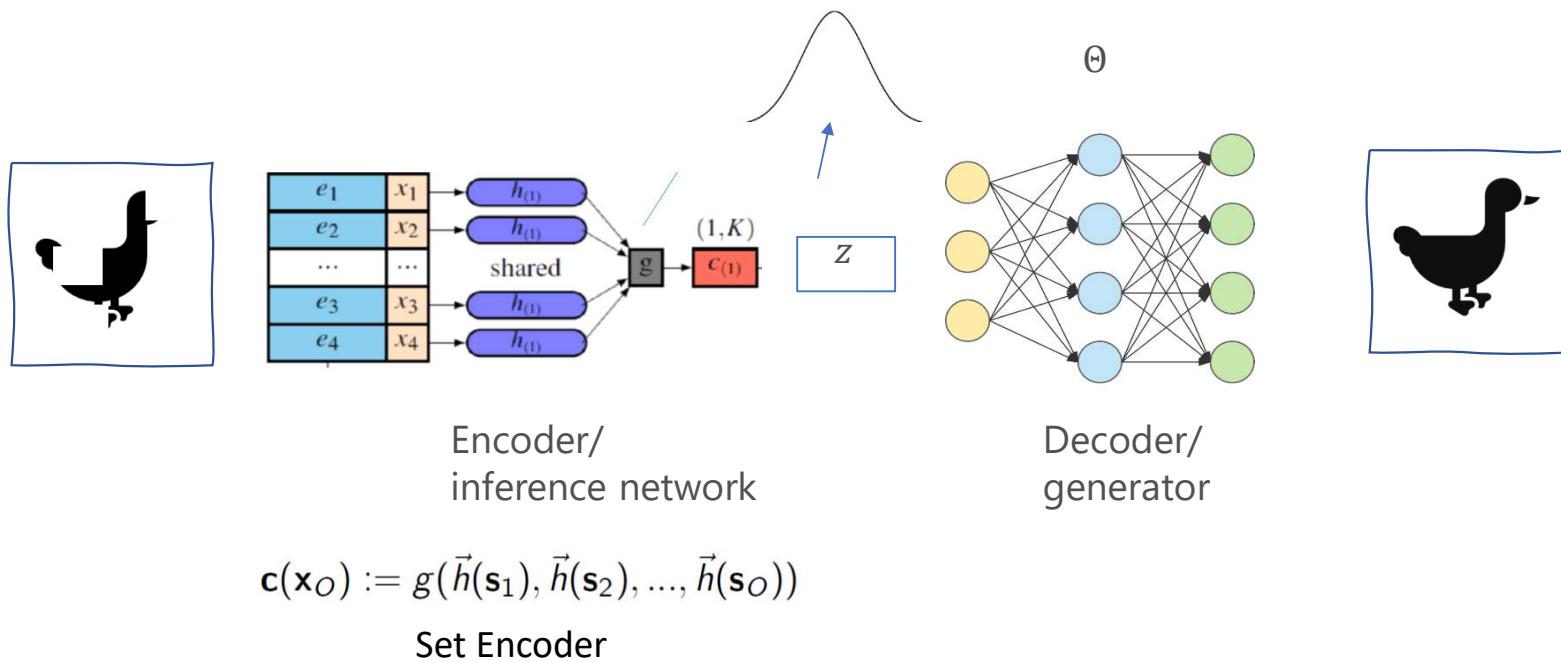
$$L_{amortized} = \log p(\mathbf{x}) - KL[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})] \\ = E_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})] - KL[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]$$

$$L_{amortized} = \log p(\mathbf{x}_o) - KL[q(\mathbf{z}|\mathbf{x}_o)||p(\mathbf{z}|\mathbf{x}_o)] \\ = E_{\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}_o)} [\log p_\theta(\mathbf{x}|\mathbf{z})] - KL[q(\mathbf{z}|\mathbf{x}_o)||p(\mathbf{z})]$$

The ELBO still holds.

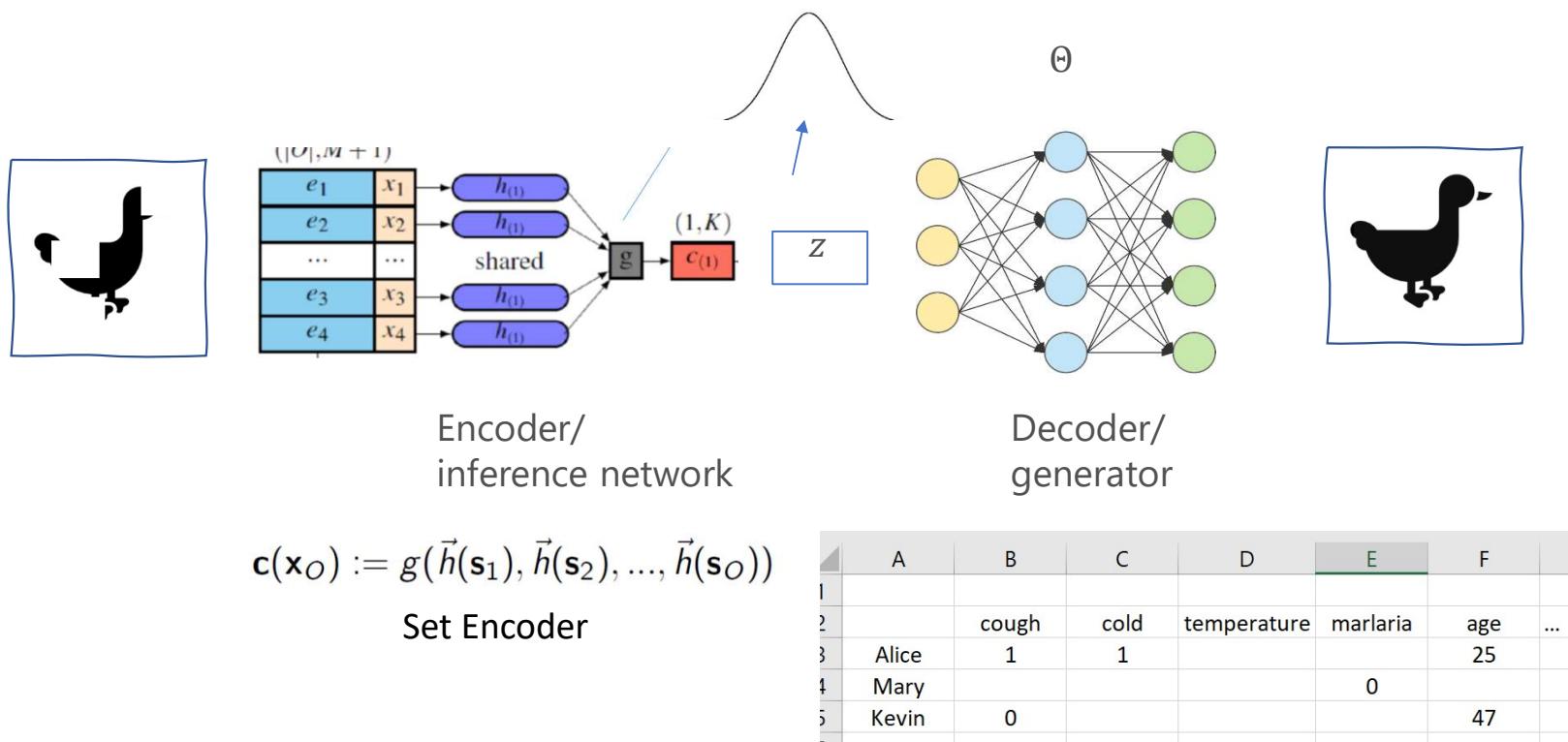
The challenge is how to design an inference net.

# VAE to Partial VAE



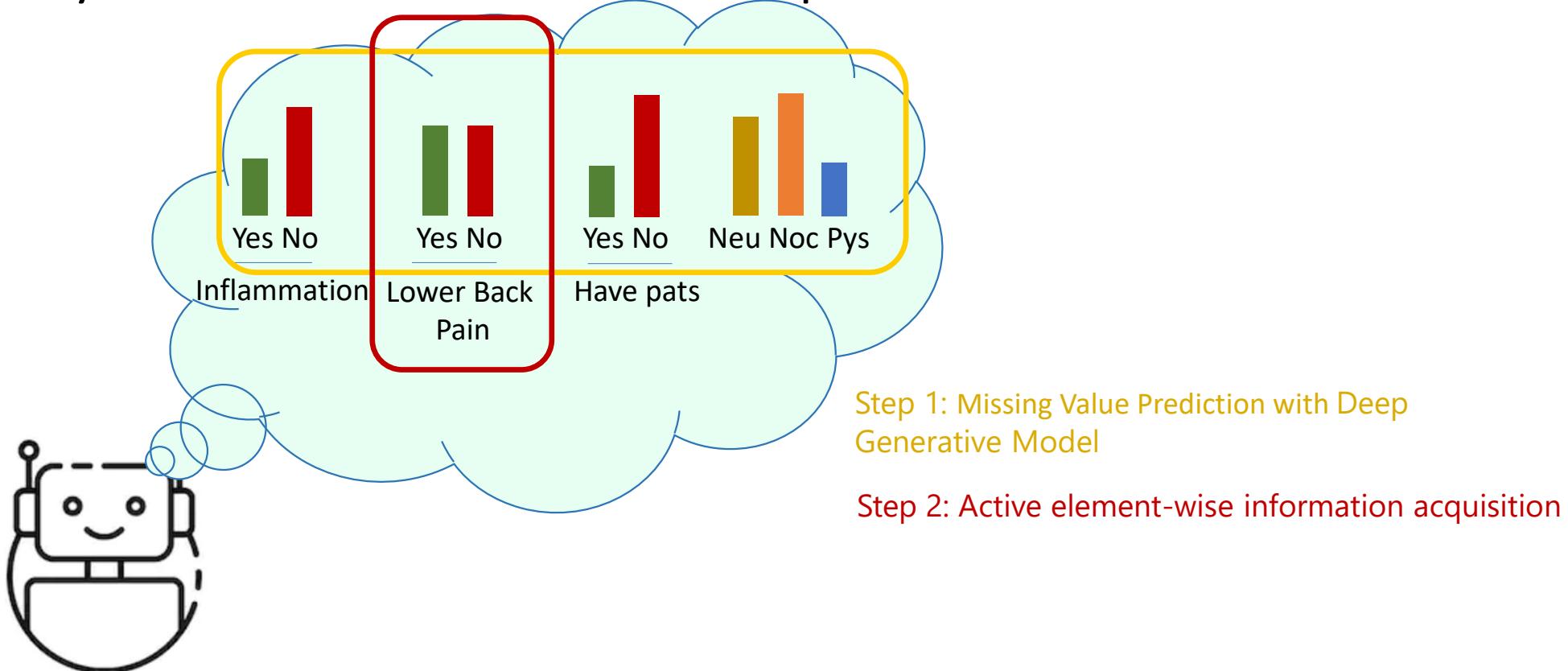
Qi et al. Pointnet: Deep learning on point sets for 3d classification and segmentation. CVPR 2017  
Zaheer et al. Deep sets. NeurIPS 2017

# VAE to Partial VAE

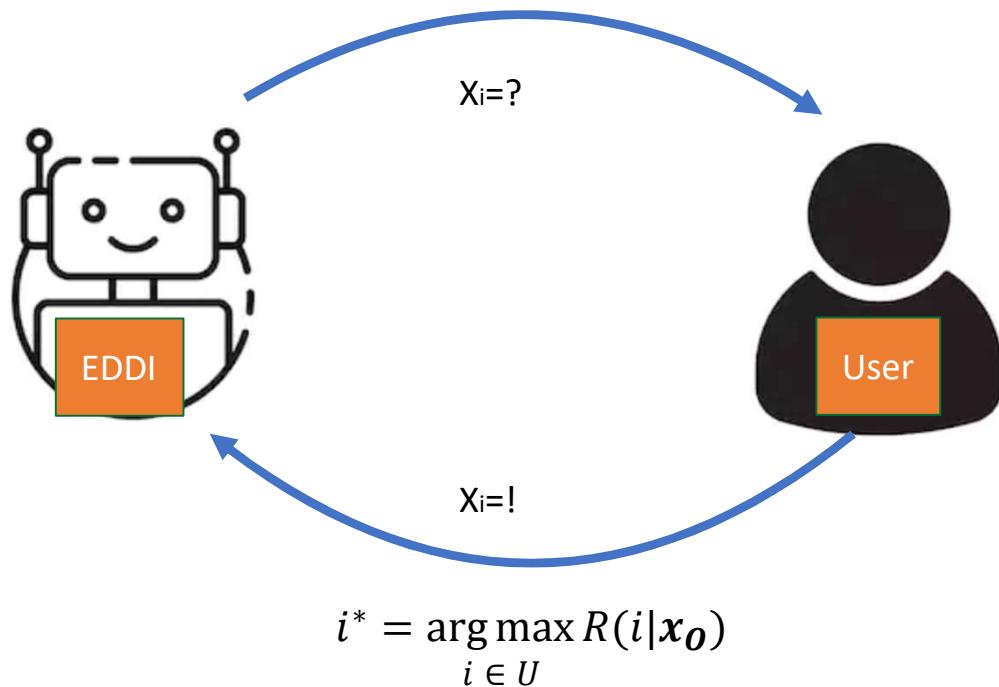


Ma et al. Eddi: Efficient dynamic discovery of high-value information with partial VAE. ICML 2019

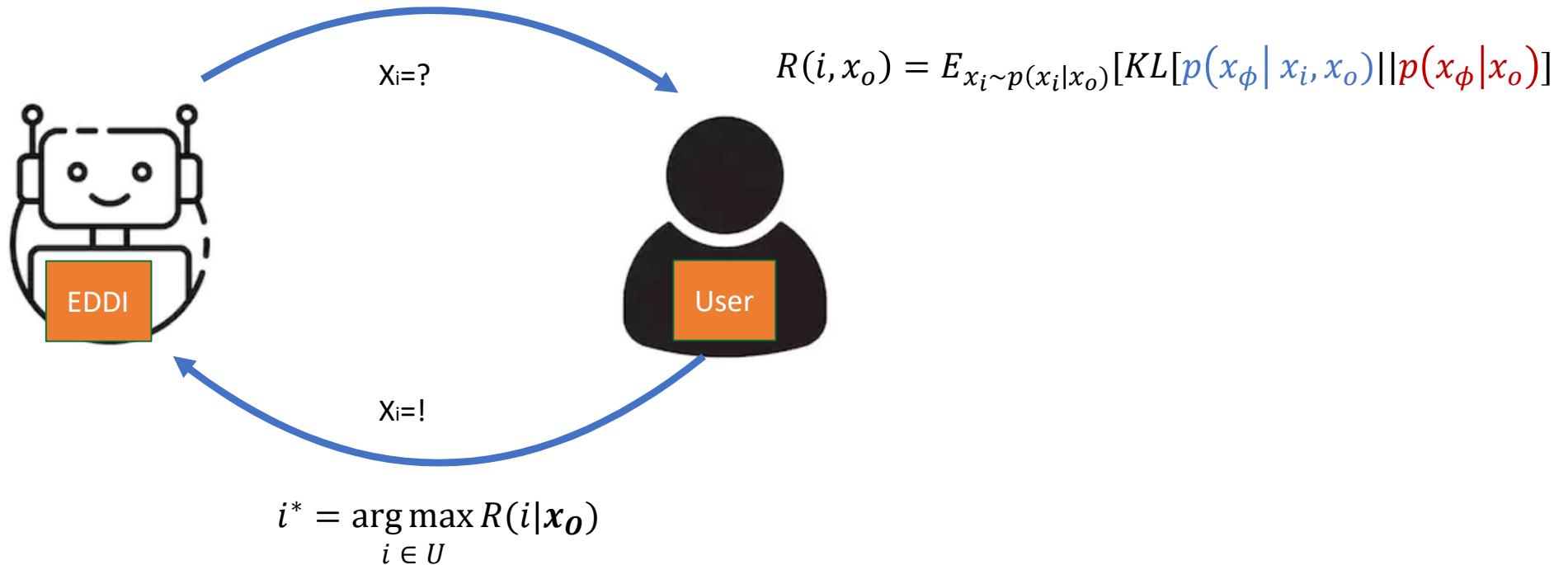
# Dynamic Information Acquisition



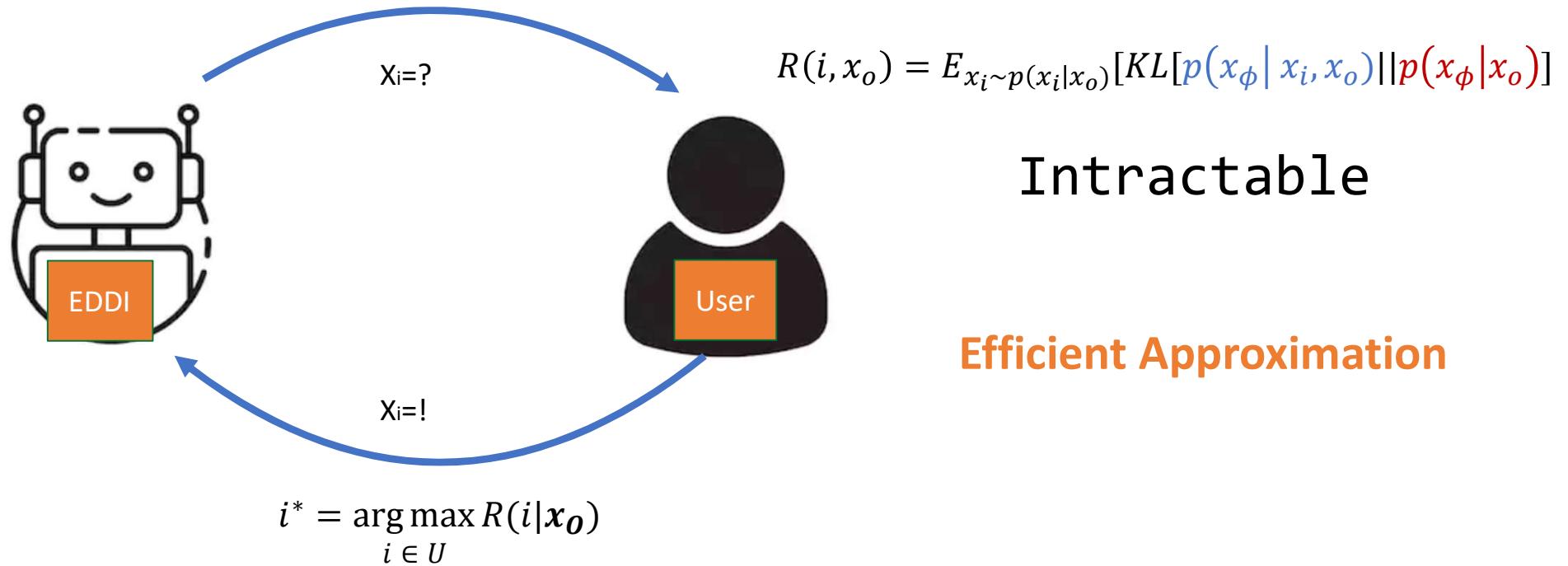
# Actively Select the Next Variable



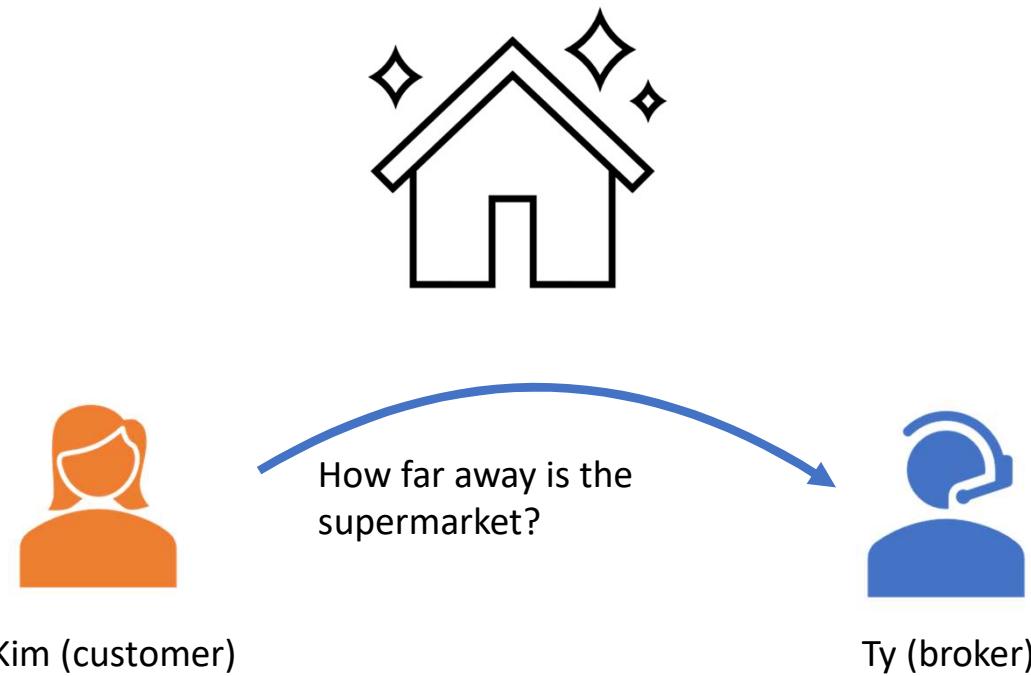
# Actively Select the Next Variable



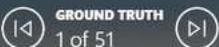
# Actively Select the Next Variable



# Predicting a House's Price



# Our solution



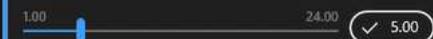
Model Questions

**TARGET VARIABLE**

median value of owner-occupied homes in \$1000's

**QUESTION 1**

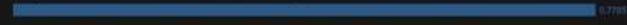
index of accessibility to radial highways



full-value property-tax rate per \$10.000



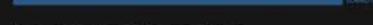
proportion of non-retail business acres per town



pupil-teacher ratio by town



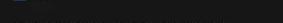
1000(B - 0.63)^2 where B is the proportion of students by town



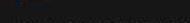
% lower status of the population



average number of rooms per dwelling



per capita crime rate by town



proportion of residential land zoned for lots over 25.000 sq.ft



charles river dummy variable (true if tract bounds river false otherwise)



nitrous oxides concentration (parts per 10 million)



proportion of owner-occupied units built prior to 1940



weighted distances to five Boston employment centres



Target Variable

Target Variable

Random Questions

# Baseline



Model Questions

**TARGET VARIABLE**

median value of owner-occupied homes in \$1000's

**QUESTION 1**

weighted distances to five Boston employment centres



median value of owner-occupied homes in \$1000's



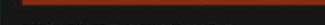
pupil-teacher ratio by town



proportion of residential land zoned for lots over 25.000 sq.ft



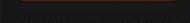
proportion of owner-occupied units built prior to 1940



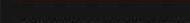
proportion of non-retail business acres per town



per capita crime rate by town



nitrous oxides concentration (parts per 10 million)



index of accessibility to radial highways



full-value property-tax rate per \$10.000



charles river dummy variable (true if tract bounds river false otherwise)



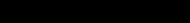
average number of rooms per dwelling



1000(B - 0.63)^2 where B is the proportion of students by town



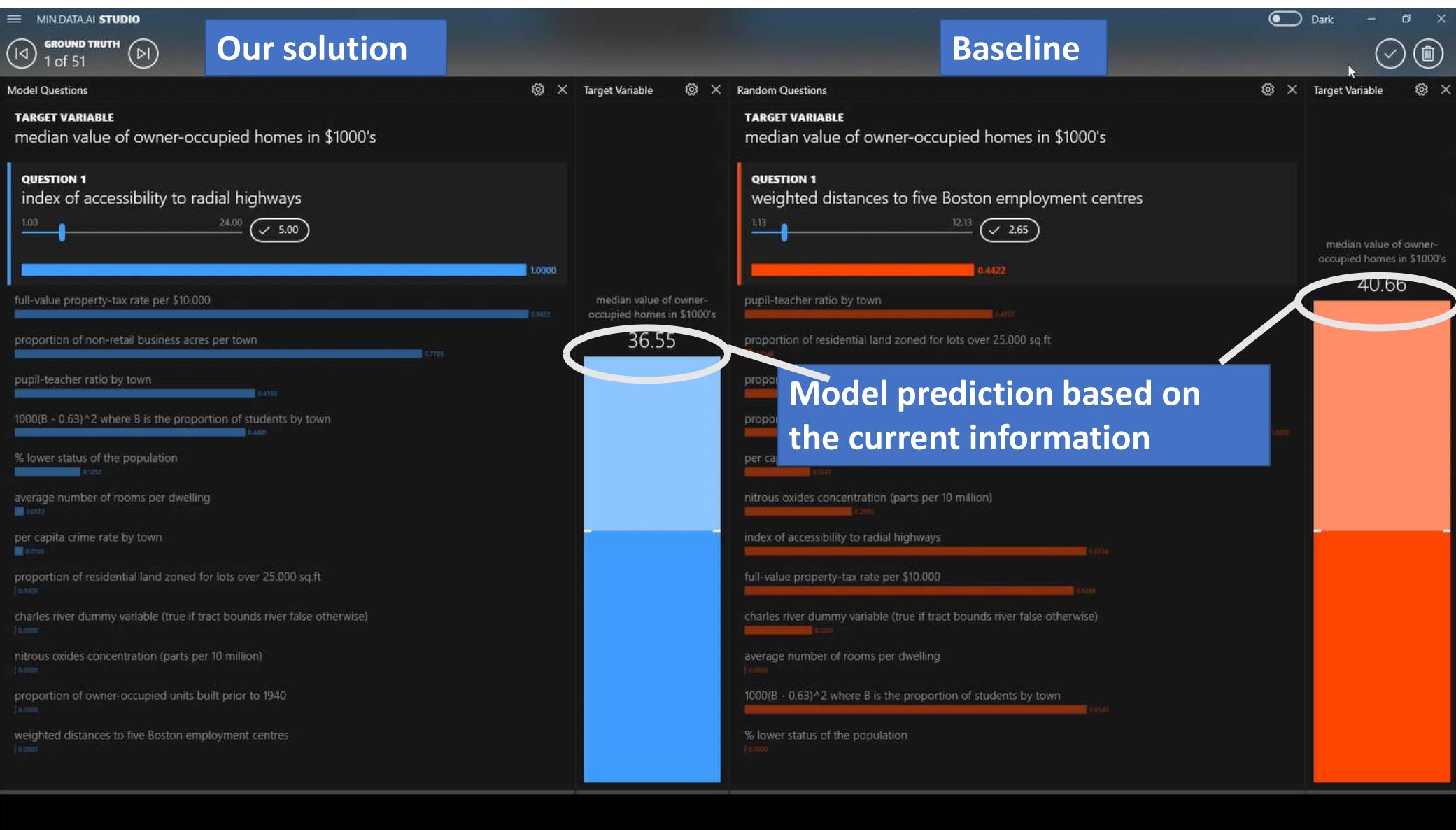
% lower status of the population

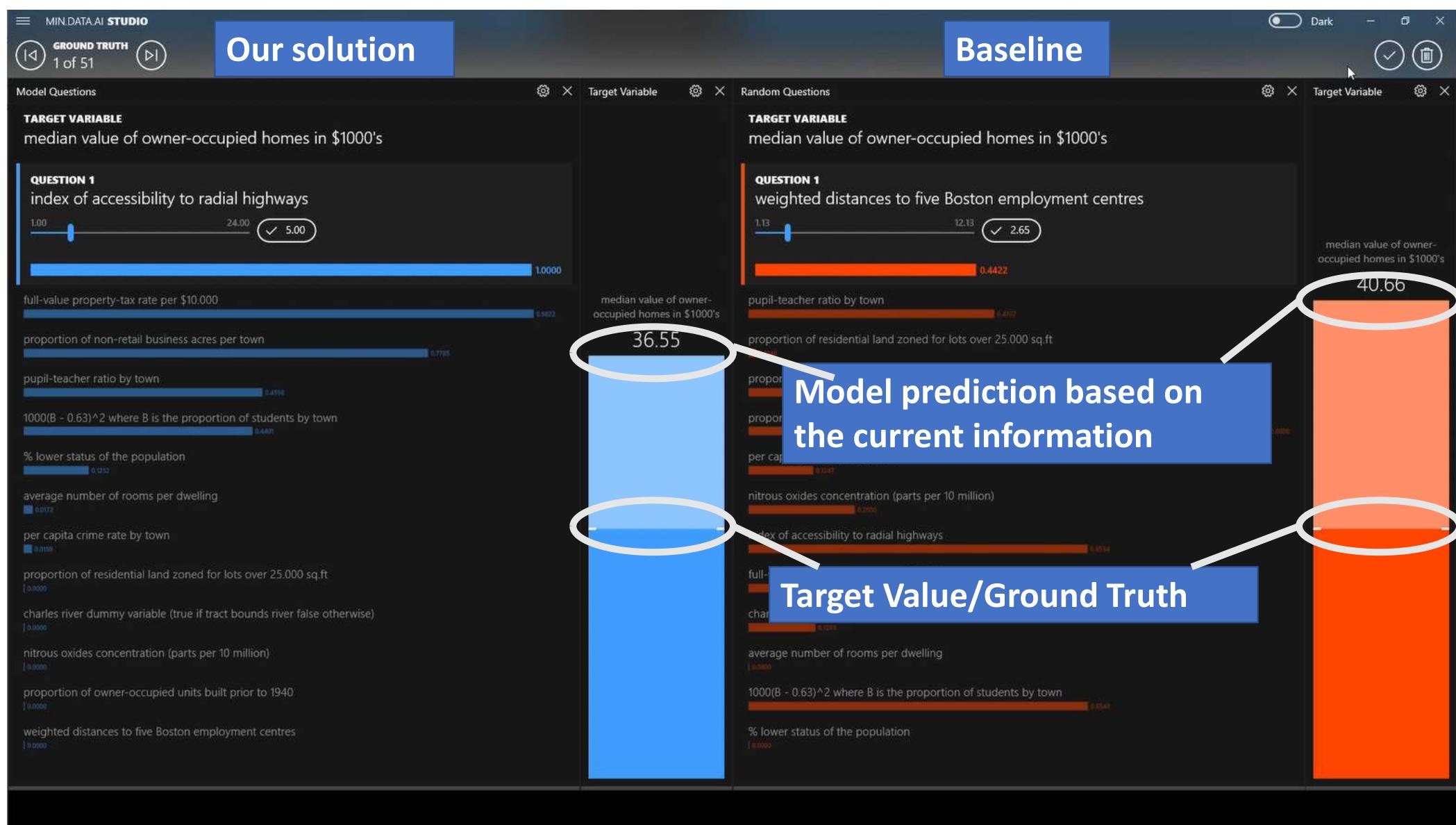


40.66

median value of owner-occupied homes in \$1000's







## Our solution

## Baseline



GROUND TRUTH

1 of 51



Model Questions

## TARGET VARIABLE

median value of owner-occupied homes in \$1000's

## List of questions that we could ask

homes in \$1000's

## QUESTION 1

index of accessibility to radial highways



full-value property-tax rate per \$10.000

proportion of non-retail business acres per town

pupil-teacher ratio by town

1000(B - 0.63)^2 where B is the proportion of students by town

% lower status of the population

average number of rooms per dwelling

per capita crime rate by town

proportion of residential land zoned for lots over 25.000 sq.ft

charles river dummy variable (true if tract bounds river false otherwise)

nitrous oxides concentration (parts per 10 million)

proportion of owner-occupied units built prior to 1940

weighted distances to five Boston employment centres

36.55

## QUESTION 1

weighted distances to five Boston employment centres



pupil-teacher ratio by town

proportion of residential land zoned for lots over 25.000 sq.ft

proportion of owner-occupied units built prior to 1940

proportion of non-retail business acres per town

per capita crime rate by town

nitrous oxides concentration (parts per 10 million)

index of accessibility to radial highways

full-value property-tax rate per \$10.000

charles river dummy variable (true if tract bounds river false otherwise)

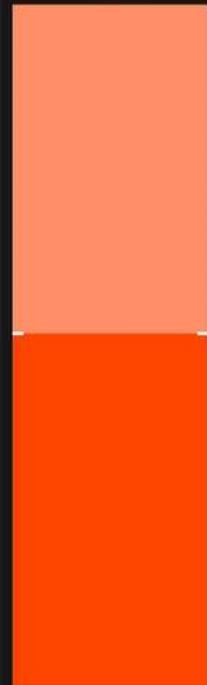
average number of rooms per dwelling

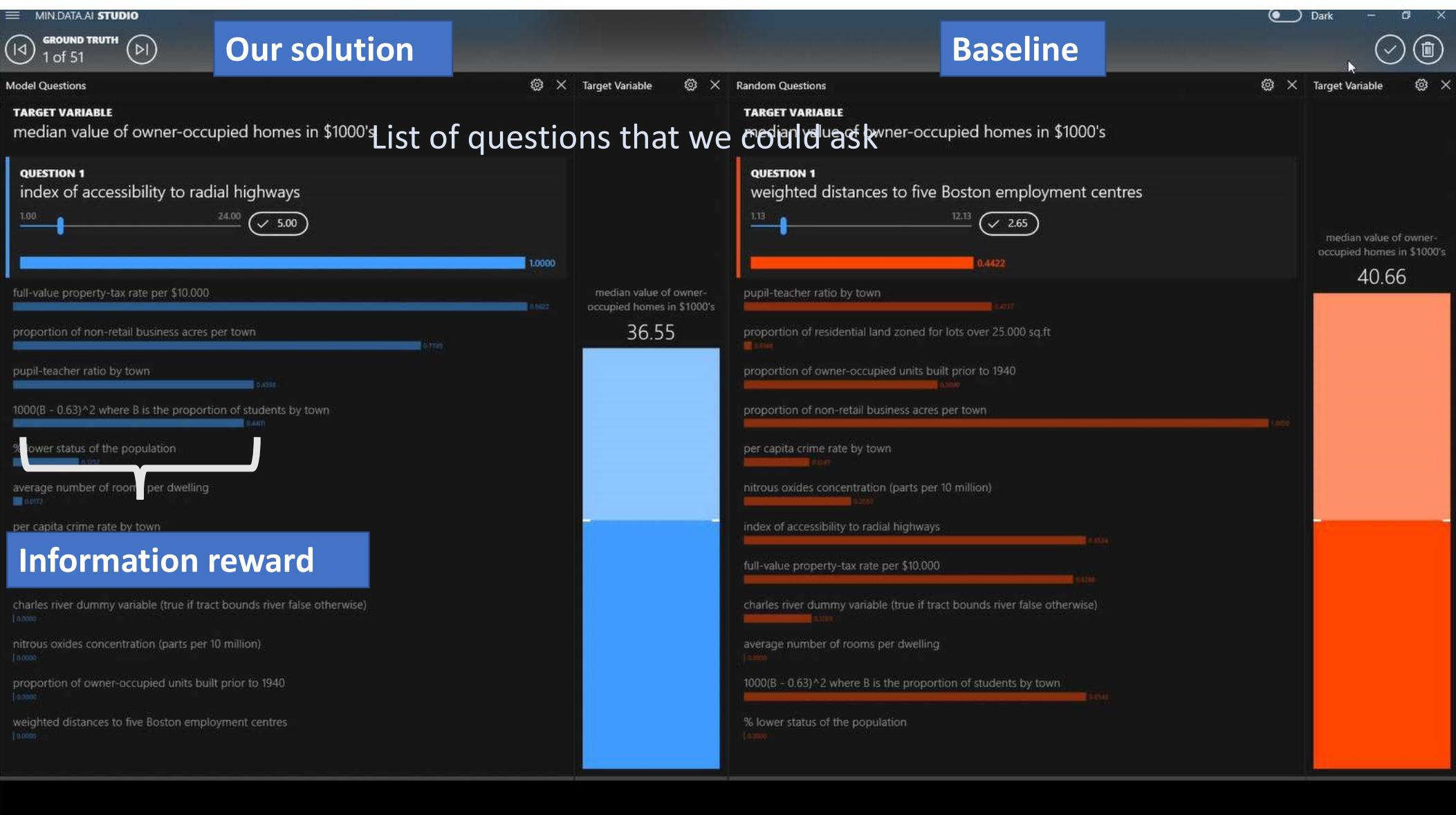
1000(B - 0.63)^2 where B is the proportion of students by town

% lower status of the population

median value of owner-occupied homes in \$1000's

40.66





MIN.DATA.AI STUDIO

MODEL

Boston - Media Home Value

Refresh

GROUND TRUTH

1 of 51

C:\Users\mgrayson\Desktop\min-data-ai\boston\test.csv

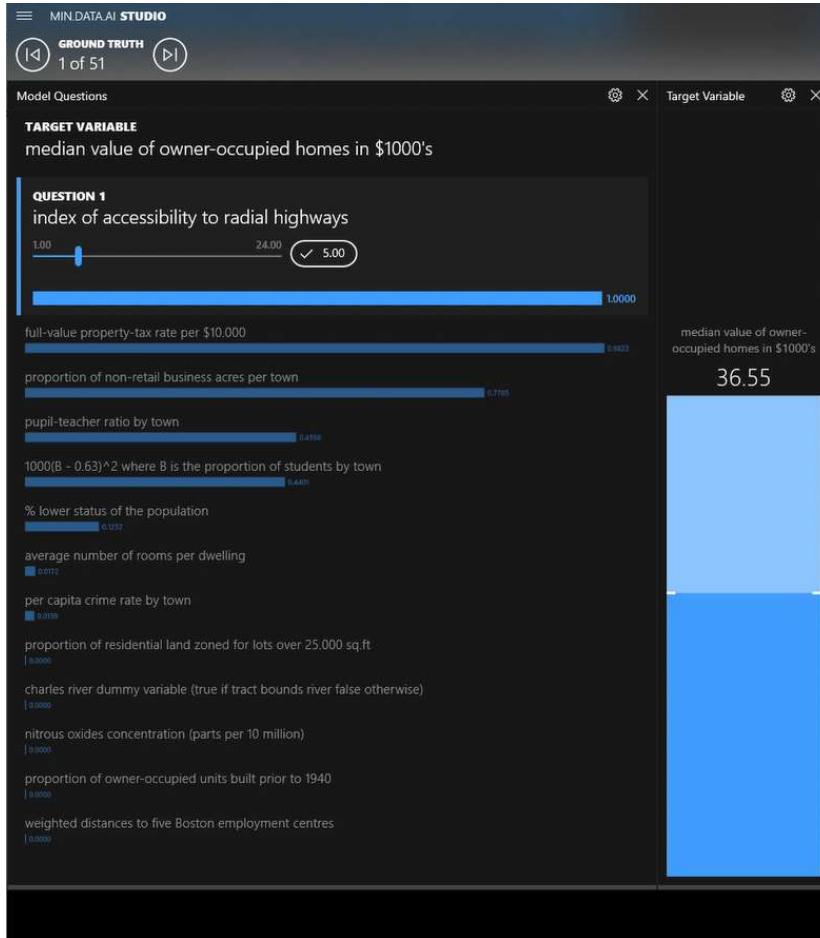
Browse...

PANELS

✓ Model questions

✗ Random questions

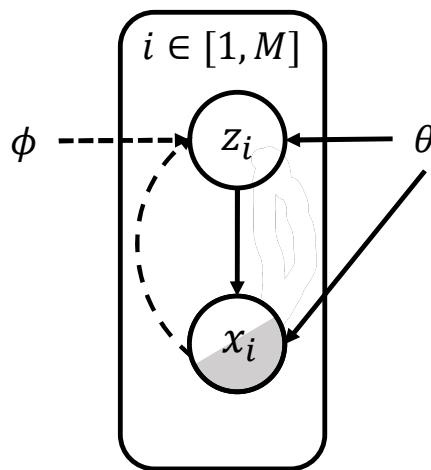
Tools



- List of questions from survey (e.g. US veteran) for mental health monitoring
- List of technical questions from interview for recruiting
- List of medical tests for diagnosis
- ... ...

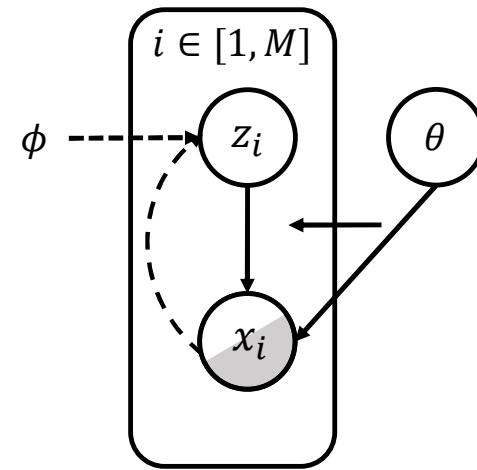
Does it work when there is few training data?

# Partial Amortized Bayesian Deep Latent Gaussian Model (PA-BELGAM)



- Point estimate of global parameter  $\theta$

$$p(\mathbf{x}_o, \mathbf{z}) = \prod_{i=1}^N \prod_{d \in O_i} p(x_{i,d} | z_i) p(z_i)$$



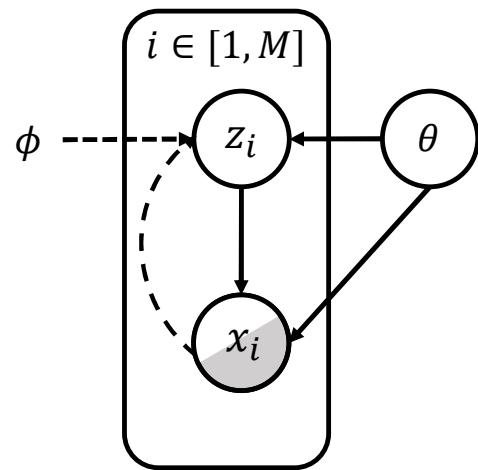
- Stochastic variable  $\theta$

$$p(\mathbf{x}_o, \theta, \mathbf{z}) = p(\theta) \prod_{i=1}^N \prod_{d \in O_i} p(x_{i,d} | z_i, \theta) p(z_i)$$

# Partial Amortized Bayesian Deep Latent Gaussian Model (PA-BELGAM)

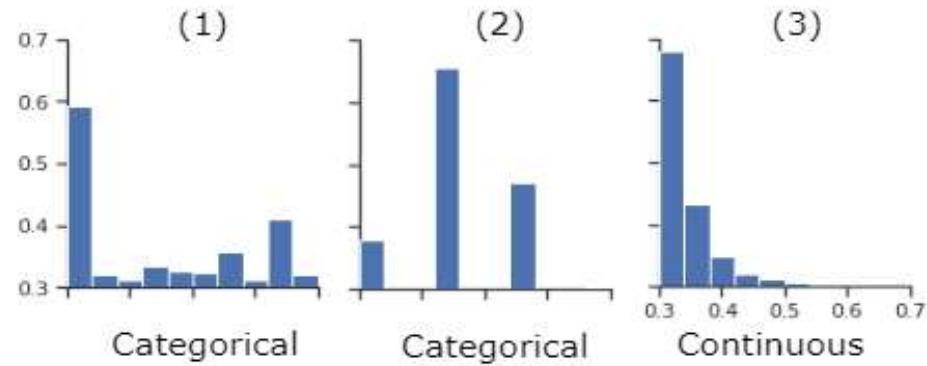
Amortized inference  
for local latent  
variables

SGHMC for global  
latent variables

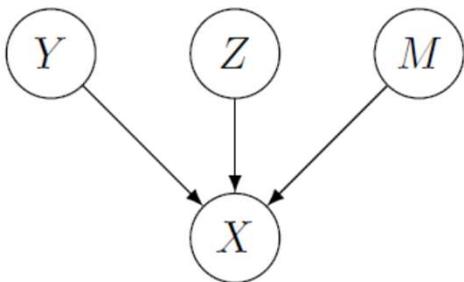


$$q(\theta, \mathbf{z} | \mathbf{x}_o) \approx q(\theta | \mathbf{x}_o) q_\phi(\mathbf{z} | \mathbf{x}_o)$$

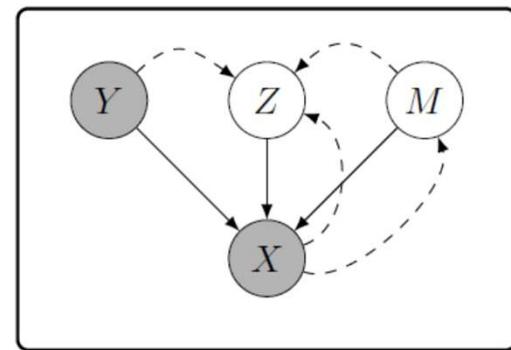
# When Data are Heterogenous



# When Robustness is Needed



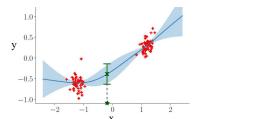
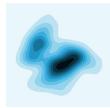
Causal Reasoning



Deep Casual  
Manipulation  
Augmented Model

# Summary

The probability distribution  $\pi(\theta)$  is intractable

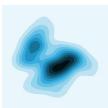


Approximate Inference

$$\int F(\theta) \pi(\theta) d\theta$$

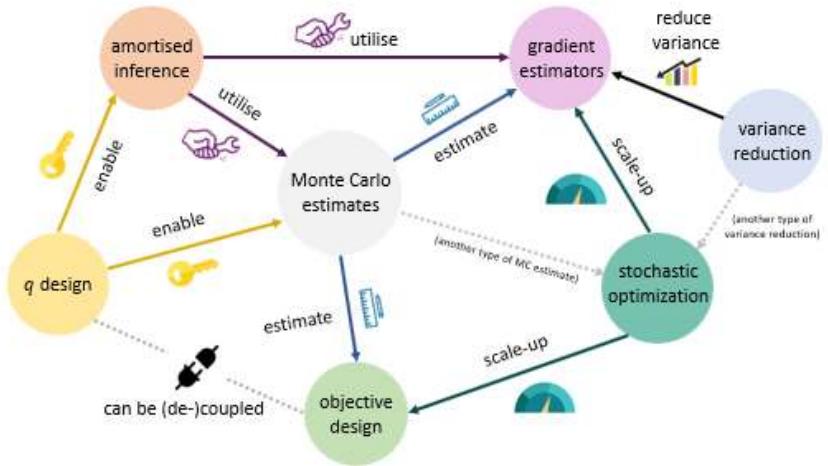
# Summary

The probability distribution  $\pi(\theta)$  is intractable



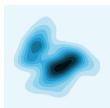
Approximate Inference

$$\int F(\theta) [\pi(\theta)] d\theta$$



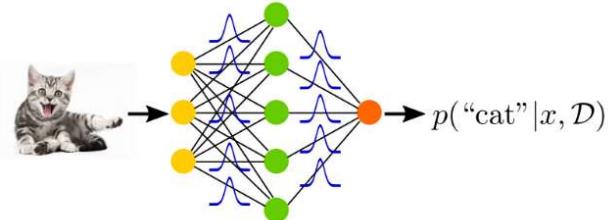
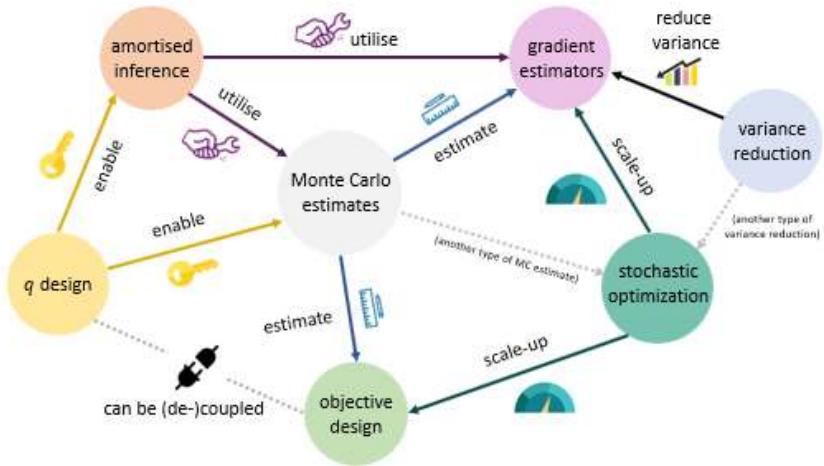
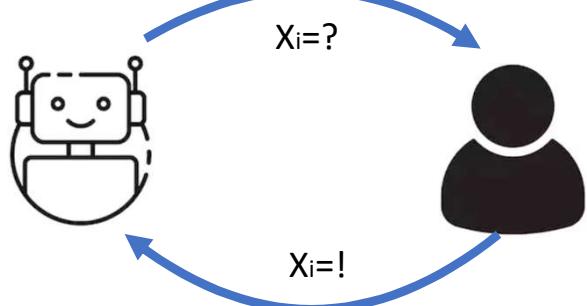
# Summary

The probability distribution  $\pi(\theta)$  is intractable



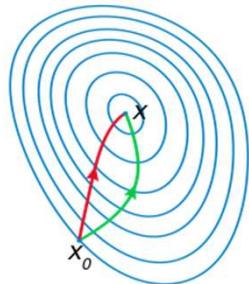
Approximate Inference

$$\int F(\theta) [\pi(\theta)] d\theta$$



# Future Directions: Methodology

Better optimization



Zhang et al. Noisy natural gradient as variational inference. ICML 2018

Khan et al. Fast and Scalable Bayesian Deep Learning by Weight-Perturbation in Adam. ICML 2018

# Future Directions: Methodology

## Better optimization

- $q$  distribution design
- objective design
- amortised inference
- scalable inference

...

## Combined approaches

- rejection sampling
- importance sampling
- SMC, MCMC, Quasi MC

...



- $q$  distribution design
- objective design
- amortised inference
- scalable inference

...

Burda et al. Importance Weighted Auto-encoders. ICLR 2016

Domke and Sheldon. Divide and Couple: Using Monte Carlo Variational Objectives for Posterior Approximation. NeurIPS 2019

# Future Directions: Methodology

## Better optimization

- $q$  distribution design
- objective design
- amortised inference
- scalable inference

...

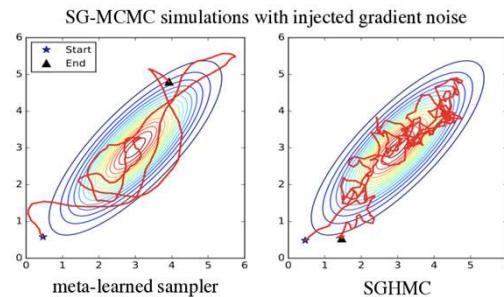
## Combined approaches



- $q$  distribution design
- objective design
- amortised inference
- scalable inference

...

## Meta-learning inference algorithms



Gong et al. Meta-learning for Stochastic Gradient MCMC. ICLR 2019  
Zhang et al. Meta-Learning for Variational Inference. AABI 2019

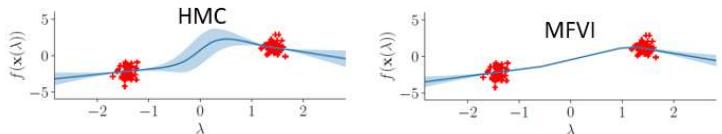
# Future Directions: Error Analyses

## Errors in inference

$$D[q(\theta) || p(\theta|D)] = ?$$

$$D[q(y^*|x^*) || p(y^*|x^*, D)] = ?$$

$$= \int p(y^*|x^*, \theta) q(\theta) d\theta$$



Analysis needed for **deep** probabilistic models!

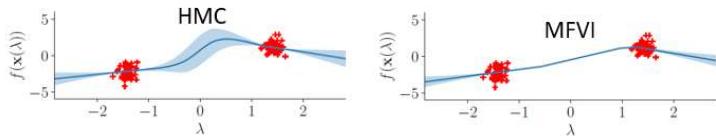
- Optimization error
- Approximation gap

# Future Directions: Error Analyses

## Errors in inference

$$D[q(\theta)||p(\theta|D)] = ?$$

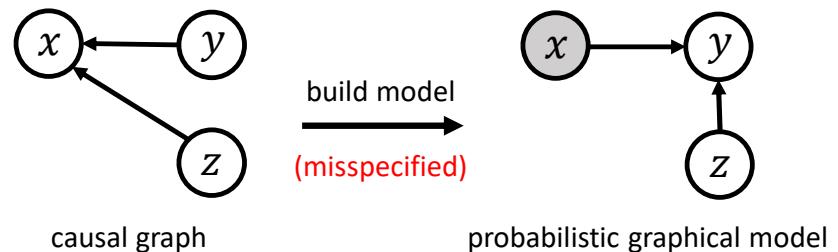
$$D[q(y^*|x^*)||p(y^*|x^*, D)] = ?$$
$$= \int p(y^*|x^*, \theta) q(\theta) d\theta$$



Analysis needed for **deep** probabilistic models!

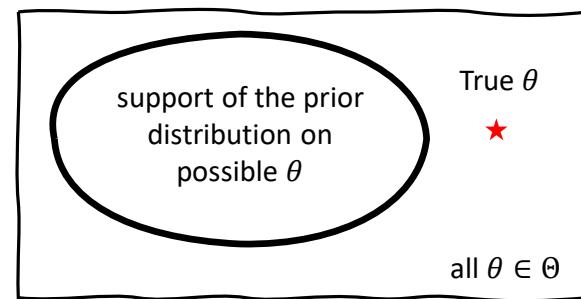
- Optimization error
- Approximation gap

## Model misspecification



causal graph

probabilistic graphical model

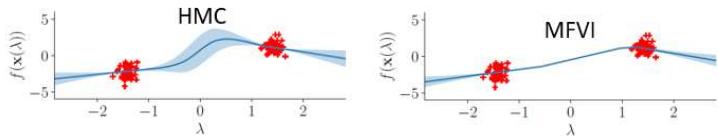


# Future Directions: Error Analyses

## Errors in inference

$$D[q(\theta)||p(\theta|D)] = ?$$

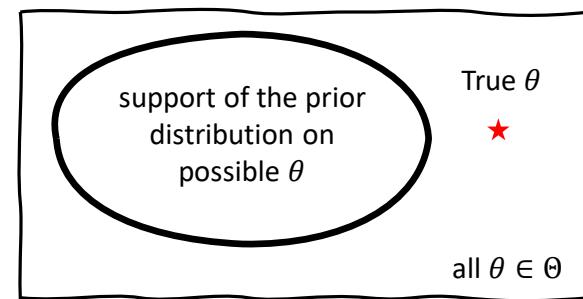
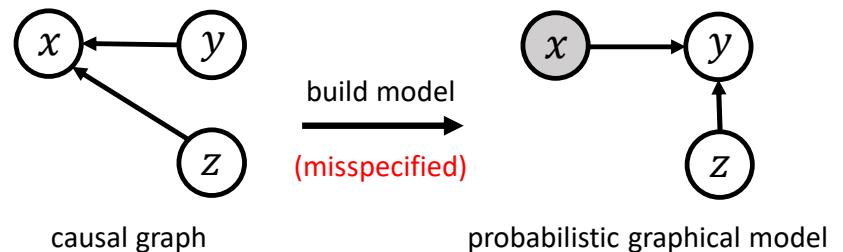
$$D[q(y^*|x^*)||p(y^*|x^*, D)] = ?$$
$$= \int p(y^*|x^*, \theta) q(\theta) d\theta$$



Analysis needed for **deep** probabilistic models!

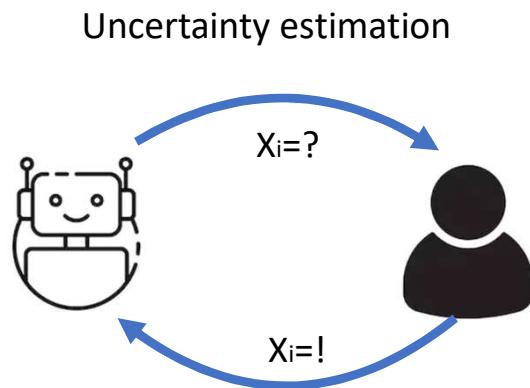
- Optimization error
- Approximation gap

## Model misspecification

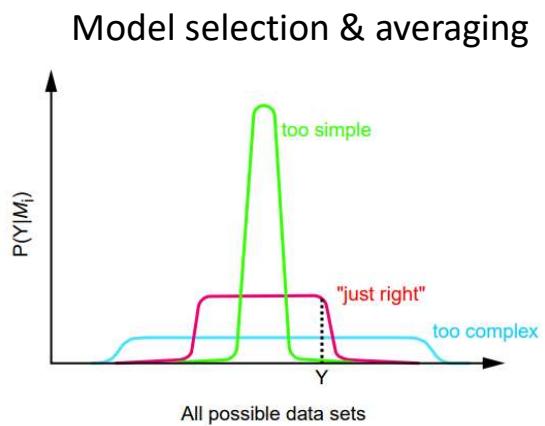
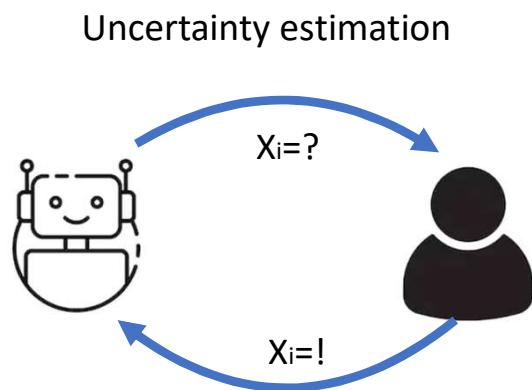


**Separation of inference & modelling?**

# Future Directions: Applications



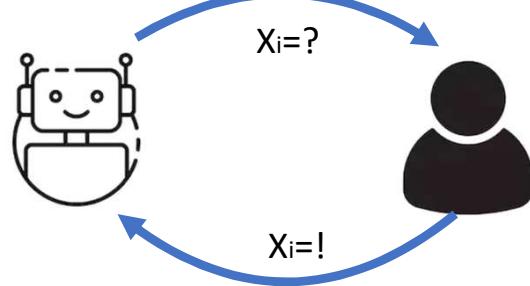
# Future Directions: Applications



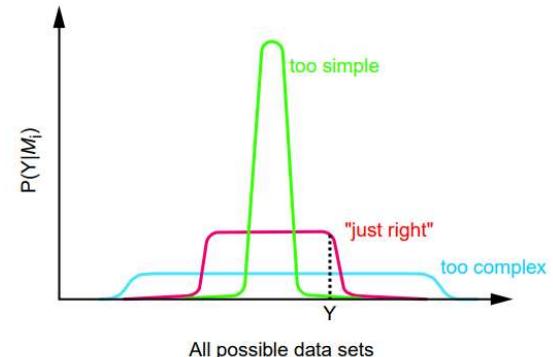
Rasmussen and Ghahramani. Occam's Razor. NeurIPS 2001.

# Future Directions: Applications

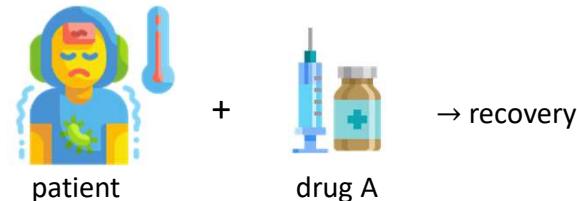
Uncertainty estimation



Model selection & averaging



Causal reasoning



"what if the patient was treated with drug B?"



# Thank You!

Questions? Ask at:

[liyzen2@gmail.com](mailto:liyzen2@gmail.com) (Yingzhen Li)  
[Cheng.Zhang@microsoft.com](mailto:Cheng.Zhang@microsoft.com) (Cheng Zhang)