Advances in Approximate Inference

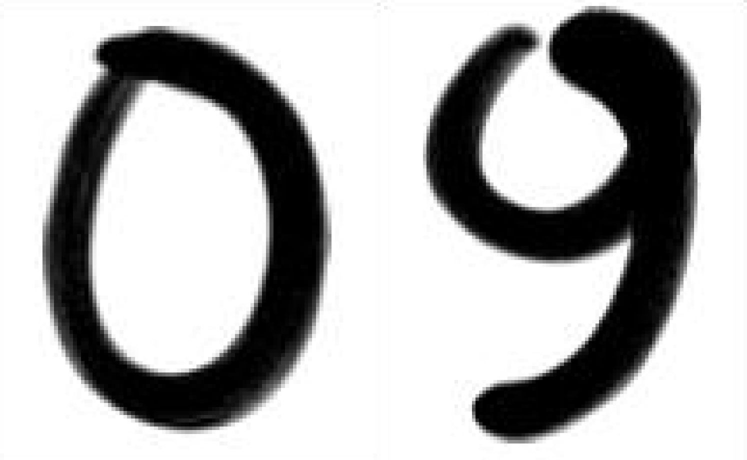
Yingzhen Li, Cheng Zhang

Microsoft Research Cambridge

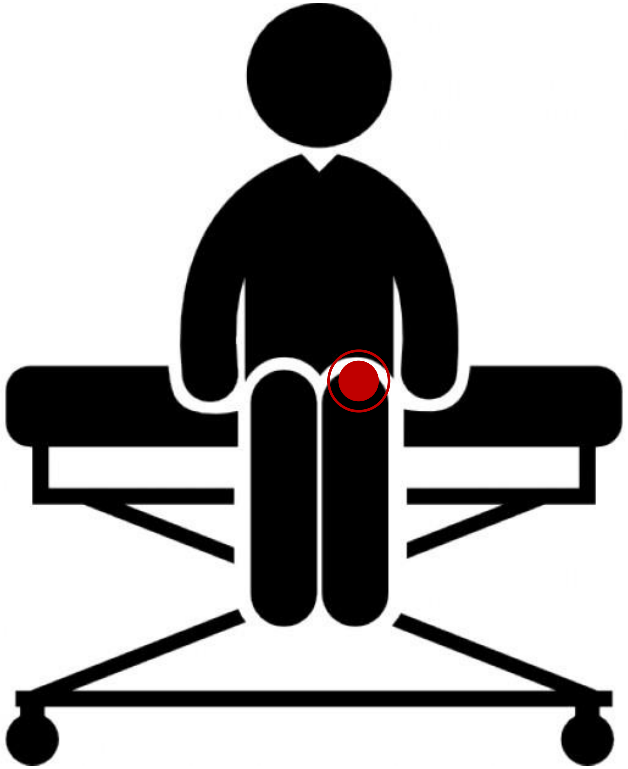
What is the Number?

07? 09?
67? 69?
...

What is the Number?



What is the Diagnosis?

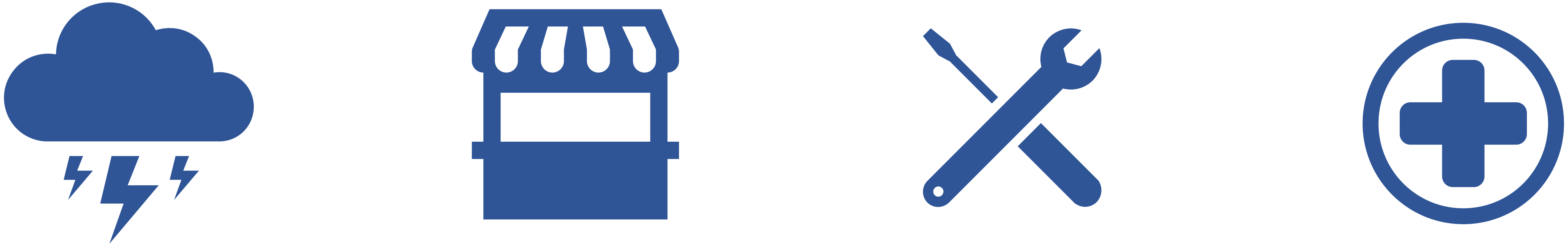


Injury?
Osteoarthritis?
Neuropathic pain?
… …

What is the Diagnosis?

Neuropathic pain
(might have spine injury)

Uncertainty is Important

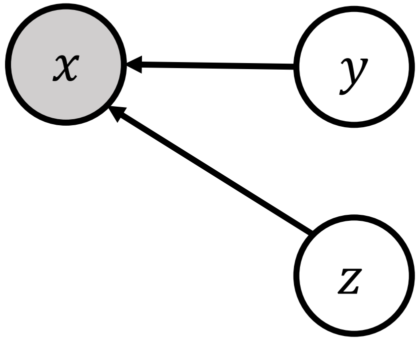


Bayesian ML / Probability Theory

Decision making under uncertainty

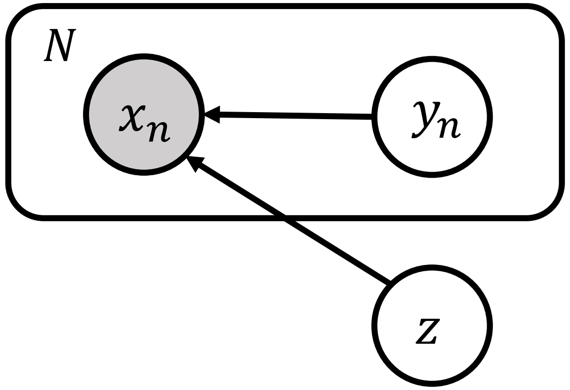
Image courtesy of Sebastian Nowozin
Re-use of the image for any other purpose is not allowed

Graphical Representation



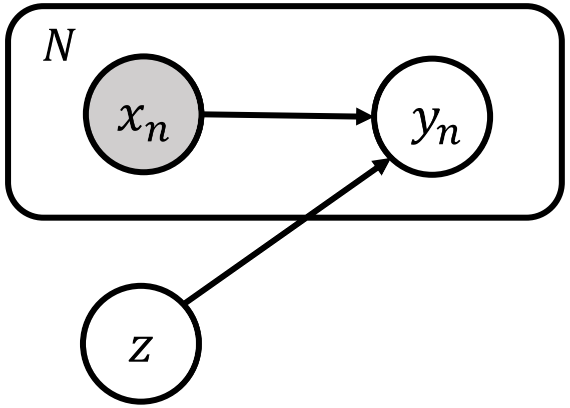
$$p(x, y, z) = p(y)p(z)p(x|y, z)$$

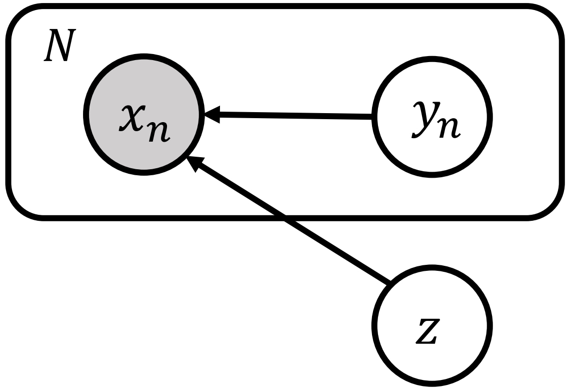
Graphical Representation



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

Graphical Representation

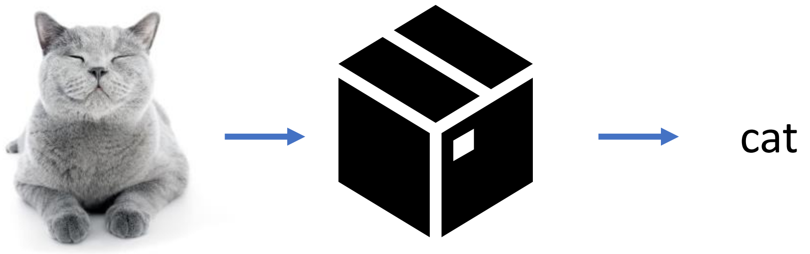




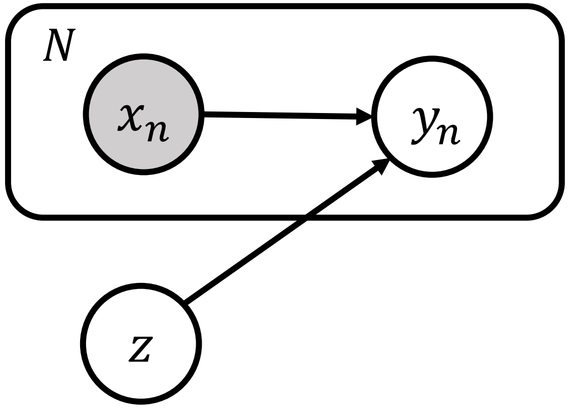
$$p(\boldsymbol{x}, \boldsymbol{y}, z) = p(z) \prod_{n}^{N} p(x_n) P(y_n | x_n, z)$$

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

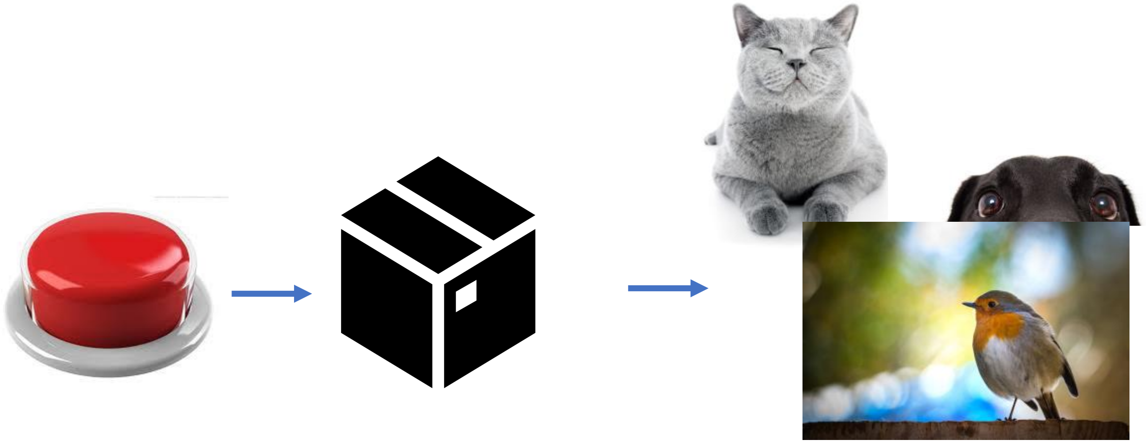
Discriminative Model vs Generative Model

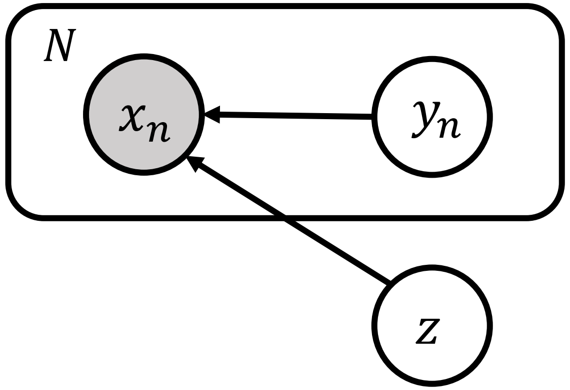


cat



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





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

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)

$$\underbrace{X_1 \qquad X_2}_{X} \qquad Y$$

Discriminative Model Example:

- Bayesian Logistic Regression

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

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)

$$\underbrace{X_1 \qquad X_2}_{X} \qquad Y$$

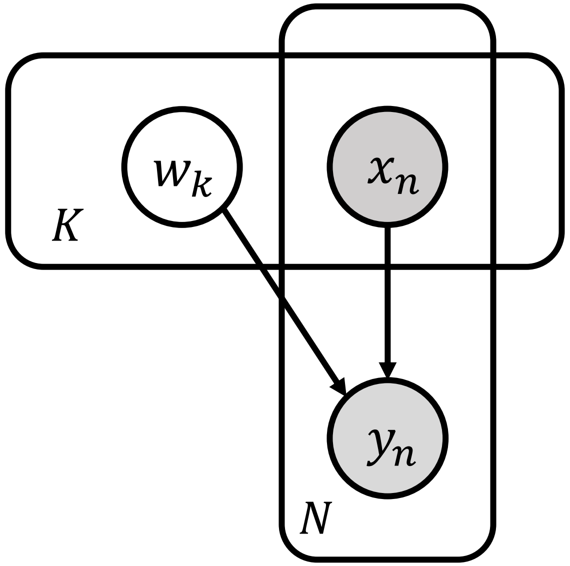
Discriminative Model Example

- 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)

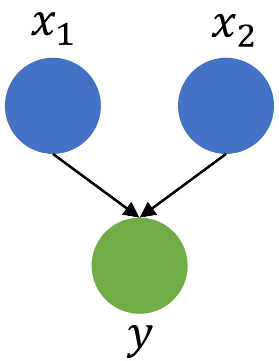
$$\underbrace{X_1 \qquad X_2}_{X} \qquad\qquad Y$$

$$p(y=1) = \frac{1}{1+e^{w_0+w_1x_1+w_2x_2}}$$

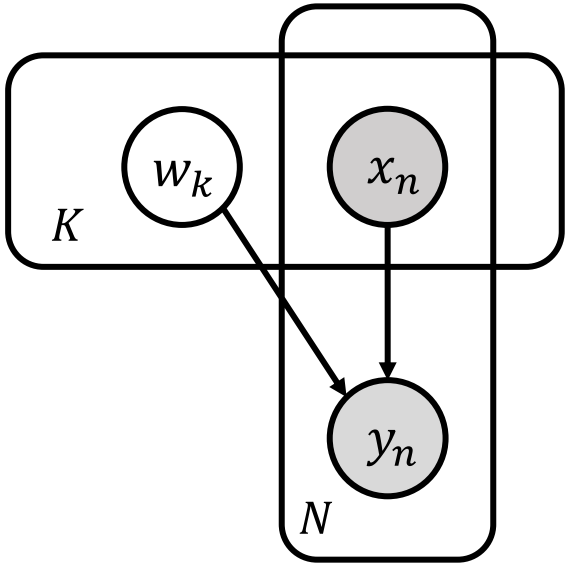


Discriminative Model Example

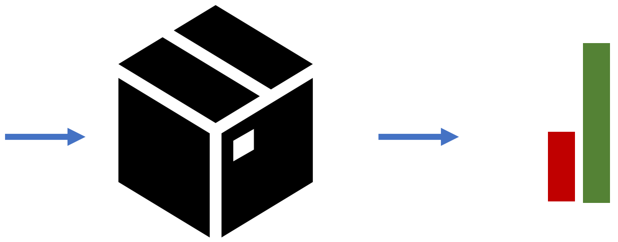
Computation Graph



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

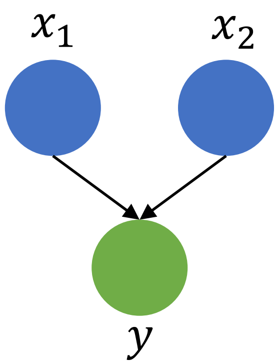


[97, 1]



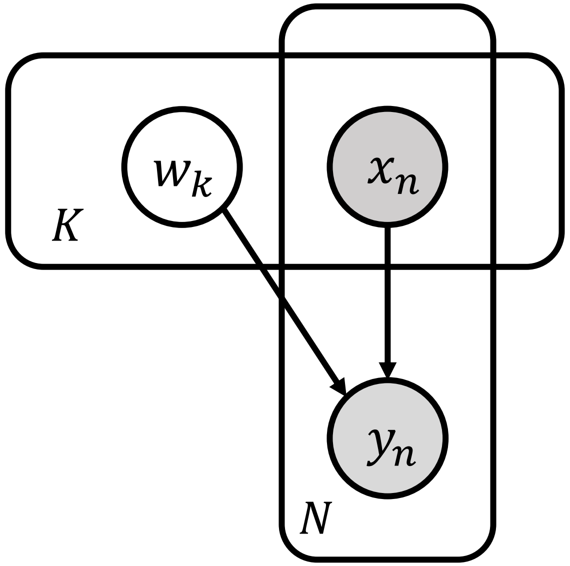
Discriminative Model Example

Computation Graph



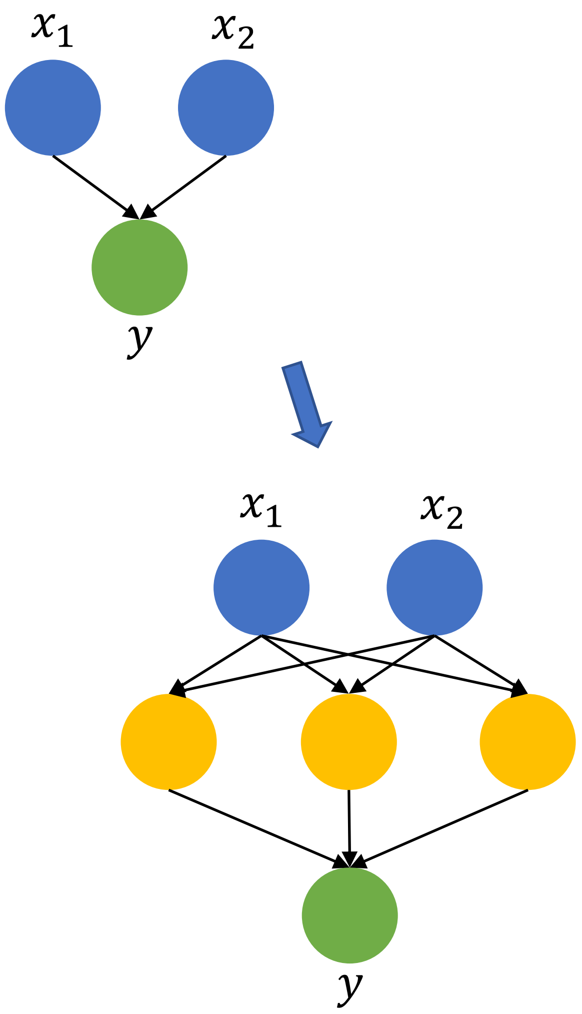
$$\begin{array}{c} \boldsymbol{W}^T X \\ \Big\downarrow \\ \boldsymbol{W}^T \Phi(X), \end{array}$$

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



Discriminative Model Example:

Computation Graph

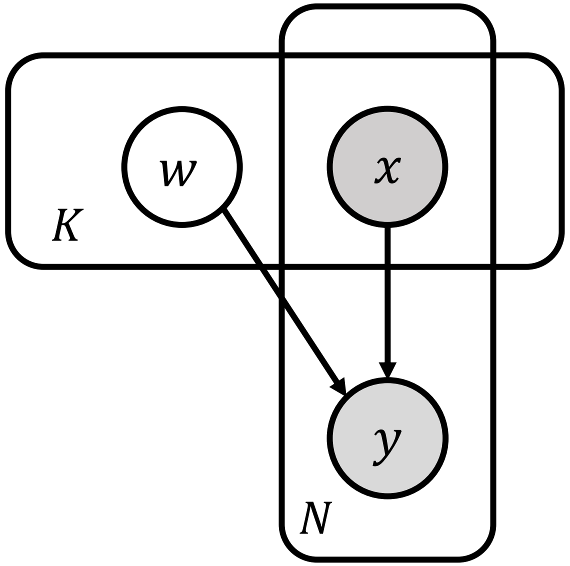


$\boldsymbol{W}^T X$

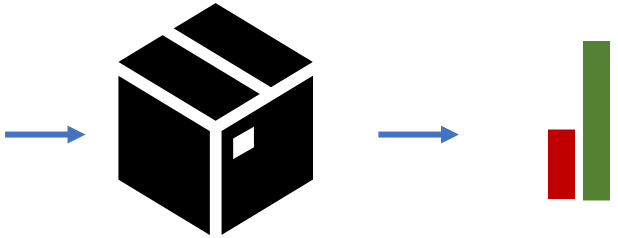
↓

$\boldsymbol{W}^T \Phi(X)$,

$$p(y=1)=\frac{1}{1+\boxed{e^{w_0+w_1x_1+w_2x_2}}}$$

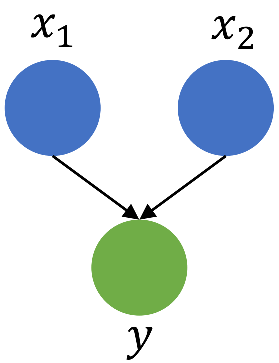


[97, 1]



Discriminative Model Example

Computation Graph

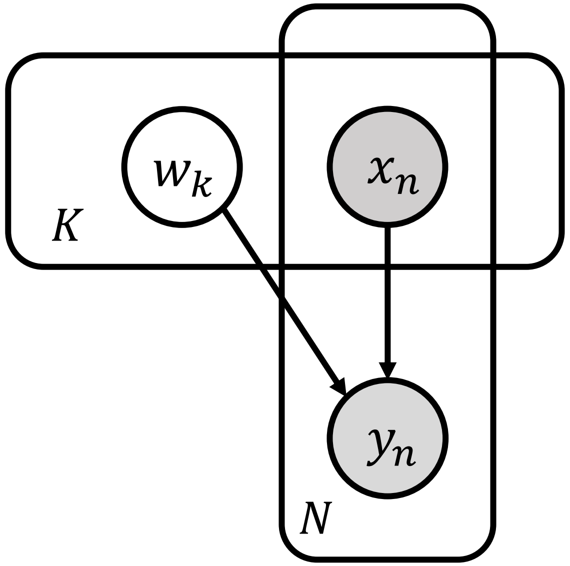


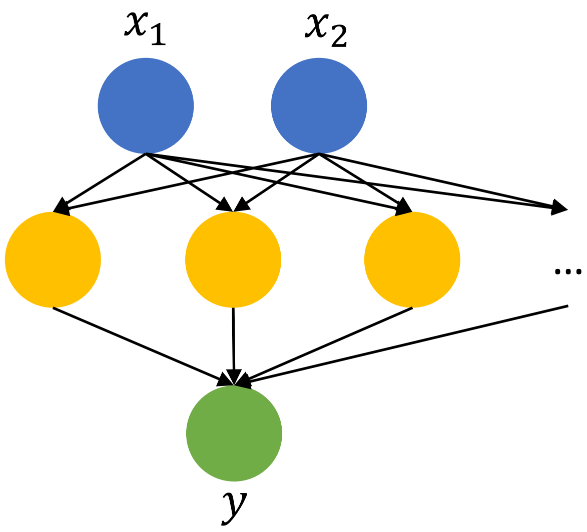
$$p(y=1) = \frac{1}{1+\boxed{e^{w_0+w_1x_1+w_2x_2}}}$$

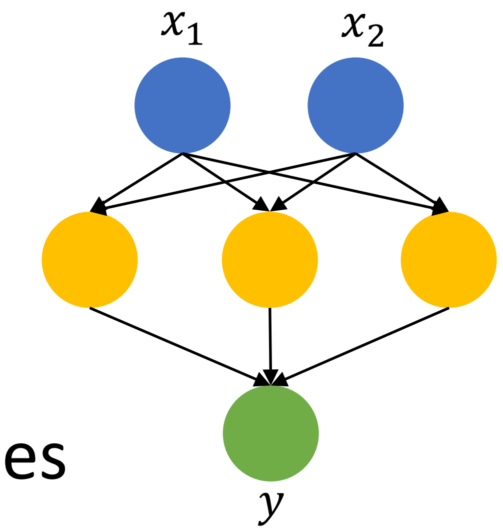
$\boldsymbol{W}^T X$

$\downarrow$

$\boldsymbol{W}^T \Phi(X),$





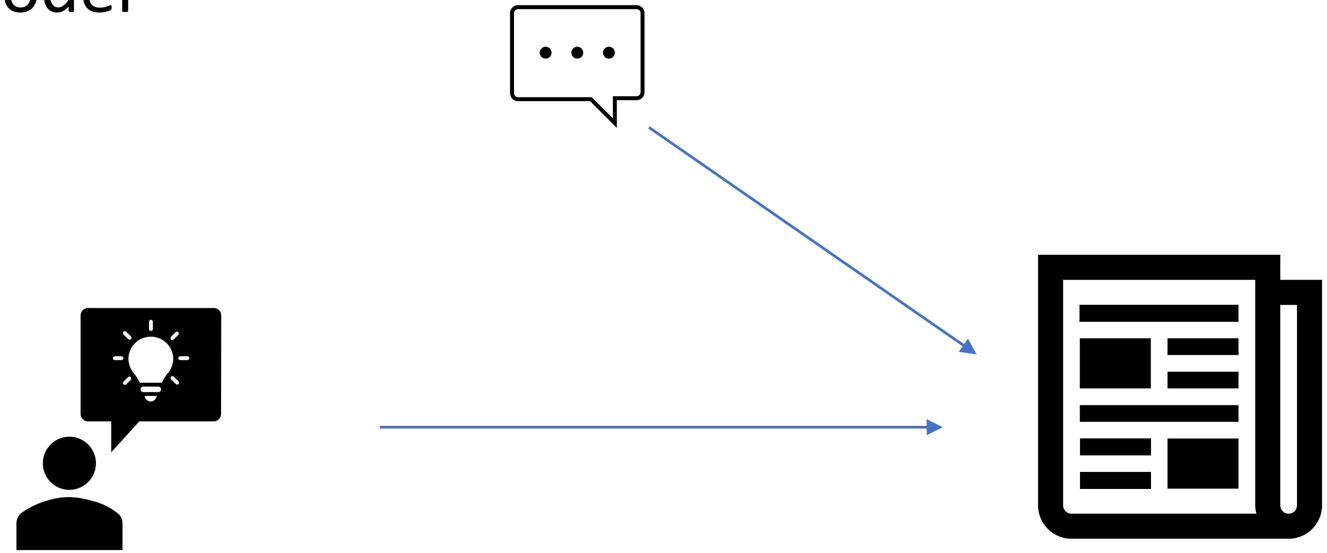


Gaussian Processes

Radford Neal’s PhD thesis (1994)

Generative Model Example

- Topic Model

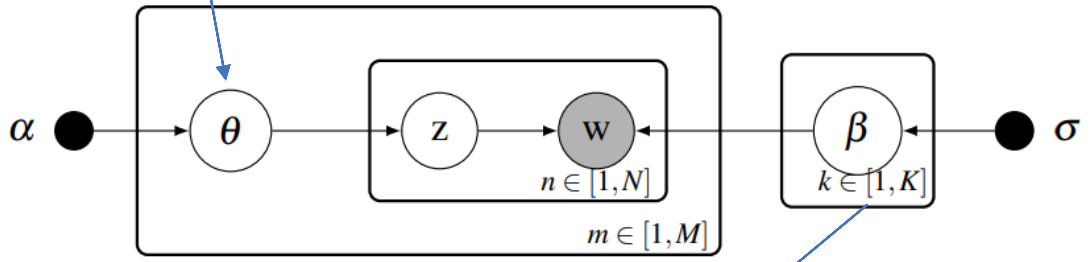


Generative Model Example:

- Latent Dirichlet Allocation

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

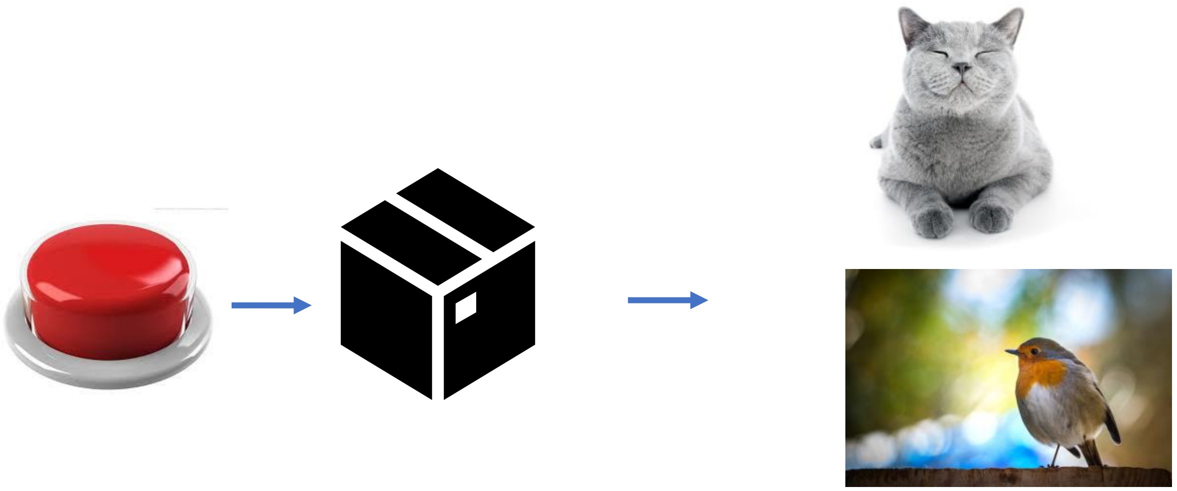
Topic distribution per document:
e.g. 30% “topic model”, 40% “natural language processing”, 30% “interpretability”



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

Blei et al. Latent Dirichlet Allocation. JMLR 2003.

Generative Model Example



Topics	Documents
User embedding	Movie rating
Underlying health conditions	Symptoms

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

$$\int \underbrace{F(\theta)}_{\text{integrand function}}\;\underbrace{\overset{\text{prob. density}}{\boxed{\pi(\overset{\text{Random variable (unobserved)}}{\boxed{\theta}})}}\,d\theta}_{\text{probability measure}}$$

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

Bayesian Inference

$\pi(\theta) = p(\theta|data)$

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

- $P(\theta)$: prior distribution
- $P(data \mid \theta)$: likelihood of θ given $data$
- $P(\theta|\, data)$: posterior distribution of θ given $data$
- $P(data)$: marginal likelihood/model evidence

$$P(data) = \int P(\theta)P(data \mid \theta)$$

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

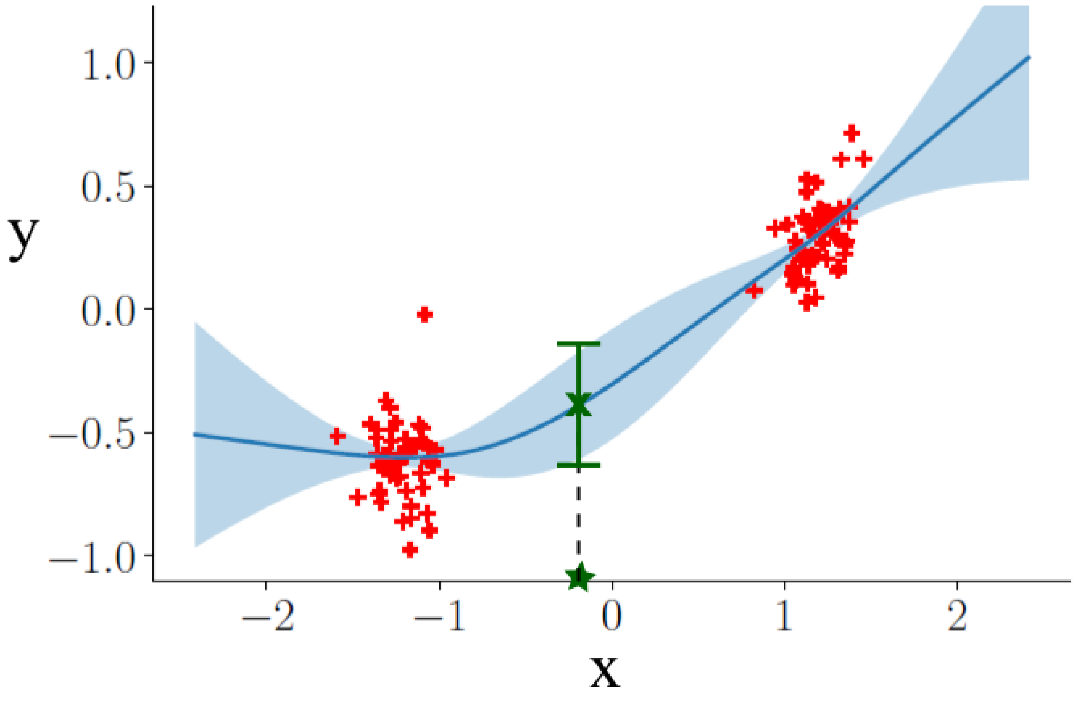
Computation Challenge

- 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?”

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



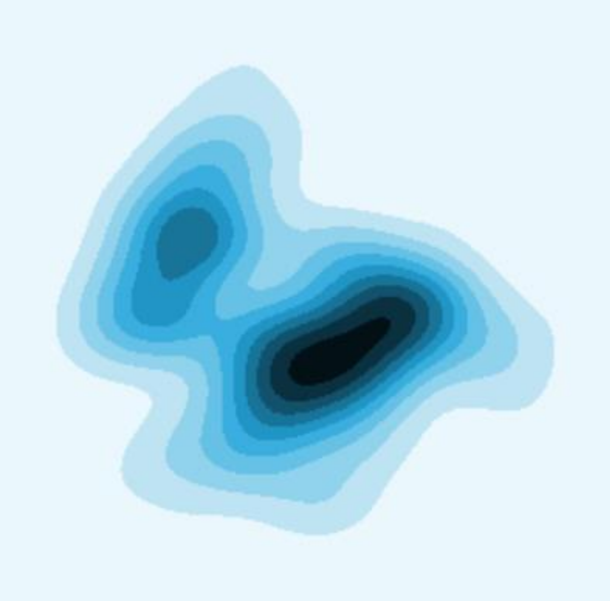
Computation Challenge

- 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



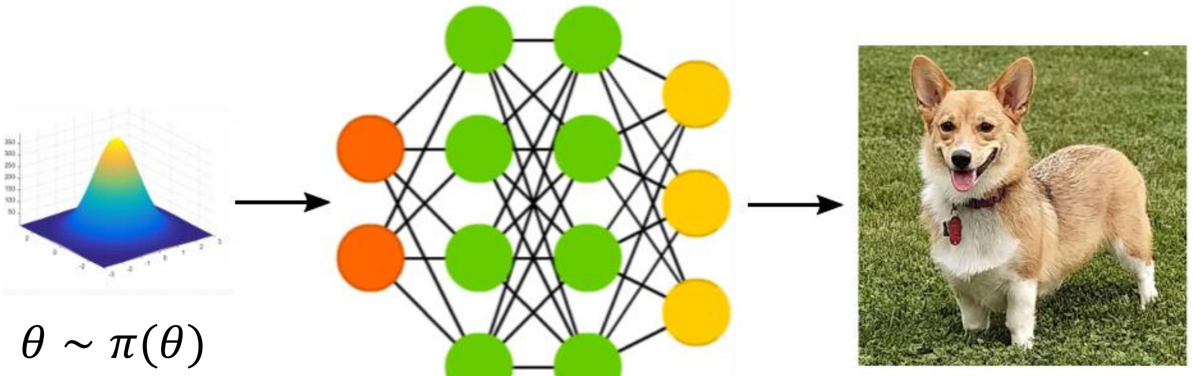
Computation Challenge

- The central equation for inference:

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

“What is the probability of generating this image?”

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



$\theta \sim \pi(\theta)$

x_0

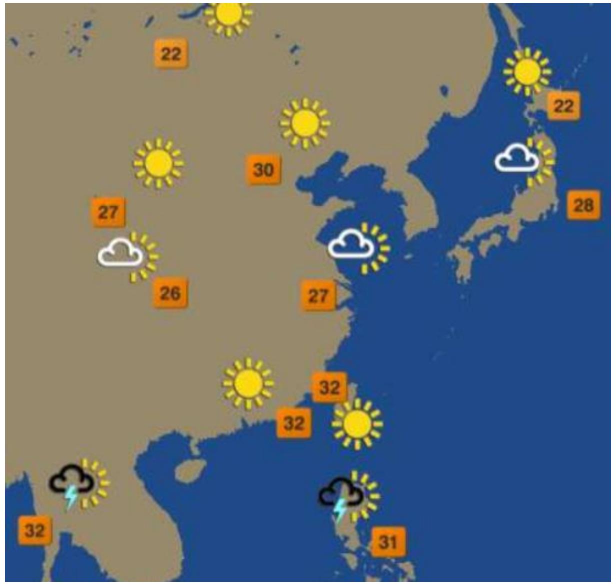
Computation Challenge

- 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:
θ: forecasting simulator settings
D: historical weather record
$F(\theta) = Simulator(\theta), \pi(\theta) = p(\theta \mid D)$



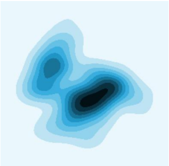
Nature laughs at the difficulties of integration.

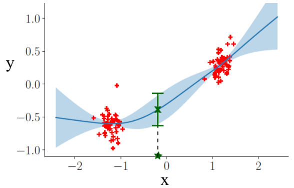
--Pierre-Simon Laplace

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

Integration in Bayesian Computation

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





Approximate Inference

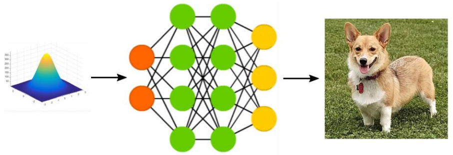
(in a strict sense)

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

This tutorial

Integration in Bayesian Computation

The integrand $F(\theta)$ is intractable

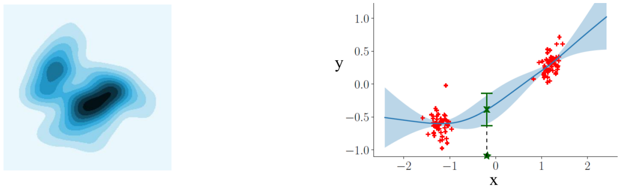


Implicit Models

Bayesian Optimisation, Probabilistic Numerics

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

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



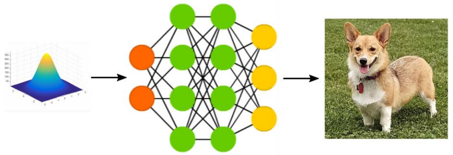
Approximate Inference

(in a strict sense)

This tutorial

Integration in Bayesian Computation

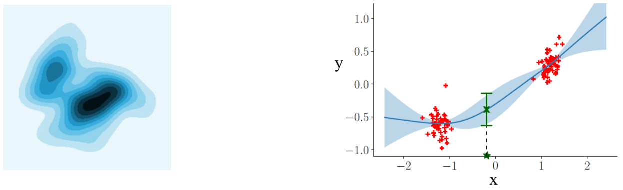
The integrand $F(\theta)$ is intractable



Implicit Models

Bayesian Optimization, Probabilistic Numerics

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



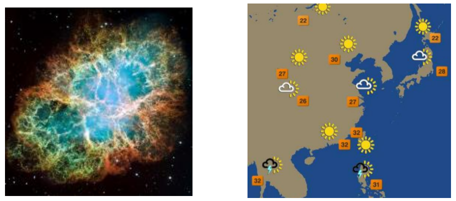
Approximate Inference

(in a strict sense)

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

This tutorial

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.)

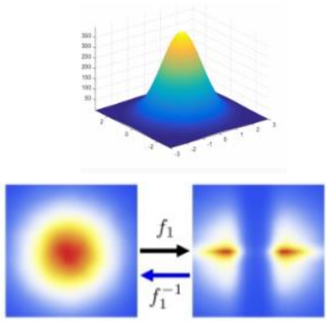
Approximate Inference

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

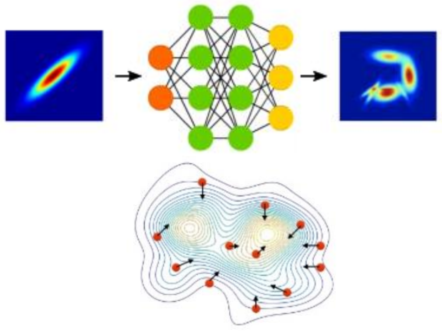
😇 😈

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

Approximate distribution design



Explicit distributions



Implicit distributions

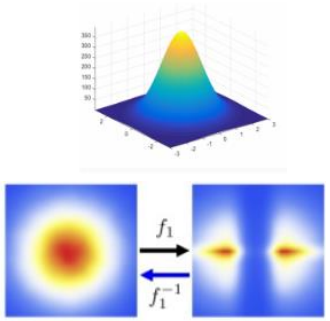
Approximate Inference

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

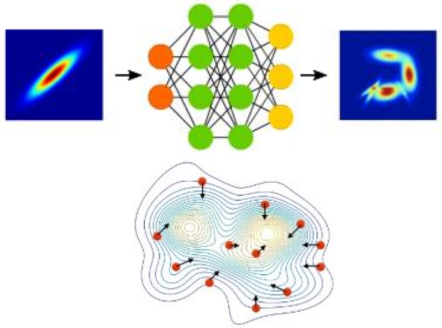
😇 😈

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

Approximate distribution design



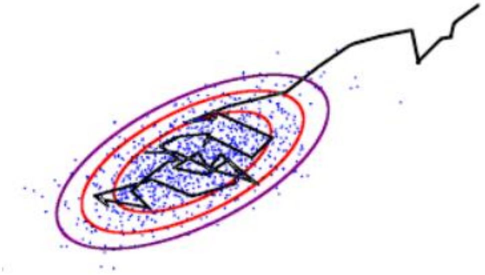
Explicit distributions



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

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

- $P(\theta)$: prior
- $P(D \mid \theta)$: 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)

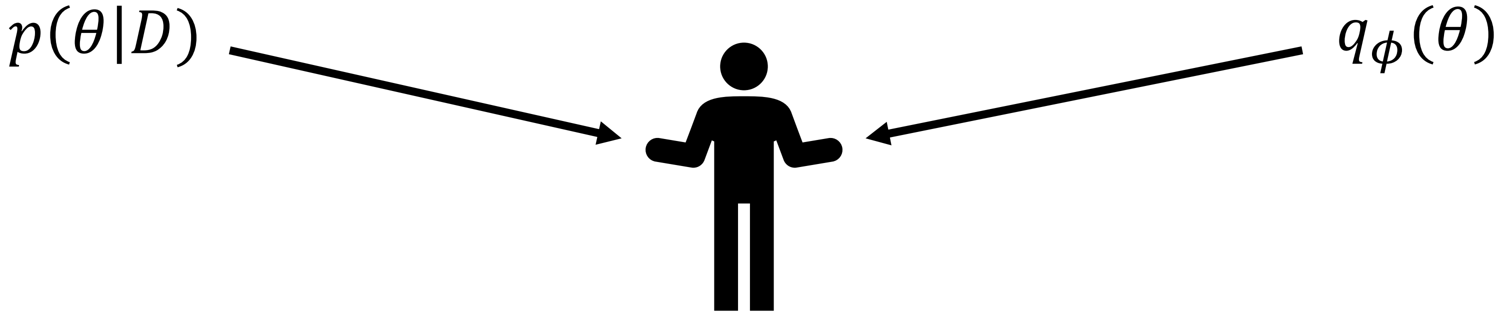
The posterior

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

The variational distribution

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

Inference as Optimization



$p(\theta|D)$

$q_\phi(\theta)$

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

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

Let’s Derive the Objective of VI

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

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

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

Let’s Derive the Objective of VI

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

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

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

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

Let’s Derive the Objective of VI

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

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

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

Let’s Derive the Objective of VI

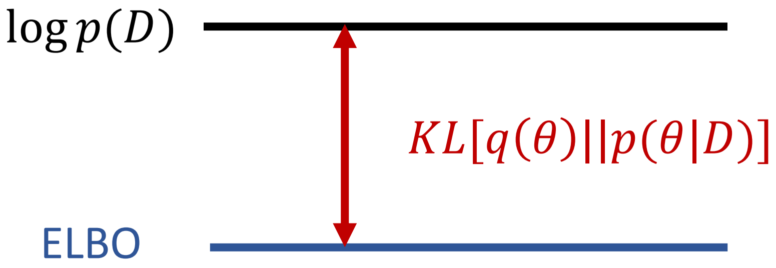
Minimize $KL[q(\theta)||p(\theta|D)]$

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

Model Evidence

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

Evidence Lower Bound (ELBO)



$\log p(D)$

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

ELBO

“Model Evidence = ELBO + KL”

Alternative Derivation

Let's start with the model evidence

$$\log p(D)$$

Alternative Derivation

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

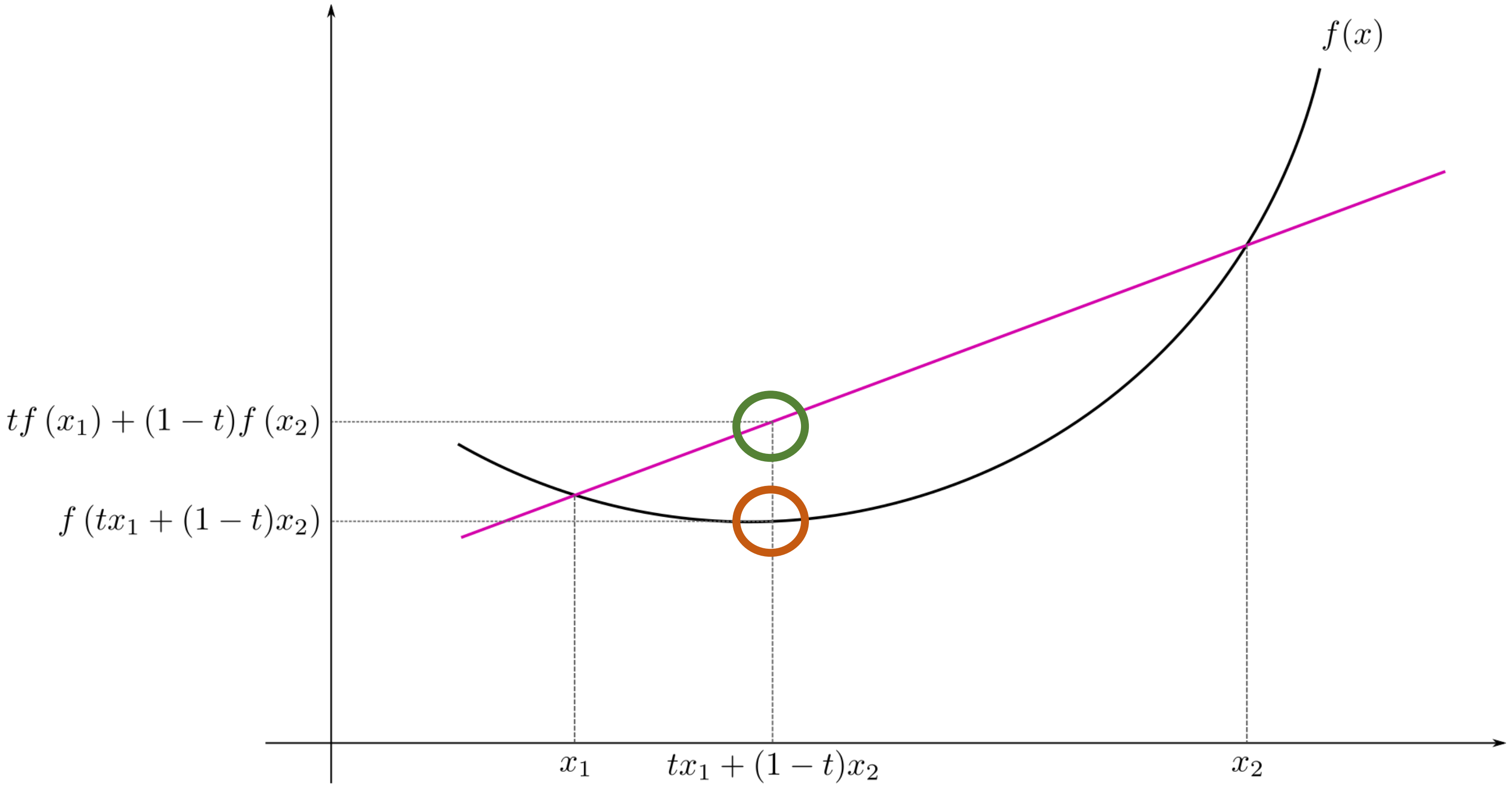
Alternative Derivation

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

$$= \log \int \frac{\mathrm{p}(\theta, D)\, q(\theta)}{q(\theta)}\, d\theta$$

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

Jensen’s Inequality



If f is a convex function, then

$$\boxed{f(E[X])} \leq \boxed{E[f(X)]}$$

If f is a concave function, then

$$f(E[X]) \geq E[f(X)]$$

Image: Wikipedia

Alternative Derivation

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

$$= \log \int \frac{\mathrm{p}(\theta, D)\ q(\theta)}{q(\theta)}\ d\theta$$

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

Jensen's inequality → $$\geq E_{q(\theta)}\left[\log \frac{\mathrm{p}(\theta, D)}{q(\theta)}\right]$$

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

Alternative Derivation

Model Evidence

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

$$= \log \int \frac{\mathrm{p}(\theta, D)\ q(\theta)}{q(\theta)}\ d\theta$$

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

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

Evidence Lower Bound (ELBO)

Variational Inference (VI)

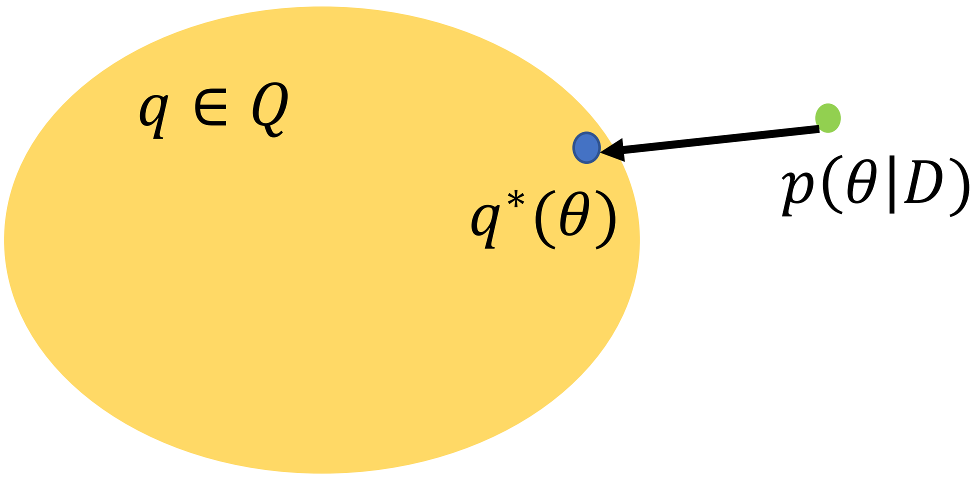
The posterior

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

The variational distribution

$$q_\phi(\theta)$$

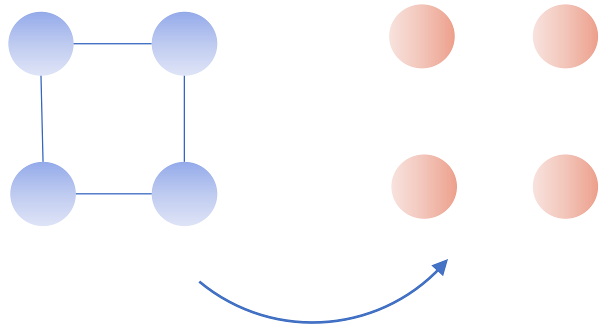
$$L = E_{q_{\phi(\theta)}}\left[\log \frac{\mathrm{p(D},\theta)}{q_\phi(\theta)}\right] = \log p(D) - KL[\, q_\phi(\theta)||p(\theta)]$$



Mean-field Variational Inference

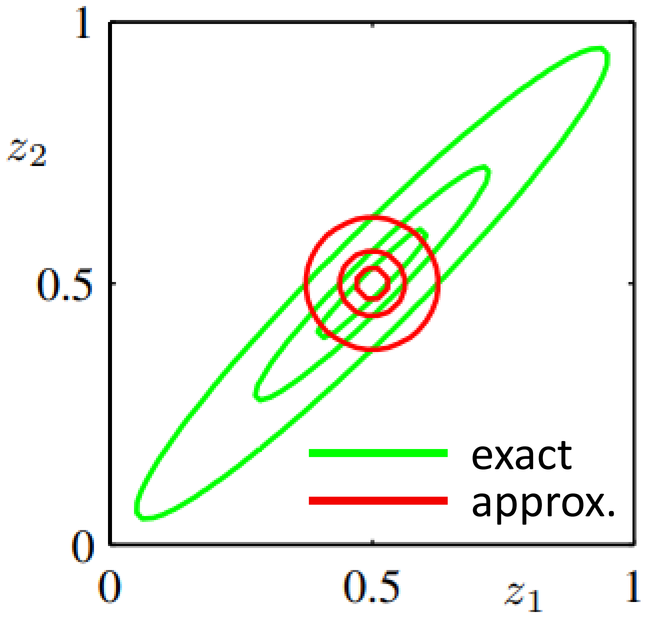
- A type of choices of the variational distribution
- The name origins in the mean field theory of physics
- The variational distribution factorizes

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



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

A Gaussian Example



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

$$q(z) = q(z_1)q(z_2)$$

Bishop (2006). Pattern recognition and machine learning. Springer.

Mean-field Variational Inference

ELBO

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

Fully Factorized Variational Distribution

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

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

Mean-field Variational Inference

ELBO

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

←

Fully Factorized Variational Distribution

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

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

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

Mean-field Variational Inference

ELBO

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

←

Fully Factorized Variational Distribution

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

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

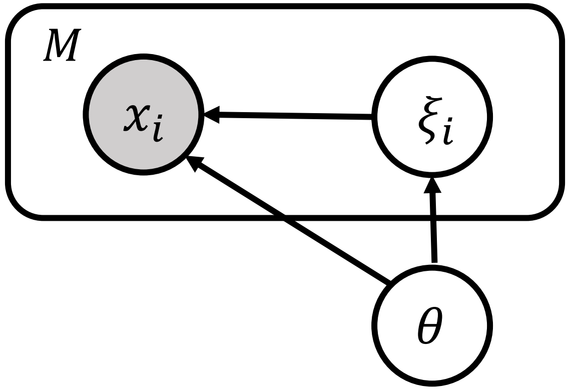
$$q^*(\theta_j) \propto \exp(E_{q(\theta_{\neg j})}[\log p(\theta_j, D \mid \theta_{\neg j})])$$

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

Part II: Advances

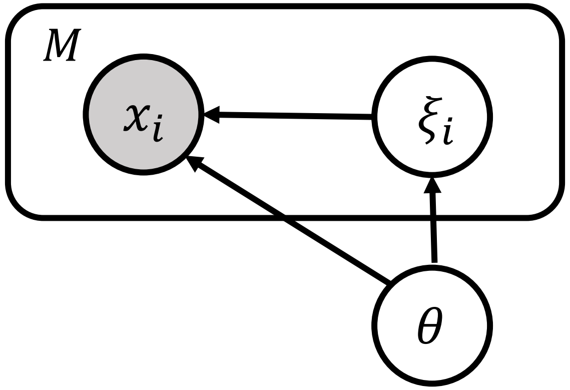
- **Scalable variational inference**
- **Monte Carlo methods**
- **Amortized inference**
- Approximate distribution design
- Optimization objective design

Stochastic Variational Inference



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

Stochastic Variational Inference



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

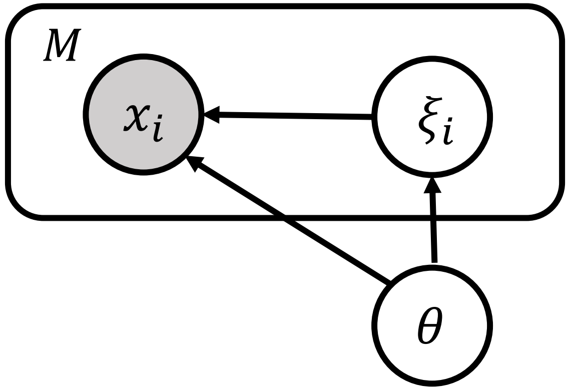
$$L = E_q\left[\log \frac{\mathrm{p}(\theta, \boldsymbol{\xi}, \boldsymbol{x})}{q(\theta, \xi)}\right]$$

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

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

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

Stochastic Variational Inference



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

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

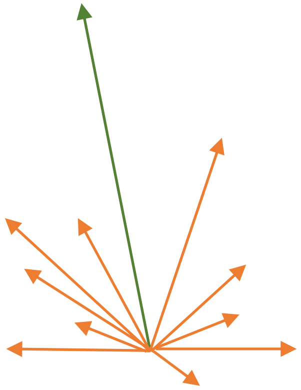
stochastic approximation
with $S \ll M$

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

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

Hoffman et al. Stochastic Variational Inference. JMLR 2013.

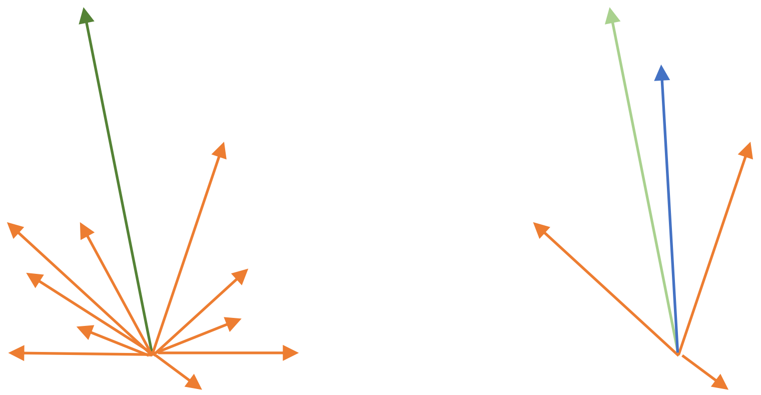
How Stochastic Gradient Works



→ $\nabla F(x_i)$ gradient of each single data point x_i

→ $E_x[\nabla F(x)]$ batch gradient considering all data points

How Stochastic Gradient Works

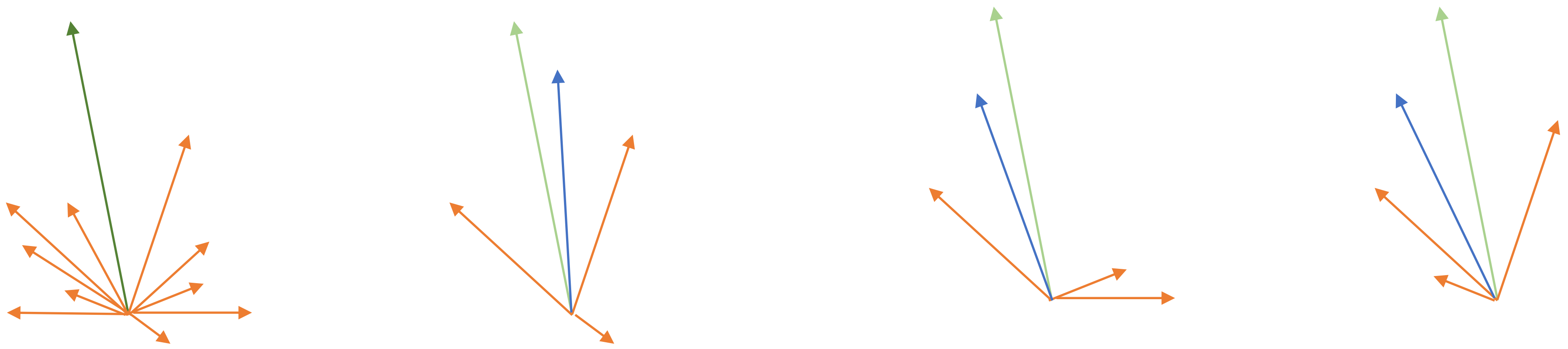


→ $\nabla F(x_i)$ gradient of each single data point x_i

→ $E_x[\nabla F(x)]$ 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

→ $E_x[\nabla F(\boldsymbol{x})]$ 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

$$\begin{tikzcd}
L = E_q\left[\log\frac{p(\theta)}{q(\theta)}\right] + \sum_{i=1}^{M} E_q\left[\log\frac{p(\xi_i|\theta)p(x_i|\xi_i)}{q(\xi_i)}\right] \arrow[r, "\text{gradient}"] \arrow[d, "\text{Stochastic approximation with } S \ll M"] & \nabla L = \nabla E_q\left[\log\frac{p(\theta)}{q(\theta)}\right] + \sum_{i=1}^{M} E_q\left[\nabla\log\frac{p(\xi_i|\theta)p(x_i|\xi_i)}{q(\xi_i)}\right] \arrow[d, "\text{Stochastic approximation with } S \ll M"] \\
\hat{L} = E_q\left[\log\frac{p(\theta)}{q(\theta)}\right] + \frac{M}{S}\sum_{i=1}^{S} E_q\left[\log\frac{p(\xi_i|\theta)p(x_i|\xi_i)}{q(\xi_i)}\right] \arrow[r, "\text{gradient}"] & \nabla\hat{L} = \nabla E_q\left[\log\frac{p(\theta)}{q(\theta)}\right] + \frac{M}{S}\sum_{i=1}^{S} E_q\left[\nabla\log\frac{p(\xi_i|\theta)p(x_i|\xi_i)}{q(\xi_i)}\right]
\end{tikzcd}$$

Stochastic Gradient

Hoffman et al. Stochastic Variational Inference. JMLR 2013.

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 $x_1, x_2, \ldots . . , x_K \sim p(x)$
    2. Evaluate $f(x_i)$ for each sample
    3. Compute $E[f(x)] \approx \underbrace{\frac{1}{K} \sum_{i=1}^{K} f(x_i)}_{\text{Unbiased Monte Carlo estimate}}$

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

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

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

REINFORCE Gradients

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

Gradient of the ELBO

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

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

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

Gradient of the ELBO

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

$$= \int \nabla_\phi q_{\phi}(\theta)\log\frac{\mathrm{p}(\theta,D)}{q_\phi(\theta)}\, d\theta + \int q_\phi(\theta)\, \nabla_\phi \log\frac{\mathrm{p}(\theta,D)}{q_\phi(\theta)}\, d\theta$$

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

Log-derivative trick

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

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

Gradient of the ELBO

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

$$= \int \underline{\nabla_\phi q_\phi(\theta)} \log \frac{\mathrm{p}(\theta,D)}{q_\phi(\theta)}\, d\theta + \int q_\phi(\theta)\, \nabla_\phi \log \frac{\mathrm{p}(\theta,D)}{q_\phi(\theta)}\, d\theta$$

$$= \int \underline{q_\phi(\theta) \nabla_\phi \log q_\phi(\theta)}\ \log \frac{\mathrm{p}(\theta,D)}{q_\phi(\theta)}\, d\theta - \int \nabla_\phi q_\phi(\theta) d\theta$$

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

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

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

Gradient of the ELBO

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

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

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

Gradient of the ELBO

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

$$= \int \nabla_\phi q_\phi(\theta) \log \frac{\mathrm{p}(\theta,D)}{q_\phi(\theta)} d\theta + \int q_\phi(\theta)\, \nabla_\phi \log \frac{\mathrm{p}(\theta,D)}{q_\phi(\theta)} d\theta$$

$$= \int q_\phi(\theta) \nabla_\phi \log q_\phi(\theta)\ \log \frac{\mathrm{p}(\theta,D)}{q_\phi(\theta)} d\theta - \underline{\int \nabla_\phi q_\phi(\theta) d\theta}$$

$$= \nabla_\phi \int q_\phi(\theta) d\theta = \nabla_\phi 1 = 0$$

$$= \mathrm{E}_{q_\phi(\theta)}\left[\underbrace{\nabla_\phi \log q_\phi(\theta)}_{\text{Score function}} \log \frac{\mathrm{p}(\theta,D)}{q_\phi(\theta)}\right]$$

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

$f(\theta)$

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

- To approximate: $E[f(x)]$
- MC Approximation:
  1. Sample $x_1, x_2, \ldots., x_K \sim p(x)$
  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

$f(\theta)$

$$\nabla_\phi L = E_{q_\phi(\theta)}\left[\boxed{\underbrace{\nabla_\phi \log q_\phi(\theta)}_{\text{Score function}} \log\frac{p(\theta,D)}{q_\phi(\theta)}}\right]$$

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

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

- To approximate: $E[f(x)]$
- MC Approximation:
  1. Sample $x_1, x_2, \ldots\ldots, x_K \sim p(x)$
  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 $\theta_1, \theta_2, \ldots\ldots, \theta_K \sim q_\phi(\theta)$

Ranganath et al. Black box variational inference. AISTATS 2014

BBVI

Gradient of the ELBO

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

- To approximate: $E[f(x)]$
- MC Approximation:
  1. Sample $x_1, x_2, \ldots.., x_K \sim p(x)$
  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 $\theta_1, \theta_2, \ldots.., \theta_K \sim q_\phi(\theta)$
2. Evaluate $\boxed{\nabla_\phi \log q_\phi(\theta) \log \frac{\mathrm{p}(\theta,D)}{q_\phi(\theta)}}$ for each sample

Ranganath et al. Black box variational inference. AISTATS 2014

BBVI

- To approximate: $E[f(x)]$
- MC Approximation:
  1. Sample $x_1, x_2, \ldots.., x_K \sim p(x)$
  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

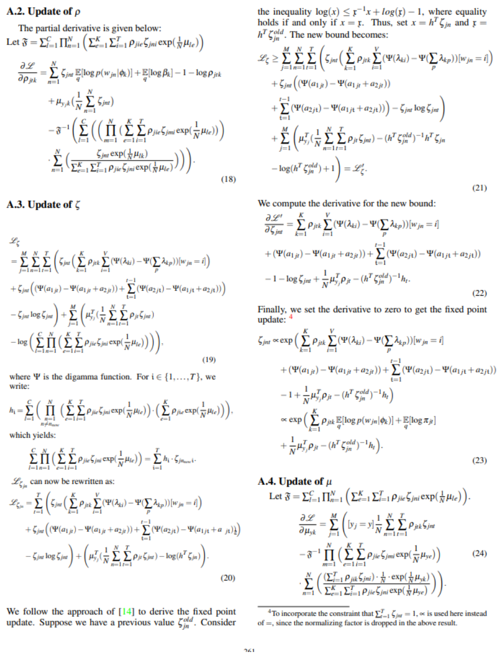
$$\nabla_\phi L = E_{q_\phi(\theta)}\left[\boxed{\nabla_\phi \log q_\phi(\theta) \log \frac{p(\theta,D)}{q_\phi(\theta)}}\right]$$

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

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

Ranganath et al. Black box variational inference. AISTATS 2014

Black-box Variational Inference (BBVI)



A.2. Update of ρ

The partial derivative is given below:

Let $\mathfrak{F} = \sum_{l=1}^{C} \prod_{n=1}^{N} \Big(\sum_{e=1}^{K} \sum_{i=1}^{T} \rho_{jie} \zeta_{jni} \exp(\frac{1}{N}\mu_{le}) \Big)$

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial \rho_{jtk}} &= \sum_{n=1}^{N} \zeta_{jnt} \mathop{\mathbb{E}}_{q}[\log p(w_{jn}|\phi_k)] + \mathop{\mathbb{E}}_{q}[\log \beta_k] - 1 - \log \rho_{jtk} \\ &+ \mu_{y_jk}(\frac{1}{N}\sum_{n=1}^{N} \zeta_{jnt}) \\ &- \mathfrak{F}^{-1}\Bigg(\sum_{l=1}^{C}\Bigg(\Bigg(\prod_{m=1}^{N}(\sum_{e=1}^{K}\sum_{i=1}^{T}\rho_{jie}\zeta_{jmi}\exp(\frac{1}{N}\mu_{le}))\Bigg) \\ &\cdot \sum_{n=1}^{N}\Big(\frac{\zeta_{jnt}\exp(\frac{1}{N}\mu_{lk})}{\sum_{e=1}^{K}\sum_{i=1}^{T}\rho_{jie}\zeta_{jni}\exp(\frac{1}{N}\mu_{le})}\Big)\Bigg)\Bigg).\end{aligned} \tag{18}$$

A.3. Update of ζ

$$\begin{aligned}&\mathscr{L}_\zeta \\ &= \sum_{j=1}^{M}\sum_{n=1}^{N}\sum_{t=1}^{T}\Bigg(\zeta_{jnt}\Big(\sum_{k=1}^{K}\rho_{jtk}\sum_{i=1}^{V}(\Psi(\lambda_{ki}) - \Psi(\sum_{p}\lambda_{kp}))[w_{jn}=i]\Big) \\ &+ \zeta_{jnt}\Big((\Psi(a_{1jt}) - \Psi(a_{1jt}+a_{2jt})) + \sum_{\mathrm{t}=1}^{t-1}(\Psi(a_{2j\mathrm{t}}) - \Psi(a_{1j\mathrm{t}}+a_{2j\mathrm{t}}))\Big) \\ &- \zeta_{jnt}\log\zeta_{jnt}\Bigg) + \sum_{j=1}^{M}\Bigg(\mu_{y_j}^{T}(\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T}\rho_{jt}\zeta_{jnt}) \\ &- \log\Bigg(\sum_{l=1}^{C}\prod_{n=1}^{N}\Big(\sum_{e=1}^{K}\sum_{i=1}^{T}\rho_{jie}\zeta_{jni}\exp(\frac{1}{N}\mu_{le})\Big)\Bigg)\Bigg),\end{aligned} \tag{19}$$

where Ψ is the digamma function. For $\mathrm{i} \in \{1,\ldots,T\}$, we write:

$$h_{\mathrm{i}} = \sum_{l=1}^{C}\Big(\prod_{\substack{n=1\\ n\neq n_{now}}}^{N}\Big(\sum_{e=1}^{K}\sum_{i=1}^{T}\rho_{jie}\zeta_{jni}\exp(\frac{1}{N}\mu_{le})\Big)\cdot\Big(\sum_{e=1}^{K}\rho_{ji e}\exp(\frac{1}{N}\mu_{le})\Big)\Big),$$

which yields:

$$\sum_{l=1}^{C}\prod_{n=1}^{N}\Big(\sum_{e=1}^{K}\sum_{i=1}^{T}\rho_{jie}\zeta_{jni}\exp(\frac{1}{N}\mu_{le})\Big) = \sum_{\mathrm{i}=1}^{T} h_{\mathrm{i}}\cdot\zeta_{jn_{now}\mathrm{i}}.$$

$\mathscr{L}_{\zeta_{jn}}$ can now be rewritten as:

$$\begin{aligned}\mathscr{L}_{\zeta_{jn}} &= \sum_{t=1}^{T}\Bigg(\zeta_{jnt}\Big(\sum_{k=1}^{K}\rho_{jtk}\sum_{i=1}^{V}(\Psi(\lambda_{ki}) - \Psi(\sum_{p}\lambda_{kp}))[w_{jn}=i]\Big) \\ &+ \zeta_{jnt}\Big((\Psi(a_{1jt}) - \Psi(a_{1jt}+a_{2jt})) + \sum_{\mathrm{t}=1}^{t-1}(\Psi(a_{2j\mathrm{t}}) - \Psi(a_{1j\mathrm{t}}+a\ _{j\mathrm{t}}))_{2}\Big) \\ &- \zeta_{jnt}\log\zeta_{jnt}\Bigg) + \Bigg(\mu_{y_j}^{T}(\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T}\rho_{jt}\zeta_{jnt}) - \log(h^T\zeta_{jn})\Bigg).\end{aligned} \tag{20}$$

We follow the approach of [14] to derive the fixed point update. Suppose we have a previous value ζ_{jn}^{old}. Consider the inequality $\log(x) \leq \mathfrak{r}^{-1}x + log(\mathfrak{r}) - 1$, where equality holds if and only if $x = \mathfrak{r}$. Thus, set $x = h^T\zeta_{jn}$ and $\mathfrak{r} = h^T\zeta_{jn}^{old}$. The new bound becomes:

$$\begin{aligned}\mathscr{L}_\zeta \geq \sum_{j=1}^{M}\sum_{n=1}^{N}\sum_{t=1}^{T}&\Bigg(\zeta_{jnt}\Big(\sum_{k=1}^{K}\rho_{jtk}\sum_{i=1}^{V}(\Psi(\lambda_{ki}) - \Psi(\sum_{p}\lambda_{kp}))[w_{jn}=i]\Big) \\ &+ \zeta_{jnt}\Big((\Psi(a_{1jt}) - \Psi(a_{1jt}+a_{2jt})) \\ &+ \sum_{\mathrm{t}=1}^{t-1}(\Psi(a_{2j\mathrm{t}}) - \Psi(a_{1j\mathrm{t}}+a_{2j\mathrm{t}}))\Big) - \zeta_{jnt}\log\zeta_{jnt}\Bigg) \\ &+ \sum_{j=1}^{M}\Bigg(\mu_{y_j}^{T}(\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T}\rho_{jt}\zeta_{jnt}) - (h^T\zeta_{jn}^{old})^{-1}h^T\zeta_{jn} \\ &- \log(h^T\zeta_{jn}^{old}) + 1\Bigg) = \mathscr{L}_\zeta'.\end{aligned} \tag{21}$$

We compute the derivative for the new bound:

$$\begin{aligned}\frac{\partial \mathscr{L}'}{\partial \zeta_{jnt}} &= \sum_{k=1}^{K}\rho_{jtk}\sum_{i=1}^{V}(\Psi(\lambda_{ki}) - \Psi(\sum_{p}\lambda_{kp}))[w_{jn}=i] \\ &+ (\Psi(a_{1jt}) - \Psi(a_{1jt}+a_{2jt})) + \sum_{\mathrm{t}=1}^{t-1}(\Psi(a_{2j\mathrm{t}}) - \Psi(a_{1j\mathrm{t}}+a_{2j\mathrm{t}})) \\ &- 1 - \log\zeta_{jnt} + \frac{1}{N}\mu_{y_j}^{T}\rho_{jt} - (h^T\zeta_{jn}^{old})^{-1}h_t.\end{aligned} \tag{22}$$

Finally, we set the derivative to zero to get the fixed point update: [4]

$$\begin{aligned}\zeta_{jnt} \propto \exp\Big(&\sum_{k=1}^{K}\rho_{jtk}\sum_{i=1}^{V}(\Psi(\lambda_{ki}) - \Psi(\sum_{p}\lambda_{kp}))[w_{jn}=i] \\ &+ (\Psi(a_{1jt}) - \Psi(a_{1jt}+a_{2jt})) + \sum_{\mathrm{t}=1}^{t-1}(\Psi(a_{2j\mathrm{t}}) - \Psi(a_{1j\mathrm{t}}+a_{2j\mathrm{t}})) \\ &- 1 + \frac{1}{N}\mu_{y_j}^{T}\rho_{jt} - (h^T\zeta_{jn}^{old})^{-1}h_t\Big) \\ &\propto \exp\Big(\sum_{k=1}^{K}\rho_{jtk}\mathop{\mathbb{E}}_{q}[\log p(w_{jn}|\phi_k)] + \mathop{\mathbb{E}}_{q}[\log\pi_{jt}] \\ &+ \frac{1}{N}\mu_{y_j}^{T}\rho_{jt} - (h^T\zeta_{jn}^{old})^{-1}h_t\Big).\end{aligned} \tag{23}$$

A.4. Update of μ

Let $\mathfrak{F} = \sum_{l=1}^{C} \prod_{n=1}^{N} \Big(\sum_{e=1}^{K} \sum_{i=1}^{T} \rho_{jie} \zeta_{jni} \exp(\frac{1}{N}\mu_{le}) \Big)$.

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial \mu_{yk}} &= \sum_{j=1}^{M}\Bigg([y_j = y]\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T}\rho_{jtk}\zeta_{jnt} \\ &- \mathfrak{F}^{-1}\prod_{m=1}^{N}\Big(\sum_{e=1}^{K}\sum_{i=1}^{T}\rho_{jie}\zeta_{jmi}\exp(\frac{1}{N}\mu_{ye})\Big) \\ &\cdot\sum_{n=1}^{N}\Big(\frac{(\sum_{i=1}^{T}\rho_{jik}\zeta_{jni})\cdot\frac{1}{N}\cdot\exp(\frac{1}{N}\mu_{yk})}{\sum_{e=1}^{K}\sum_{i=1}^{T}\rho_{jie}\zeta_{jni}\exp(\frac{1}{N}\mu_{ye})}\Big)\Bigg).\end{aligned} \tag{24}$$

[4] To incorporate the constraint that $\sum_{t=1}^{T}\zeta_{jnt} = 1$, $\propto$ is used here instead of $=$, since the normalizing factor is dropped in the above result.

Go beyond conjugate exponential family

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

$$\theta \sim N(\mu, \sigma^2)$$
$$\epsilon \sim N(0, 1), \theta = \mu + \sigma\epsilon$$

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

$$\theta \sim N(\mu, \sigma^2)$$
$$\epsilon \sim N(0, 1), \theta = \mu + \sigma\epsilon$$

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

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

$$\theta \sim N(\mu, \sigma^2)$$
$$\epsilon \sim N(0,1), \theta = \mu + \sigma\epsilon$$

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

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

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

$$\theta \sim N(\mu, \sigma^2)$$
$$\epsilon \sim N(0, 1), \theta = \mu + \sigma\epsilon$$

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

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

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

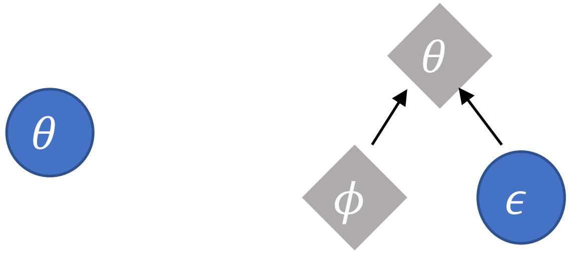
$$\theta \sim N(\mu, \sigma^2)$$
$$\epsilon \sim N(0, 1), \theta = \mu + \sigma\epsilon$$

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

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

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

Reparameterization Trick



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

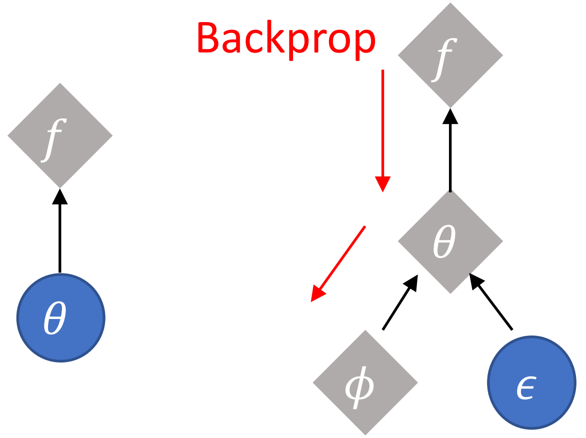
$$\theta \sim N(\mu, \sigma^2)$$
$$\epsilon \sim N(0,1), \theta = \mu + \sigma\epsilon$$

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

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

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

Reparameterization Trick



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

$$\theta \sim N(\mu, \sigma^2)$$
$$\epsilon \sim N(0,1), \theta = \mu + \sigma\epsilon$$

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

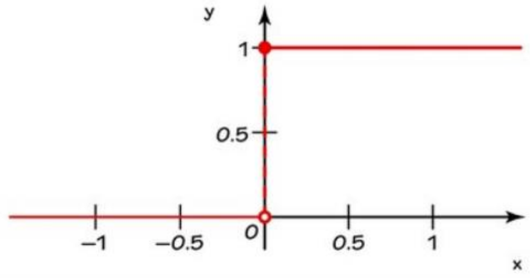
Gradient $\nabla_\phi L = \nabla_\phi E_{r(\epsilon)}\left[\log \frac{\mathrm{p}(g(\epsilon,\phi), D)}{q_\phi(g(\epsilon,\phi))}\right] = E_{r(\epsilon)}\left[\nabla_\phi \log \frac{\mathrm{p}(g(\epsilon,\phi), D)}{q_\phi(g(\epsilon,\phi))}\right]$

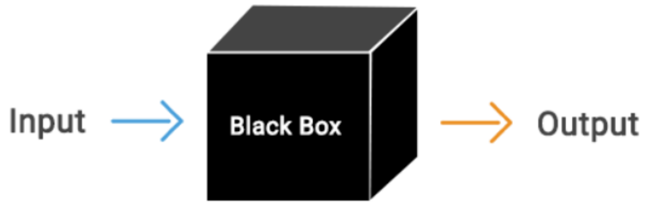
$$\nabla_\phi \hat{L} = \frac{1}{K}\sum_{k=1}^{K} \nabla_\phi \log \frac{\mathrm{p}(g(\epsilon_k,\phi), D)}{q_\phi(g(\epsilon_k,\phi))}, \epsilon_k \sim r(\epsilon)$$

Variance Reduction Techniques in MCVI

- When non-differentiable, falls back to REINFORCE gradient

High variance!

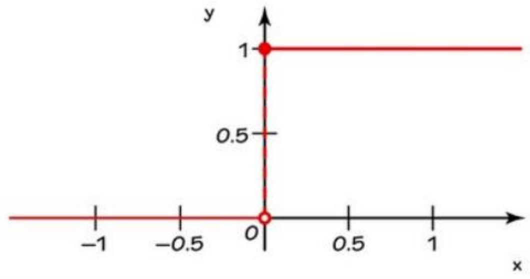


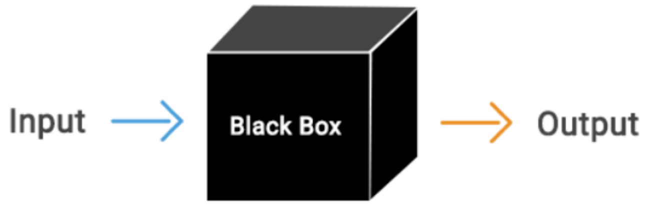


Variance Reduction Techniques in MCVI

- When non-differentiable, falls back to REINFORCE gradient

High variance!



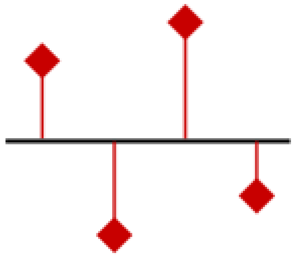


- Solutions to high variance REINFORCE gradients:
  - Low variance unbiased estimators with control variates
  - Biased estimators to enable reparam. trick (potentially low variance)

Variance Reduction Techniques in MCVI

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

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

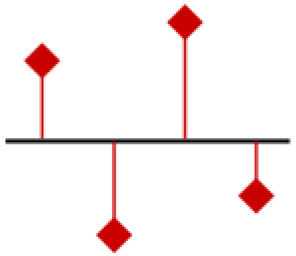


$F(\theta)$

Variance Reduction Techniques in MCVI

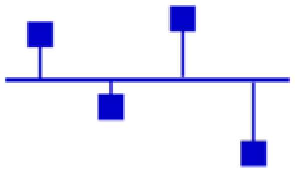
- Control variate method:
    - Assume we want to estimate with MC simulation

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



$F(\theta)$

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

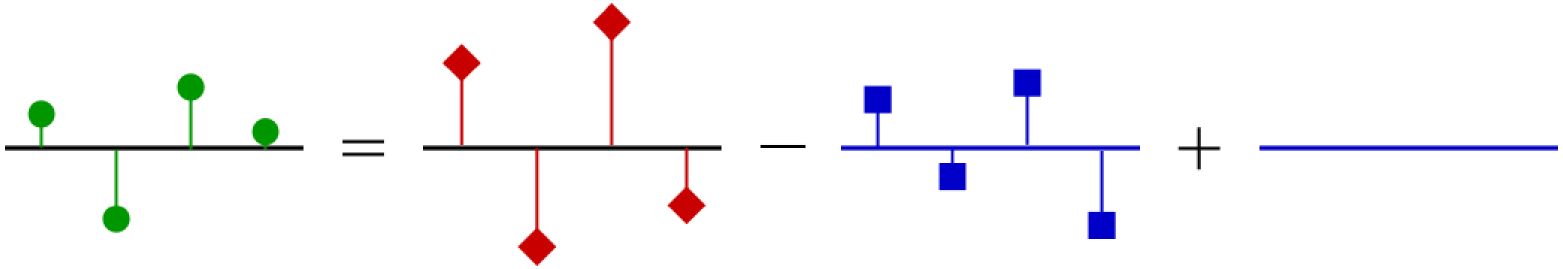


$G(\theta)$

Variance Reduction Techniques in MCVI

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

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



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

$$V_{q(\theta)}[\hat{F}(\theta)] = V_{q(\theta)}[F(\theta)] + \underbrace{V_{q(\theta)}[G(\theta)] - 2\, Cov_{q(\theta)}[F(\theta), G(\theta)]}_{<\, 0 \text{ if } F \text{ and } G \text{ are strongly and positively correlated}}$$

Variance Reduction Techniques in MCVI

- Application to REINFORCE gradient:
  - $F(\theta) = \underbrace{\log \frac{p(D,\theta)}{q_\phi(\theta)}}_{:= f(\theta)} \nabla_\phi \log q_\phi(\theta)$
  - Define $G(\theta) = g(\theta) \nabla_\phi \log q_\phi(\theta)$

Variance Reduction Techniques in MCVI

- Application to REINFORCE gradient:
  - $F(\theta) = \underbrace{\log \frac{p(D,\theta)}{q_\phi(\theta)}}_{:= f(\theta)} \nabla_\phi \log q_\phi(\theta)$
  - Define $G(\theta) = g(\theta) \nabla_\phi \log q_\phi(\theta)$
  - Variance reduced gradient: $\hat{F}(\theta) = \underbrace{\big(f(\theta) - g(\theta)\big)}_{:= \Delta(\theta)} \nabla_\phi \log \mathrm{q}_\phi(\theta) + E_{\mathrm{q}_\phi(\theta)}[G(\theta)]$

Variance Reduction Techniques in MCVI

- Application to REINFORCE gradient:
  - $F(\theta) = \underbrace{\log \frac{p(D,\theta)}{q_\phi(\theta)}}_{:= f(\theta)} \nabla_\phi \log q_\phi(\theta)$
  - Define $G(\theta) = g(\theta)\nabla_\phi \log q_\phi(\theta)$
  - Variance reduced gradient: $\hat{F}(\theta) = \underbrace{\big(f(\theta) - g(\theta)\big)}_{:= \Delta(\theta)} \nabla_\phi \log \mathrm{q}_\phi(\theta) + E_{\mathrm{q}_\phi(\theta)}[G(\theta)]$
  - "Baseline" approach:

$$g(\theta) = b$$

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

(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

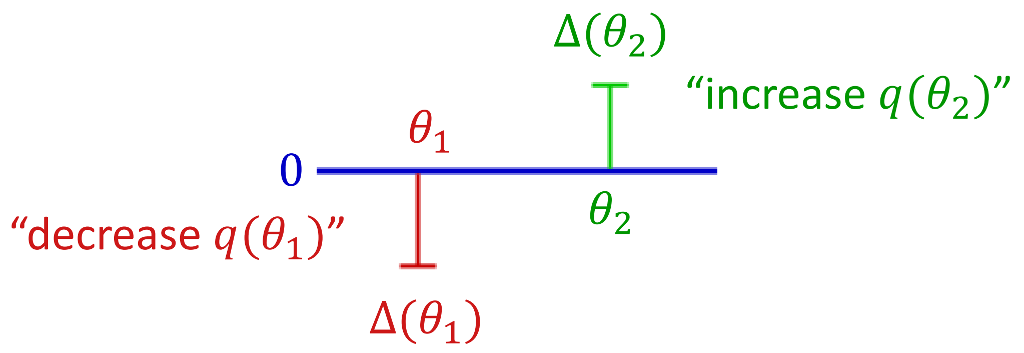
Variance Reduction Techniques in MCVI

- Application to REINFORCE gradient:
  - $F(\theta) = \underbrace{\log \frac{p(D,\theta)}{q_\phi(\theta)}}_{:= f(\theta)} \nabla_\phi \log q_\phi(\theta)$
  - Define $G(\theta) = g(\theta) \nabla_\phi \log q_\phi(\theta)$
  - Variance reduced gradient: $\hat{F}(\theta) = \underbrace{\big(f(\theta) - g(\theta)\big)}_{:= \Delta(\theta)} \nabla_\phi \log q_\phi(\theta) + E_{q_\phi(\theta)}[G(\theta)]$
  - "Baseline" approach:

$$g(\theta) = b$$

$$\Rightarrow E_{q(\theta)}[G(\theta)] = \underbrace{b E_{q(\theta)}\big[\nabla_\phi \log q(\theta)\big] = 0}_{\text{log-derivative trick}}$$

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



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:
  - $F(\theta) = \underbrace{\log \frac{p(D,\theta)}{q_\phi(\theta)}}_{:= f(\theta)} \nabla_\phi \log q_\phi(\theta)$
  - Define $G(\theta) = g(\theta) \nabla_\phi \log q_\phi(\theta)$
  - Variance reduced gradient: $\hat{F}(\theta) = \underbrace{\left(f(\theta) - g(\theta)\right)}_{:= \Delta(\theta)} \nabla_\phi \log \mathrm{q}_\phi(\theta) + E_{\mathrm{q}_\phi(\theta)}[G(\theta)]$
  - "Taylor expansion" approach (e.g. 1st order):

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

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

- Application to REINFORCE gradient:
  - $F(\theta) = \underbrace{\log \frac{p(D,\theta)}{q_\phi(\theta)}}_{:= f(\theta)} \nabla_\phi \log q_\phi(\theta)$
  - Define $G(\theta) = g(\theta) \nabla_\phi \log q_\phi(\theta)$
  - Variance reduced gradient: $\hat{F}(\theta) = \underbrace{\big(f(\theta) - g(\theta)\big)}_{:= \Delta(\theta)} \nabla_\phi \log \mathrm{q}_\phi(\theta) + E_{\mathrm{q}_\phi(\theta)}[G(\theta)]$
  - "Taylor expansion" approach (e.g. 1st order):

$$g(\theta) = f(\theta_0) + \nabla_{\theta_0} f(\theta_0)(\theta - \theta_0) \qquad \Rightarrow E_{q(\theta)}[G(\theta)] = \overbrace{\big(f(\theta_0) - \nabla_{\theta_0} f(\theta_0)\theta_0\big) E_{q(\theta)}\big[\nabla_\phi \log q(\theta)\big]}^{= 0 \text{ (log-derivative trick)}}$$

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

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

- Application to REINFORCE gradient:
  - $F(\theta) = \underbrace{\log\frac{p(D,\theta)}{q_\phi(\theta)}}_{:= f(\theta)} \nabla_\phi \log q_\phi(\theta)$
  - Define $G(\theta) = g(\theta)\nabla_\phi \log q_\phi(\theta)$
  - Variance reduced gradient: $\hat{F}(\theta) = \underbrace{\big(f(\theta) - g(\theta)\big)}_{:= \Delta(\theta)}\nabla_\phi \log \mathrm{q}_\phi(\theta) + E_{\mathrm{q}_\phi(\theta)}[G(\theta)]$
  - "Taylor expansion" approach (e.g. 1st order):

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

$$\Rightarrow E_{q(\theta)}[G(\theta)] = \overbrace{\big(f(\theta_0) - \nabla_{\theta_0} f(\theta_0)\theta_0\big)E_{q(\theta)}[\nabla_\phi \log q(\theta)]}^{=\,0\ \text{(log-derivative trick)}}$$

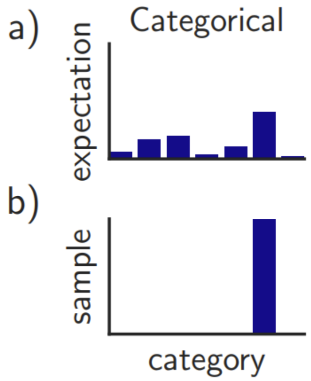
$$+\underbrace{\nabla_{\theta_0} f(\theta_0) E_{q(\theta)}[\theta \nabla_\phi \log q(\theta)]}_{= \nabla_{\theta_0} f(\theta_0)\nabla_\phi E_{q(\theta)}[\theta]\ \text{(log-derivative trick)}}$$

$$\Rightarrow \hat{F}(\theta) = \Delta(\theta)\nabla_\phi \log q_\phi(\theta) + \underbrace{\nabla_{\theta_0} f(\theta_0)\nabla_\phi E_{q(\theta)}[\theta]}_{:= \nabla_\phi f(\theta_0)\ \text{if}\ \theta_0 = E_{q(\theta)}[\theta]}$$

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

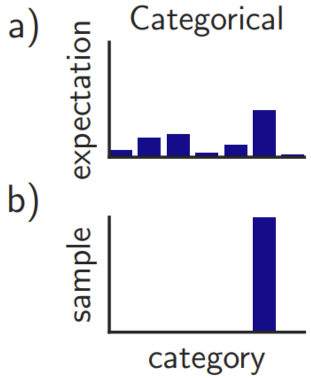


Categorical distribution:

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

Variance Reduction Techniques in MCVI

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

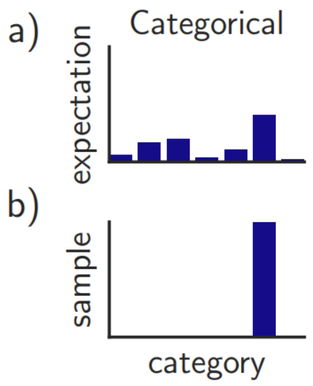


Gumbel trick to sample y:

$$y = \arg max\ [g_k + \log \pi_k],$$
$$g_k \sim Gumbel(0, 1)$$

Variance Reduction Techniques in MCVI

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



Gumbel trick to sample y:

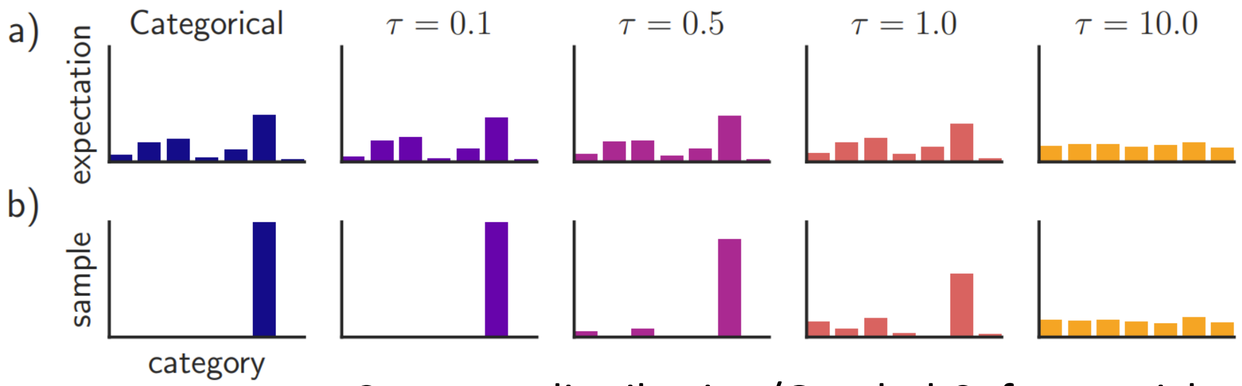
$$y = \boxed{\arg max}\,[g_k + \log \pi_k],$$
$$g_k \sim Gumbel(0,1)$$

replace with softmax

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

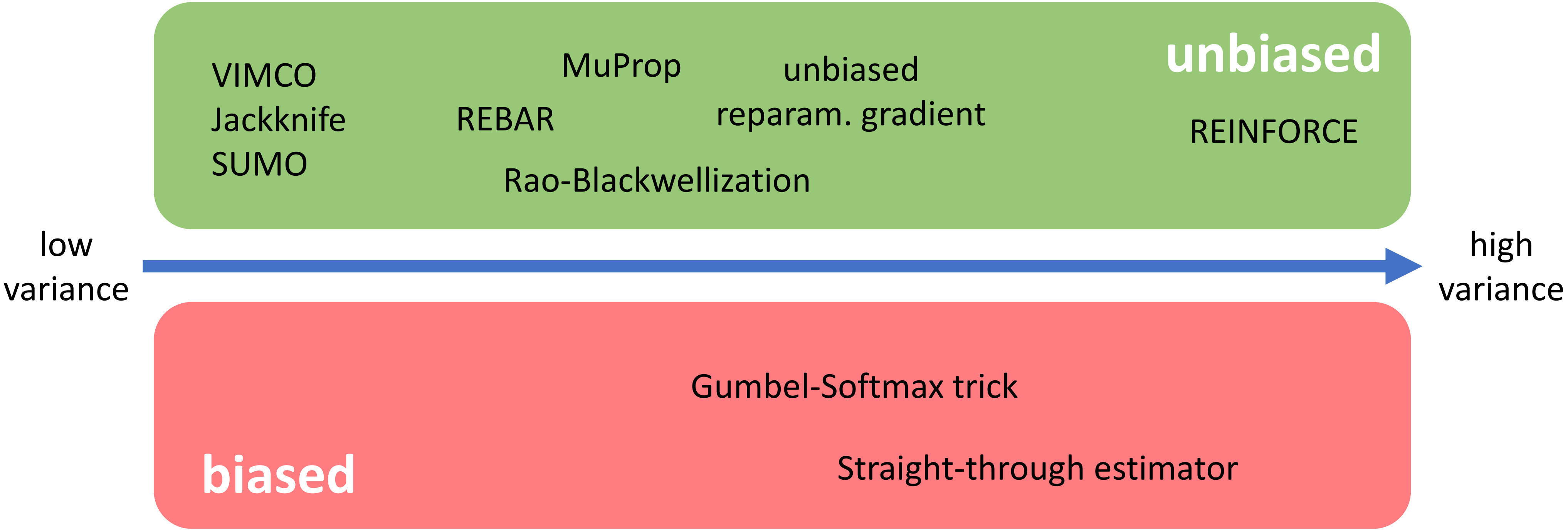


Concrete distribution/Gumbel-Softmax trick:
sample the “soft vector” (instead of one-hot encoding of y)

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

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



unbiased

VIMCO
Jackknife
SUMO

MuProp

REBAR

Rao-Blackwellization

unbiased
reparam. gradient

REINFORCE

low
variance

high
variance

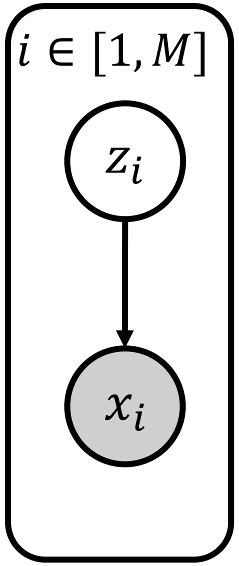
Gumbel-Softmax trick

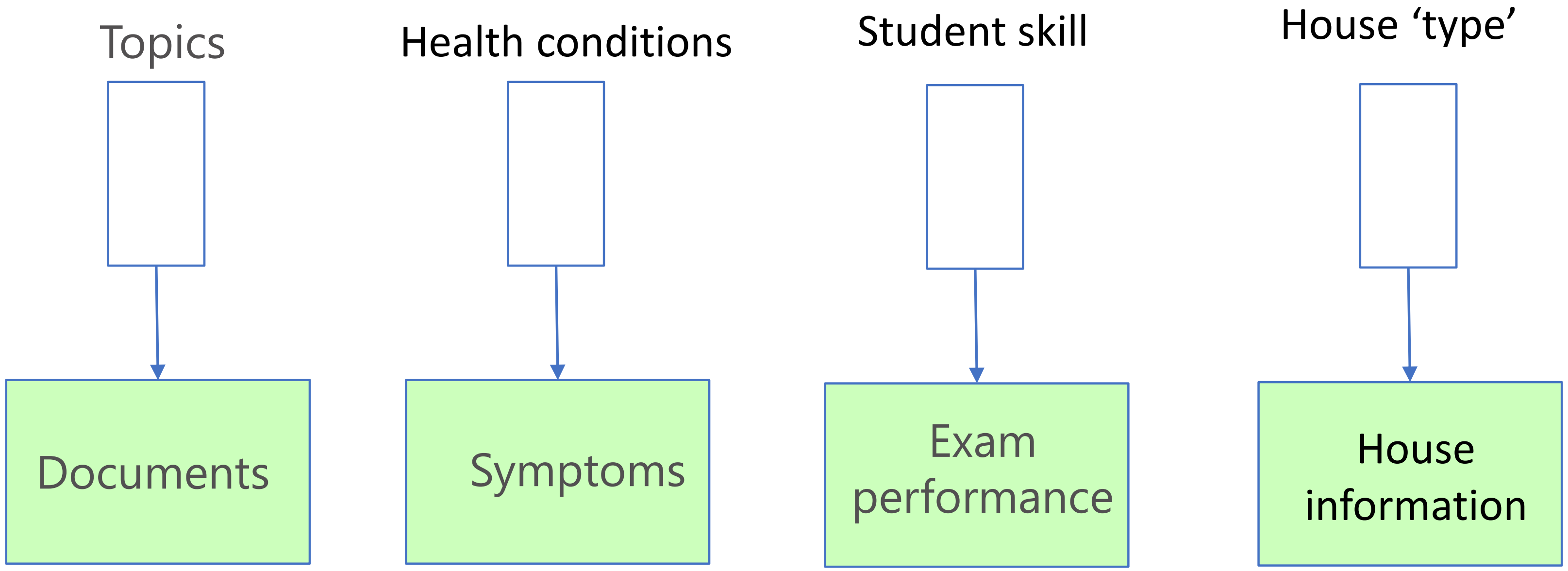
Straight-through estimator

biased

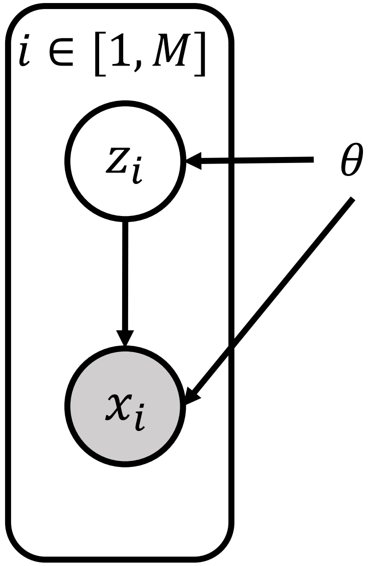
For an incomplete list of variance reduced gradient estimators, see http://yingzhenli.net/home/en/?page_id=1262

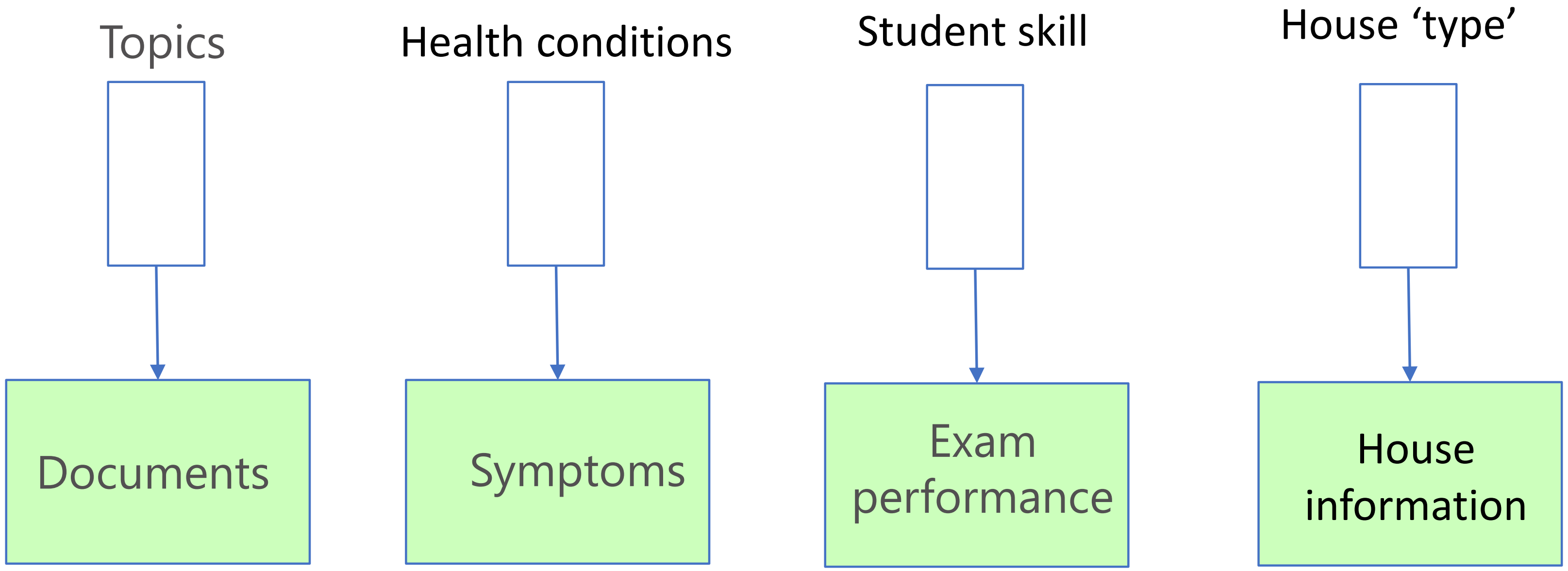
Latent Variable Model





Deep Latent Variable Model





Topics

Documents

Health conditions

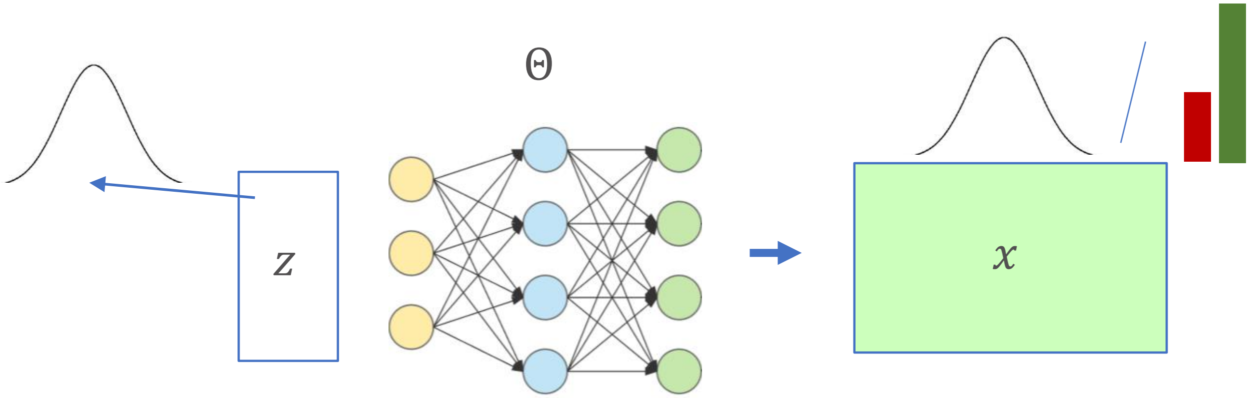
Symptoms

Student skill

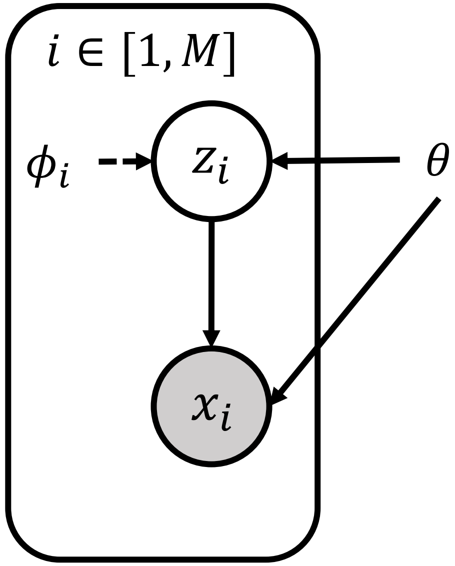
Exam performance

House ‘type’

House information



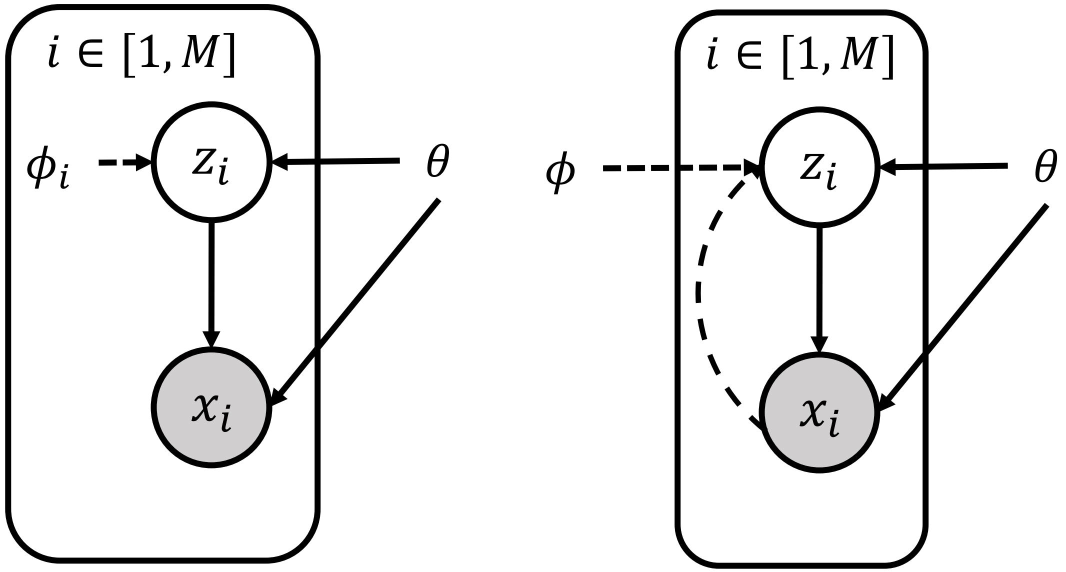
Amortized Inference



ϕ parameter for variational distribution

θ decoder parameter

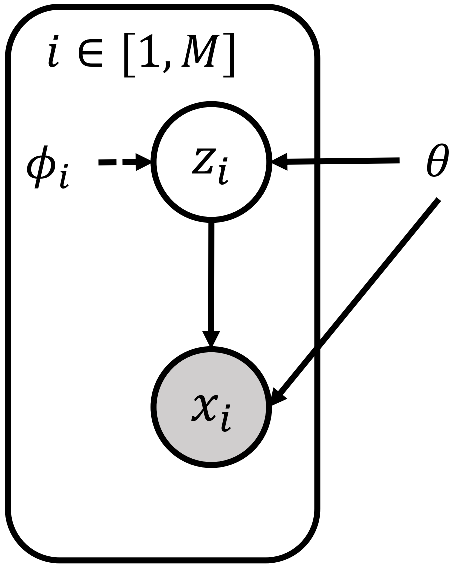
Amortized Inference

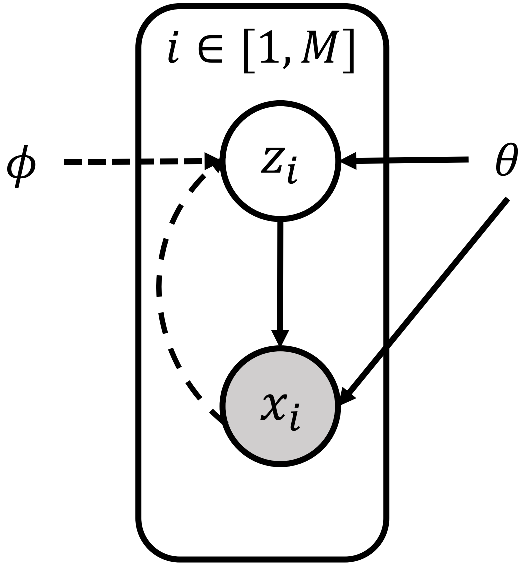


ϕ parameter for variational distribution

θ decoder parameter

Amortized Inference



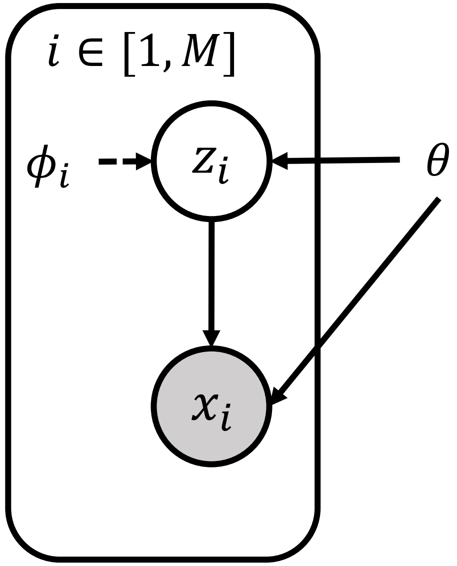


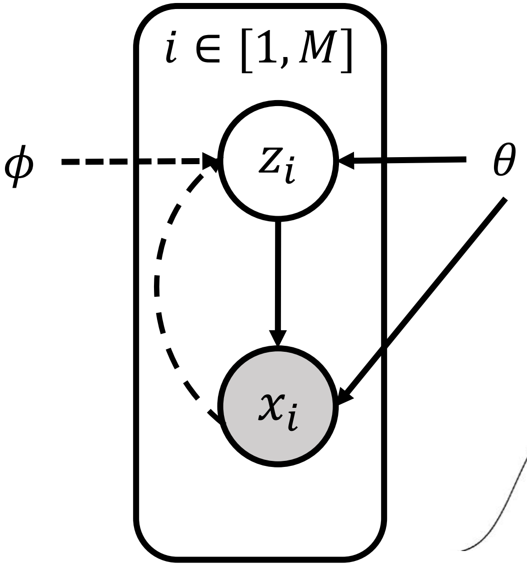
$$L = \log p(\boldsymbol{x}) - KL[q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x})]$$

↓

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

Variational Auto-Encoders (VAE)

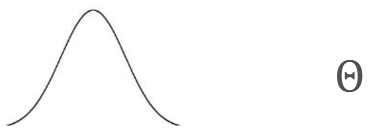


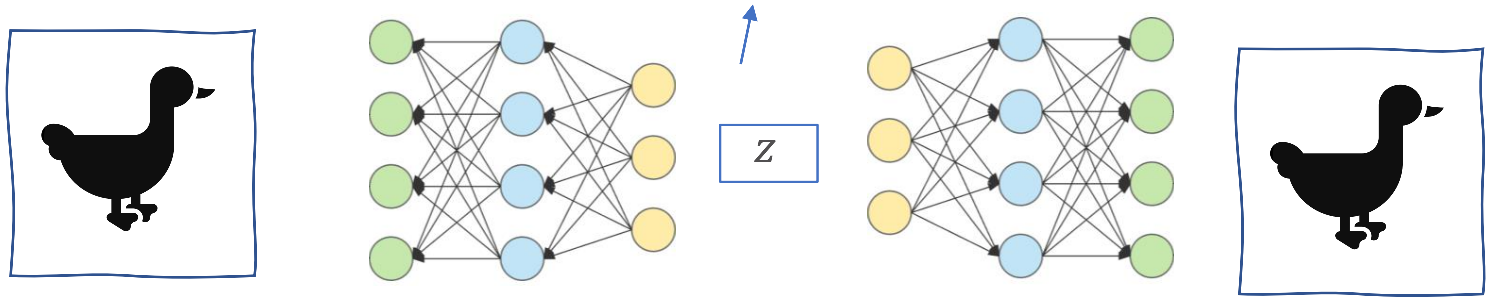


$$L = \log p(\boldsymbol{x}) - KL[q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x})]$$

$$\downarrow$$

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

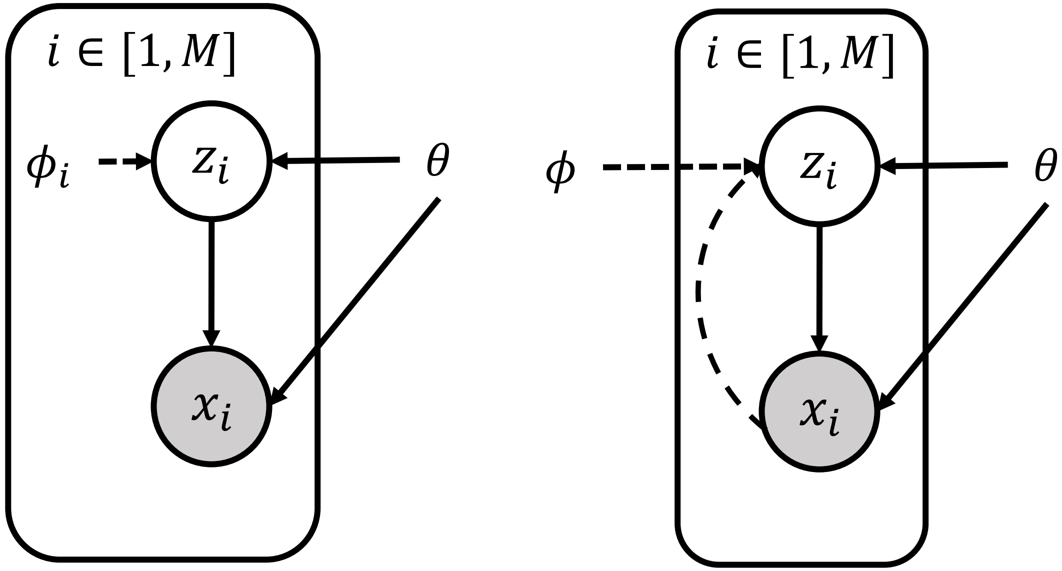




Encoder/
inference network

Decoder/
generator

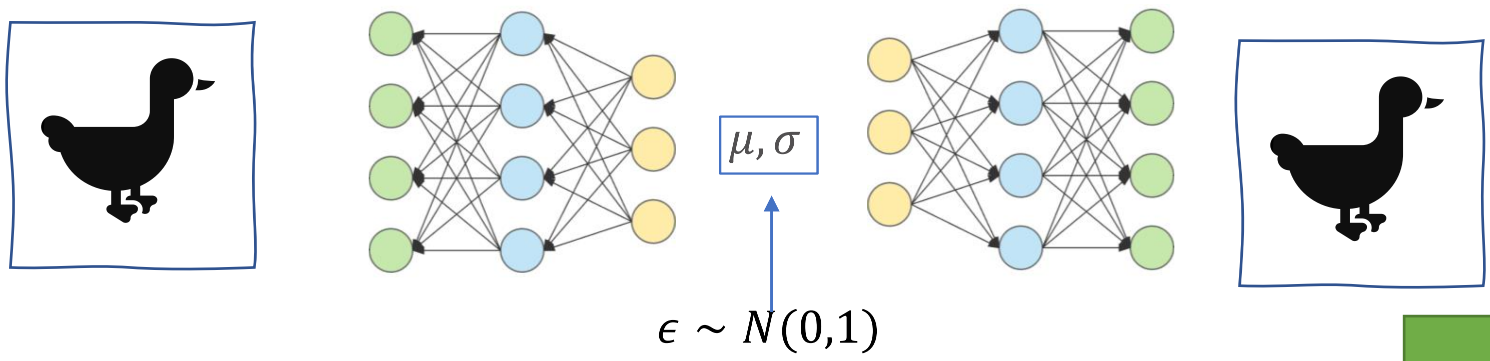
Variational Auto-Encoders (VAE)



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

↓

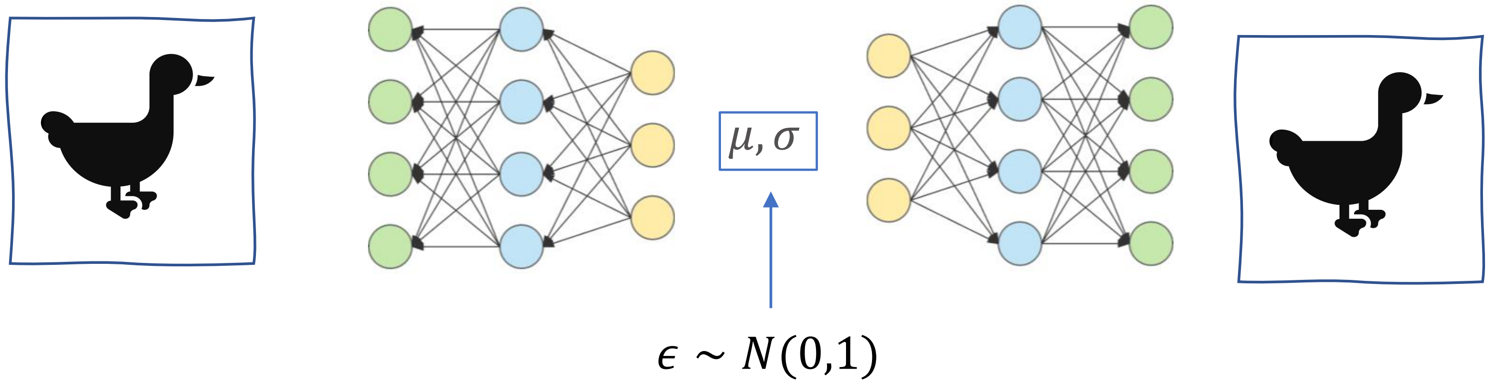
$$L_{amortized} = \log p(\boldsymbol{x}) - KL[q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}|\boldsymbol{x})]$$



$$z \sim N(\mu, \sigma^2)$$
$$\epsilon \sim N(0, 1), \theta = \mu + \sigma\epsilon$$

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)



μ, σ

$\epsilon \sim N(0,1)$

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

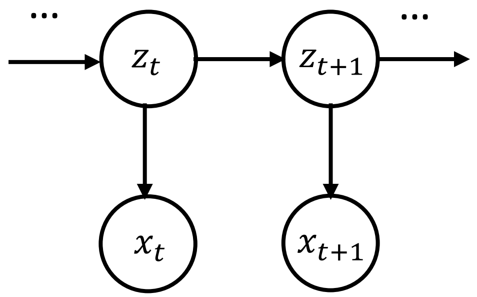
$$= E_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}[\log \mathrm{p}_\theta(\boldsymbol{x}|\boldsymbol{z})] - KL[q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}|\boldsymbol{x})]$$

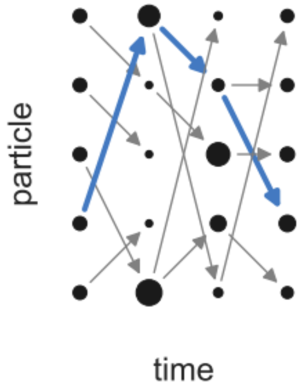
Kingma and Welling. Auto-encoding variational bayes. ICLR 2014.
Rezende et al. Stochastic backpropagation and approximate inference in deep generative models. ICML 2014.

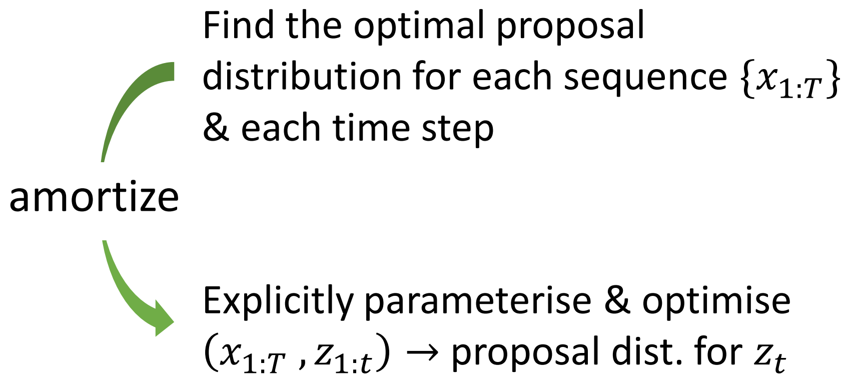
How to apply amortization to other inference methods?

Amortized Inference: Further Examples

- Amortized SMC







amortize

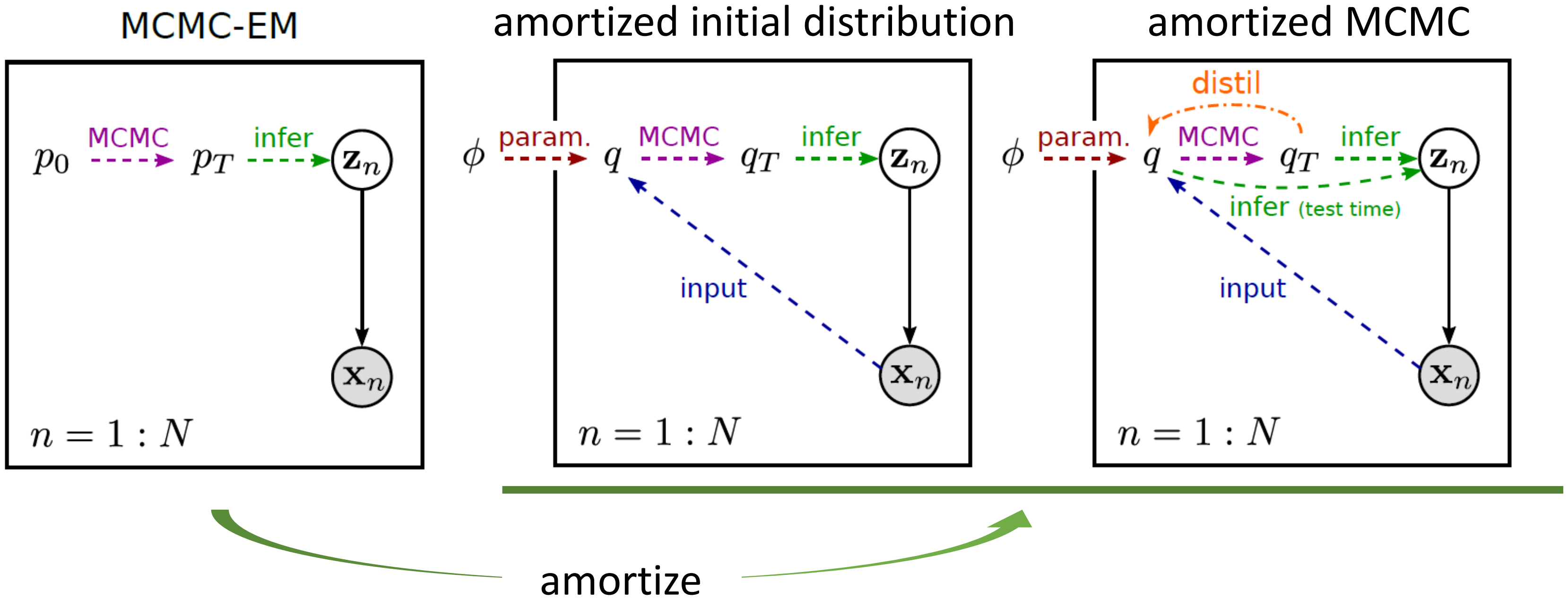
Find the optimal proposal distribution for each sequence $\{x_{1:T}\}$ & each time step

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

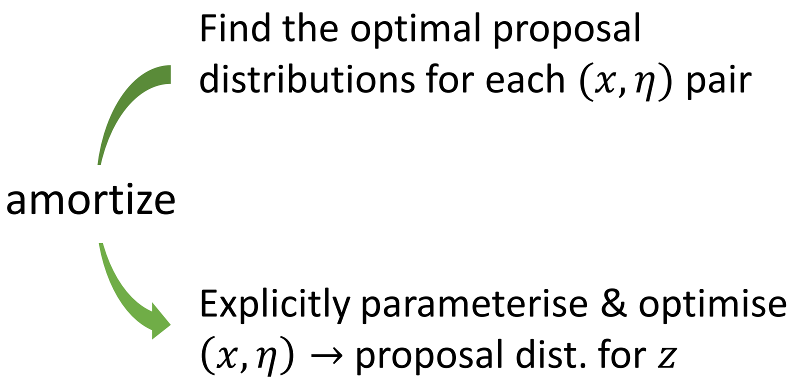
Li et al. Approximate Inference with Amortised MCMC. ICML 2017 Workshop on Implicit Models

Amortized Inference: Further Examples

- Amortized Monte Carlo integration

Goal: estimate with importance sampling

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



Find the optimal proposal distributions for each (x, η) pair

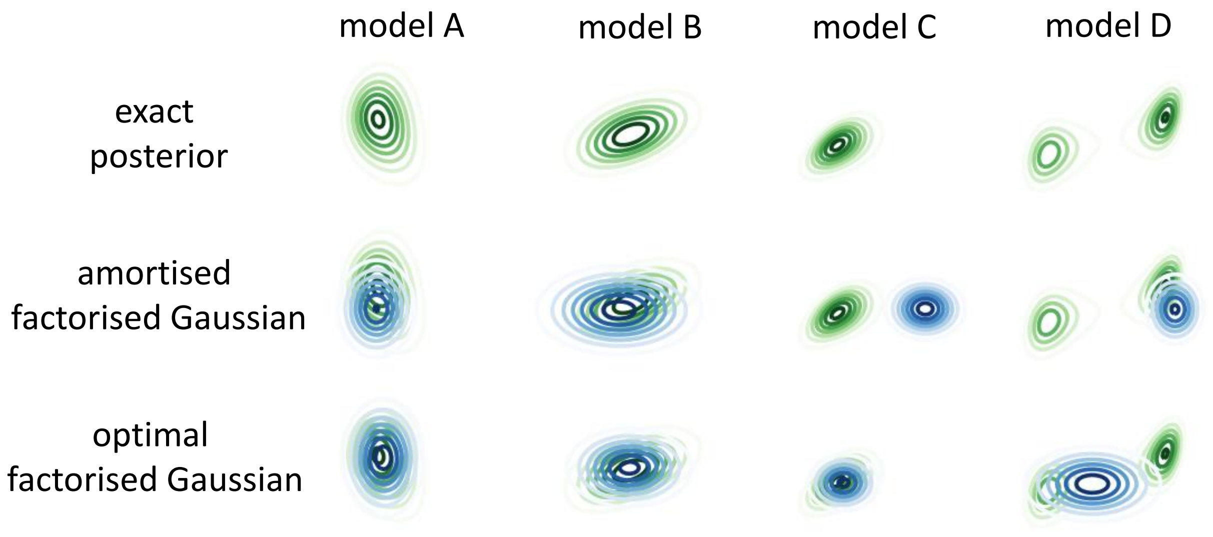
amortize

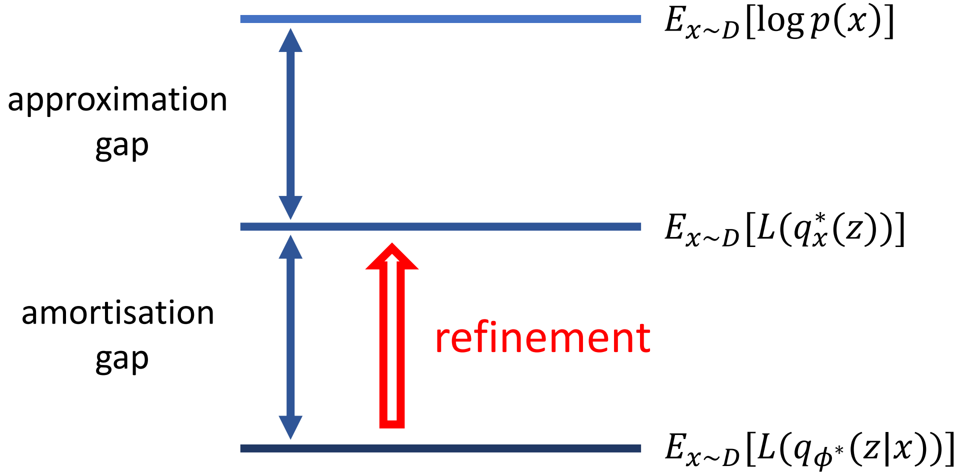
Explicitly parameterise & optimise $(x, \eta) \rightarrow$ proposal dist. for z

Goliński et al. Amortized Monte Carlo Integration. ICML 2019

Amortized Inference: Limitations

- Amortised approximate posteriors in practice are sub-optimal





$E_{x\sim D}[\log p(x)]$

approximation gap

$E_{x\sim D}[L(q_x^*(z))]$

amortisation gap

refinement

$E_{x\sim D}[L(q_{\phi^*}(z|x))]$

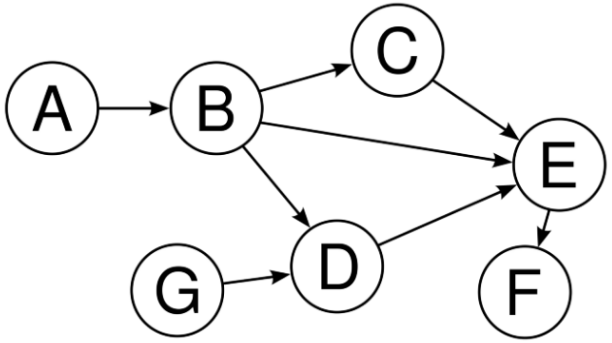
- The “refinement” idea:
  - Initialise $q(z|x) = N(z; \mu, \sigma^2)$ with the amortised solution $\mu \leftarrow \mu_\phi(x), \sigma \leftarrow \sigma_\phi(x)$
  - Then run T more VI gradient steps to update μ, σ

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

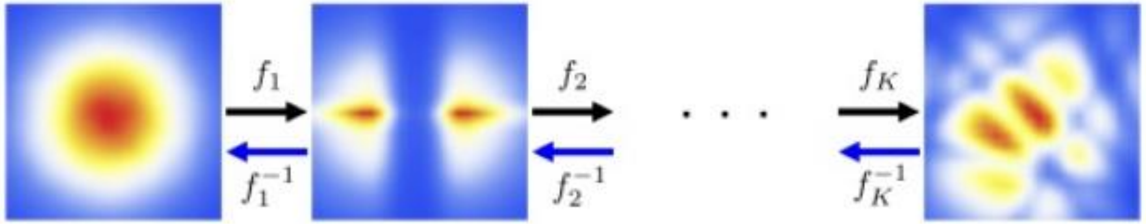
Part II: Advances

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

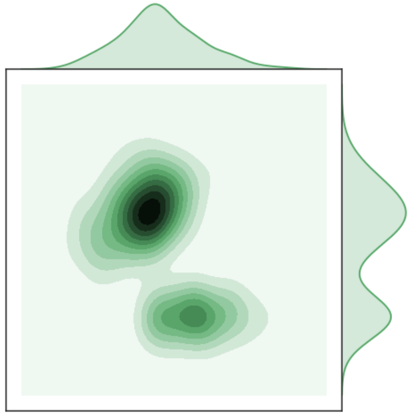
Designing q Distributions



Structured approximations



Normalizing flows

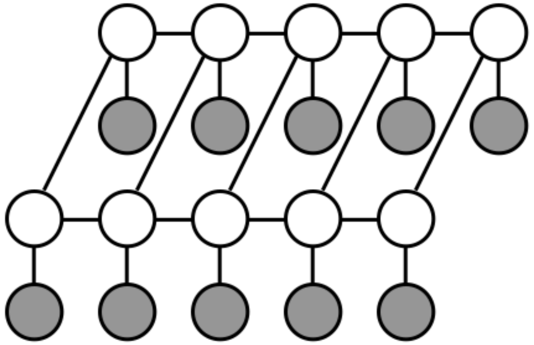


Auxiliary variables & mixture distributions

Implicit approximate posteriors

Structured Approximations

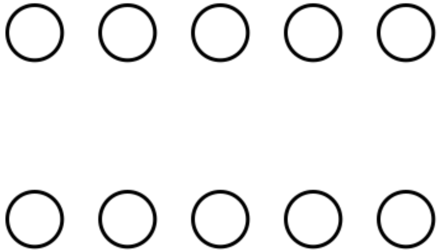
- introduce dependencies between random variables for q:



Hidden Markov Model

Exact posterior $p(z \mid x)$

$z_i \not\perp z_j \mid x$

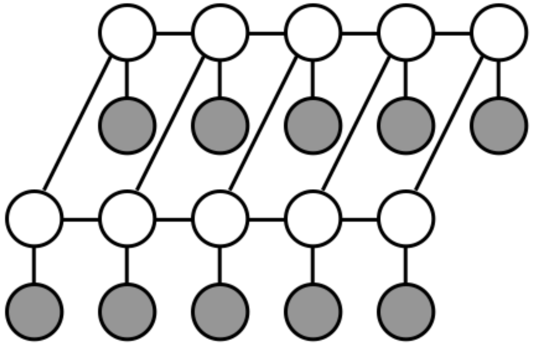


Mean-field approximation

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

Structured Approximations

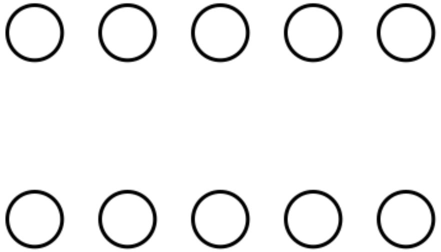
- introduce dependencies between random variables for q:



Hidden Markov Model

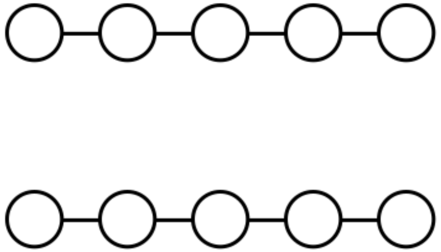
Exact posterior $p(z \mid x)$

$z_i \not\perp z_j \mid x$



Mean-field approximation

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



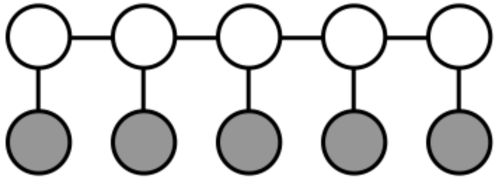
Structured approximation

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

Main design question: the grouping and conditional dependency structure

Structured Approximations

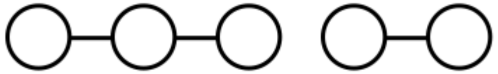
- Auto-regressive distributions (as a specific dependency structure)



Hidden Markov Model

Exact posterior $p(z \mid x)$

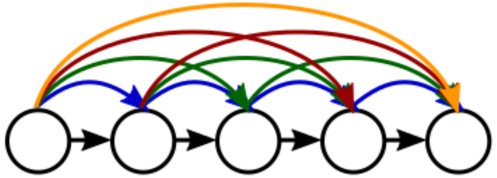
$z_i \not\perp z_j \mid x$



Structured approximation

$q(z) = \prod_s q(z_s)$

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



Auto-regressive approximation

$$q(z) = \prod_i q(z_i \mid z_{<i})$$

$$q(z_1 \mid z_{<1}) = q(z_1)$$

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) $p_X(x)$
  - $\mathrm{y} = f(x)$ is an invertible mapping

Normalizing Flows

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

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

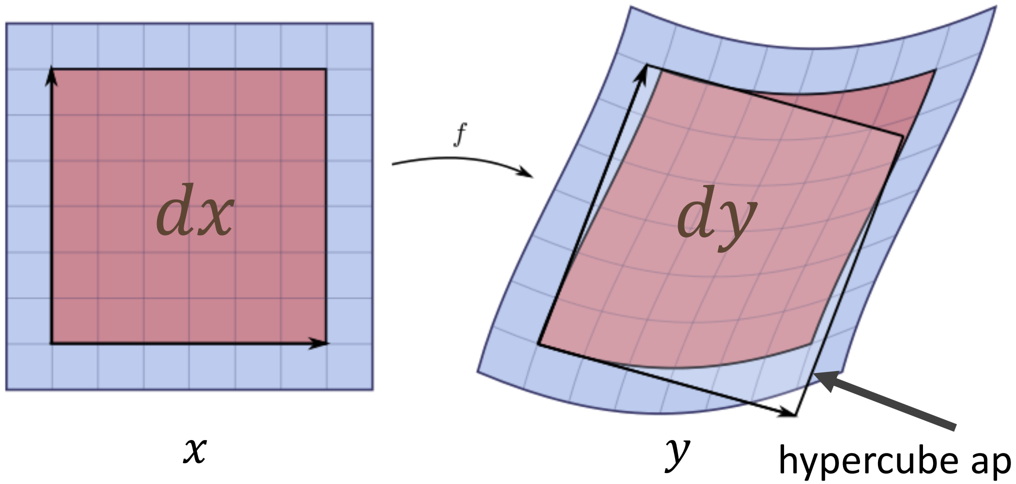
Normalizing Flows

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

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

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

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



hypercube approximation
(exact when $\mathrm{vol}(dx) \to 0$)

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 ϕ

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

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

Rezende and Mohamed. Variational Inference with Normalizing Flows. ICML 2015

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 ϕ

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

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

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

Rezende and Mohamed. Variational Inference with Normalizing Flows. ICML 2015

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 ϕ

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

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

by def. of $q(z)$

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

Rezende and Mohamed. Variational Inference with Normalizing Flows. ICML 2015

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 ϕ

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

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

$$\begin{aligned} L\big(q(z)\big) &= E_{q(z)}[\log p(x \mid z) + \log p(z) \; - \log q(z)] \\ &= E_{q(z)}\left[\log p(x,\; z) - \boxed{\log q_0(z_0 = f_\phi^{-1}(z))|\det\left(\frac{dz}{dz_0}\right)|^{-1}}\right] \\ &= \underline{E_{q_0(z_0)}\left[\log p(x, f_\phi(z_0)) \; - \log q_0(z_0) + \log|\det\left(\frac{df_\phi}{dz_0}\right)|\right]} \end{aligned}$$

by def. of $q(z)$

reparam. trick:

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

Rezende and Mohamed. Variational Inference with Normalizing Flows. ICML 2015

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 ϕ

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

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

$$\begin{aligned} L\big(q(z)\big) &= E_{q(z)}[\log p(x \mid z) + \log p(z) \; - \log q(z)] \\ &= E_{q(z)}\left[\log p(x,\; z) - \boxed{\log q_0(z_0 = f_\phi^{-1}(z))|\det\left(\frac{dz}{dz_0}\right)|^{-1}}\right] \\ &= \underline{E_{q_0(z_0)}\left[\log p(x, f_\phi(z_0)) \; - \log q_0(z_0) + \log|\det\left(\frac{df_\phi}{dz_0}\right)|\right]} \end{aligned}$$

by def. of $q(z)$

reparam. trick:

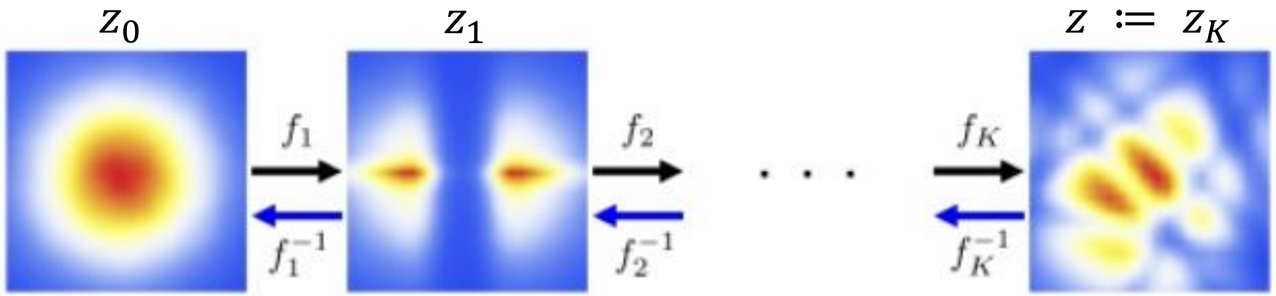
$z \sim q(z) \Leftrightarrow z_0 \sim q_0(z_0), z = f_\phi(z_0)$

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

Rezende and Mohamed. Variational Inference with Normalizing Flows. ICML 2015

Normalizing Flows

- Variational inference with Normalizing flow
  - Idea: define f_ϕ 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



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

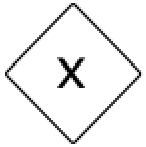
    - For each simple mapping, hopefully the Jacobian log-determinant is easy to compute

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

Rezende and Mohamed. Variational Inference with Normalizing Flows. ICML 2015

Normalizing Flows

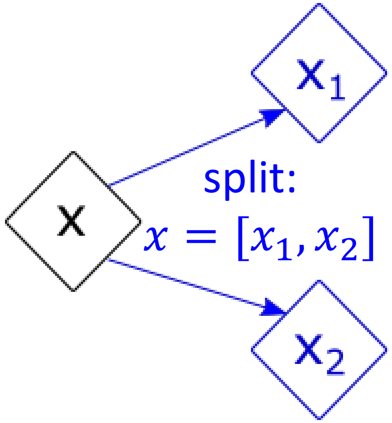
- Goal: construct f_k to enable fast compute of $\log|\det(\frac{dz_k}{dz_{k-1}})|$
    - Example (RealNVP): $y \coloneqq f_{\phi_k}(x)$ computed as follows



Dinh et al. Density Estimation using Real NVP. ICLR 2017

Normalizing Flows

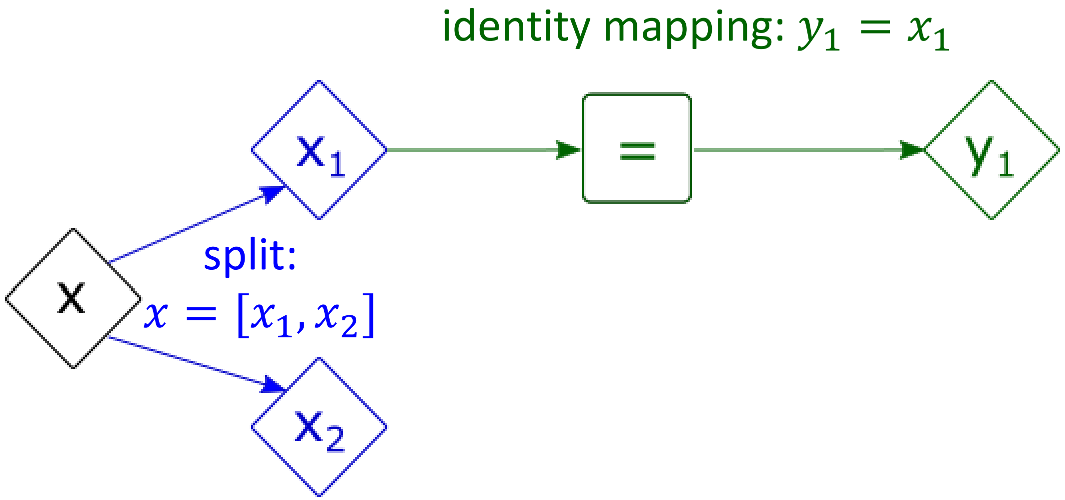
- Goal: construct f_k to enable fast compute of $\log|\det(\frac{dz_k}{dz_{k-1}})|$
  - Example (RealNVP): $y \coloneqq f_{\phi_k}(x)$ computed as follows



Dinh et al. Density Estimation using Real NVP. ICLR 2017

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

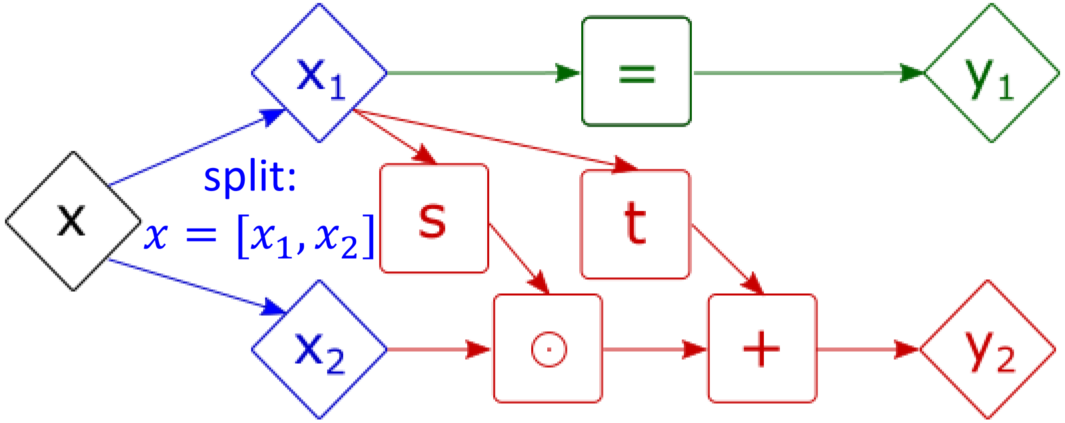


Dinh et al. Density Estimation using Real NVP. ICLR 2017

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

identity mapping: $y_1 = x_1$



split: $x = [x_1, x_2]$

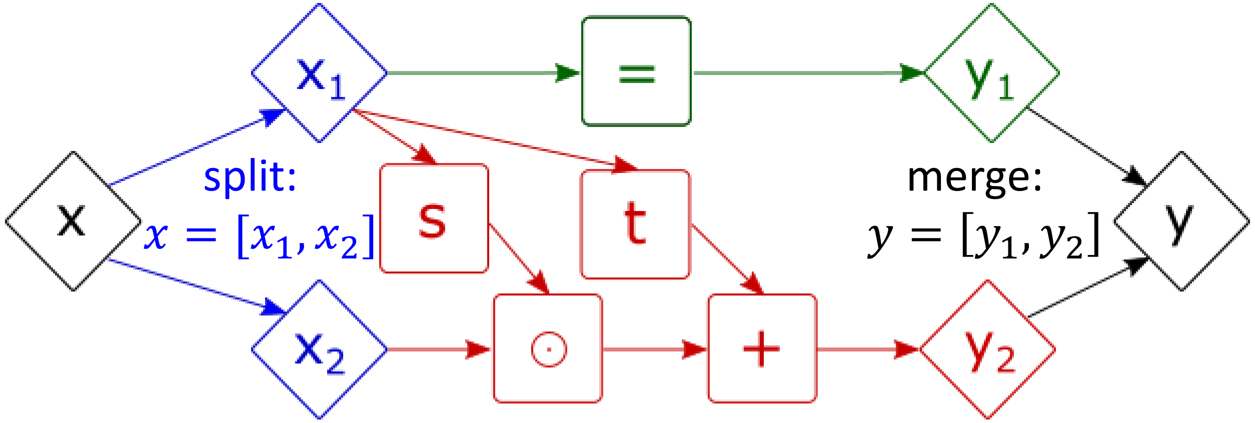
affine transform: $y_2 = x_2 \odot \exp\big(s(x_1)\big) + t(x_1)$

Dinh et al. Density Estimation using Real NVP. ICLR 2017

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

identity mapping: $y_1 = x_1$



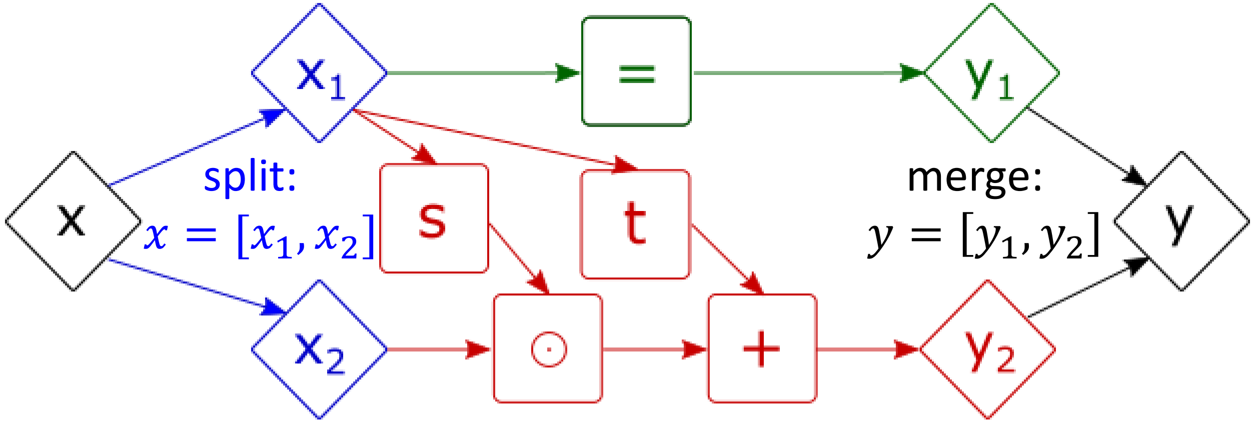
affine transform: $y_2 = x_2 \odot \exp\big(s(x_1)\big) + t(x_1)$

Dinh et al. Density Estimation using Real NVP. ICLR 2017

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

identity mapping: $y_1 = x_1$



split: $x = [x_1, x_2]$

merge: $y = [y_1, y_2]$

affine transform: $y_2 = x_2 \odot \exp\big(s(x_1)\big) + t(x_1)$

Jacobian:

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

Log-determinant of Jacobian:

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

Dinh et al. Density Estimation using Real NVP. ICLR 2017

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

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

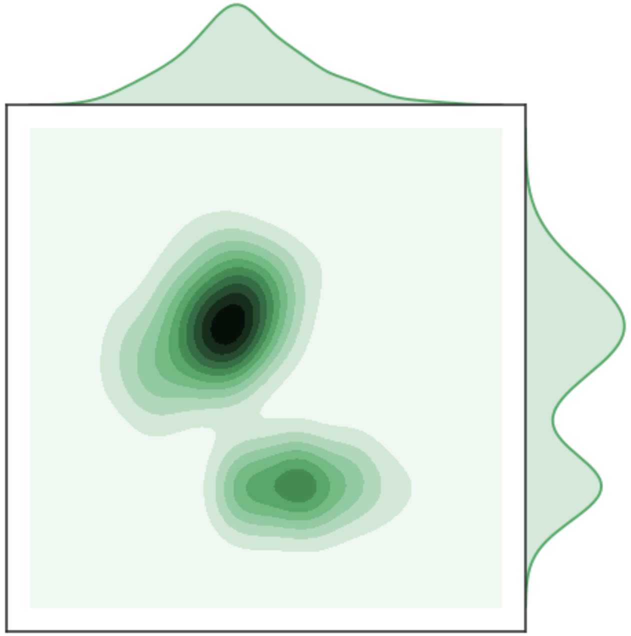
  - a is the auxiliary variable used to enrich the approximate posterior

  - Example: Mixture of Gaussians

$$a \sim q(a) = Categorical(\pi_1, \ldots, \pi_K)$$

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

Can be very flexible with many components!



Auxiliary Variables & Mixture Distributions

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

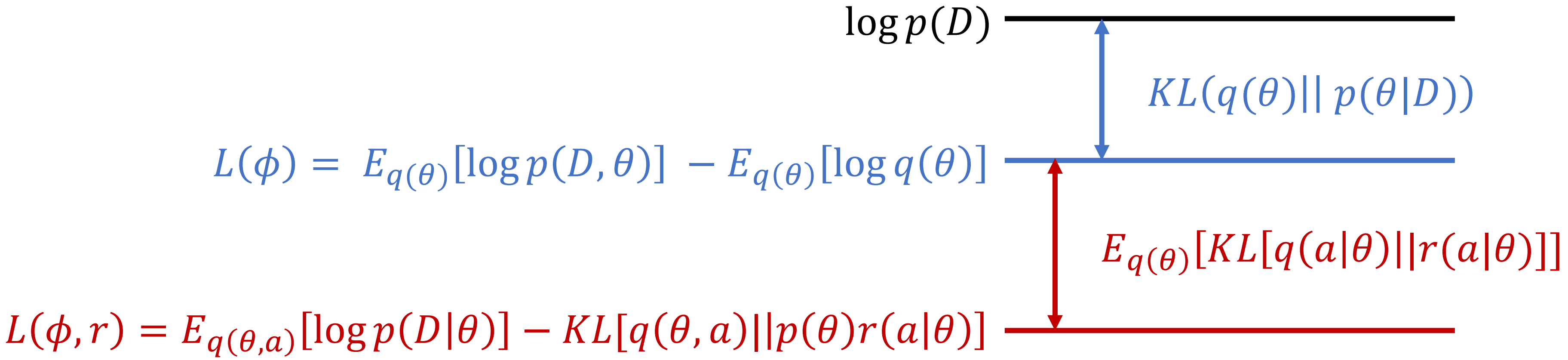
$$q(\theta) = \int q(\theta \,|a)\, q(a)\, da$$

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

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

Auxiliary Variables & Mixture Distributions

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



$\log p(D)$

$KL(q(\theta)|| \, p(\theta|D))$

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

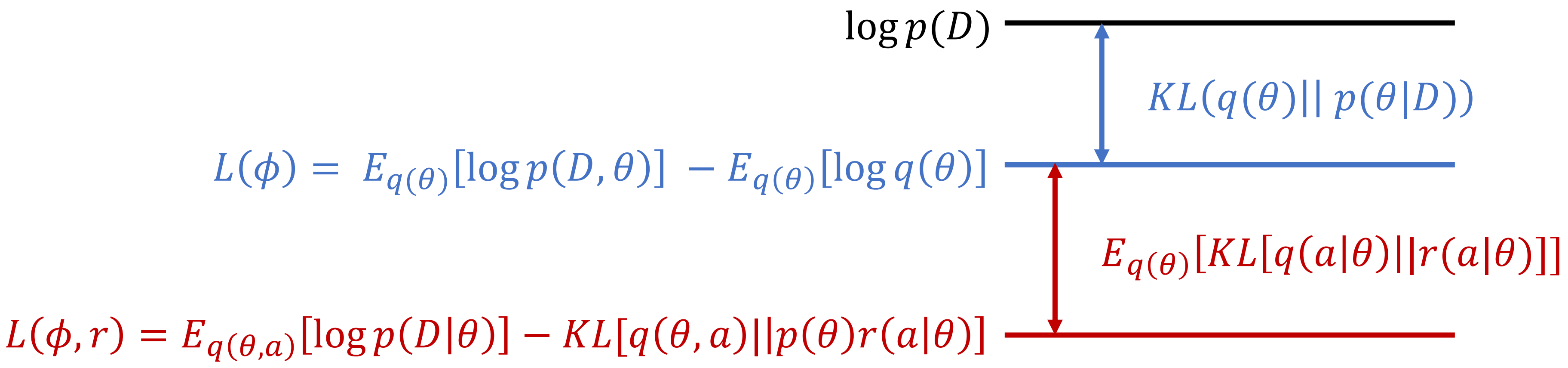
$E_{q(\theta)}[KL[q(a|\theta)||r(a|\theta)]]$

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

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 $r(a|\theta)$:



$$\log p(D)$$

$$KL(q(\theta)||\, p(\theta|D))$$

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

$$E_{q(\theta)}[KL[q(a|\theta)||r(a|\theta)]]$$

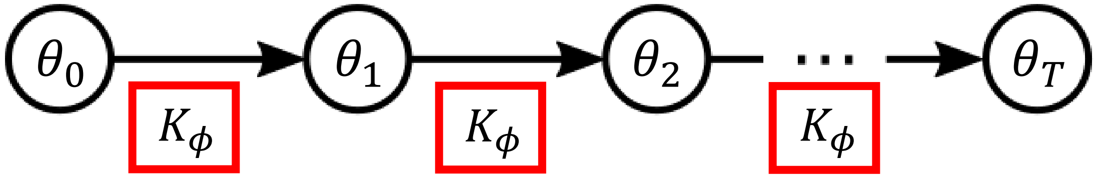
$$L(\phi, r) = E_{q(\theta,a)}[\log p(D|\theta)] - KL[q(\theta, a)||p(\theta)r(a|\theta)]$$

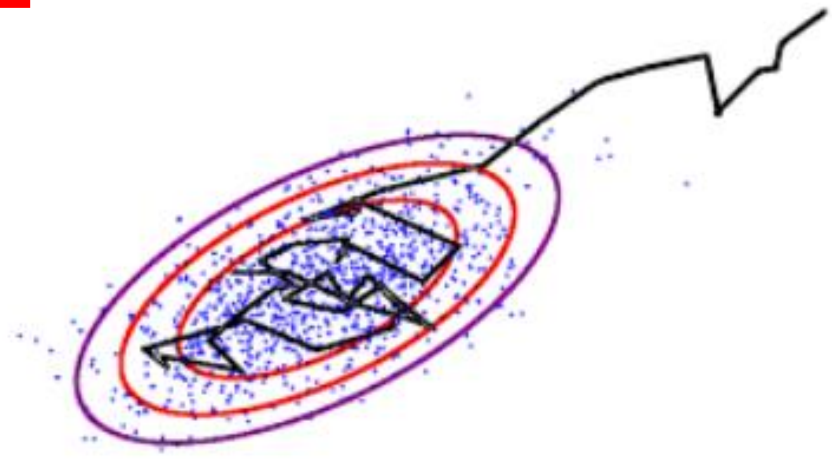
- Optimize $r(a|\theta)$ to close the gap!
- $L(\phi, r)$ estimated by Monte Carlo:$a_k \sim q(a), \theta_k \sim q(\theta \mid a_k)$

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

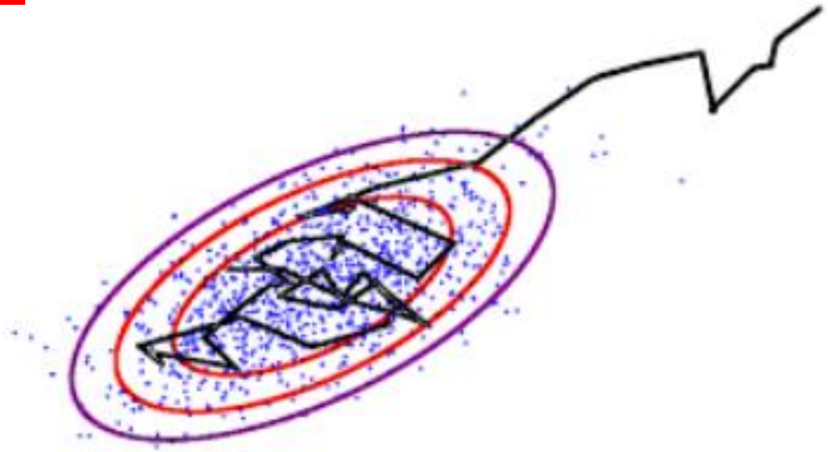
- Hierarchical mixture distributions for $q(\theta, a)$
  - VI-MCMC hybrid: build $q(\theta)$ with a Markov Chain:

$$\begin{tikzcd}
\theta_0 \arrow[r, "K_\phi"'] & \theta_1 \arrow[r, "K_\phi"'] & \theta_2 \arrow[r, "K_\phi"', "\cdots"] & \theta_T
\end{tikzcd}$$

learn the transition kernel with VI:

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

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



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

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

approximate posterior moments

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

Implicit Approximate Posteriors

- Two quantities computed in (approximate) Bayesian inference:

approximate Bayesian predictive

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

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

approximate posterior moments

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

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

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

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

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

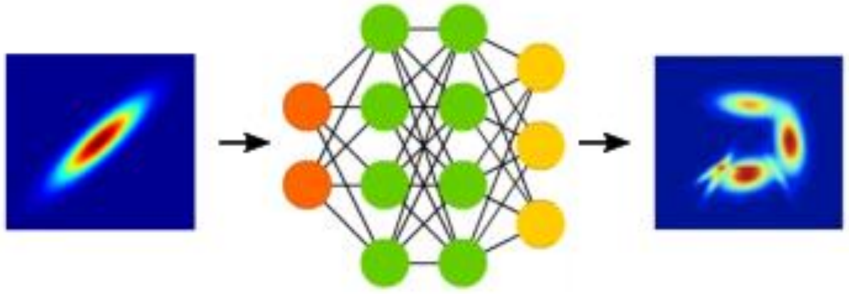
approximate posterior moments

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

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

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

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

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

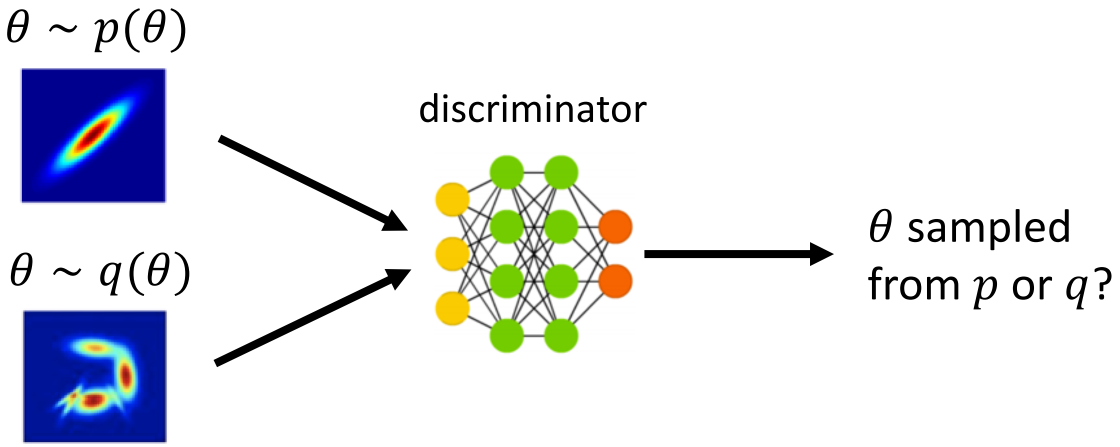
Li and Turner. Gradient Estimators for Implicit Models. ICLR 2018

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

Implicit Approximate Posteriors

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

Approximated by using a discriminator (AVB):



$\theta \sim p(\theta)$

$\theta \sim q(\theta)$

discriminator

θ sampled from p or q?

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

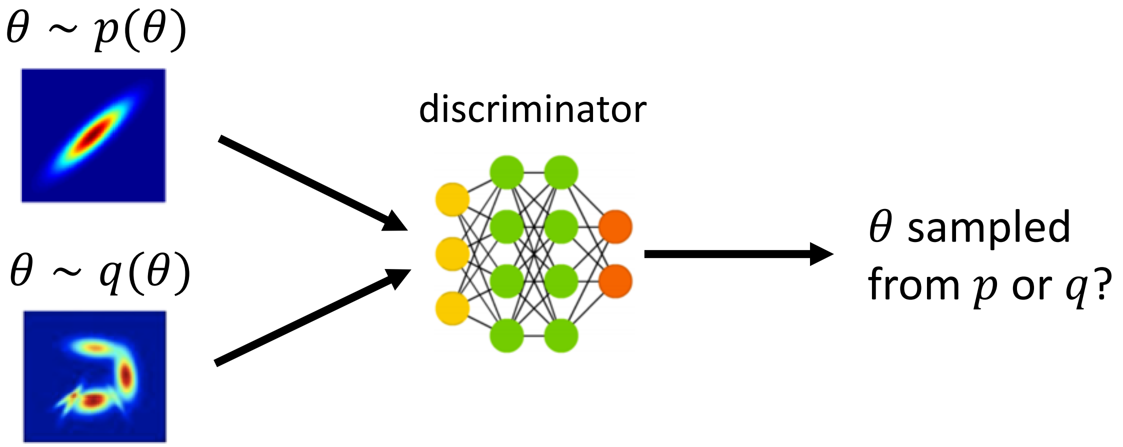
Li and Turner. Gradient Estimators for Implicit Models. ICLR 2018

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

Implicit Approximate Posteriors

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

Approximated by using a discriminator (AVB):

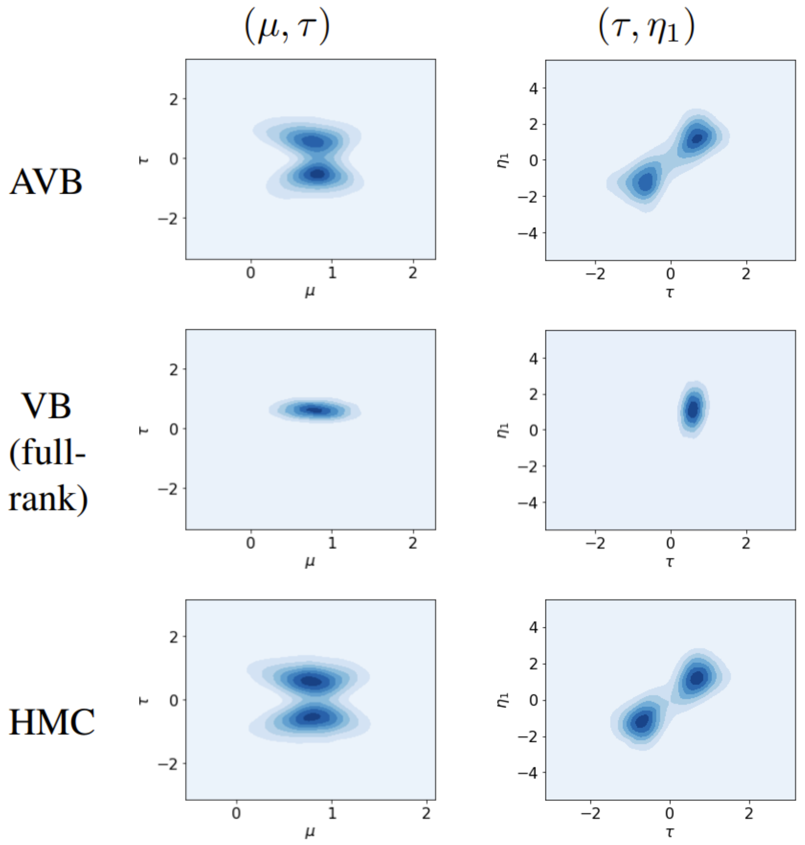


$\theta \sim p(\theta)$

$\theta \sim q(\theta)$

discriminator

θ sampled from p or q?



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

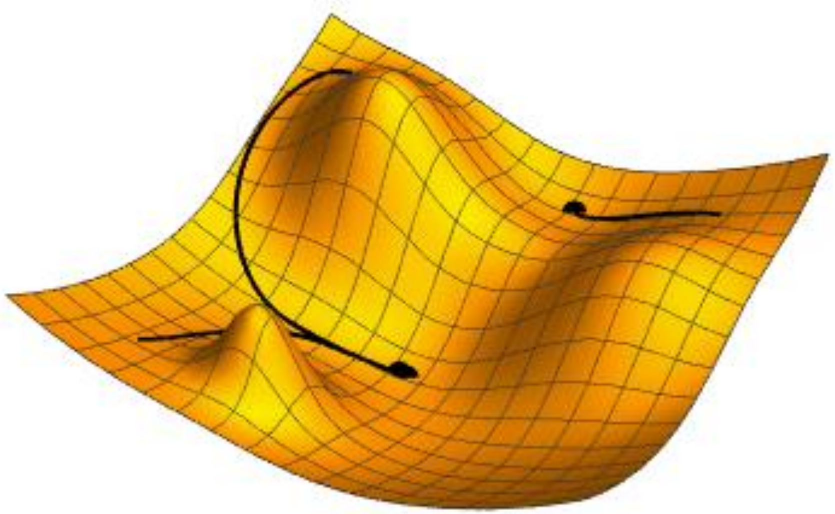
Tran et al. Hierarchical Implicit Models and Likelihood-Free Variational Inference. NeurIPS 2017

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



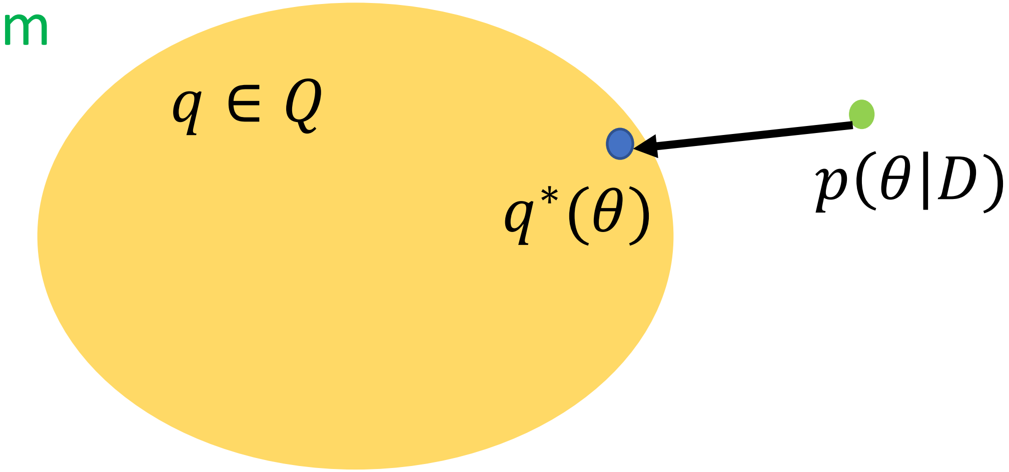
VI Ingredients

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

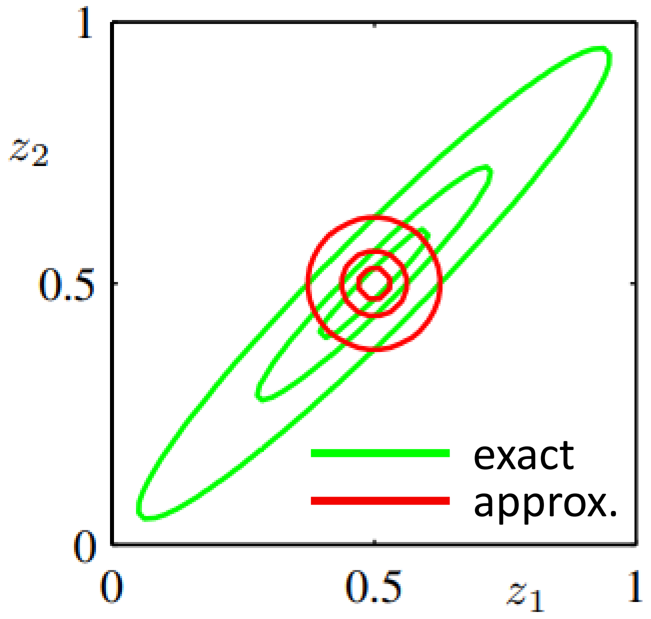
p: your model design

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

KL: defines the algorithm



Does It Work?



VI underestimate the uncertainty

Bishop (2006). Pattern recognition and machine learning. Springer.

(Rényi) α-Divergence

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

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

$\alpha = 1$

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

VI with α-Divergence

ELBO

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

Variational Rényi bound:

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

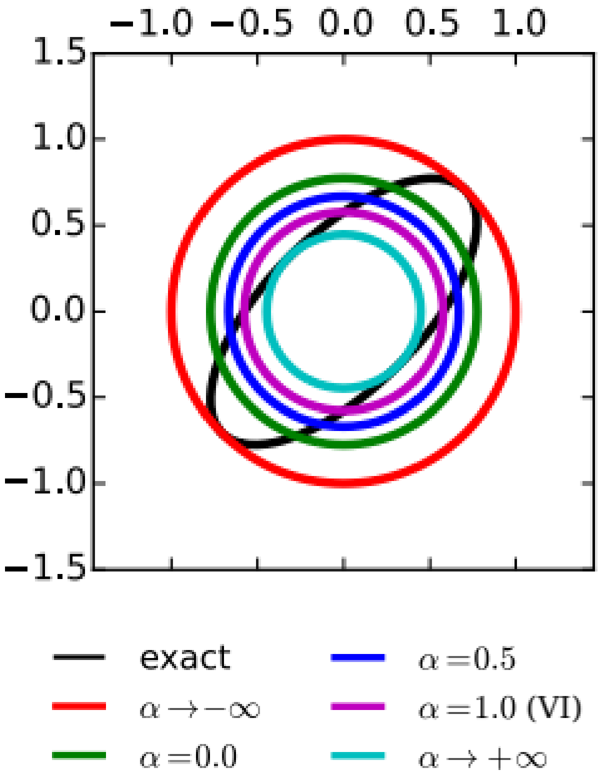
$$\lim_{\alpha\to1} L_\alpha = L$$

Li and Turner. Rényi Divergence Variational Inference. NeurIPS 2016

Dieng et al. Variational Inference via χ-Upper Bound Minimization. NeurIPS 2017

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

Does It Work?

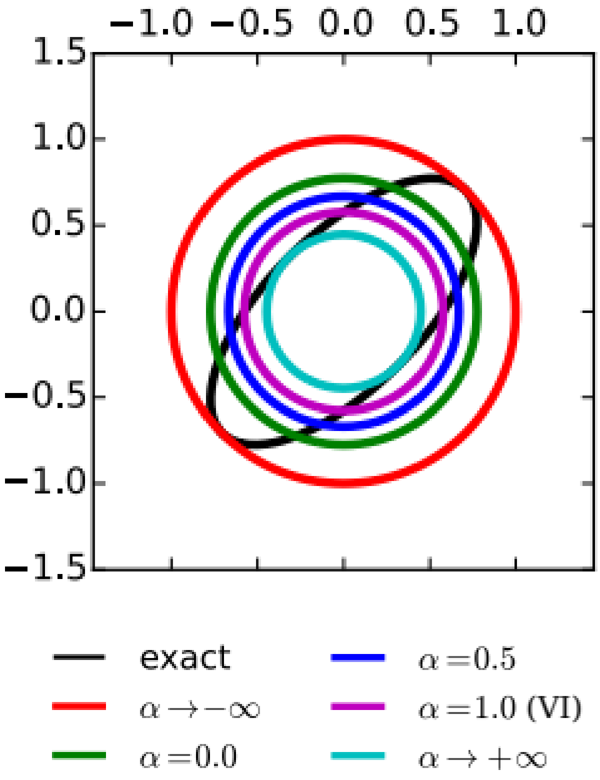


Li and Turner. Rényi Divergence Variational Inference. NeurIPS 2016

Dieng et al. Variational Inference via χ-Upper Bound Minimization. NeurIPS 2017

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

Does It Work?



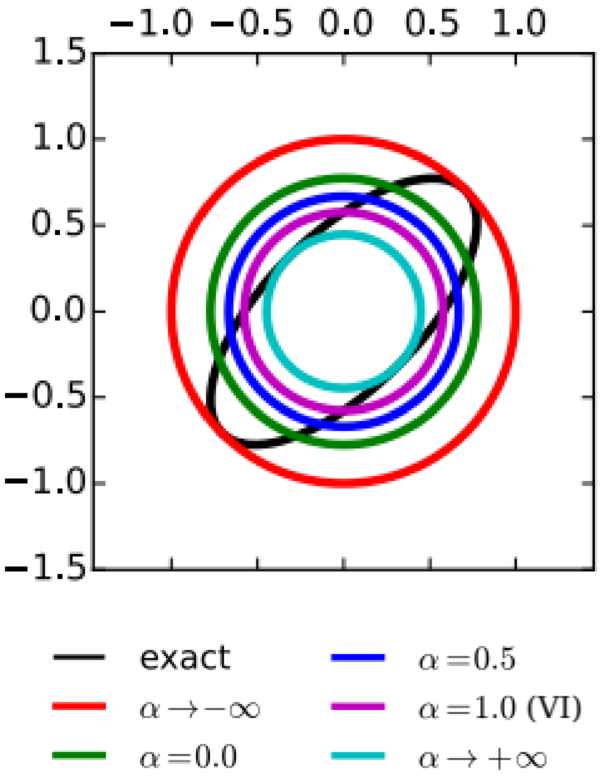
How to choose alpha?

Li and Turner. Rényi Divergence Variational Inference. NeurIPS 2016

Dieng et al. Variational Inference via χ-Upper Bound Minimization. NeurIPS 2017

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

Does It Work?



How to choose alpha?

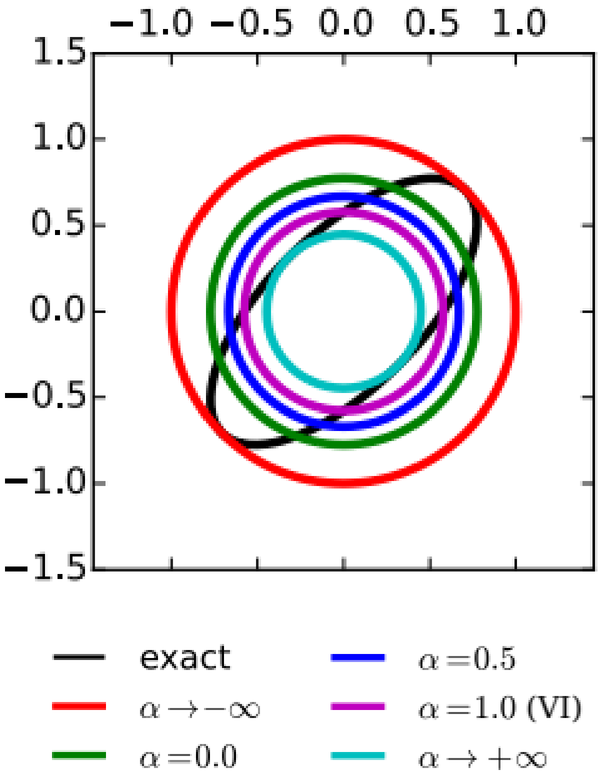
$$L_\alpha = \frac{1}{1-\alpha} E_{\theta\sim q_\phi}\left[\left(\log\frac{\mathrm{p}(\mathrm{D},\theta)}{q_\phi(\theta)}\right)^{1-\alpha}\right]$$

Li and Turner. Rényi Divergence Variational Inference. NeurIPS 2016

Dieng et al. Variational Inference via χ-Upper Bound Minimization. NeurIPS 2017

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

Does It Work?



How to choose alpha?

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

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 χ-Upper Bound Minimization. NeurIPS 2017

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

Revisiting Perturbation Theory for VI

$$V(x,z) \equiv \log q_\lambda(z) - \log p(x,z)$$

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

Bamler*, **Zhang*** et al. Perturbative Black Box Variational Inference. NeurIPS 2017

Revisiting Perturbation Theory for VI

$$V(x,z) \equiv \log q_\lambda(z) - \log p(x,z)$$

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

Taylor expansion around $\beta = 1$:

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

Bamler*, **Zhang*** et al. Perturbative Black Box Variational Inference. NeurIPS 2017

Revisiting Perturbation Theory for VI

$$V(x,z) \equiv \log q_\lambda(z) - \log p(x,z)$$

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

Taylor expansion around $\beta = 1$:

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

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

Bamler*, **Zhang*** et al. Perturbative Black Box Variational Inference. NeurIPS 2017

Revisiting Perturbation Theory for VI

$$V(x,z) \equiv \log q_\lambda(z) - \log p(x,z)$$

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

Taylor expansion around $\beta = 1$:

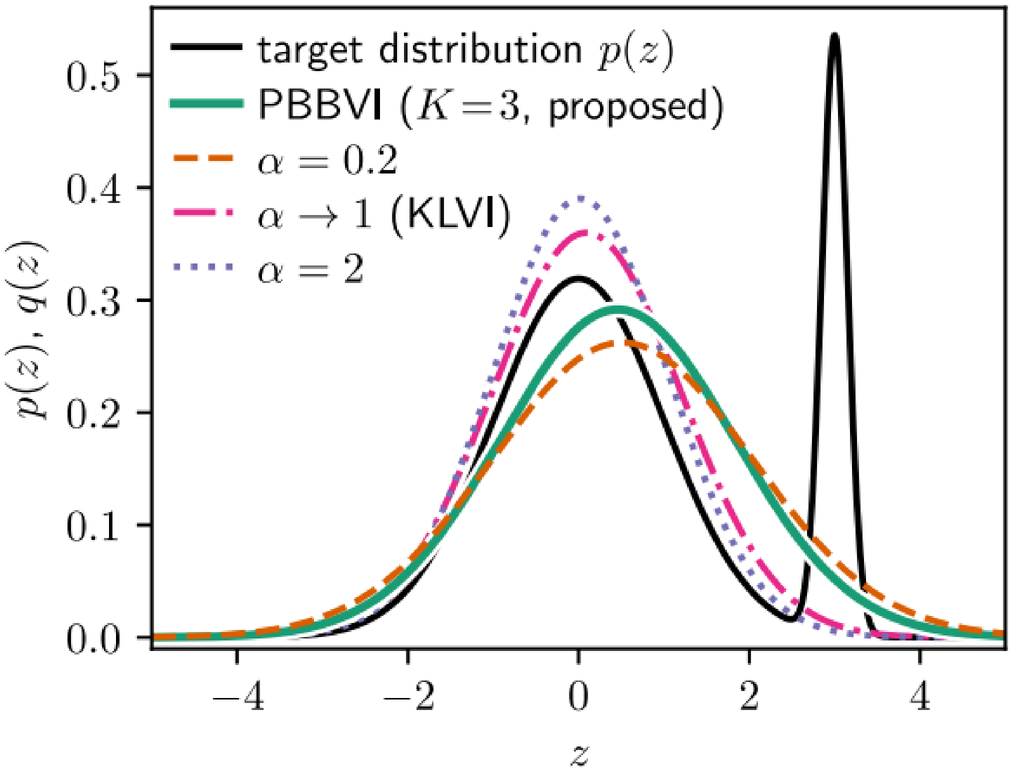
$$\log p(x) \approx \boxed{E_{q_\lambda}[-V]} + \frac{1}{2}\left[\left(V - E_{q_\lambda}[-V]\right)^2\right] - \frac{1}{3!}\left[\left(V - E_{q_\lambda}[-V]\right)^3\right] + \frac{1}{4!}\left[\left(V - E_{q_\lambda}[-V]\right)^4\right] - \ \ldots$$

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

Truncation at any odd number term provides a bound.

Bamler*, **Zhang*** et al. Perturbative Black Box Variational Inference. NeurIPS 2017

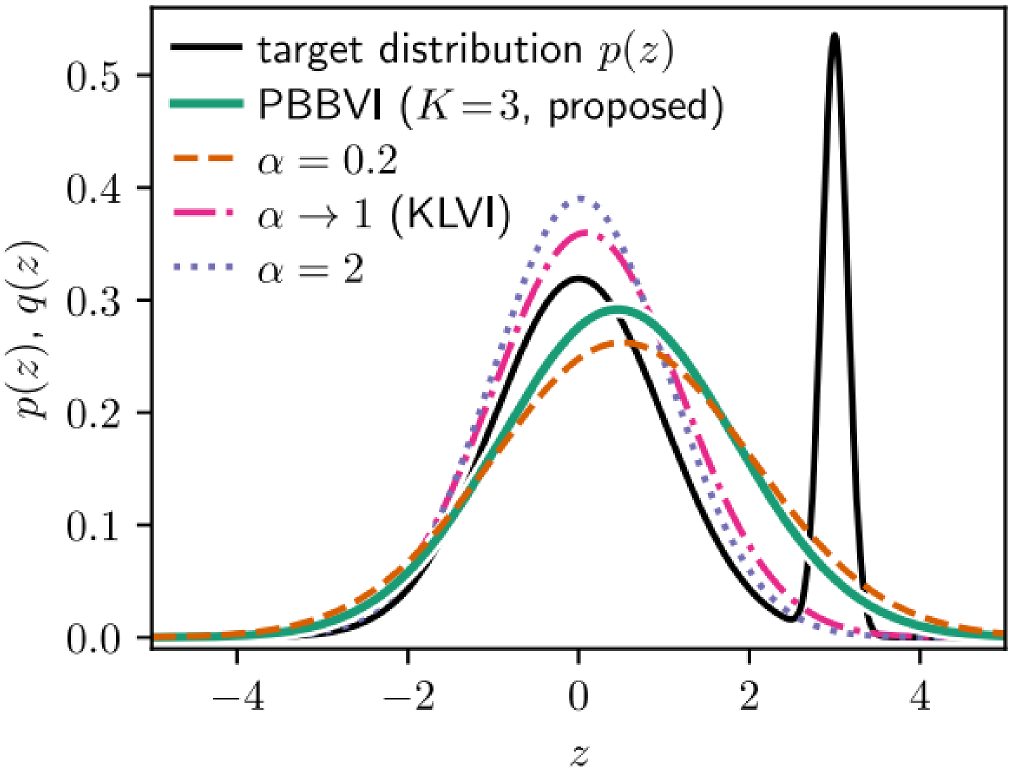
Behaviour of PBBVI



- Better uncertainty estimation than KLVI
- Better bias-variance trade-off comparing to α-VI

Bamler*, **Zhang*** et al. Perturbative Black Box Variational Inference. NeurIPS 2017

Behaviour of PBBVI



- Better uncertainty estimation than KLVI
- Better bias-variance trade-off comparing to α-VI

Where do we truncate?
Is it flexible enough?

Bamler*, **Zhang*** et al. Perturbative Black Box Variational Inference. NeurIPS 2017

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

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

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

$$f(t) = \mathrm{t}\log t \longrightarrow KL(p||q)$$

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

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:

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

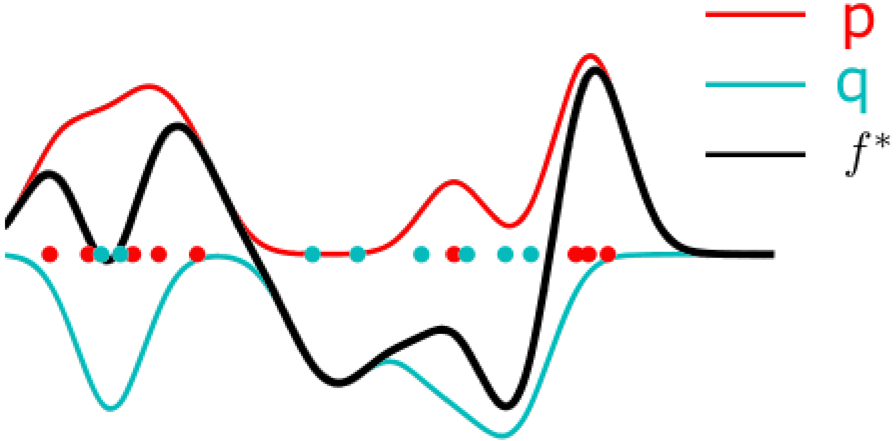


Figure adapted, source: Dougal Sutherland

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 $\nabla_z \log p(z|x) = \nabla_z \log p(z,x)$

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

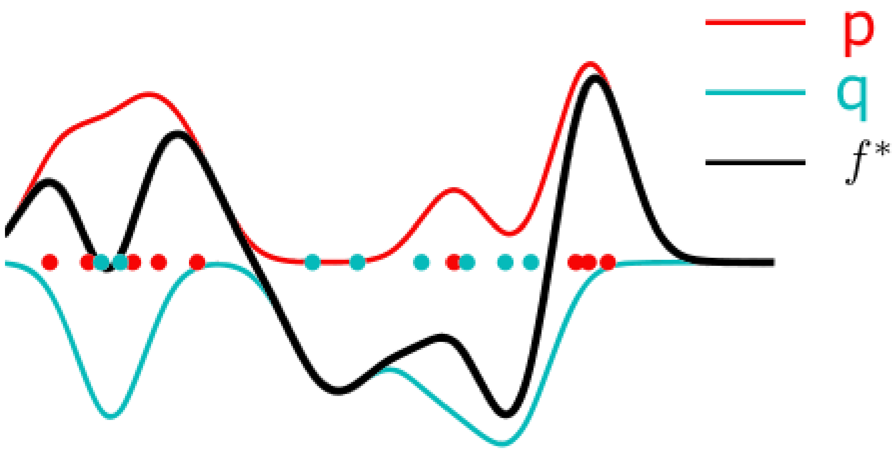
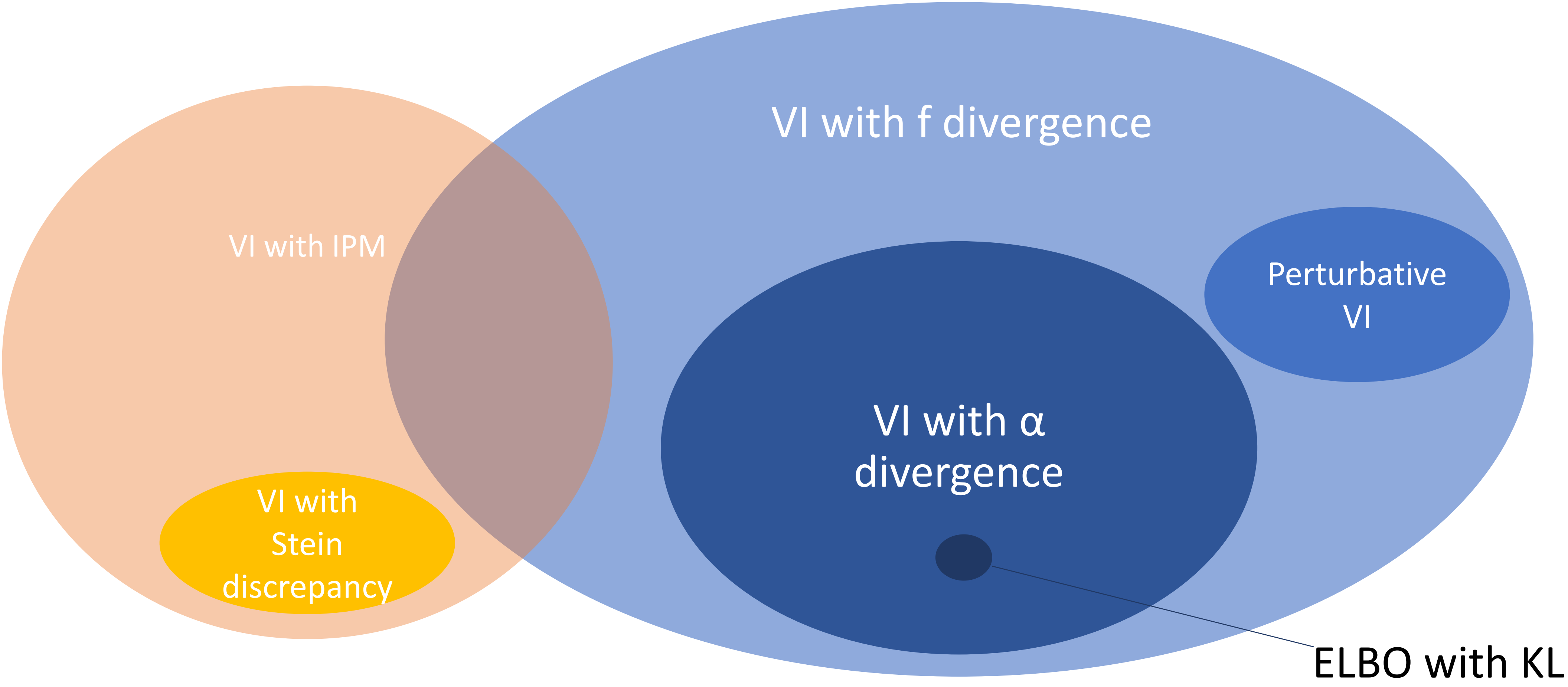


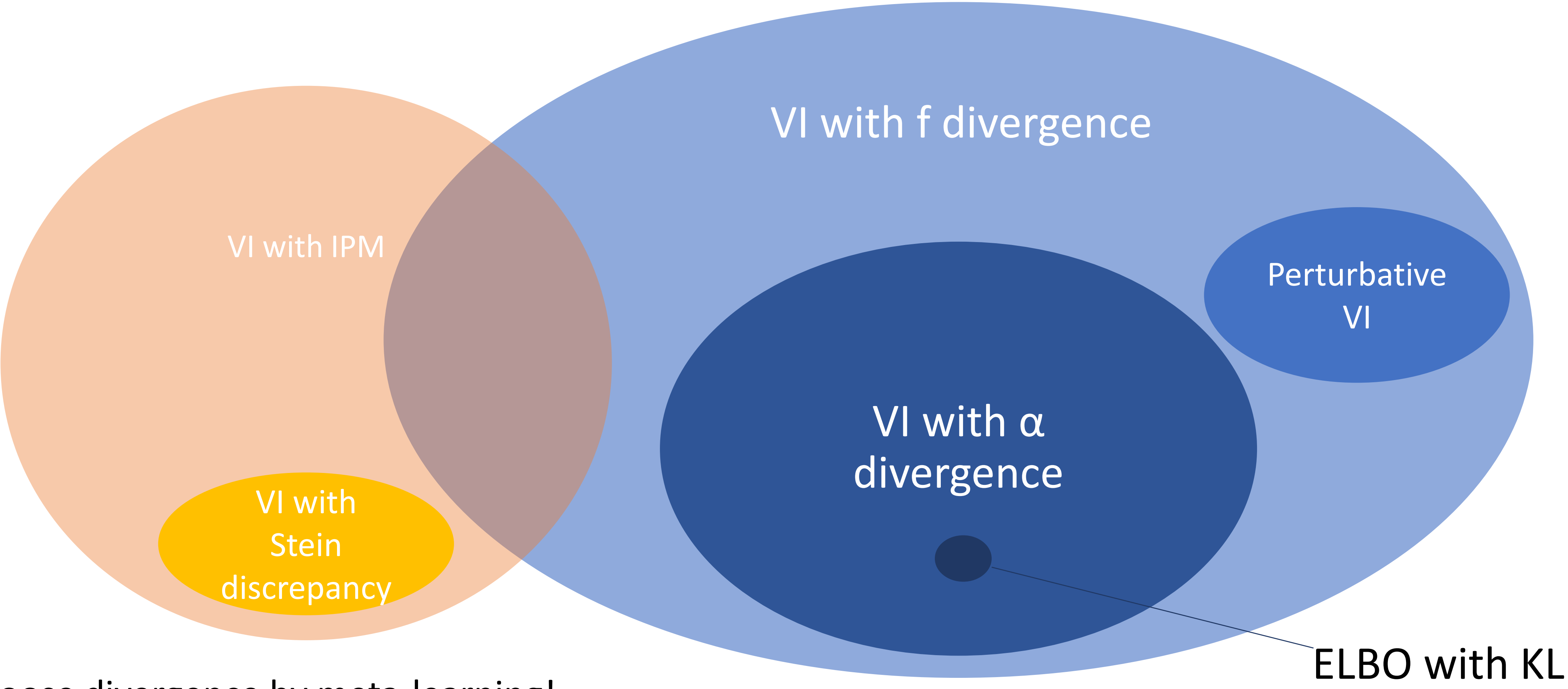
Figure adapted, source: Dougal Sutherland

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:

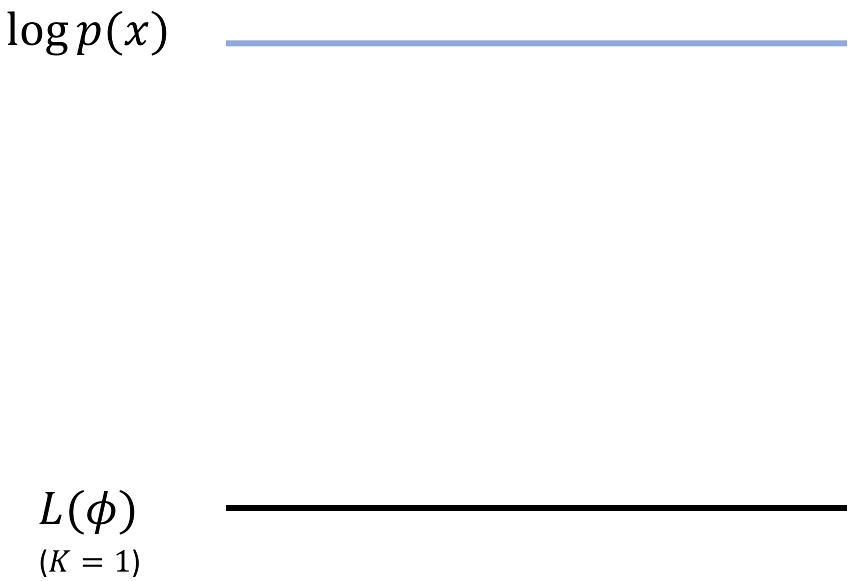
$$L_K(\phi) = E_{z_1,\ldots,z_K \sim\, q(z)}\left[\log \underbrace{\frac{1}{K}\sum_{k=1}^{K}\frac{p(x,z_k)}{q(z_k)}}_{\text{Importance sampling estimate of } p(x)}\right]$$

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:

$$L_K(\phi) = E_{z_1,\ldots,z_K \sim\, q(z)}\left[\log \underbrace{\frac{1}{K}\sum_{k=1}^{K}\frac{p(x,z_k)}{q(z_k)}}_{\text{Importance sampling estimate of } p(x)}\right]$$

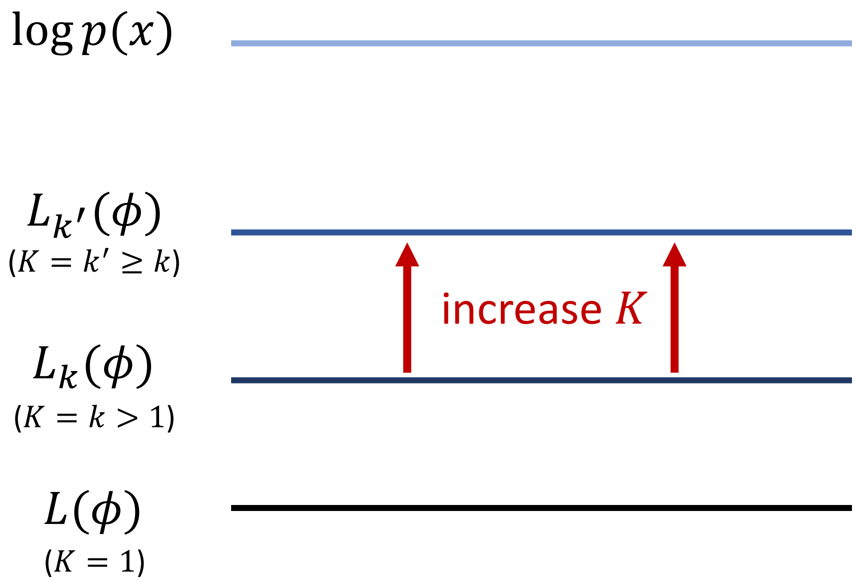


Burda et al. Importance Weighted Auto-encoders. ICLR 2016
Naesseth et al. Variational Sequential Monte Carlo. AISTATS 2018
Maddison et al. Filtering Variational Objectives. NeurIPS 2017
Le et al. Auto-encoding Sequential Monte Carlo. ICLR 2018
Masrani et al. The Thermodynamic Variational Objective. NeurIPS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

Improved Monte Carlo Bounds

- Importance weighted auto-encoder (IWAE) bound:

$$L_K(\phi) = E_{z_1,\ldots,z_K \sim\, q(z)}\left[\log \underbrace{\frac{1}{K}\sum_{k=1}^{K}\frac{p(x, z_k)}{q(z_k)}}_{\text{Importance sampling estimate of } p(x)}\right]$$

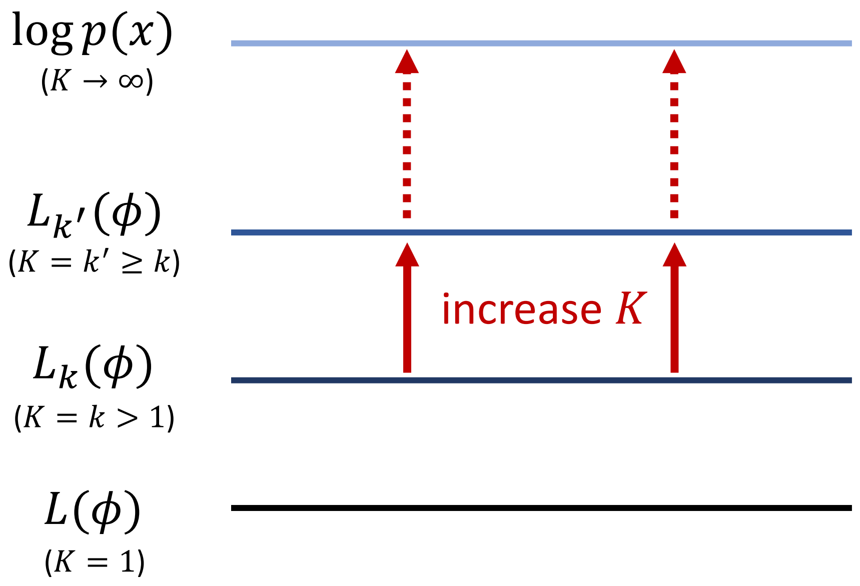


Burda et al. Importance Weighted Auto-encoders. ICLR 2016
Naesseth et al. Variational Sequential Monte Carlo. AISTATS 2018
Maddison et al. Filtering Variational Objectives. NeurIPS 2017
Le et al. Auto-encoding Sequential Monte Carlo. ICLR 2018
Masrani et al. The Thermodynamic Variational Objective. NeurIPS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

Improved Monte Carlo Bounds

- Importance weighted auto-encoder (IWAE) bound:

$$L_K(\phi) = E_{z_1,\ldots,z_K \sim q(z)}\left[\log \underbrace{\frac{1}{K}\sum_{k=1}^{K}\frac{p(x,z_k)}{q(z_k)}}_{\text{Importance sampling estimate of } p(x)}\right]$$

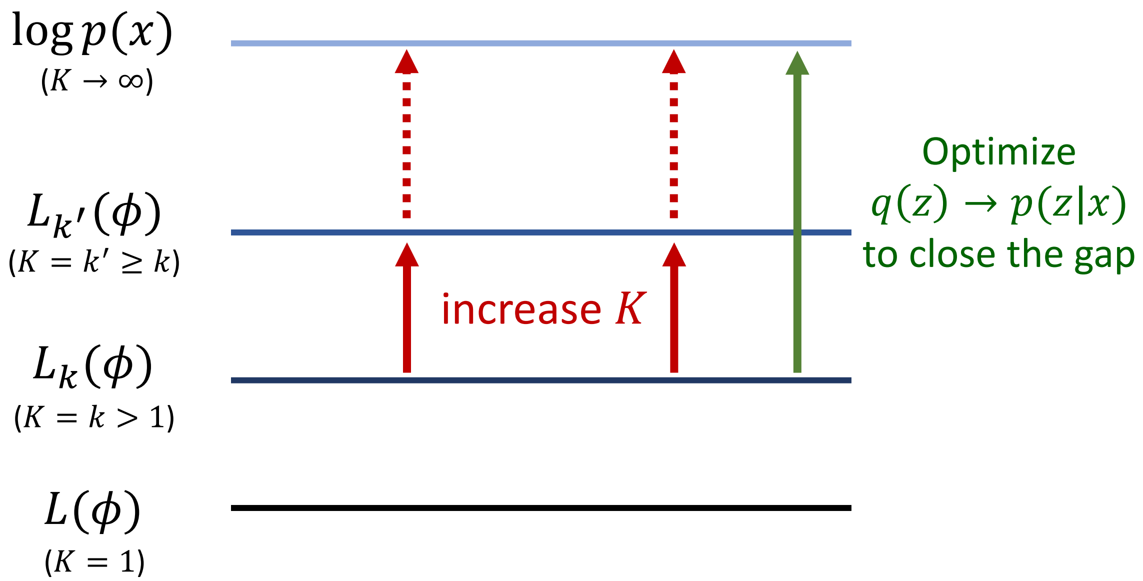


Burda et al. Importance Weighted Auto-encoders. ICLR 2016
Naesseth et al. Variational Sequential Monte Carlo. AISTATS 2018
Maddison et al. Filtering Variational Objectives. NeurIPS 2017
Le et al. Auto-encoding Sequential Monte Carlo. ICLR 2018
Masrani et al. The Thermodynamic Variational Objective. NeurIPS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

Improved Monte Carlo Bounds

- Importance weighted auto-encoder (IWAE) bound:

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



$\log p(x)$
$(K \to \infty)$

$L_{k'}(\phi)$
$(K = k' \geq k)$

$L_k(\phi)$
$(K = k > 1)$

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

increase K

Optimize
$q(z) \to p(z|x)$
to close the gap

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:

$$E_{q(h)}[R(h,x)] = p(x) \quad \Rightarrow \quad \underbrace{E_{q(h)}[\log R(h,x)] \leq \log p(x)}_{\text{Jensen's inequality}}$$

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

Improved Monte Carlo Bounds

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

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

Jensen's inequality

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

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

Improved Monte Carlo Bounds

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

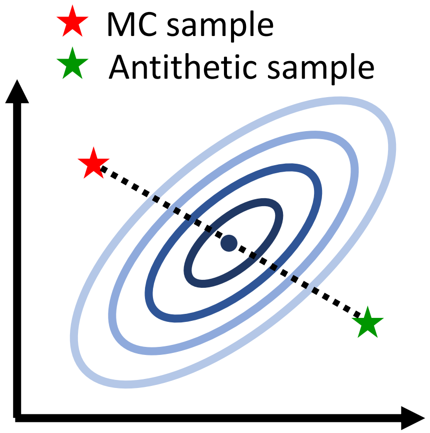
$$E_{q(h)}[R(h,x)] = p(x) \quad \Rightarrow \quad \underline{E_{q(h)}[\log R(h,x)] \leq \log p(x)}$$

Jensen's inequality

  - 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 $q(z)$:

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



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

Improved Monte Carlo Bounds

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

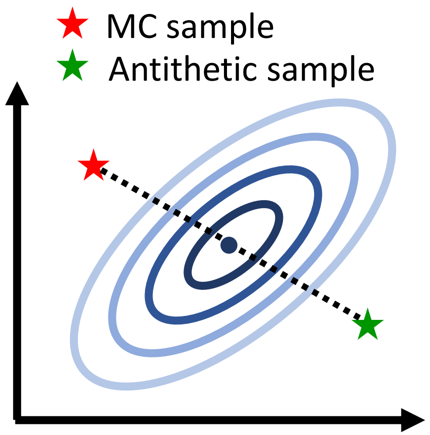
$$E_{q(h)}[R(h,x)] = p(x) \quad \Rightarrow \quad \underline{E_{q(h)}[\log R(h,x)] \leq \log p(x)}$$

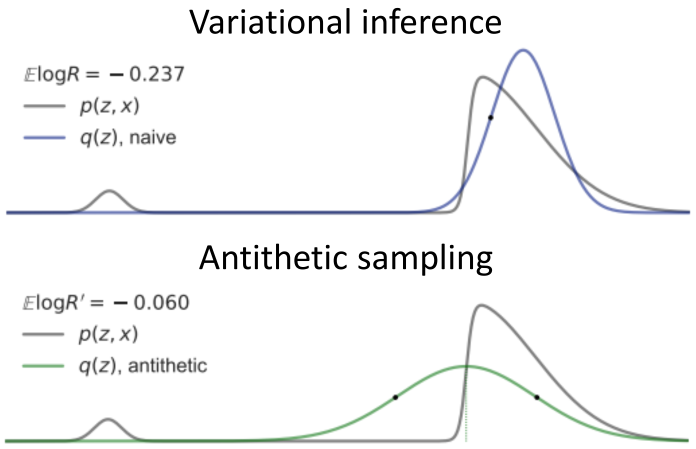
Jensen’s inequality

  - 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 $q(z)$:

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

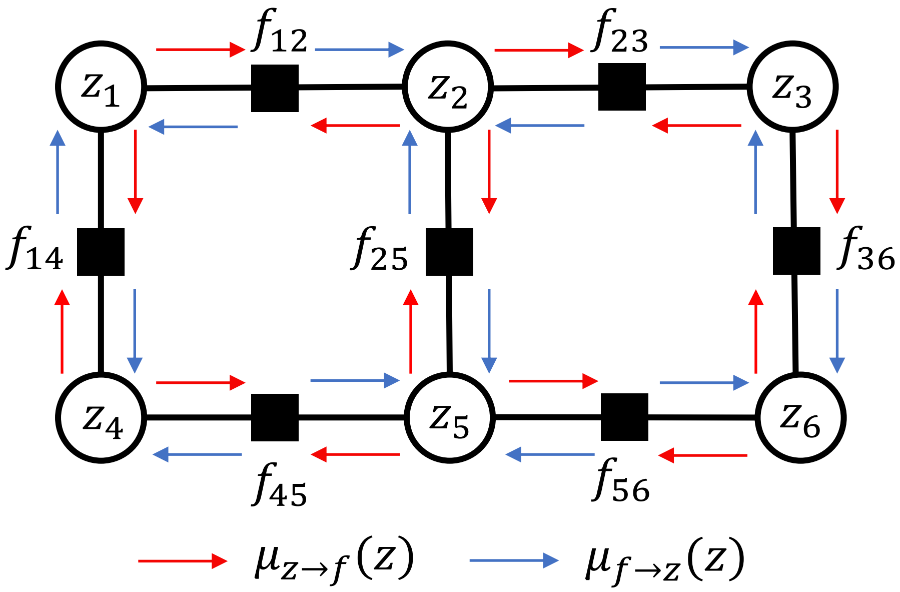




Domke and Sheldon. Divide and Couple: Using Monte Carlo Variational Objectives for Posterior Approximation. NeurIPS 2019

Free-energy as an Objective

- Bethe free-energy & message passing:

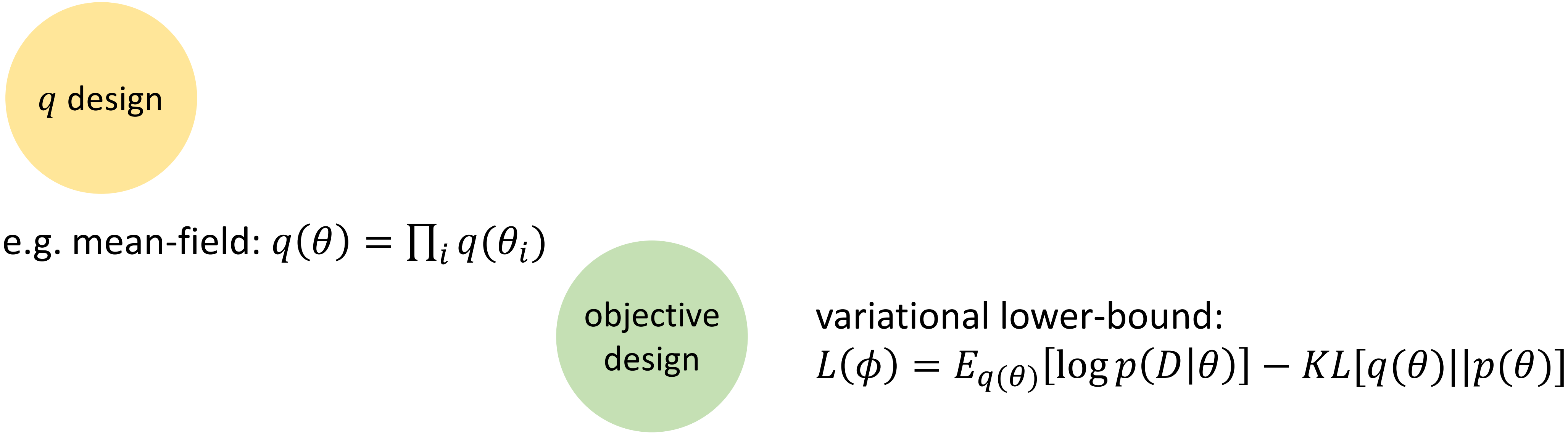


- Both q and the inference algorithm are defined by the *factor graph*
- Optimal q achieved at the fixed point of the *Bethe free energy*

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



q design

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

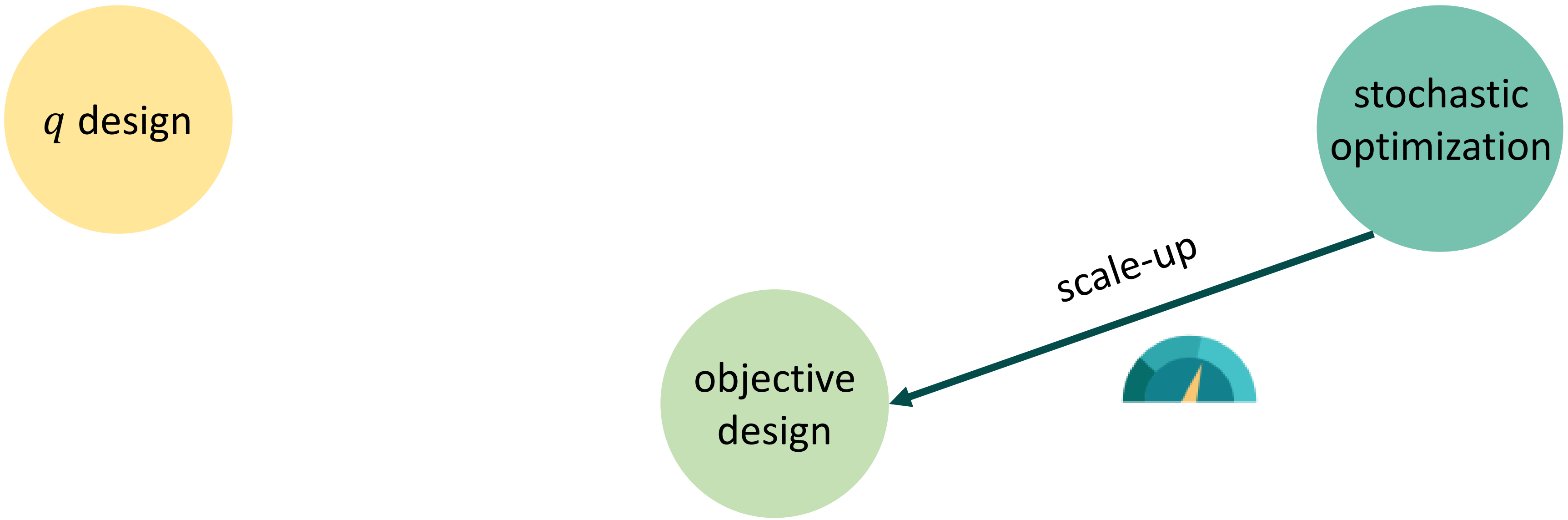
objective design

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

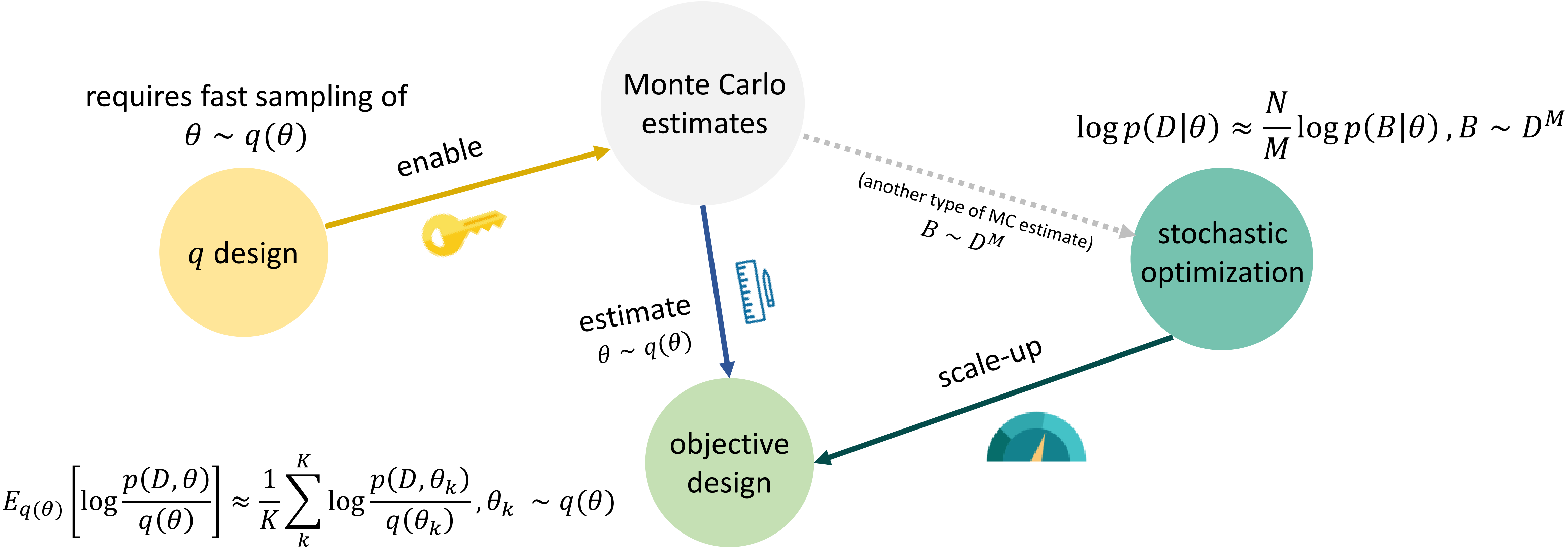
Landscape of Advances

mini-batch training:

$$\log p(D|\theta) \approx \frac{N}{M} \log p(B|\theta)\,, B \sim D^{M}$$



Landscape of Advances



requires fast sampling of
$\theta \sim q(\theta)$

q design

enable

Monte Carlo estimates

estimate
$\theta \sim q(\theta)$

(another type of MC estimate)
$B \sim D^M$

stochastic optimization

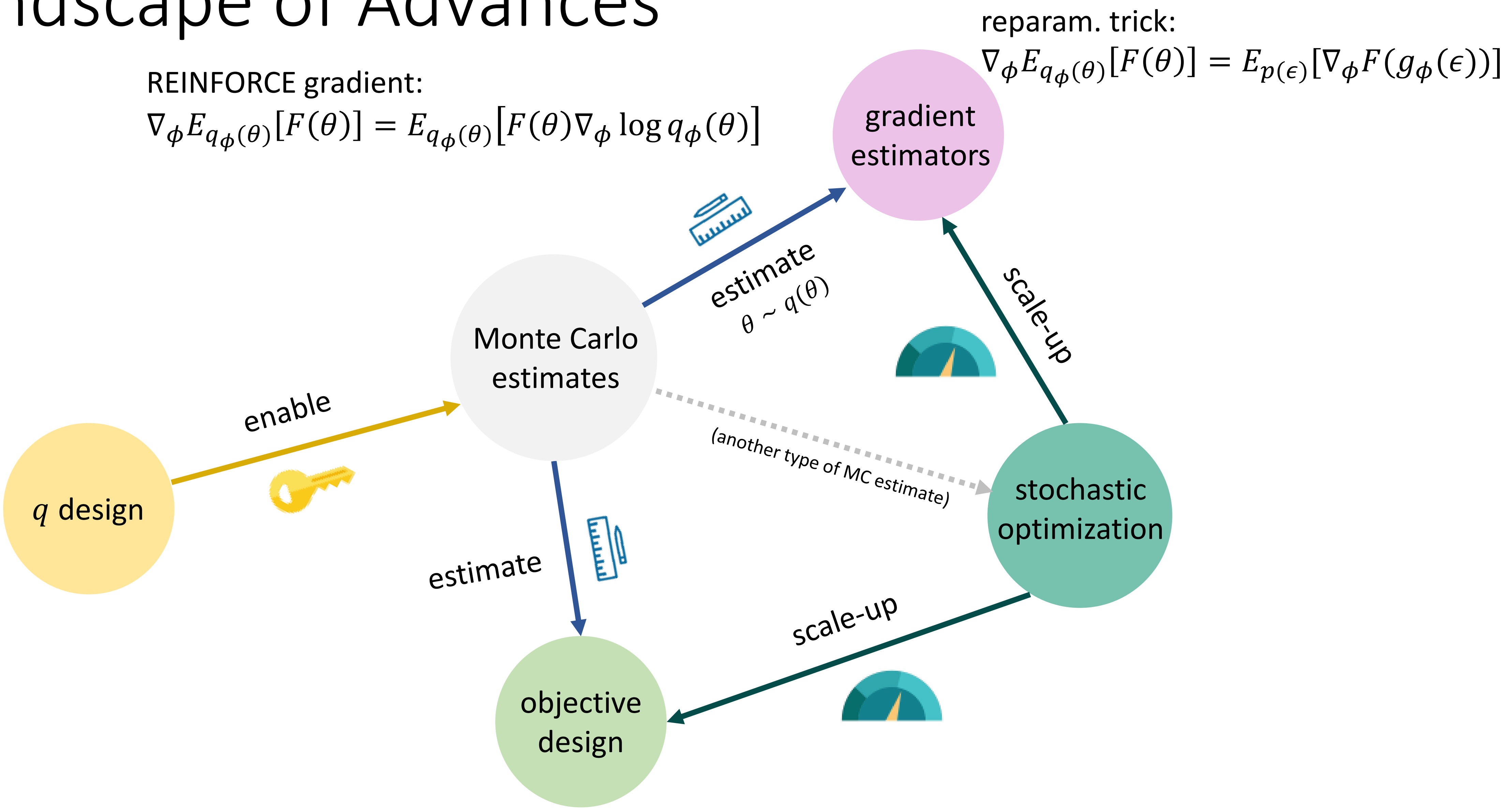
$$\log p(D|\theta) \approx \frac{N}{M} \log p(B|\theta)\,, B \sim D^M$$

scale-up

objective design

$$E_{q(\theta)}\left[\log \frac{p(D,\theta)}{q(\theta)}\right] \approx \frac{1}{K} \sum_{k}^{K} \log \frac{p(D,\theta_k)}{q(\theta_k)}, \theta_k \ \sim q(\theta)$$

Landscape of Advances



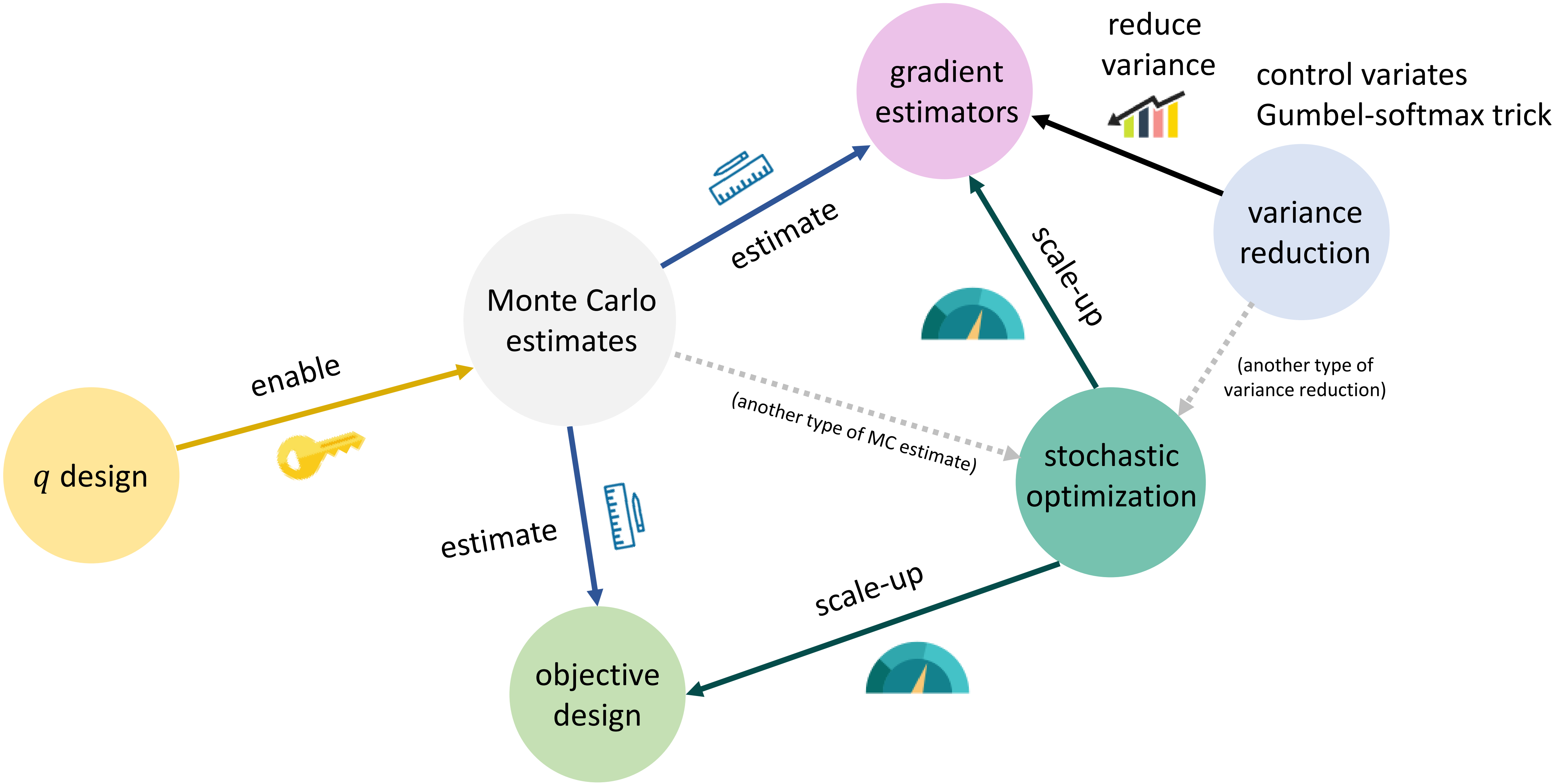
REINFORCE gradient:

$$\nabla_\phi E_{q_\phi(\theta)}[F(\theta)] = E_{q_\phi(\theta)}\left[F(\theta)\nabla_\phi \log q_\phi(\theta)\right]$$

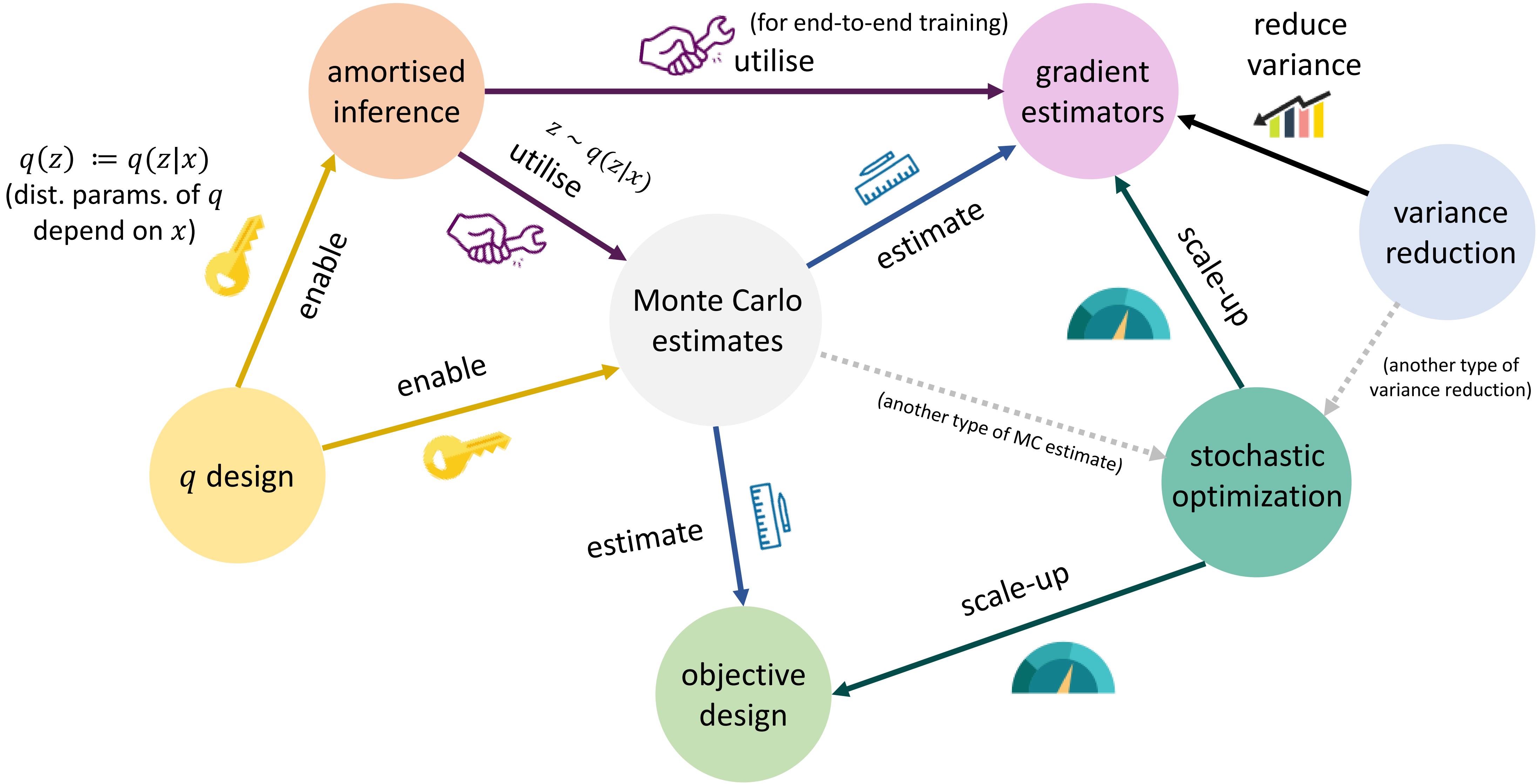
reparam. trick:

$$\nabla_\phi E_{q_\phi(\theta)}[F(\theta)] = E_{p(\epsilon)}[\nabla_\phi F(g_\phi(\epsilon))]$$

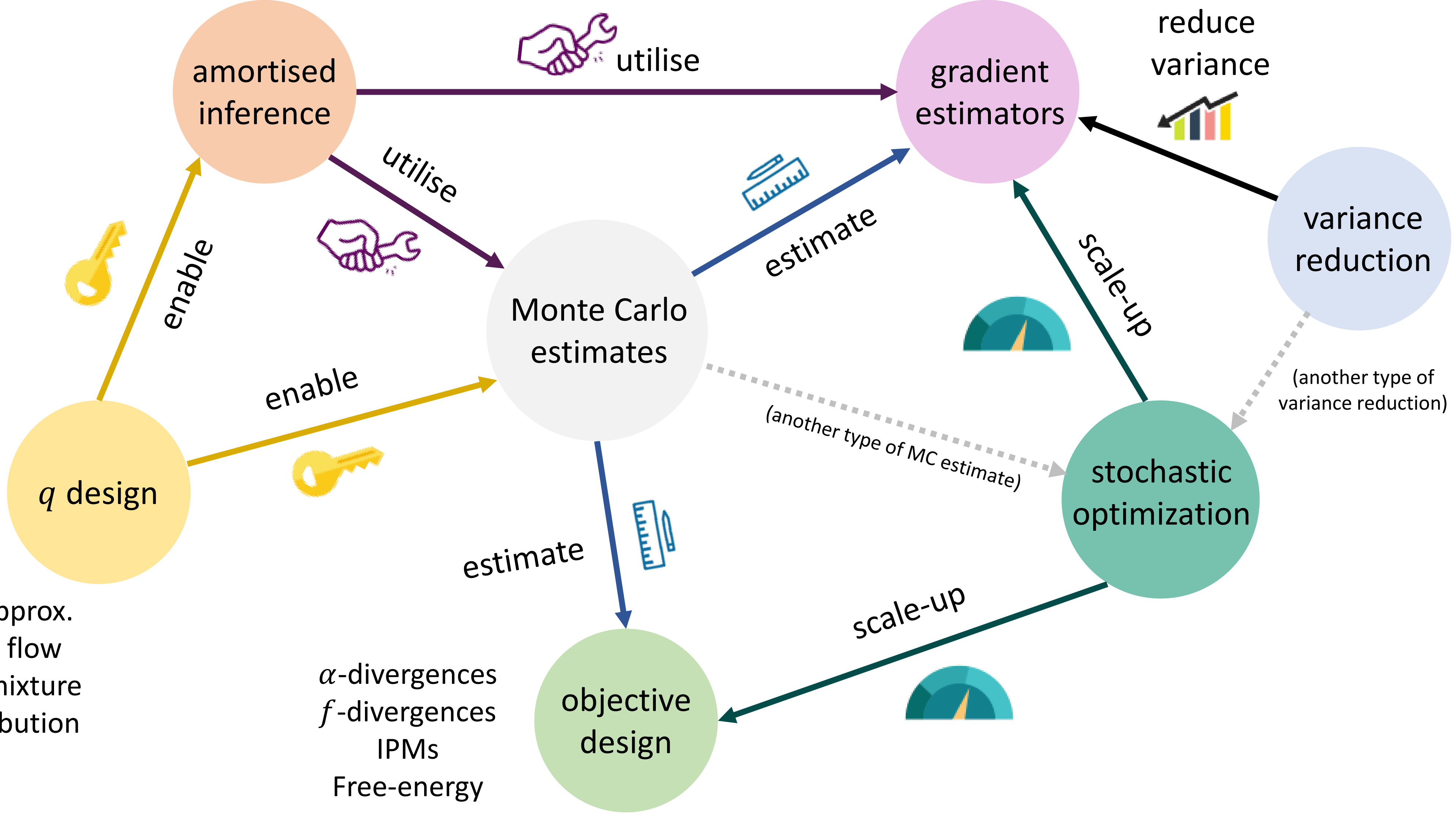
Landscape of Advances



Landscape of Advances



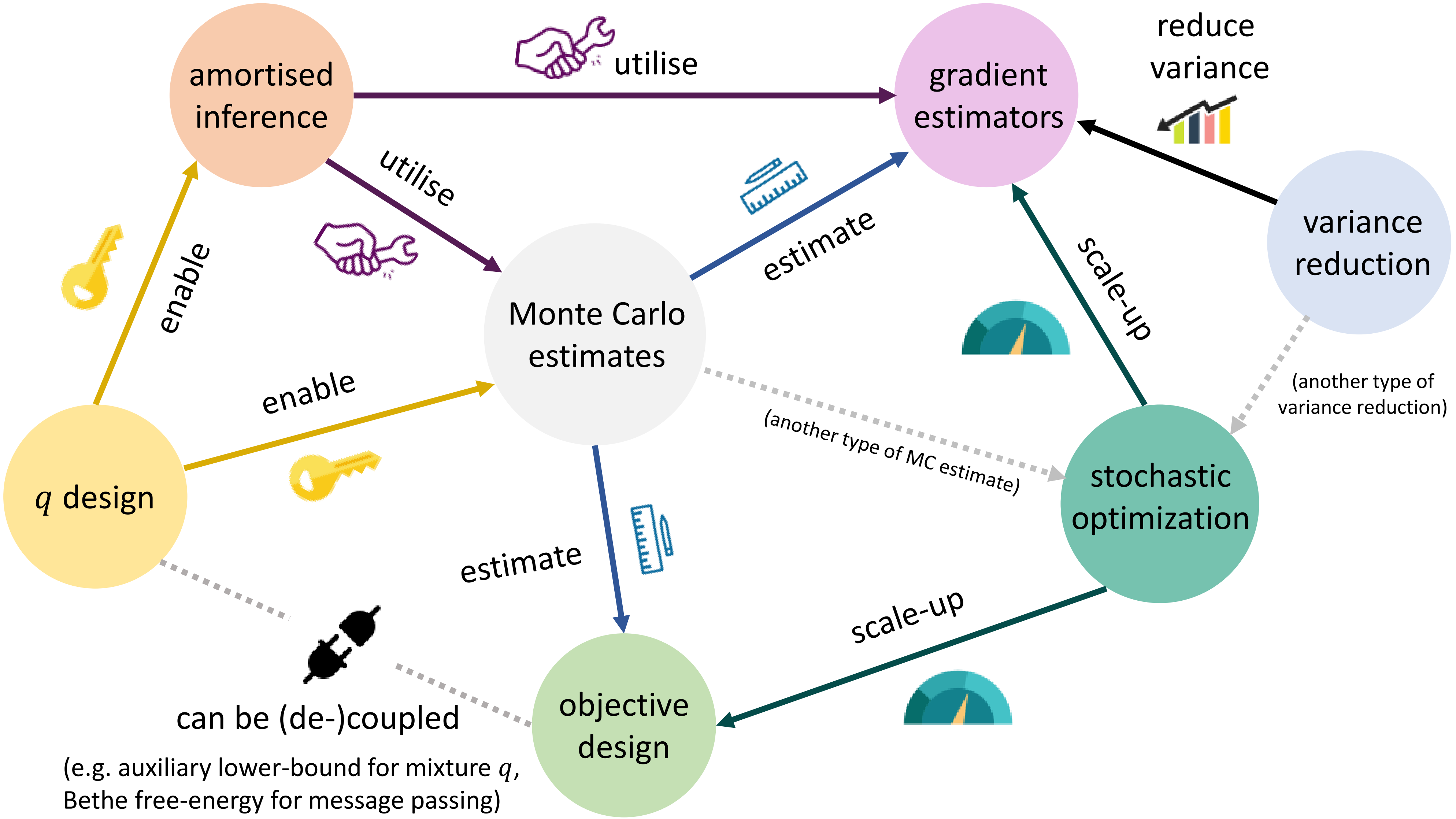
Landscape of Advances



structured approx.
normalizing flow
hierarchical mixture
implicit distribution

α-divergences
f-divergences
IPMs
Free-energy

Landscape of Advances



amortised inference

gradient estimators

Monte Carlo estimates

variance reduction

q design

stochastic optimization

objective design

utilise

utilise

enable

enable

estimate

estimate

reduce variance

scale-up

scale-up

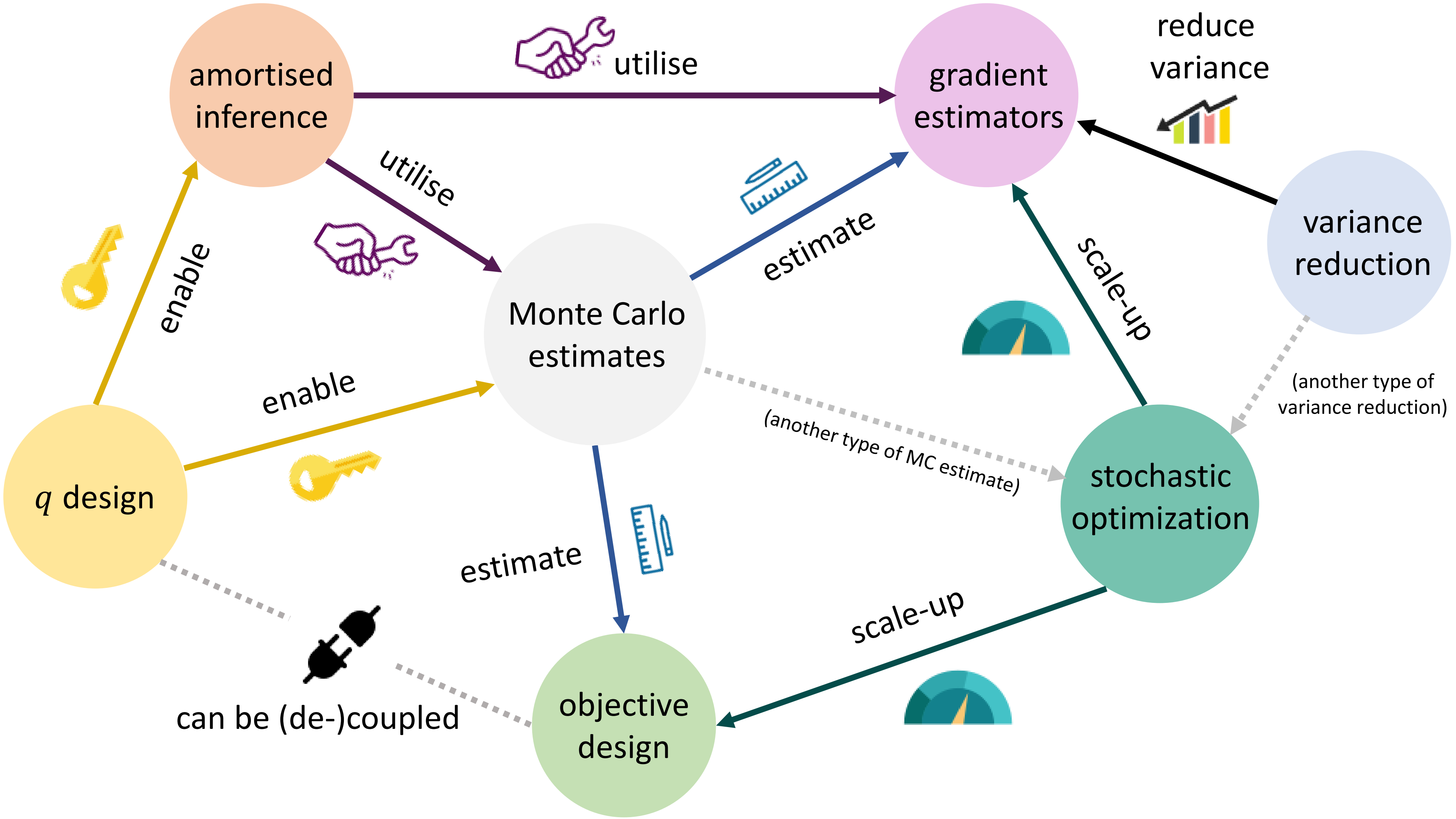
(another type of MC estimate)

(another type of variance reduction)

can be (de-)coupled

(e.g. auxiliary lower-bound for mixture q, Bethe free-energy for message passing)

Landscape of Advances



Part III: Applications

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

Why Estimating Uncertainty in DL?

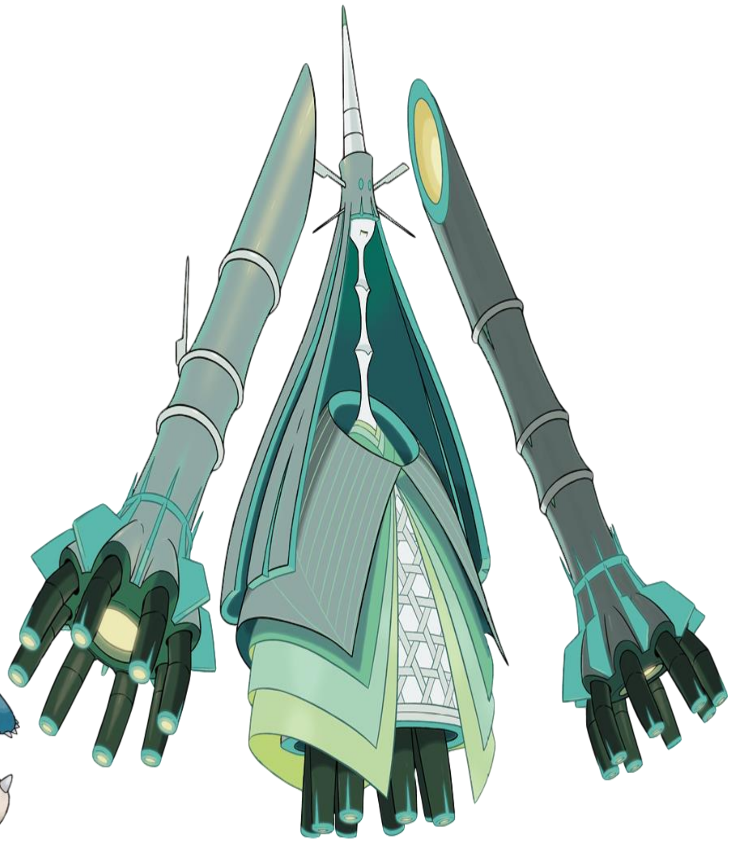
- Models are often over-parameterised
  - E.g. BERT, GPT-3 in NLP
  - E.g. ResNet-152 for vision tasks

Why Estimating Uncertainty in DL?

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

ResNet-152
(~60 million)

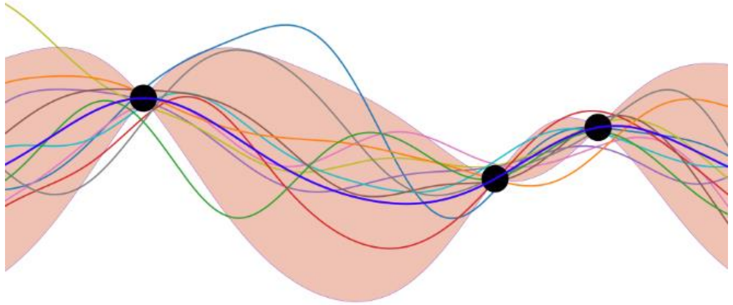
BERT
(~110 million)



GPT-3
(~175 billion)

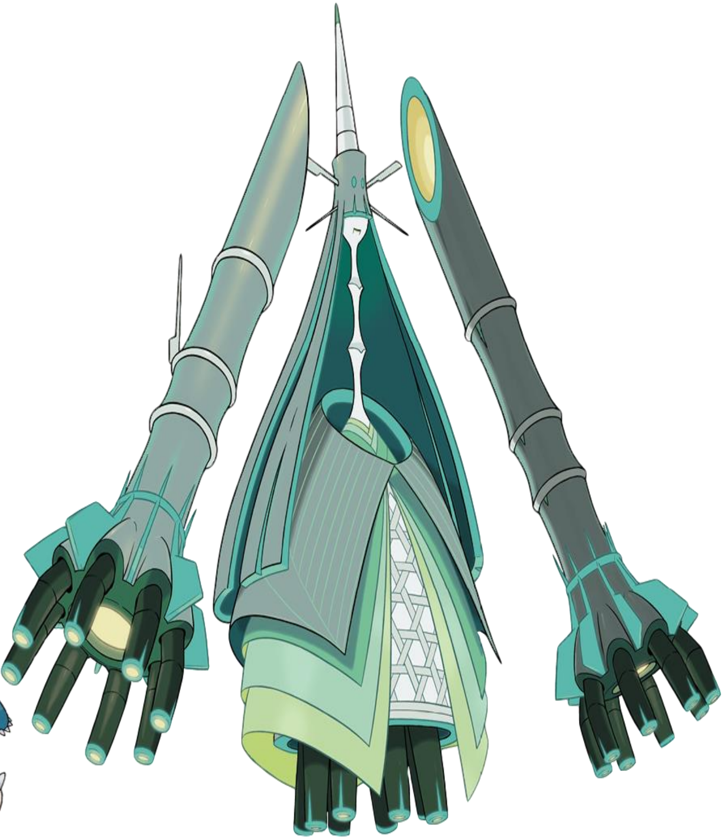
Why Estimating Uncertainty in DL?

- 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
(~110 million)



GPT-3
(~175 billion)

Why Estimating Uncertainty in DL?

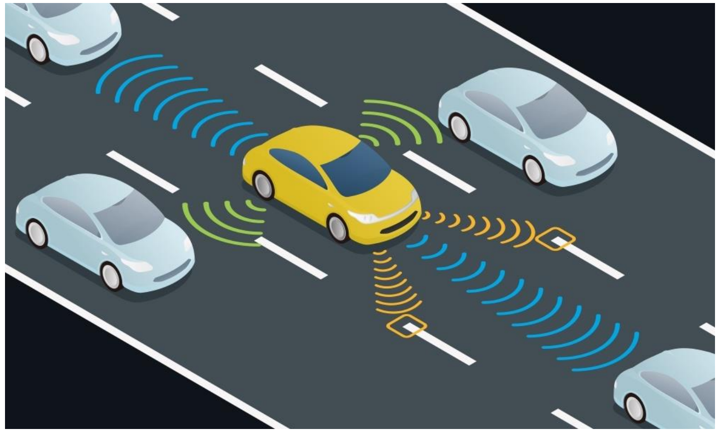
- Critical tasks need uncertainty estimates to assist decision making
  - Inform end users when uncertain, for safe decision making

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



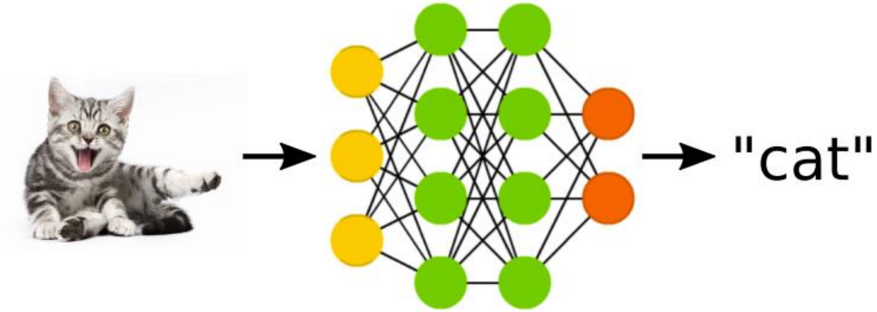
Autonomous driving

Bayesian Neural Network (BNN) 101

Classifying different types of animals:

- x: input image; y: output label
- Build a neural network with parameters θ:

$$p(y|x,\theta) = softmax(f_\theta(x))$$

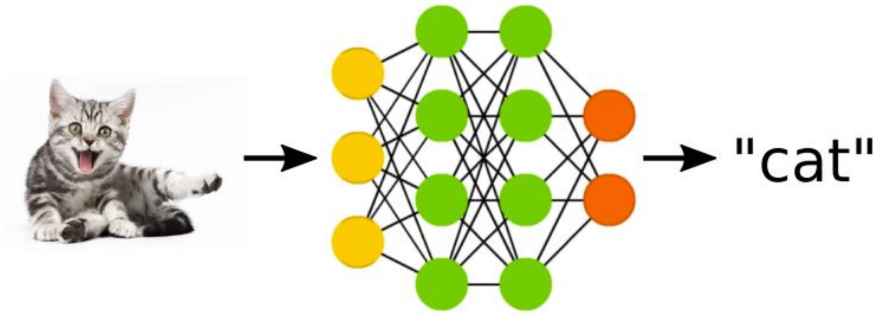


Bayesian Neural Network (BNN) 101

Classifying different types of animals:

- x: input image; y: output label
- Build a neural network with parameters θ:

$$p(y|x,\theta) = softmax(f_\theta(x))$$



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

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

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

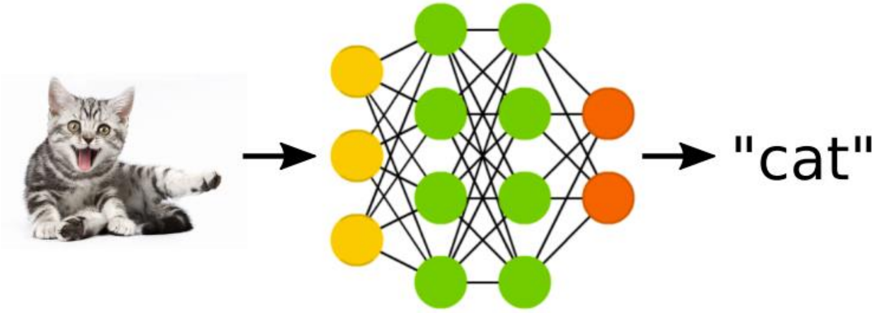
Neural network parameters: $\theta = \{W^l, b^l\}_{l=1}^L$

Bayesian Neural Network (BNN) 101

Classifying different types of animals:

- x: input image; y: output label
- Build a neural network with parameters θ:

$$p(y|x,\theta) = softmax(f_\theta(x))$$



Typical deep learning solution:

- Optimize θ to obtain a point estimates (MLE):

$$\theta^* = argmax \ \log p(D \mid \theta) \ ,$$
$$\log p(D \mid \theta) = \sum_{n=1}^{N} \log p(y_n \mid x_n, \theta) \ , D = \{(x_n, y_n)\}_{n=1}^{N}$$

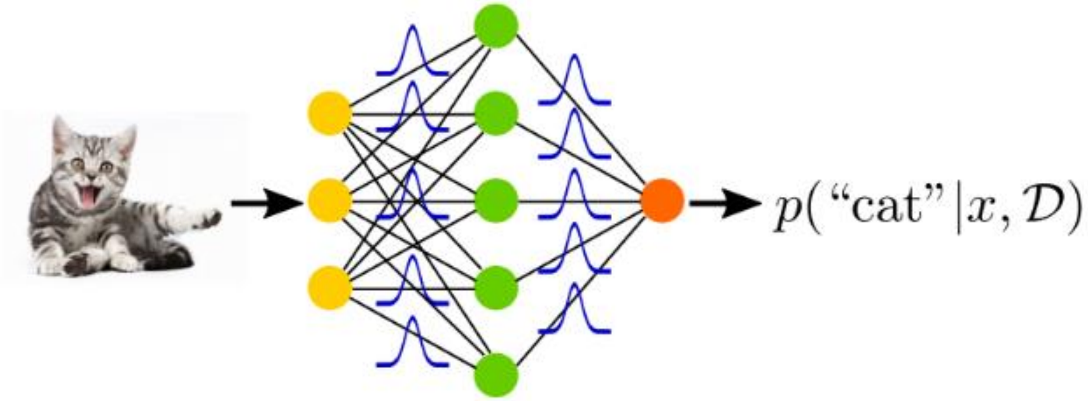
- Prediction: using $p(y^* \mid x^*, \theta^*)$

Bayesian Neural Network (BNN) 101

Classifying different types of animals:

- x: input image; y: output label
- Build a neural network with parameters θ:

$$p(y|x,\theta) = softmax(f_\theta(x))$$



Bayesian solution:

- Put a prior $p(\theta)$ on network parameters θ, e.g. Gaussian prior

$$p(\theta) = N(\theta; 0, \sigma^2\, I)$$

- Compute the posterior distribution $p(\theta \mid D)$:

$$p(\theta \mid D) \propto p(D \mid \theta)\, p(\theta)$$

- Bayesian predictive inference:

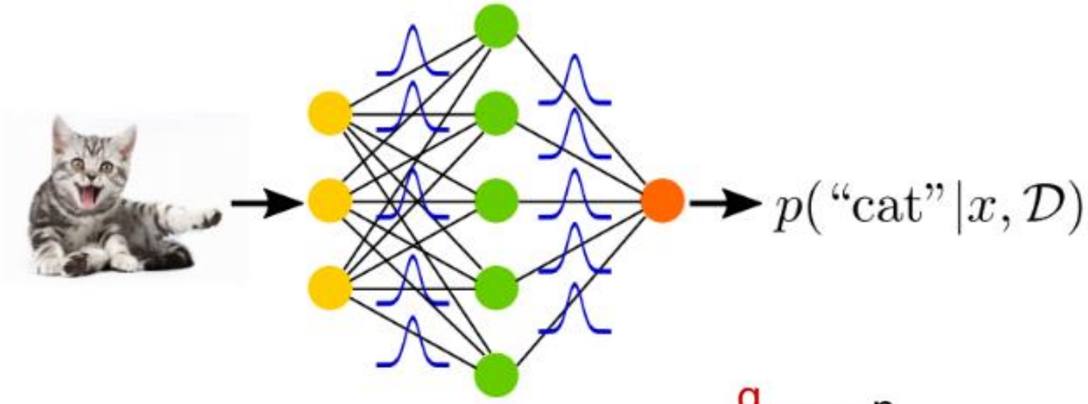
$$p(y^* \mid x^*, D) = E_{p(\theta \mid D)}[p(y^* \mid x^*, \theta)]$$

Bayesian Neural Network (BNN) 101

Classifying different types of animals:

- x: input image; y: output label
- Build a neural network with parameters θ:

$$p(y|x, \theta) = softmax(f_\theta(x))$$



Approximate (Bayesian) inference solution:

- Exact posterior intractable, use approximate posterior:

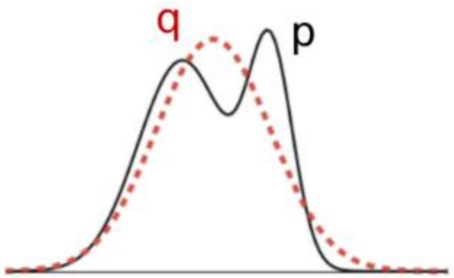
$$q(\theta) \approx p(\theta \mid D)$$

- Approximate Bayesian predictive inference:

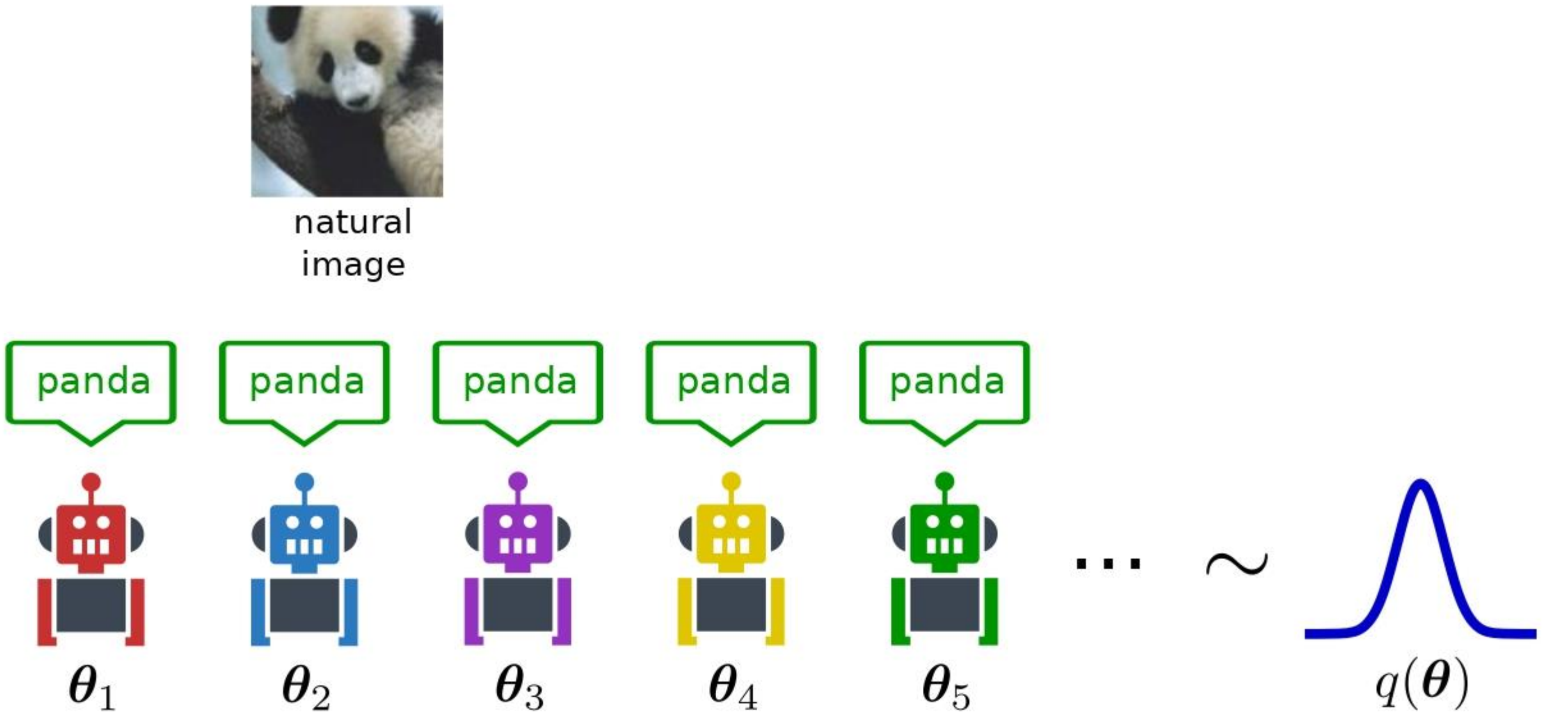
$$p(y^* \mid x^*, D) \approx E_{q(\theta)}[p(y^* \mid x^*, \theta)]$$

- Monte Carlo approximation:

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

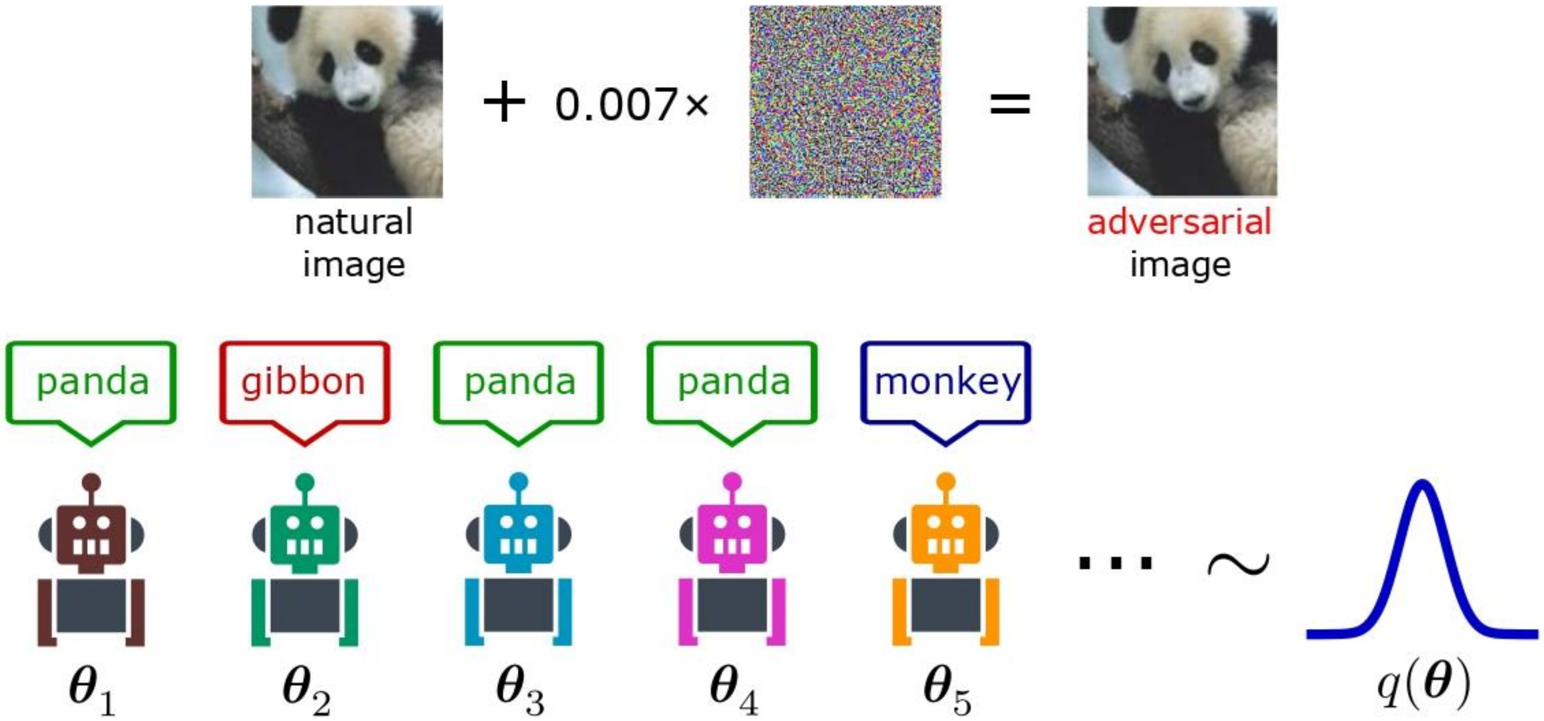


Bayesian Neural Network (BNN) 101



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

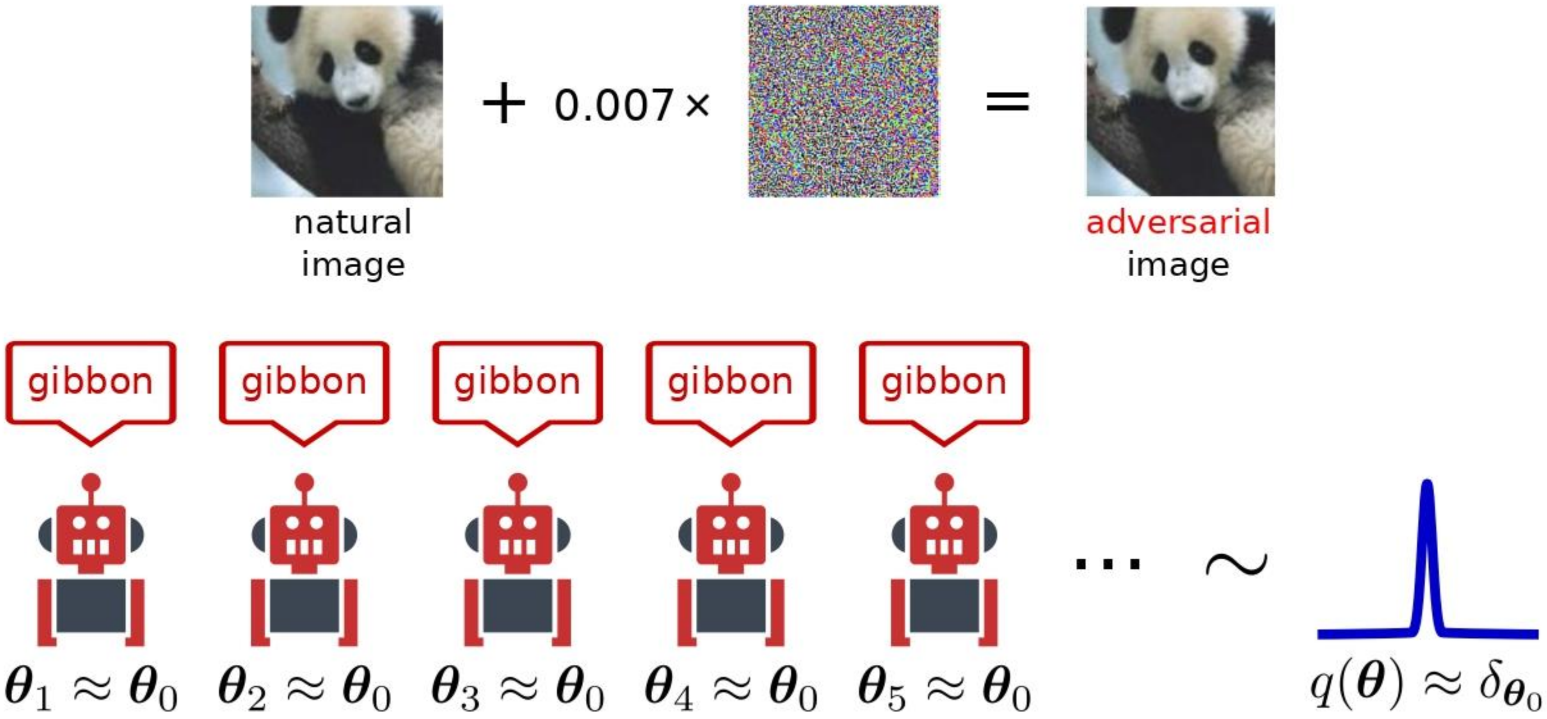
Bayesian Neural Network (BNN) 101



Prediction on OOD/noisy/adversarial data:

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

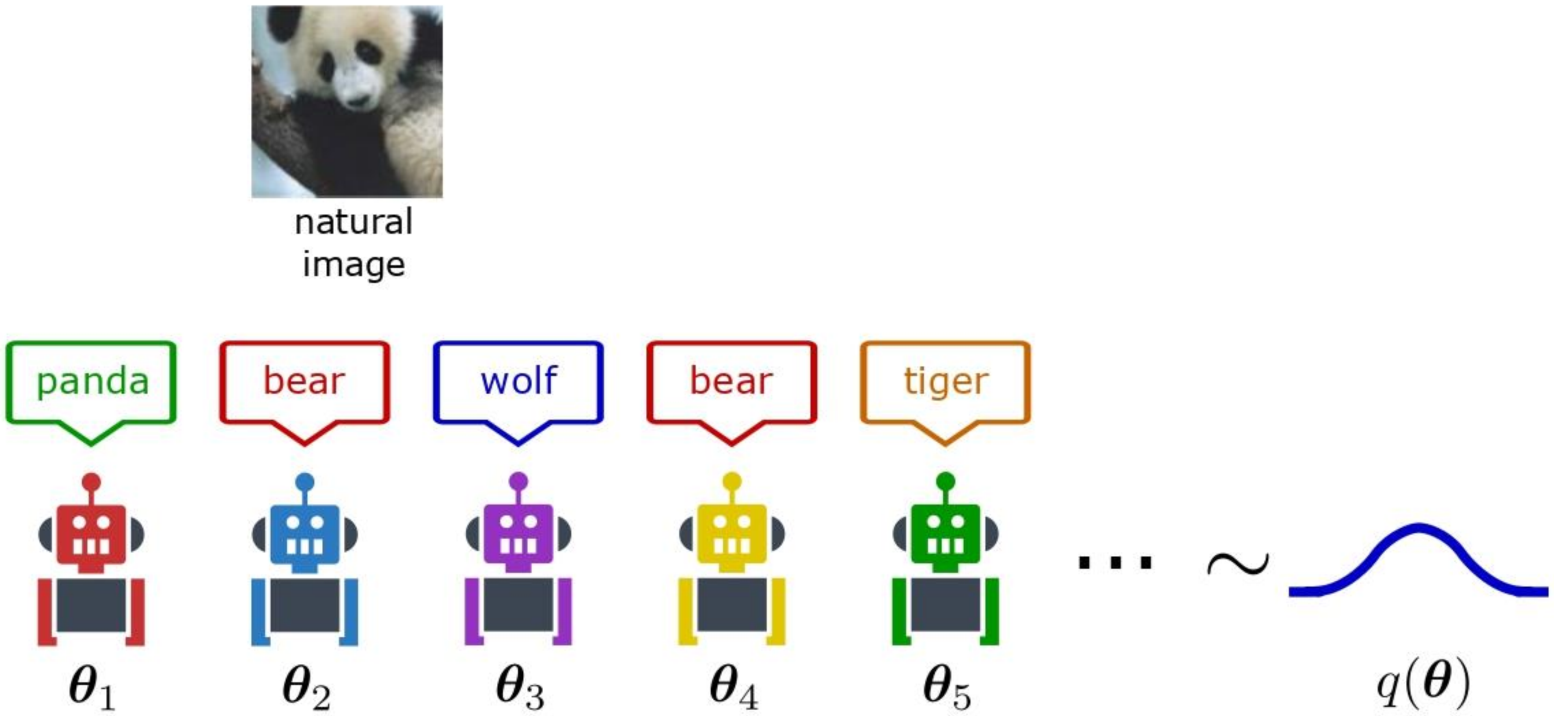
Bayesian Neural Network (BNN) 101



natural image
+ 0.007 ×
=
adversarial image
gibbon
gibbon
gibbon
gibbon
gibbon
$\boldsymbol{\theta}_1 \approx \boldsymbol{\theta}_0$
$\boldsymbol{\theta}_2 \approx \boldsymbol{\theta}_0$
$\boldsymbol{\theta}_3 \approx \boldsymbol{\theta}_0$
$\boldsymbol{\theta}_4 \approx \boldsymbol{\theta}_0$
$\boldsymbol{\theta}_5 \approx \boldsymbol{\theta}_0$
$\cdots \sim$
$q(\boldsymbol{\theta}) \approx \delta_{\boldsymbol{\theta}_0}$

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

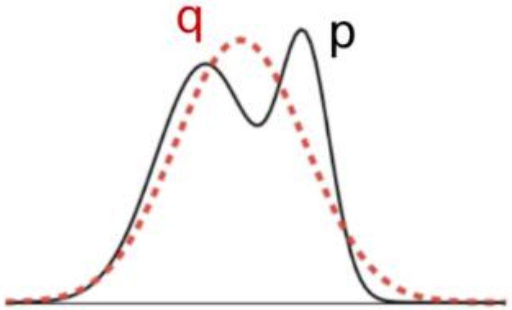
Bayesian Neural Network (BNN) 101



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

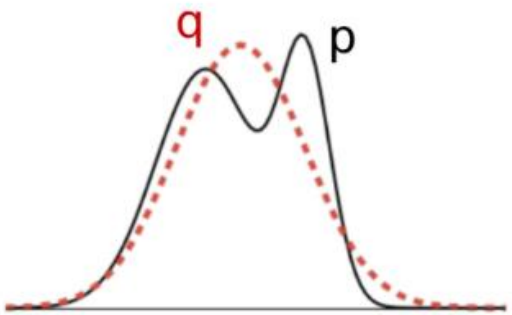
Approximate Inference in BNNs

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



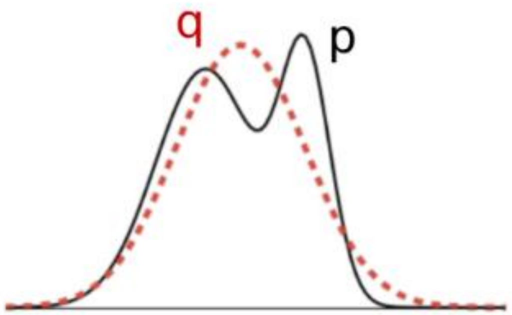
Approximate Inference in BNNs

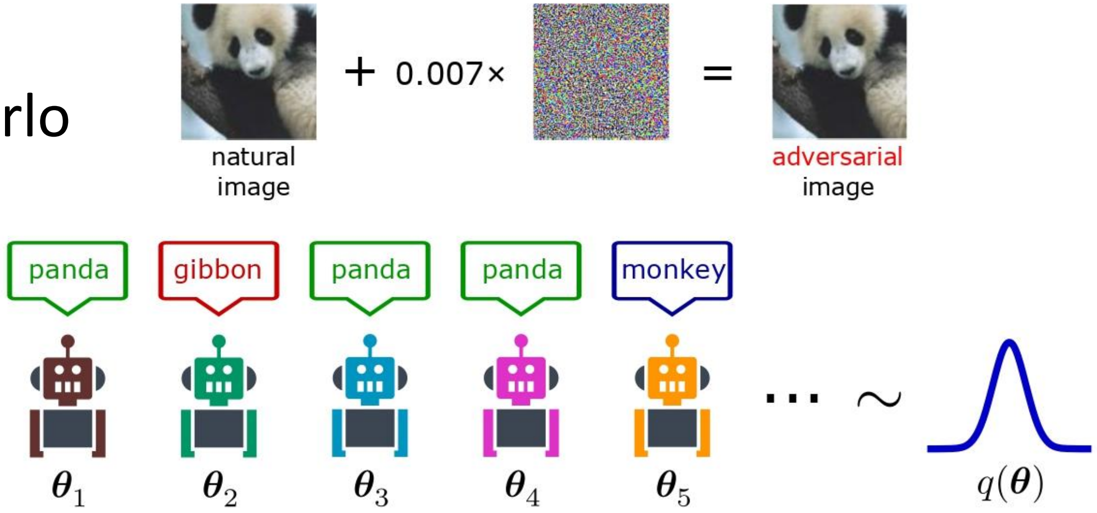
- Key steps of approximate inference in BNNs
  1. Construct the $q(\theta) \approx p(\theta \mid D)$ distribution
     - Simple distributions: e.g. Mean-field Gaussian
     - Structured approximations, e.g. low-rank Gaussians
     - Others (non-Gaussian)
  2. Fit the $q(\theta)$ distribution
     - E.g.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



Approximate Inference in BNNs

- Key steps of approximate inference in BNNs
  1. Construct the $q(\theta) \approx p(\theta \mid D)$ 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





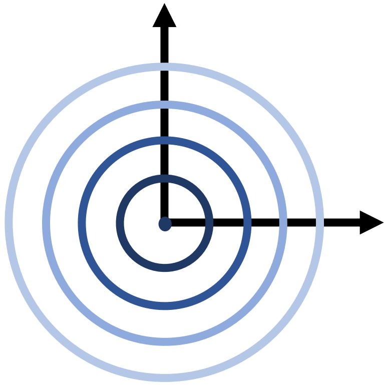
Approximate Inference in BNNs

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

$$q(\theta) = \prod_{l=1}^{L} q\left(W^{l}\right) q(b^{l})$$

$$q(W_l) = \prod_{ij} q(W_{ij}^{l}), \qquad q\left(W_{ij}^{l}\right) = N(W_{ij}^{l}; M_{ij}^{l}, V_{ij}^{l})$$

$$q\left(b^{l}\right) = \prod_i q\left(b_i^{l}\right),\ \ q\left(b_i^{l}\right) = N(b_i^{l}; m_i^{l}, v_i^{l})$$



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

Blundell et al. Weight Uncertainty in Neural Networks. ICML 2015

Approximate Inference in BNNs

- Step 2: fit the $q(\theta)$ distribution:
  - Variational inference: $\phi^* = argmax\ L(\phi)$

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

Blundell et al. Weight Uncertainty in Neural Networks. ICML 2015

Approximate Inference in BNNs

- Step 2: fit the $q(\theta)$ distribution:
  - Variational inference: $\phi^* = argmax\ L(\phi)$
    $$L(\phi) = E_{q(\theta)}[\log p(D \mid \theta)] - KL[q(\theta) \,||\, p(\theta)]$$
  - First scalable technique: Stochastic optimization
    - i.i.d. assumption of data: $\log p(D \mid \theta) = \sum_{n=1}^{N} \log p(y_n \mid x_n, \theta)$
    - Enable mini-batch training with $\{(x_m, y_m)\} \sim D^M$:

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

Blundell et al. Weight Uncertainty in Neural Networks. ICML 2015

Approximate Inference in BNNs

- Step 2: fit the $q(\theta)$ distribution:
  - Variational inference: $\phi^* = argmax\ L(\phi)$

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

  - First scalable technique: Stochastic optimization
    - i.i.d. assumption of data: $\log p(D \mid \theta) = \sum_{n=1}^{N} \log p(y_n \mid x_n, \theta)$
    - Enable mini-batch training with $\{(x_m, y_m)\} \sim D^M$:

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

reweighting to ensure calibrated posterior concentration

Blundell et al. Weight Uncertainty in Neural Networks. ICML 2015

Approximate Inference in BNNs

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

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

Blundell et al. Weight Uncertainty in Neural Networks. ICML 2015

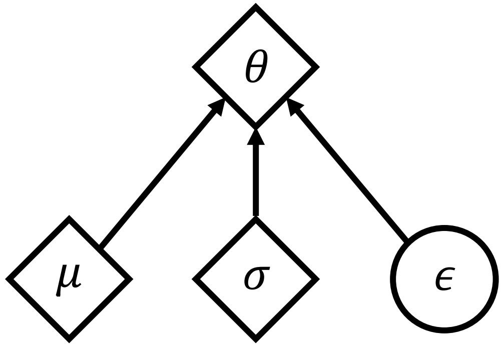
Approximate Inference in BNNs

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

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

    - Reparameterization trick to sample mean-field Gaussians:

$$\theta_k \sim q(\theta) \Leftrightarrow \theta_k = m_\theta + \sigma_\theta\,\epsilon_k,\ \ \epsilon_k \sim N(0,I)$$

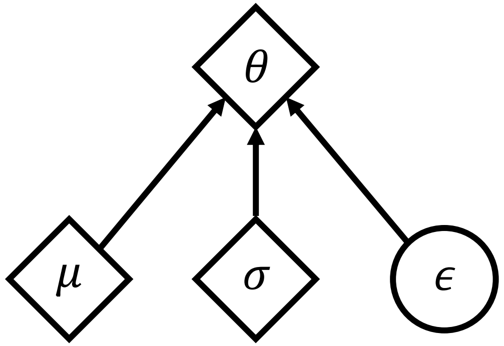


Blundell et al. Weight Uncertainty in Neural Networks. ICML 2015

Approximate Inference in BNNs

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

$$E_{q(\theta)}[\log p(y \mid x, \theta)] \approx \frac{1}{K} \sum_{k}^{K} \log p(y \mid x, \theta_k)\,, \qquad \theta_k \sim q(\theta)$$



- Reparameterization trick to sample mean-field Gaussians:

$$\theta_k \sim q(\theta) \Leftrightarrow \theta_k = m_\theta + \sigma_\theta\, \epsilon_k,\ \ \epsilon_k \sim N(0, I)$$

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

Blundell et al. Weight Uncertainty in Neural Networks. ICML 2015

Approximate Inference in BNNs

- Combining both steps:

$$L(\phi) \approx \frac{N}{M}\sum_{m=1}^{M}\frac{1}{K}\sum_{k=1}^{K}\log p(y_m \mid x_m, \theta_k) - \underbrace{KL[q(\theta) \,||\, p(\theta)]}_{\text{analytic between two Gaussians}}, \theta_k \sim q(\theta)$$

(if not, can also be estimated with Monte Carlo)

Blundell et al. Weight Uncertainty in Neural Networks. ICML 2015

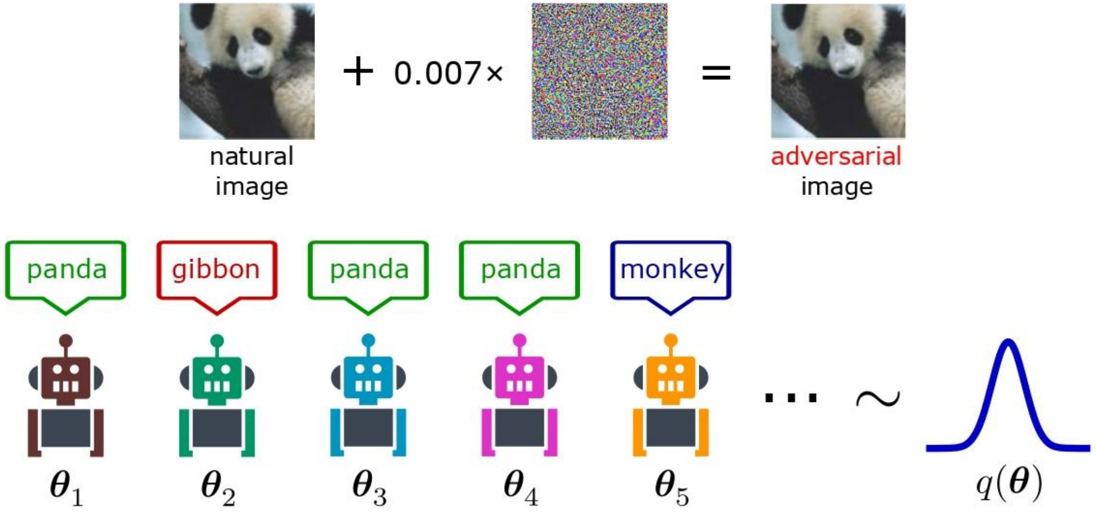
Approximate Inference in BNNs

- Step 3: compute prediction with Monte Carlo approximations:

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

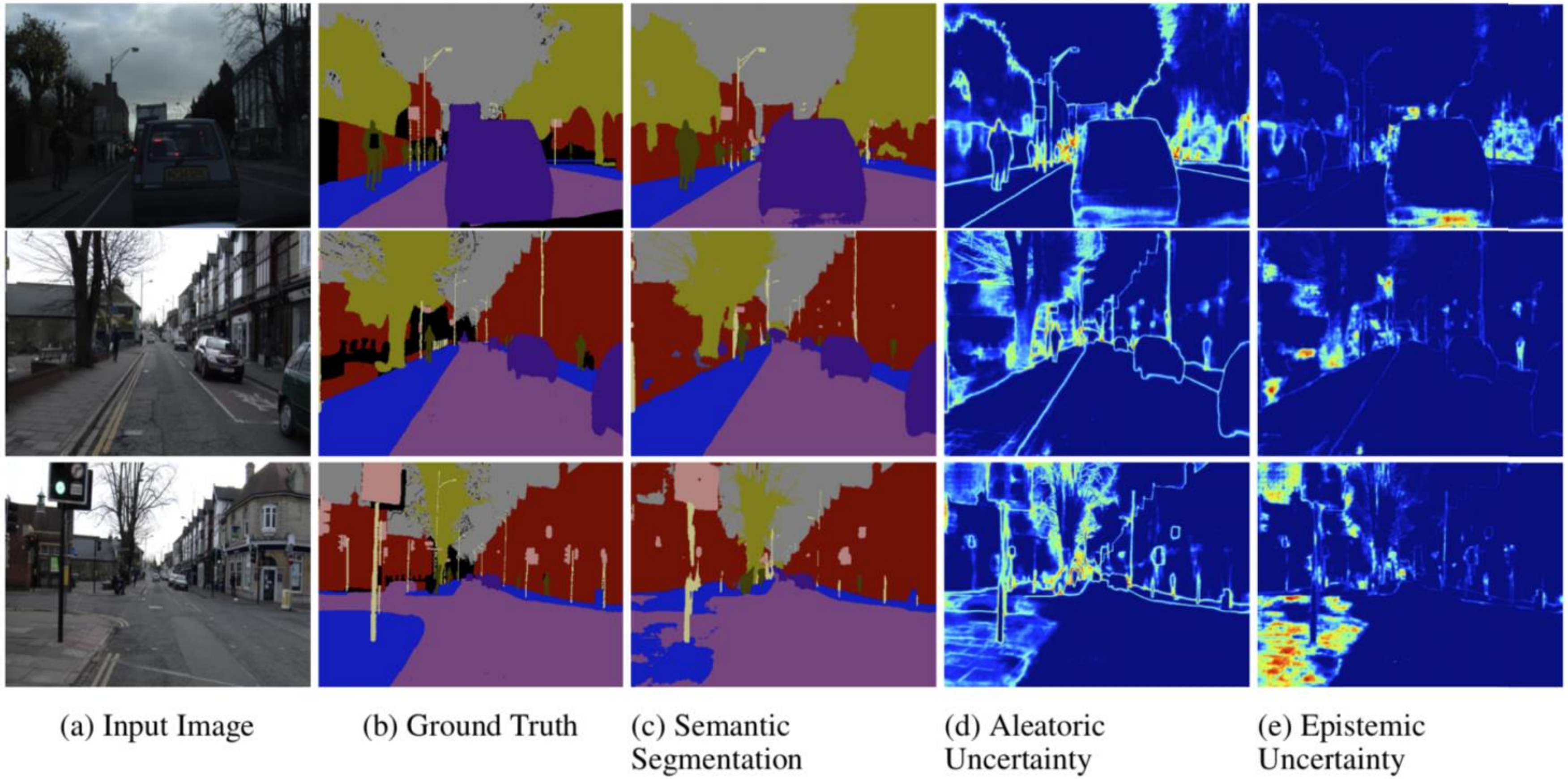
Mean-field Gaussian case:

$\theta_k = m_\theta + \sigma_\theta\, \epsilon_k,\ \ \epsilon_k \sim N(0, I)$



Blundell et al. Weight Uncertainty in Neural Networks. ICML 2015

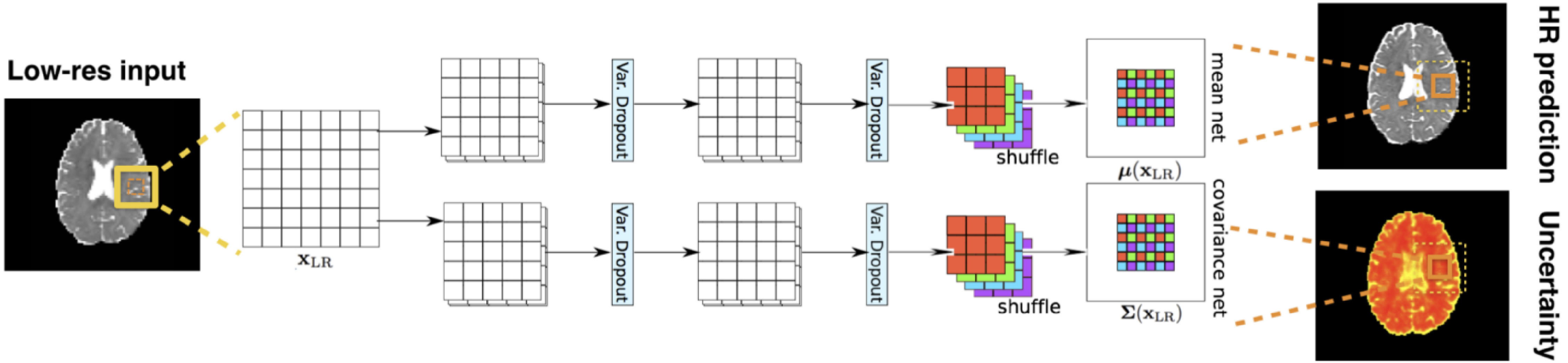
Applications of BNNs: Image Segmentation

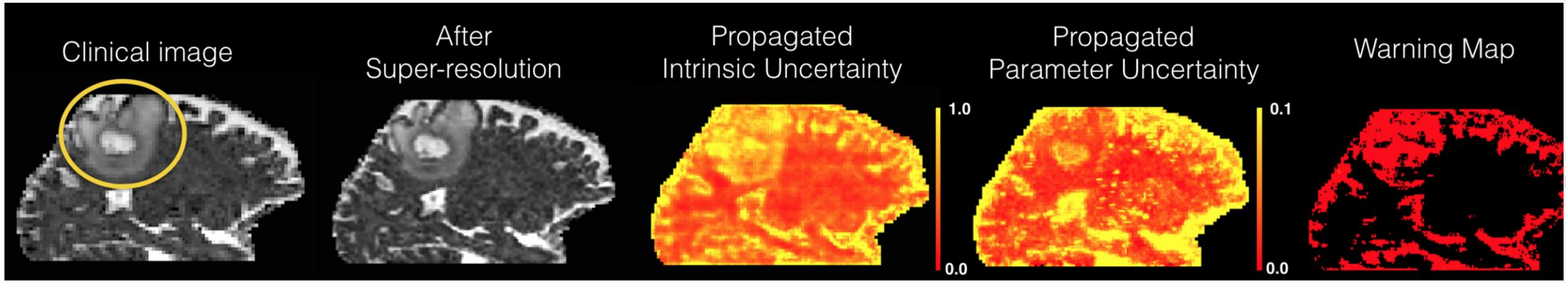


(a) Input Image (b) Ground Truth (c) Semantic Segmentation (d) Aleatoric Uncertainty (e) Epistemic Uncertainty

Kendall and Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? NeurIPS 2017

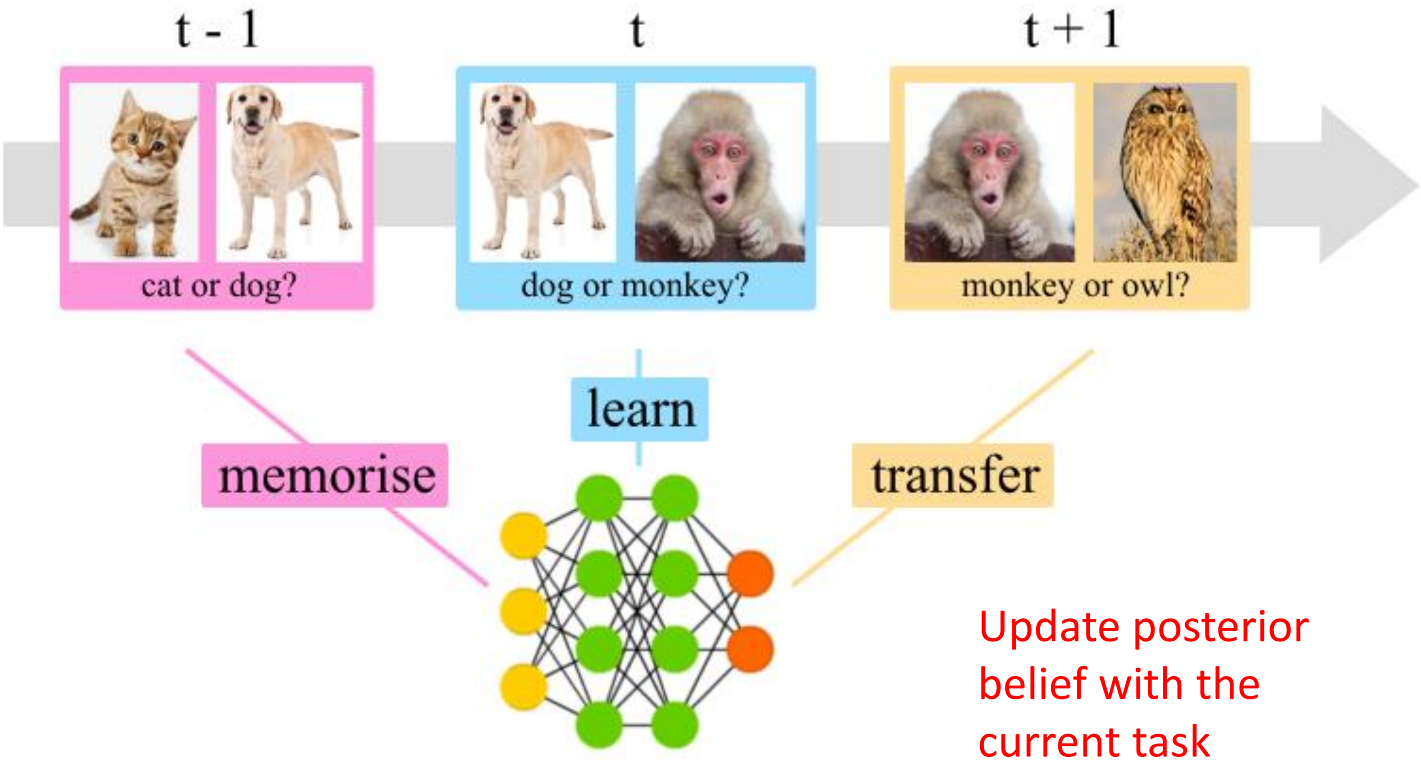
Applications of BNNs: Super Resolution




Clinical image
After Super-resolution
Propagated Intrinsic Uncertainty
1.0
0.0
Propagated Parameter Uncertainty
0.1
0.0
Warning Map

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

Applications of BNNs: Continual Learning



Update posterior belief with the current task

Posterior from the previous tasks as prior for the current task

$$L_{VCL}^{t}\left(q_t(\theta)\right) = E_{q_{t(\theta)}}[\log p(D_t \mid \theta)] - KL[\boxed{q_t(\theta)} \,||\, \boxed{q_{t-1}(\theta)}]$$

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

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

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

Stochastic gradient MCMC

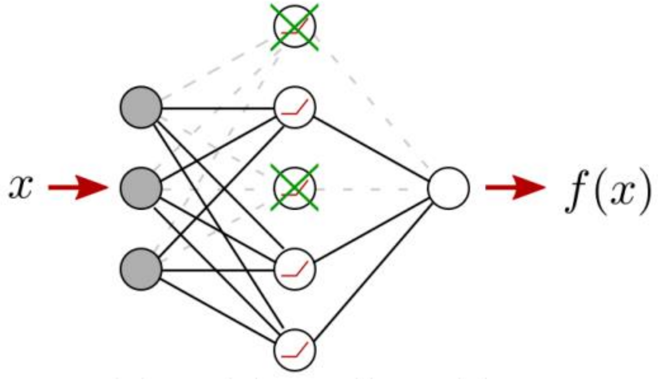
Li et al. Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks. AAAI 2016
Zhang et al. Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning. ICLR 2020

Recent Progress in BNNs: Inference

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

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

Stochastic gradient MCMC



Monte Carlo dropout

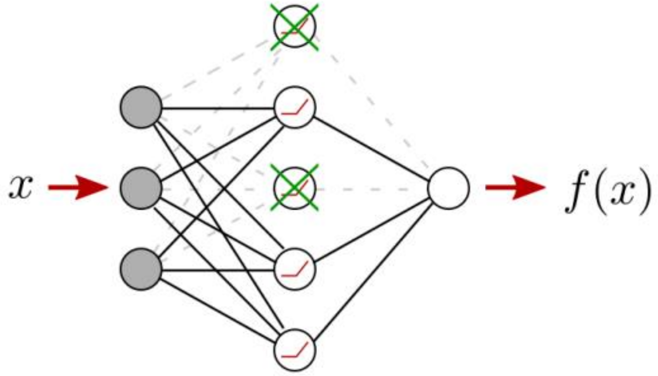
Gal and Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. ICML 2016

Recent Progress in BNNs: Inference

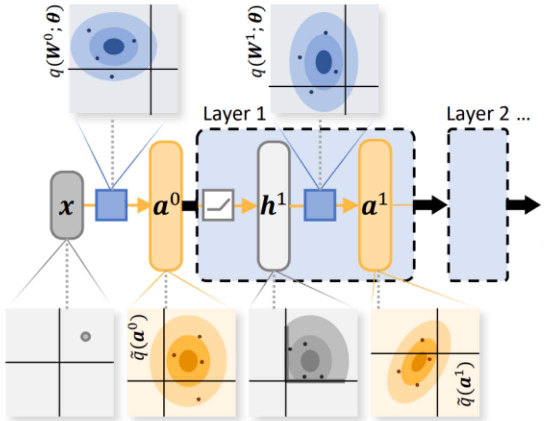
SGD: $\theta_{t+1} = \theta_t - \eta \nabla_{\theta_t} \widetilde{U}(\theta_t)$

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

Stochastic gradient MCMC



Monte Carlo dropout



Deterministic approximations

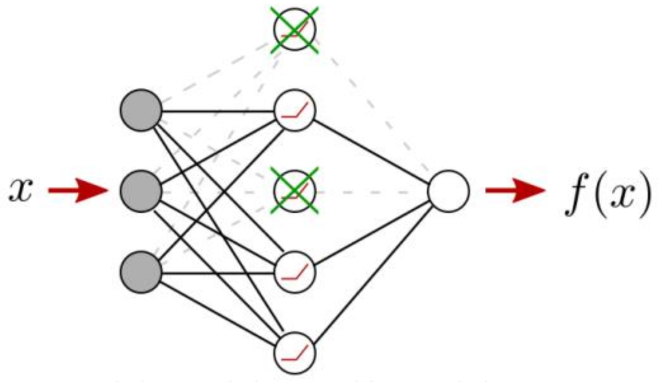
Hernandez-Lobato and Adams. Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. ICML 2015
Wu et al. Deterministic Variational Inference for Robust Bayesian Neural Networks. ICLR 2019

Recent Progress in BNNs: Inference

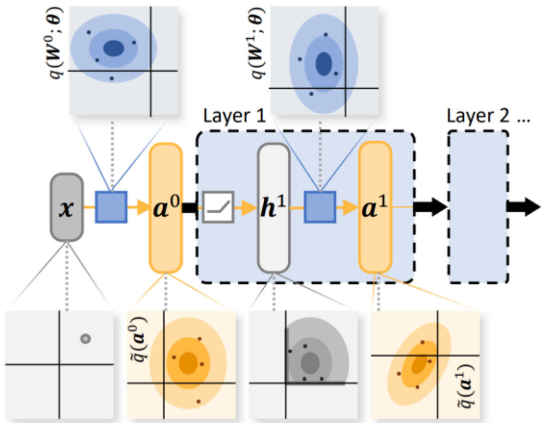
SGD: $\theta_{t+1} = \theta_t \ - \eta\nabla_{\theta_t}\widetilde{U}(\theta_t)$

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

Stochastic gradient MCMC

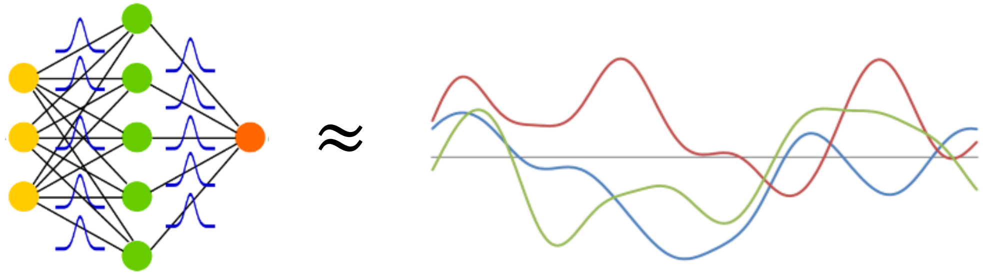


Monte Carlo dropout



$q(\boldsymbol{W}^0;\boldsymbol{\theta})$
$q(\boldsymbol{W}^1;\boldsymbol{\theta})$
Layer 1
Layer 2 ...
$\boldsymbol{x}$
$\boldsymbol{a}^0$
$\boldsymbol{h}^1$
$\boldsymbol{a}^1$
$\tilde{q}(\boldsymbol{a}^0)$
$\tilde{q}(\boldsymbol{a}^1)$

Deterministic approximations

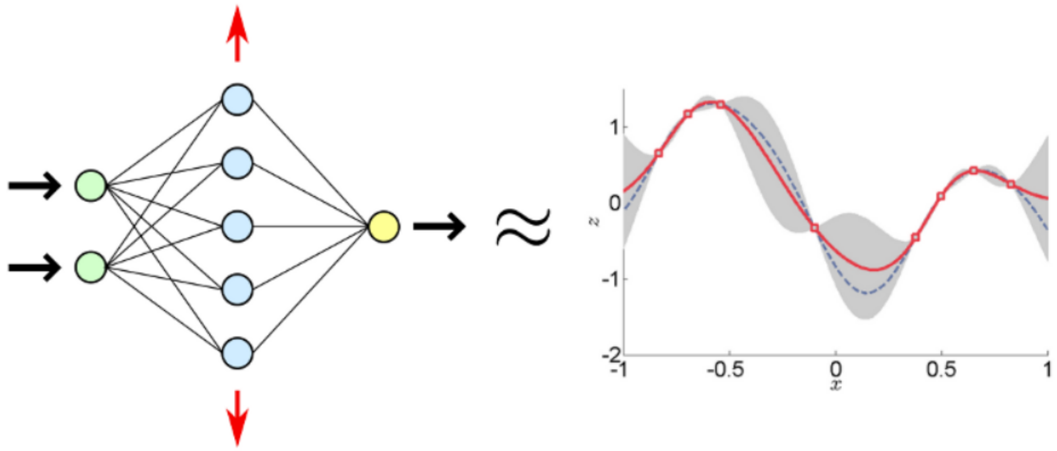


Function space approximate inference

Ma et al. Variational Implicit Processes. ICML 2019

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

Recent Progress in BNNs: Theory

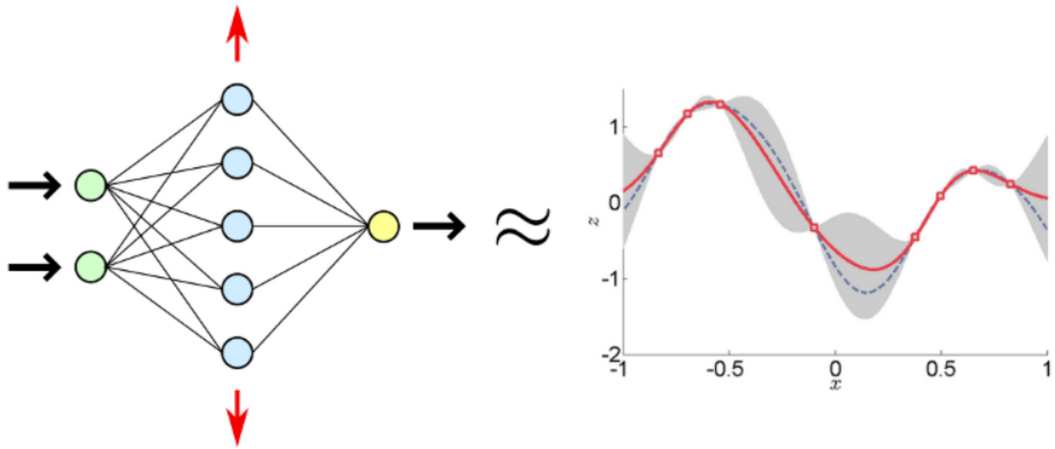


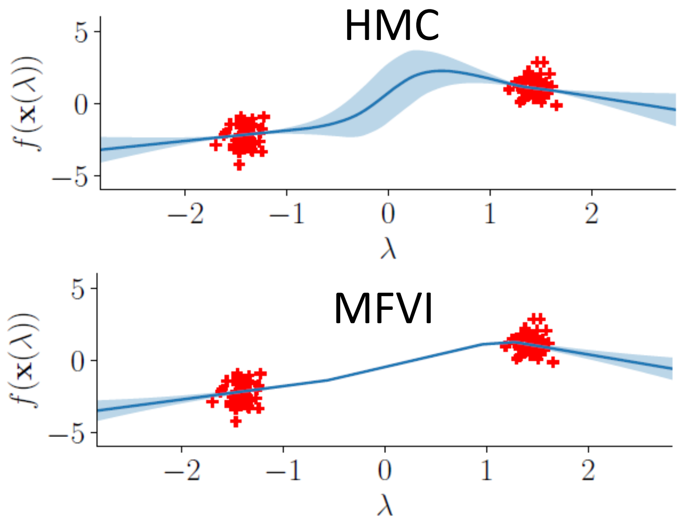
Connections to GPs:

- 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

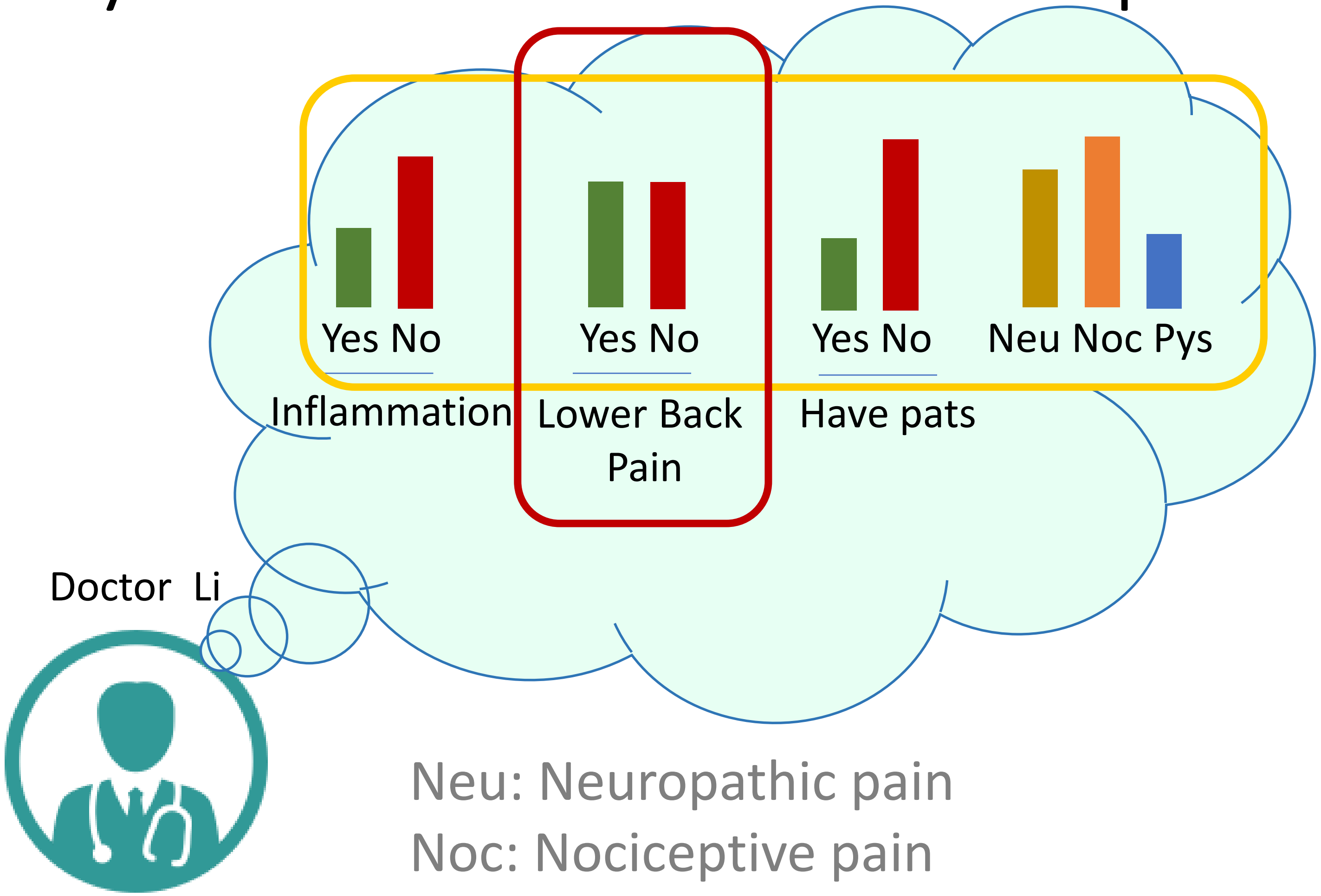
Foong et al. On the Expressiveness of Approximate Inference in Bayesian Neural Networks. NeurIPS 2020

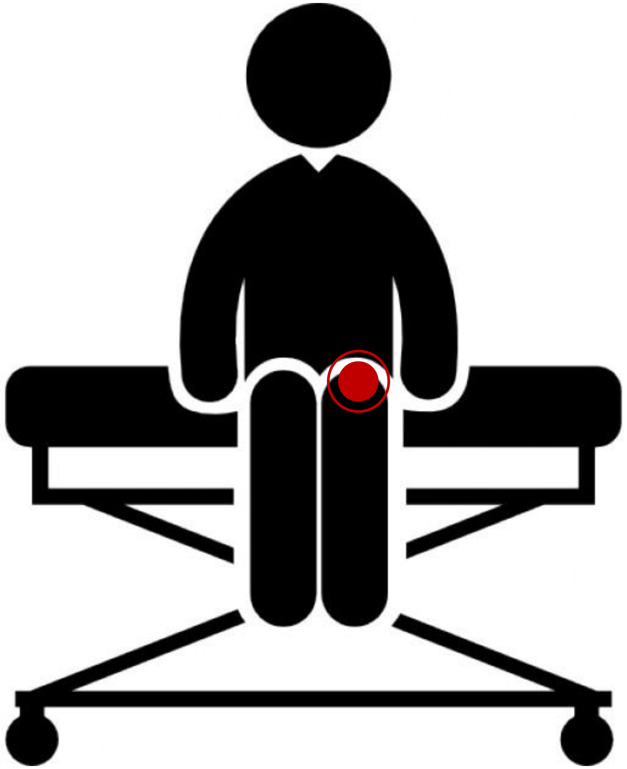
Farquhar et al. Liberty or Depth: Deep Bayesian Neural Nets Do Not Need Complex Weight Posterior Approximations. NeurIPS 2020

Dynamic Information Acquisition

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

Dynamic Information Acquisition





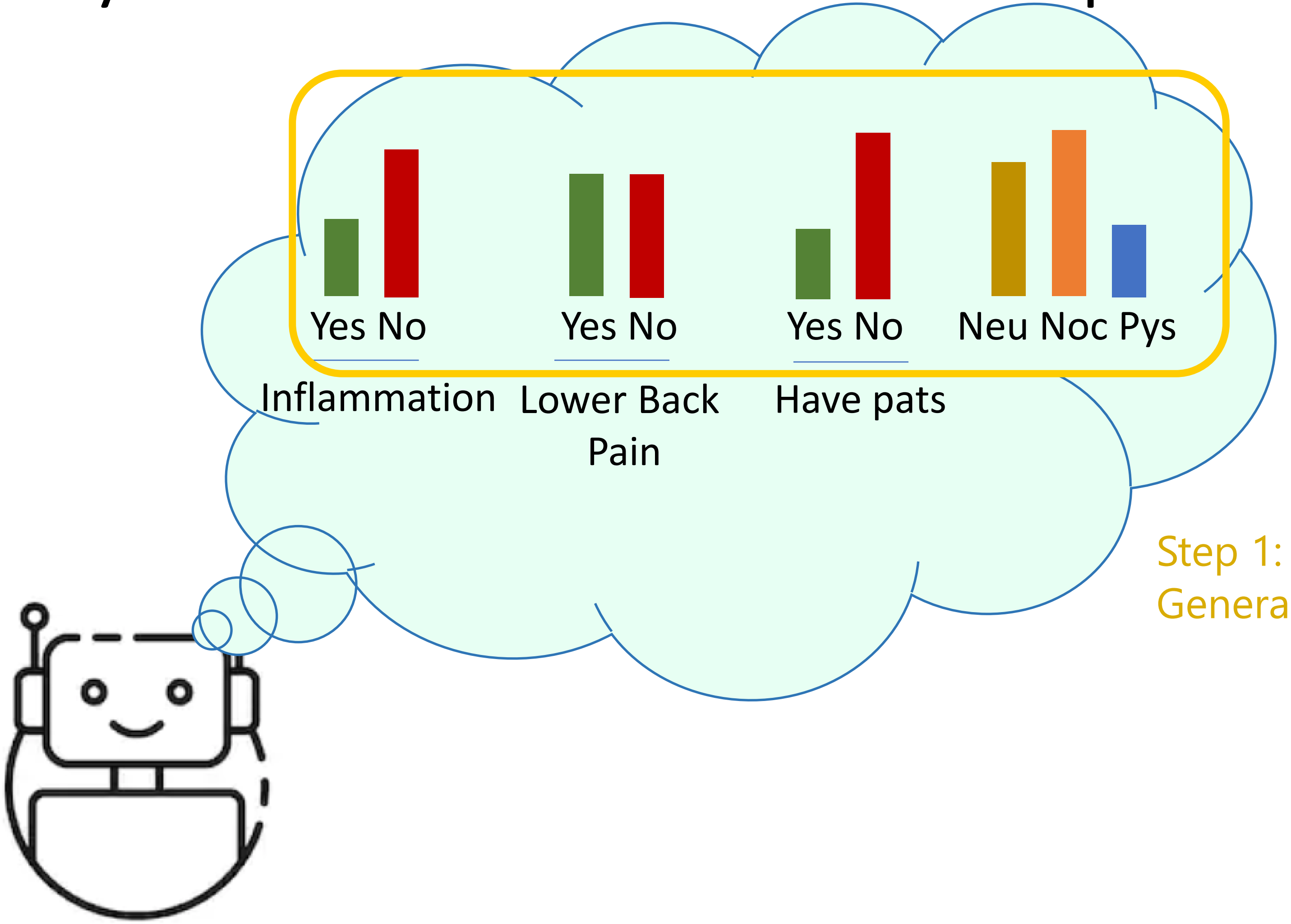
Neu: Neuropathic pain

Noc: Nociceptive pain

Pys: Psychiatric pain

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

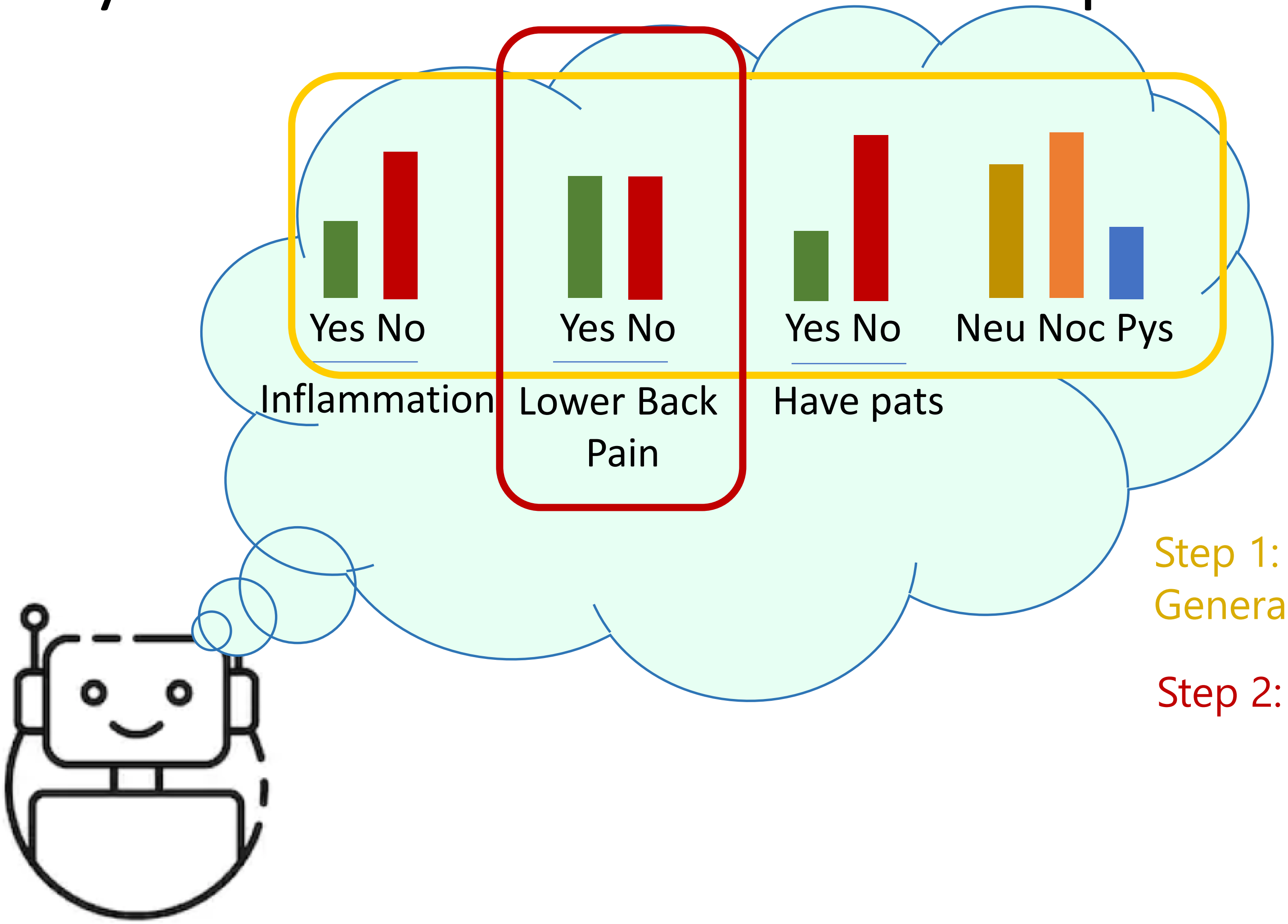
Dynamic Information Acquisition



Step 1: Missing Value Prediction with Deep Generative Model

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

Dynamic Information Acquisition

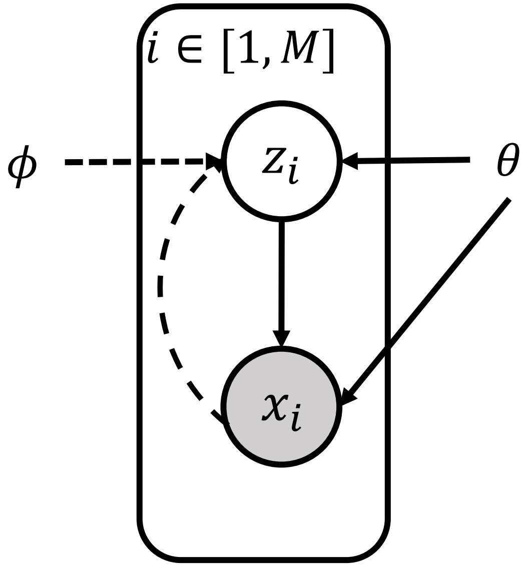


Step 1: Missing Value Prediction with Deep Generative Model

Step 2: Active element-wise information acquisition

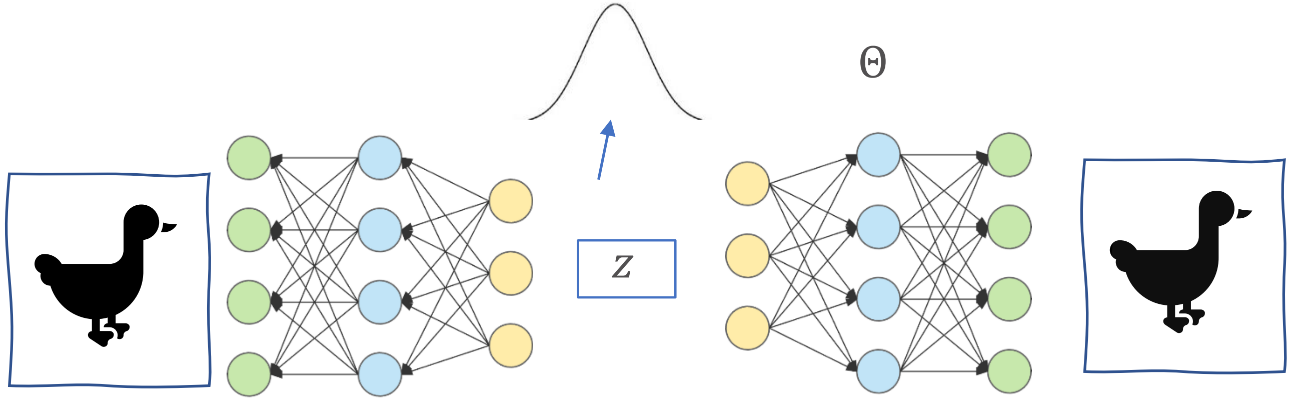
Ma et al. Eddi: Efficient dynamic discovery of high-value information with partial VAE. ICML 2019

A Deep Generative Model



Variational Auto-encoder (VAE)

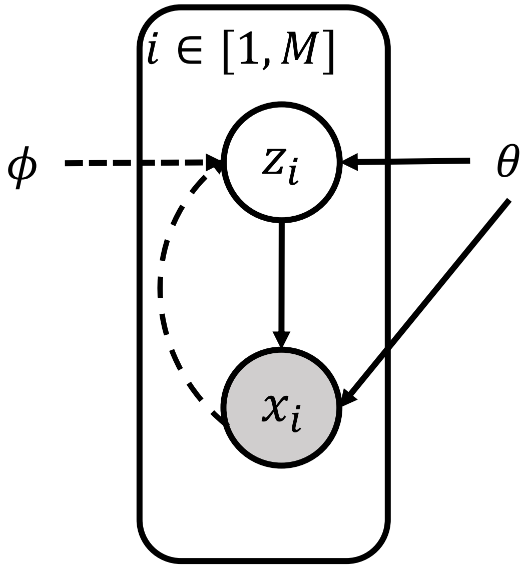
$$L_{amortized} = \log p(\boldsymbol{x}) - KL\,(q(\boldsymbol{z}\,|\boldsymbol{x}\,)|p(\boldsymbol{z}|\boldsymbol{x}))$$



Encoder/
inference network

Decoder/
generator

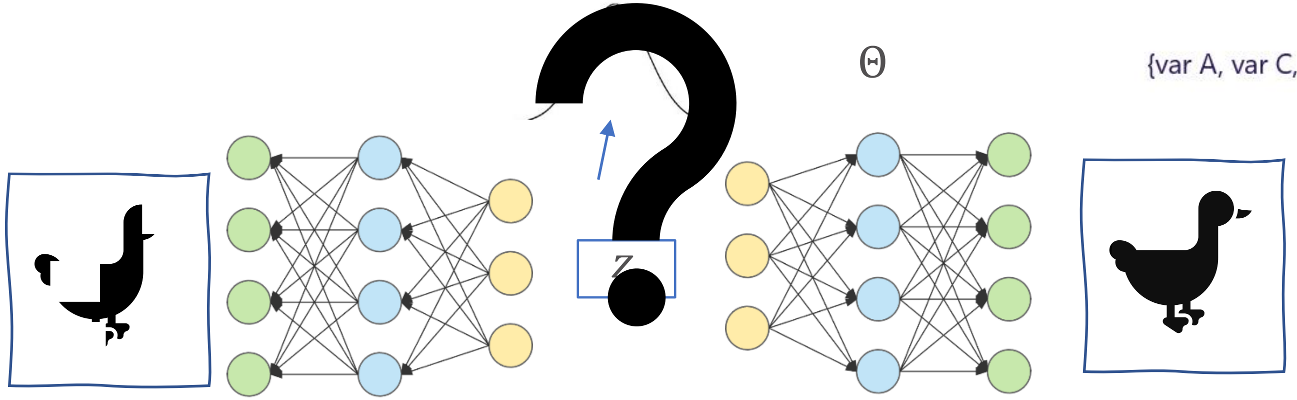
Challenges



- Utilizing Deep Learning for scalable inference
- Cannot handle partial observation

{var A, var C, var D, var E}

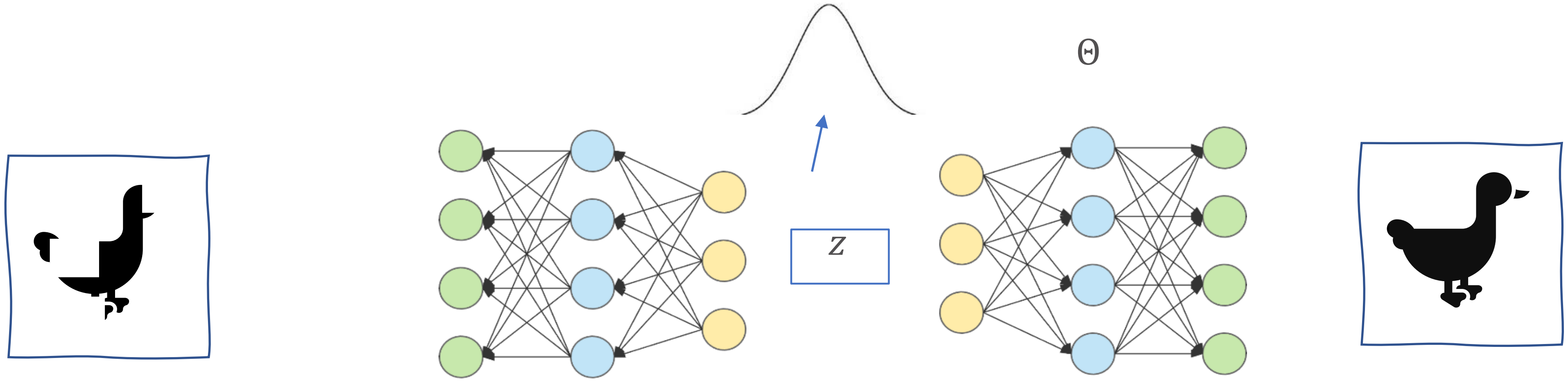
{var A, var B}



Encoder/
inference network

Decoder/
generator

VAE to Partial VAE

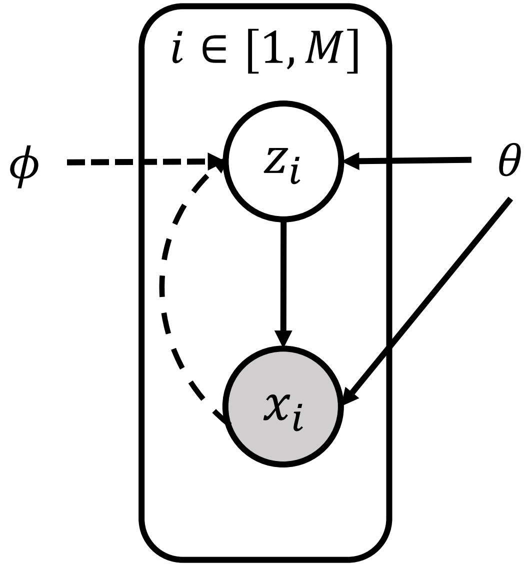


Encoder/
inference network

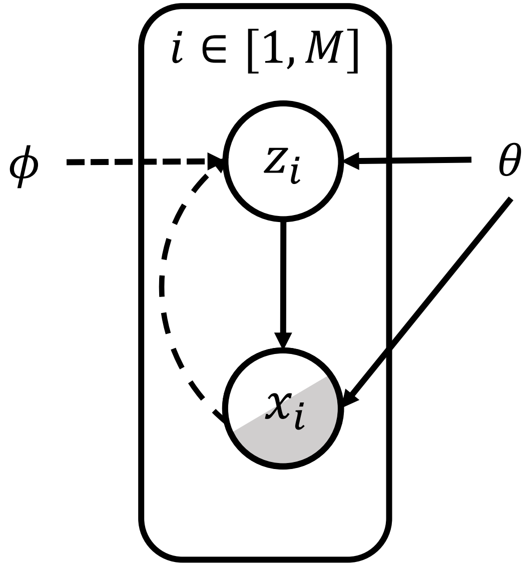
Decoder/
generator

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

VAE to Partial VAE



X is fully observed.

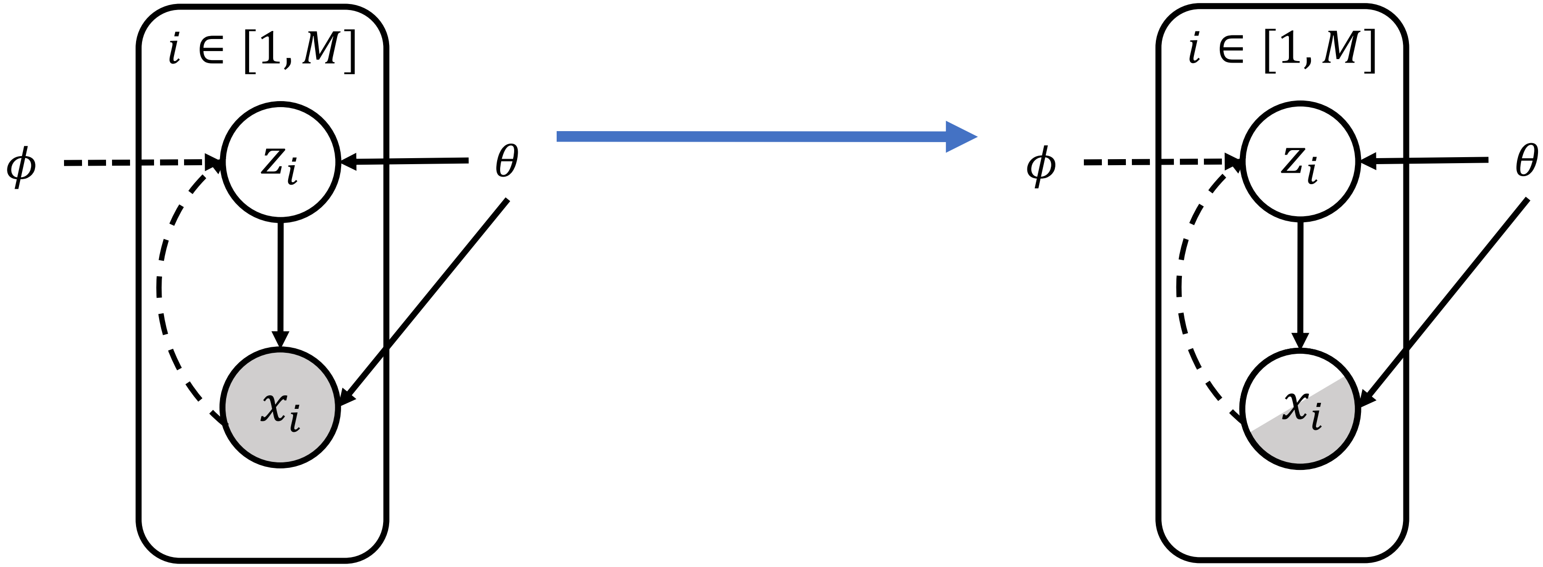


X is partially observed.

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

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

VAE to Partial VAE



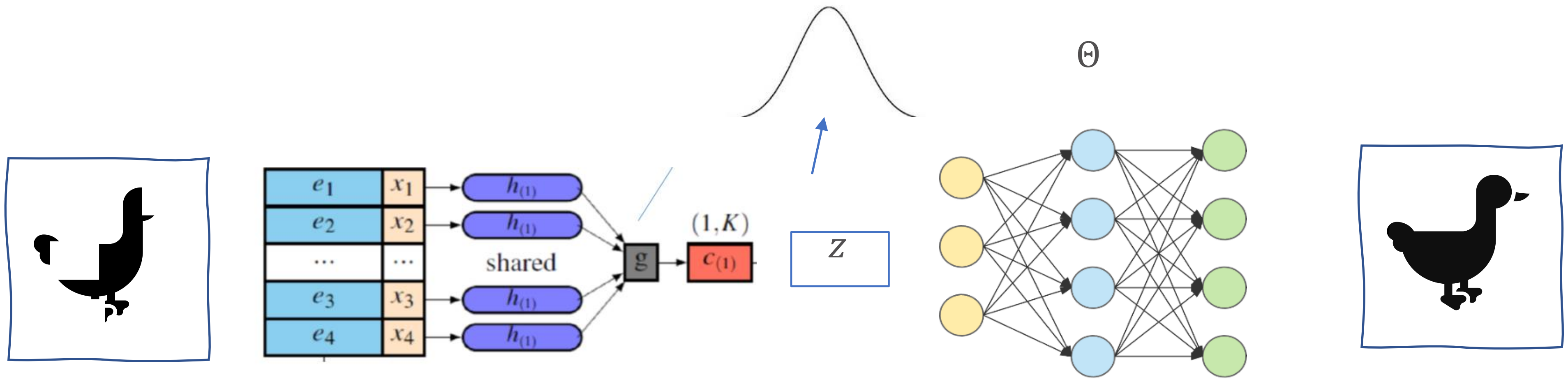
$$L_{amortized} = \log p(\boldsymbol{x}) - KL[q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}|\boldsymbol{x})]$$
$$= E_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})\ ] \ - KL[q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})]$$

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

The ELBO still holds.
The challenge is how to design an inference net.

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

VAE to Partial VAE



Encoder/
inference network

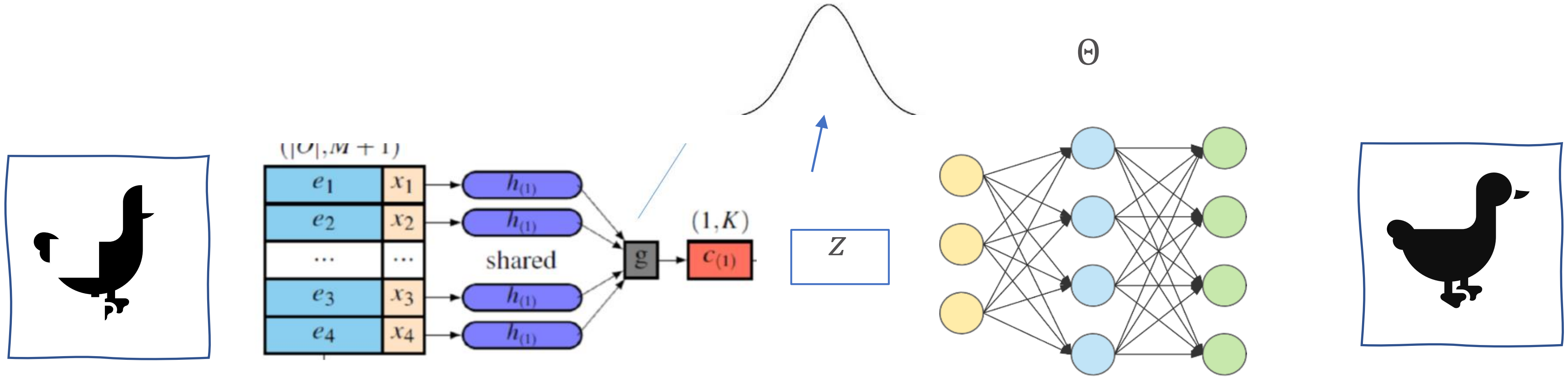
Decoder/
generator

$$\mathbf{c}(\mathbf{x}_O) := g(\vec{h}(\mathbf{s}_1), \vec{h}(\mathbf{s}_2), ..., \vec{h}(\mathbf{s}_O))$$

Set Encoder

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

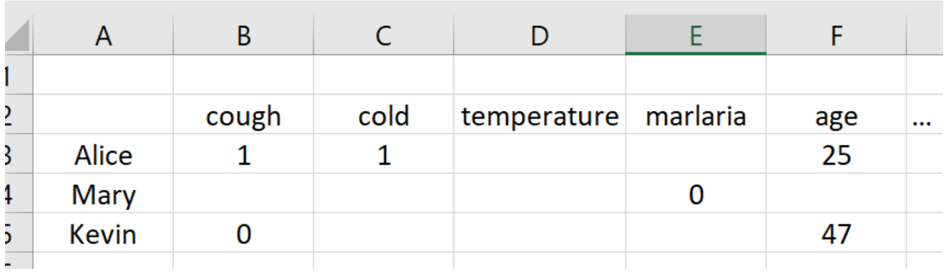


Encoder/
inference network

Decoder/
generator

$$\mathbf{c}(\mathbf{x}_O) := g(\vec{h}(\mathbf{s}_1), \vec{h}(\mathbf{s}_2), ..., \vec{h}(\mathbf{s}_O))$$

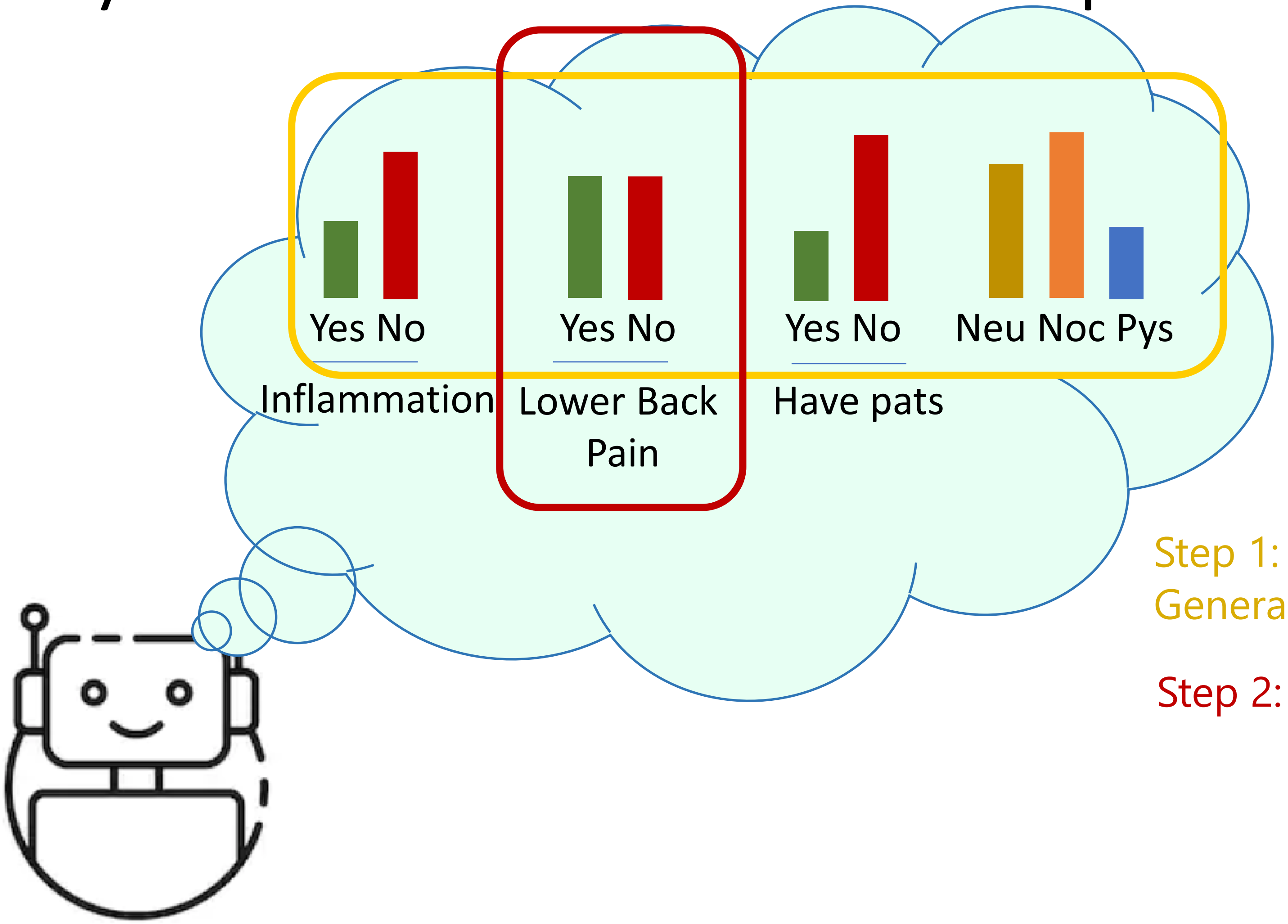
Set Encoder



	A	B	C	D	E	F	
1							
2		cough	cold	temperature	marlaria	age	...
3	Alice	1	1			25	
4	Mary				0		
5	Kevin	0				47	

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

Dynamic Information Acquisition

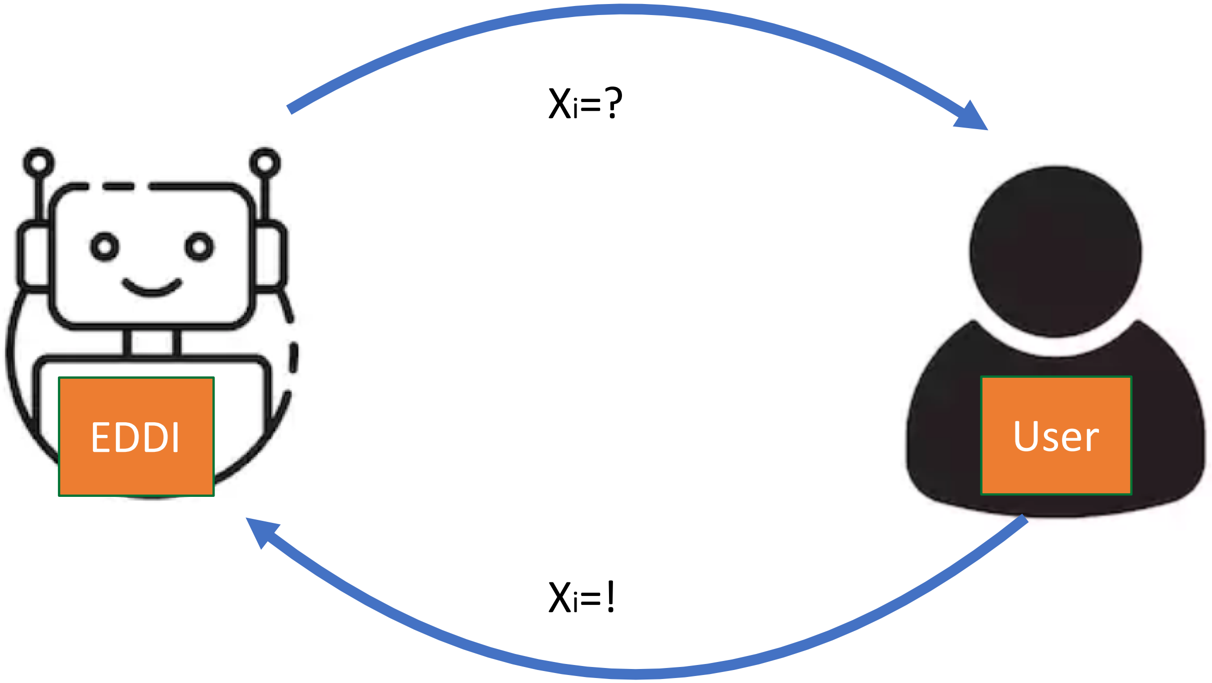


Step 1: Missing Value Prediction with Deep Generative Model

Step 2: Active element-wise information acquisition

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

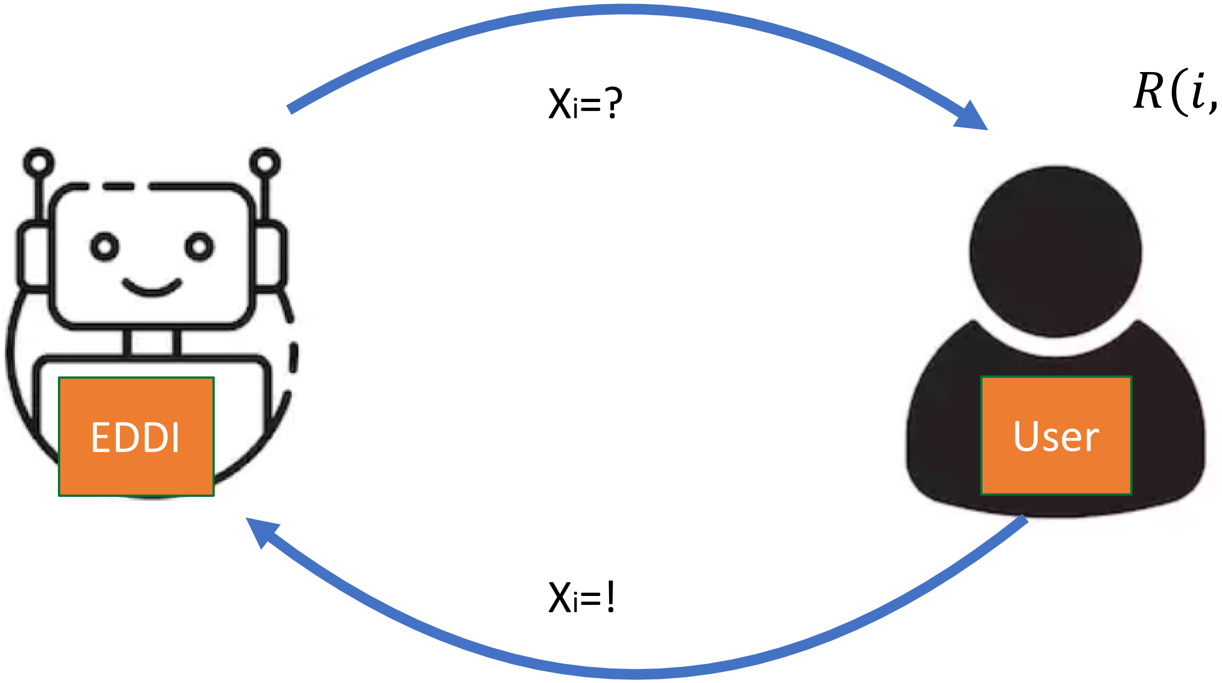
Actively Select the Next Variable



$$i^* = \underset{i \in U}{\arg\max}\, R(i|\boldsymbol{x_O})$$

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

Actively Select the Next Variable

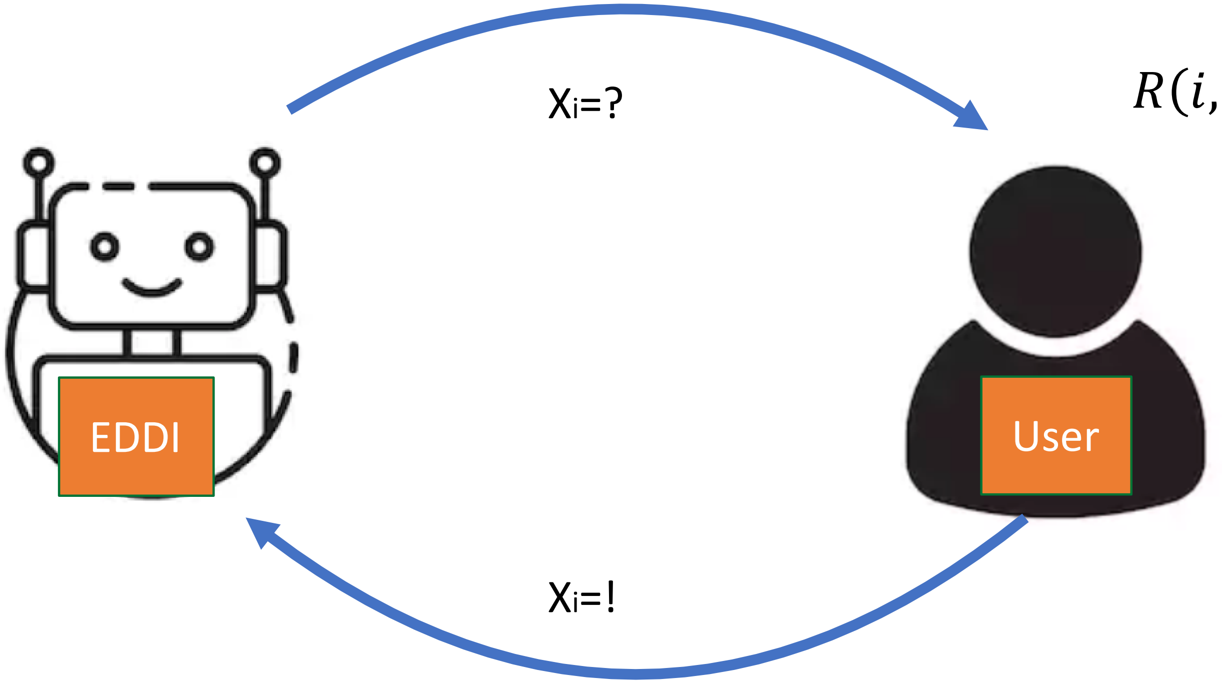


$$R(i, x_o) = E_{x_i \sim p(x_i|x_o)}[KL[p(x_\phi| x_i, x_o)||p(x_\phi|x_o)]$$

$$i^* = \arg\max_{i \in U} R(i|\boldsymbol{x_o})$$

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

Actively Select the Next Variable



$$i^* = \arg\max_{i \in U} R(i|\boldsymbol{x_o})$$

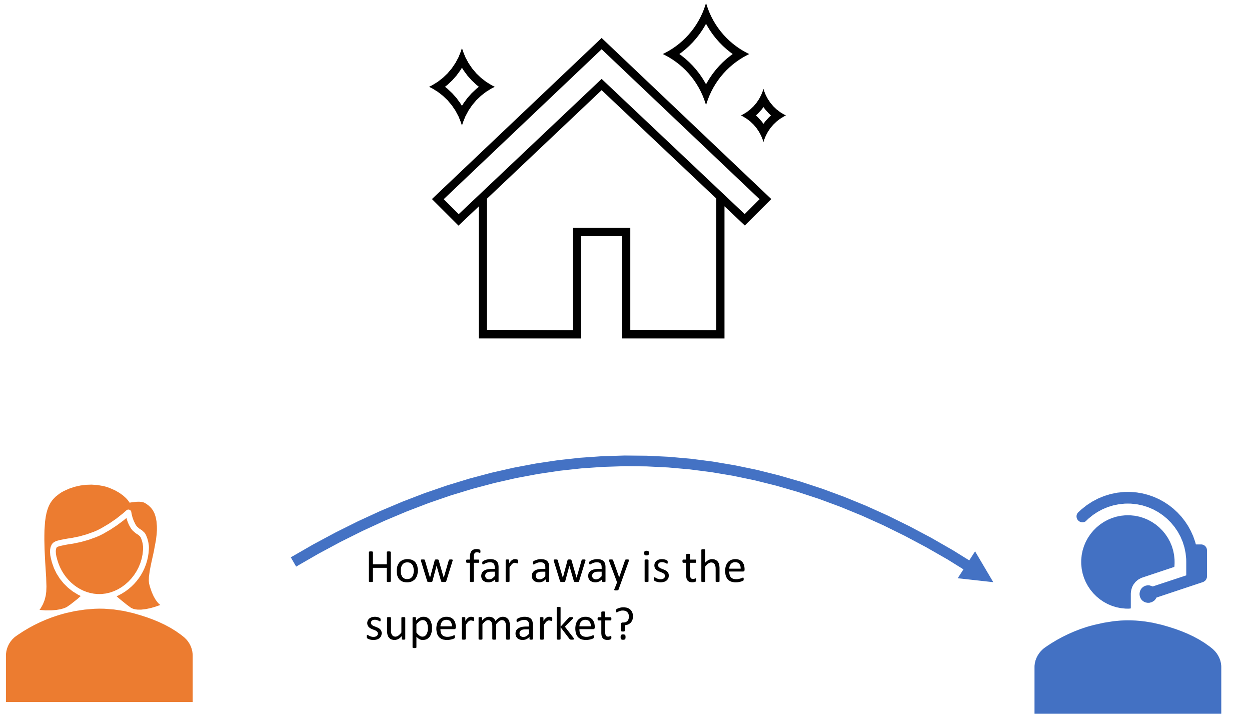
$$R(i, x_o) = E_{x_i \sim p(x_i|x_o)}[KL[p(x_\phi|\, x_i, x_o)||p(x_\phi|x_o)]$$

Intractable

Efficient Approximation

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

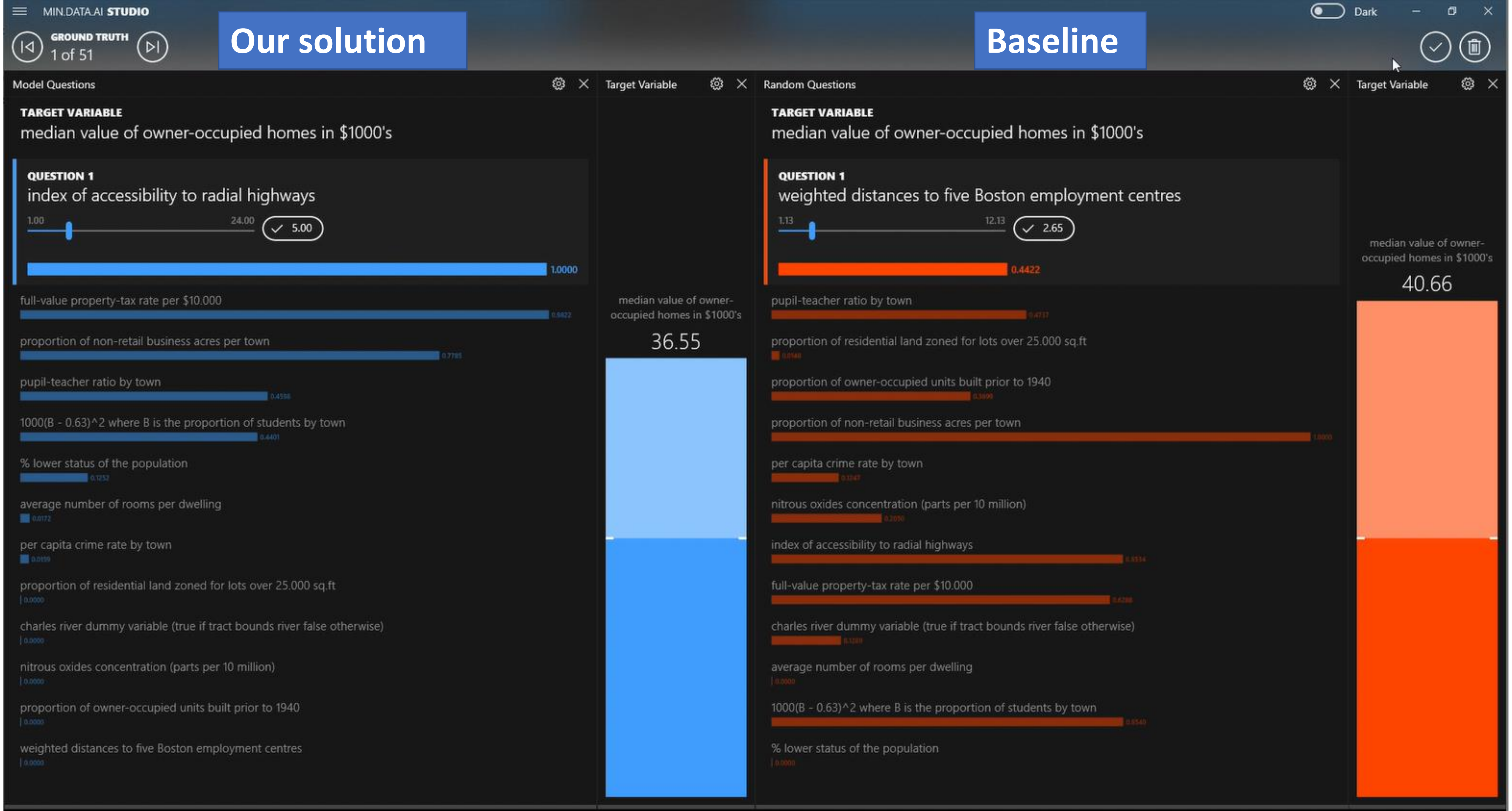
Predicting a House’s Price



How far away is the supermarket?

Kim (customer)

Ty (broker)



Model Questions

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

QUESTION 1
index of accessibility to radial highways

1.00 24.00 ✓ 5.00

1.0000

full-value property-tax rate per $10.000 0.9822

proportion of non-retail business acres per town 0.7785

pupil-teacher ratio by town 0.4598

1000(B - 0.63)^2 where B is the proportion of students by town 0.4401

% lower status of the population 0.1252

average number of rooms per dwelling 0.0172

per capita crime rate by town 0.0159

proportion of residential land zoned for lots over 25.000 sq.ft 0.0000

charles river dummy variable (true if tract bounds river false otherwise) 0.0000

nitrous oxides concentration (parts per 10 million) 0.0000

proportion of owner-occupied units built prior to 1940 0.0000

weighted distances to five Boston employment centres 0.0000

Target Variable

median value of owner-occupied homes in $1000's
36.55

Random Questions

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

QUESTION 1
weighted distances to five Boston employment centres

1.13 12.13 ✓ 2.65

0.4422

pupil-teacher ratio by town 0.4737

proportion of residential land zoned for lots over 25.000 sq.ft 0.0148

proportion of owner-occupied units built prior to 1940 0.3699

proportion of non-retail business acres per town 1.0000

per capita crime rate by town 0.1247

nitrous oxides concentration (parts per 10 million) 0.2050

index of accessibility to radial highways 0.6534

full-value property-tax rate per $10.000 0.6288

charles river dummy variable (true if tract bounds river false otherwise) 0.1289

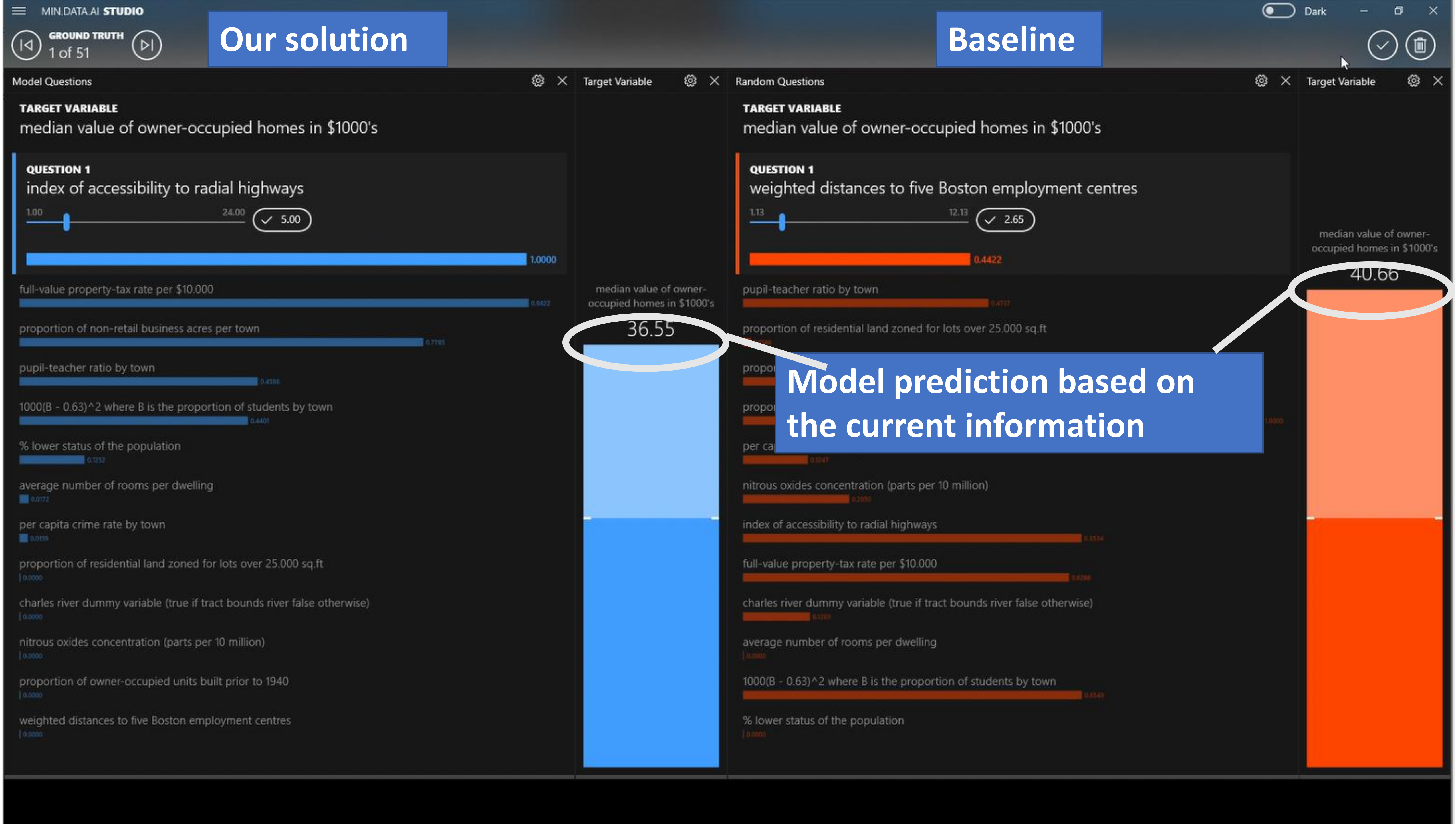
average number of rooms per dwelling 0.0000

1000(B - 0.63)^2 where B is the proportion of students by town 0.6540

% lower status of the population 0.0000

Target Variable

median value of owner-occupied homes in $1000's
40.66



Our solution

Baseline

Model Questions

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

QUESTION 1
index of accessibility to radial highways
1.00 24.00 ✓ 5.00
1.0000

full-value property-tax rate per $10.000 0.9822
proportion of non-retail business acres per town 0.7785
pupil-teacher ratio by town 0.4598
1000(B - 0.63)^2 where B is the proportion of students by town 0.4401
% lower status of the population 0.1252
average number of rooms per dwelling 0.0172
per capita crime rate by town 0.0159
proportion of residential land zoned for lots over 25.000 sq.ft 0.0000
charles river dummy variable (true if tract bounds river false otherwise) 0.0000
nitrous oxides concentration (parts per 10 million) 0.0000
proportion of owner-occupied units built prior to 1940 0.0000
weighted distances to five Boston employment centres 0.0000

Target Variable
median value of owner-occupied homes in $1000's
36.55

Random Questions

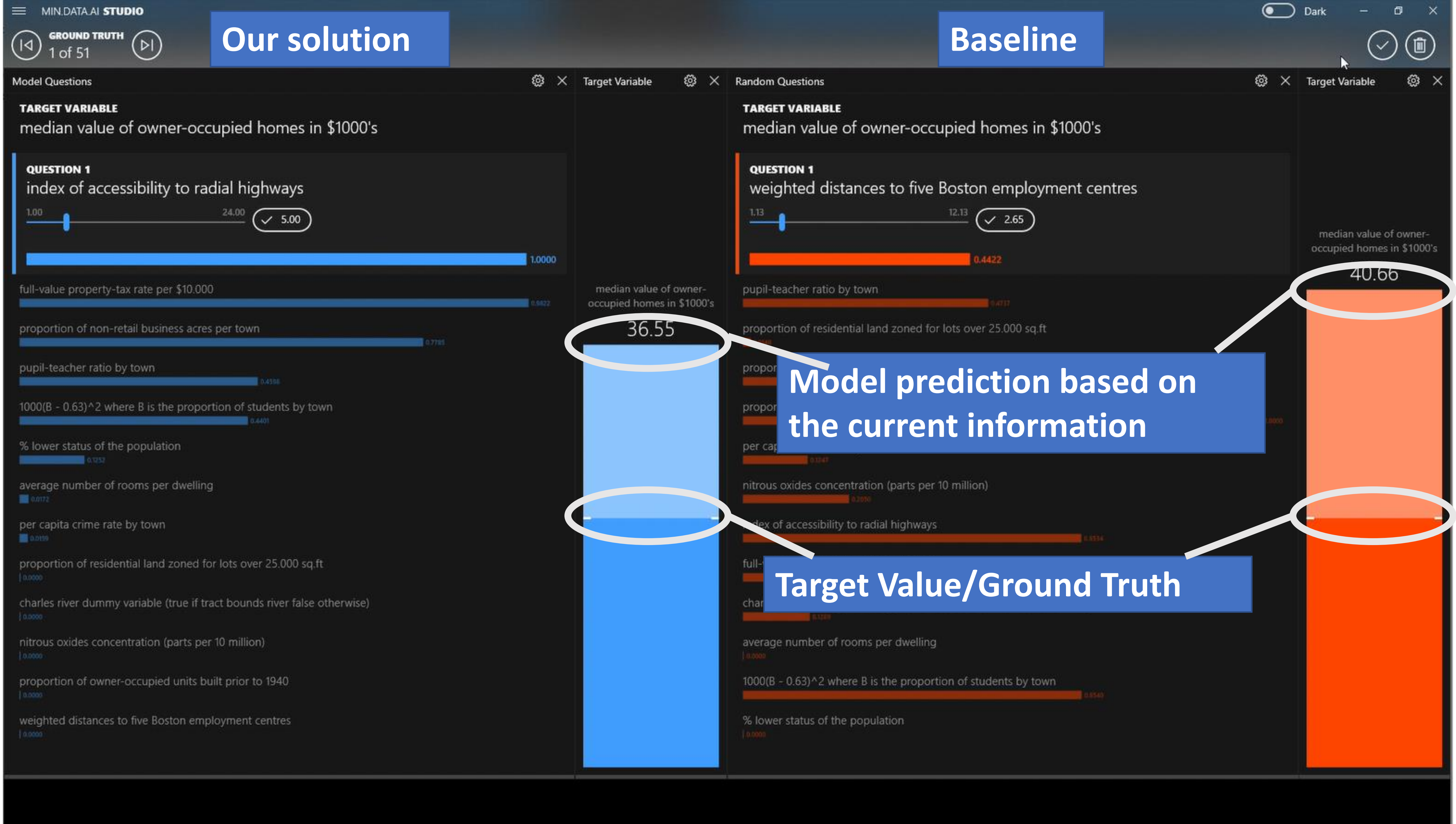
TARGET VARIABLE
median value of owner-occupied homes in $1000's

QUESTION 1
weighted distances to five Boston employment centres
1.13 12.13 ✓ 2.65
0.4422

pupil-teacher ratio by town 0.4737
proportion of residential land zoned for lots over 25.000 sq.ft 0.0348
propo
propo 1.0000
per ca 0.1247
nitrous oxides concentration (parts per 10 million) 0.2050
index of accessibility to radial highways 0.5534
full-value property-tax rate per $10.000 0.5288
charles river dummy variable (true if tract bounds river false otherwise) 0.1289
average number of rooms per dwelling 0.0000
1000(B - 0.63)^2 where B is the proportion of students by town 0.5540
% lower status of the population 0.0000

Target Variable
median value of owner-occupied homes in $1000's
40.66

Model prediction based on the current information



Our solution

Baseline

Model Questions

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

QUESTION 1
index of accessibility to radial highways

1.00 24.00 ✓ 5.00

1.0000

full-value property-tax rate per $10.000 0.9822

proportion of non-retail business acres per town 0.7785

pupil-teacher ratio by town 0.4598

1000(B - 0.63)^2 where B is the proportion of students by town 0.4401

% lower status of the population 0.1252

average number of rooms per dwelling 0.0172

per capita crime rate by town 0.0159

proportion of residential land zoned for lots over 25.000 sq.ft 0.0000

charles river dummy variable (true if tract bounds river false otherwise) 0.0000

nitrous oxides concentration (parts per 10 million) 0.0000

proportion of owner-occupied units built prior to 1940 0.0000

weighted distances to five Boston employment centres 0.0000

Target Variable

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

36.55

Random Questions

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

QUESTION 1
weighted distances to five Boston employment centres

1.13 12.13 ✓ 2.65

0.4422

pupil-teacher ratio by town 0.4737

proportion of residential land zoned for lots over 25.000 sq.ft 0.0548

propor

propor 1.0000

per cap 0.1247

nitrous oxides concentration (parts per 10 million) 0.2050

index of accessibility to radial highways 0.8534

full-

char 0.1289

average number of rooms per dwelling 0.0000

1000(B - 0.63)^2 where B is the proportion of students by town 0.8540

% lower status of the population 0.0000

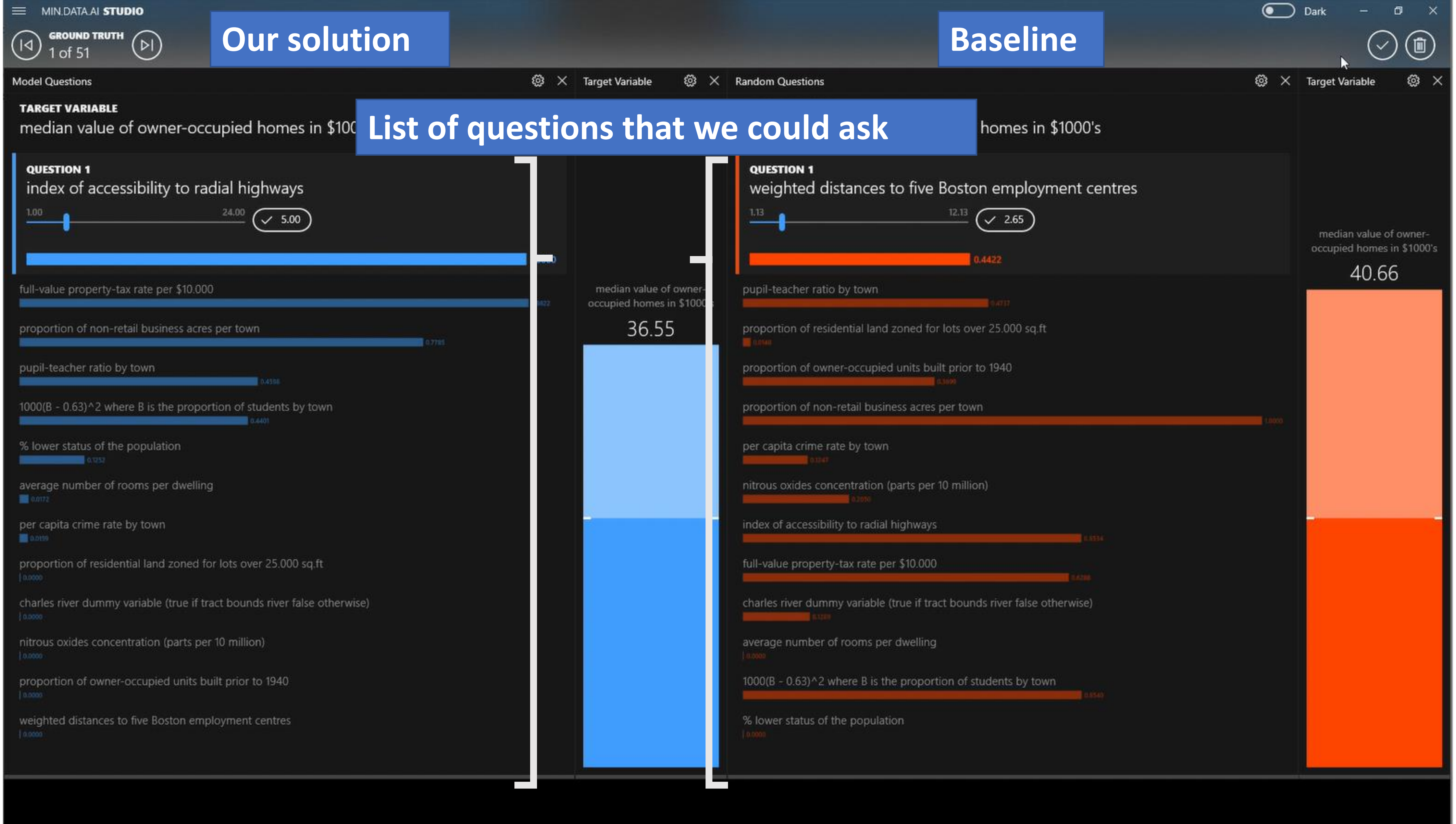
Target Variable

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

40.66

Model prediction based on the current information

Target Value/Ground Truth



Our solution

Baseline

List of questions that we could ask

Model Questions

TARGET VARIABLE
median value of owner-occupied homes in $100

QUESTION 1
index of accessibility to radial highways
1.00 24.00 ✓ 5.00

full-value property-tax rate per $10.000
proportion of non-retail business acres per town 0.7785
pupil-teacher ratio by town 0.4598
1000(B - 0.63)^2 where B is the proportion of students by town 0.4401
% lower status of the population 0.1252
average number of rooms per dwelling 0.0172
per capita crime rate by town 0.0159
proportion of residential land zoned for lots over 25.000 sq.ft 0.0000
charles river dummy variable (true if tract bounds river false otherwise) 0.0000
nitrous oxides concentration (parts per 10 million) 0.0000
proportion of owner-occupied units built prior to 1940 0.0000
weighted distances to five Boston employment centres 0.0000

Target Variable

median value of owner-occupied homes in $1000's
36.55

Random Questions

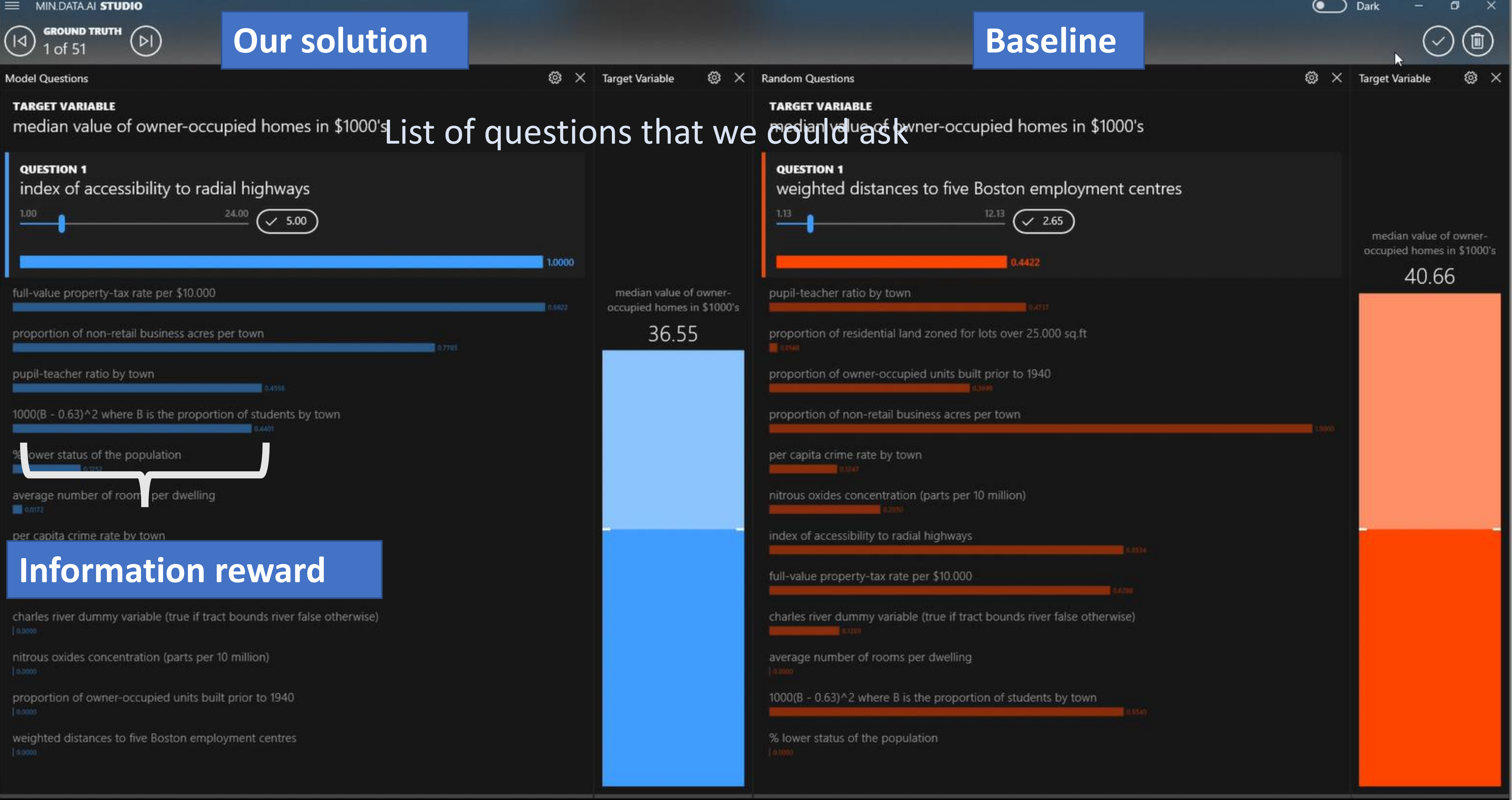
homes in $1000's

QUESTION 1
weighted distances to five Boston employment centres
1.13 12.13 ✓ 2.65
0.4422

pupil-teacher ratio by town 0.4737
proportion of residential land zoned for lots over 25.000 sq.ft 0.0148
proportion of owner-occupied units built prior to 1940 0.3699
proportion of non-retail business acres per town 1.0000
per capita crime rate by town 0.1247
nitrous oxides concentration (parts per 10 million) 0.2050
index of accessibility to radial highways 0.6534
full-value property-tax rate per $10.000 0.6288
charles river dummy variable (true if tract bounds river false otherwise) 0.1289
average number of rooms per dwelling 0.0000
1000(B - 0.63)^2 where B is the proportion of students by town 0.6540
% lower status of the population 0.0000

Target Variable

median value of owner-occupied homes in $1000's
40.66



Our solution

Baseline

List of questions that we could ask

Information reward

MIN.DATA.AI STUDIO

GROUND TRUTH
1 of 51

Dark

Model Questions

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

QUESTION 1
index of accessibility to radial highways
1.00 24.00 ✓ 5.00
1.0000

full-value property-tax rate per $10.000 0.9822

proportion of non-retail business acres per town 0.7785

pupil-teacher ratio by town 0.4558

1000(B - 0.63)^2 where B is the proportion of students by town 0.4401

% lower status of the population 0.1257

average number of rooms per dwelling 0.0172

per capita crime rate by town

charles river dummy variable (true if tract bounds river false otherwise) 0.0000

nitrous oxides concentration (parts per 10 million) 0.0000

proportion of owner-occupied units built prior to 1940 0.0000

weighted distances to five Boston employment centres 0.0000

Target Variable

median value of owner-occupied homes in $1000's
36.55

Random Questions

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

QUESTION 1
weighted distances to five Boston employment centres
1.13 12.13 ✓ 2.65
0.4422

pupil-teacher ratio by town 0.4737

proportion of residential land zoned for lots over 25.000 sq.ft 0.0148

proportion of owner-occupied units built prior to 1940 0.3699

proportion of non-retail business acres per town 1.0000

per capita crime rate by town 0.1247

nitrous oxides concentration (parts per 10 million) 0.2050

index of accessibility to radial highways 0.6514

full-value property-tax rate per $10.000 0.6288

charles river dummy variable (true if tract bounds river false otherwise) 0.1269

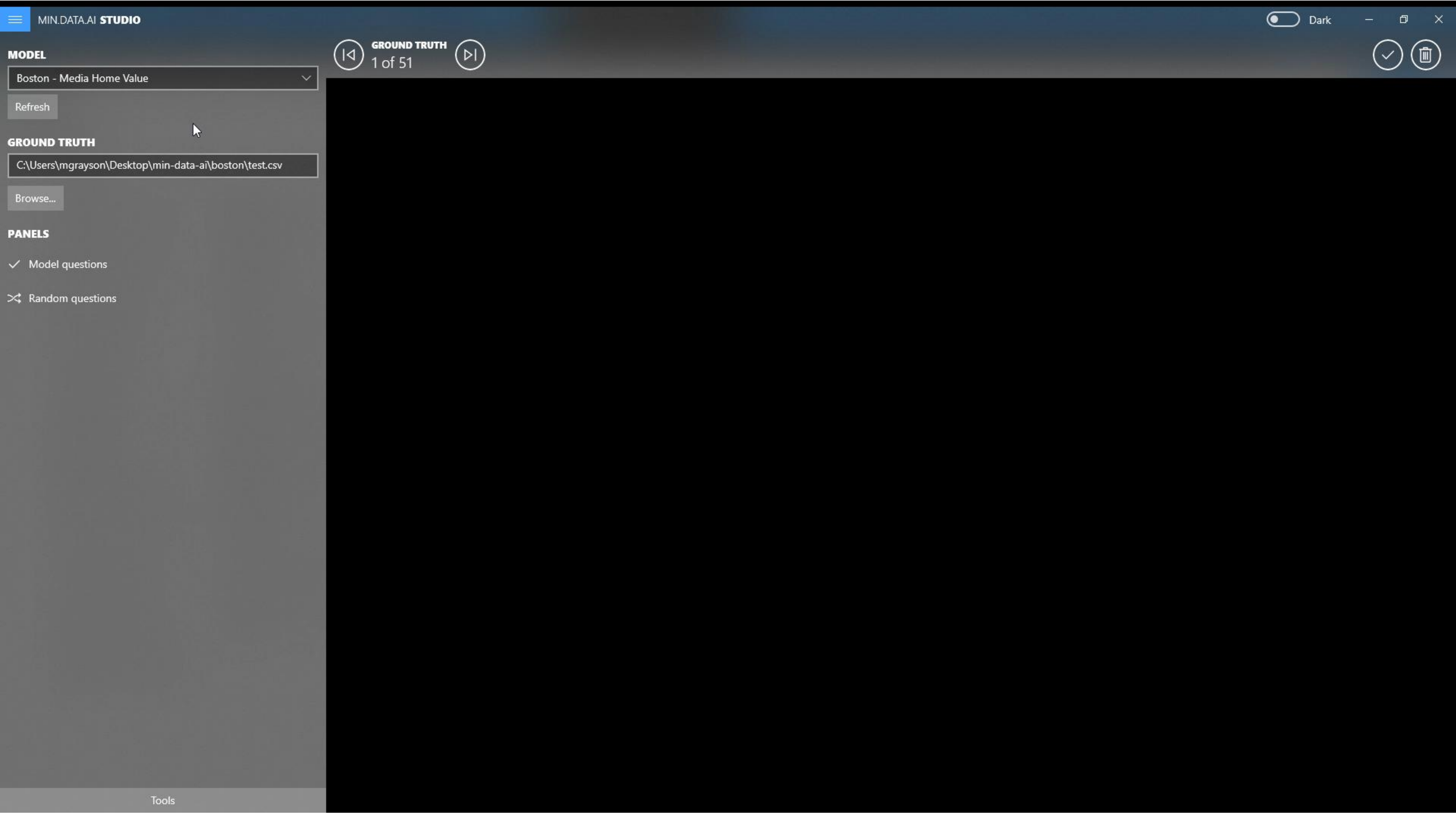
average number of rooms per dwelling 0.0000

1000(B - 0.63)^2 where B is the proportion of students by town 0.6540

% lower status of the population 0.0000

Target Variable

median value of owner-occupied homes in $1000's
40.66



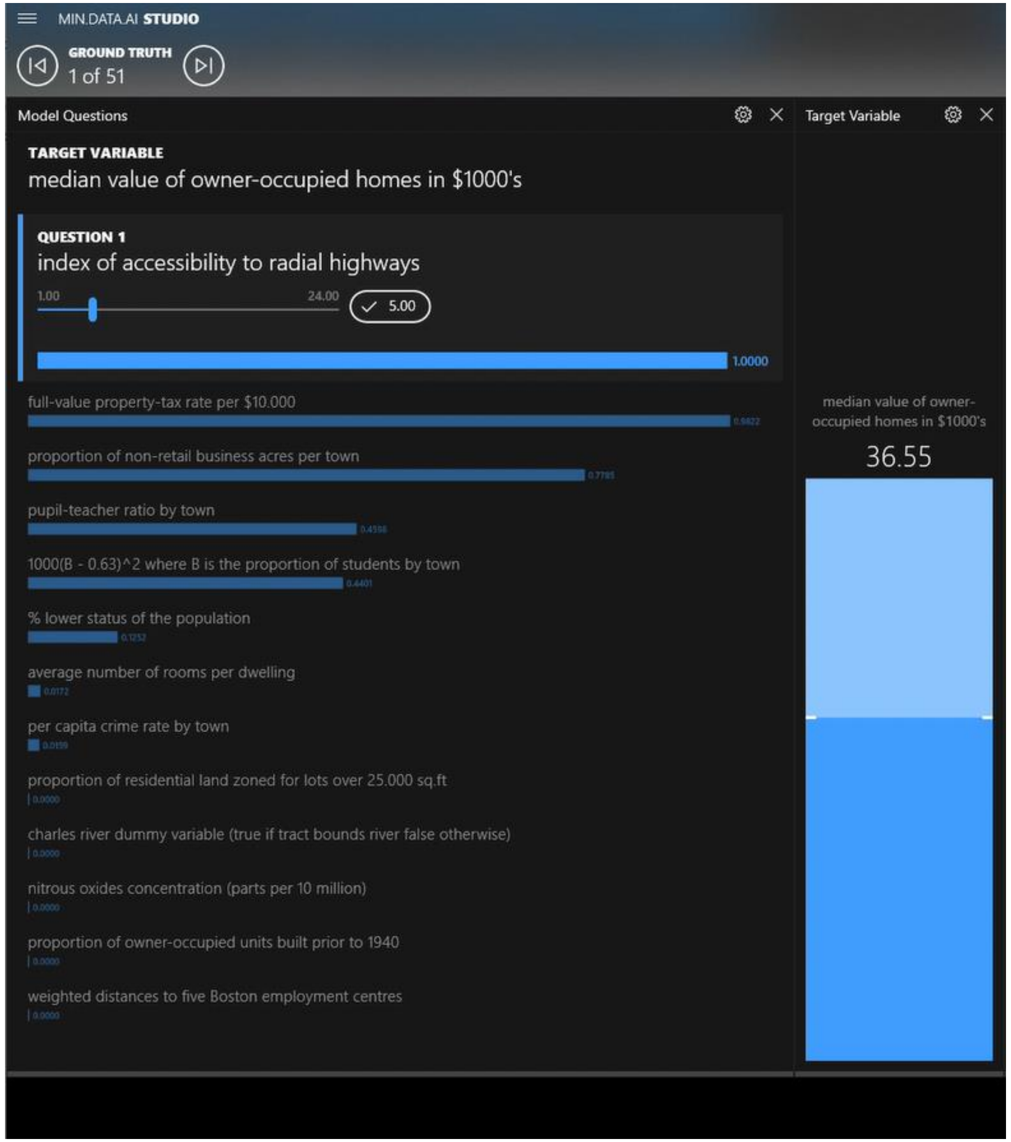
MODEL

Boston - Media Home Value

GROUND TRUTH

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

GROUND TRUTH 1 of 51



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

QUESTION 1
index of accessibility to radial highways
1.00 24.00 ✓ 5.00
1.0000

full-value property-tax rate per $10.000
0.9822

proportion of non-retail business acres per town
0.7785

pupil-teacher ratio by town
0.4598

1000(B - 0.63)^2 where B is the proportion of students by town
0.4401

% lower status of the population
0.1252

average number of rooms per dwelling
0.0172

per capita crime rate by town
0.0159

proportion of residential land zoned for lots over 25.000 sq.ft
0.0000

charles river dummy variable (true if tract bounds river false otherwise)
0.0000

nitrous oxides concentration (parts per 10 million)
0.0000

proportion of owner-occupied units built prior to 1940
0.0000

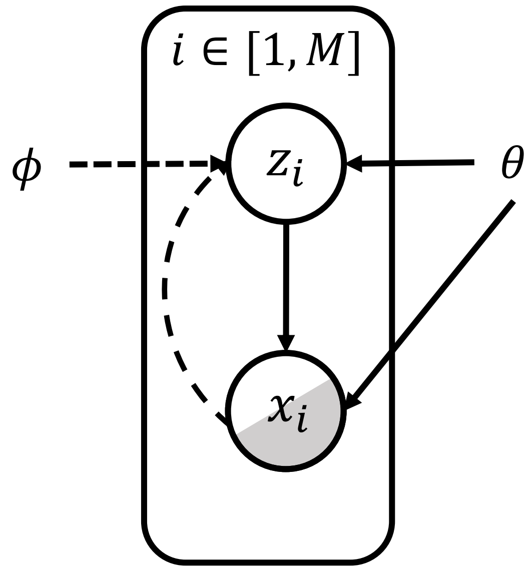
weighted distances to five Boston employment centres
0.0000

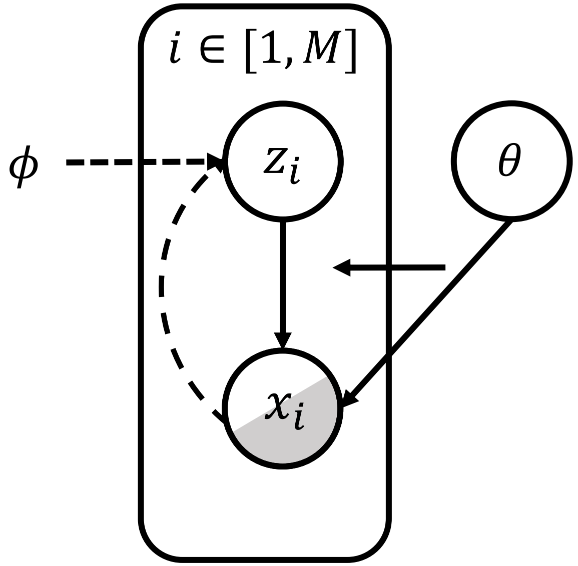
median value of owner-occupied homes in $1000's
36.55

- 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 θ

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

- Stochastic variable θ

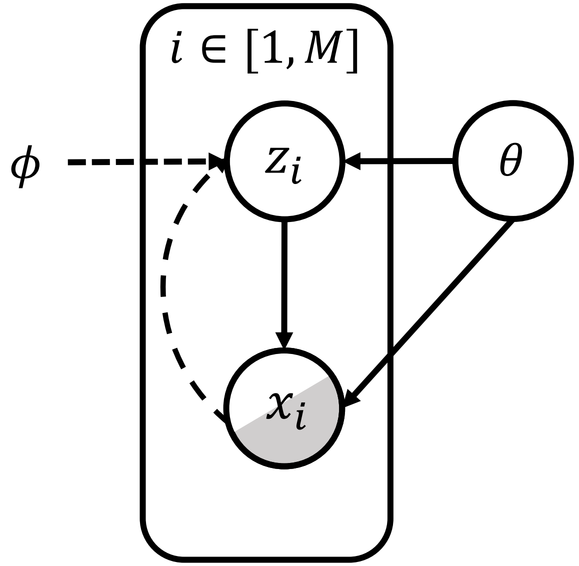
$$p(\boldsymbol{x_o}, \theta, \boldsymbol{z}) = p(\theta) \prod_{i=1}^{N} \prod_{d \in O_i} p(x_{i,d} | z_i, \theta) p(z_i)$$

Gong et al. Icebreaker: Element-wise efficient information acquisition with a Bayesian deep latent gaussian model. NeurIPS 2019

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

Amortized inference for local latent variables

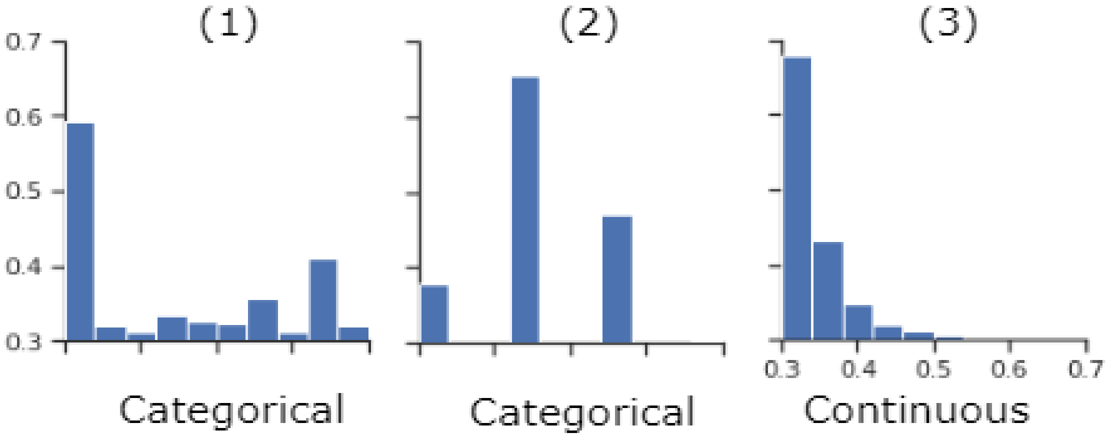
SGHMC for global latent variables



$$q(\theta, \boldsymbol{z} \mid \boldsymbol{x_o}) \approx q(\theta \mid \boldsymbol{x_o}) q_\phi(\boldsymbol{z} \mid \boldsymbol{x_o})$$

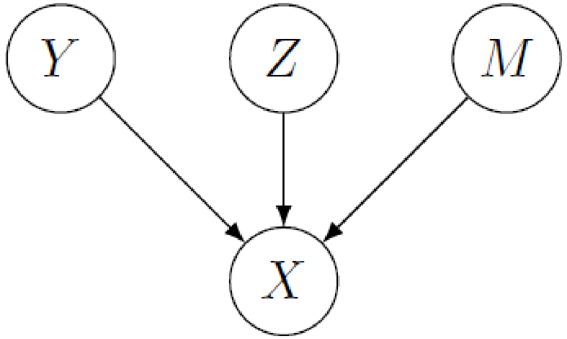
Gong et al. Icebreaker: Element-wise efficient information acquisition with a Bayesian deep latent gaussian model. NeurIPS 2019

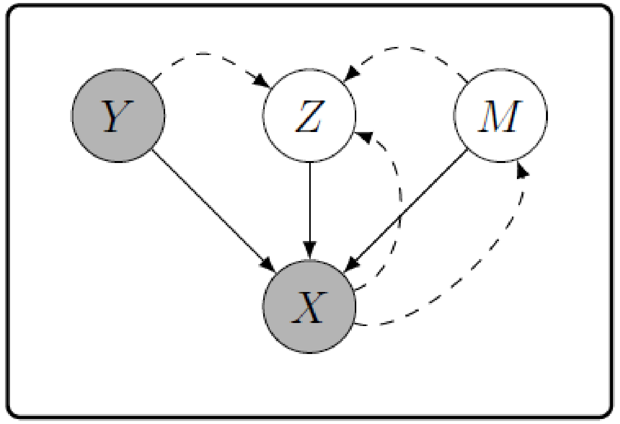
When Data are Heterogenous



Ma et.al. VAEM: a Deep Generative Model for Heterogeneous Mixed Type Data. NeurIPS 2020

When Robustness is Needed





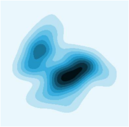
Causal Reasoning

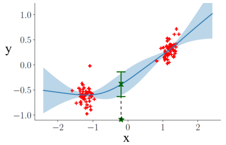
Deep Casual
Manipulation
Augmented Model

Zhang et.al. A Causal View on Robustness of Neural Networks. NeurIPS 2020

Summary

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



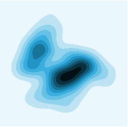


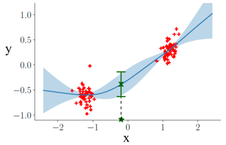
Approximate Inference

$$\int F(\theta)\,\boxed{\pi(\theta)}\,d\theta$$

Summary

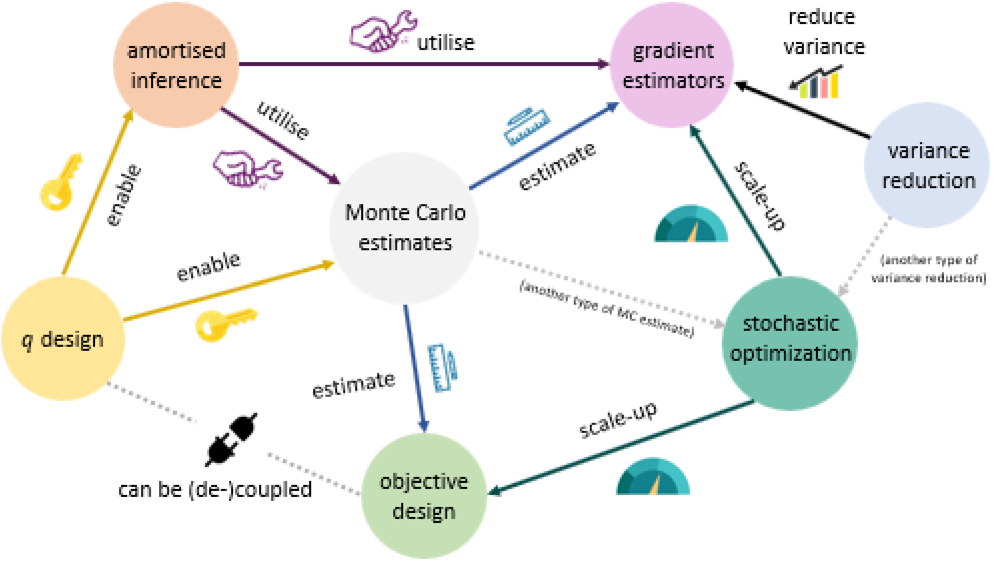
The probability distribution $\pi(\theta)$ is intractable





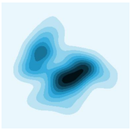
Approximate Inference

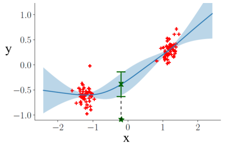
$$\int F(\theta)\boxed{\pi(\theta)}d\theta$$



Summary

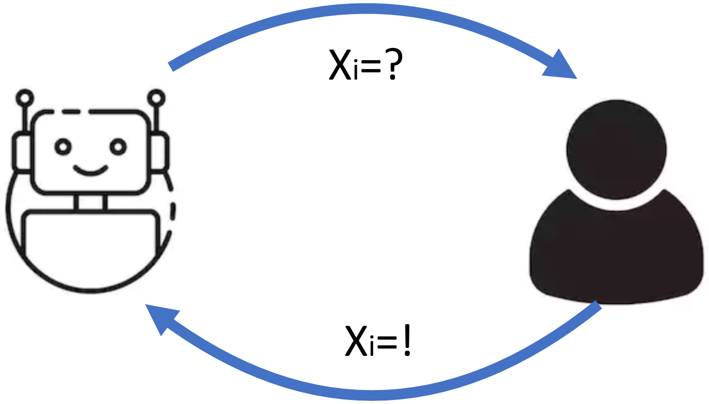
The probability distribution $\pi(\theta)$ is intractable

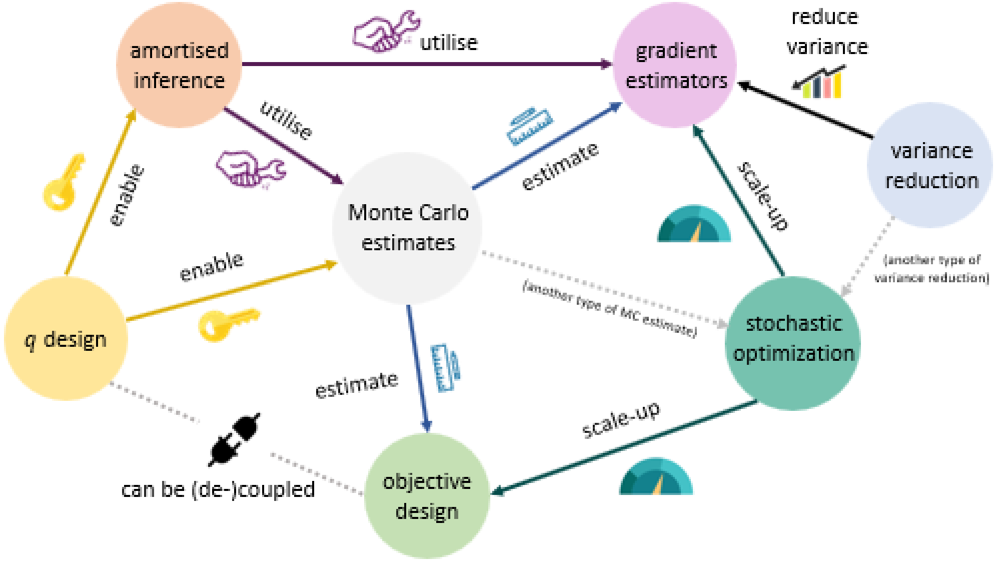




Approximate Inference

$$\int F(\theta)\boxed{\pi(\theta)}d\theta$$





amortised inference

utilise

gradient estimators

reduce variance

utilise

estimate

enable

Monte Carlo estimates

scale-up

variance reduction

enable

(another type of variance reduction)

(another type of MC estimate)

stochastic optimization

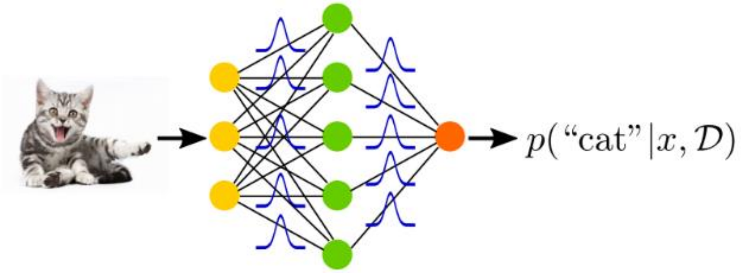
q design

estimate

scale-up

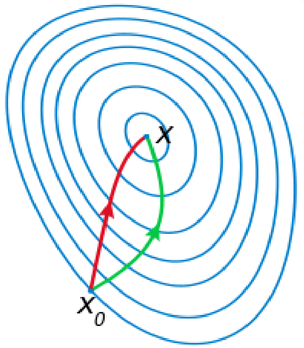
can be (de-)coupled

objective design



Future Directions: Methodology

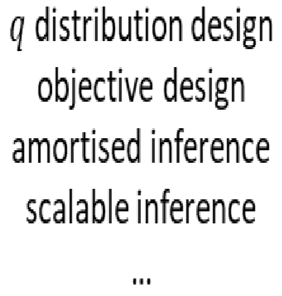
Better optimization



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

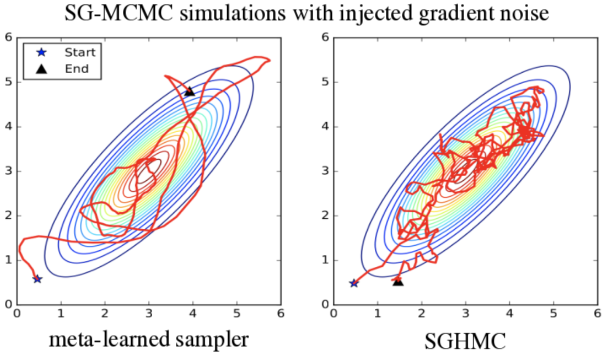
…

Combined approaches

rejection sampling
importance sampling
SMC, MCMC, Quasi MC
…

q distribution design
objective design
amortised inference
scalable inference
…

Meta-learning inference algorithms



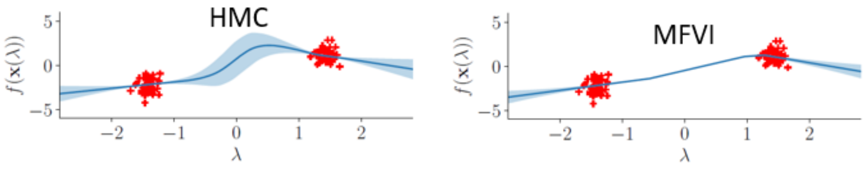
Gong et al. Meta-learning for Stochastic Gradient MCMC. ICLR 2019
Zhang et al. Meta-Learning for Variational Inference. AABI 2019

Future Directions: Error Analyses

Errors in inference

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

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



Analysis needed for deep probabilistic models!

- Optimization error
- Approximation gap

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

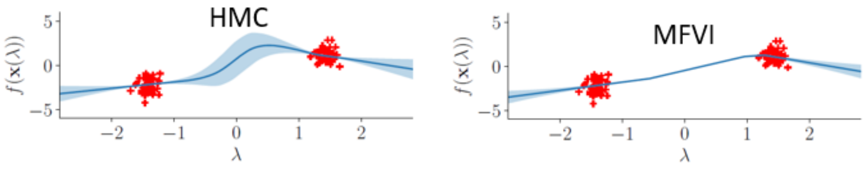
Future Directions: Error Analyses

Errors in inference

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

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

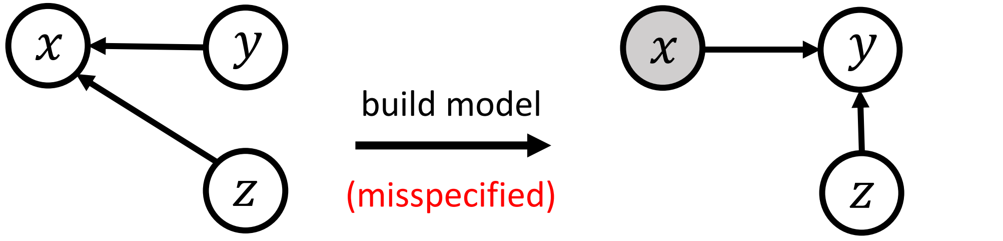
$$= \int p(y^*|x^*, \theta)\; q(\theta) d\theta$$



Analysis needed for deep probabilistic models!

- Optimization error
- Approximation gap

Model misspecification

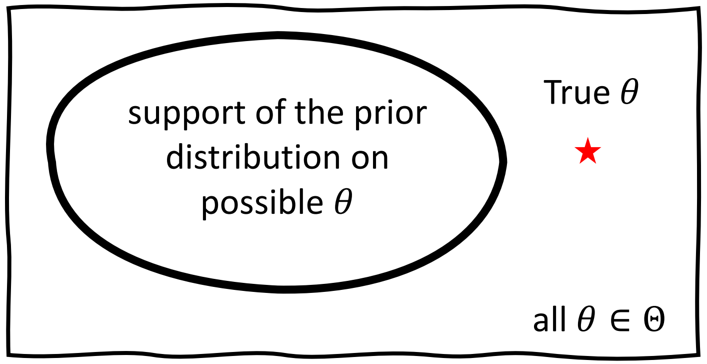


build model

(misspecified)

causal graph

probabilistic graphical model



support of the prior distribution on possible θ

True θ

★

all $\theta \in \Theta$

Wang and Blei. Variational Bayes under Model Misspecification. NeurIPS 2019

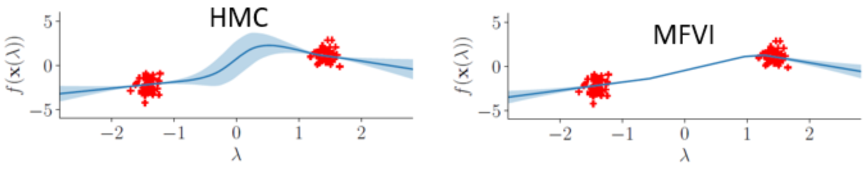
Future Directions: Error Analyses

Errors in inference

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

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

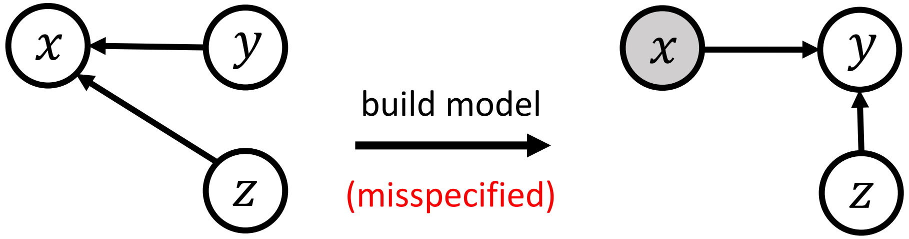
$$= \int p(y^*|x^*,\theta)\; q(\theta) d\theta$$



Analysis needed for deep probabilistic models!

- Optimization error
- Approximation gap

Model misspecification

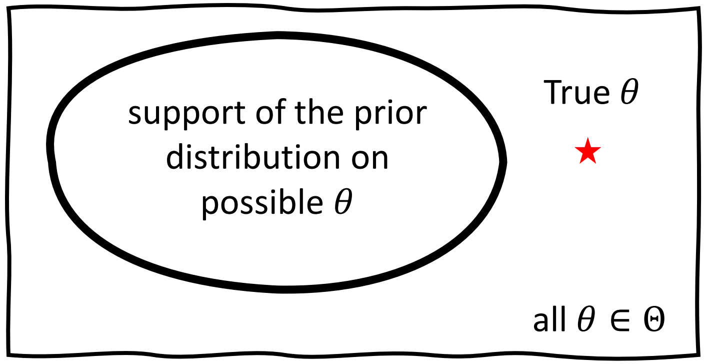


build model

(misspecified)

causal graph

probabilistic graphical model



support of the prior distribution on possible θ

True θ

★

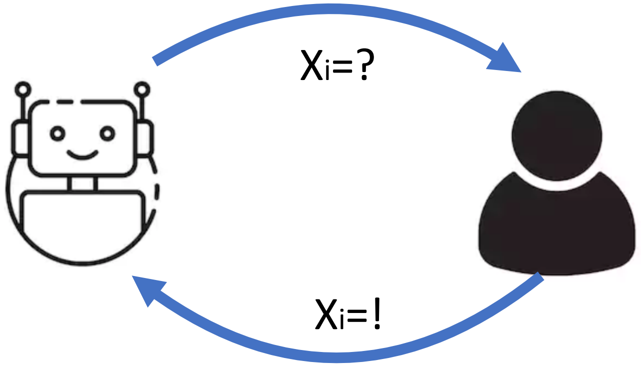
all $\theta \in \Theta$

Separation of inference & modelling?

Wang and Blei. Variational Bayes under Model Misspecification. NeurIPS 2019

Future Directions: Applications

Uncertainty estimation

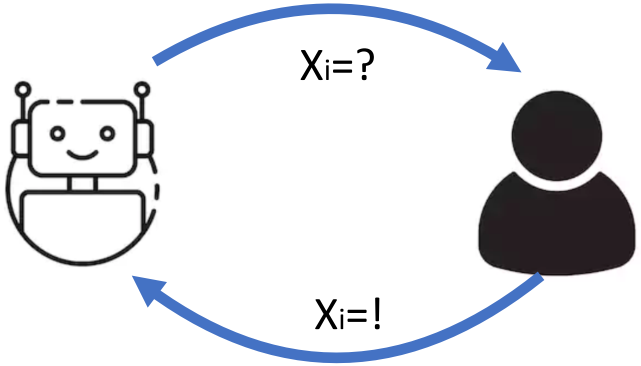


X_i=?

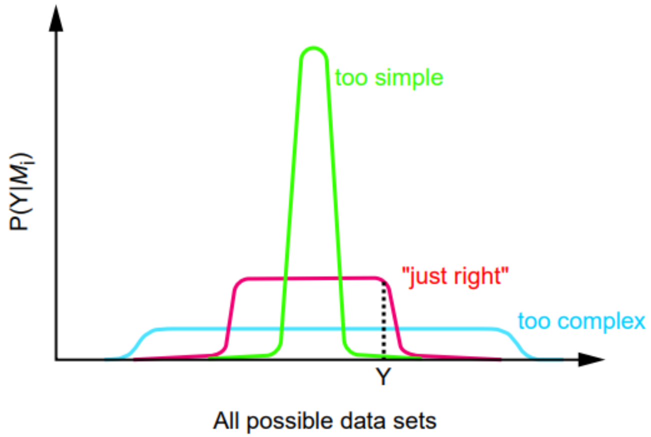
X_i=!

Future Directions: Applications

Uncertainty estimation



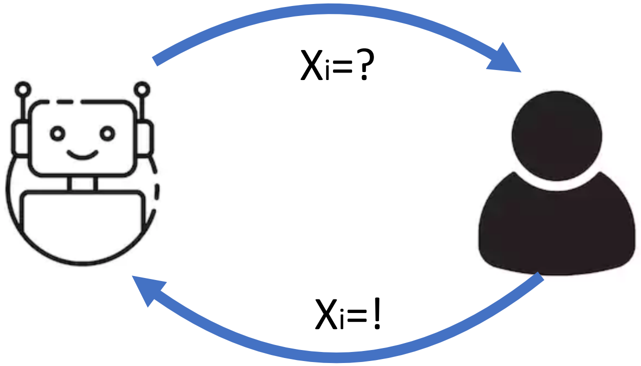
Model selection & averaging



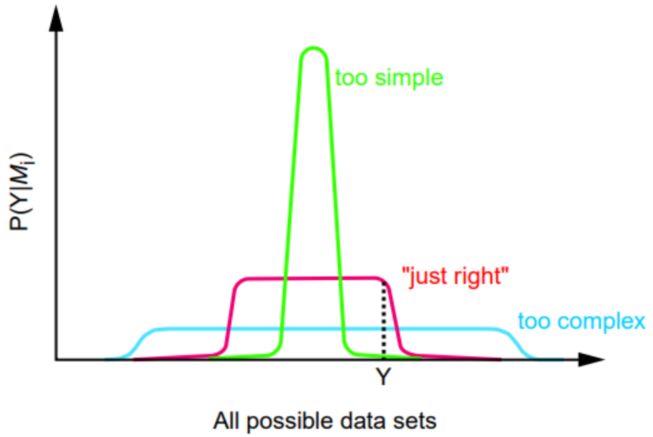
Rasmussen and Ghahramani. Occam's Razor. NeurIPS 2001.

Future Directions: Applications

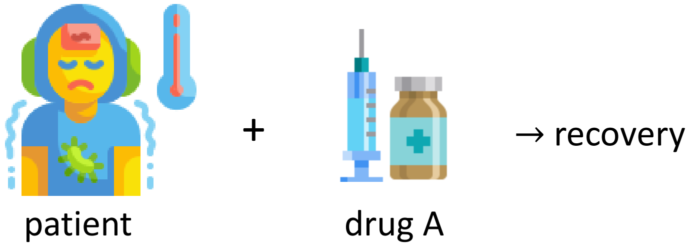
Uncertainty estimation



Model selection & averaging



Causal reasoning



“what if the patient was treated with drug B?”

Thank You!

Questions? Ask at:
liyzhen2@gmail.com (Yingzhen Li)
Cheng.Zhang@microsoft.com (Cheng Zhang)