Probabilistic Reasoning & Variational Inference

Foundations | Tricks | Algorithms

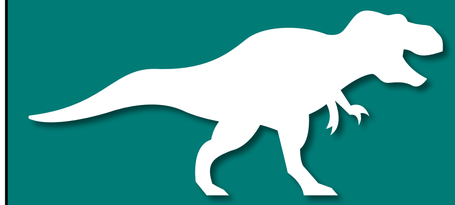
Shakir Mohamed
Research Scientist, DeepMind
@shakir_za

#MLSS2019 London

Principles to Products

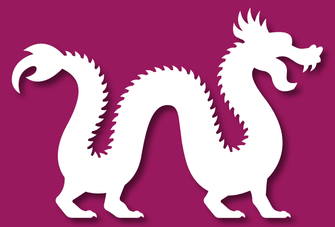
Applications	Assistive Technology	Advancing Science	Climate and Energy	Healthcare	Fairness and Safety	Autonomous systems
Reasoning	Planning	Explanation	Rapid Learning	World Simulation	Objects and Relations	
Information	Uncertainty	Information Gain	Causality	Prediction		
Principles	Probability Theory	Bayesian Analysis	Hypothesis Testing	Estimation Theory	Asymptotics	

Part 1
Probabilistic Foundations



1. Language to think about the Philosophy of Machine Learning

2. Understand the Model-Inference-Algorithm paradigm



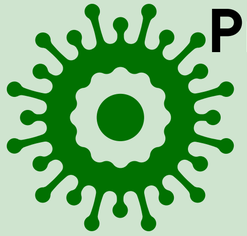
3. Use probabilistic thinking for in supervised, unsupervised, and reinforcement learning.

Probability

Some Definitions for probability

Statistical Probability
Frequency ratio of items

Logical Probability
Degree of confirmation of a hypothesis based on logical analysis



Probability as Propensity
Probability used for predictions

Subjective Probability
Probability as a degree of belief

Probability is sufficient for the task of reasoning under uncertainty

Probability

Probability as a Degree of Belief

Probability is a measure of the belief in a proposition **given** evidence.
A description of a state of knowledge.

No such thing as
the probability
of an event, since the value depends on the evidence used.

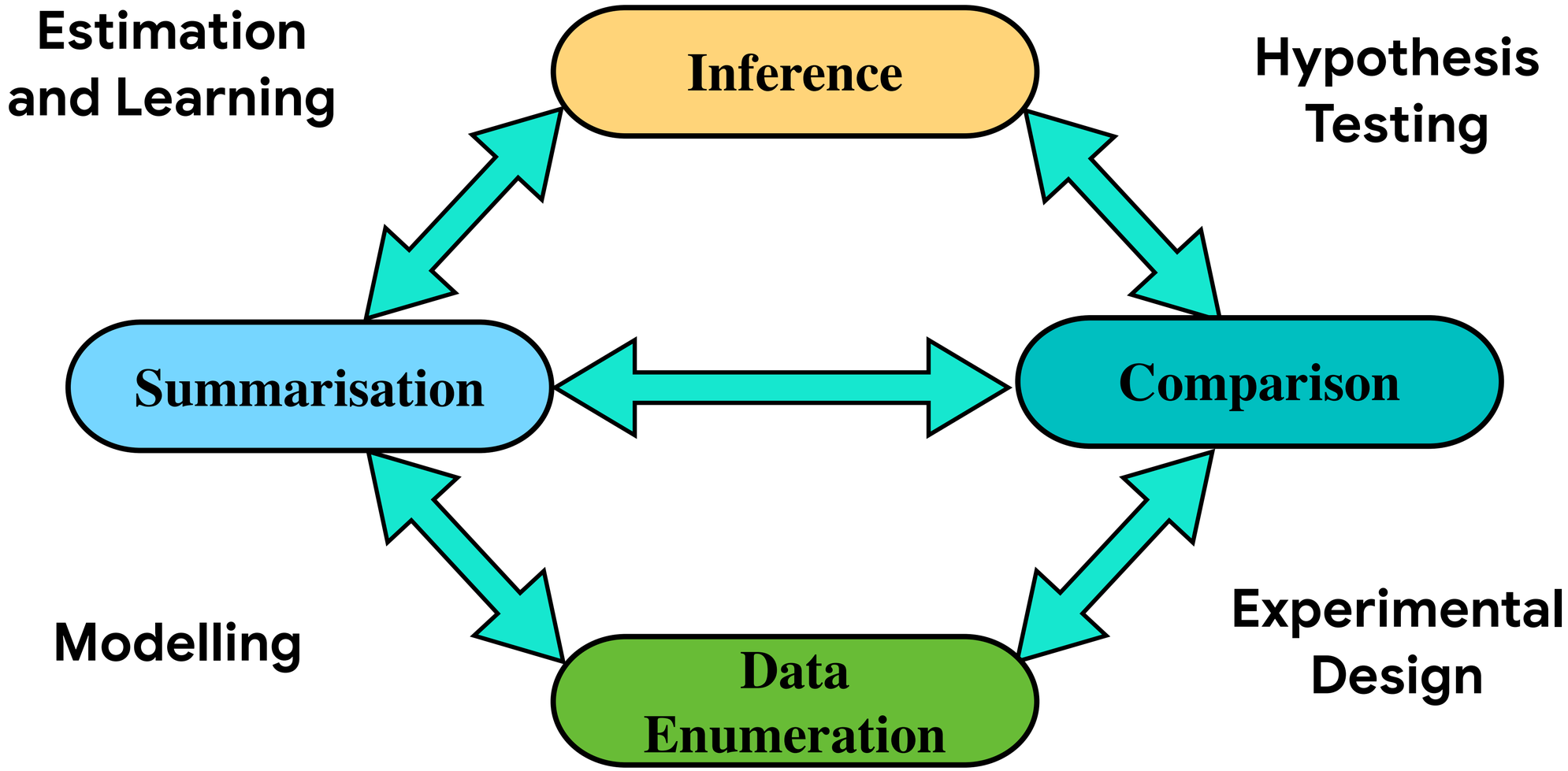
Inherently subjective in that it depends on the believer's information

Different observers with different information will have different beliefs.

Probabilistic Quantities

Probability	$p(\mathbf{x}) \qquad p^*(\mathbf{x}) \qquad q(\mathbf{x})$
Conditions	$p(\mathbf{x}) \geq 0 \qquad \int p(\mathbf{x}) d\mathbf{x} = 1$
Bayes Rule	$p(\mathbf{z}\|\mathbf{x}) = \dfrac{p(\mathbf{x}\|\mathbf{z})p(\mathbf{z})}{p(\mathbf{x})}$
Parameterisation	$p_\theta(\mathbf{x}\|\mathbf{z}) \equiv p(\mathbf{x}\|\mathbf{z};\boldsymbol{\theta})$
Expectation	$\mathbb{E}_{p_\theta(\mathbf{x}\|\mathbf{z})}[f(\mathbf{x};\boldsymbol{\phi})] = \int p_\theta(\mathbf{x}\|\mathbf{z}) f(\mathbf{x};\boldsymbol{\phi}) d\mathbf{x}$
Gradient	$\nabla_\phi f(\mathbf{x};\boldsymbol{\phi}) = \dfrac{\partial f(\mathbf{x};\boldsymbol{\phi})}{\partial \boldsymbol{\phi}}$

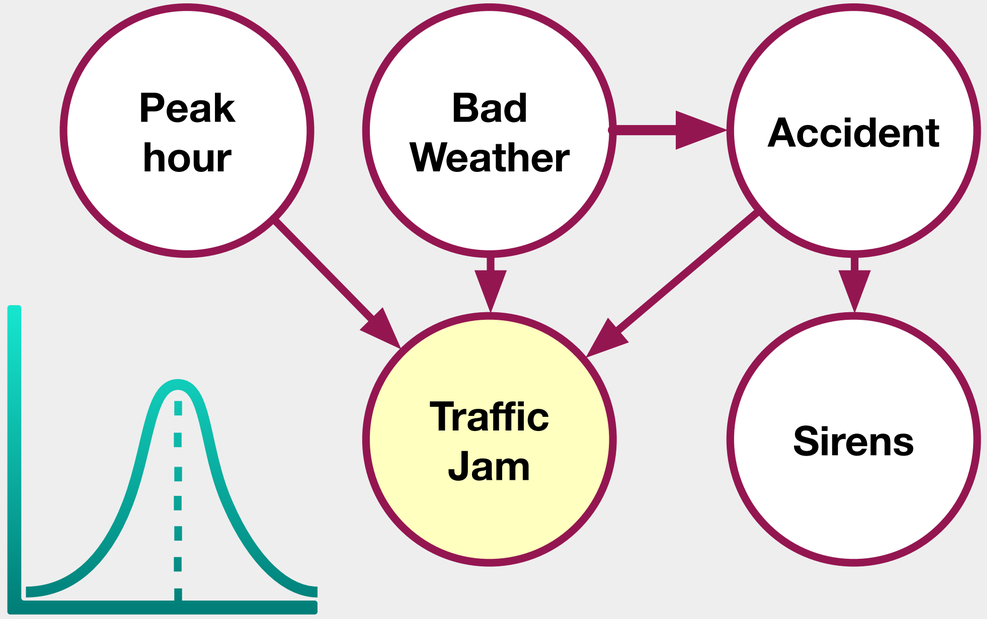
Statistical Operations



Probabilistic Models

Model: Description of the world, of data, of potential scenarios, of processes.

A probabilistic model writes out these models using the language of probability



prob(traffic Jam)

prob(sirens | Accident)

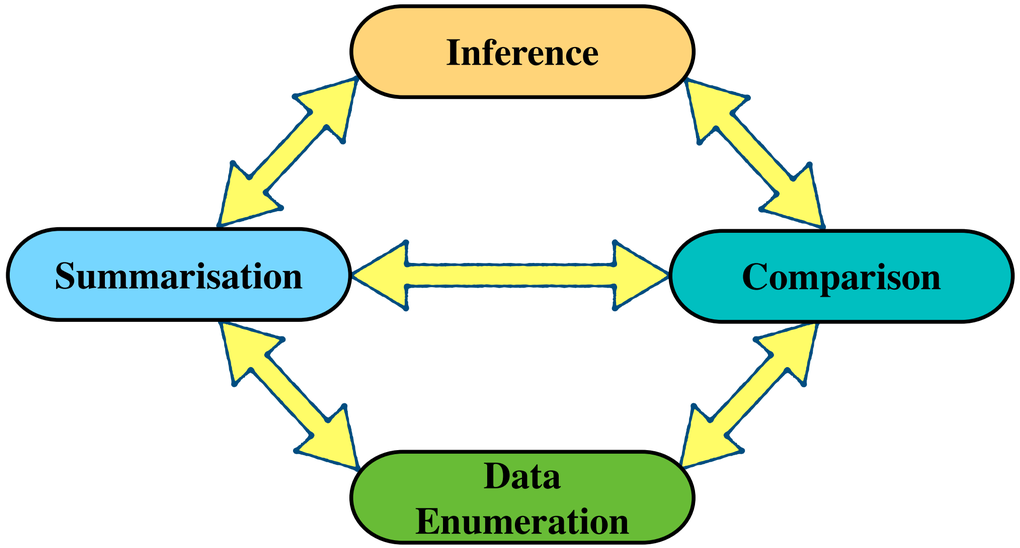
prob(peak hour | Traffic Jam)

Most models in machine learning are probabilistic.

Probabilistic models let you learn probability distributions of data.

You can choose what to learn: Just the mean. Or the entire distribution.

Centrality of Inference



Artificial Intelligence will be the refined instantiation of these statistical operations.

The core questions of AI will be those of probabilistic inference

Inference and Decision-making

Inference

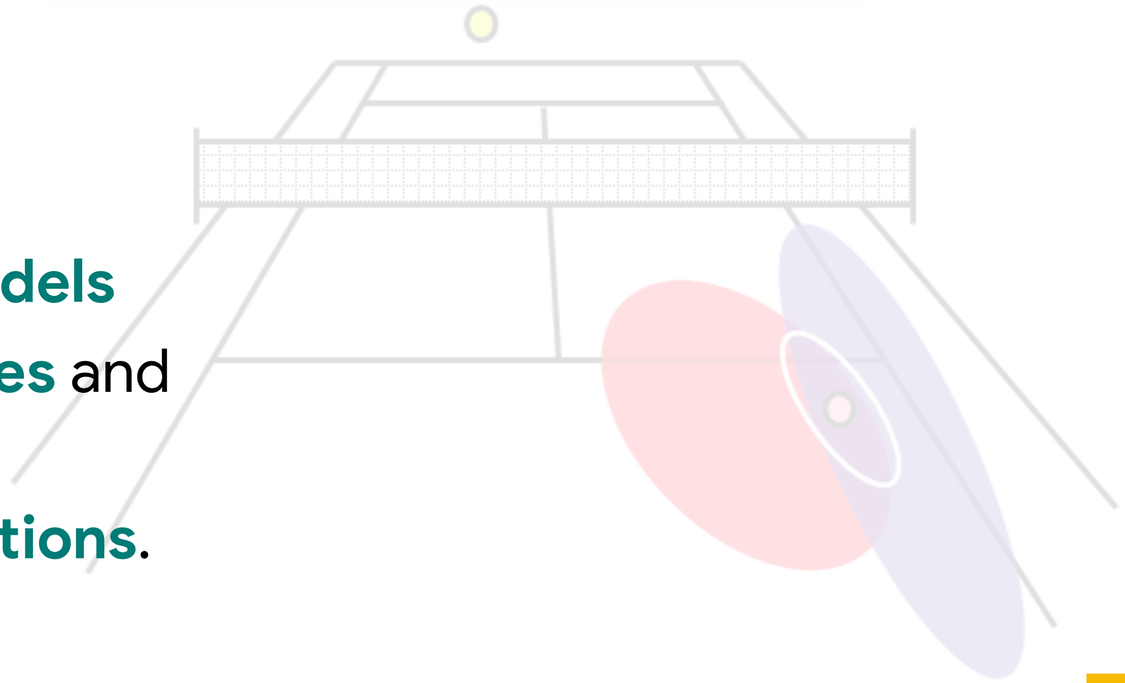
What we can *know* about our data

Decision-making

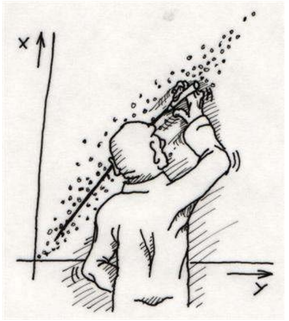
What we can *do* with our data.

Have many of the tools needed to build plausible reasoning systems:

1. Flexible ways of building rich **probabilistic models**
2. Ability to learn and **make consistent inferences** and maintain beliefs
3. Reason about potential outcomes and **take actions**.



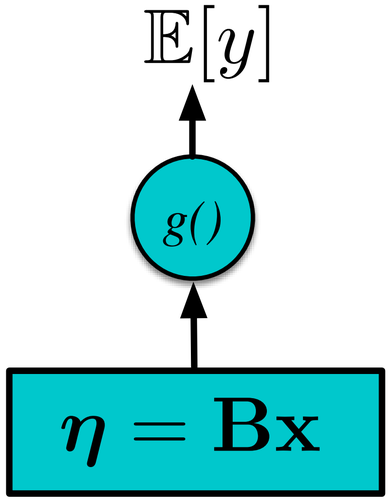
Linear Regression



Generalised Linear Regression

$$\eta = \mathbf{w}^\top \mathbf{x} + b$$

$$p(y|\mathbf{x}) = p(y|g(\eta);\theta)$$



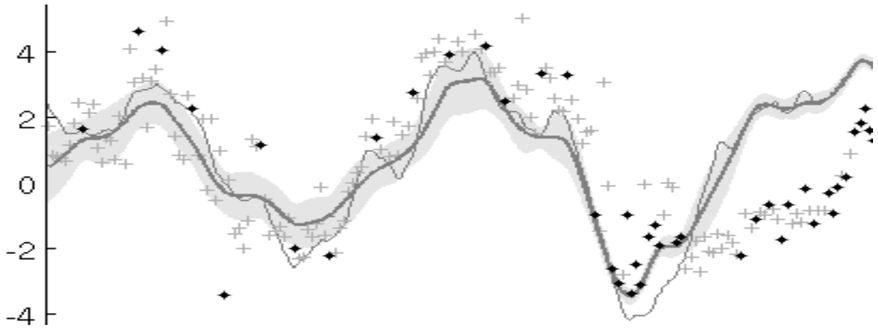
- The basic function can be any linear function, e.g., affine, convolution.
- g(.) is an **inverse link function** that we'll refer to as an activation function.

Target	Regression	Link	Inv link	Activation
Real	Linear	Identity	Identity	
Binary	Logistic	Logit $\log \frac{\mu}{1-\mu}$	Sigmoid $\frac{1}{1+\exp(-\eta)}$	Sigmoid
Binary	Probit	Inv Gauss CDF $\Phi^{-1}(\mu)$	Gauss CDF $\Phi(\eta)$	Probit
Binary	Gumbel	Compl. log-log $\log(-\log(\mu))$	Gumbel CDF $e^{-e^{-x}}$	
Binary	Logistic		Hyperbolic Tangent $\tanh(\eta)$	Tanh
Categorical	Multinomial		Multin. Logit $\frac{\eta_i}{\sum_j \eta_j}$	Softmax
Counts	Poisson	$\log(\mu)$	$\exp(\nu)$	
Counts	Poisson	$\sqrt{(}\mu)$	ν^2	
Non-neg.	Gamma	Reciprocal $\frac{1}{\mu}$	$\frac{1}{\nu}$	
Sparse	Tobit		max $\max(0;\nu)$	ReLU
Ordered	Ordinal		Cum. Logit $\sigma(\phi_k - \eta)$	

Optimise the negative log-likelihood

$$\mathcal{L} = -\log p(y|g(\eta);\theta)$$

Deep Networks

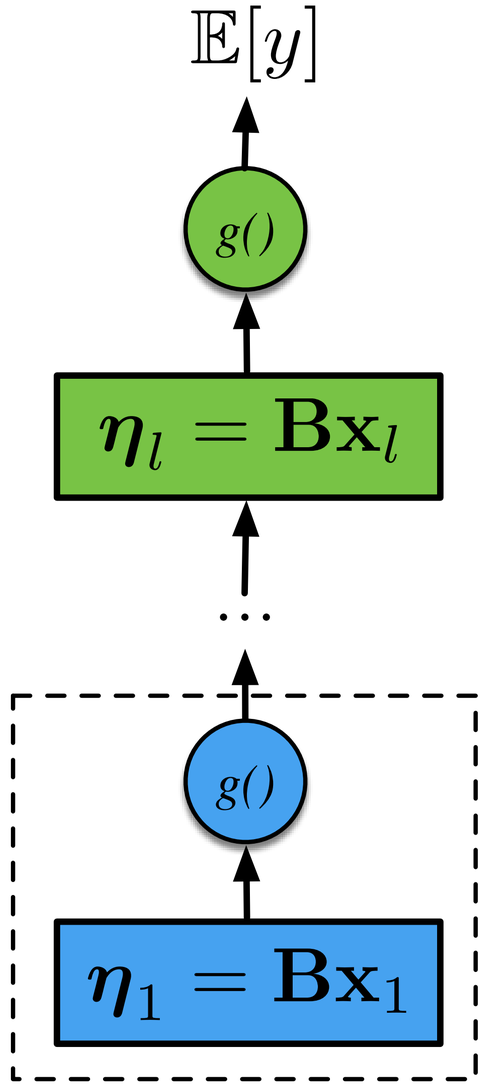


Recursive Generalised Linear Regression

- Recursively compose the basic linear functions.
- Gives a deep neural network.

$$\mathbb{E}[y] = h_L \circ \ldots \circ h_l \circ h_0(\mathbf{x})$$

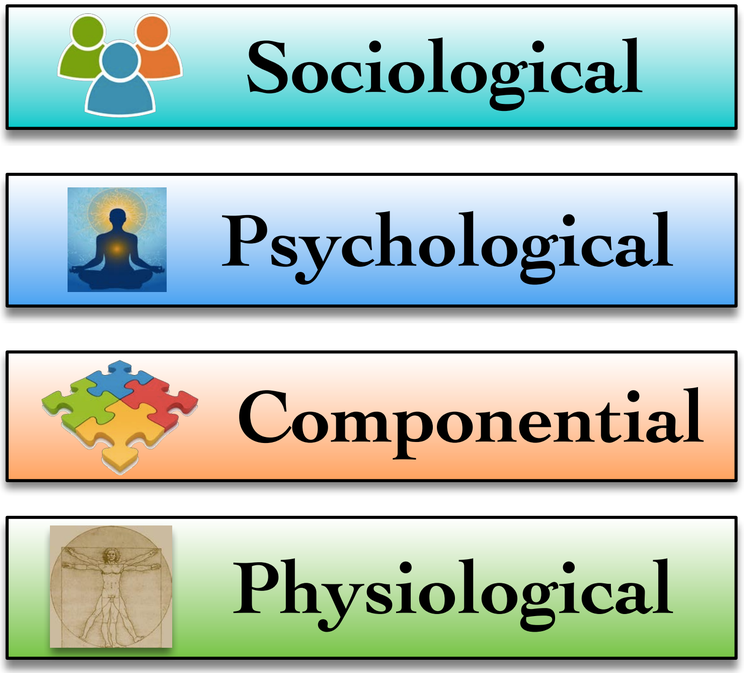
A general, flexible framework for building non-linear, parametric models



Foundations

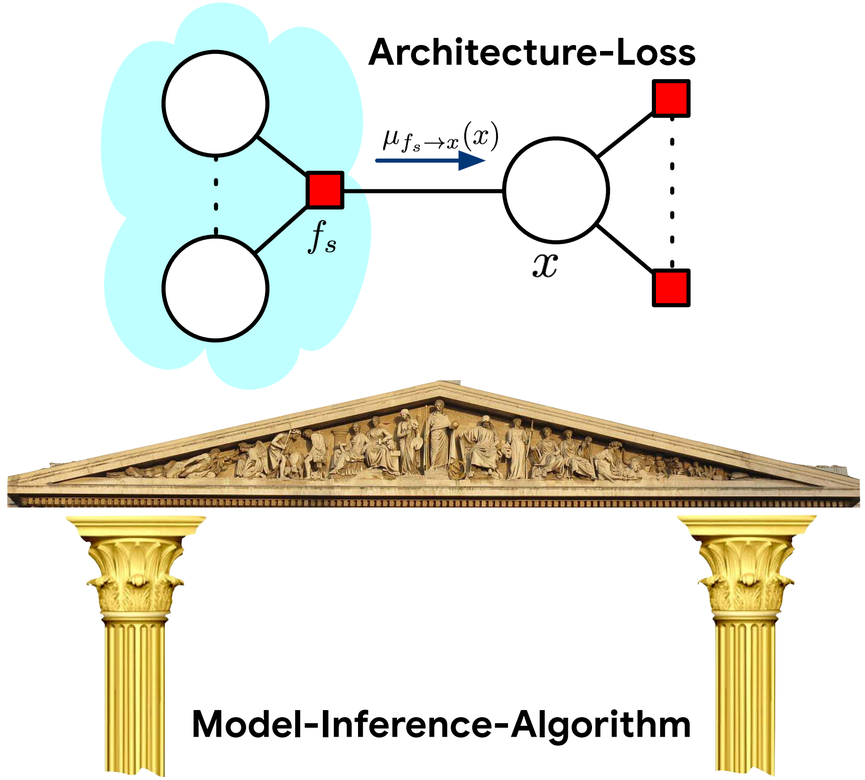
How will you approach your ML research and practice?

In general:
Human-centred,
interdisciplinary approach

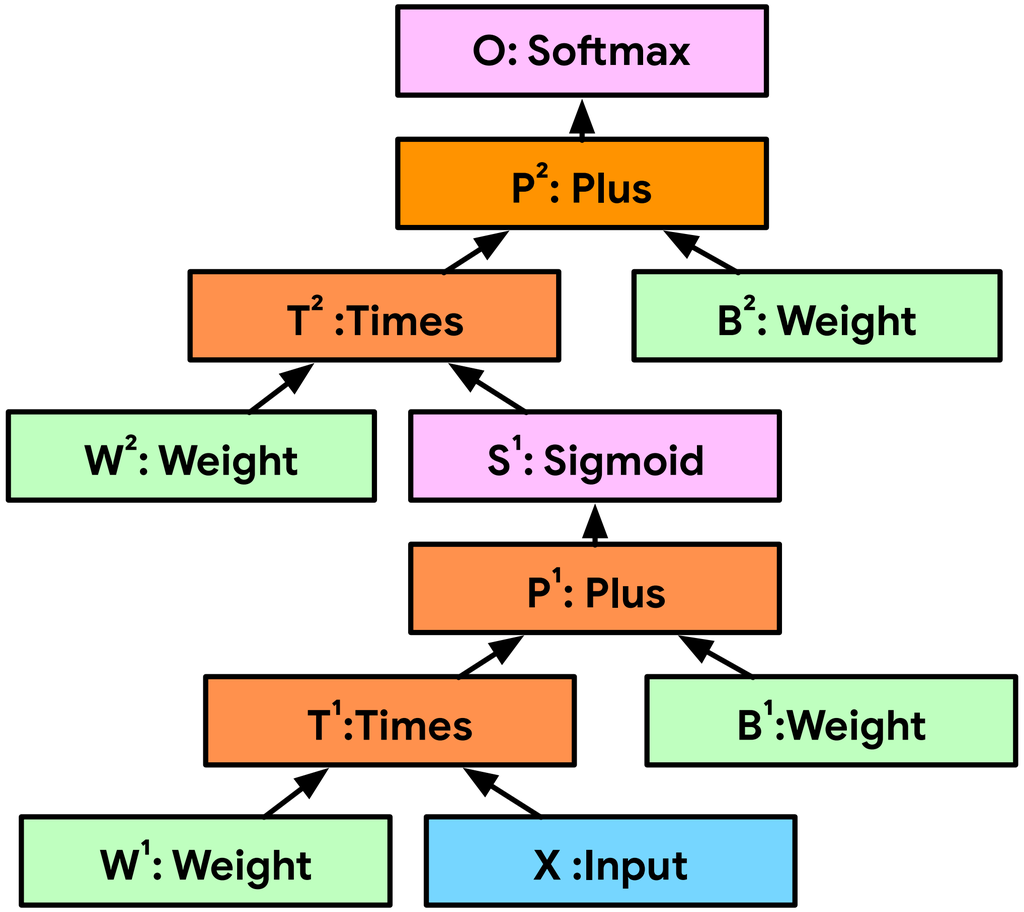


Sun's Phenomenological Levels

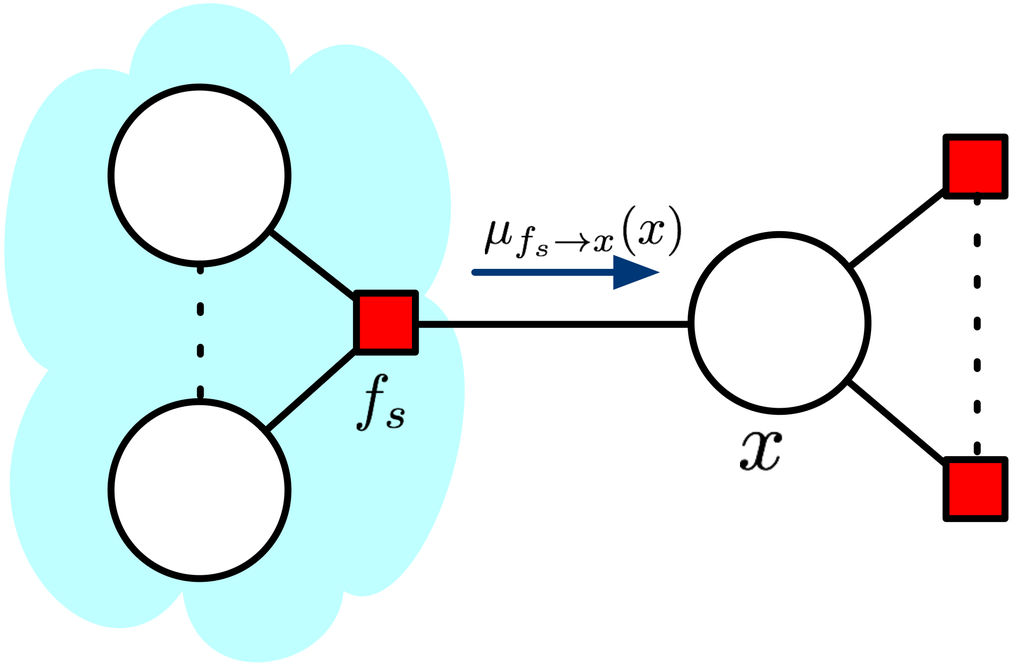
For the ML Core:
Probabilistic and pragmatic in approach



Architecture-Loss



1. Computational Graphs



2. Error propagation

Model-Inference-Algorithm

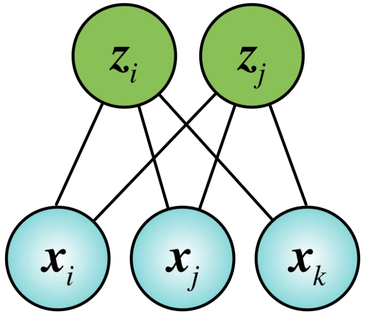
3. Algorithms

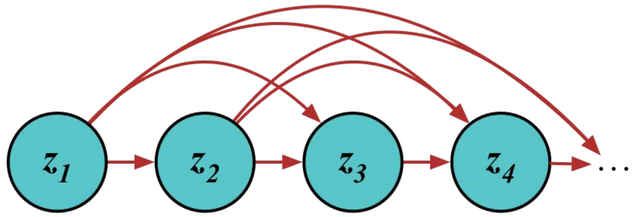
1. Models

2. Learning Principles

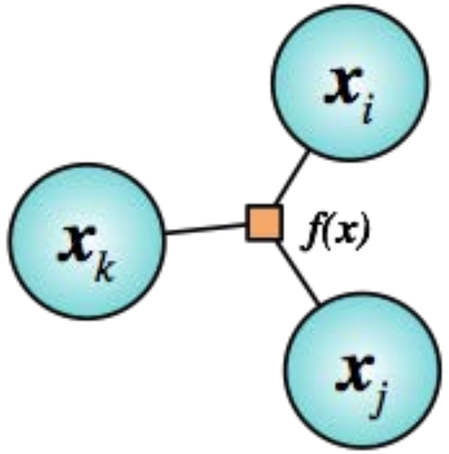
Models

Directed and Undirected

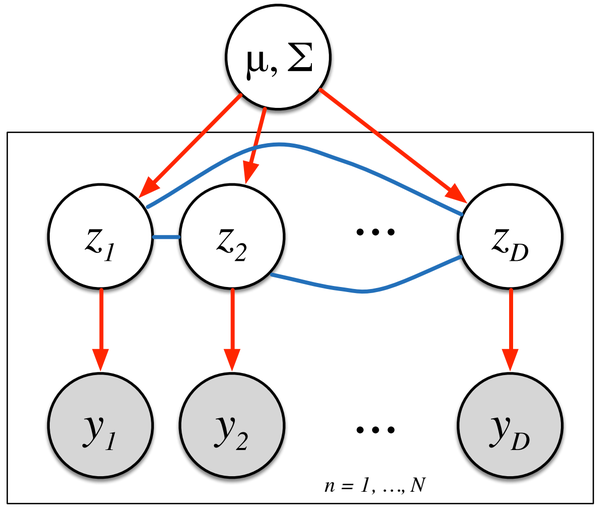




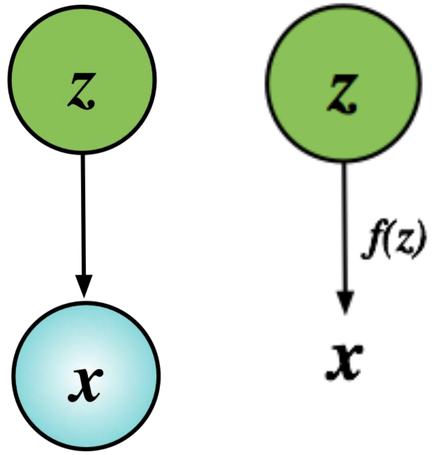
Fully-observed



Parametric, Non-parametric And semi-parametric



Latent Variable



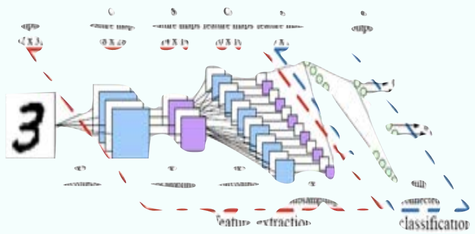
Learning Principles

- **Statistical Inference**
  - **Direct**
    - Laplace approximation
    - Maximum a posteriori
    - Cavity Methods
    - Expectation Maximisation
    - Noise Contrastive
    - Maximum Likelihood
    - Variational Inference
    - Integr. Nested Laplace Approx
    - Markov chain Monte Carlo
    - Sequential Monte Carlo
  - **Indirect**
    - Two Sample Comparison
    - Approx Bayesian Computation
    - Max Mean Discrepency
    - Method of Moments
    - Transportation methods

Algorithms

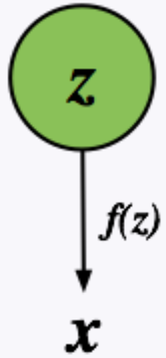
A given model and learning principle can be implemented in many ways.

Convolutional neural network + penalised maximum likelihood



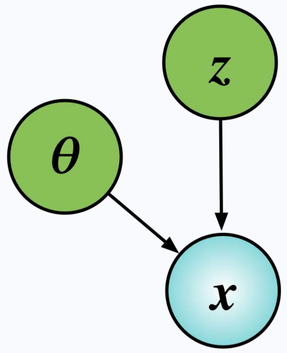
- Optimisation methods (SGD, Adagrad)
- Regularisation (L1, L2, batchnorm, dropout)

Implicit Generative Model + Two-sample testing



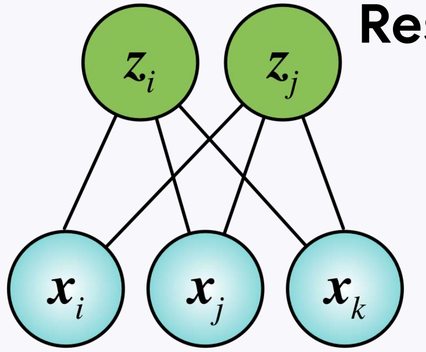
- Unsupervised-as-supervised learning
- Approximate Bayesian Computation (ABC)
- Generative adversarial network (GAN)

Latent variable model + variational inference



- VEM algorithm
- Expectation propagation
- Approximate message passing
- Variational auto-encoders (VAE)

Restricted Boltzmann Machine + maximum likelihood



- Contrastive Divergence
- Persistent CD
- Parallel Tempering
- Natural gradients

Likelihood Functions

Probabilistic Model

$$p(y|\mathbf{x}) = p(y|h(\mathbf{x});\boldsymbol{\theta})$$

Likelihood function

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_n \log p(y_n|\mathbf{x}_n;\boldsymbol{\theta})$$

Likelihood of parameters

Prescribed Likelihoods

Efficient Estimators
- Statistically efficient (Cramer-Rao lower bound)
- Asymptotically unbiased, consistent
- Maximum entropy (principle of indifference)

Tests with Good Power
- Likelihood ratio tests
- Can construct small confidence regions

Widely-applicable
- Handle data that is incompletely observed, distorted, samples with bias
- Can offset or correct these issues.

Pool Information
- Combine different data sources
- Knowledge outside the data can be used, like constraints on domain or prior probabilities.

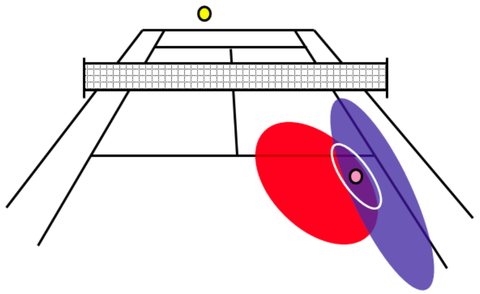
Misspecification: Inefficient estimates; or confidence intervals/tests can fail completely.

Bayesian Analysis

Bayesian approach follows the idea that all components of a model should be probabilistic and be described by probability distributions

Bayes' Theorem

$$\underbrace{p(z|y)}_{\text{Posterior}} = \frac{\overbrace{p(y|z)}^{\text{Likelihood}}\ \overbrace{p(z)}^{\text{Prior}}}{\underbrace{\int p(y,z)dz}_{\text{Marginal likelihood/ Model evidence}}}$$



Bayesian analysis is an approach to modelling that follows:

- Decide on a priori beliefs.
- Posit an explanation of how the observed data is generated, i.e. provide a probabilistic description.

Rule for inverse probabilities.
Go from prior states of knowledge to new states based on evidence.

Bayesian Analysis

Interested in reasoning about two important quantities

Evidence

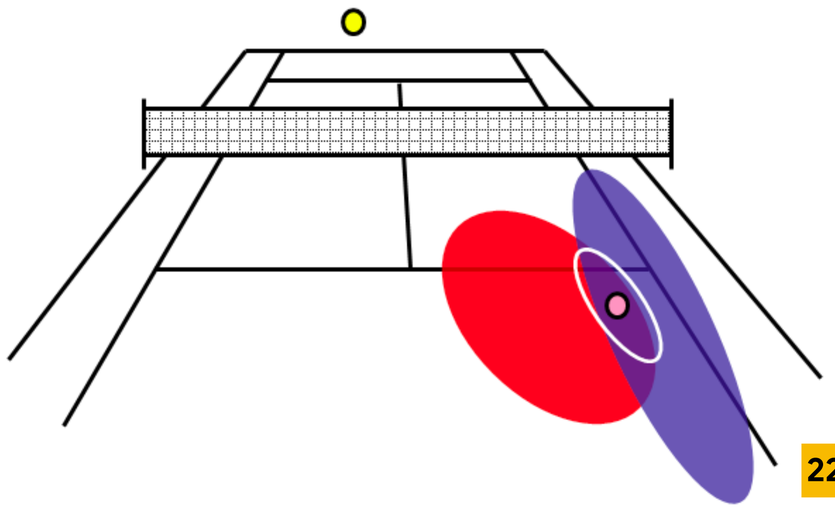
$$p(y|\mathbf{x}) = \int p(y|h(\mathbf{x});\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$$

Posterior

$$p(\boldsymbol{\theta}|y,\mathbf{x}) \propto p(y|h(\mathbf{x});\boldsymbol{\theta})p(\boldsymbol{\theta})$$

- In Bayesian analysis, things that are *not. observed* must be integrated over - averaged out.
- This makes computation difficult.
- Integration is the central operation.

Intractable Integrals: Will often see this phrasing.
- Don't know the integral in closed form
- Very high-dimensional quantities and can't compute (e.g., using quadrature)



Learning and Inference

Statistics, no distinction between learning and inference - only inference (or **estimation**).

Bayesian statistics, all quantities are probability distributions, so there is only the problem of **inference**.

Machine learning makes a distinction between **inference and learning**:

- **Inference**: reason about (and compute) unknown probability distributions.
- (Parameter) **Learning** is finding point estimates of quantities in the model.

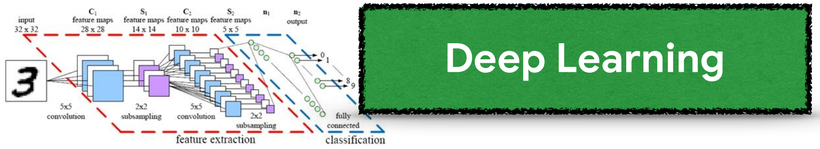
Software engineering, **inference** is the forward evaluation of a trained model (to get predictions).

Decision making and AI, refer to **learning** in general as the means of understanding and acting based on past experience (data).

Break

10min

Two Streams of Machine Learning



Deep Learning

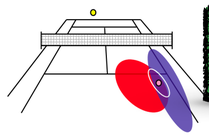
+ Rich non-linear models for classification and sequence prediction.

+ Scalable learning using stochastic approximation and conceptually simple.

+ Easily composable with other gradient-based methods

- Only point estimates

- Hard to score models, do selection and complexity penalisation.



Bayesian Reasoning

- Mainly conjugate and linear models

- Potentially intractable inference, computationally expensive or long simulation time.

+ Unified framework for model building, inference, prediction and decision making

+ Explicit accounting for uncertainty and variability of outcomes

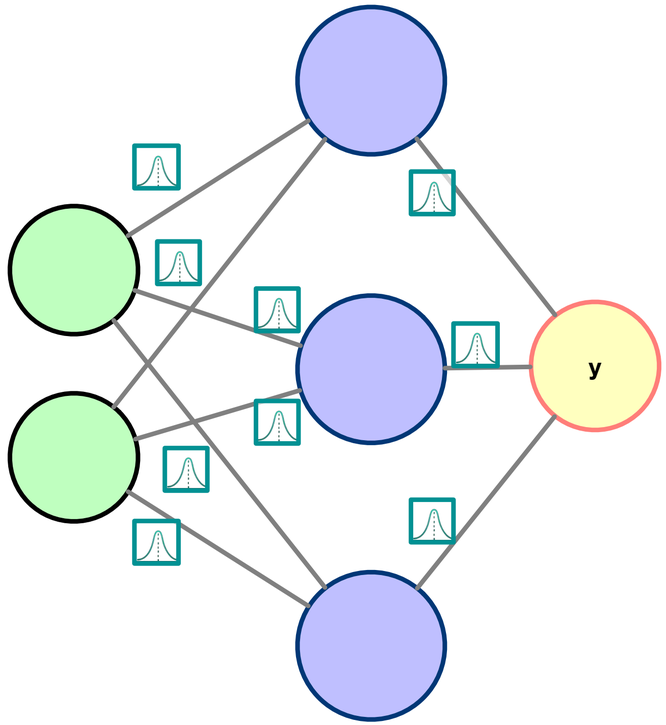
+ Robust to overfitting; tools for model selection and composition.

Natural to consider the marriage of these approaches: Bayesian Deep Learning

Regression and Classification

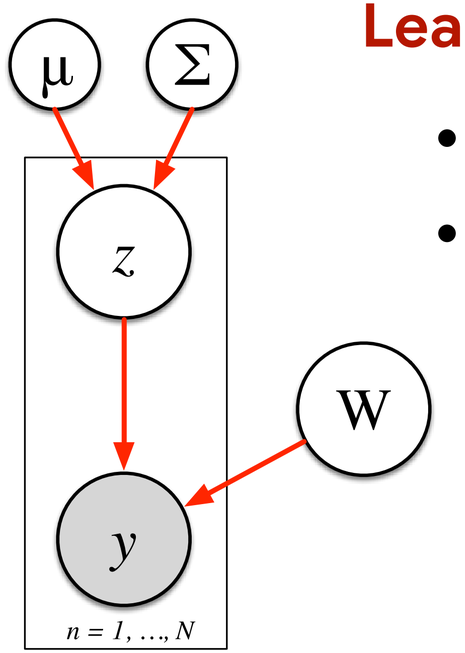
Probabilistic models over functions

Prior	$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}\|0, \mathbf{I})$
Observation model	$p(y\|\mathbf{x}, \boldsymbol{\theta}) = Categorical(\pi(\mathbf{x}; \boldsymbol{\theta}))$
Posterior	$p(\boldsymbol{\theta}\|y, \mathbf{x})$



- Make predictions of future based on past correlations.
- Ways of learning distributions over functions and maintaining uncertainty over functions.
- Many ways to learn the posterior distribution.

Density Estimation



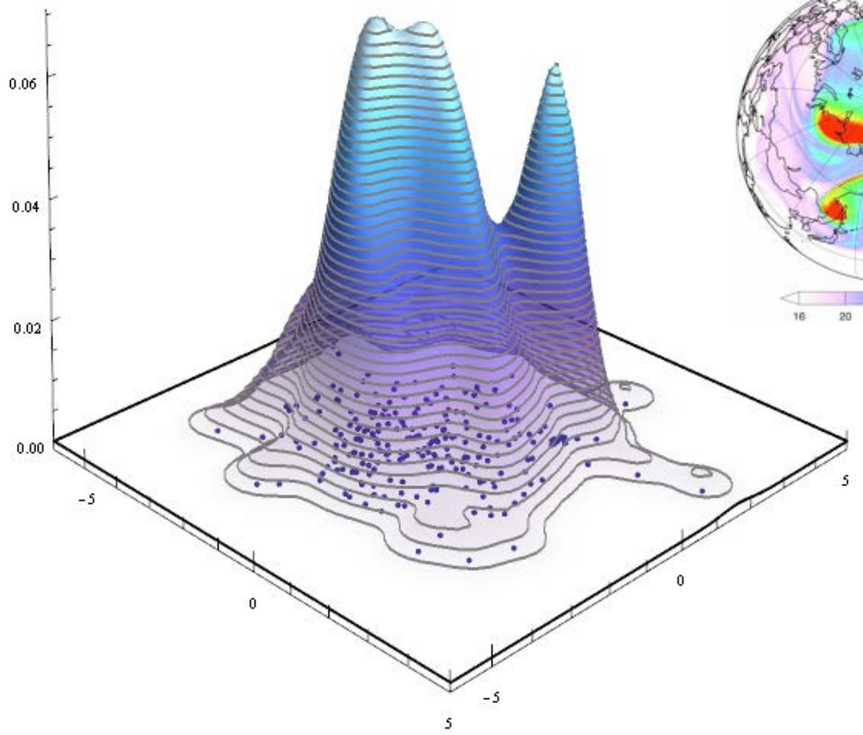
Factor Analysis / PCA

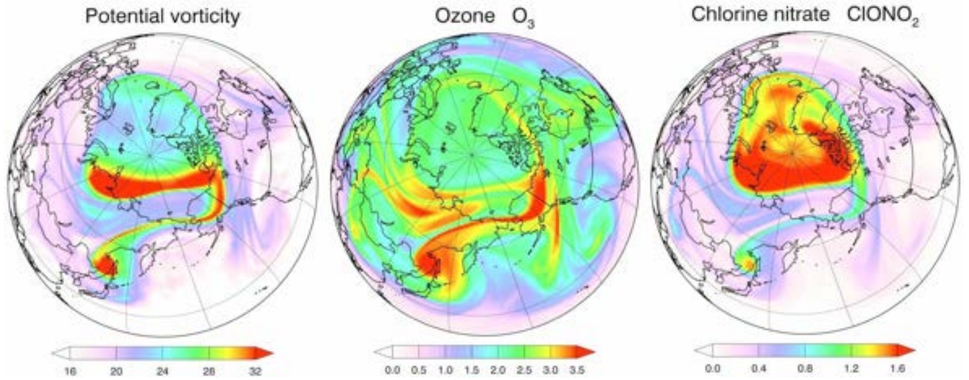
$$z \sim \mathcal{N}(z|\mu, \Sigma)$$

$$y \sim \mathcal{N}(y|Wz, \sigma_y^2 I)$$

Learn probability distributions over the data itself

- How can you learn from data without any labels. Structure of the data.
- Deep Generative Models and Unsupervised learning.





NO LABELS

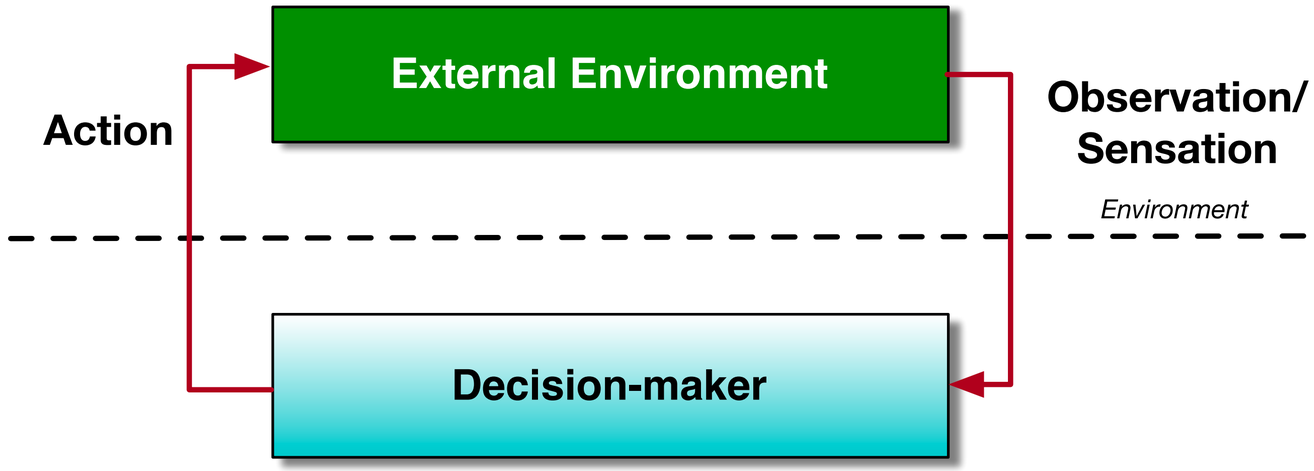
Decision-making

Probabilistic models of environments and actions

Prior over actions $a \sim p(a)$

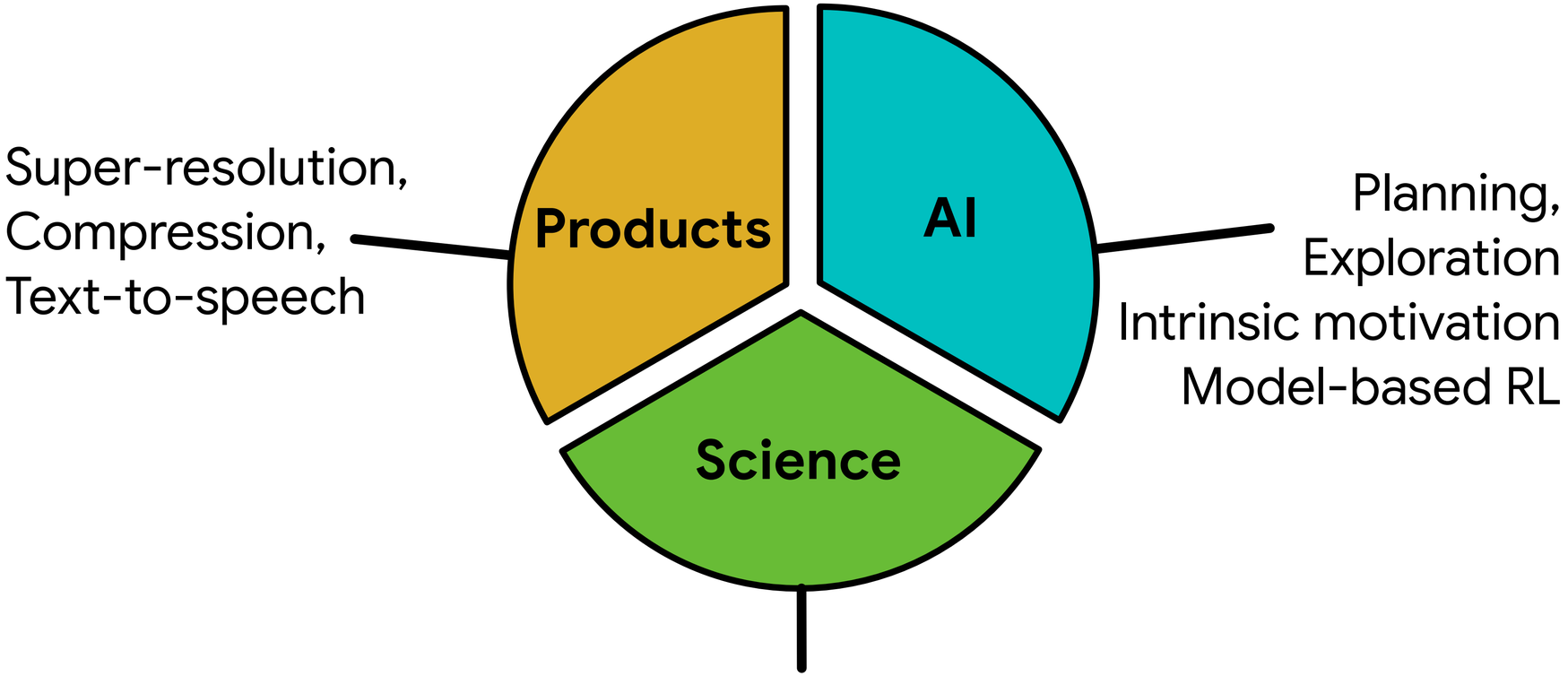
Interaction only $u(s,a) \sim \text{Environment}(a)$

Reward/Utility $p(R(s)|a) \propto \exp(u(s,a))$

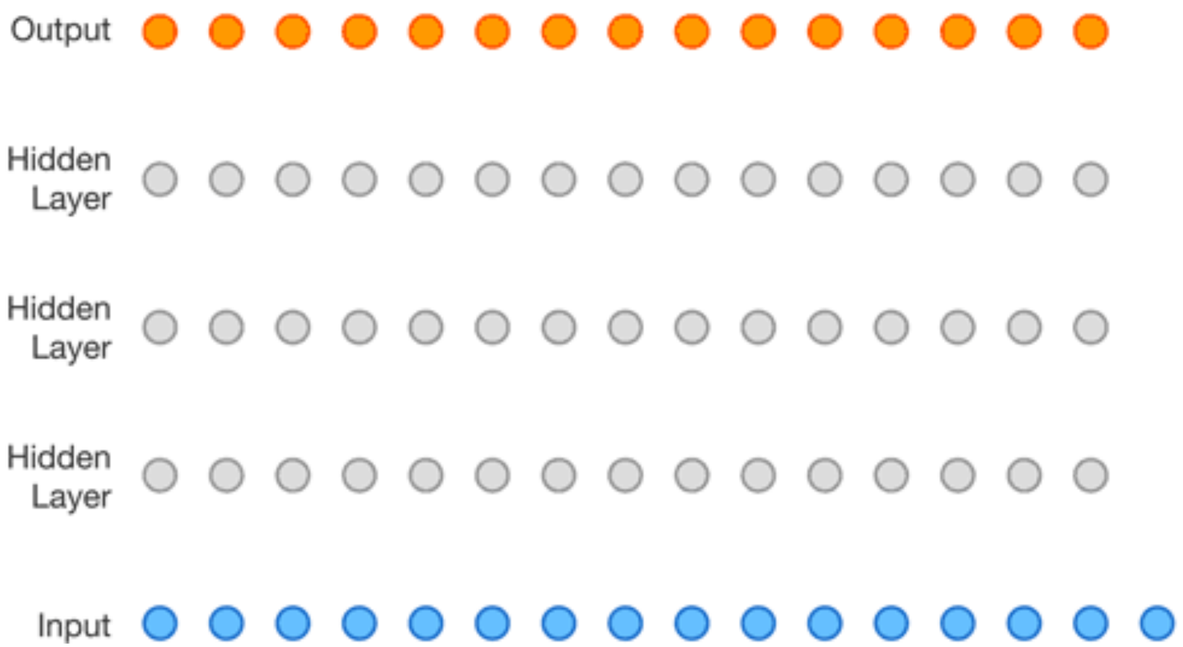


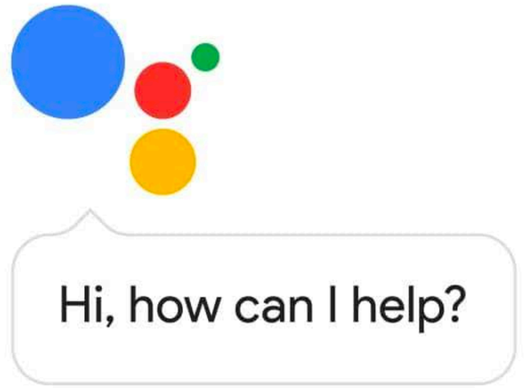
Setup is common in experimental design, causal learning, reinforcement learning.

Applications



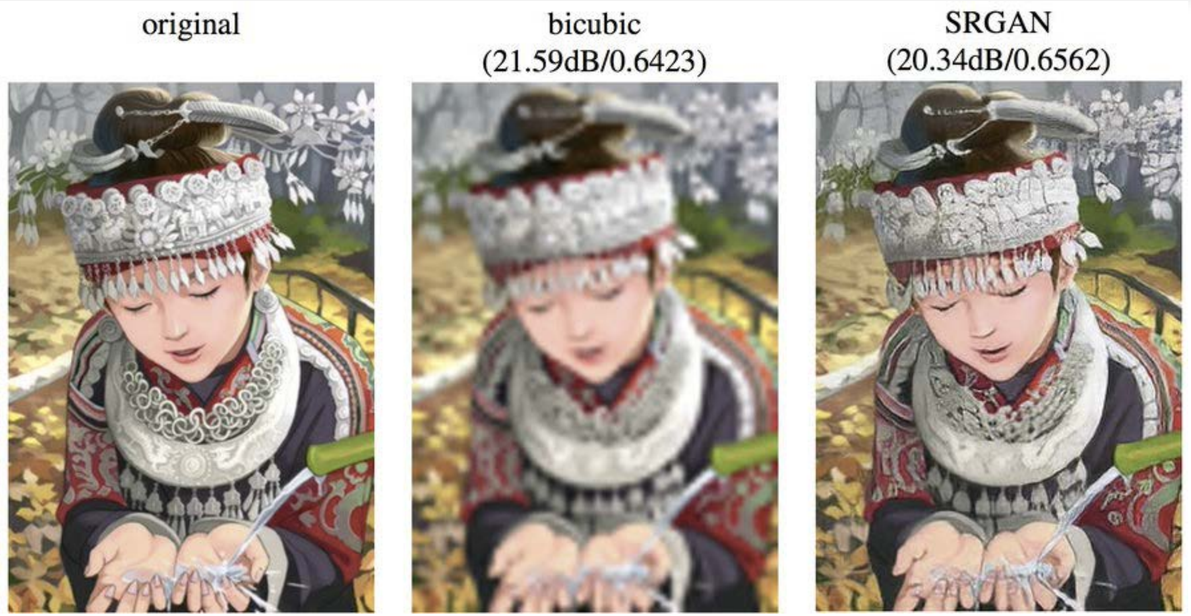
Assistive Technologies





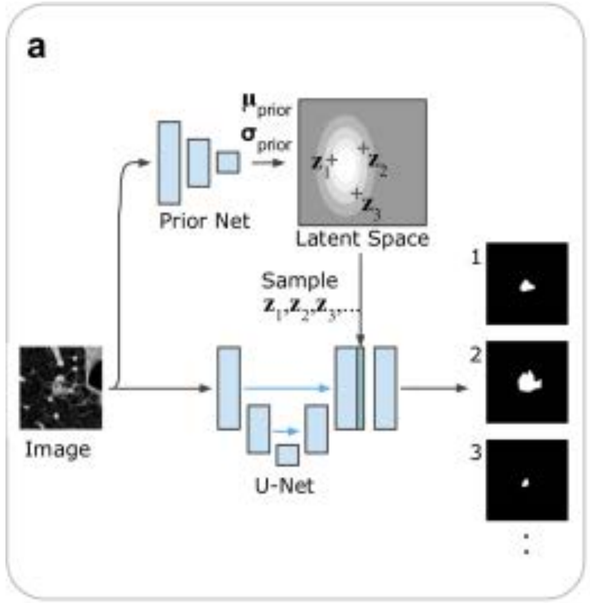
Fully-observed conditional generative model

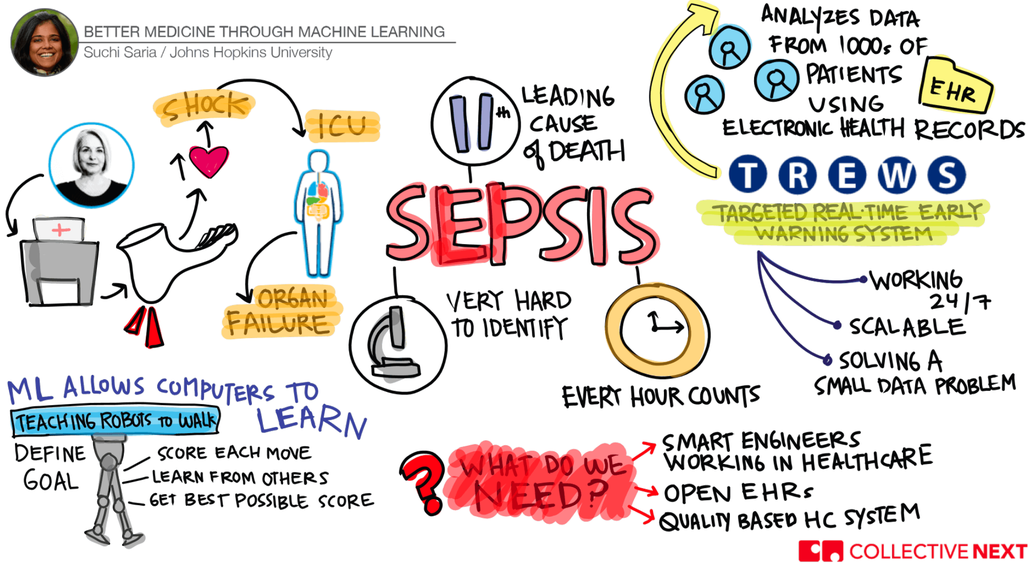
Compression-Communication

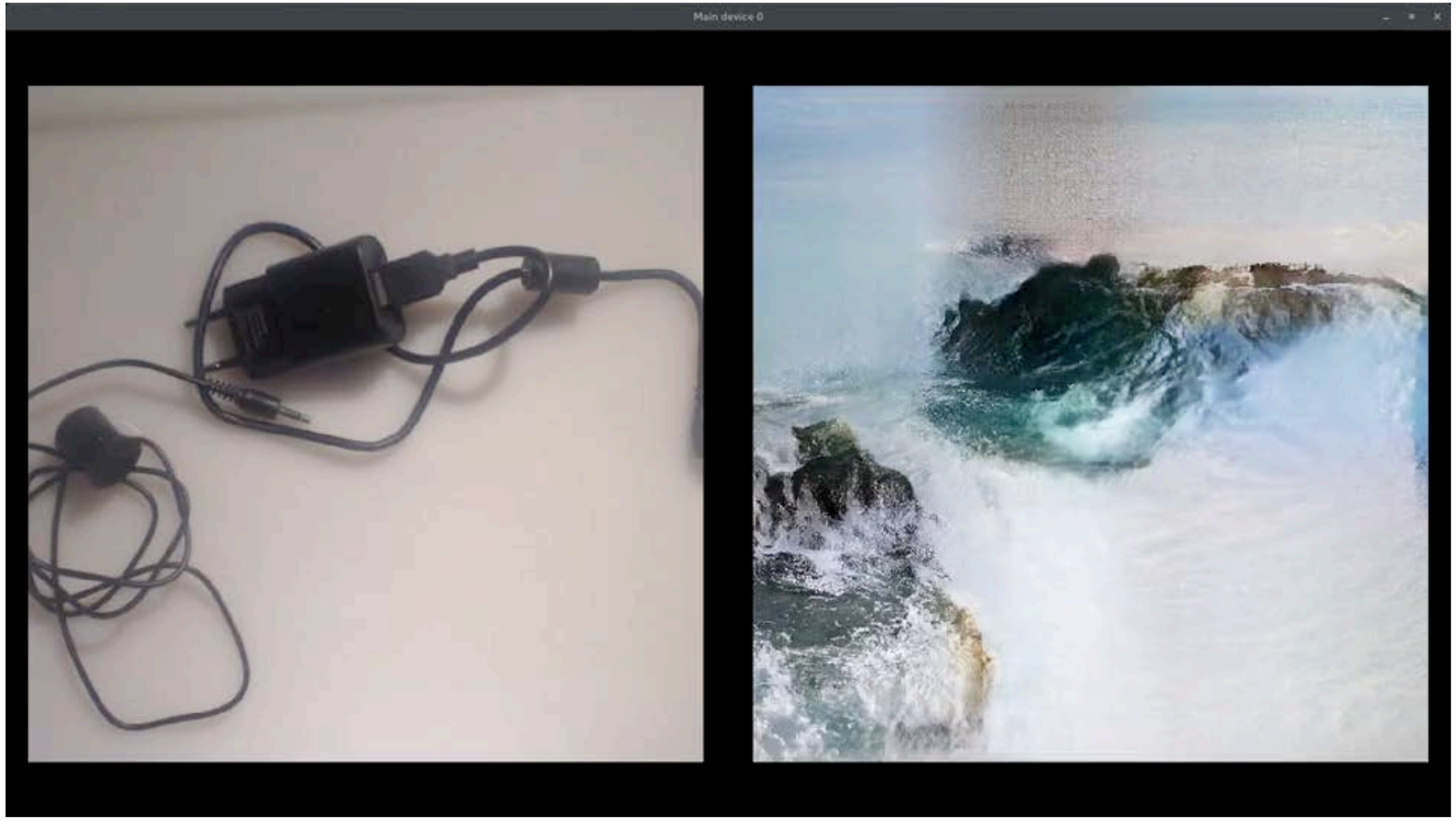


Compression rate:
0.2bits/dimension

Advancing Healthcare

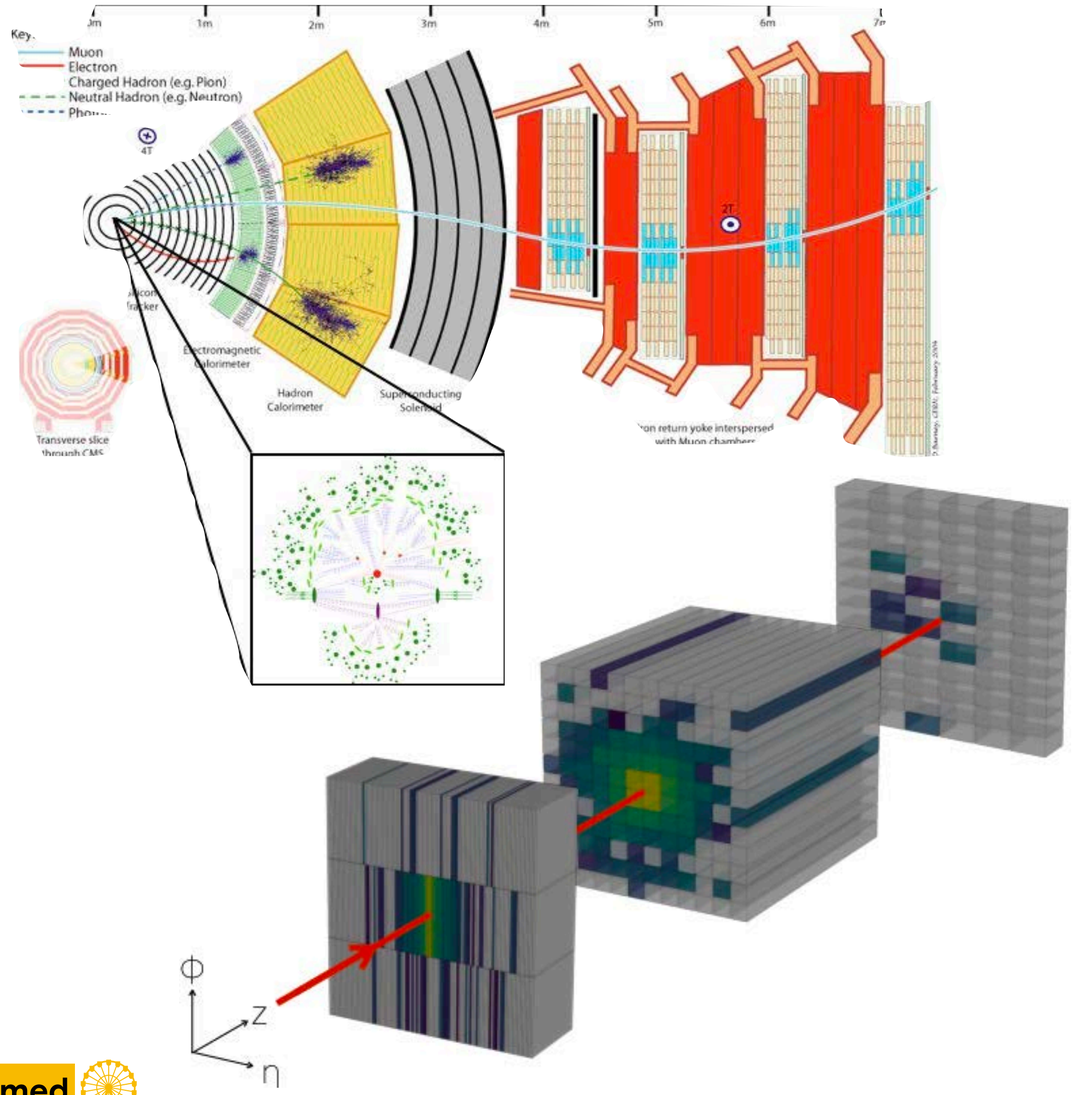


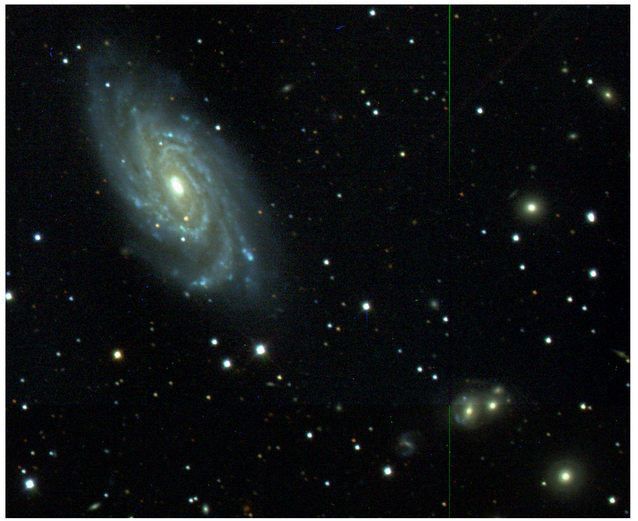


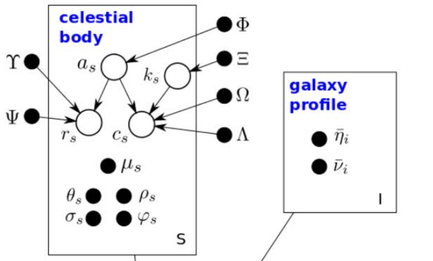


Video from work of Memo Aktem

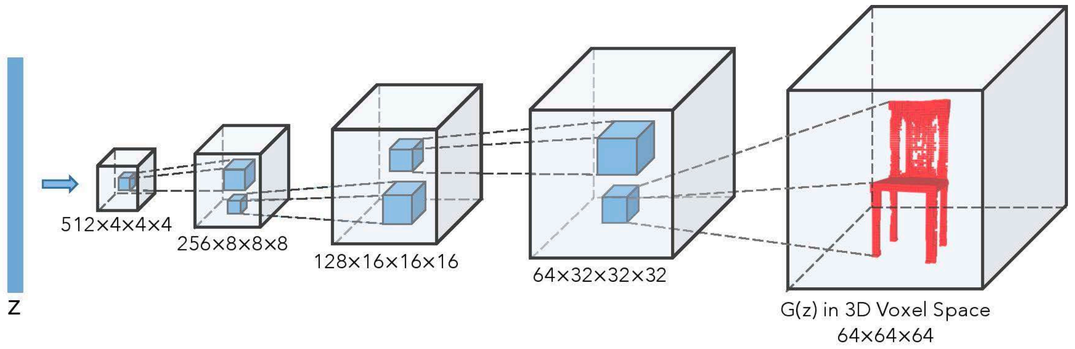
Advancing Science

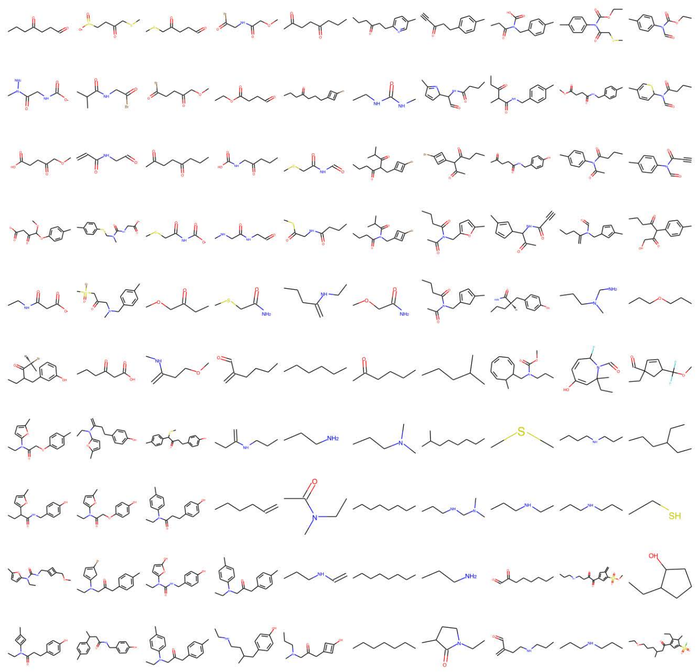




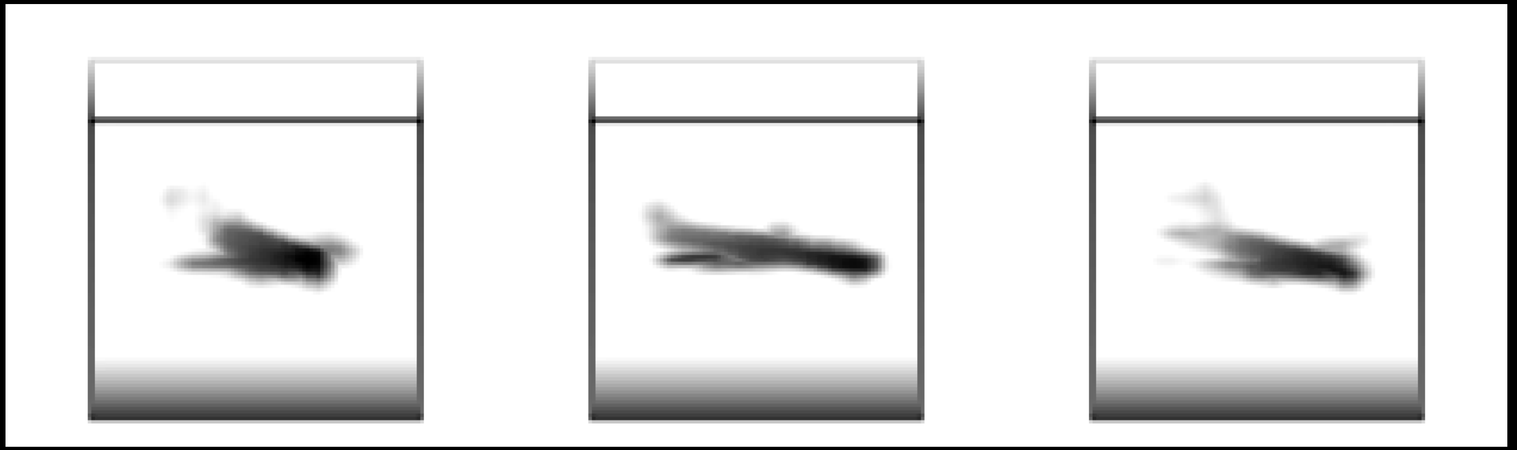


Generative Design

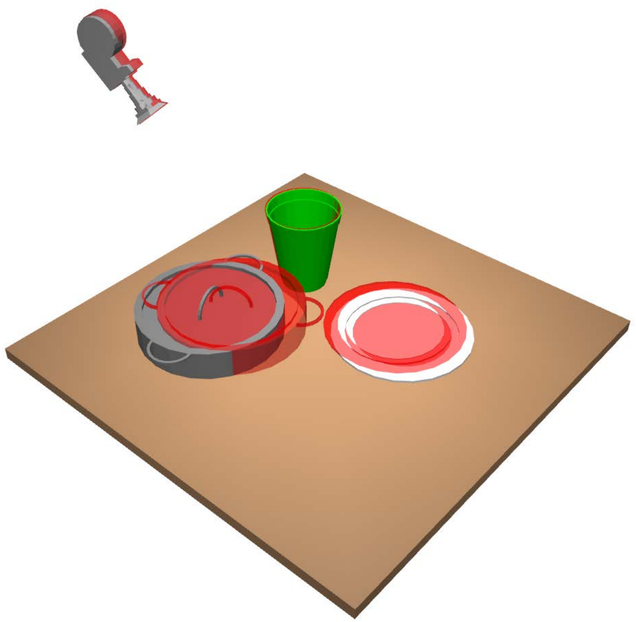


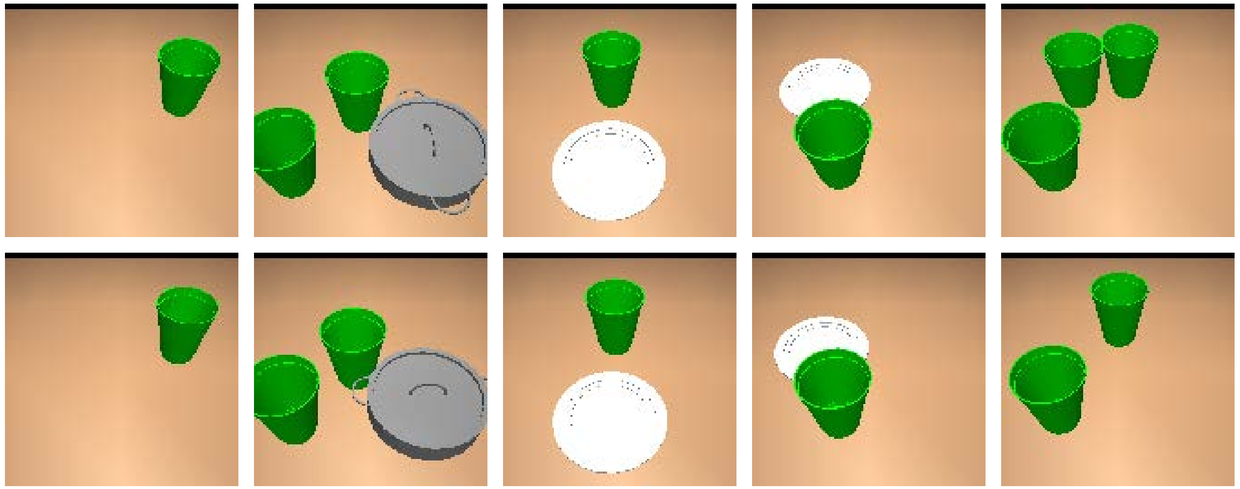


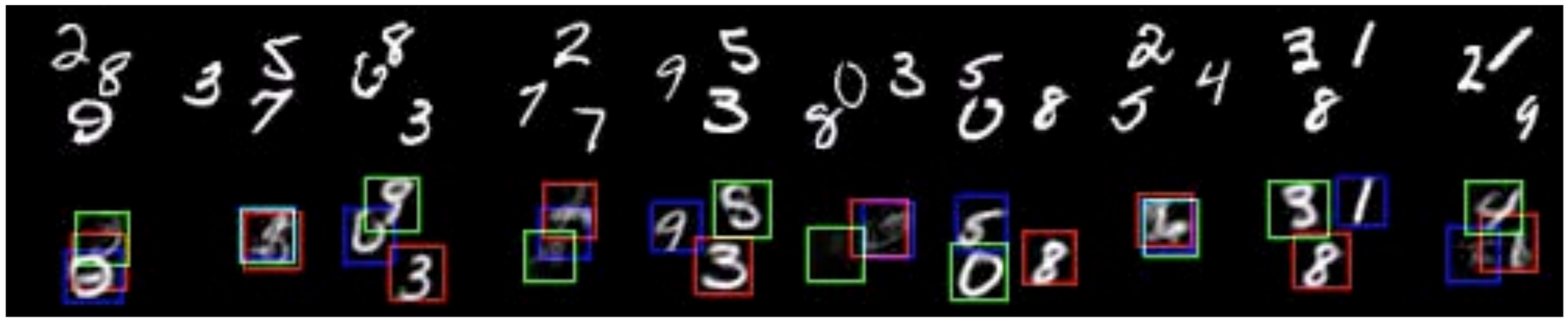
3D Scene Generation



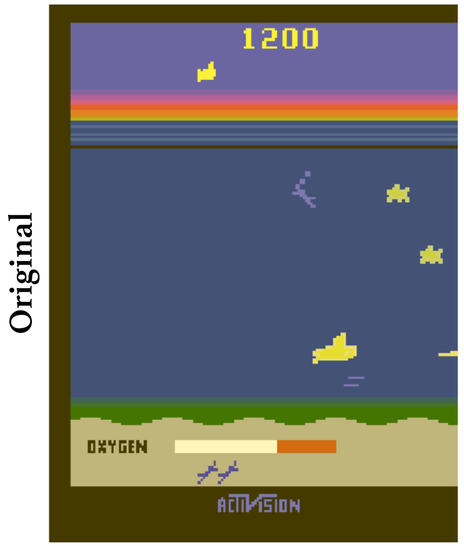
Rapid Scene Understanding





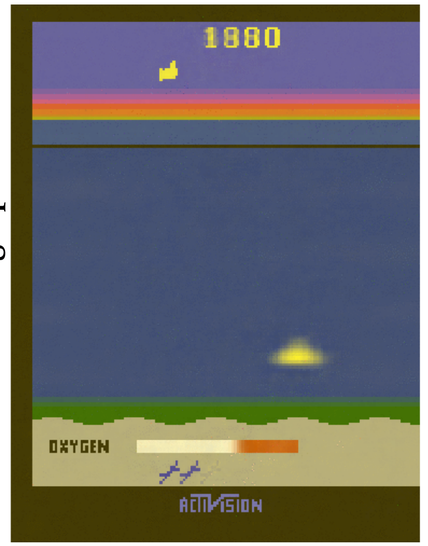


Visual Concept Learning

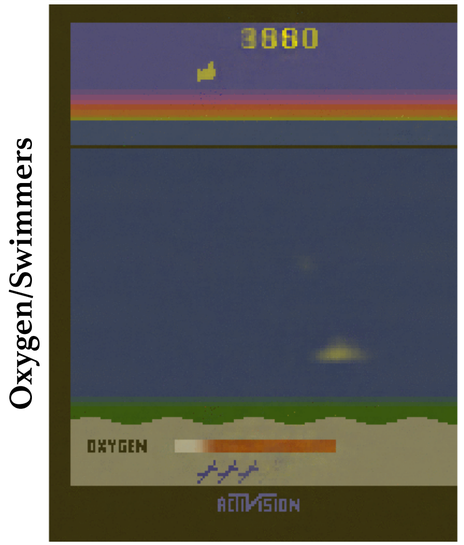


Original

Score



Moving Up



Oxygen/Swimmers

Score/Lives

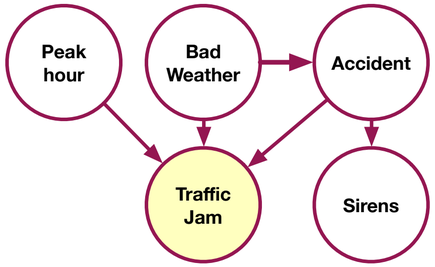
Moving Left

Progress in Generative Models

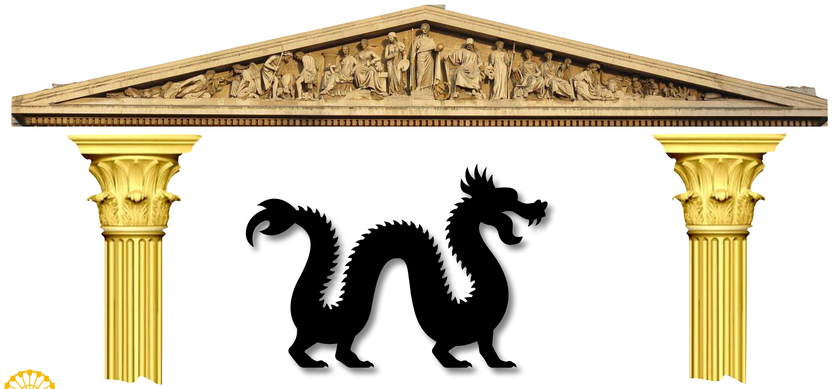
Summary of Part I

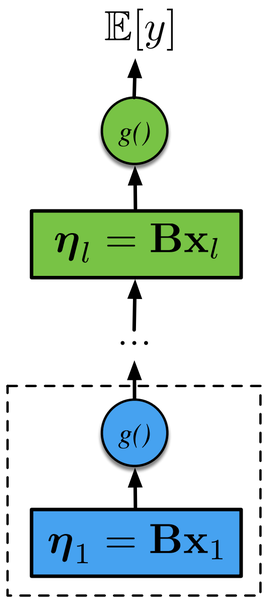
Subjective Probability
Probability as a degree of belief

Probabilistic descriptions of systems and data

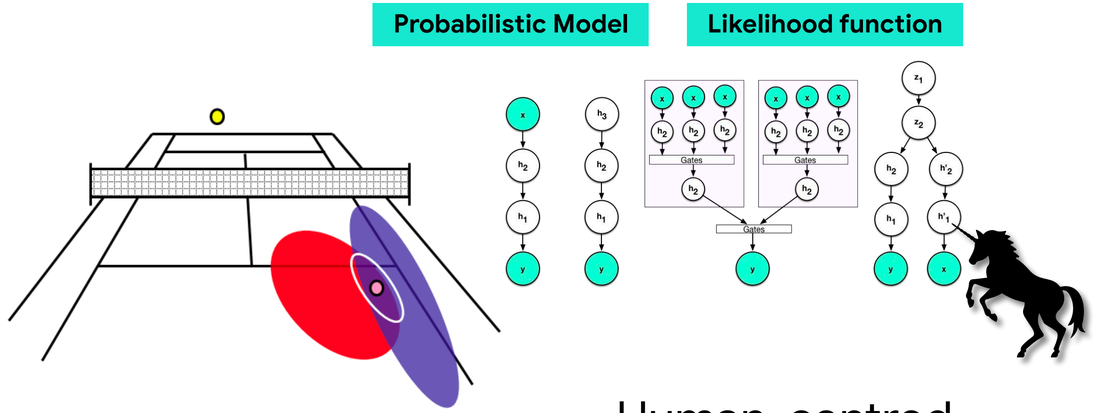


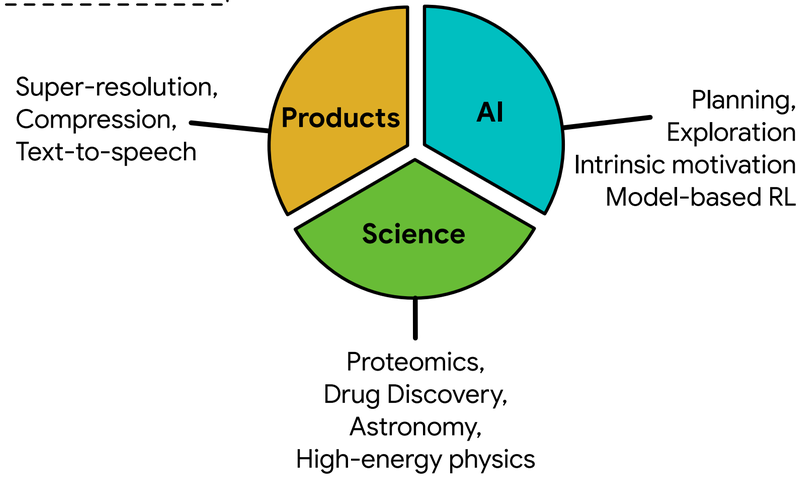
Model-Inference-Algorithm



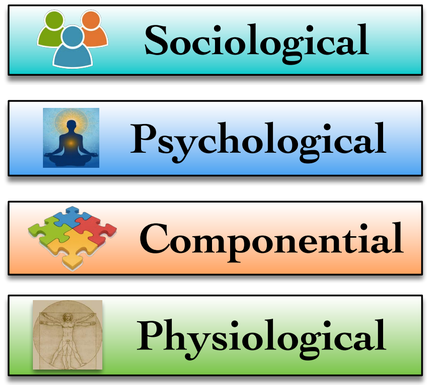


Deep Learning, Estimation theory, hierarchical models, Bayesian analysis



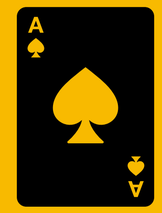


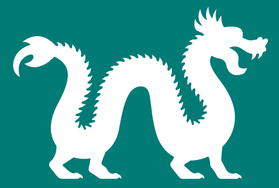
Human-centred



Sun's Phenomenological Levels

Part 2
Manipulating Probabilities





1. Develop tools to manipulate distributions by studying 6 probability questions.

2. Build connections between concepts in machine learning and those in other computational sciences.

Inferential Questions

Evidence Estimation

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}$$

Moment Computation

$$\mathbb{E}[f(\mathbf{z})|\mathbf{x}] = \int f(\mathbf{z}) p(\mathbf{z}|\mathbf{x}) d\mathbf{z}$$

Parameter Estimation

$$p(\boldsymbol{\theta}|\mathbf{x}_{0:N})$$

Prediction

$$p(\mathbf{x}_{t+1}|\mathbf{x}_{0:t})$$

Planning

$$\mathcal{J} = \mathbb{E}_p \left[\int_0^\infty C(\mathbf{x}_t) dt | \mathbf{x}_0, \mathbf{u} \right]$$

Hypothesis Testing

$$\mathcal{B} = \log p(\mathbf{x}|H_1) - \log p(\mathbf{x}|H_2)$$

Experimental Design

$$\mathcal{IG} = D[p(\mathbf{x}_{t:T}|u) \| p(\mathbf{x}_{0:t})]$$

Identity Trick

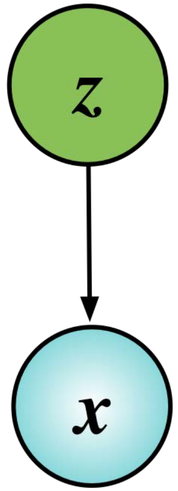
Transform an expectation w.r.t. distribution *p*,
into an expectation w.r.t. distribution *q*.

$$\int p(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} = \mathbb{E}_{p(\mathbf{x})}[f(\mathbf{x})]$$

$$\mathbb{E}_{q(\mathbf{x})}[g(\mathbf{x}; f)] = \int q(\mathbf{x}) g(\mathbf{x}, f) d\mathbf{x}$$

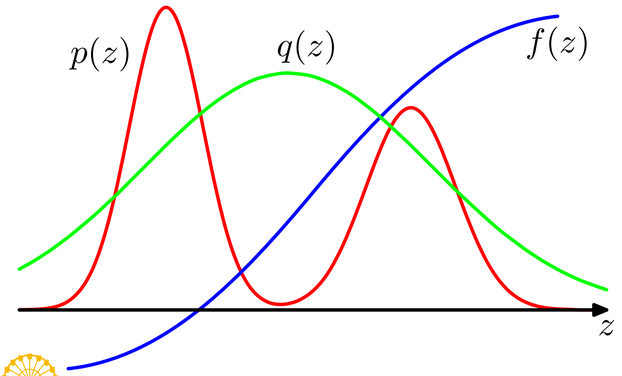
Do this by introducing a **probabilistic one** $\frac{p(\mathbf{x})}{p(\mathbf{x})}$

Identity Trick



Conditions

- $q(z)>0$, when $p(x|z)p(z) \neq 0$.
- *$q(z)$ is known/easy to handle.*



Integral problem

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

Probabilistic one

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\frac{q(\mathbf{z})}{q(\mathbf{z})}d\mathbf{z}$$

Re-group/re-weight

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z})}q(\mathbf{z})d\mathbf{z}$$

$$p(\mathbf{x}) = \mathbb{E}_{q(\mathbf{z})}\left[p(\mathbf{x}|\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z})}\right]$$

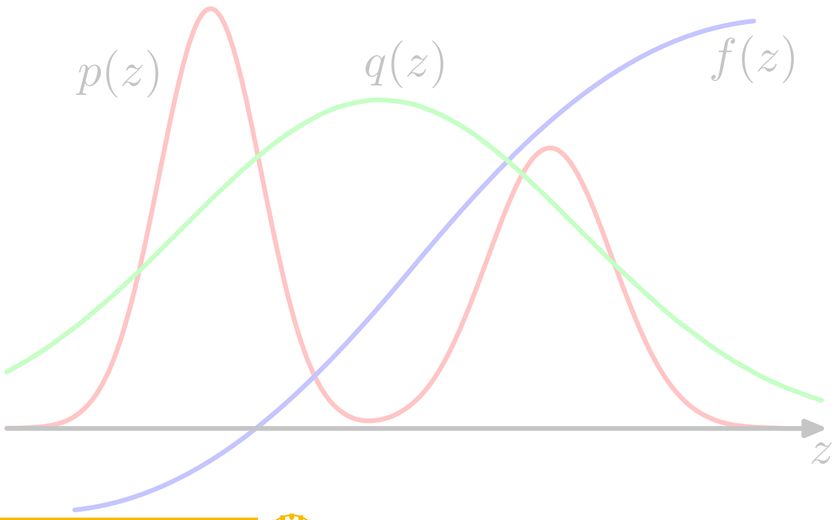
Importance Sampling

$$p(\mathbf{x}) = \mathbb{E}_{q(\mathbf{z})}\left[p(\mathbf{x}|\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z})}\right]$$

Monte Carlo Estimator

$$p(\mathbf{x}) = \frac{1}{S}\sum_{s} w^{(s)} p(\mathbf{x}|\mathbf{z}^{(s)})$$

$$w^{(s)} = \frac{p(z)}{q(z)} \quad z^{(s)} \sim q(z)$$



Identity Trick Elsewhere

- Manipulate stochastic gradients
- Derive probability bounds
- RL for policy corrections

Probability Flow Tricks

Distribution and sample

$$\hat{x} \sim p(x)$$

Transformation

$$\hat{y} = g(\hat{x}; \theta)$$

Unconscious Statistician

$$\mathbb{E}_{p(x)}[g(x;\theta)]$$

Change of Variables

$$p(\mathbf{y}) = p(\mathbf{x}) \left| \frac{dg}{d\mathbf{x}} \right|^{-1}$$

Begin with a diagonal Gaussian and improve by change of variables.

Makes entropy computation and backpropagation easy.

Triangular Jacobians allow for computational efficiency.

Compute

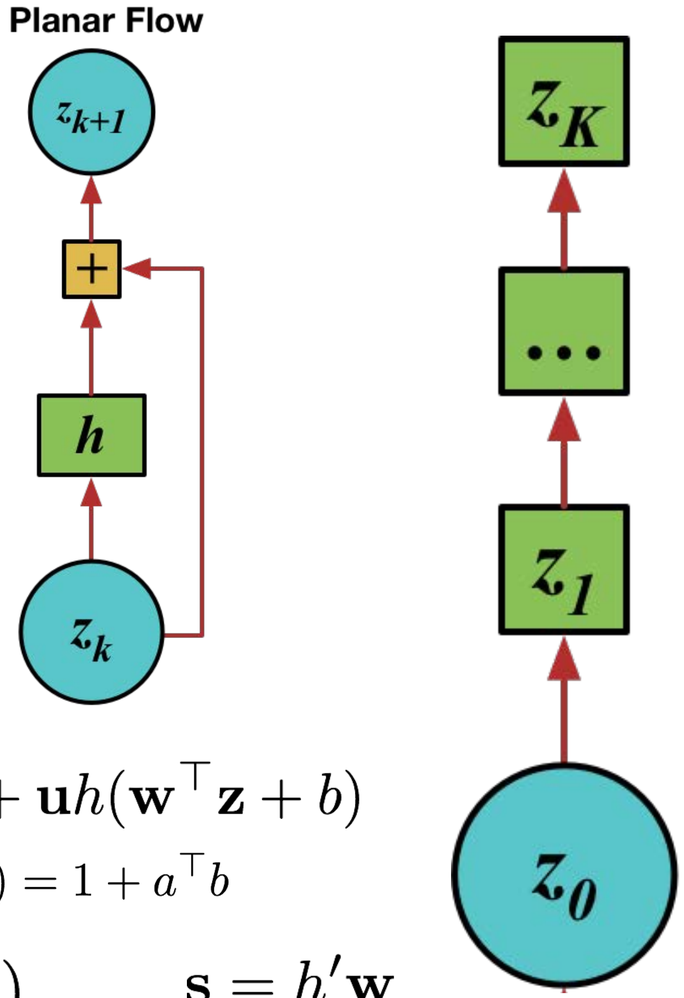
$$\log \det \left(\frac{\partial f(\mathbf{z})}{\partial \mathbf{z}} \right)$$

$$f(\mathbf{z}) = \mathbf{z} + \mathbf{u} h(\mathbf{w}^\top \mathbf{z} + b)$$

$$\det(I + ab^\top) = 1 + a^\top b$$

$$\det(\mathbf{I} + \mathbf{u}\mathbf{s}^\top) = (1 + \mathbf{u}^\top \mathbf{s}) \qquad \mathbf{s} = h'\mathbf{w}$$

Linear time computation of the determinant and its gradient.



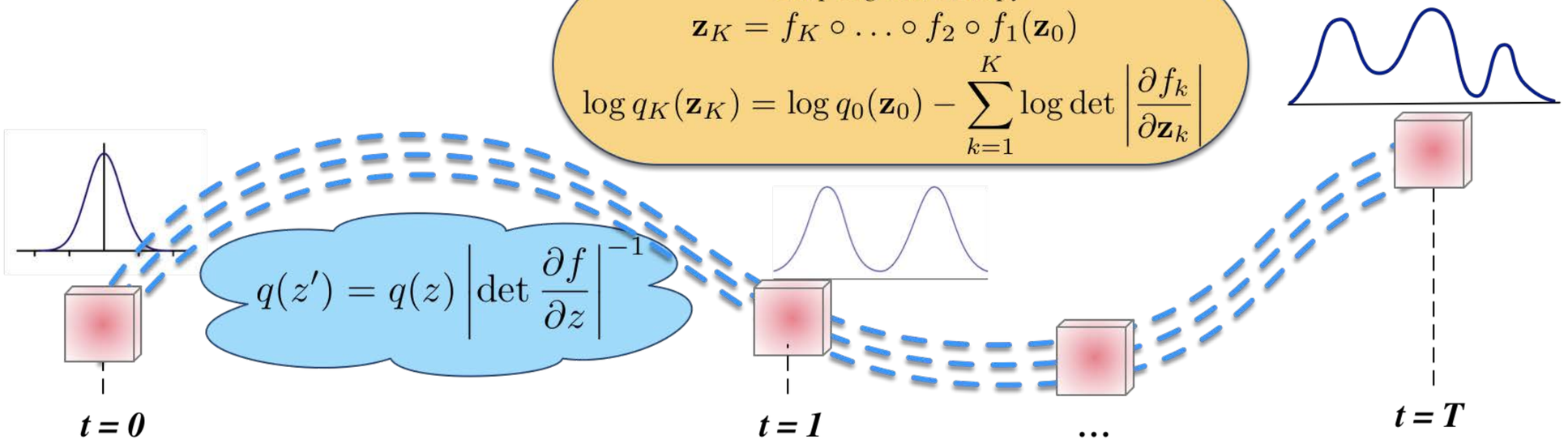
Normalising Flows

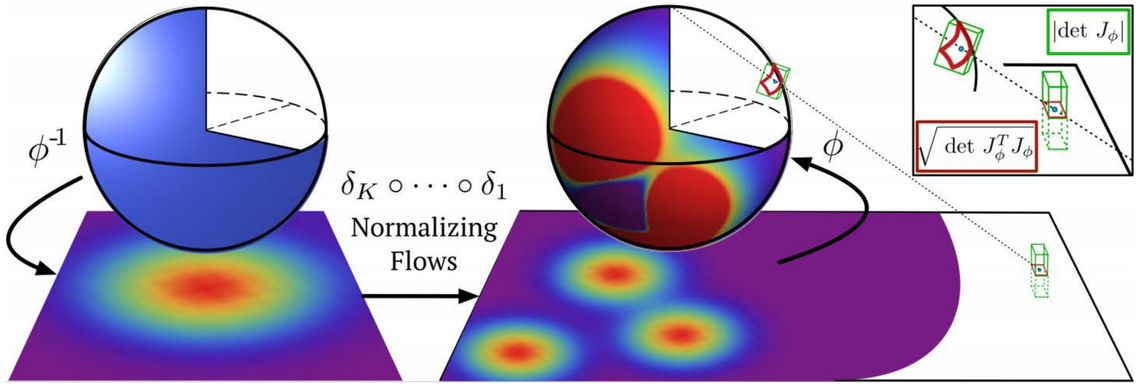
Sampling and Entropy

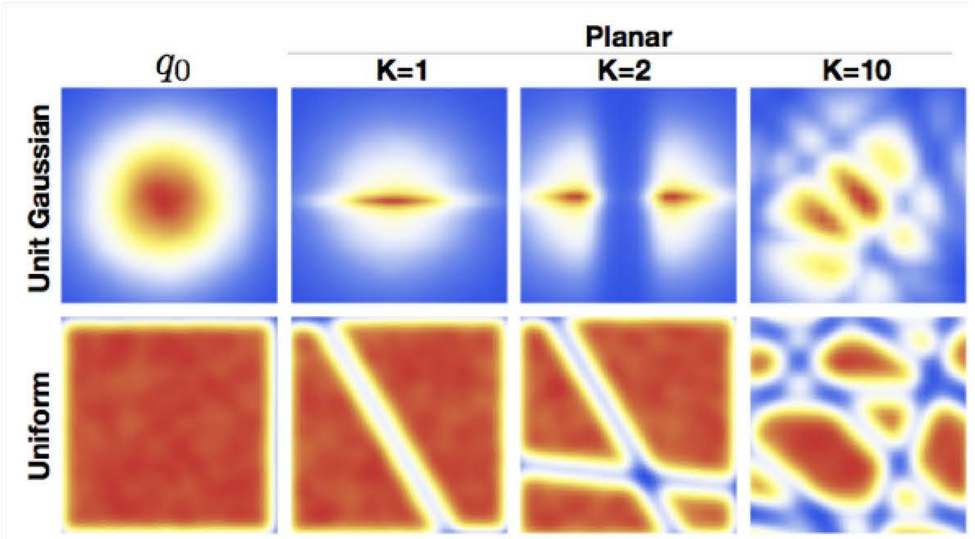
$$\mathbf{z}_K = f_K \circ \ldots \circ f_2 \circ f_1(\mathbf{z}_0)$$

$$\log q_K(\mathbf{z}_K) = \log q_0(\mathbf{z}_0) - \sum_{k=1}^{K} \log \det \left| \frac{\partial f_k}{\partial \mathbf{z}_k} \right|$$

$$q(z') = q(z) \left| \det \frac{\partial f}{\partial z} \right|^{-1}$$







Stochastic Optimisation

Common gradient problem

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})}[f_\theta(\mathbf{z})] = \nabla \int q_\phi(\mathbf{z}) f_\theta(\mathbf{z}) d\mathbf{z}$$

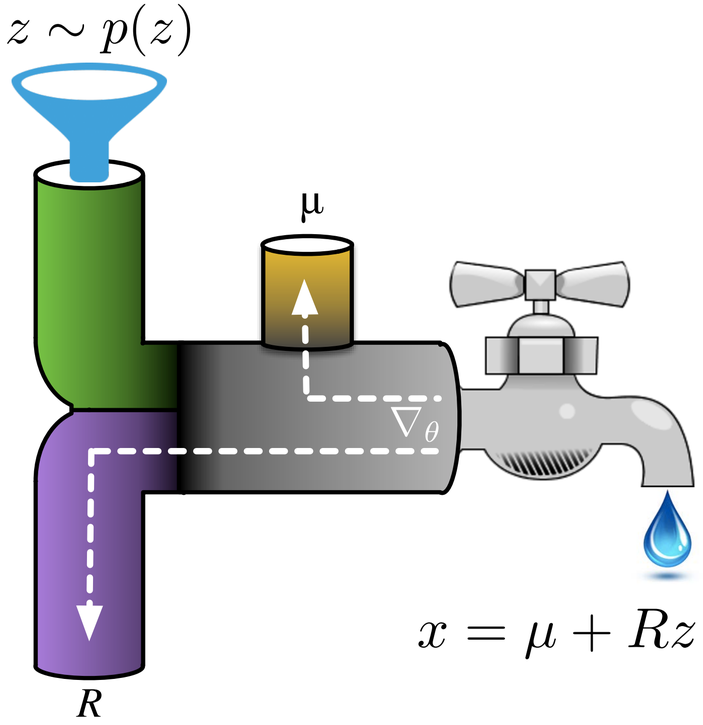
1. **Pathwise estimator**: Differentiate the function **f**(**z**)
2. **Score-function estimator**: Differentiate the density **q**(**z**|**x**)

- Don't know this expectation in general.
- Gradient is of the parameters of the distribution w.r.t. which the expectation is taken.

Typical problem areas

- Sensitivity analysis
- Generative models and inference
- Reinforcement learning and control
- Operations research and inventory control
- Monte Carlo simulation
- Finance and asset pricing

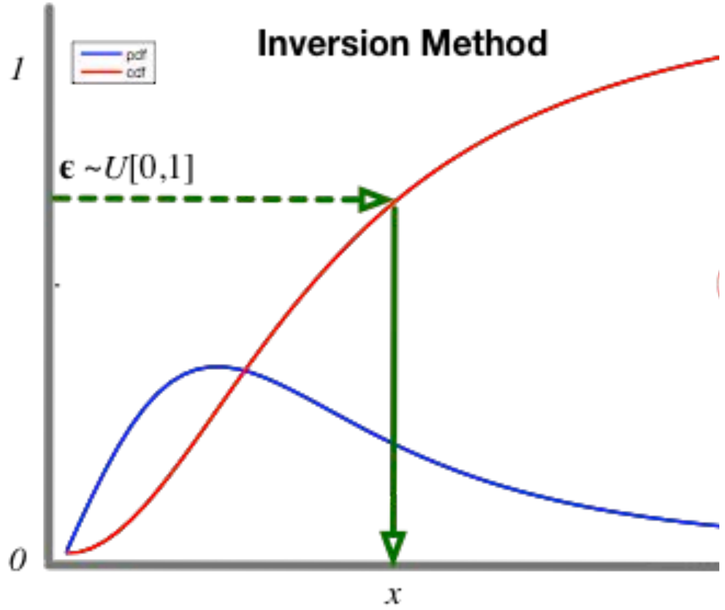
Reparameterisation Tricks



Distributions can be expressed as a transformations of other distributions.

$$z \sim q_\phi(\mathbf{z})$$

$$\mathbf{z} = g(\epsilon, \phi) \quad \epsilon \sim p(\epsilon)$$



Samplers, one-liners and change-of-variables

$$p(z) = \left|\frac{d\epsilon}{dz}\right| p(\epsilon) \implies |p(z)dz| = |p(\epsilon)d\epsilon|$$

Pathwise Estimator

$$\nabla_\phi \mathbb{E}_{q(z)}[f(\mathbf{z})] = \nabla_\phi \int q_\phi(\mathbf{z}) f(\mathbf{z}) d\mathbf{z}$$

Known transformation

$$z = g(\epsilon, \phi); \quad \epsilon \sim p(\epsilon)$$

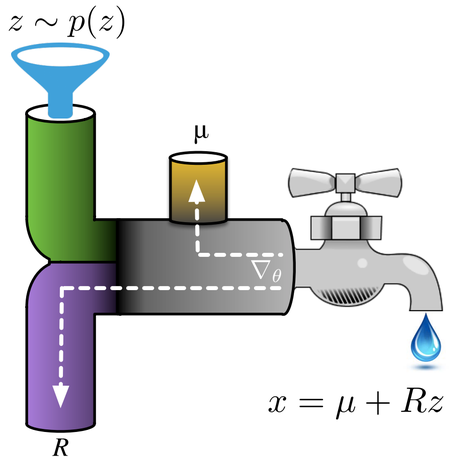
$$= \nabla_\phi \int p(\boldsymbol{\epsilon}) \frac{d\boldsymbol{\epsilon}}{d\mathbf{z}} f(g(\epsilon, \phi)) g'(\epsilon, \phi) d\boldsymbol{\epsilon}$$

Change of variables

$$= \nabla_\phi \mathbb{E}_{p(\epsilon)}[f(g(\phi, \epsilon)] = \mathbb{E}_{p(\epsilon)}[\nabla_\phi f(g(\phi, \epsilon)]$$

Inv fn Thm

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})}[f_\theta(\mathbf{z})] = \mathbb{E}_{p(\epsilon)}[\nabla_\phi f_\theta(g(\epsilon, \phi))]$$



Other names

- Unconscious statistician
- Stochastic backpropagation
- Perturbation analysis
- Reparameterisation trick
- Affine-independent inference

When to use

- Function *f* is differentiable
- Density *q* is known with a suitable transform of a simpler base distribution: inverse CDF, location-scale transform, or other co-ordinate transform.
- Easy to sample from base distribution.

Log-derivative Trick

Score function is the derivative of a log-likelihood function.

$$\nabla_\phi \log q_\phi(\mathbf{z}) = \frac{\nabla_\phi q_\phi(\mathbf{z})}{q_\phi(\mathbf{z})}$$

Several useful properties

Expected score

$$\mathbb{E}_{q(z)}\left[\nabla_\phi \log q_\phi(\mathbf{z})\right] = 0$$

←Show this

$$\mathbb{E}_{q(z)}[\nabla_\phi \log q_\phi(\mathbf{z})] = \int q(z) \frac{\nabla_\phi q_\phi(\mathbf{z})}{q_\phi(\mathbf{z})} = \nabla \int q_\phi(\mathbf{z}) = \nabla 1 = 0$$

Fisher Information

$$\mathbb{V}[\nabla_\theta \log p(\mathbf{x};\theta)] = \mathcal{I}(\theta) = \mathbb{E}_{p(x;\theta)}[\nabla_\theta \log p(\mathbf{x};\theta) \nabla_\theta \log p(\mathbf{x};\theta)^\top]$$

Score-function Estimator

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})}[f_\theta(\mathbf{z})] = \nabla \int q_\phi(\mathbf{z}) f_\theta(\mathbf{z}) d\mathbf{z}$$

Leibnitz integral rule

$$= \int \frac{q_\phi(\mathbf{z})}{q_\phi(\mathbf{z})} \nabla_\phi q_\phi(\mathbf{z}) f(\mathbf{z}) d\mathbf{z}$$

Identity

$$= \int q_\phi(\mathbf{z}) \nabla_\phi \log q_\phi(\mathbf{z}) f(\mathbf{z}) d\mathbf{z}$$

Log-deriv $\quad \nabla_\phi \log q_\phi(\mathbf{z}) = \frac{\nabla_\phi q_\phi(\mathbf{z})}{q_\phi(\mathbf{z})}$

$$= \mathbb{E}_{q_\phi(\mathbf{z})} \left[f(\mathbf{z}) \nabla_\phi \log q_\phi(\mathbf{z}) \right]$$

Gradient

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})}[f_\theta(\mathbf{z})] = \mathbb{E}_{q_\phi(\mathbf{z})} \left[(f(\mathbf{z}) - c) \nabla_\phi \log q_\phi(\mathbf{z}) \right]$$

Control Variate

Other names

- Likelihood ratio method
- REINFORCE and policy gradients
- Automated & Black-box inference

When to use

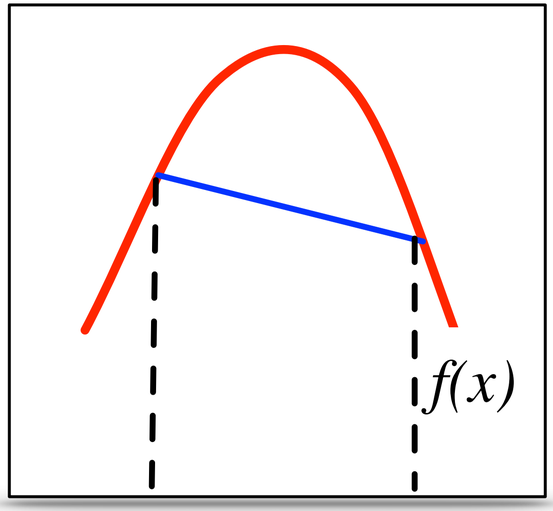
- Function is not differentiable, not analytical.
- Distribution *q* is easy to sample from.
- Density *q* is known and differentiable.

Bounding Tricks

An important result from convex analysis lets us move expectations through a function:

For concave functions $f(.)$

$$f(\mathbb{E}[x]) \geq \mathbb{E}[f(x)]$$



Logarithms are strictly *concave* allowing us to use Jensen's inequality.

$$\log \int p(x)g(x)dx \geq \int p(x) \log g(x)dx$$

Bounding Trick Elsewhere

Optimisation; Variational Inference; Rao-Blackwell Theorem;

Other Bounding Tricks

- Fenchel duality
- Holder's inequality
- Monge-Kantorovich Inequality

Evidence Bounds

Integral problem

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

Proposal

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\frac{q(\mathbf{z})}{q(\mathbf{z})}d\mathbf{z}$$

Importance Weight

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z})}q(\mathbf{z})d\mathbf{z}$$

Jensen's inequality

$$\log \int p(x)g(x)dx \geq \int p(x) \log g(x)dx$$

$$\log p(\mathbf{x}) \geq \int q(\mathbf{z}) \log \left(p(\mathbf{x}|\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z})} \right) d\mathbf{z}$$

$$= \int q(\mathbf{z}) \log p(\mathbf{x}|\mathbf{z}) - \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z})}$$

Lower bound

$$\mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - KL[q(\mathbf{z})||p(\mathbf{z})]$$

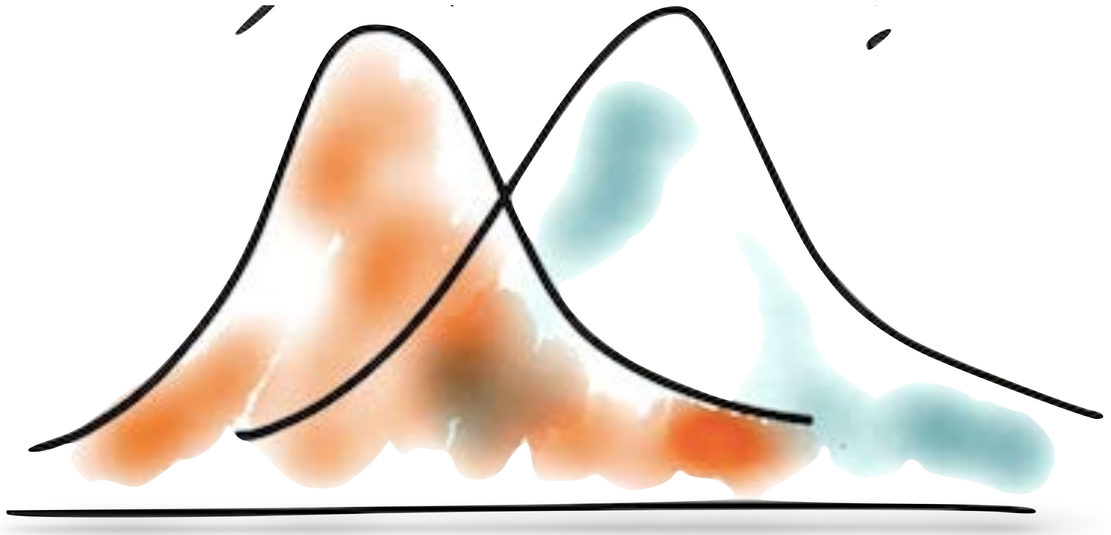
Density Ratio Trick

The ratio of two densities can be computed using a classifier of using samples drawn from the two distributions.

$$\frac{p^*(\mathbf{x})}{q(\mathbf{x})} = \frac{p(y=1|\mathbf{x})}{p(y=-1|\mathbf{x})}$$

Density Ratio Trick Elsewhere

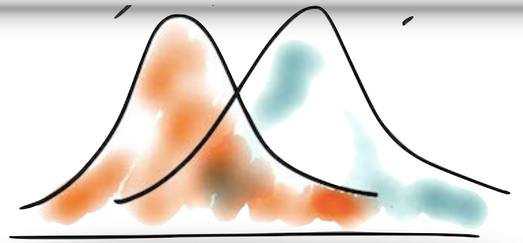
- Generative Adversarial Networks (GANs)
- Noise contrastive estimation, Classifier-ABC
- Two-sample testing
- Covariate-shift, calibration



Density Ratio Estimation

Assign labels $\{y_1, \ldots, y_N\} = \{+1, \ldots, +1, -1, \ldots, -1\}$

Equivalence $p^*(\mathbf{x}) = p(\mathbf{x}|y=1) \quad q(\mathbf{x}) = p(\mathbf{x}|y=-1)$



Density Ratio $\dfrac{p^*(\mathbf{x})}{q(\mathbf{x})}$

Bayes' Rule $p(\mathbf{x}|y) = \dfrac{p(y|\mathbf{x})p(\mathbf{x})}{p(y)}$

Conditional

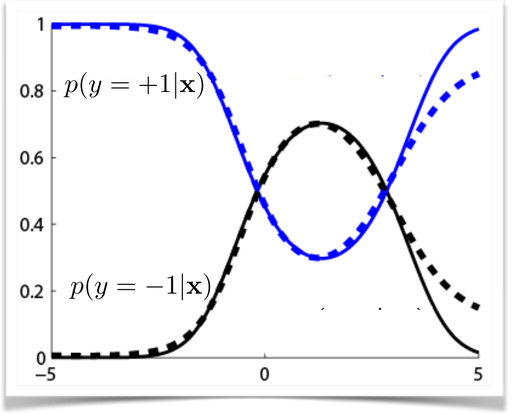
$$\frac{p^*(\mathbf{x})}{q(\mathbf{x})} = \frac{p(\mathbf{x}|y=1)}{p(\mathbf{x}|y=-1)}$$

Bayes' Subst.

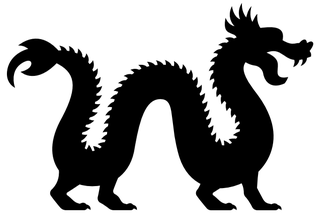
$$= \frac{p(y=+1|\mathbf{x})p(\mathbf{x})}{p(y=+1)} \Big/ \frac{p(y=-1|\mathbf{x})p(\mathbf{x})}{p(y=-1)}$$

Class probability

$$\frac{p^*(\mathbf{x})}{q(\mathbf{x})} = \frac{p(y=1|\mathbf{x})}{p(y=-1|\mathbf{x})}$$

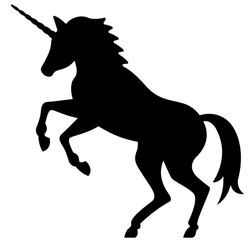


Computing a density ratio is equivalent to class probability estimation.

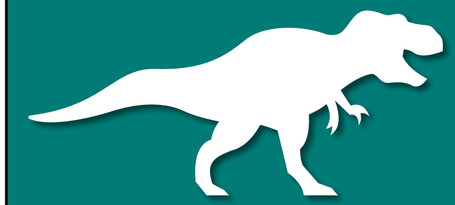


Strengthen your probabilistic dexterity.

Identity	$\frac{p(\mathbf{x})}{p(\mathbf{x})}$
Hutchinson's	$\mathbb{E}[\mathbf{z}\mathbf{z}^\top] = \mathbf{I}$
Flows	$p(\mathbf{y}) = p(\mathbf{x}) \left\lvert \frac{dg}{d\mathbf{x}} \right\rvert^{-1}$
Log-derivative	$\nabla_\phi \log q_\phi(\mathbf{z}) = \frac{\nabla_\phi q_\phi(\mathbf{z})}{q_\phi(\mathbf{z})}$
Reparameterisation	$\mathbf{z} = g(\epsilon, \phi) \quad \epsilon \sim p(\epsilon)$
Density Ratio	$\frac{p^*(\mathbf{x})}{q(\mathbf{x})} = \frac{p(y=1\lvert\mathbf{x})}{p(y=-1\lvert\mathbf{x})}$

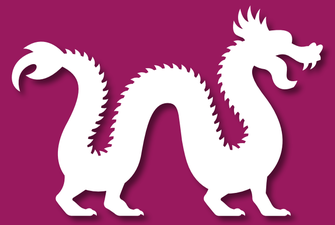


Part 3
Inference in Latent Variable Models



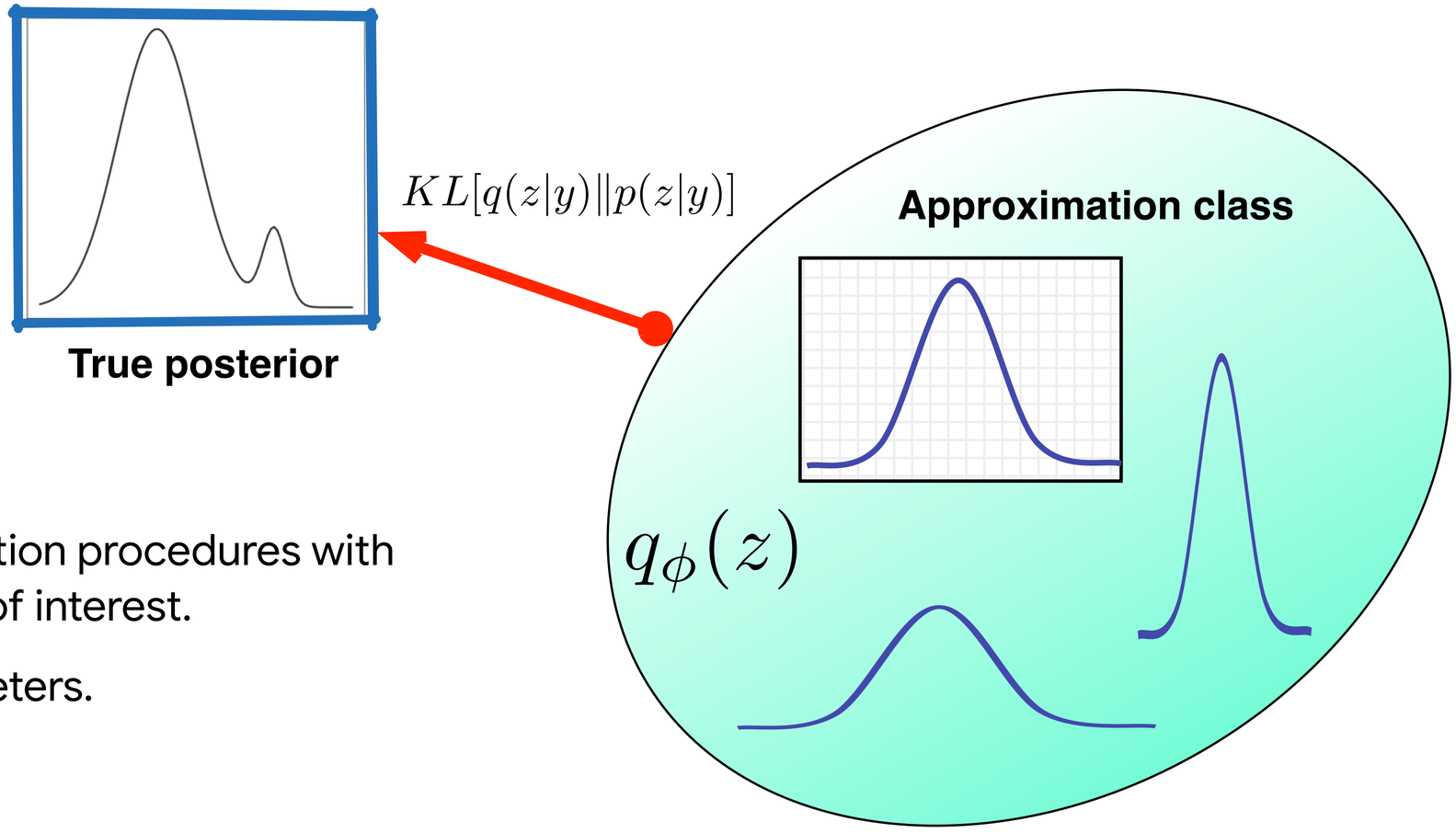
1. Understand the General use for Variational Methods in Machine Learning

2. Derive and Think of Varieties of Variational Autoencoders



3. Consider the Applications, Evaluation and Critical use of Machine Learning

Variational Methods



Deterministic approximation procedures with bounds on probabilities of interest.

Fit the variational parameters.

Variational Calculus

Called a variational method because it derives from the **Calculus of Variations**.

Functions:

- Variables as input, output is a value.
- Full and partial derivatives $\frac{df}{dx}$
- E.g., Maximise likelihood p(x|θ) w.r.t. parameters θ

Functionals:

- Functions as input, output is a value.
- Functional derivatives $\frac{\delta F}{\delta f}$
- E.g., Maximise the entropy H[*p(x)*] w.r.t. *p(x)*

We exploit both types of derivatives in variational inference.

Variational Calculus

Two basic rules

- **Functional derivative:** $$\frac{\delta f(x)}{\delta f(x')} = \delta(x - x')$$

- **Commutative rule:** $$\frac{\delta}{\delta f(x')} \frac{\partial f(x)}{\partial x} = \frac{\partial}{\partial x} \frac{\delta f(x)}{\delta f(x')}$$

Simple example: Maximise the entropy w.r.t.

$$H[p(x)] = -\int p(x) \log p(x) dx$$

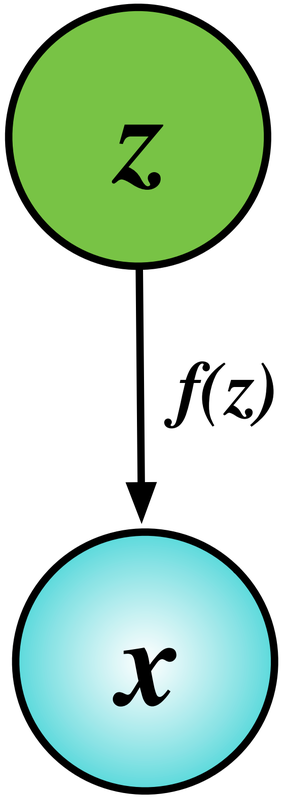
Compute: $$\frac{\delta H[p(x)]}{\delta p(x)}$$

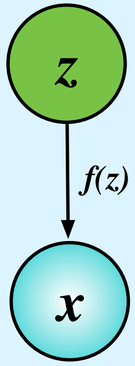
$$-\frac{\delta}{\delta p(x)} \int p(x) \log p(x) dx$$

$$-\int p(x) \frac{1}{p(x)} \delta(x - x') dx' - \int \log p(x) \delta(x - x') dx'$$

$$-1 - \log p(x)$$

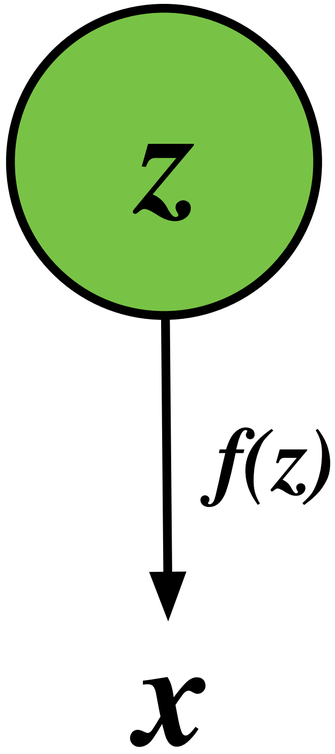
Latent Variable Models





Latent variable models

Introduce an unobserved **local random variables** that represents hidden causes.



Prescribed models

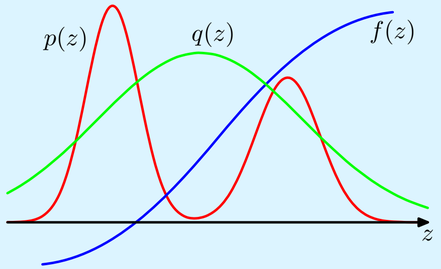
Use observer likelihoods and assume observation noise.

Implicit models

Likelihood-free or simulation-based models.

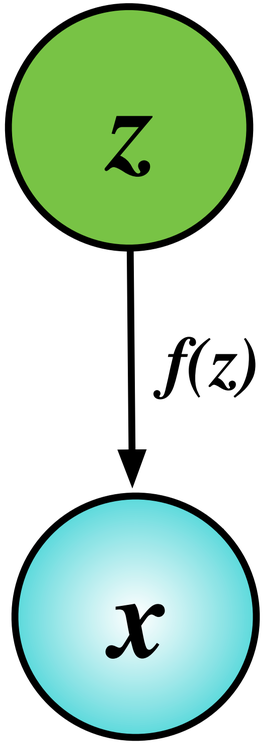
Model Evidence

Model evidence (or marginal likelihood, partition function):
Integrating out any global and local variables enables
model scoring, comparison, selection, moment estimation,
normalisation, posterior computation and prediction.



Learning principle: Model Evidence

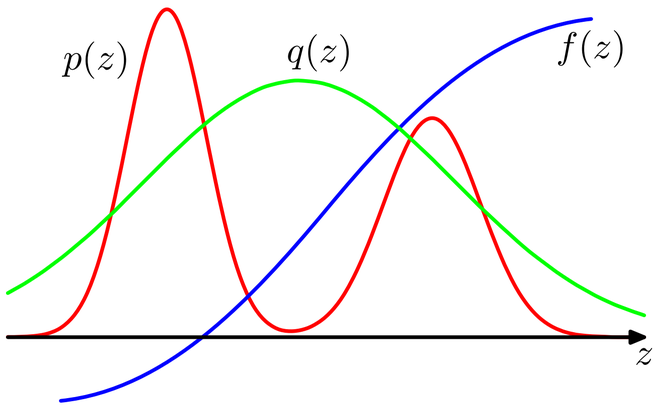
$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}$$



Integral is intractable in general
and requires approximation.

Basic idea: Transform the
integral into an expectation over a simple,
known distribution.

Importance Sampling



Notation

Always think of *q(**z|x**)* but often will write *q(**z**)* for simplicity.

Conditions

- *q(z|x)*>0, when *f(z)p(z)* ≠ 0.
- Easy to sample from *q(z)*.

Integral problem

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

Proposal

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\frac{q(\mathbf{z})}{q(\mathbf{z})}d\mathbf{z}$$

Importance Weight

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z})}q(\mathbf{z})d\mathbf{z}$$

$$w^{(s)} = \frac{p(z)}{q(z)} \quad z^{(s)} \sim q(z)$$

Monte Carlo

$$p(\mathbf{x}) = \frac{1}{S}\sum_{s} w^{(s)}p(\mathbf{x}|\mathbf{z}^{(s)})$$

IS to Variational Inference

Integral problem

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

Proposal

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\frac{q(\mathbf{z})}{q(\mathbf{z})}d\mathbf{z}$$

Importance Weight

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z})}q(\mathbf{z})d\mathbf{z}$$

Jensen's inequality

$$\log \int p(x)g(x)dx \geq \int p(x) \log g(x)dx$$

$$\log p(\mathbf{x}) \geq \int q(\mathbf{z}) \log \left(p(\mathbf{x}|\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z})} \right) d\mathbf{z}$$

$$= \int q(\mathbf{z}) \log p(\mathbf{x}|\mathbf{z}) - \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p(\mathbf{z})}$$

Variational lower bound

$$\mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - KL[q(\mathbf{z})||p(\mathbf{z})]$$

Families of Variational Bounds

Variational Free Energy

$$\mathcal{F}(\mathbf{x}, q) = \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - KL[q(\mathbf{z})||p(\mathbf{z})]$$

Multi-sample Variational Objective

$$\mathcal{F}(\mathbf{x}, q) = \mathbb{E}_{q(z)}\left[\log \frac{1}{S}\sum_s \frac{p(\mathbf{z})}{q(\mathbf{z})} p(\mathbf{x}|\mathbf{z})\right]$$

Renyi Variational Objective

$$\mathcal{F}(\mathbf{x}, q) = \frac{1}{1-\alpha}\mathbb{E}_{q(z)}\left[\left(\log \frac{1}{S}\sum_s \frac{p(\mathbf{z})}{q(\mathbf{z})} p(\mathbf{x}|\mathbf{z})\right)^{1-\alpha}\right]$$

Other generalised families exist. Optimal solution is the same for all objectives.

Variational Bound

$$\mathcal{F}(\mathbf{x}, q) = \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - KL[q(\mathbf{z})||p(\mathbf{z})]$$

Approx. Posterior

Reconstruction

Penalty

- **Approximate posterior distribution *q(z):*** Best match to true posterior *p(z|y)*, one of the unknown inferential quantities of interest to us.
- **Reconstruction cost**: The expected log-likelihood measure how well samples from *q(z)* are able to explain the data *y*.
- **Penalty:** Ensures the the explanation of the data *q(z)* doesn't deviate too far from your beliefs *p(z)*. A mechanism for realising Okham's razor.

Variational Bound

$$\mathcal{F}(\mathbf{x}, q) = \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - KL[q(\mathbf{z})||p(\mathbf{z})]$$

Approx. Posterior

Reconstruction

Penalty

Some comments on *q*:

- **Integration is now optimisation**: optimise for *q(z)* directly.
  - I write *q(z)* to simplify the notation, but it depends on the data, *q(z|x)*.
  - *Easy convergence assessment* since we wait until the free energy (loss) reaches convergence.
- **Variational parameters:** parameters of *q(z)*
  - E.g., if a Gaussian, variational parameters are mean and variance.

Why Variational Inference

Disadvantages:

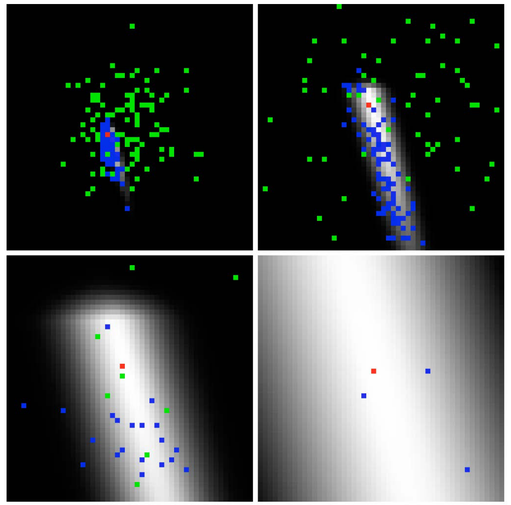
- An **approximate posterior** only - not always guaranteed to find exact posterior in the limit.
- **Difficulty in optimisation** — can get stuck in local minima.
- Typically **under-estimates the variance** of the posterior and can bias maximum likelihood parameter estimates.
- **Limited theory** and guarantees for variational methods.

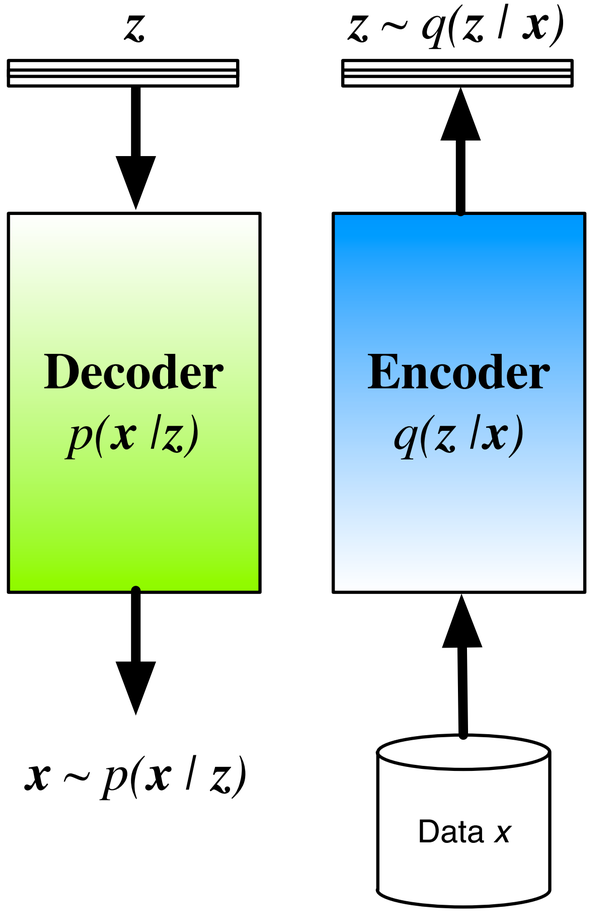
Advantages:

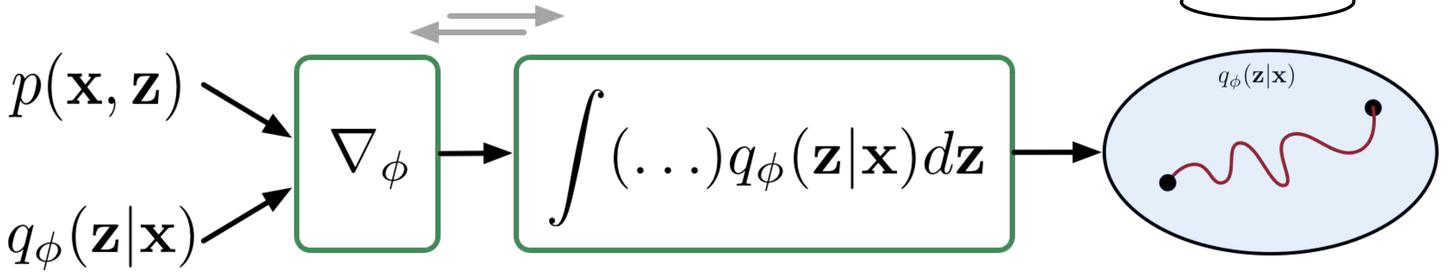
- Applicable to almost **all probabilistic models:** non-linear, non-conjugate, high-dimensional, directed and undirected.
- Transforms problem of **integration into one of optimisation**.
- Easy **convergence assessment**.
- Principled and scalable approach for **model selection**.
- **Compact representation** of the posterior distribution.
- Can be **faster to converge** than competing methods.
- **Numerically stable.**
- Can be used on **modern computing architectures** (CPUs and GPUs)

- Probabilistic Modelling and Inference
- Variational Inference
- Approximate Posteriors
- Variational Optimisation
- Gradient Computation
- Implementation

$$\mathcal{F}(\mathbf{x}, q) = \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - KL[q(\mathbf{z})\|p(\mathbf{z})]$$

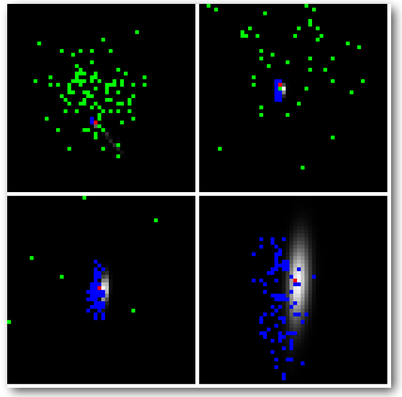


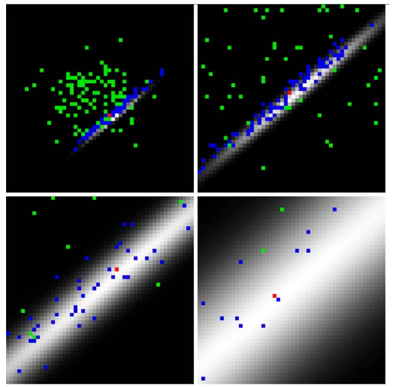


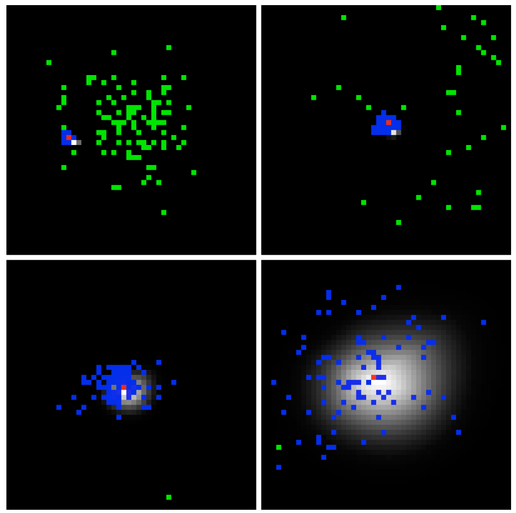


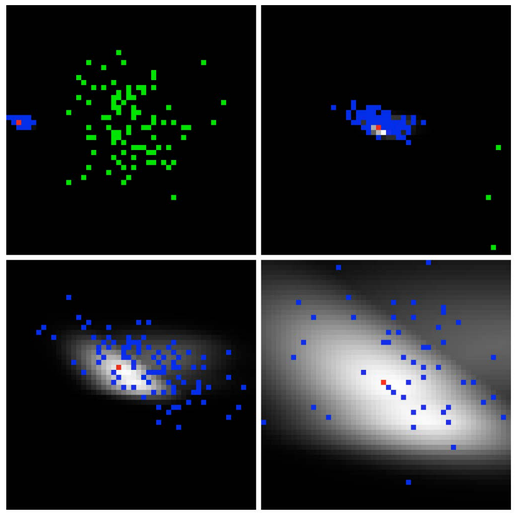
Real Posteriors

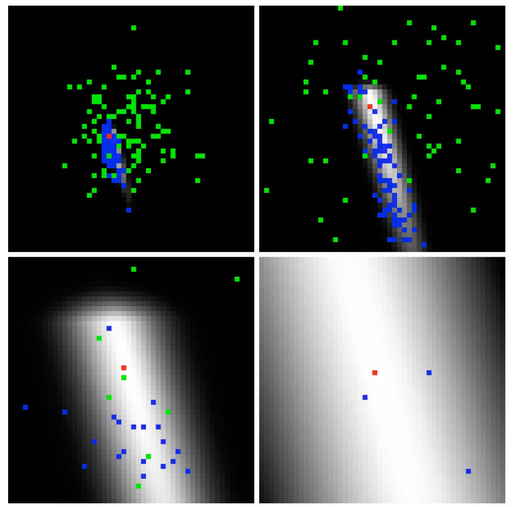
Require flexible approximations for the types of posteriors we are likely to see.





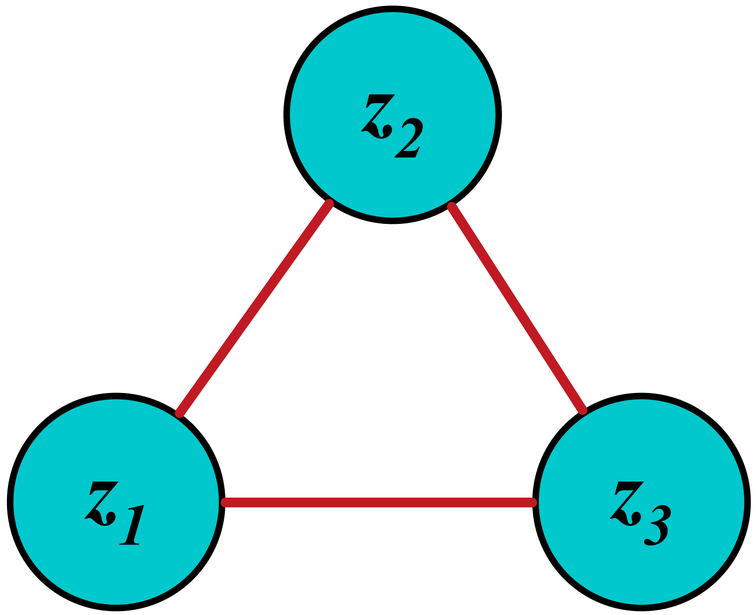


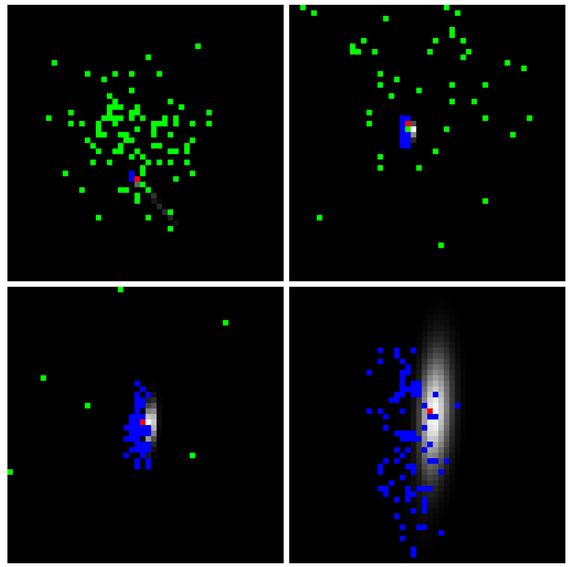




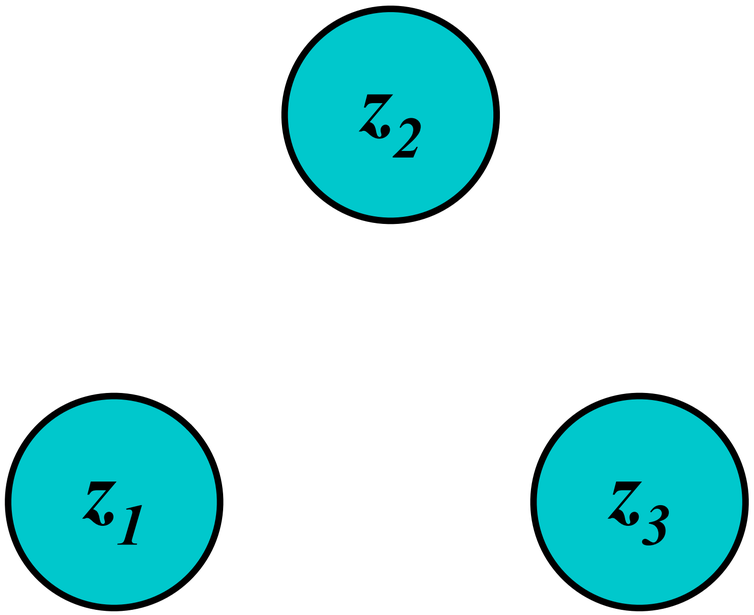
Mean-Fields

True Posterior





Fully-factorised



Most Expressive ⟵ ⟶ *Least Expressive*

$$q^*(z|x) \propto p(x|z)p(z)$$

$$q_{MF}(z|x) = \prod_k q(z_k)$$

Restricted class of approximations: every dimension (or subset of dimensions) of the posterior is independent.

Latent Gaussian Models

Probabilistic Model $z \sim \mathcal{N}(z|0,1) \qquad y \sim p(y|f_\theta(z))$

Mean-field approx $q(z) = \prod_i \mathcal{N}(z_i|\mu_i, \sigma_i^2)$

Variational bound $\mathcal{F}(y,q) = \mathbb{E}_{q(z)}[\log p(y|z)] - KL[q(z)\|p(z)]$

$$\mathcal{F}(y,q) = \mathbb{E}_{q(z)}[\log p(y|z)] - \sum_i KL[q(z_i)\|p(z_i)]$$

$$\mathcal{F}(y,q) = \mathbb{E}_{q(z)}[\log p(y|z)] - \sum_i KL[\mathcal{N}(z_i|\mu_i, \sigma_i^2)\|\mathcal{N}(z_i|0,1)]$$

$$\mathcal{F}(y,q) = \mathbb{E}_{q(z)}[\log p(y|f_\theta(z))] - \frac{1}{2}\sum_i (\sigma_i^2 + \mu_i^2 - 1 - \ln \sigma_i^2)$$

Variational Optimisation

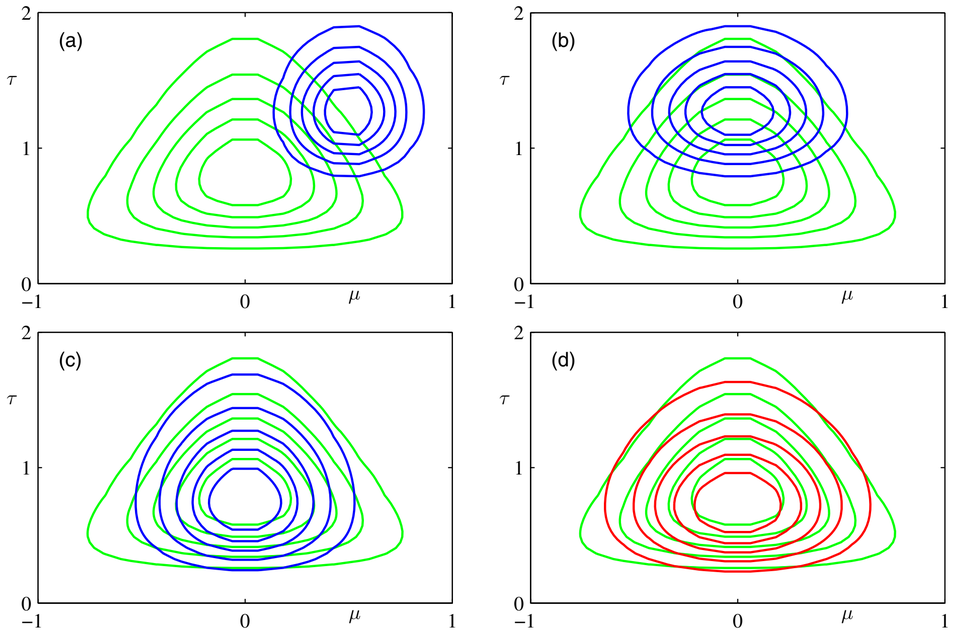
$$\mathcal{F}(\mathbf{x}, q) = \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - KL[q(\mathbf{z})||p(\mathbf{z})]$$

Approx. Posterior

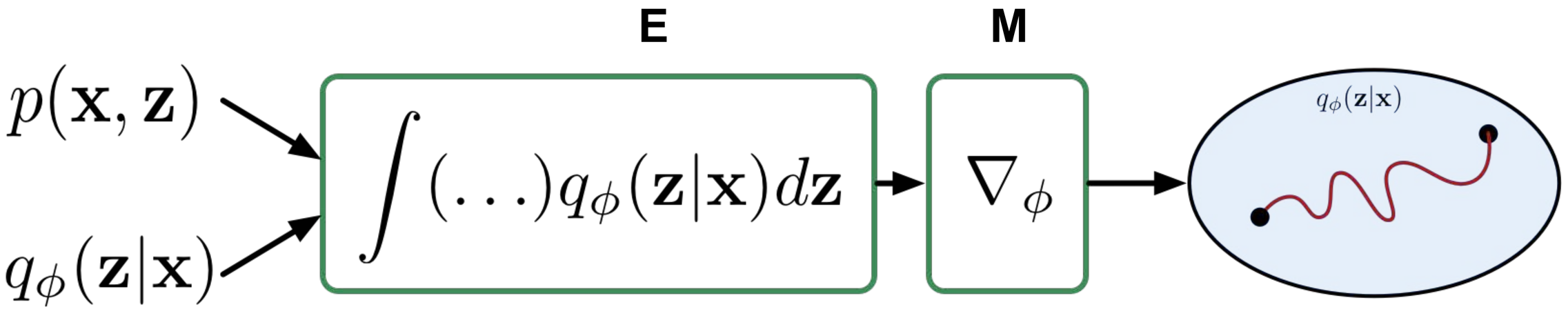
Reconstruction

Penalty

- Variational EM
- Stochastic Variational Inference
- Doubly Stochastic Variational Inference
- Amortised Inference



Classical Inference



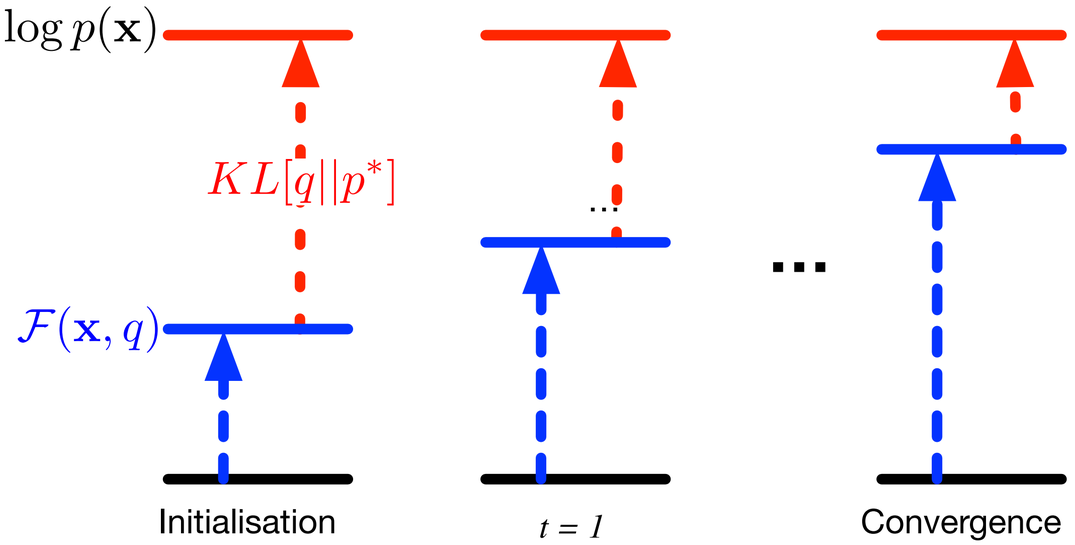
Compute expectations then M-step gradients

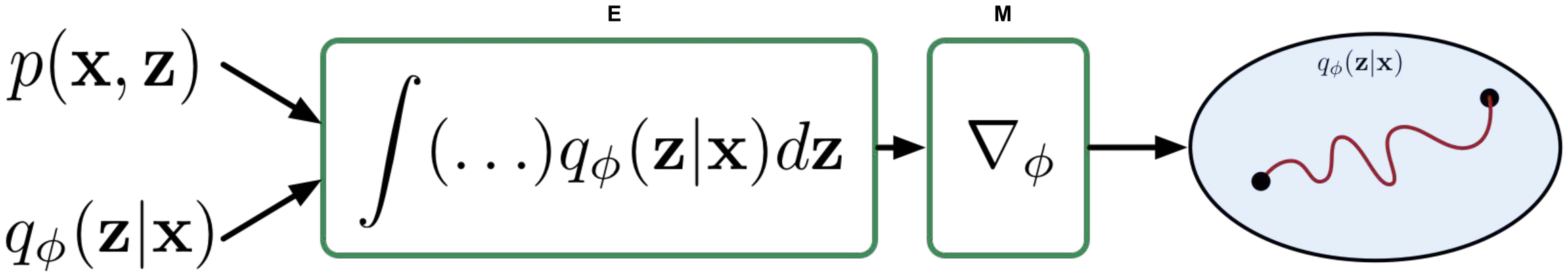
Variational EM

$$\mathcal{F}(\mathbf{x}, q) = \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - KL[q(\mathbf{z})||p(\mathbf{z})]$$

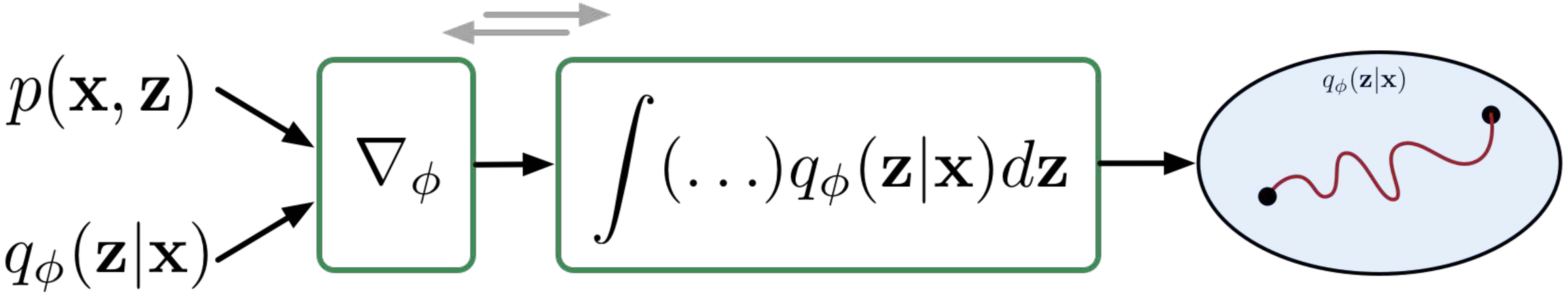
Repeat:

E-step	$\phi \propto \nabla_\phi \mathcal{F}(\mathbf{x}, q)$	*Var. params*
M-step	$\theta \propto \nabla_\theta \mathcal{F}(\mathbf{x}, q)$	*Model params*





Stochastic Inference Approach

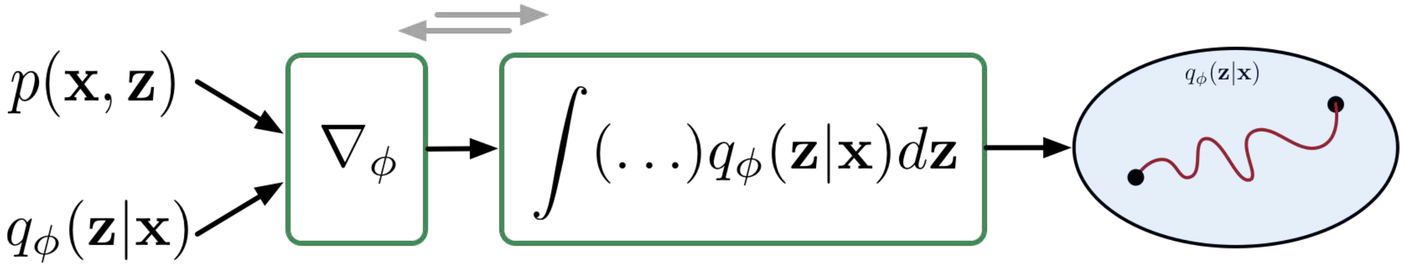


In general, we won't know the expectations.

Gradient is of the parameters of the distribution w.r.t. which the expectation is taken.

Stochastic Gradients

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})}[f_\theta(\mathbf{z})] = \nabla \int q_\phi(\mathbf{z}) f_\theta(\mathbf{z}) d\mathbf{z}$$



Doubly stochastic estimators

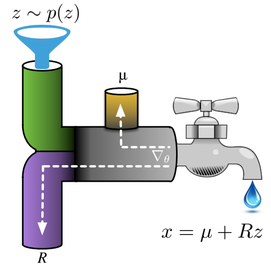
Pathwise Estimator

When easy to use transformation is available and differentiable function *f*.

$$= \mathbb{E}_{p(\epsilon)}[\nabla_\phi f_\theta(g(\epsilon, \phi))]$$

$$z \sim q_\phi(\mathbf{z})$$

$$\mathbf{z} = g(\epsilon, \phi) \quad \epsilon \sim p(\epsilon)$$



Score-function estimator

When function *f* non-differentiable and *q(z)* is easy to sample from.

$$= \mathbb{E}_{q(z)}[f_\theta(\mathbf{z}) \nabla_\phi \log q_\phi(\mathbf{z}))]$$

Amortised Inference

Repeat:

E-step (compute q)

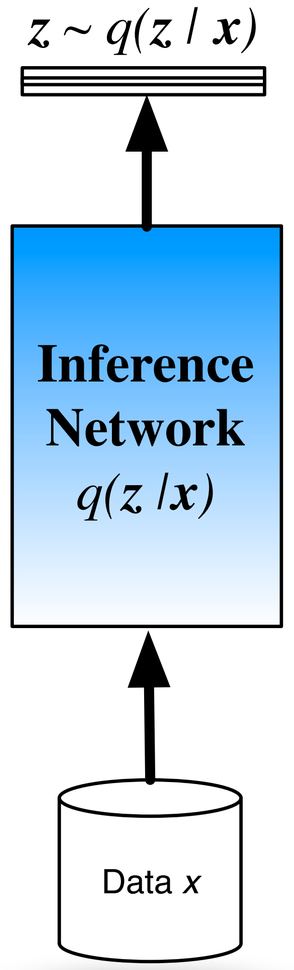
For i = 1, ... N

$$\phi_n \propto \nabla_\phi \mathbb{E}_{q_\phi(z)}[\log p_\theta(\mathbf{x}_n|z_n)] - \nabla_\phi KL[q(z_n)||p(z)]$$

Instead of solving for every observation, amortise using a model.

M-step

$$\theta \propto \frac{1}{N} \sum_n \mathbb{E}_{q_\phi(z)}[\nabla_\theta \log p_\theta(\mathbf{x}_n|z_n)]$$



- **Inference network**: q is an ***encoder,*** an ***inverse*** model, ***recognition model***.
- Parameters of q are now a set of *global parameters* used for inference of all data points - test and train.
- **Amortise (spread) the cost of inference over all data.**
- Joint optimisation of variational and model parameters.

Inference networks provide an efficient mechanism for **posterior inference with memory**

Variational Autoencoder

$$\mathcal{F}(\mathbf{x}, q) = \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - KL[q(\mathbf{z})\|p(\mathbf{z})]$$

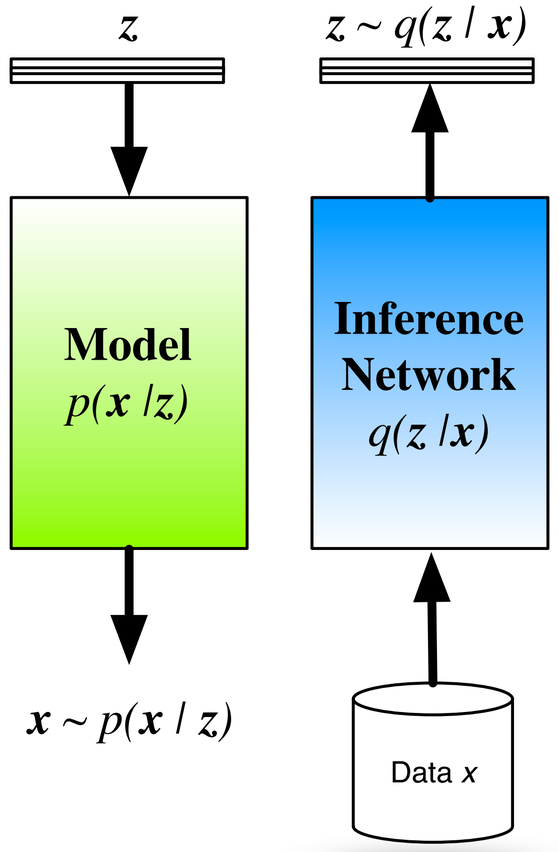
Approx. Posterior

Reconstruction

Penalty

Stochastic encoder-decoder system to implement variational inference.

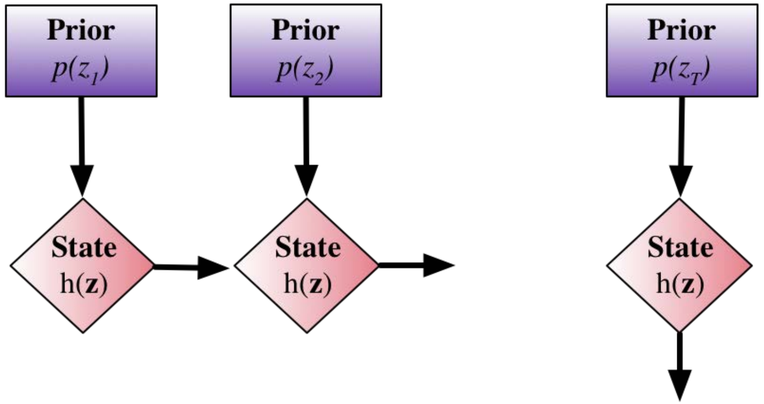
Specific combination of **variational inference** in **latent variable models** using **inference networks**
Variational Auto-encoder



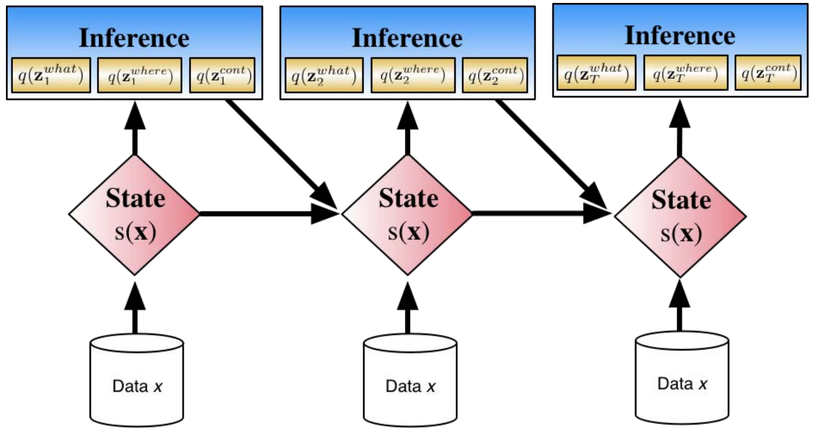
But don't forget what your model is, and what inference you use.

Richer VAES

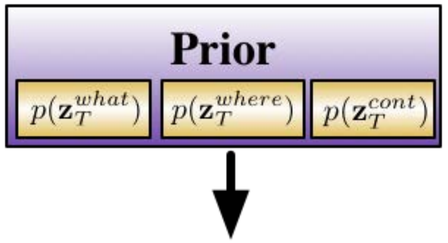
DRAW: Recurrent/Dependent Priors



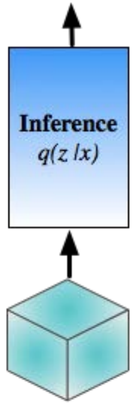
Recurrent/Dependent Inference Networks



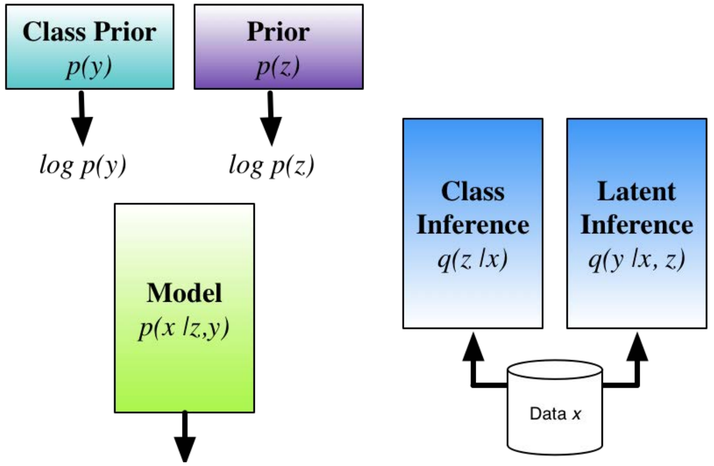
AIR: Structured Priors



Volumetric and Sequence data



Semi-supervised Learning



Minimum Description Length

$$\mathcal{F}(\mathbf{x}, q) = \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - KL[q(\mathbf{z})\|p(\mathbf{z})]$$

Stochastic encoder

Data code-length

Hypothesis code

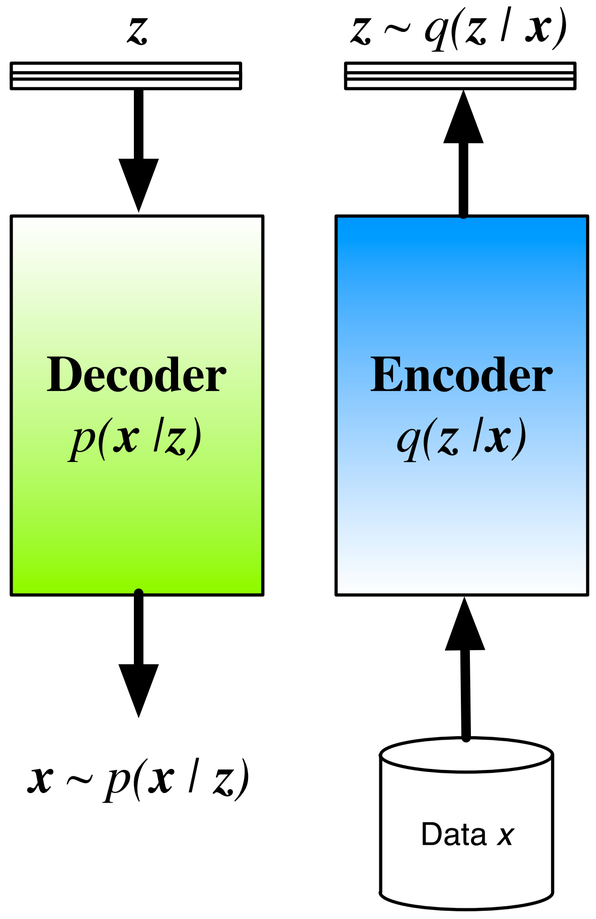
Stochastic encoder-decoder systems implement amortised variational inference.

Regularity in our data that can be explained with latent variables, implies that the data is ***compressible***.

- Must introduce an approximation to the ideal message.
- **Encoder:** variational distribution *q(**z**|**x**)*,
- **Decoder:** likelihood *p(**x**|**z**)*.

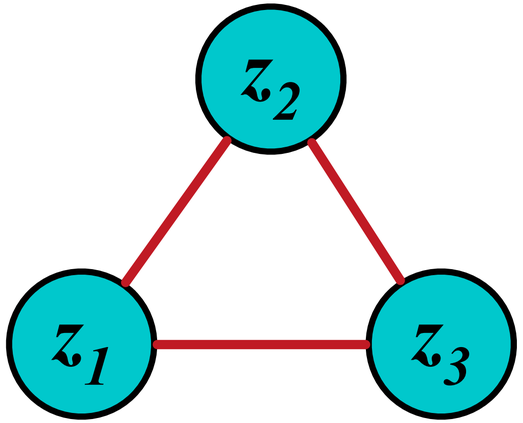
Minimum Description Length (MDL): Inference is a problem of Compression.

we must find the ideal shortest message of our data ***x***: marginal likelihood.

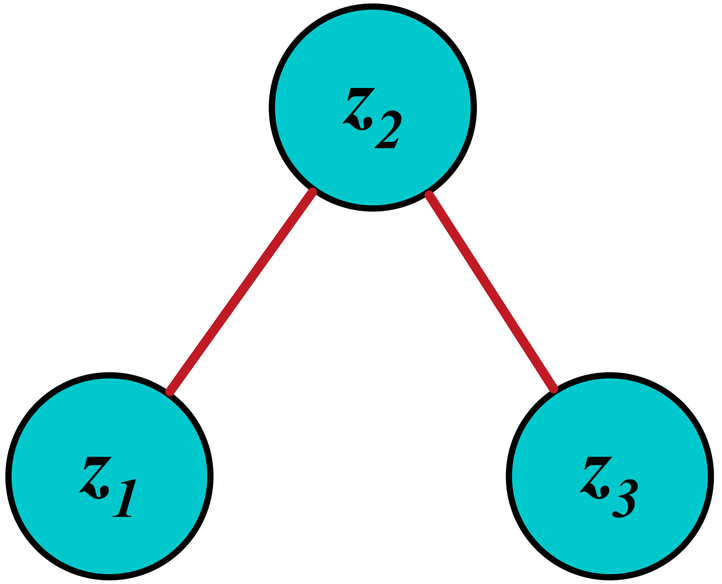


Structured Mean-field

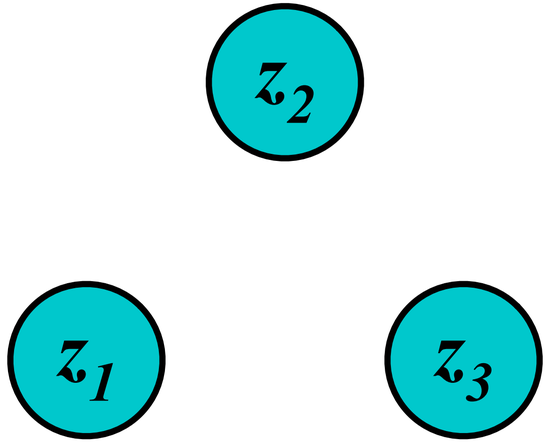
True Posterior



Structured Approx.



Fully-factorised



Most Expressive ⟵ | ⟶ *Least Expressive*

$$q^*(z|x) \propto p(x|z)p(z) \qquad q(z) = \prod_k q_k(z_k|\{z_j\}_{j\neq k}) \qquad q_{MF}(z|x) = \prod_k q(z_k)$$

Structured mean-field: introduce dependencies into our factorisation.

Families of Approximate Posteriors

Covariance Models

$\mathrm{diag}(\alpha_1, \ldots, \alpha_K)$

Mean-field

$\mathrm{diag}(\alpha_1, \ldots, \alpha_K) + \mathbf{u}\mathbf{u}^\top$

Rank-1

$\mathrm{diag}(\alpha_1, \ldots, \alpha_K) + \sum_j \mathbf{u_j}\mathbf{u_j}^\top$

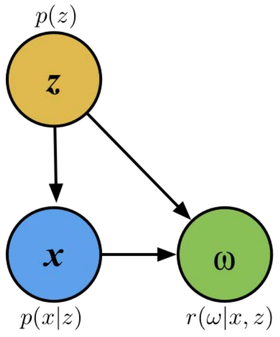
Rank-J

$\mathbf{U}\mathbf{U}^\top$

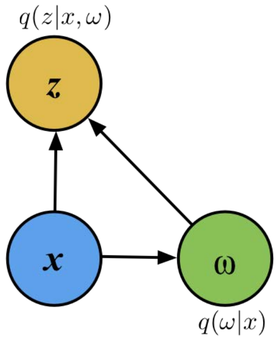
Full

Auxiliary Variable Models

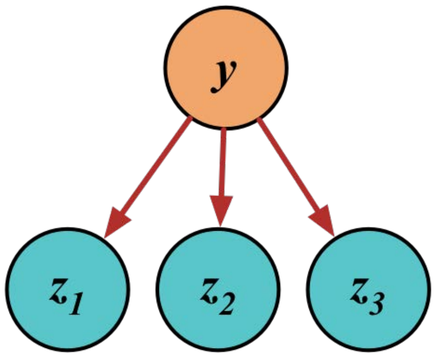
Auxiliary latent variable model p(x,z,ω)



Inference model q(z,ω)

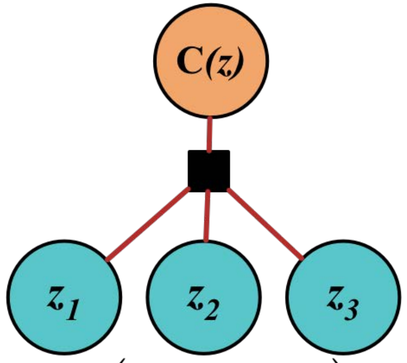


Mixture model



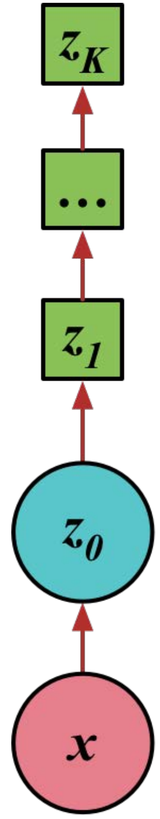
$$q_{mm}(\mathbf{z};\boldsymbol{\nu}) = \sum_r \rho_r q_r(\mathbf{z}_r|\boldsymbol{\nu}_r)$$

Copula Methods



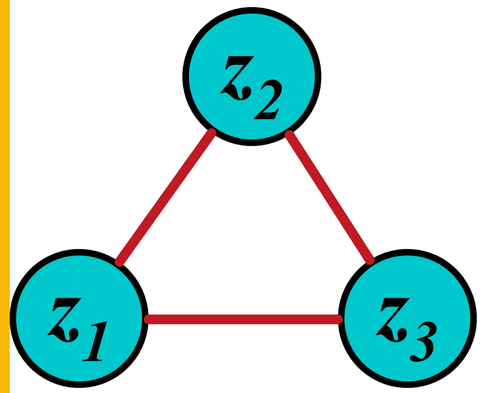
$$q_{lm}(\mathbf{z};\boldsymbol{\nu}) = \left(\prod_k q_k(z_k|\boldsymbol{\nu}_k)\right) C(\mathbf{z};\boldsymbol{\nu}_{k+1})$$

Normalising Flows

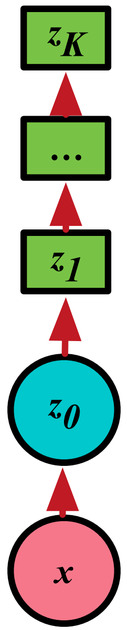


Families of Posterior Approximations

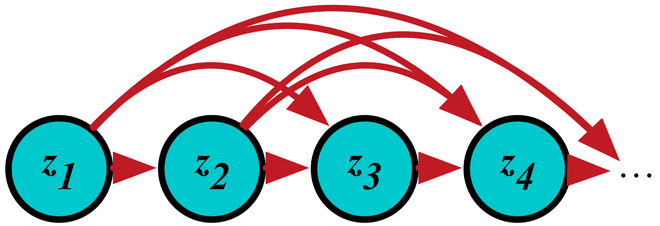
True Posterior



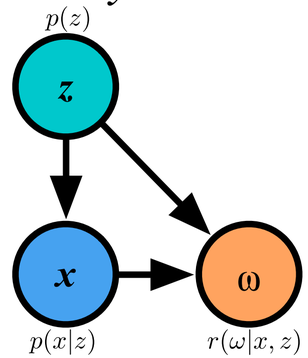
Normalising flows



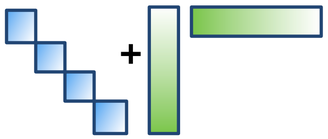
Structured mean-field



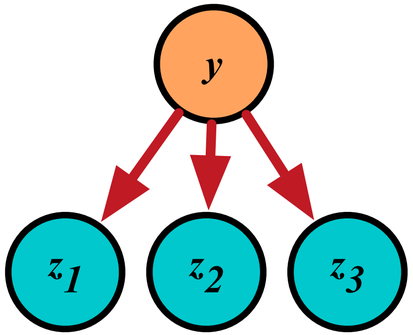
Auxiliary variables



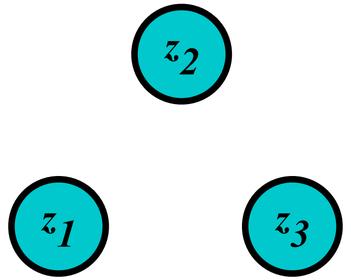
Covariance models



Mixtures



Fully-factorised



Most Expressive ⟵ ⟶ ***Least Expressive***

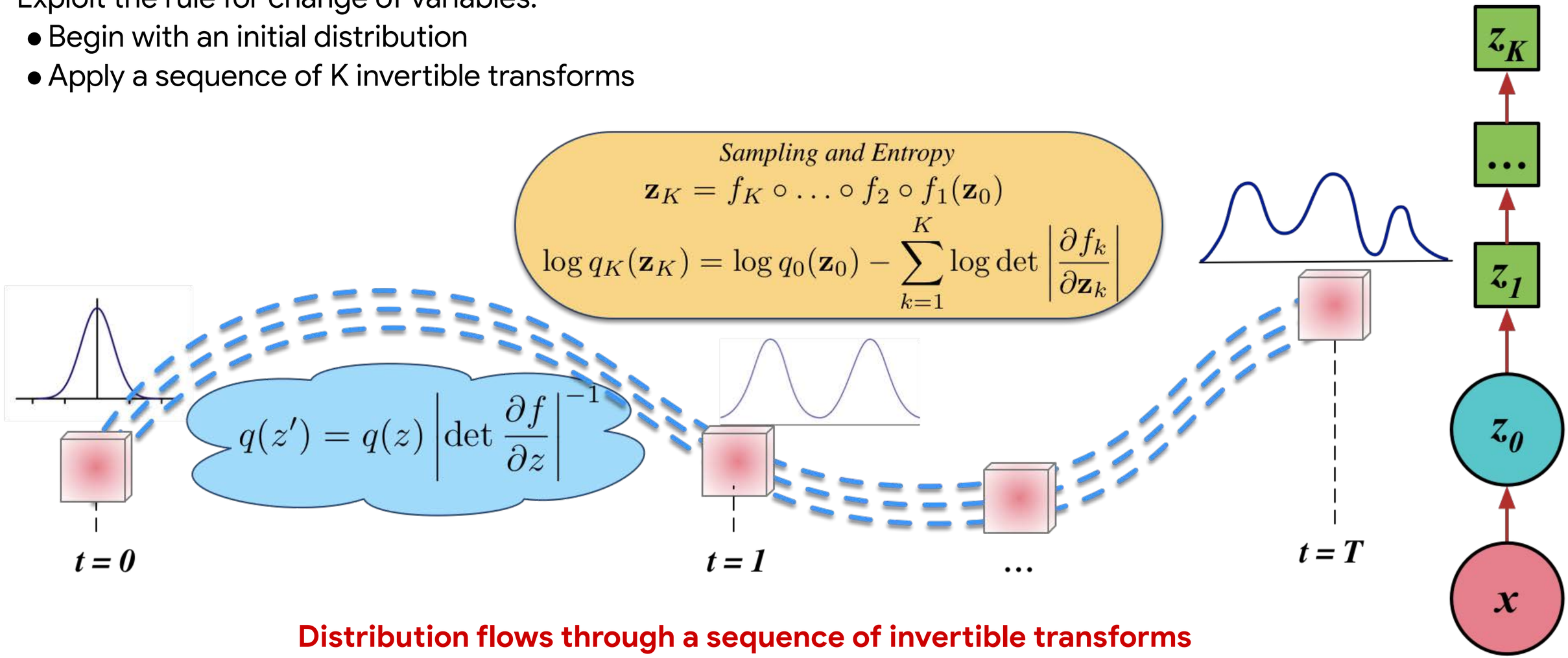
$$q^*(z|x) \propto p(x|z)p(z)$$

$$q_{MF}(z|x) = \prod_k q(z_k$$

Normalising Flows

Exploit the rule for change of variables:

- Begin with an initial distribution
- Apply a sequence of K invertible transforms



Distribution flows through a sequence of invertible transforms

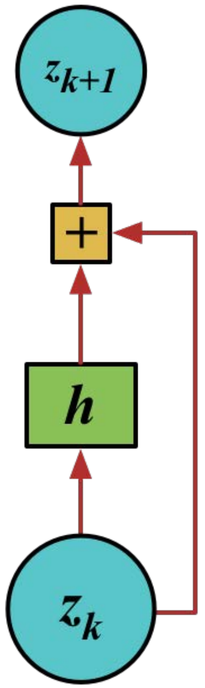
Choice of Transformation

$$\mathcal{L} = \mathbb{E}_{q_0(\mathbf{z}_0)}[\log p(\mathbf{x}, \mathbf{z}_K)] - \mathbb{E}_{q_0(\mathbf{z}_0)}[\log q_0(\mathbf{z}_0)] - \mathbb{E}_{q_0(\mathbf{z}_0)}\left[\sum_{k=1}^{K} \log \det \left|\frac{\partial f_k}{\partial \mathbf{z}_k}\right|\right]$$

Begin with a fully-factorised Gaussian and improve by change of variables.

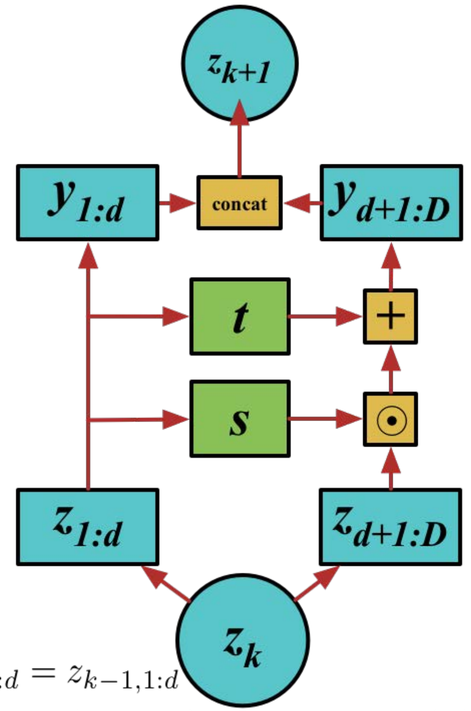
Triangular Jacobians allow for computational efficiency.

Planar Flow



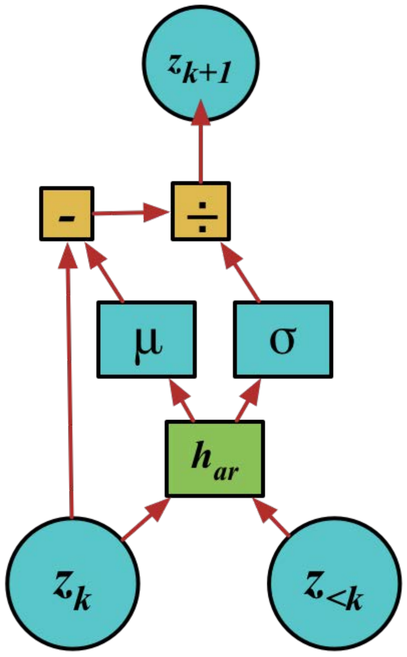
$$z_k = z_{k-1} + uh(w^\top z_{k-1} + b)$$

Real NVP



$$y_{1:d} = z_{k-1,1:d}$$
$$y_{d+1:D} = t(z_{k-1,1:d}) + z_{d+1:D} \odot \exp(s(z_{k-1,1:d}))$$

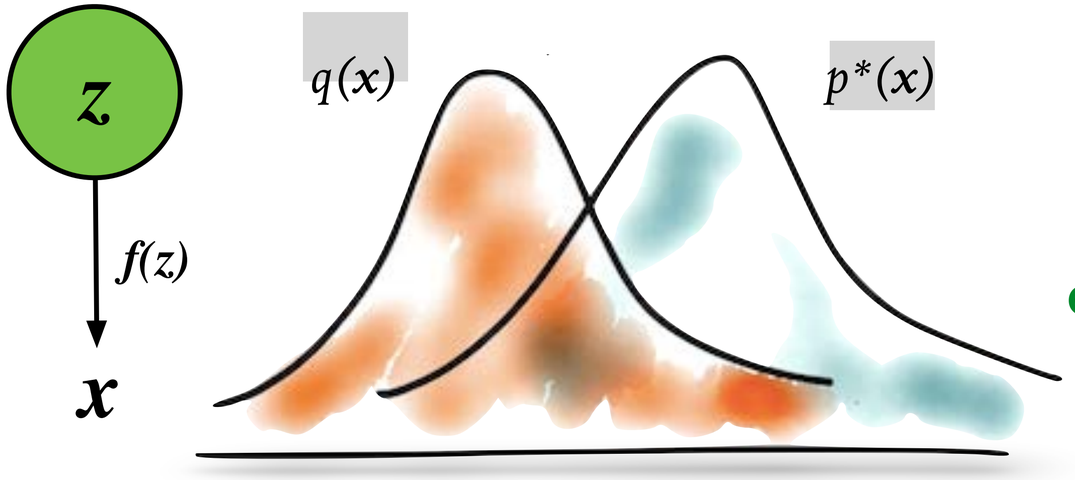
Inverse AR Flow



$$z_k = \frac{z_{k-1} - \mu_k(z_{<k}, x)}{\sigma_k(z_{<k}, x)}$$

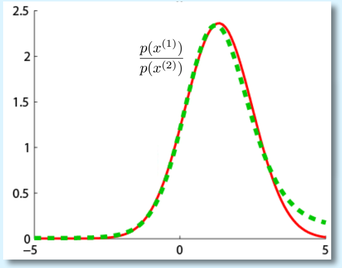
Linear time computation of the determinant and its gradient.

Learning by Comparison



We compare the estimated distribution q(x) to the true distribution p*(x) using samples.

Basic idea: Transform into learning a model of the density ratio.



Learning principle: Two-sample tests

$$\frac{p^*(\mathbf{x})}{q(\mathbf{x})} = 1 \qquad p^*(\mathbf{x}) = q(\mathbf{x})$$

Interest is not in estimating the marginal probabilities, only in how they are related.

Evaluating Generative Models

Likelihood approaches:

- Straightforward for models which have tractable likelihoods
- For VAEs, we can compare evidence lower bounds (ELBO) to log-likelihoods

Evaluating in target domains

- Evaluate performance on final task, whether transfer, generalisation, reward-maximisation, detection acc.
- Question what is being evaluated - diversity or quality.

Baselines and confidence:

- Spend time tuning your baselines (architecture, learning rate, optimizer etc.).
- Be amazed (rather than dejected) at how well they can perform.
- Use random seeds for reproducibility Report results averaged over multiple random seeds along with confidence intervals

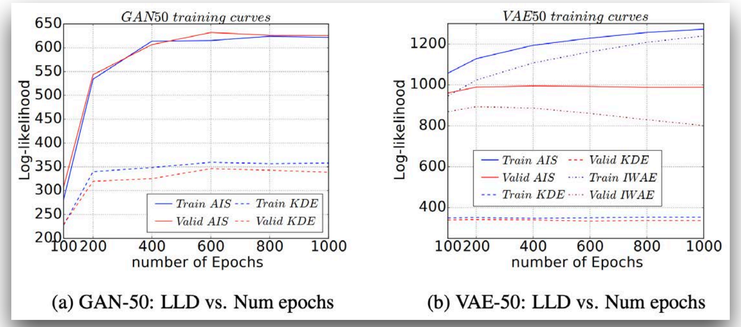
Human evaluations:

- Scoring based on human judgements(e.g., Mechanical Turk).
- Can be expensive, biased, hard to reproduce.

Some Evaluation Methods

Annealed Importance Sampling (AIS)

A Monte Carlo algorithm used to estimate ratios of normalising constants. Computes a sequence of importance sampling-based estimates.

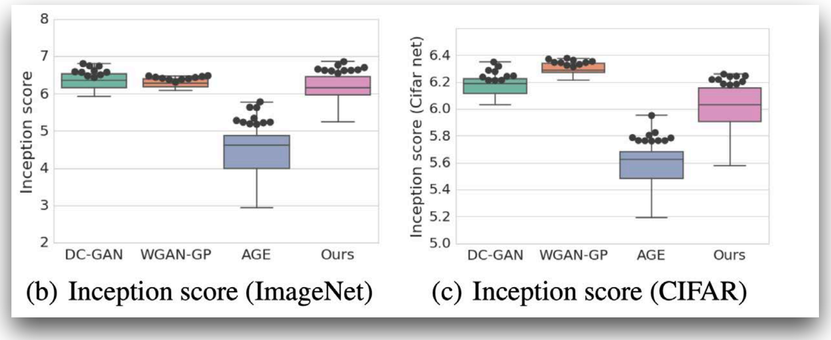


(a) GAN-50: LLD vs. Num epochs (b) VAE-50: LLD vs. Num epochs

- Will need to choose an evaluation likelihood.
- Difficult to implement in practice, sensitive to schedule and instabilities.
- Can give insight into behaviour and directions for improvement.

Inception Score

A feature-based score that uses the KL divergence between features obtained from a pre-trained classifier.

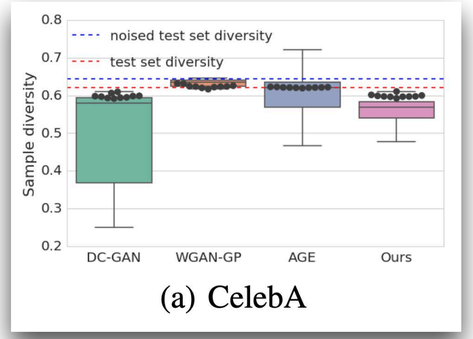


(b) Inception score (ImageNet) (c) Inception score (CIFAR)

- Use features from an inception network classifier.
- Hard to use outside of the data that the classifier was trained on.
- But can correlate with human judgements on image quality,

Similarity Scores

Use knowledge of computer vision to judge the structural similarity of images.



(a) CelebA

- Multiscale structural similarity is one popular one.
- But many kinds of summary stats are possible.
- Won't say when a model works well, but rather when it deviates from the test set.

We et al. (2016); Salimans et al., (2016); Heusel et al., (2017); Wang et al. (2003)

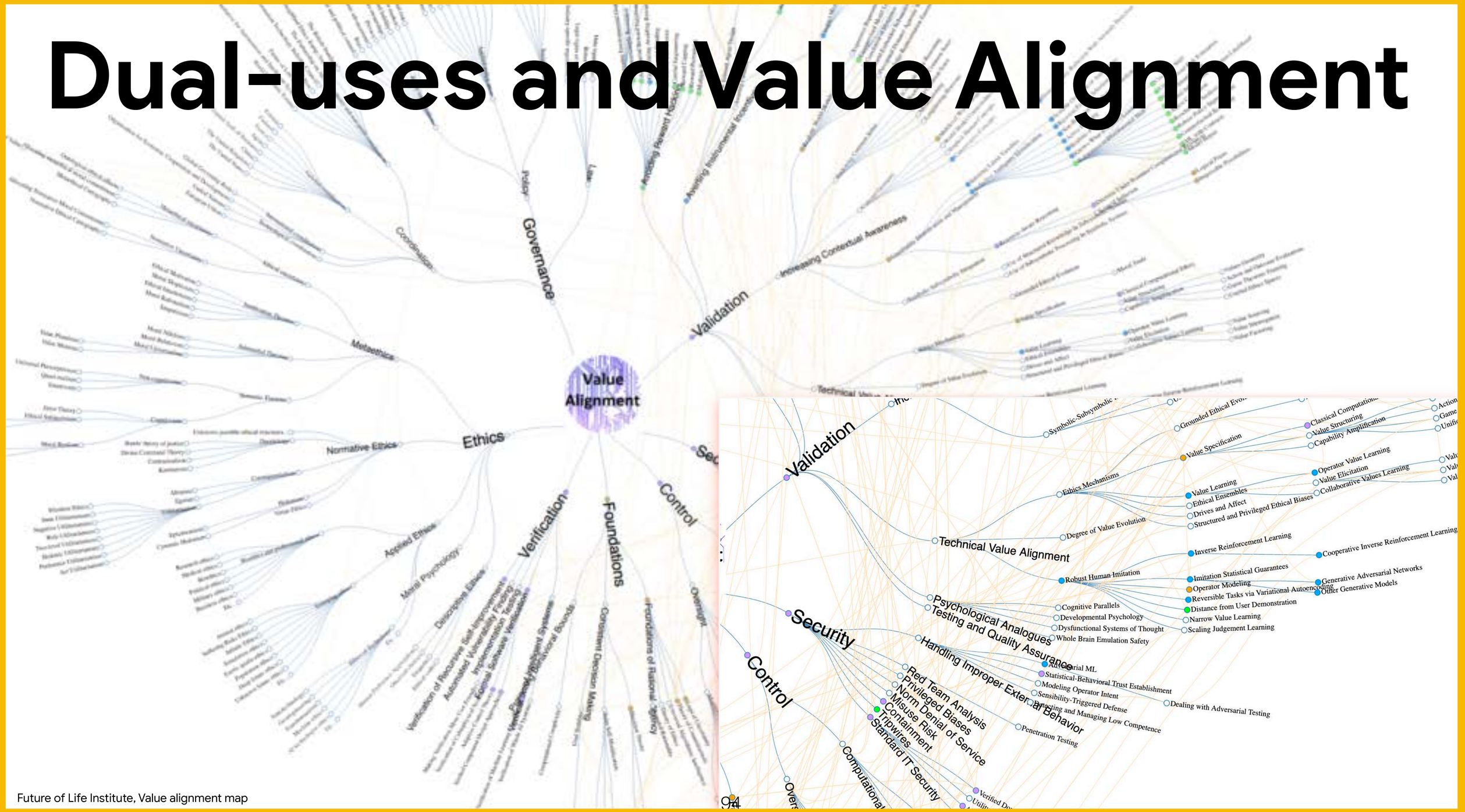
Critical Practice for ML

Bansal et al. (2018)

Consider the uses of our models.

What are the dual uses of generative models. How do we think critically about these uses, educate, regulate, co-design these tools.

Dual-uses and Value Alignment



Future of Life Institute, Value alignment map

Neutrality and Universality

Neutrality Traps

- **The Portability Trap:** Failure to understand how repurposing algorithmic solutions designed for one social context may be inaccurate / do harm when applied to a different context.
- **The Formalism Trap:** Failure to account for the full meaning of social concepts such as fairness, which be resolved through mathematical formalisms.
- **The Ripple Effect Trap**: Failure to understand how the insertion of technology into an existing social system changes the behaviours and embedded values of the pre-existing system .
- **The Solutionism Trap**: Failure to recognise the possibility that the best solution to a problem may not involve technology.

Universality

'A mono-cultural view of ethics conceives itself as the only valid one. In order to avoid this kind of ethical chauvinism and colonialism it is necessary that transcultural ethics arise from an intercultural dialogue instead of thinking of itself as universal without noticing its own cultural bias.' Capurro, 2004

SUSTAINABLE DEVELOPMENT GOALS

Probabilistic Thinking

Not exhaustive list, and many references to be updated.

Cheeseman, P.C., 1985, August. In Defense of Probability. In *IJCAI* (Vol. 2, pp. 1002-1009).

Murphy, K.P., 2012. Machine Learning: A Probabilistic Perspective.

Efron, B., 1982. Maximum likelihood and decision theory. *The annals of Statistics*, pp.340-356.

Stochastic Optimisation

S. Mohamed, M. Rosca, M. Figurnov, A. Mnih. Monte Carlo Gradient Estimation in Machine Learning. 2019.

P L'Ecuyer, Note: On the interchange of derivative and expectation for likelihood ratio derivative estimators, Management Science, 1995

Peter W Glynn, Likelihood ratio gradient estimation for stochastic systems, Communications of the ACM, 1990

Michael C Fu, Gradient estimation, Handbooks in operations research and management science, 2006

Ronald J Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine learning, 1992

Paul Glasserman, Monte Carlo methods in financial engineering, , 2003

Luc Devroye, Random variate generation in one line of code, Proceedings of the 28th conference on Winter simulation, 1996

L. Devroye, Non-uniform random variate generation, , 1986

Omiros Papaspiliopoulos, Gareth O Roberts, Martin Skold, A general framework for the parametrization of hierarchical models, Statistical Science, 2007

Michael C Fu, Gradient estimation, Handbooks in operations research and management science, 2006

Ranganath, Rajesh, Sean Gerrish, and David M. Blei. "Black Box Variational Inference." In AISTATS, pp. 814-822. 2014.

Mnih, Andriy, and Karol Gregor. "Neural variational inference and learning in belief networks." arXiv preprint arXiv:1402.0030 (2014).

Lázaro-Gredilla, Miguel. "Doubly stochastic variational Bayes for non-conjugate inference." (2014).

Wingate, David, and Theophane Weber. "Automated variational inference in probabilistic programming." arXiv preprint arXiv: 1301.1299 (2013).

Paisley, John, David Blei, and Michael Jordan. "Variational Bayesian inference with stochastic search." arXiv preprint arXiv:1206.6430 (2012).

- Applications of Deep Generative Models
- Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra. "Stochastic backpropagation and approximate inference in deep generative models." ICML 2014
- Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." ICLR 2014
- Gregor, Karol, et al. "Towards Conceptual Compression." arXiv preprint arXiv:1604.08772 (2016).
- Eslami, S. M., Heess, N., Weber, T., Tassa, Y., Kavukcuoglu, K., & Hinton, G. E. (2016). Attend, Infer, Repeat: Fast Scene Understanding with Generative Models. arXiv preprint arXiv:1603.08575.
- Oh, Junhyuk, Xiaoxiao Guo, Honglak Lee, Richard L. Lewis, and Satinder Singh. "Action-conditional video prediction using deep networks in atari games." In Advances in Neural Information Processing Systems, pp. 2863-2871. 2015.
- Rezende, Danilo Jimenez, Shakir Mohamed, Ivo Danihelka, Karol Gregor, and Daan Wierstra. "One-Shot Generalization in Deep Generative Models." arXiv preprint arXiv:1603.05106 (2016).
- Rezende, Danilo Jimenez, S. M. Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. "Unsupervised Learning of 3D Structure from Images." arXiv preprint arXiv:1607.00662 (2016).
- Kingma, Diederik P., Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. "Semi-supervised learning with deep generative models." In Advances in Neural Information Processing Systems, pp. 3581-3589. 2014.
- Maaløe, Lars, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. "Auxiliary Deep Generative Models." arXiv preprint arXiv:1602.05473 (2016).
- Odena, Augustus. "Semi-Supervised Learning with Generative Adversarial Networks." arXiv preprint arXiv:1606.01583 (2016).
- Springenberg, Jost Tobias. "Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks." arXiv preprint arXiv:1511.06390 (2015).
- Blundell, Charles, Benigno Uria, Alexander Pritzel, Yazhe Li, Avraham Ruderman, Joel Z. Leibo, Jack Rae, Daan Wierstra, and Demis Hassabis. "Model-Free Episodic Control." arXiv preprint arXiv:1606.04460 (2016).
- Higgins, Irina, Loic Matthey, Xavier Glorot, Arka Pal, Benigno Uria, Charles Blundell, Shakir Mohamed, and Alexander Lerchner. "Early Visual Concept Learning with Unsupervised Deep Learning." arXiv preprint arXiv:1606.05579 (2016).

- **Fully-observed Models**

- Oord, Aaron van den, Nal Kalchbrenner, and Koray Kavukcuoglu. "Pixel recurrent neural networks." arXiv preprint arXiv:1601.06759 (2016).
- Larochelle, Hugo, and Iain Murray. "The Neural Autoregressive Distribution Estimator." In AISTATS, vol. 1, p. 2. 2011.
- Uria, Benigno, Iain Murray, and Hugo Larochelle. "A Deep and Tractable Density Estimator." In ICML, pp. 467-475. 2014.
- Veness, Joel, Kee Siong Ng, Marcus Hutter, and Michael Bowling. "Context tree switching." In 2012 Data Compression Conference, pp. 327-336. IEEE, 2012.
- Rue, Havard, and Leonhard Held. Gaussian Markov random fields: theory and applications. CRC Press, 2005.
- Wainwright, Martin J., and Michael I. Jordan. "Graphical models, exponential families, and variational inference." Foundations and Trends® in Machine Learning 1, no. 1-2 (2008): 1-305.

- **Implicit Probabilistic Models**

- Tabak, E. G., and Cristina V. Turner. "A family of nonparametric density estimation algorithms." Communications on Pure and Applied Mathematics 66, no. 2 (2013): 145-164.
- Rezende, Danilo Jimenez, and Shakir Mohamed. "Variational inference with normalizing flows." arXiv preprint arXiv:1505.05770 (2015).
- Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." In Advances in Neural Information Processing Systems, pp. 2672-2680. 2014.
- Verrelst, Herman, Johan Suykens, Joos Vandewalle, and Bart De Moor. "Bayesian learning and the Fokker-Planck machine." In Proceedings of the International Workshop on Advanced Black-box Techniques for Nonlinear Modeling, Leuven, Belgium, pp. 55-61. 1998.
- Devroye, Luc. "Random variate generation in one line of code." In Proceedings of the 28th conference on Winter simulation, pp. 265-272. IEEE Computer Society, 1996.
- Ravuri, S., Mohamed, S., Rosca, M. and Vinyals, O., 2018. Learning Implicit Generative Models with the Method of Learned Moments. *arXiv preprint arXiv:1806.11006*.

Latent variable models

Dayan, Peter, Geoffrey E. Hinton, Radford M. Neal, and Richard S. Zemel. "The helmholtz machine." Neural computation 7, no. 5 (1995): 889-904.

Hyvärinen, A., Karhunen, J., & Oja, E. (2004). Independent component analysis (Vol. 46). John Wiley & Sons.

Gregor, Karol, Ivo Danihelka, Andriy Mnih, Charles Blundell, and Daan Wierstra. "Deep autoregressive networks." arXiv preprint arXiv: 1310.8499 (2013).

Ghahramani, Zoubin, and Thomas L. Griffiths. "Infinite latent feature models and the Indian buffet process." In Advances in neural information processing systems, pp. 475-482. 2005.

Teh, Yee Whye, Michael I. Jordan, Matthew J. Beal, and David M. Blei. "Hierarchical dirichlet processes." Journal of the american statistical association (2012).

Adams, Ryan Prescott, Hanna M. Wallach, and Zoubin Ghahramani. "Learning the Structure of Deep Sparse Graphical Models." In AISTATS, pp. 1-8. 2010.

Lawrence, Neil D. "Gaussian process latent variable models for visualisation of high dimensional data." Advances in neural information processing systems 16.3 (2004): 329-336.

Damianou, Andreas C., and Neil D. Lawrence. "Deep Gaussian Processes." In AISTATS, pp. 207-215. 2013.

Mattos, César Lincoln C., Zhenwen Dai, Andreas Damianou, Jeremy Forth, Guilherme A. Barreto, and Neil D. Lawrence. "Recurrent Gaussian Processes." arXiv preprint arXiv:1511.06644 (2015).

Salakhutdinov, Ruslan, Andriy Mnih, and Geoffrey Hinton. "Restricted Boltzmann machines for collaborative filtering." In Proceedings of the 24th international conference on Machine learning, pp. 791-798. ACM, 2007.

Saul, Lawrence K., Tommi Jaakkola, and Michael I. Jordan. "Mean field theory for sigmoid belief networks." Journal of artificial intelligence research 4, no. 1 (1996): 61-76.

Frey, Brendan J., and Geoffrey E. Hinton. "Variational learning in nonlinear Gaussian belief networks." Neural Computation 11, no. 1 (1999): 193-213.

Inference and Learning

Jordan, Michael I., Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. "An introduction to variational methods for graphical models." Machine learning 37, no. 2 (1999): 183-233.
Hoffman, Matthew D., David M. Blei, Chong Wang, and John William Paisley. "Stochastic variational inference." Journal of Machine Learning Research 14, no. 1 (2013): 1303-1347.
Honkela, Antti, and Harri Valpola. "Variational learning and bits-back coding: an information-theoretic view to Bayesian learning." IEEE Transactions on Neural Networks 15, no. 4 (2004): 800-810.
Burda, Yuri, Roger Grosse, and Ruslan Salakhutdinov. "Importance weighted autoencoders." arXiv preprint arXiv: 1509.00519 (2015).
Li, Yingzhen, and Richard E. Turner. "Variational Inference with R\'enyi Divergence." arXiv preprint arXiv:1602.02311 (2016).
Borgwardt, Karsten M., and Zoubin Ghahramani. "Bayesian two-sample tests." arXiv preprint arXiv:0906.4032 (2009).
Gutmann, Michael, and Aapo Hyvärinen. "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models." AISTATS. Vol. 1. No. 2. 2010.
Tsuboi, Yuta, Hisashi Kashima, Shohei Hido, Steffen Bickel, and Masashi Sugiyama. "Direct Density Ratio Estimation for Large-scale Covariate Shift Adaptation." Information and Media Technologies 4, no. 2 (2009): 529-546.
Sugiyama, Masashi, Taiji Suzuki, and Takafumi Kanamori. Density ratio estimation in machine learning. Cambridge University Press, 2012.

Amortised Inference

Gershman, Samuel J., and Noah D. Goodman. "Amortized inference in probabilistic reasoning." In Proceedings of the 36th Annual Conference of the Cognitive Science Society. 2014.
Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra. "Stochastic backpropagation and approximate inference

Probabilistic Reasoning & Variational Inference

Foundations | Tricks | Algorithms

Shakir Mohamed
Research Scientist, DeepMind
@shakir_za