Unsupervised Learning and Generative Models

Shakir Mohamed

Staff Research Scientist, DeepMind

@shakir_za

UCL Course on
Advanced Topics in Machine Learning
21 March 2017

Why Unsupervised Learning

Move beyond associating inputs to outputs

Understand and imagine how the world evolves

Recognise objects in the world and their factors of variation

Detect surprising events in the world

Establish concepts as useful for reasoning and decision making

Imagine and generate rich plans for the future

Part of a suite of complementary learning systems

Lecture Overview

Part I
Probabilistic Machine Learning

Types of Generative Models

Part II
Learning in Prescribed Probabilistic Models

Families of Approximate Posterior Distributions

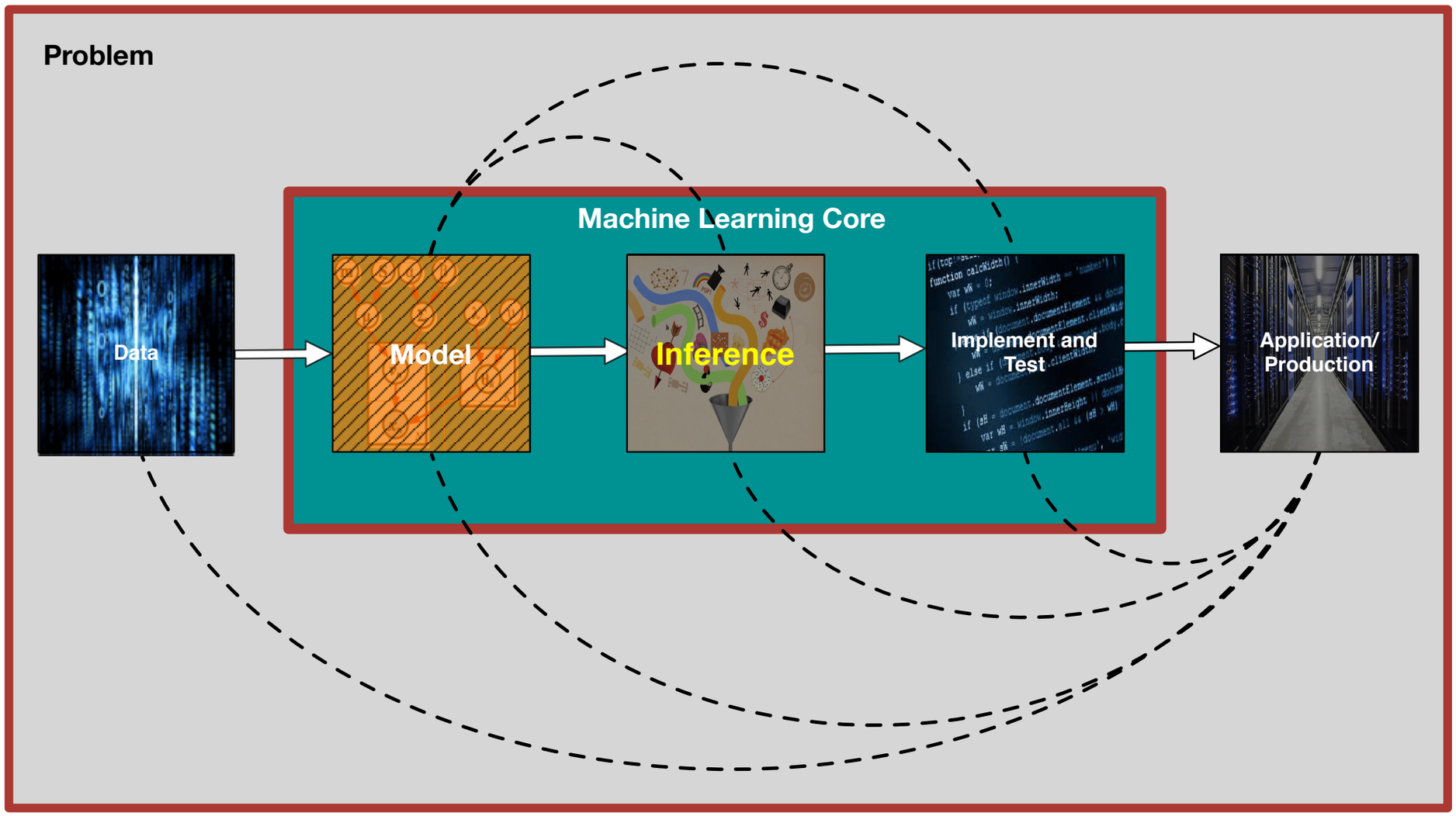
Variational Optimisation

Part III
Learning In Implicit Probabilistic Models

Part IV
Applications and Extensions of Generative Models

Probabilistic Machine Learning

Modelling and Box's Loop



In probabilistic models, we must reason over the probability of events.

Statistical Inference

Any mechanism by which we deduce the probabilities in our model based on data.

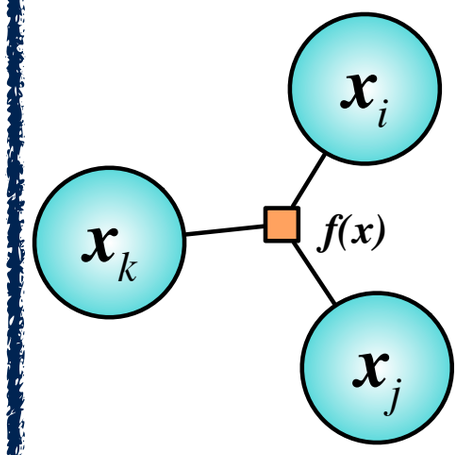
Thinking about ML

3. Algorithms

1. Models

2. Learning Principles

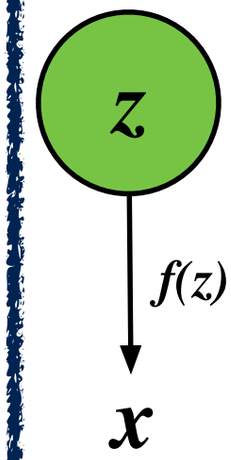
Types of Generative Models



Fully-observed models

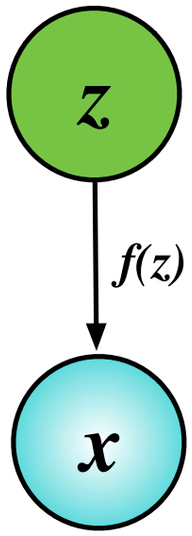
Model observed data directly without introducing any new unobserved local variables.

Models



Implicit Models

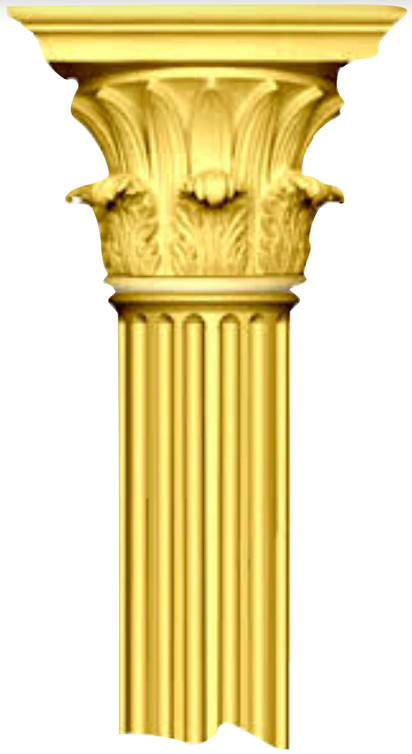
Simulator-based models; models data as a transformation of an unobserved noise source using a parameterised function.



Latent variable models

Introduce an unobserved random variable for every observed data point to explain hidden causes.

Smorgasbord of Learning Principles

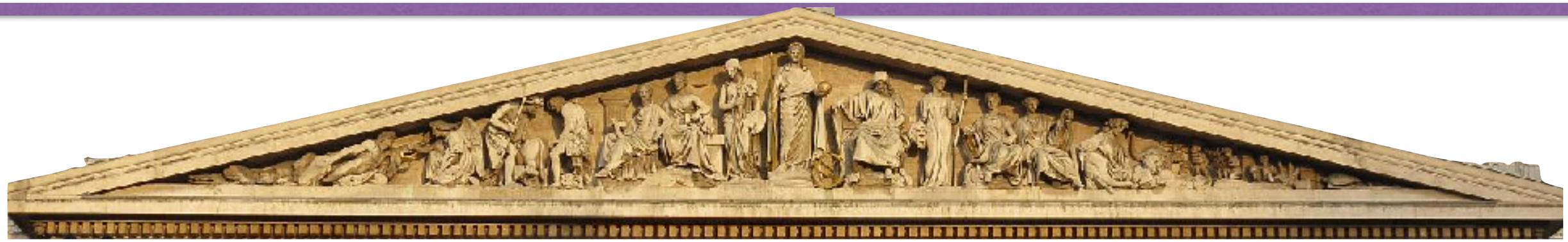


Learning Principles

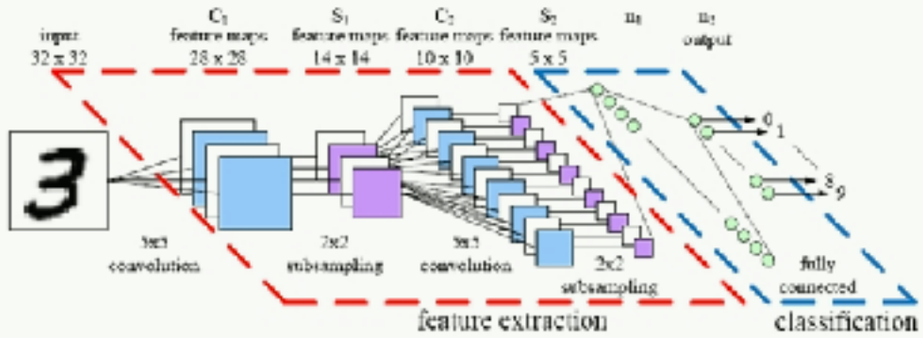
For a given model, there are many competing inference methods.

- Exact methods (conjugacy, enumeration)
- Numerical integration (Quadrature)
- Generalised method of moments
- **Maximum likelihood (ML)**
- **Maximum a posteriori (MAP)**
- Laplace approximation
- Integrated nested Laplace approximations (INLA)
- **Expectation Maximisation (EM)**
- Monte Carlo methods (MCMC, SMC, ABC)
- Noise contrastive estimation (NCE)
- Cavity Methods (EP)
- **Variational methods**

Combining Models and Inference

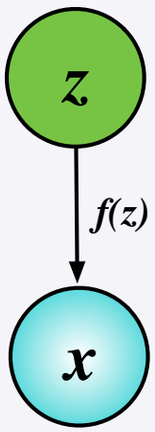


A given model and learning principle can be implemented in many ways.



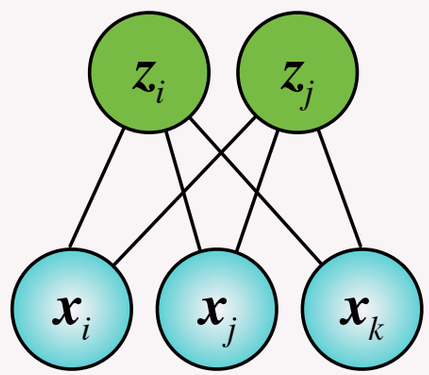
Convolutional neural network
+ penalised maximum likelihood

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



Latent variable model
+ variational inference

- VEM algorithm
- Expectation propagation
- Approximate message passing
- *Variational auto-encoders*



Restricted Boltzmann Machine
+ maximum likelihood

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

Consolidation Questions

- Think through your may machine learning solution that interests you in the framework of models-inference-algorithm. What advantages, for you, does this structured approach to reasoning about learning systems offer. Are there ways in which it hinders you thinking and understanding.
- Maximum likelihood is the dominant learning principle in machine learning. What are the problems with maximum likelihood? Are there ways to improve it?
- Question 3.4 in PMRL
- Question 9.19 in ITILA
- Question 4.4 in BRML

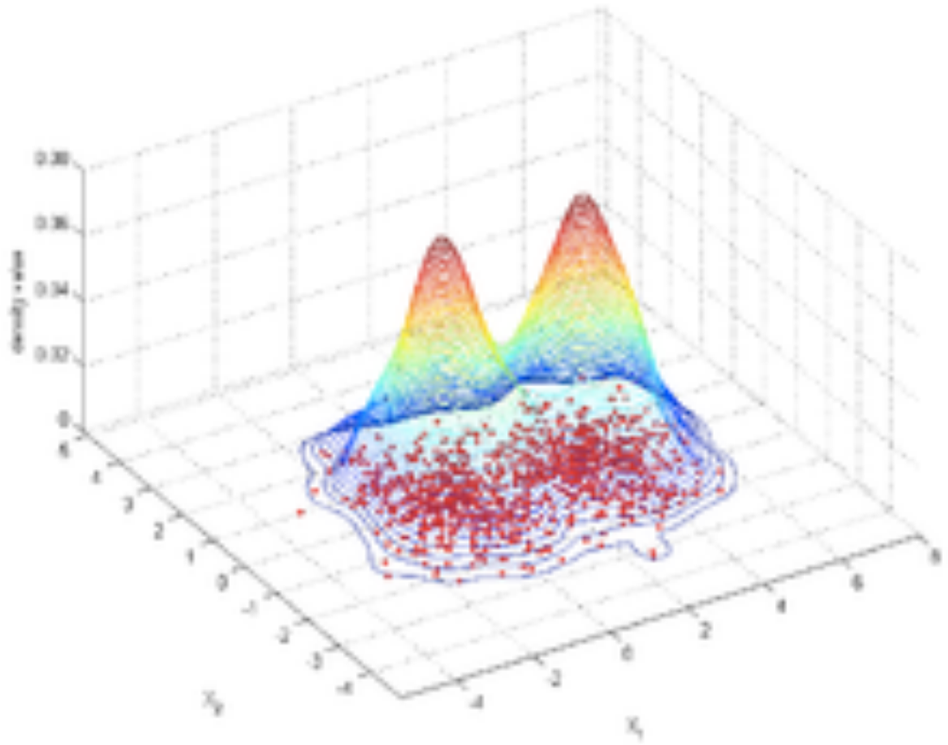
Types of Generative Models

Density Estimation

Construct and estimate of a probability density function $p(x)$ from observed data.

Already familiar with many methods:

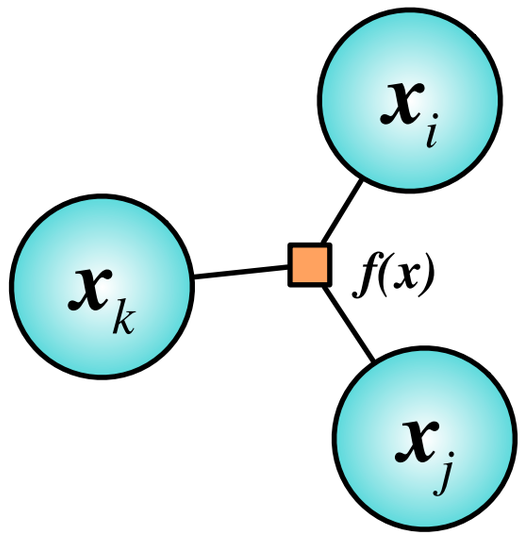
- Histograms
- Kernel Density Estimation (KDE)
- PCA and Factor Analysis
- Mixture models

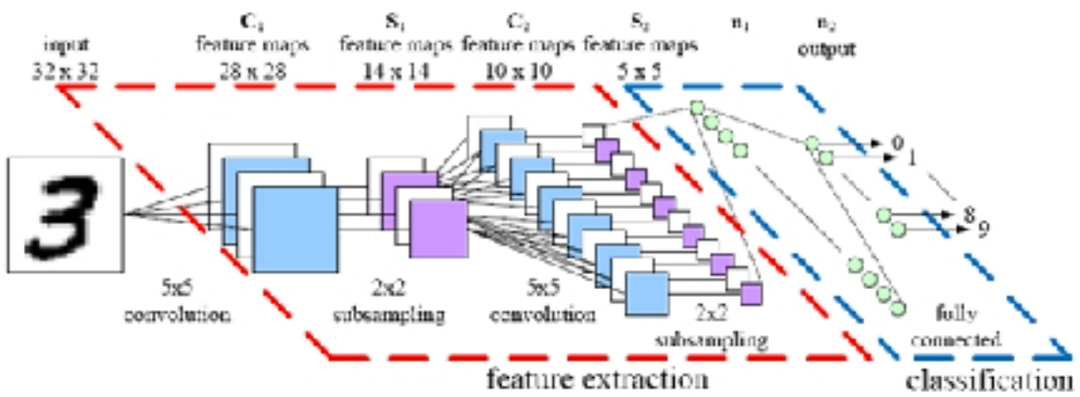


Types of Models

Unconditional models

- Unsupervised learning
- Learn $p(x)$, no targets or labels
- Density estimation
- What is commonly meant by generative model





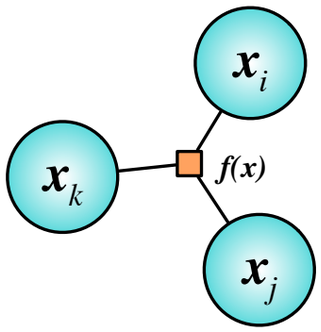
Conditional models

- Supervised learning
- Learn $p(y|x)$ for observed x,y
- Regression and classification
- Conditional density estimation:

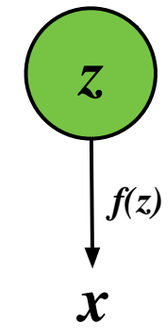
$p(x|c)$, *context* c (labels, states, actions)

Every probabilistic model is a generative model, whether conditional or not.

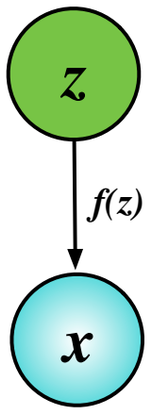
Types of Generative Models



Fully-observed models



Implicit models



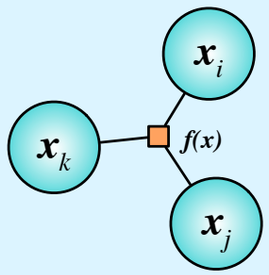
Latent variable models

Design Dimensions

- ***Data:*** binary, real-valued, nominal, strings, images.
- ***Dependency***: independent, sequential, temporal, spatial.
- ***Representation***: continuous or discrete
- ***Dimension***: parametric or non-parametric

- Computational complexity
- Modelling capacity
- Bias, uncertainty, calibration
- Interpretability

Fully-observed Models



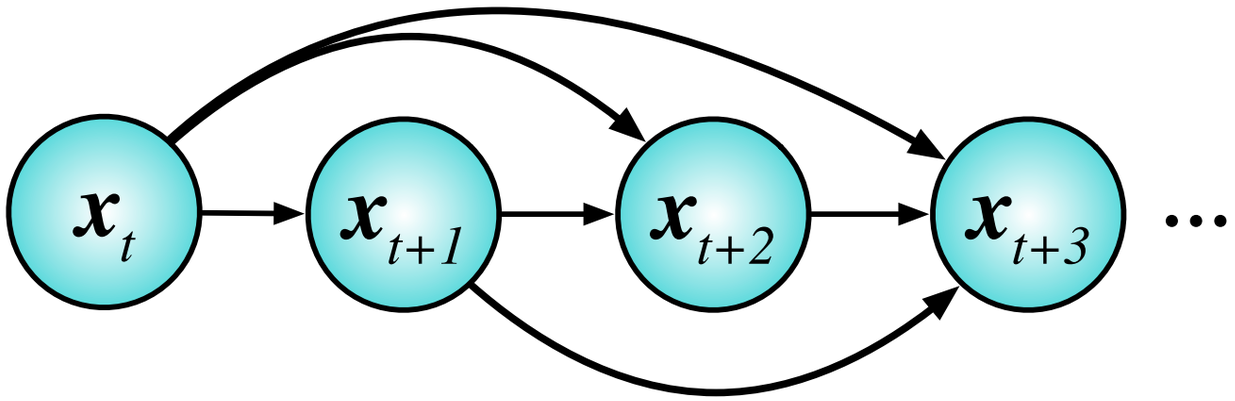
Fully-observed models

*Model observed data directly **without** introducing any new unobserved **local variables**.*

*Model Parameters are **global variables.***

*Stochastic activations & unobserved random variables are **local variables.***

Markov Models



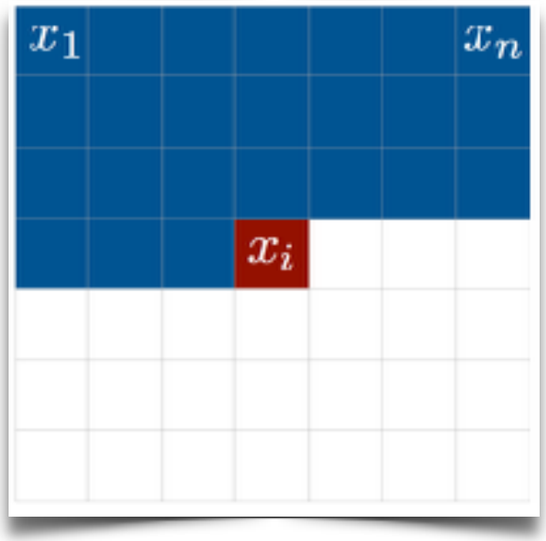
$$x_1 \sim \text{Cat}(x_1|\pi)$$

$$x_2 \sim \text{Cat}(x_2|\pi(\mathbf{x}_1))$$

$$\cdots$$

$$x_i \sim \text{Cat}(x_i|\pi(\mathbf{x}_{<n}))$$

$$p(\mathbf{x}) = \prod_i p(x_i|f(\mathbf{x}_{<i}; \boldsymbol{\theta}))$$



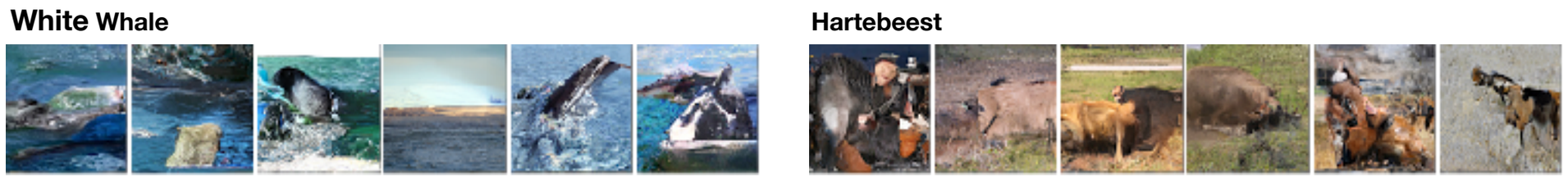
All conditional probabilities described by deep networks.

Fully-observed Models

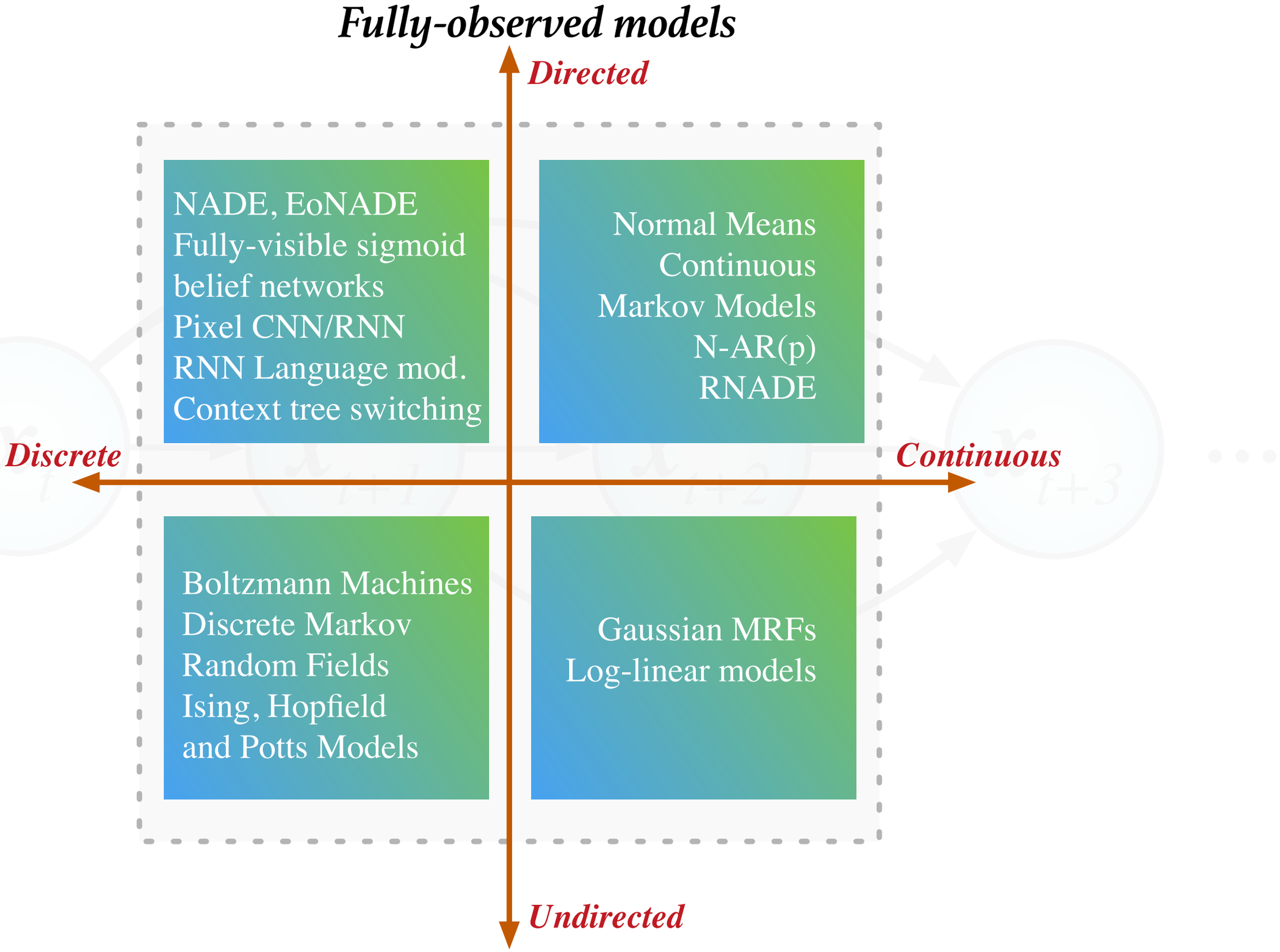
Properties

- + Can directly encode how observed points are related.
- + *Any data* type can be used
- + For directed graphical models:
  - + **Parameter learning simple:** Log-likelihood is directly computable, no approximation needed.
  - + Easy to scale-up to large models, many optimisation tools available.
  - - Order sensitive.
- - For undirected models,
  - - **Parameter learning difficult:** Need to compute normalising constants.
- - **Generation can be slow:** iterate through elements sequentially, or using a Markov chain.

Pixel CNN



Model-space Visualisation

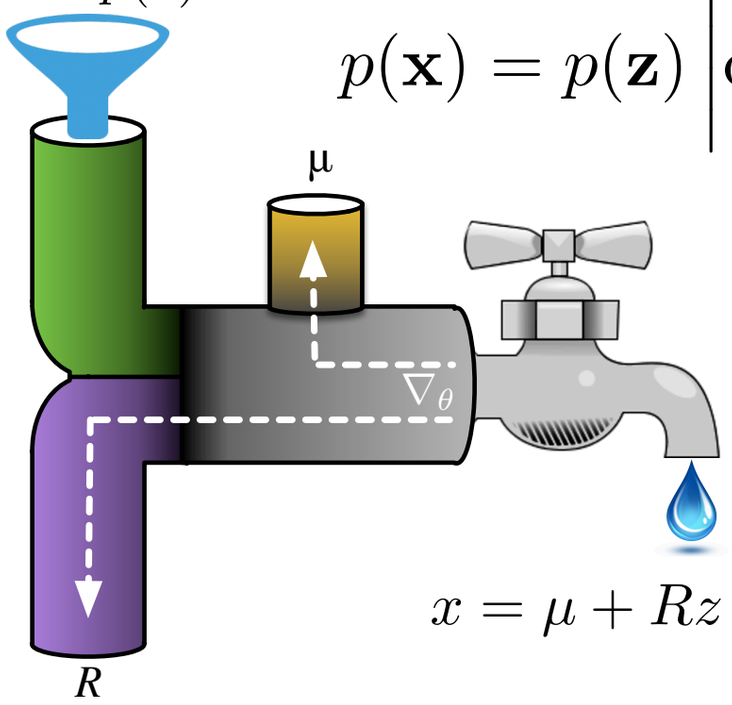


Implicit Models

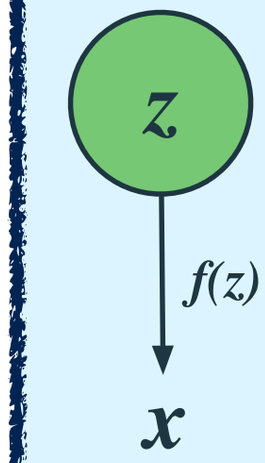
Change of variables for invertible functions

$$z \sim p(z)$$

$$p(\mathbf{x}) = p(\mathbf{z}) \left| \det \frac{\partial f}{\partial \mathbf{z}} \right|^{-1}$$



$$x = \mu + Rz$$



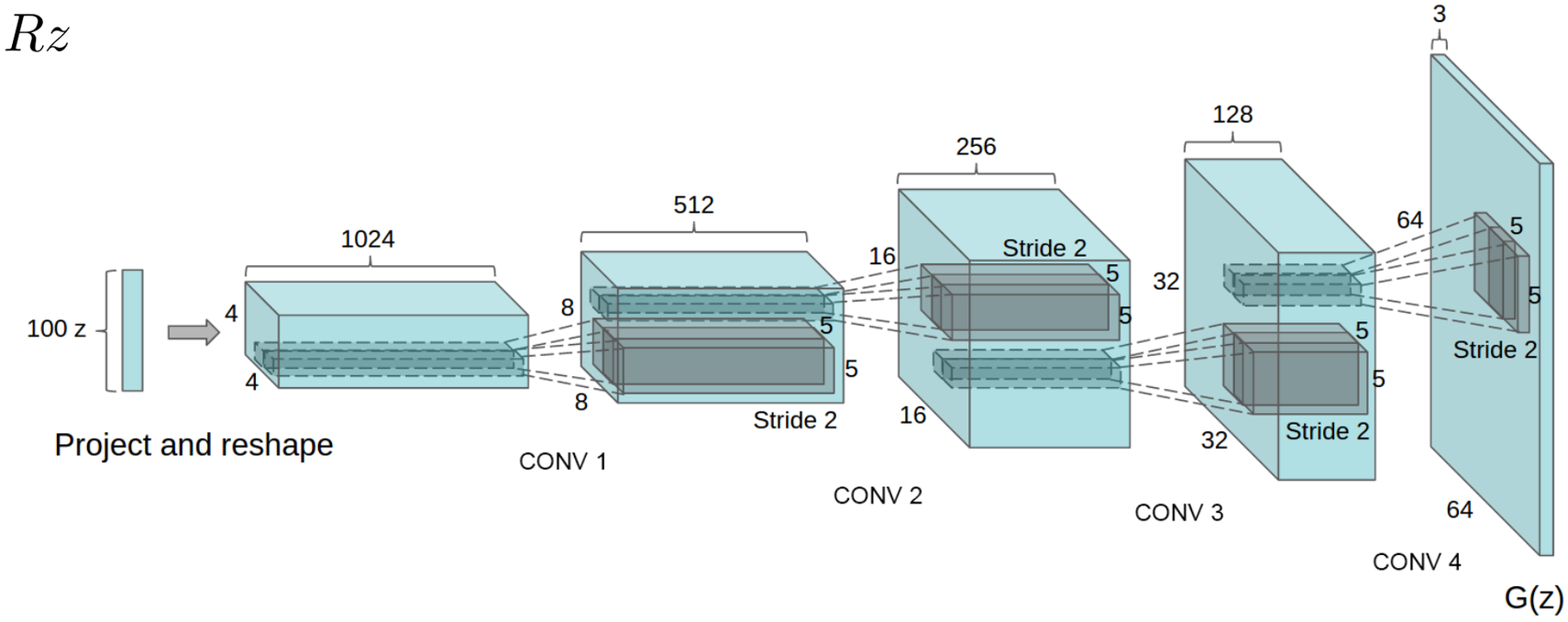
Implicit models

*Transform an unobserved **noise source** using a parameterised **function**.*

Generator Networks

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\mathbf{x} = f(\mathbf{z}; \boldsymbol{\theta})$$



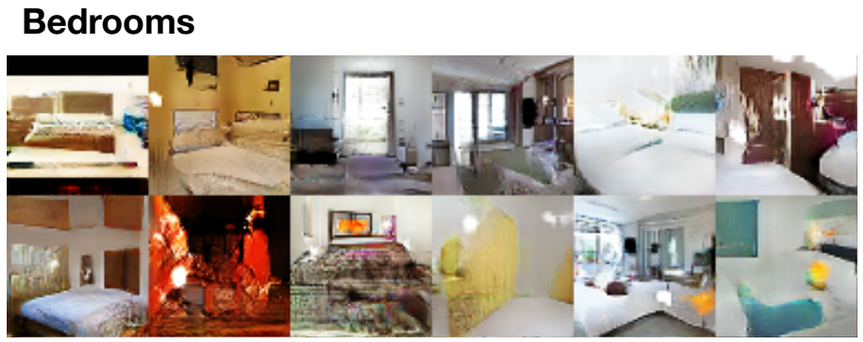
The transformation function is parameterised by a linear or deep network (fully-connected, convolutional or recurrent).

Implicit Models

Properties

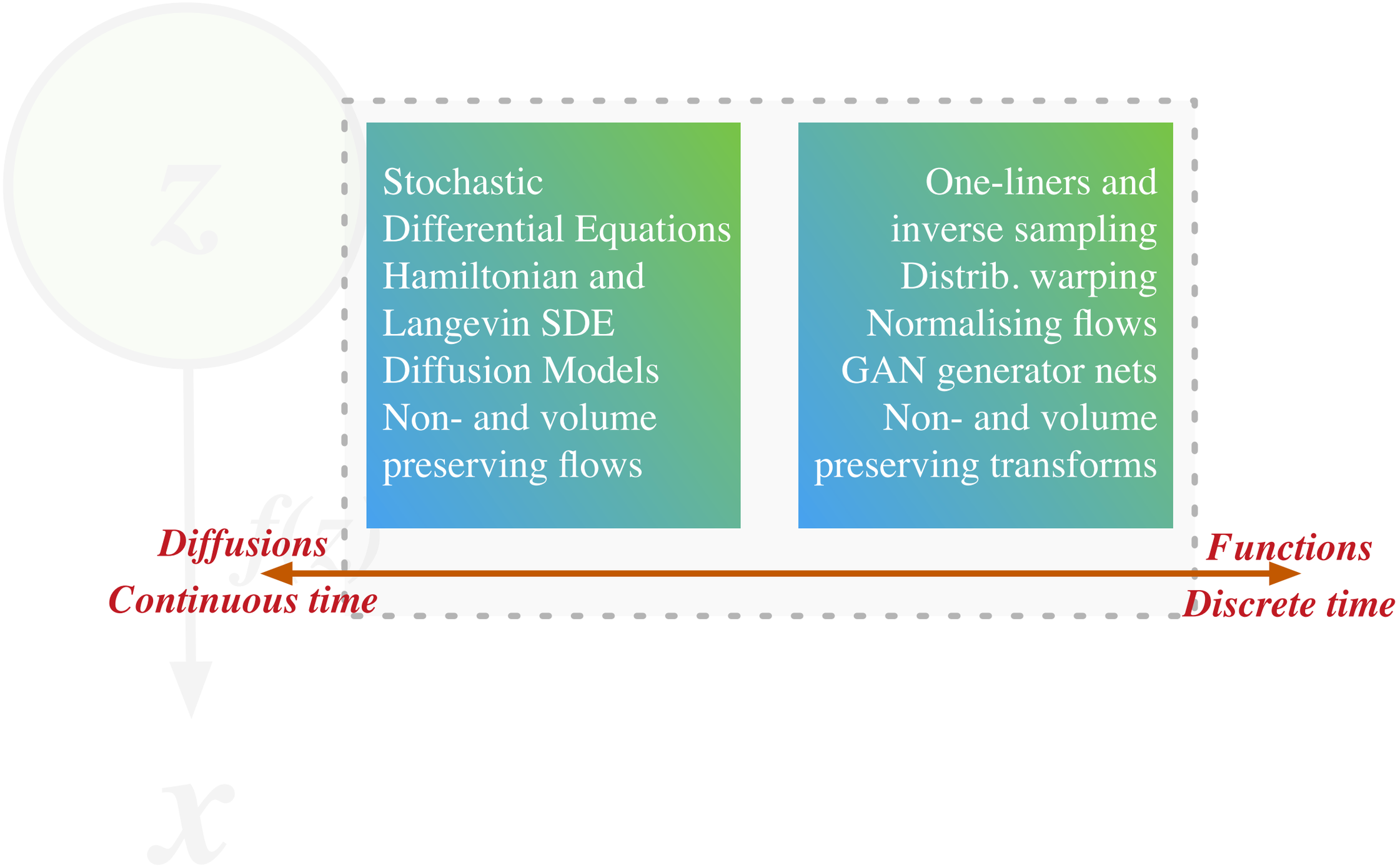
+ Easy sampling, and natural to specify.
+ Easy to compute expectations without knowing final distribution.
+ Can exploit with large-scale classifiers and convolutional networks.
- ***Difficult to satisfy constraints***: Difficult to maintain invertibility, and challenging optimisation.
- ***Lack of noise model*** (likelihood):
  - Difficult to extend to generic data types
  - Difficult to account for noise in observed data.
  - Hard to compute marginalised likelihood for model scoring, comparison and selection.

Convolutional generative adversarial network



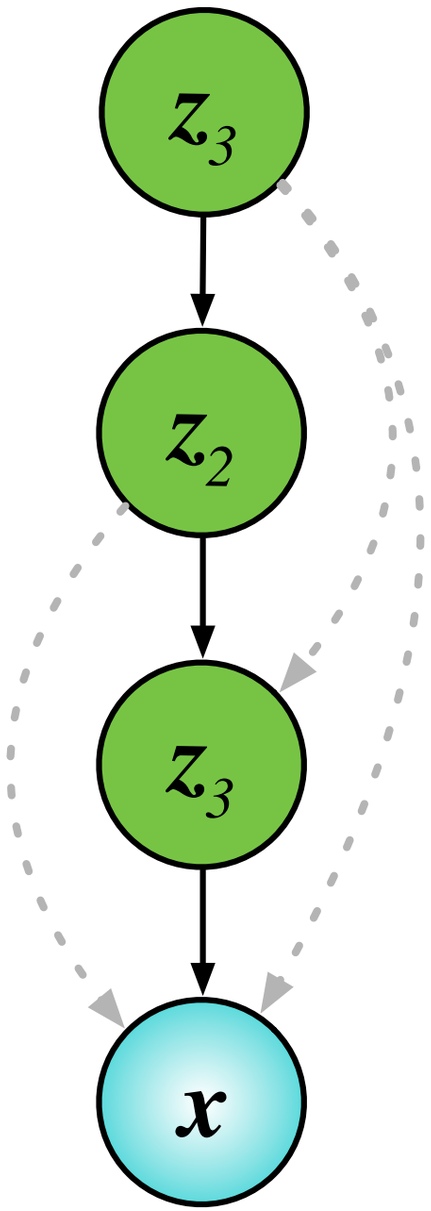
Model-space Visualisation

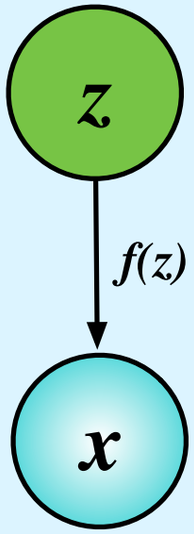
Implicit Probabilistic models



Latent Variable Models

Deep Latent Gaussian Model





Latent variable models

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

$$\mathbf{z}_3 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\mathbf{z}_2|\mathbf{z}_3 \sim \mathcal{N}(\mu(\mathbf{z}_3), \Sigma(\mathbf{z}_3))$$

$$\mathbf{z}_1|\mathbf{z}_2 \sim \mathcal{N}(\mu(\mathbf{z}_2), \Sigma(\mathbf{z}_2))$$

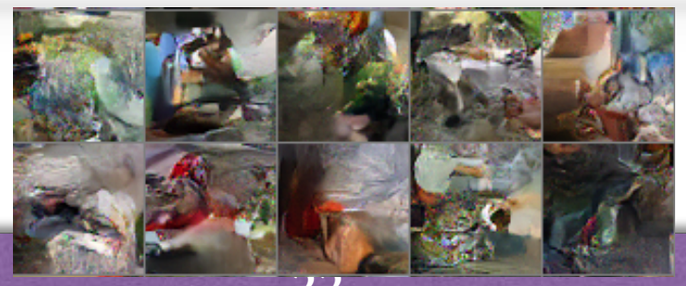
$$\mathbf{x}|\mathbf{z}_1 \sim \mathcal{N}(\mu(\mathbf{z}_1), \Sigma(\mathbf{z}_1))$$

Latent Variable Models

Properties

+ Easy sampling.
+ Easy way to include hierarchy and depth.
+ Easy to encode structure believed to generate the data
+ Avoids order dependency assumptions: marginalisation of latent variables induces dependencies.
+ Latents provide compression and representation the data.
+ Scoring, model comparison and selection possible using the marginalised likelihood.
- Inversion process to determine latents corresponding to a input is difficult in general
- Difficult to compute marginalised likelihood requiring approximations.
- Not easy to specify rich approximations for latent posterior distribution.

Convolutional DRAW



Model-space Visualisation

Latent variable models

Cascaded Indian Buffet process
Hierarchical Dirichlet process

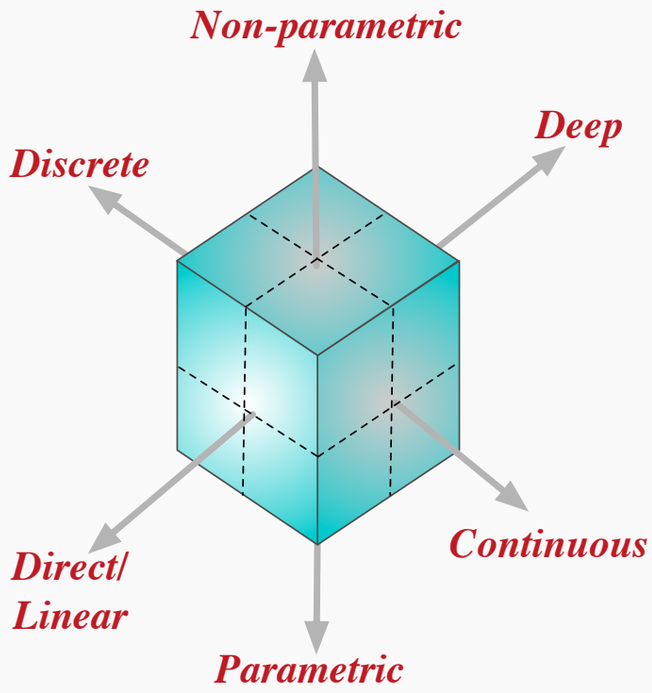
Deep Nonparametric Discrete

Sigmoid Belief Net
Deep auto-regressive networks (DARN)

Deep Parametric Discrete

Indian buffet process
Dirichlet process mixture

Direct Nonparametric Discrete



Deep Gaussian processes
Recurrent Gaussian Process
GP State space model

Deep Nonparametic Continuous

Hidden Markov Model
Discrete LVM
Sparse LVMs

Linear Parametric Discrete

Nonlinear factor analysis
Nonlinear Gaussian belief network
Deep Latent Gaussian (VAE, DRAW)

Deep Parametric Continuous

PCA, factor analysis
Independent components analysis
Gaussian LDS
Latent Gauss Field

Linear Parametric Continuous

Gaussian process LVM

Direct Nonparametric Continuous

Consolidation Questions

- Models can discussed in other ways.
  - One alternative taxonomy is of directed, undirected, bi-directed, and mixed graphs. What are the properties of these different models? Find examples of such models.
  - A further taxonomy is of graphical, chordal, hierarchical and non-hierarchical models. Again, what are the properties of such models? How do the examples you used above related to this taxonomy. Be clear about any advantages and disadvantages that stem from the type of model, the types of inference that are possible, or the the algorithmic conveniences that your choice exposes.
- Think of the models-space visualisation. Are any models you know missing. Where do they fit in? Would you have used a different set of axes?
- What are the protocols for evaluation that we use in machine learning? The common approach you have used before is known as inductive testing. What is transductive testing? Look up prequential testing, posterior predictive checks, risk minimisation, and utility theory for other examples and language.

Part II

Inference in Prescribed Probabilistic Models

Inference Problems

Common inference problems are:

Evidence Estimation

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

Moment Computation

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

Prediction

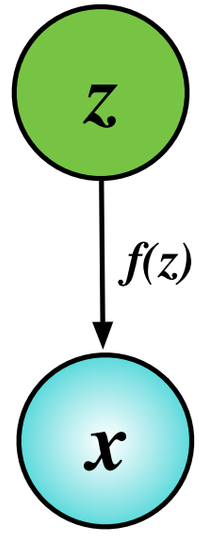
$$p(\mathbf{x}_{t+1}) = \int p(\mathbf{x}_{t+1}|\mathbf{x}_t) p(\mathbf{x}_t) d\mathbf{x}_t$$

Testing

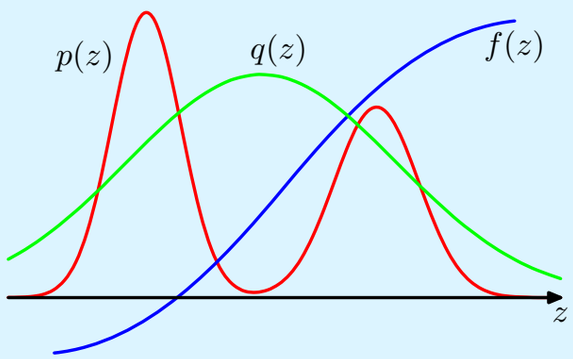
$$\mathcal{B} = \log p(\mathbf{x}|H_1) - \log p(\mathbf{x}|H_2)$$

Bayesian 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.



We take steps to improve the model evidence for given data samples.



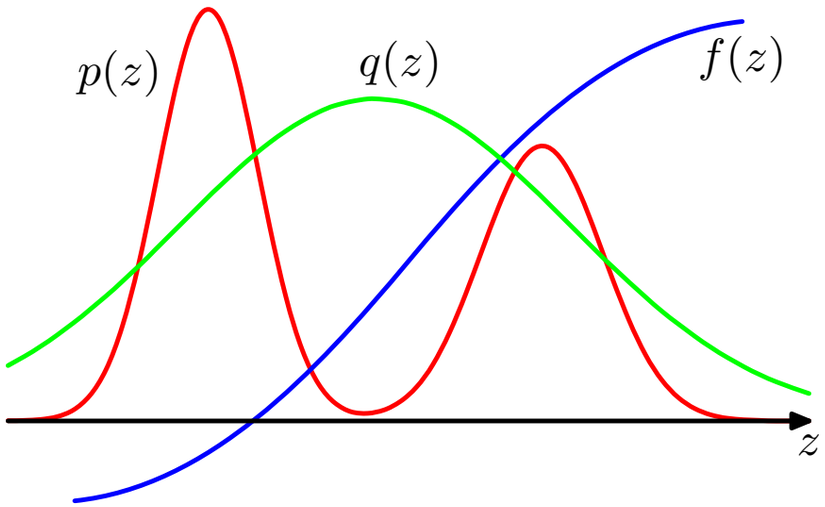
Learning principle: Model Evidence

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

Integral is intractable in general and requires approximation.

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

Importance Sampling



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

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

Notation

Always think of $q(\mathbf{z}|\mathbf{x})$ but often will write $q(\mathbf{z})$ for simplicity.

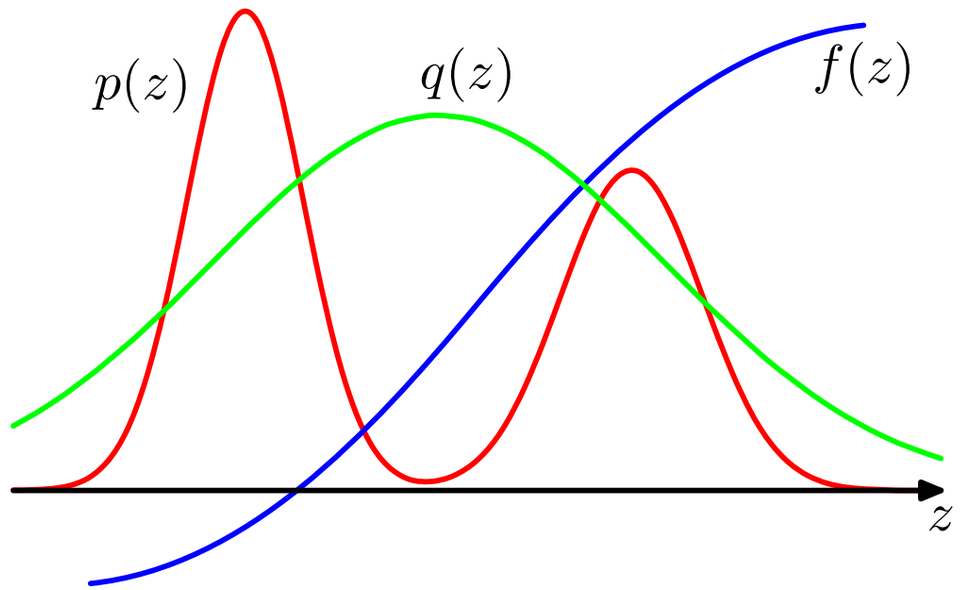
Conditions

- $q(z|x)>0$, when $f(z)p(z) \neq 0$.
- Easy to sample from $q(z)$.

Importance Sampling

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

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



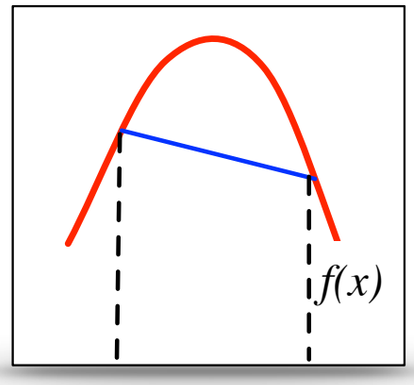
Can we take inspiration from importance sampling, but instead:

- Obtain a ***deterministic*** algorithm,
- Scale-up to ***high-dimensional and large data*** problems,
- Easy **convergence assessment**.

Now, from importance sampling to variational inference ...

Jensen's Inequality

An important result from convex analysis:



For concave functions *f(.)*

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

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

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

Instead of Monte Carlo Integration, use Jensen's inequality.

Importance Sampling 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

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

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

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

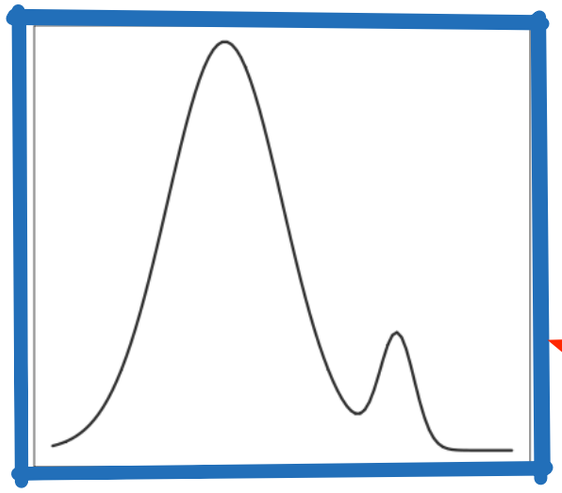
Variational lower bound

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

What is a Variational Method?

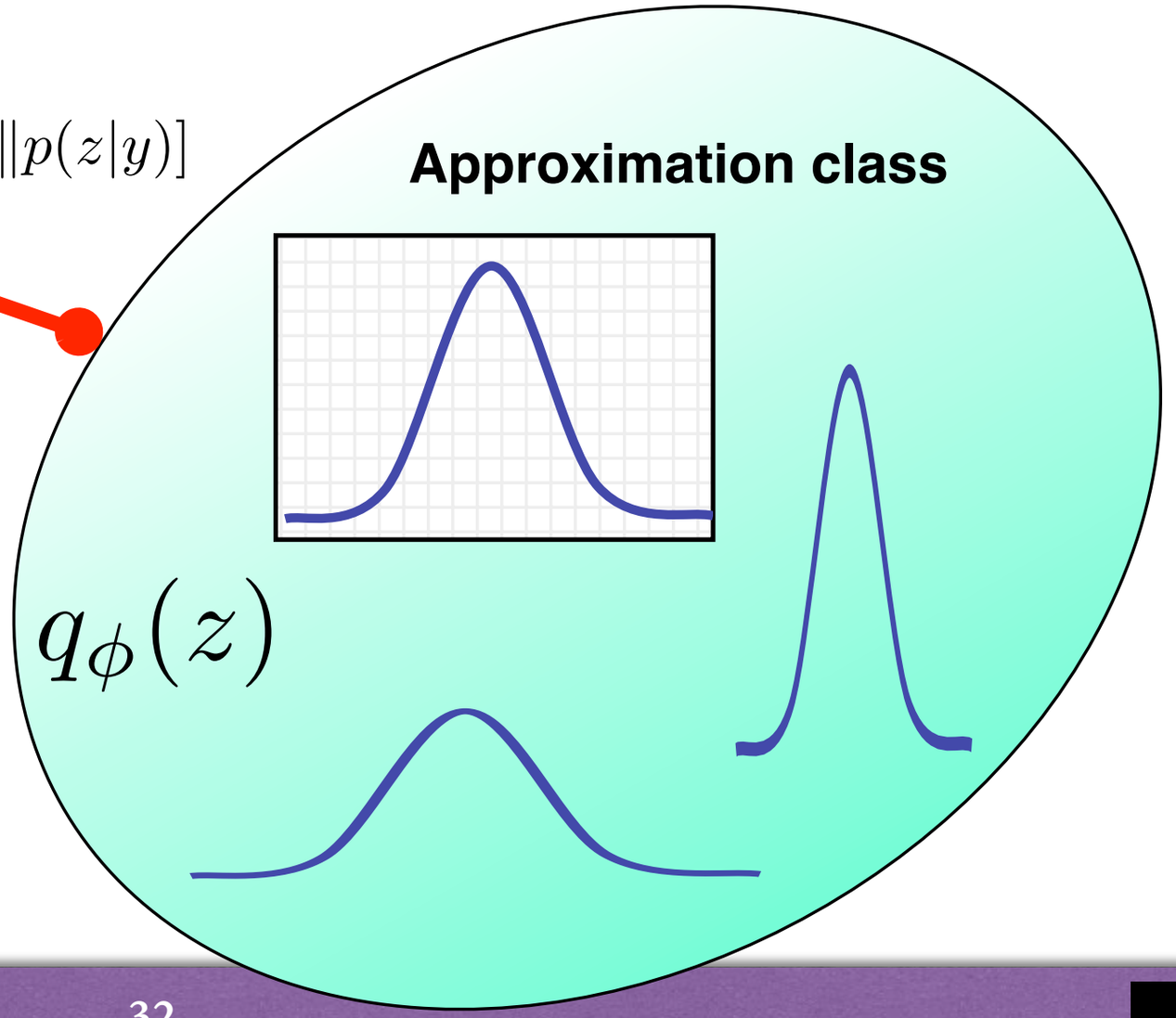
Variational Principle

General family of methods for approximating complicated densities by a simpler class of densities.



True posterior

$$KL[q(z|y)\|p(z|y)]$$



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. $p(x)$

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

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

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

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

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

Variational Inference

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

This bound is exactly of the form we are looking for.

- **Variational free energy:** We obtain a functional and are free to choose the distribution $q(z)$ that best matches the true posterior.
- **Evidence lower bound (ELBO):** principled bound on the marginal likelihood, or model evidence.
- Certain choices of $q(z)$ makes this quantity easier to compute. Examples to come.

Variational Inference

Interpreting the 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 Inference

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

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, $q(z|y)$.
  - *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.
  - Optimisation allows us to *tighten the bound* and get as close as possible to the true marginal likelihood.

Minimum Description Length (MDL)

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

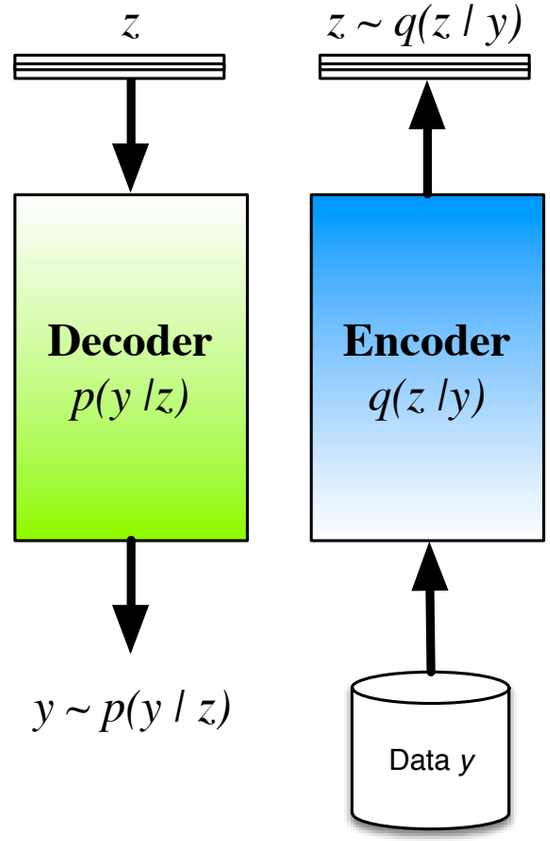
Stochastic encoder

Data code-length

Hypothesis code

Stochastic encoder-decoder systems implement variational inference.

- Regularity in our data that can be explained with latent variables, implies that the data is compressible.
- MDL: inference seen as a problem of compression — we must find the ideal shortest message of our data y: marginal likelihood.
- Must introduce an approximation to the ideal message.
- **Encoder:** variational distribution $q(z|y)$,
- **Decoder:** likelihood $p(y|z)$.



Variational Inference vs. Variational Bayes

$$p(\mathbf{x}|\mathbf{z},\boldsymbol{\theta})p(\mathbf{z},\boldsymbol{\theta})$$

Variational Inference (VI)

Apply the variational principle only to some parts of the model.

Widely-used case: latent variables are assigned probability distributions; maximum likelihood estimates for others.

$q(\mathbf{z})$ θ_{ML}

Inference *Learning*

Variational Bayesian Inference (VB)

All unknown quantities are probability distributions and use a variational approximation for all posterior distributions.

$q(\mathbf{z},\boldsymbol{\theta}|\mathbf{x})$

Inference

Why Variational Inference?

Disadvantages:

- **Choice of posterior**—can be hard to choose a rich and tractable family of approximate posterior distributions.
- An **approximate posterior** only—not always guaranteed to find exact posterior in the limit.
- **Difficulty in optimisation**—can get stuck in local minima.
- Can **under-estimates the variance** of the posterior and can bias maximum likelihood parameter estimates if using limited approximation.
- **Limited theory** and guarantees for variational methods.

Why Variational Inference?

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)

Other 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.

Consolidation Questions

- Question 1.29 in PMRL.
- Question 1.34 in PMRL.
- Explain to a colleague the difference between type-II maximum likelihood, evidence maximisation, and fully-Bayesian inference, if any.
- How have you used importance sampling in reinforcement learning.
- Jensen's inequality is used widely in convex optimisation. Can you give a geometric understanding of Jensen's inequality. Where else is this inequality used.
- Planning problems in reinforcement learning have a dual view as inference problems. These approaches are often called planning-as-inference or KL-control. Find papers that describe this and connect their reasoning to your understanding of inference and variational approximations.
- Can you derive the Renyi-based variational objective?
- The variational free energy is just one of many objectives that can be used. In other parts of statistical model you may have come across Bethe's free energy, Kikuchi's free energy, the generalised free energy. How are these related.
- Instead of a lower bound on the log-marginal likelihood, we could try to derive upper-bounds on the marginal likelihood and minimise this. Read-up on tree-reweighted partitioning.
- What is Holder's inequality.

Families of Approximate Posterior Distributions

Free-form and Fixed-form

Free-form variational method solves for the exact distribution setting the functional derivative to zero.

$$\frac{\delta \mathcal{F}(x, q)}{\delta q(z)} = 0 \quad s.t. \int q(z) dz = 1$$

$$q(z) \propto p(z) \exp(\log p(x|z, \theta))$$

Great! The optimal solution is the true posterior distribution.

But solving for the normalisation is our original problem.

Fixed-form variational method specifies an explicit form of the q-disribution.

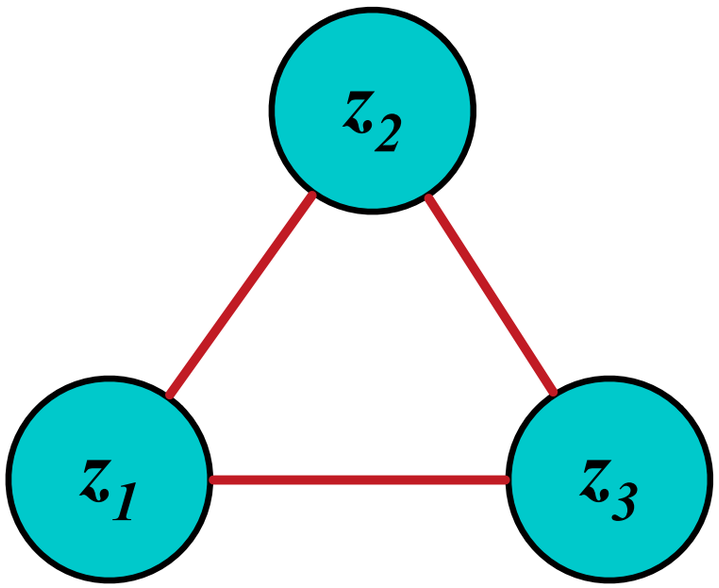
$$q_\phi(z) = f(z; \phi)$$

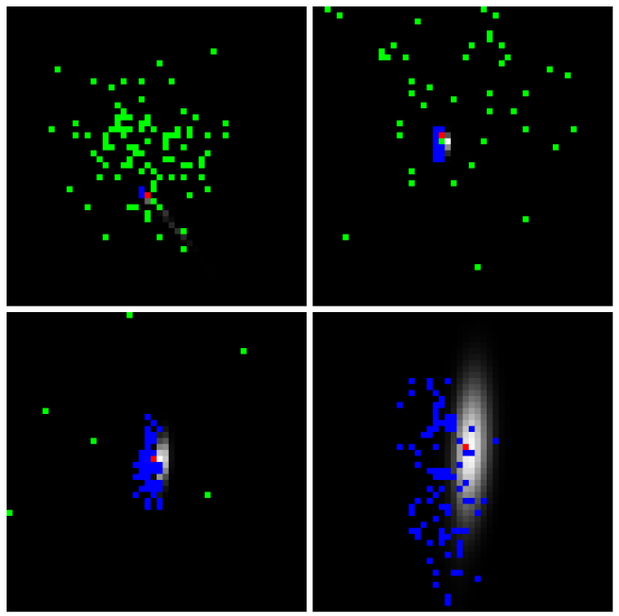
This is ideally a rich class of distributions. Parameters ϕ are called variational parameters.

Mean-field Variational Inference

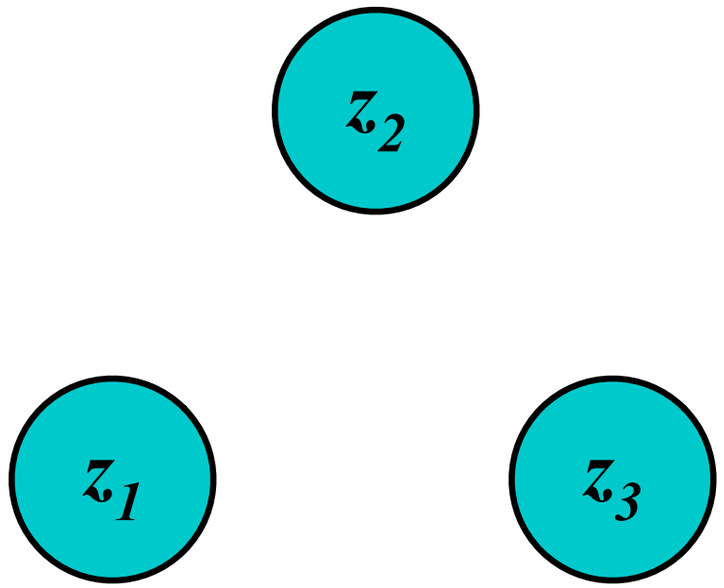
Mean-field methods assume that the distribution is factorised.

True Posterior





Fully-factorised



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

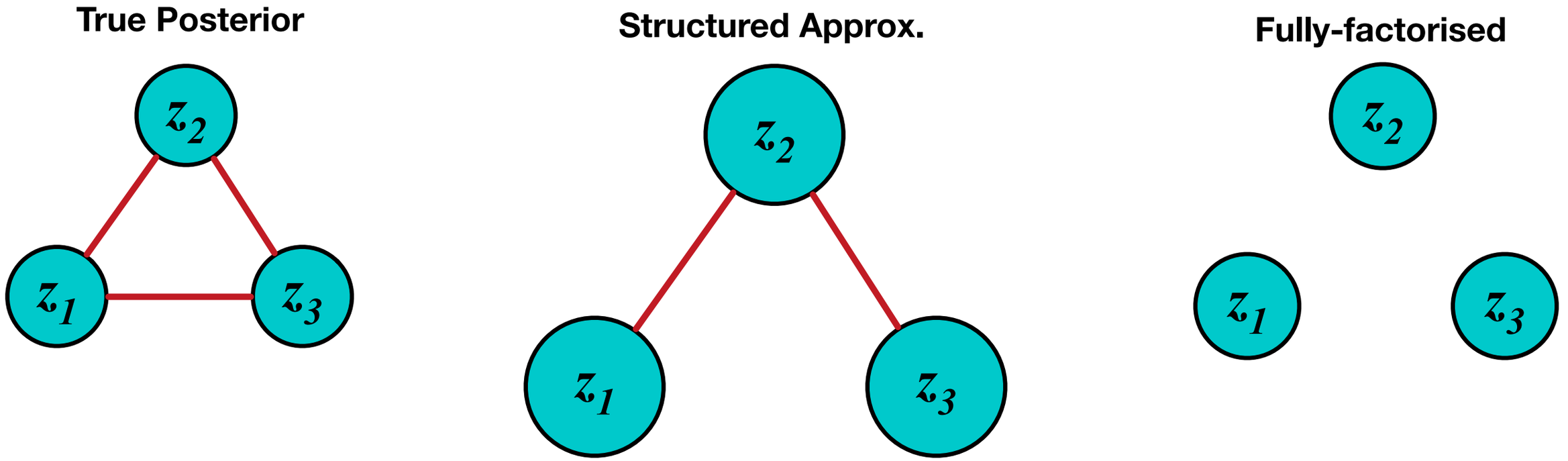
$$q^*(z|x) \propto p(x|z)p(z)$$

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

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

Structured Mean Field

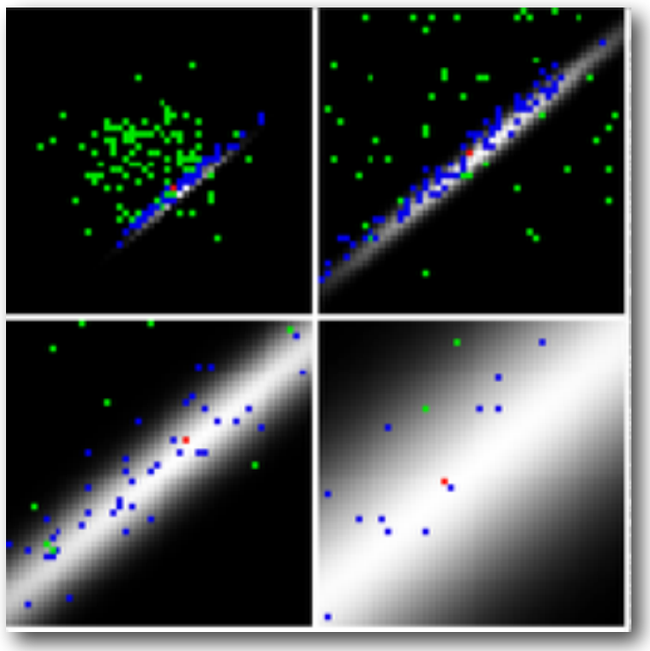
Structured mean-field: introduce dependencies into our factorisation.



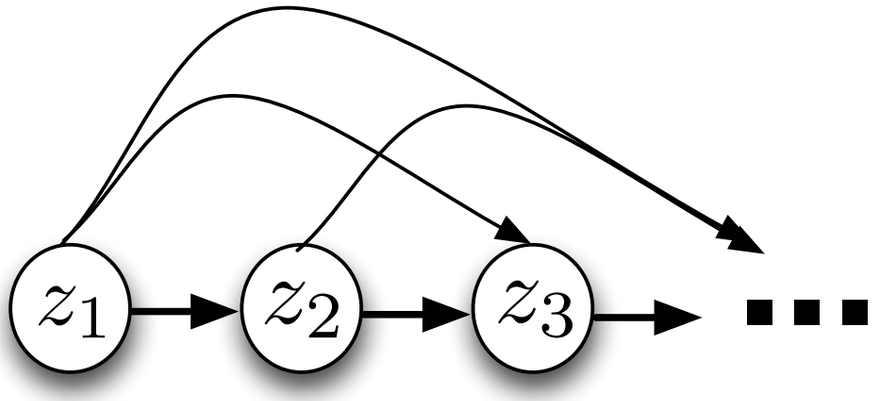
Most Expressive ← → *Least Expressive*

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

Structured Mean Field



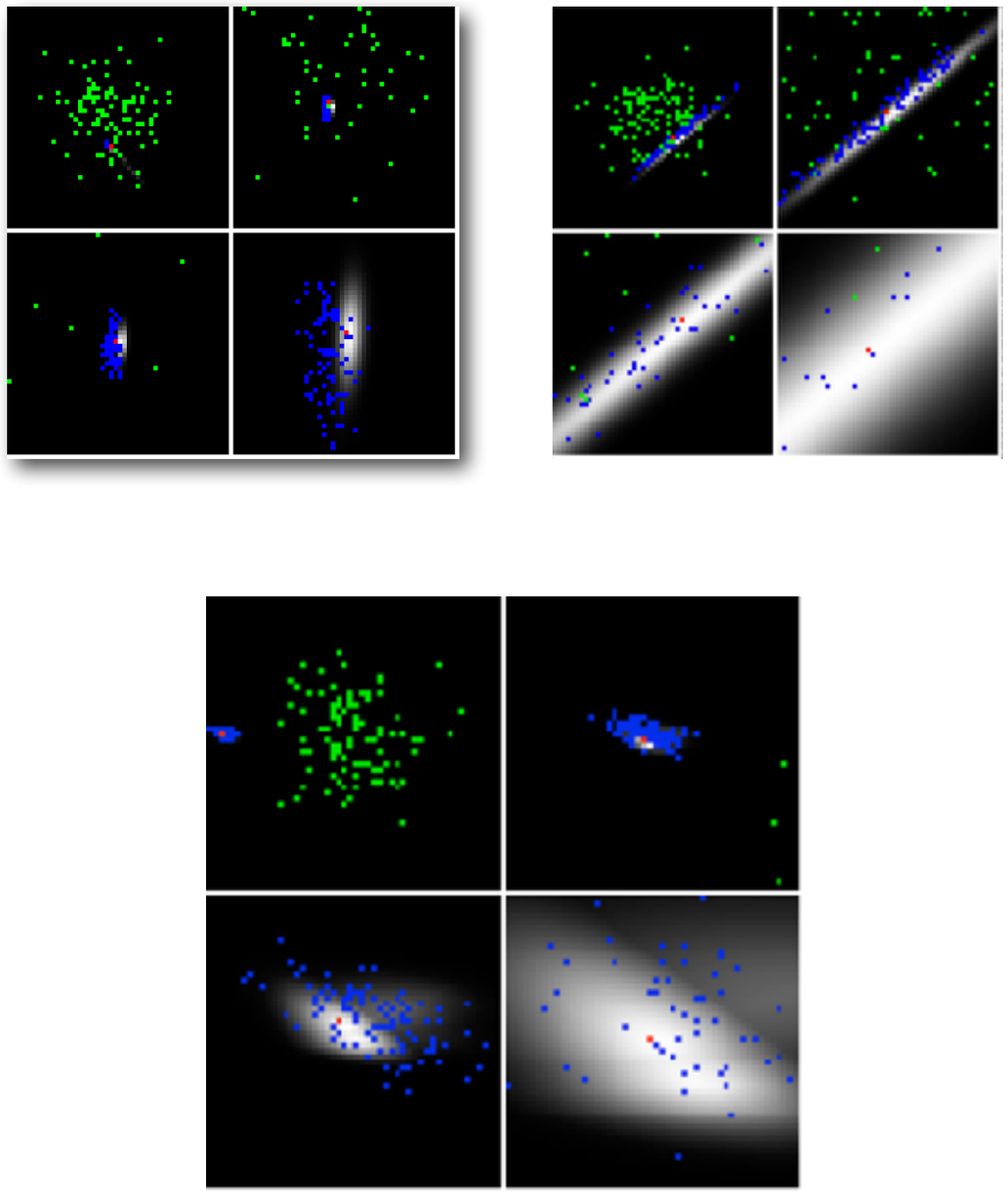
Autoregressive approximation: One very useful and powerful structured specification is to condition on all previous variables.



$$q(\mathbf{z}) = \prod_i q_i(z_i|z_{<i})$$

Fixed-form Approximations

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



Variational Latent Gaussian Models

Examples: GP regression, BXPCA or DLGM.

$$z \sim \mathcal{N}(z|0,1) \qquad y \sim p(y|f_\theta(z)) \qquad q(z) = \prod_i \mathcal{N}(z_i|\mu_i, \sigma_i^2)$$

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

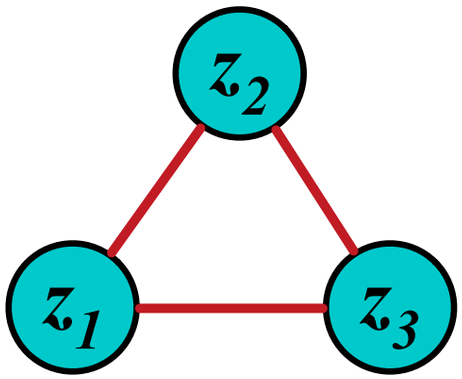
$$\mathcal{F}(y,q) = \mathbb{E}_{q(z)}[\log p(y|z)] - \sum_i KL[q(z_i)\|p(z_i)]$$

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

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

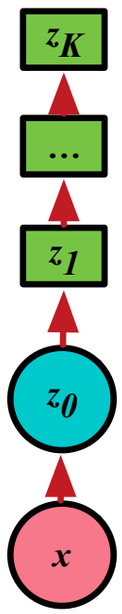
Families of Approximations

True Posterior

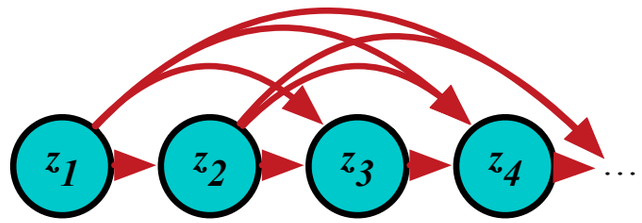


Families of Posterior Approximations

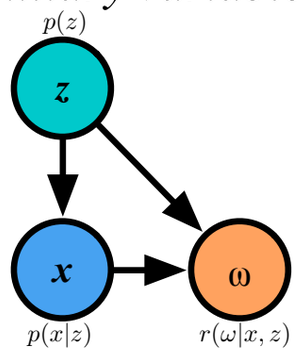
Normalising flows



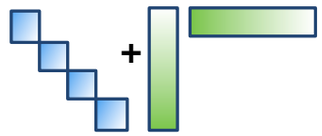
Structured mean-field



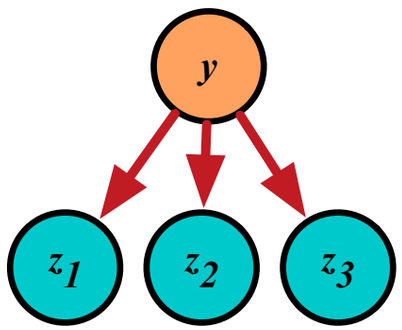
Auxiliary variables



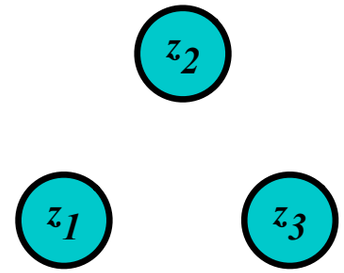
Covariance models



Mixtures



Fully-factorised



Most Expressive ← → ***Least Expressive***

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

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

Consolidation Questions

- Change of variables: Question 8.10 in BRML
- KL divergence: Q8.39, Q8.33 in BRML
- Compute the KL divergence between two members of the same exponential family of distributions. Use specific examples at the start (two Dirichlet distributions, or other favourite dirtibutions).
- How do you compute the KL divergence by Monte Carlo integration.
- Form an intuitive understanding of the change of variables for probability, and how this is extended to the concept of a normalising flow.
- Write out a latent variable model with alternative prior distributions. Consider a Dirichlet (then called a partial membership model), and Bernoulli distribution, or an autoregressive Bernoulli distribution.
- Can you transform other models, such as classifiers, into Bayesian models, by adding prior distributions. How would you apply variational inference in these settings.

Variational Optimisation

Optimising the Variational Objective

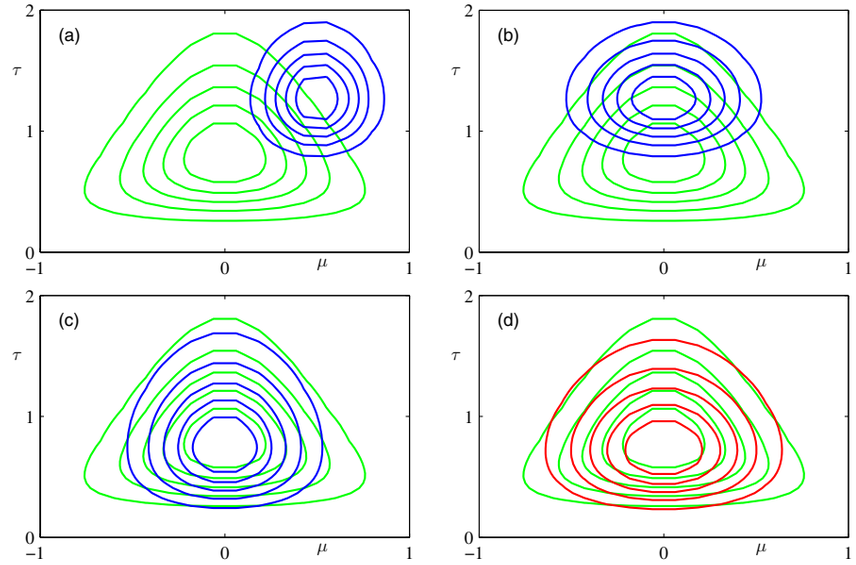
$$\mathcal{F}(\mathbf{x}, q) = \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - KL[q(\mathbf{z})\|p(\mathbf{z})]$$

Approx. Posterior

Reconstruction

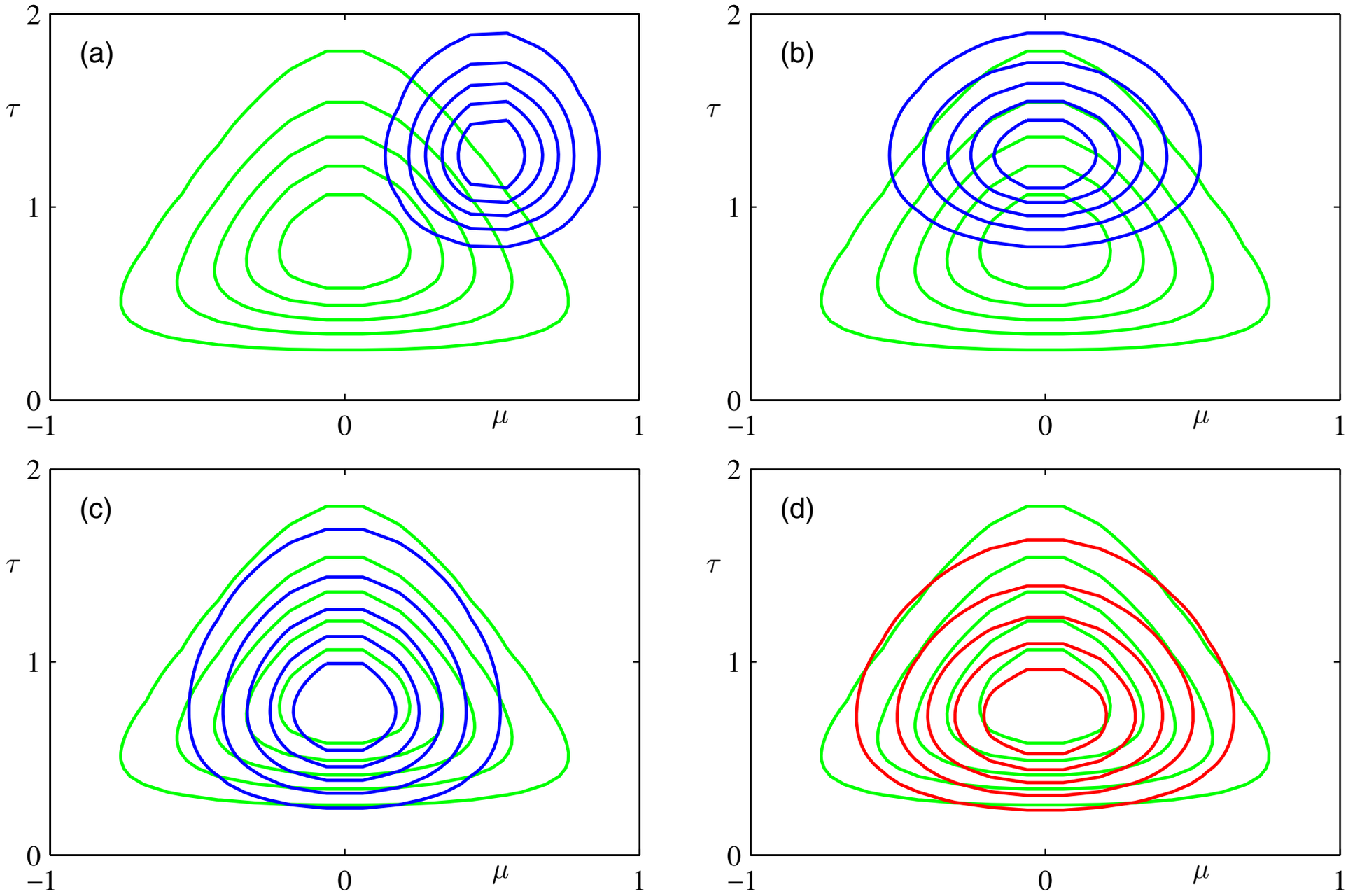
Penalty

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



Optimising the Variational Objective

Example of variational optimisation for a simple 2D density.



What optimisation schemes can we use to achieve this?

Variational EM

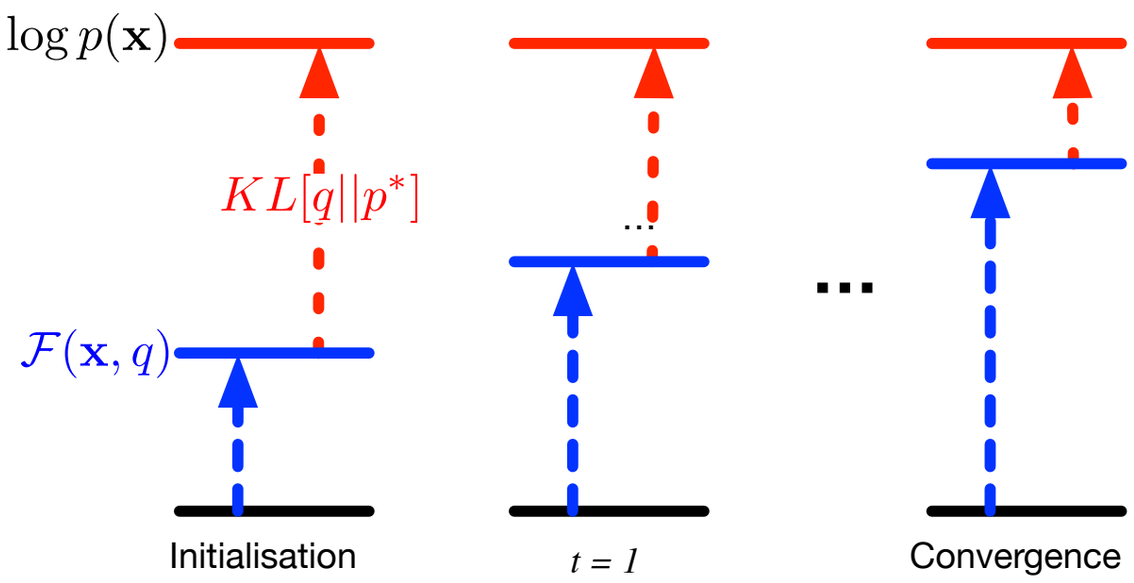
$$\mathcal{F}(\mathbf{x}, q) = \mathbb{E}_{q(\mathbf{z})}[\log p(\mathbf{x}|\mathbf{z})] - KL[q(\mathbf{z})\|p(\mathbf{z})]$$

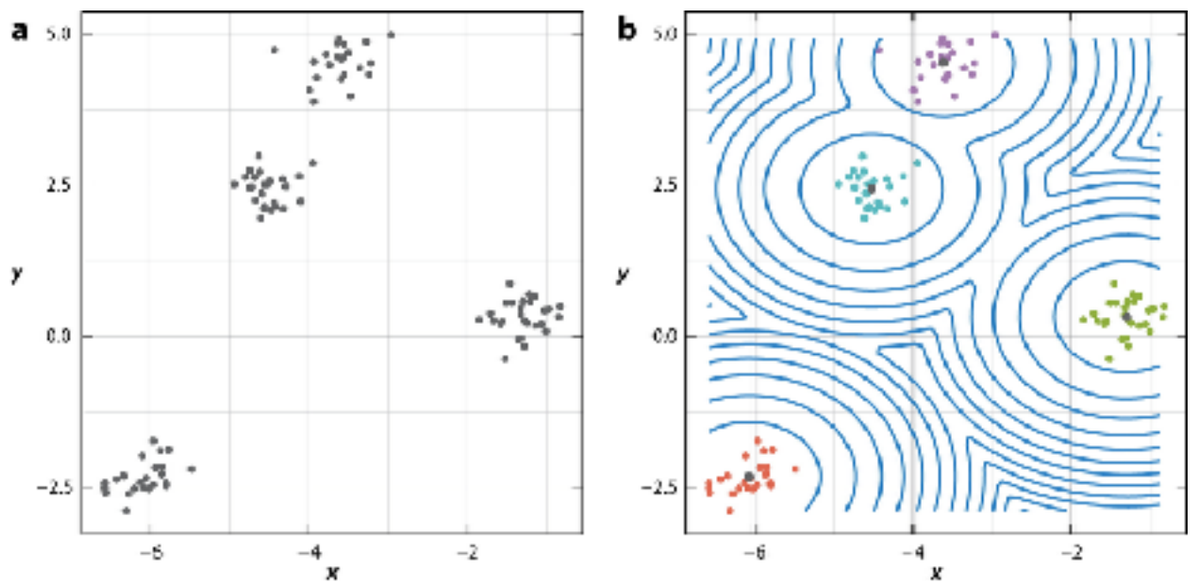
Alternating optimisation for the variational parameters and then model parameters (VEM).

Repeat:

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

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





Variational EM

Standard approach involves computation over the entire

Repeat:

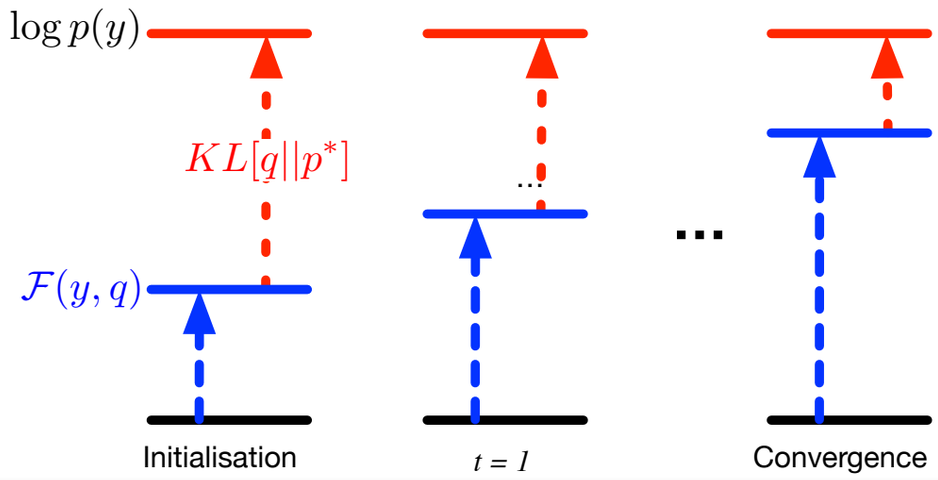
E-step *(Inference)*

For i = 1, ... N

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

M-step *(Parameter Learning)*

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



Stochastic Variational Inference

Instead use a **stochastic gradient based on a mini-batch** of data.

Many names: *online EM, stochastic approximation EM, stochastic variational inference.*

Repeat:

E-step *(Inference)*

For i = 1, ... N

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

M-step *(Parameter Learning)*

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

***N* is a mini-batch:** sampled with replacement from the full data set or received online.

Scalable - only need to operate on a small batch at a time. Can operate on large data sets.

Doubly Stochastic Variational Inference

VEM and SVI assume easy computation of the expected log-likelihood (and KL).

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

Instead compute all expectations by Monte Carlo approximation.

Doubly stochastic estimation : one source of stochasticity from the mini-batch, another from the Monte Carlo evaluation of the expectation.

Monte Carlo E-step: $\mathbf{z}_n^{(s)} \sim q_\phi(\mathbf{z}_n|\mathbf{x}_n)$

$$\phi_n \propto \nabla_\phi \frac{1}{S} \sum_s \left[\log p_\theta(\mathbf{x}_n|\mathbf{z}_n(\phi)^{(s)}) - \log \frac{q(\mathbf{z}_n(\phi)^{(s)})}{p(\mathbf{z}_n(\phi)^{(s)})}\right]$$

General idea only. Will make precise when we look at Monte Carlo estimators.

Amortised Inference

Repeat:

E-step (compute q)

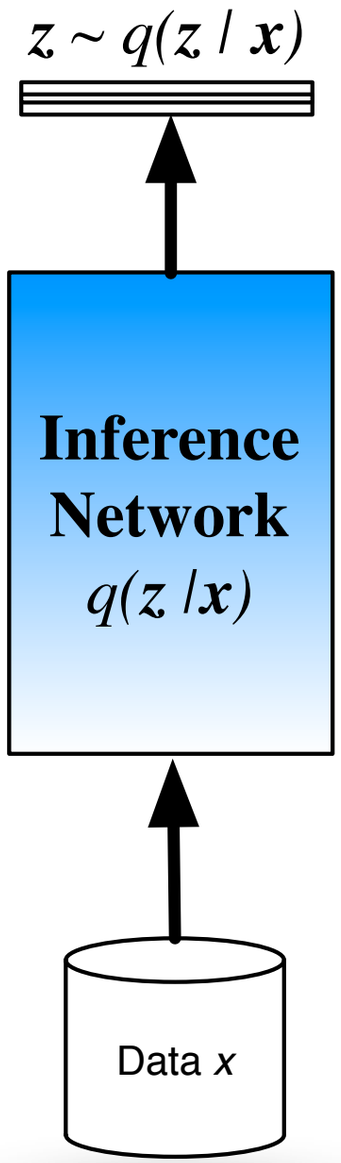
For i = 1, ... N

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

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**

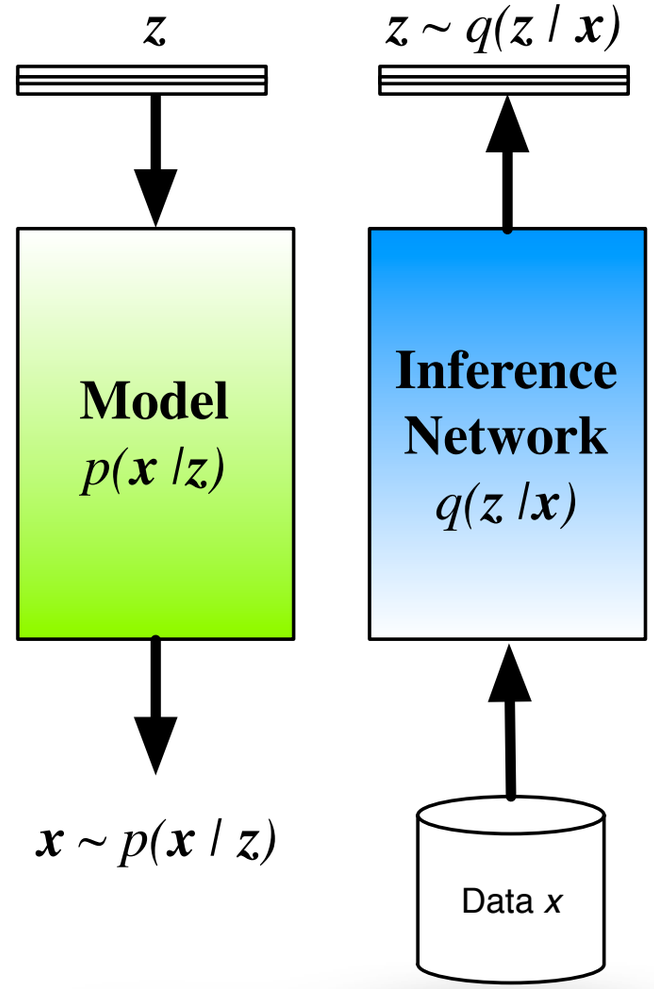
Amortised Variational Inference

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

Approx. Posterior

Reconstruction

Penalty



Stochastic encoder-decoder system to implement variational inference.

- **Model (Decoder):** likelihood $p(x|z)$.
- **Inference (Encoder):** variational distribution $q(z|x)$
- Transforms an auto-encoder into a generative model

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.

Computing the Expected Log-likelihood

An outstanding issue in all the optimisation methods is the computation of the expected log-likelihood (and KL term if unknown).

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})}[\log p_\theta(\mathbf{x}_n|\mathbf{z}_n)]$$

- We don't know this expectation in general.
- The parameters of the distribution with respect to which the expectation is taken.

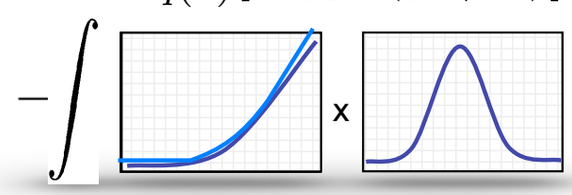
Two general approaches

- **Deterministic methods:** use additional bounds to simplify computation - local variational methods. Referred to as local variational methods.
- **Stochastic methods:** Compute the expectation by Monte Carlo and exploit properties of the distributions.

Local Variational Methods

Replace the likelihood with a simpler form — a lower bound that makes the expectation easy to compute.

$$\mathbb{E}_{q(z)}[\log p_\theta(y_n|z_n)]$$

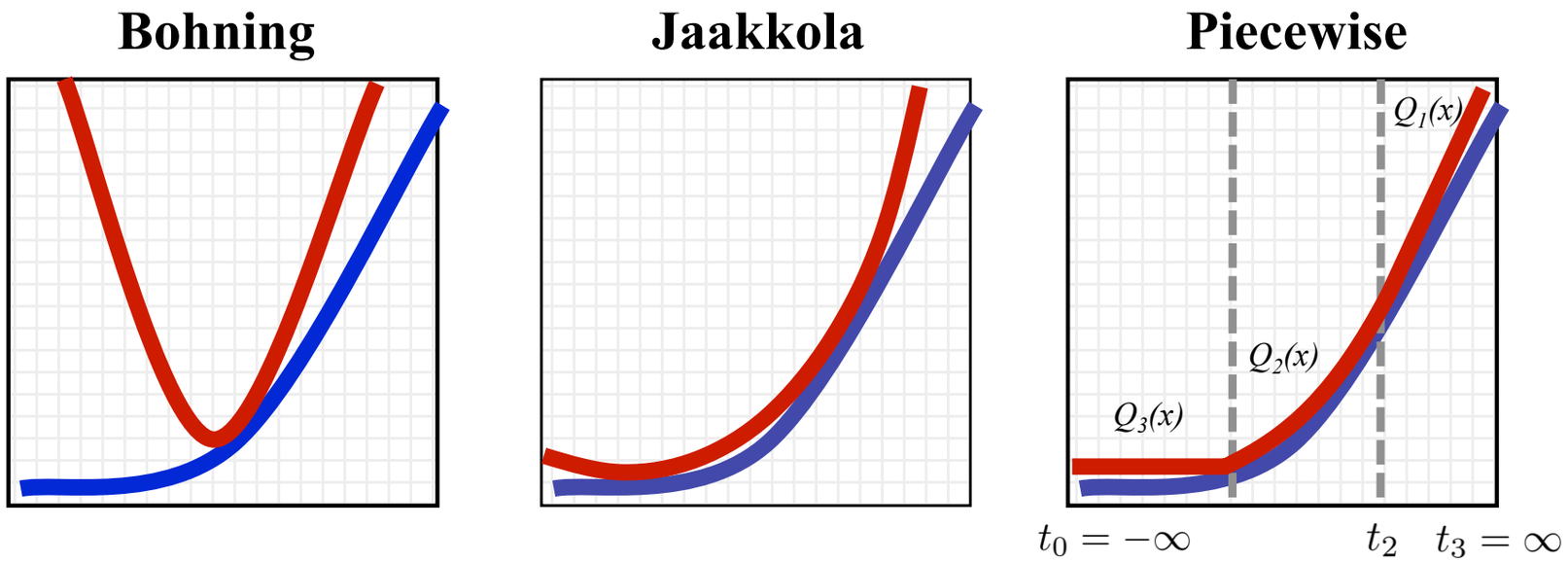


Original problem $$p(y=1|z) = \frac{1}{1+\exp(-z)} = \sigma(z)$$

Local Bound $$\sigma(z) \geq \sigma(\xi)\exp\left(\frac{z-\xi}{2} - \lambda(\xi)(z^2-\xi^2)\right)$$

Additional variational parameters ξ

Bound with only linear or quadratic terms: expectations, especially against a Gaussian, are easy to compute.



Stochastic Optimisation

Common gradient problem

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

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

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

Typical problem areas:

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

Score-function Estimators

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

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

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

Other names:
Likelihood ratio method
REINFORCE and policy gradients
Automated inference
Black-box inference

When to use:
Function is not differentiable.
Distribution q is easy to sample from.
Density q is known and differentiable.

Score-function Estimators

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

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

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

Score function and its properties

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

Deriving the estimator

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

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

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

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

Pathwise-derivative Estimators

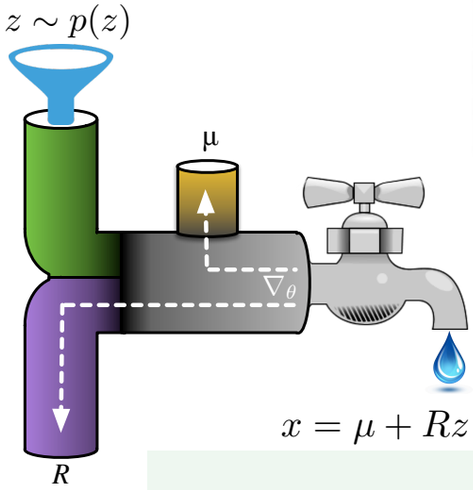
Pathwise Estimator

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

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

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

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



Other names:

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.

Pathwise-derivative Estimators

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

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

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

Samplers, one-liners and change-of-variables

$$z \sim q_\phi(\mathbf{z}) \qquad \mathbf{z} = g(\epsilon, \phi) \quad \epsilon \sim p(\epsilon) \qquad p(z) = \left|\frac{d\epsilon}{dz}\right| p(\epsilon) \implies |p(z)dz| = |p(\epsilon)d\epsilon|$$

(Non-rigorous) Deriving the Estimator

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

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

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

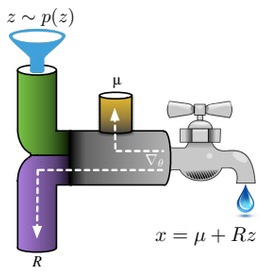
Stochastic Gradient Estimators

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

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.

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

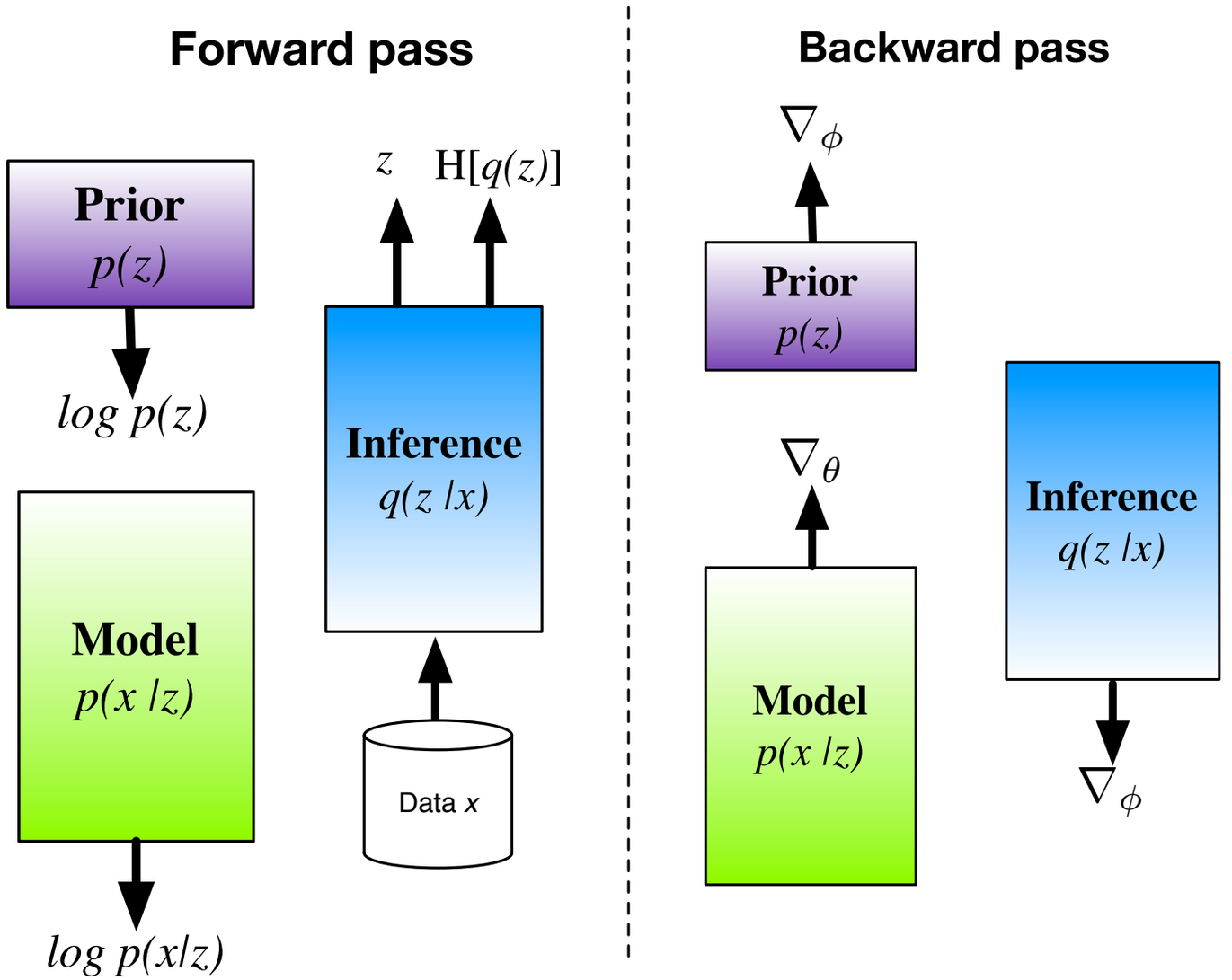
Doubly stochastic estimators

Implementing a Variational Algorithm

Variational inference turns integration into optimisation: **Automated Tools:**

- **Differentiation:** Theano, Torch7, TensorFlow, Stan.
- **Message passing:** infer.NET

- Stochastic gradient descent and other preconditioned optimisation.
- Same code can run on both GPUs or on distributed clusters.
- Probabilistic models are modular, can easily be combined.



Ideally want probabilistic programming using variational inference.

Variational Inference Theory

- **Tightness of the bound:**
  - The bound is exact is if q is the true posterior.
  - For certain classes of q-distributions, we can show that the class of distributions is rich enough to include the true posterior distributions.
- **Convexity and duality**
  - For Latent Gaussian models, and many others, we can show convexity of the variational objective. This allows us to exploit other optimisation approaches such as dual decomposition.
- **Bound correction**
  - We can obtain a tighter bound in a number of settings using perturbation analysis.

Variational Inference Theory

- **Convergence**
  - Based on VEM, we can show convergence to local minima.
  - We can also show theoretically for certain models that we have local convergence to the optimum in asymptotic settings.
- **Consistency**
  - We can show consistency of the mean of maximum likelihood parameter estimates, for some types of latent variable models using properties of the functional derivative. In other cases, we can show that we get inconsistent estimators.
- **Asymptotic normality**
  - We can use the theory for asymptotic normality of Laplace approximations to show asymptotic normality for certain classes of models using variational inference.

Other Variational Problems

- Belief propagation
- Expectation propagation
- Mutual Information maximisation
- Rate distortion theory
- Information bottleneck
- Policy search methods

Consolidation Questions

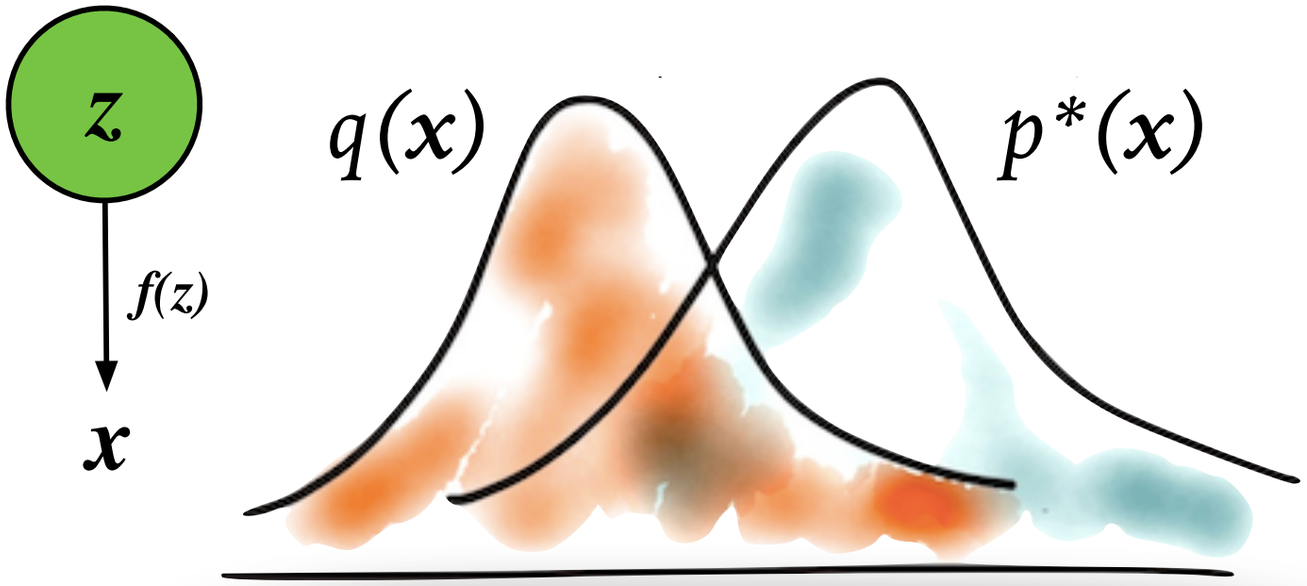
- What other examples of amortised inference can you think of. Make a list of models and describe the what parts of the algorithm implement amortised inference. What are the disadvantages of amortised inference.
- What other ways do we have of deriving the stochastic gradient estimators we considered.
- There are two other types of stochastic gradient estimators: weak derivative estimators (also known as finite difference estimators), and harmonic gradient estimators. Look these up to know of their existence.
- Finite difference estimators are an important alternative to what we discussed. Build a basic understanding of their limitations and applications. Look up SPSA.
- Why can we interchange the order of integration and differentiation in deriving the score-function estimator (Leibniz integral rules).
- Write the gradient estimator of the variational free energy for a Latent Gaussian model w.r.t. both the model and the variational parameters.
- Connect how you used the score-function estimator to the way you implemented learning in RL using REINFORCE, policy gradient methods and actor-critic methods. Is there an equivalence between concepts used in RL and here?

Learning in Implicit Probabilistic Models

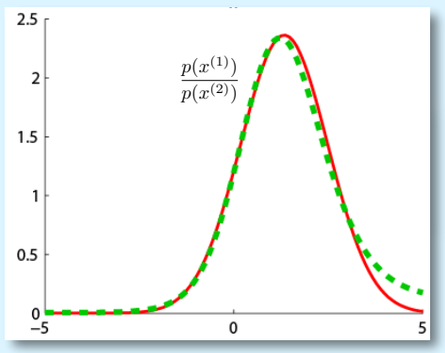
Estimation by Comparison

For some models, we only have access to an unnormalised probability or partial knowledge of the distribution.

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



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



Learning principle: Two-sample tests

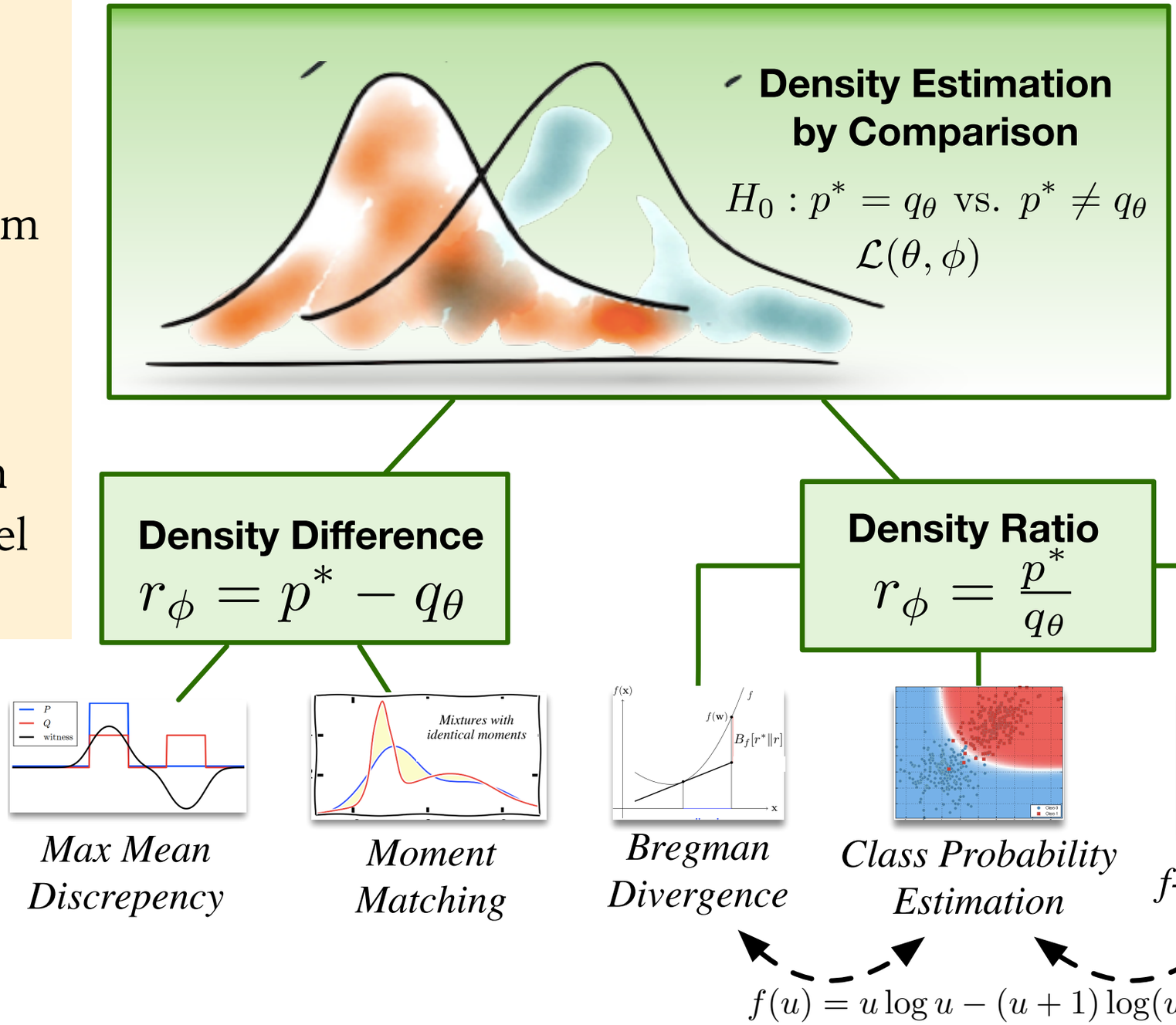
$$\frac{p^*(\mathbf{x})}{q(\mathbf{x})} = 1 \qquad p^*(\mathbf{x}) = q(\mathbf{x})$$

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

Estimation by Comparison

Two steps

1. Use a hypothesis ***test or comparison*** to obtain some model to tells how data from our model differs from observed data.
2. ***Adjust model*** to better match the data distribution using the comparison model from step 1.



Density Ratio Estimation

Combine data

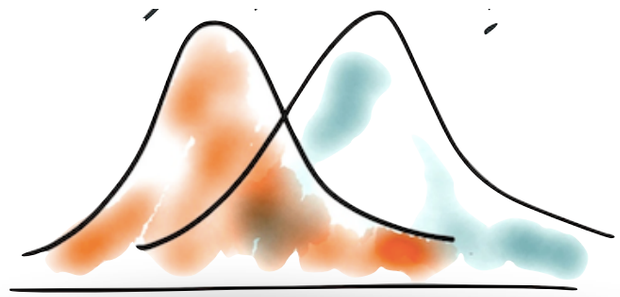
$$\{\mathbf{x}_1, \ldots, \mathbf{x}_N\} = \{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_{\hat{n}}, \tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_{\tilde{n}}\}$$

Assign labels

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

Equivalence

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



Density Ratio

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

Bayes' Rule

$$p(\mathbf{x}|y) = \frac{p(y|\mathbf{x})p(\mathbf{x})}{p(y)}$$

Conditional

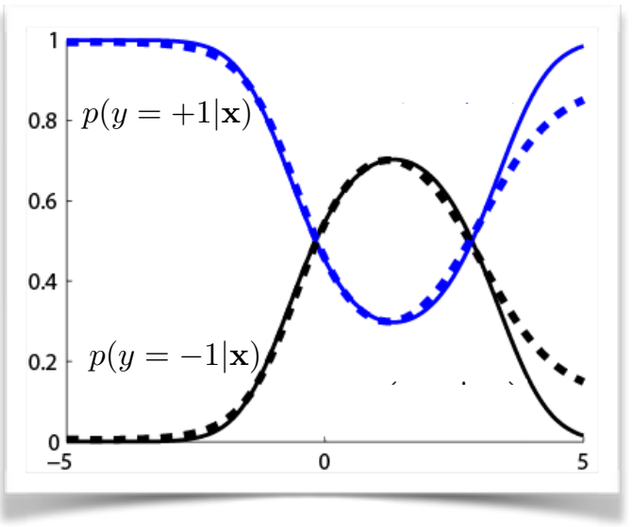
$$\frac{p^*(\mathbf{x})}{q(\mathbf{x})} = \frac{p(\mathbf{x}|y=1)}{p(\mathbf{x}|y=-1)}$$

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

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

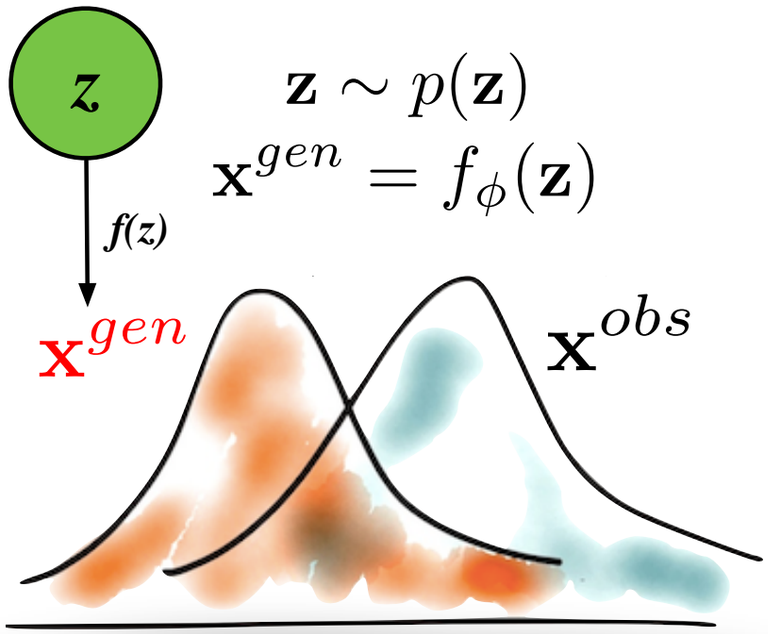


Computing a density ratio is equivalent to class probability estimation.

Adversarial Learning

Scoring Function $p(y = +1|\mathbf{x}) = D_\theta(\mathbf{x}) \qquad p(y = -1|\mathbf{x}) = 1 - D_\theta(\mathbf{x})$

Bernoulli Ratio Loss $\mathcal{F}(\mathbf{x}, \theta, \phi) = \mathbb{E}_{p^*(x)}[\log D_\theta(\mathbf{x})] + \mathbb{E}_{q_\phi(x)}[\log(1 - D_\theta(\mathbf{x})]$



$$\mathbf{z} \sim p(\mathbf{z})$$
$$\mathbf{x}^{gen} = f_\phi(\mathbf{z})$$

Instances of testing and inference:

- Unsupervised-as-supervised learning
- Classifier ABC
- Noise-contrastive estimation
- Adversarial learning and GANs

Alternating optimisation

$$\min_\phi \max_\theta \mathcal{F}(\mathbf{x}, \theta, \phi)$$

Generative Adversarial Networks

Comparison loss $\theta \propto \nabla_\theta \mathbb{E}_{p^*(x)}[\log D_\theta(\mathbf{x})] + \nabla_\theta \mathbb{E}_{q_\phi(x)}[\log(1 - D_\theta(\mathbf{x})]$

Generative loss $\phi \propto -\nabla_\phi \mathbb{E}_{q(z)}[\log(1 - D_\theta(f_\phi(\mathbf{z}))]$

Consolidation Questions

- Find some examples of non-likelihood and non-maximum likelihood estimation.
- The are many ways to do hypothesis tests and comparisons. One way is by the method of maximum mean discrepancy and empirical distance metrics (Dudley and Wassertein metrics). Find references on how these can be used.
- One approach for Bayesian likelihood-free inference is called approximate Bayesian computation (ABC) that is widely used in computational biology. Think about ABC in the framework of comparison and estimation.
- Other approaches worth looking at include mean-shift estimation, method of moments, classifier ABC.

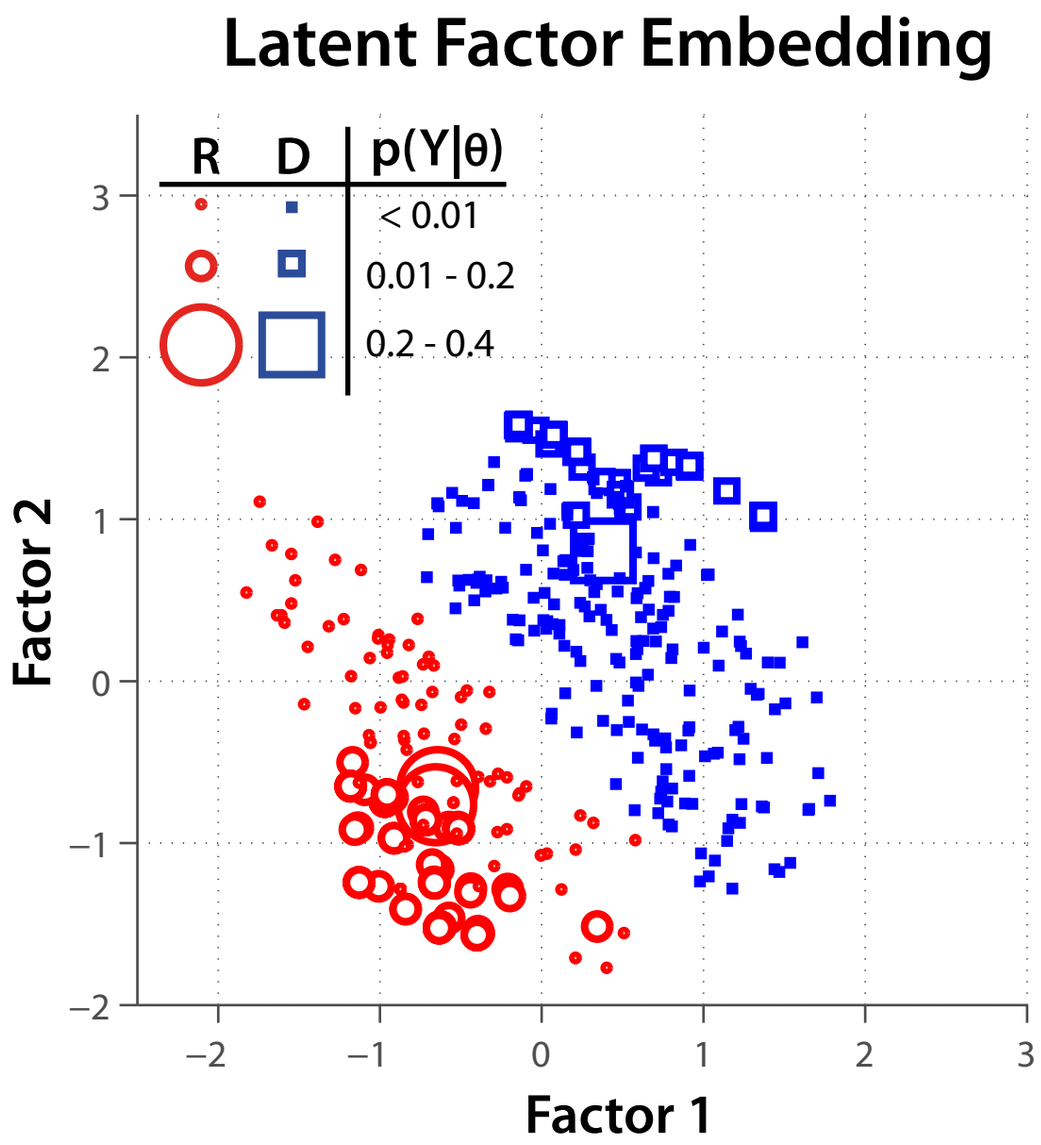
Part VI

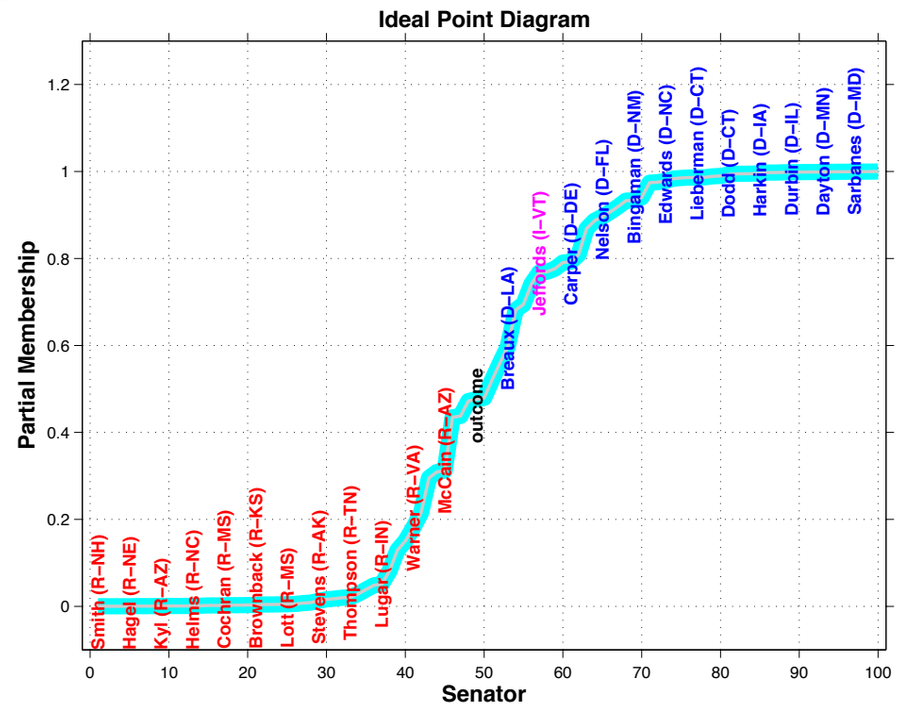
Applications and Extensions of Generative Models

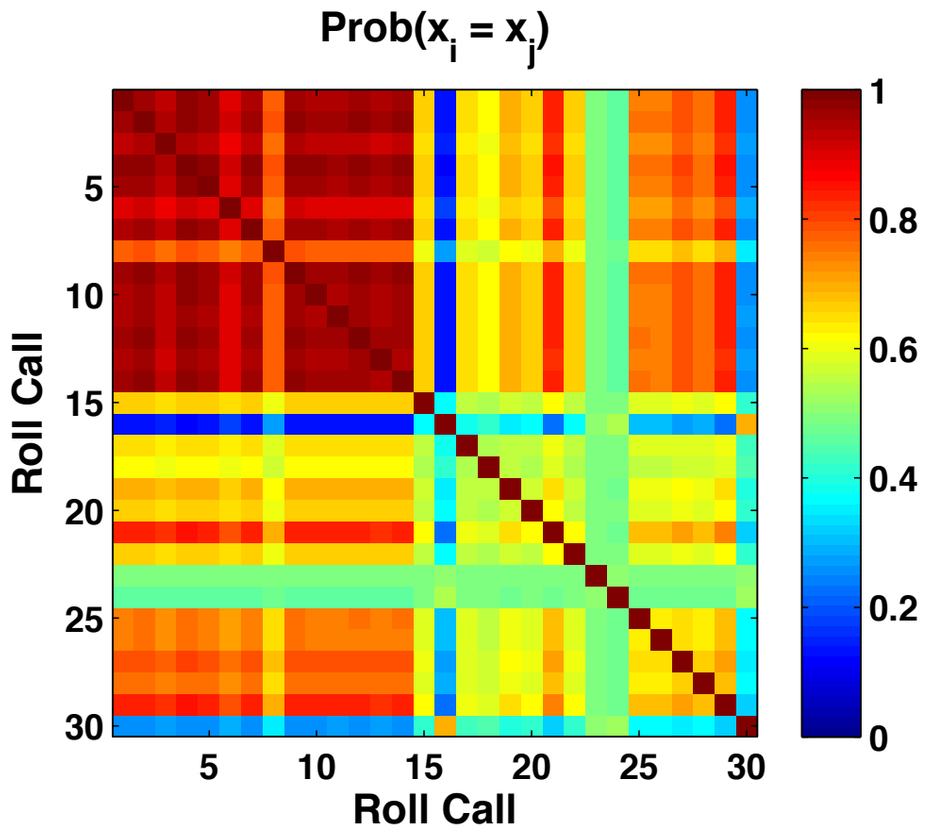
Data Visualisation

Factor Analysis

Binary data set of votes in the US senate.

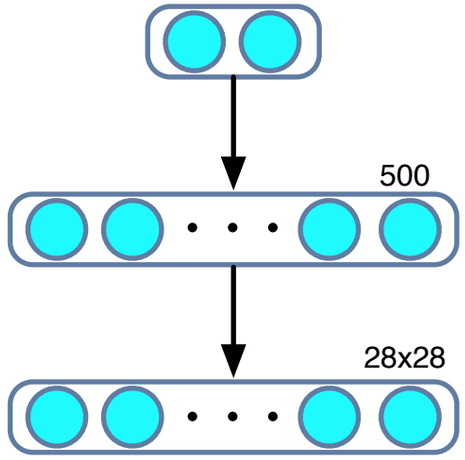




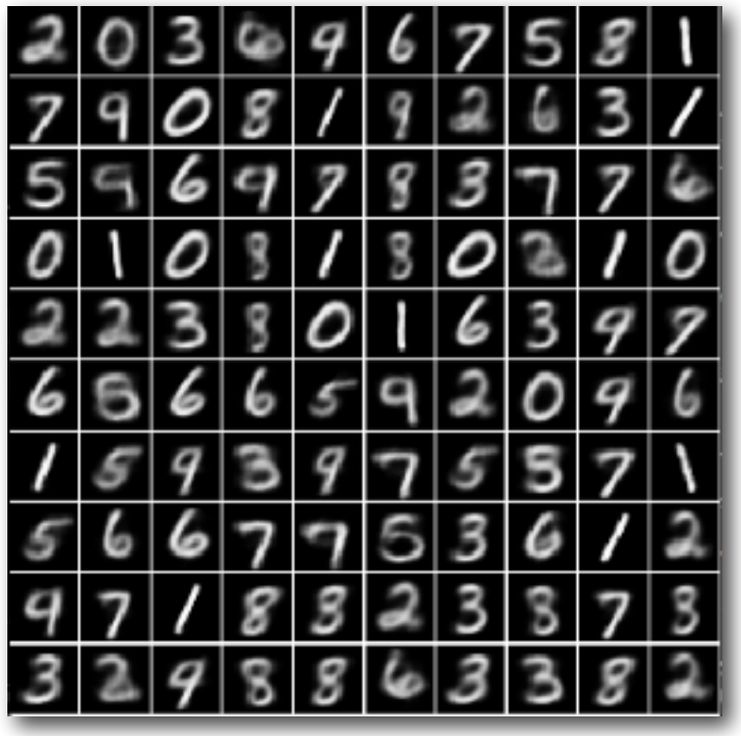


Data Visualisation

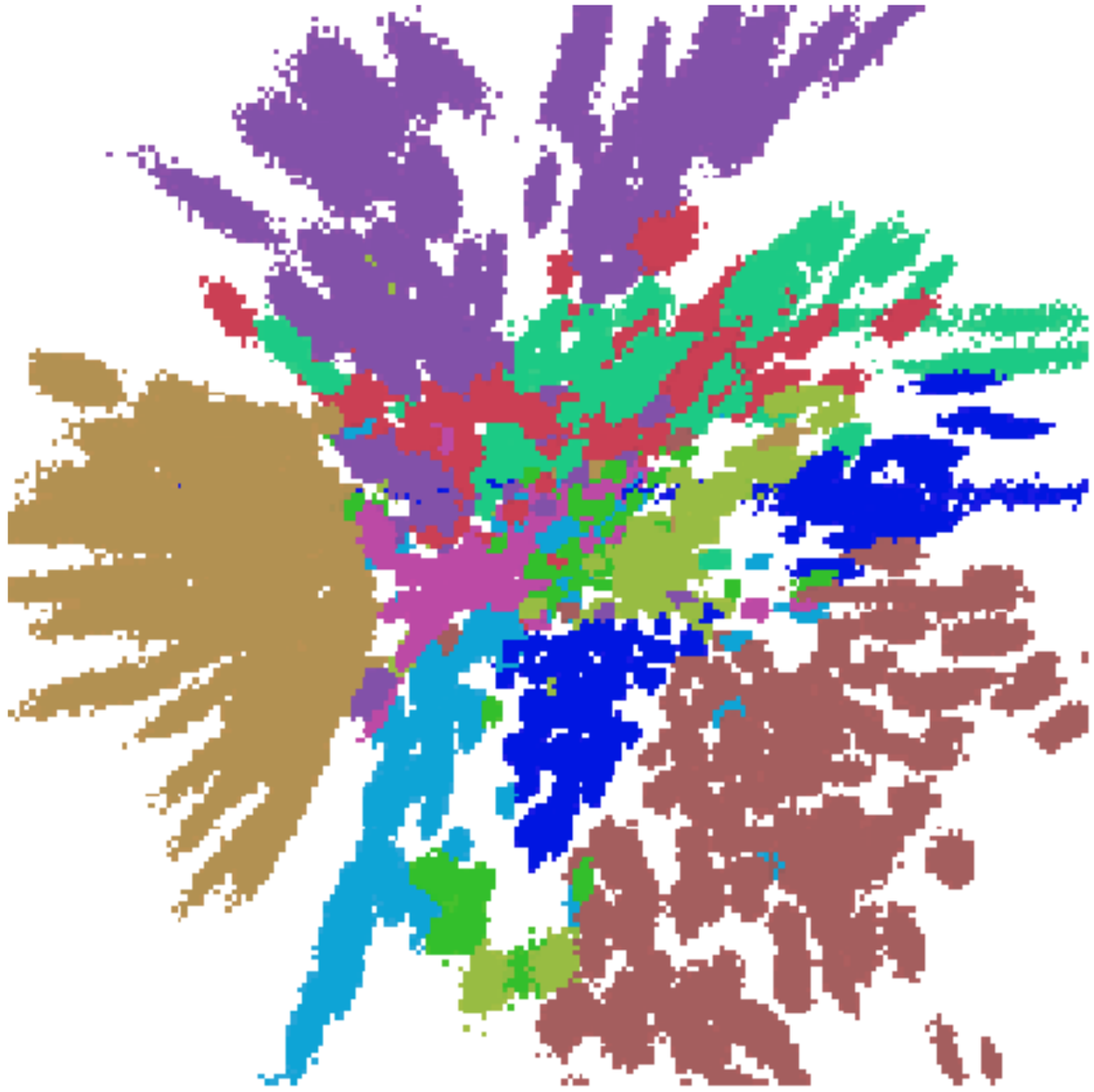
MNIST Handwritten digits



DLGM



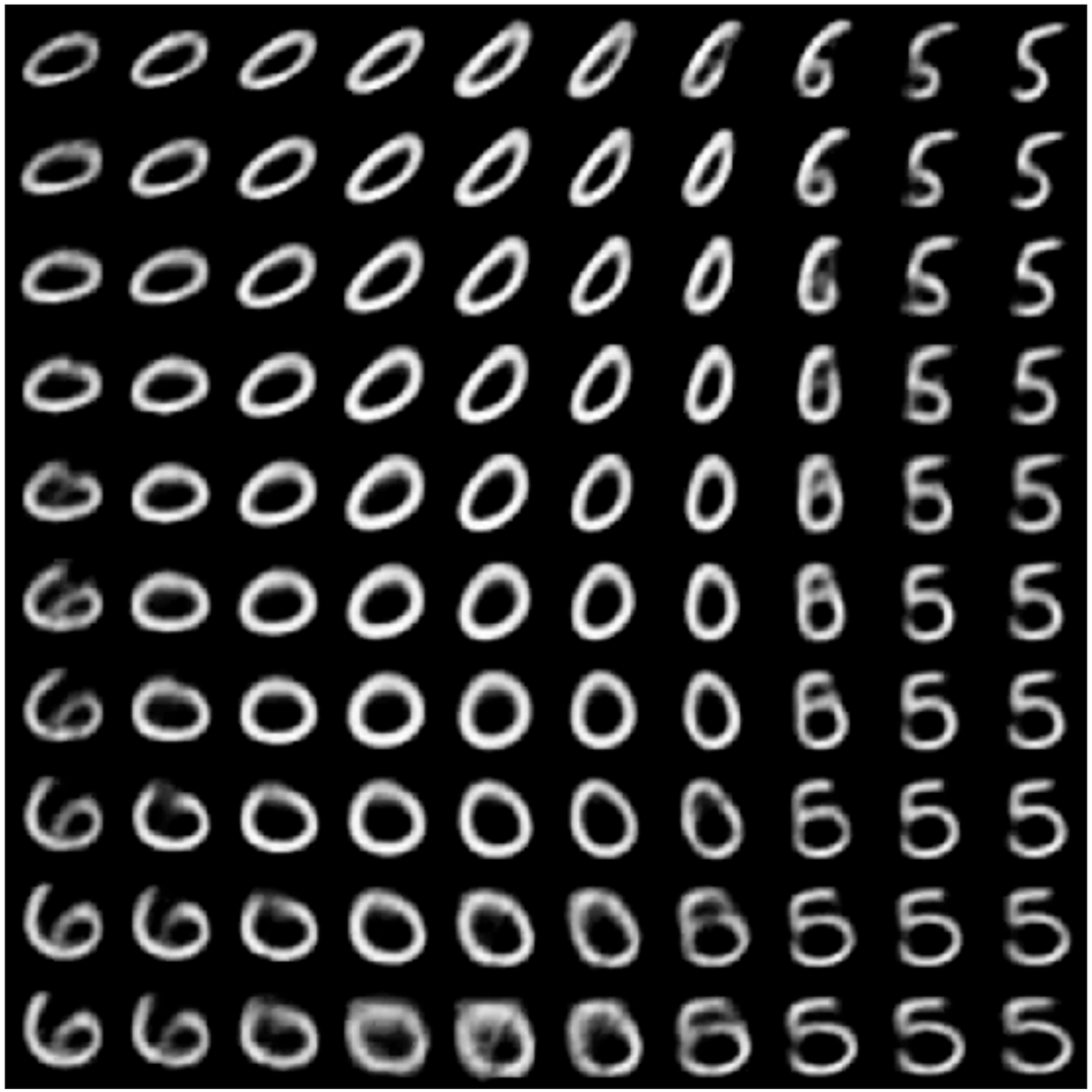
Samples from 2D latent model



Labels in 2D latent space

Visualizing MNIST in 3D

Visualising MNIST in 2D



DLGM

Data Simulation

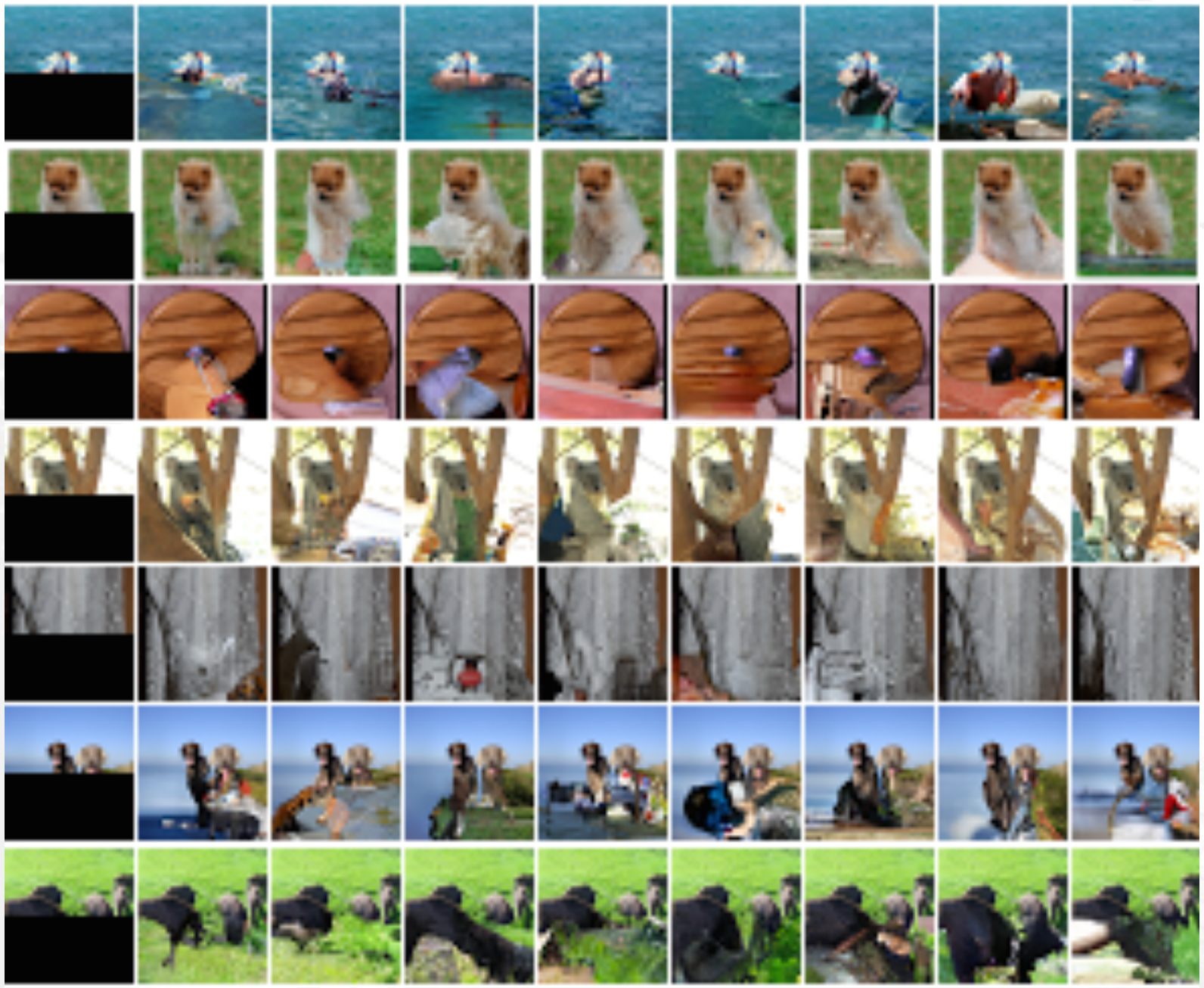
DLGM

Data

Samples

Data Imputation

Pixel CNN

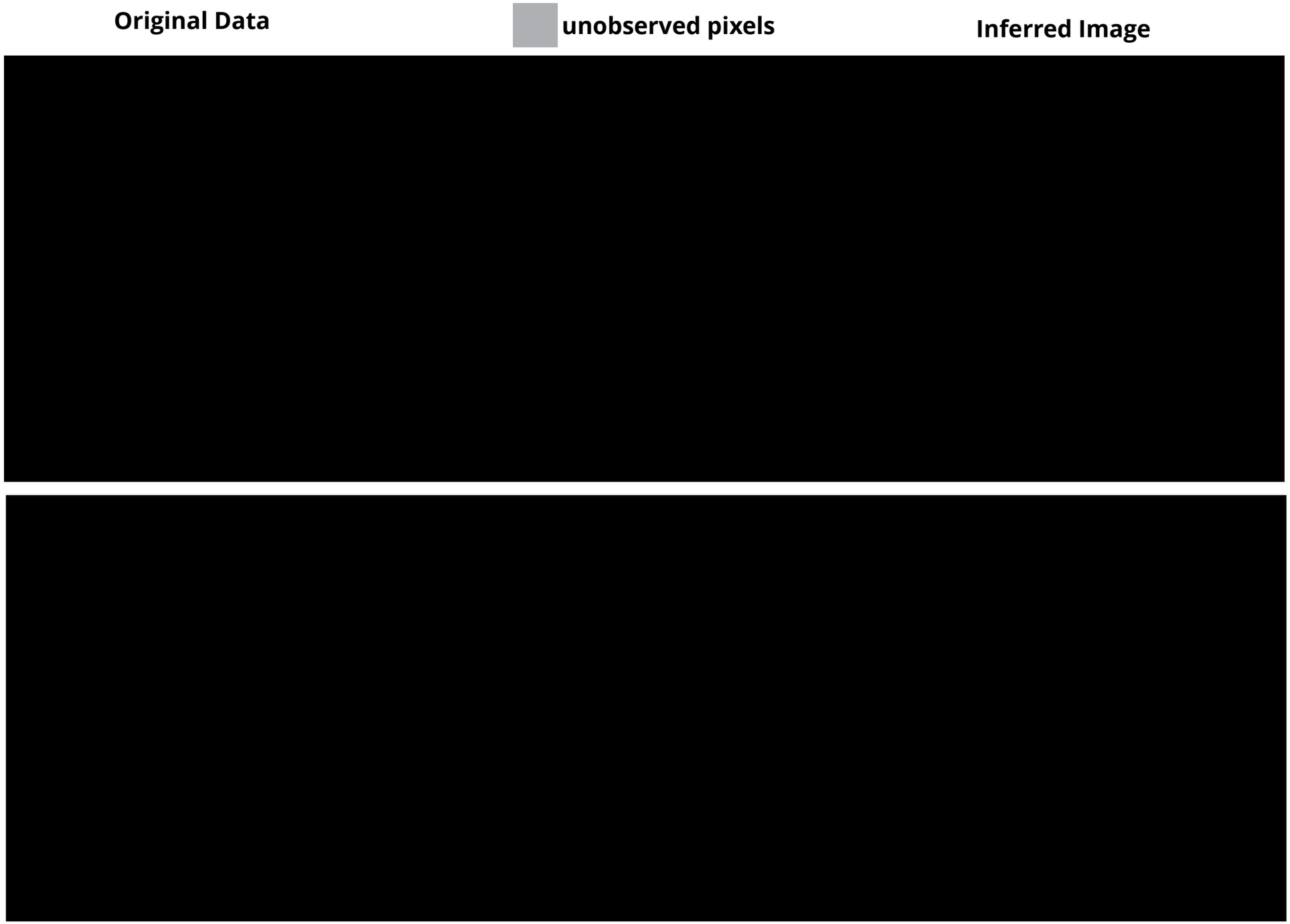


Missing Data Imputation

DLGM

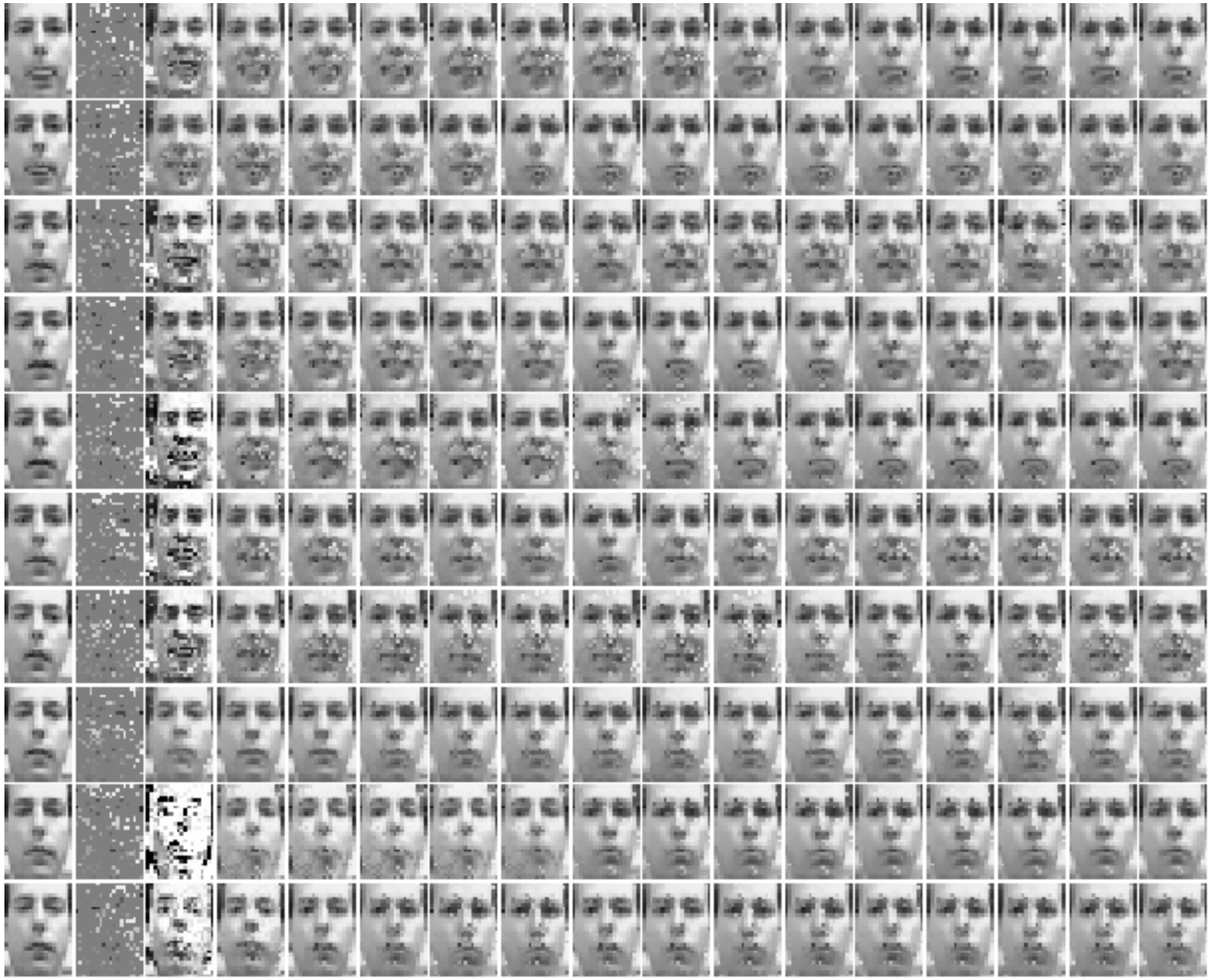
10%
observed

50%
observed



Missing Data Imputation

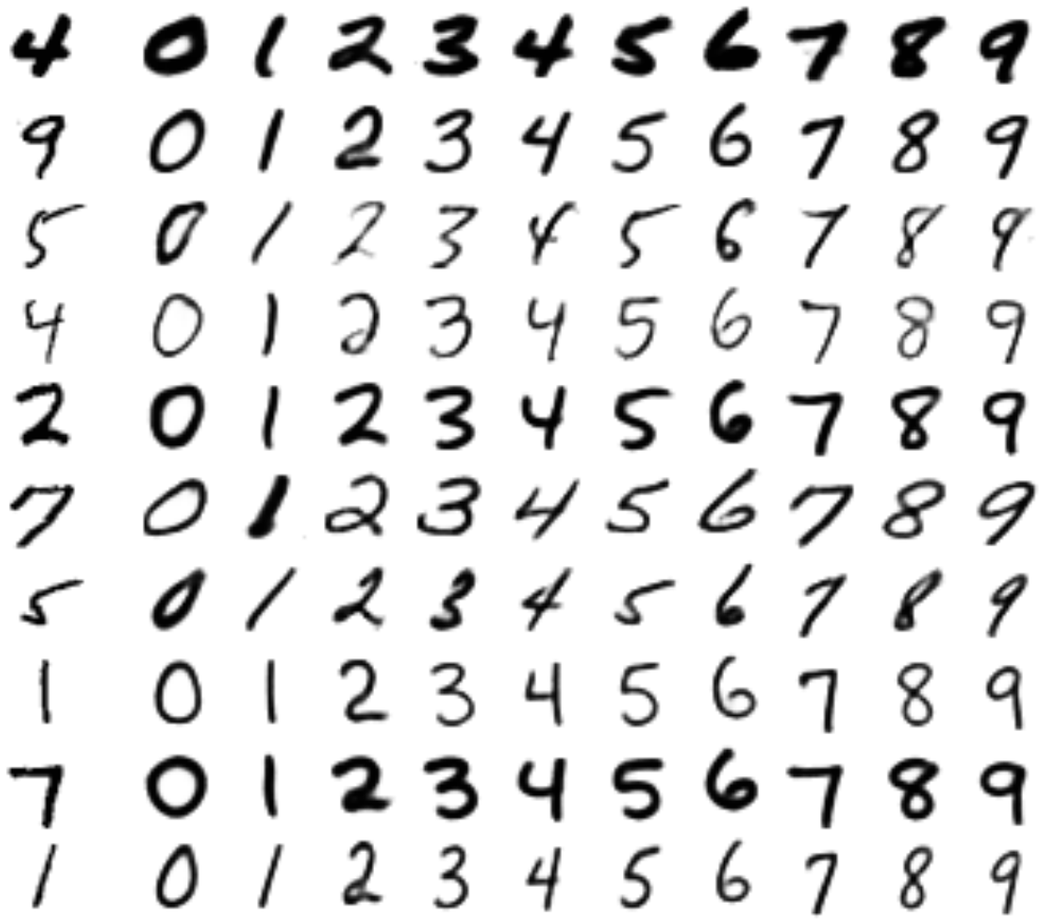
Frey Faces dataset. Completion: **80% missing pixels**

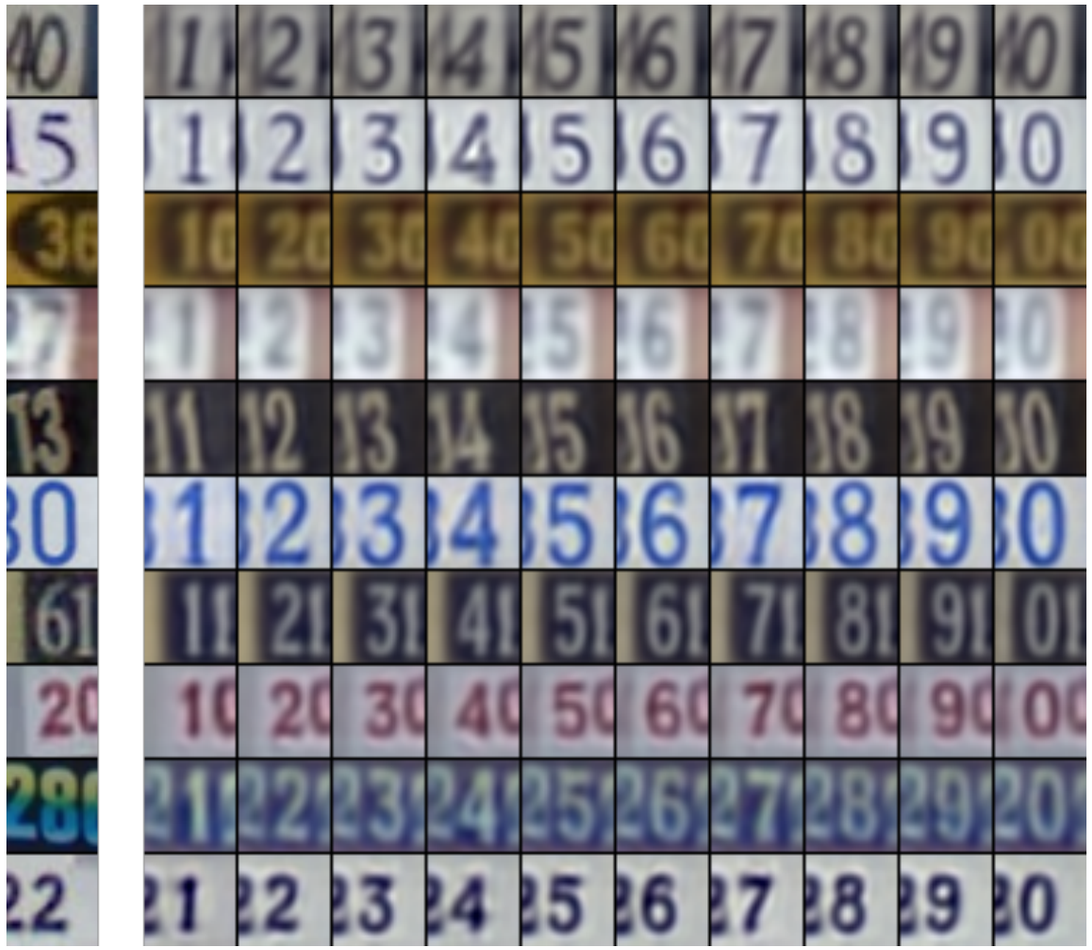


DLGM

Analogical Reasoning

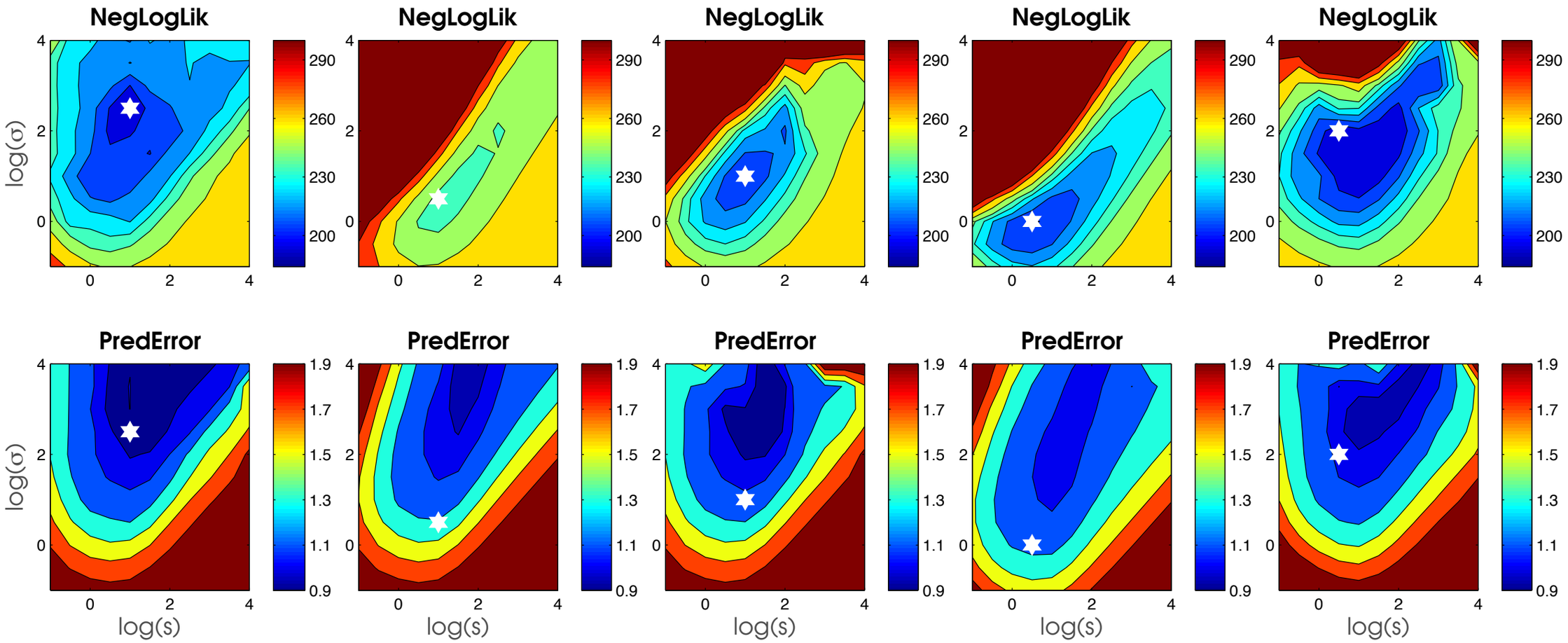
Semi-supervised DLGM





Model Selection

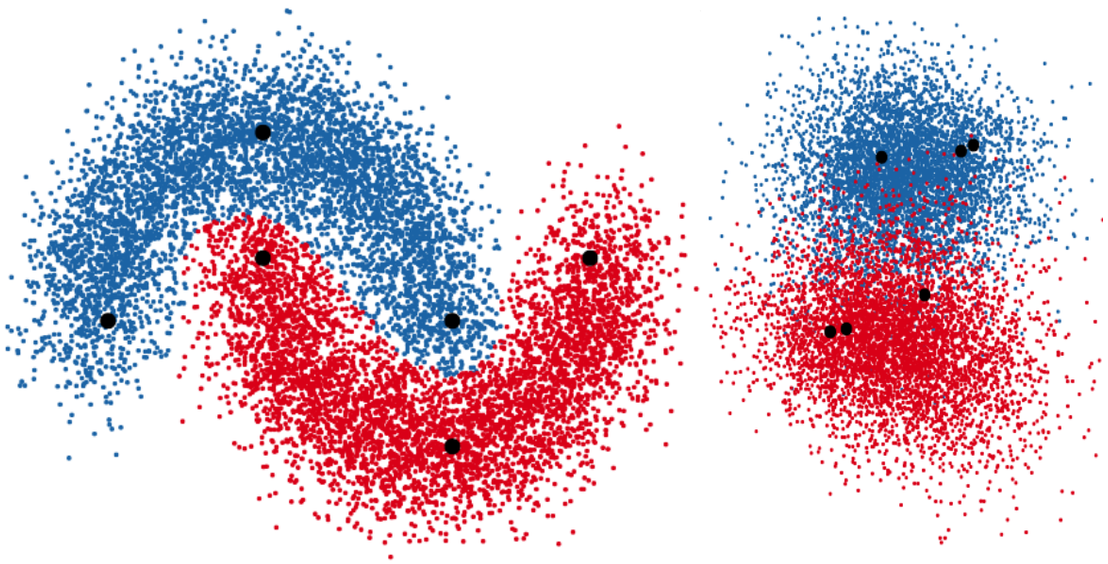
GP Regression

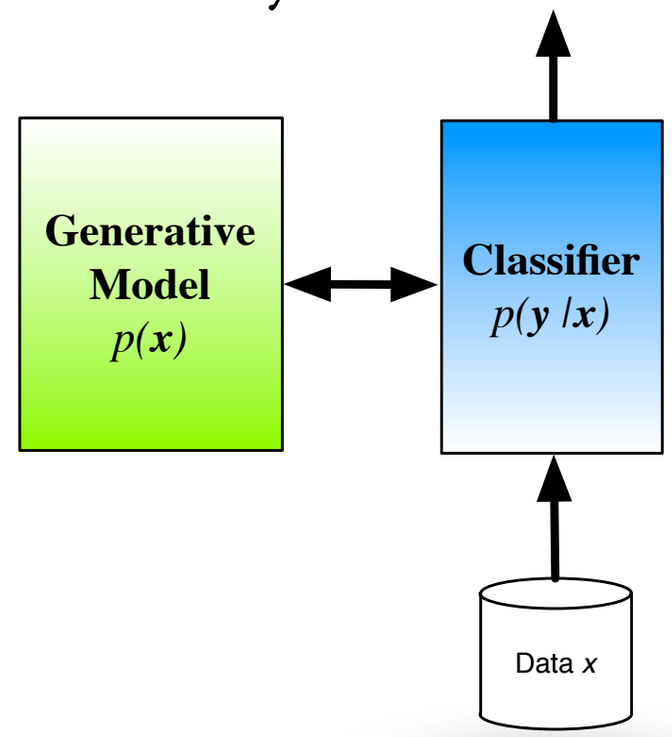


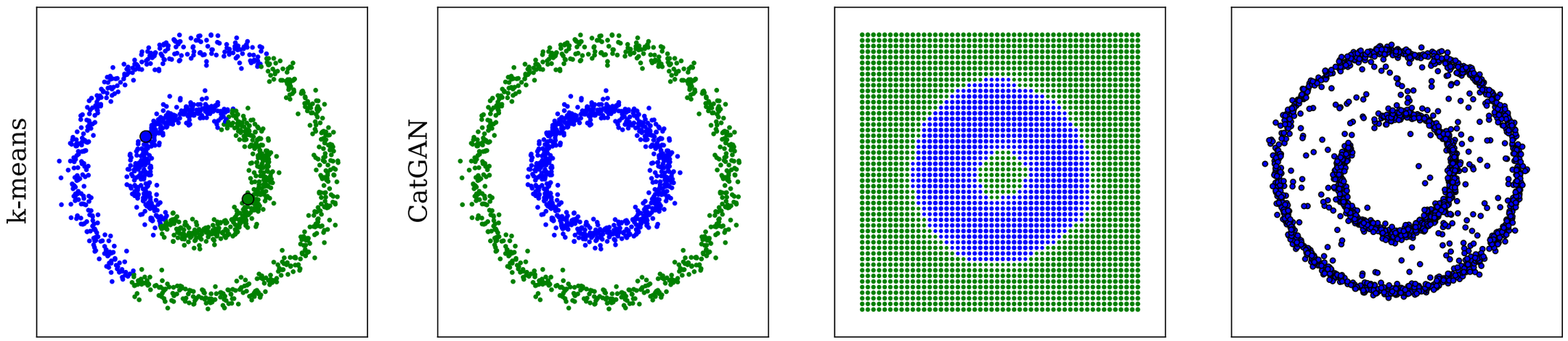
(a) Logit-HMC (b) Logit-Boh (c) Logit-Blei (d) Probit-VB (e) Stick-PW

Semi-supervised Classification

- How generative models can be used to improve the ability of discriminative models
- Allow for data-efficiency.

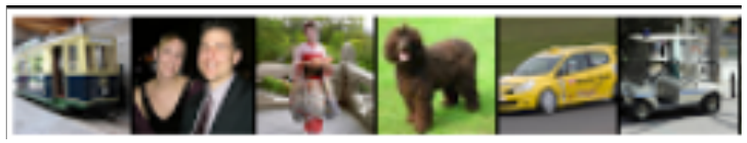






Communication and Compression

Original Image



0.1 bits/pixel

0.4 bits/pixel

jpeg

jpeg 2000

generative

mean

jpeg

jpeg 2000

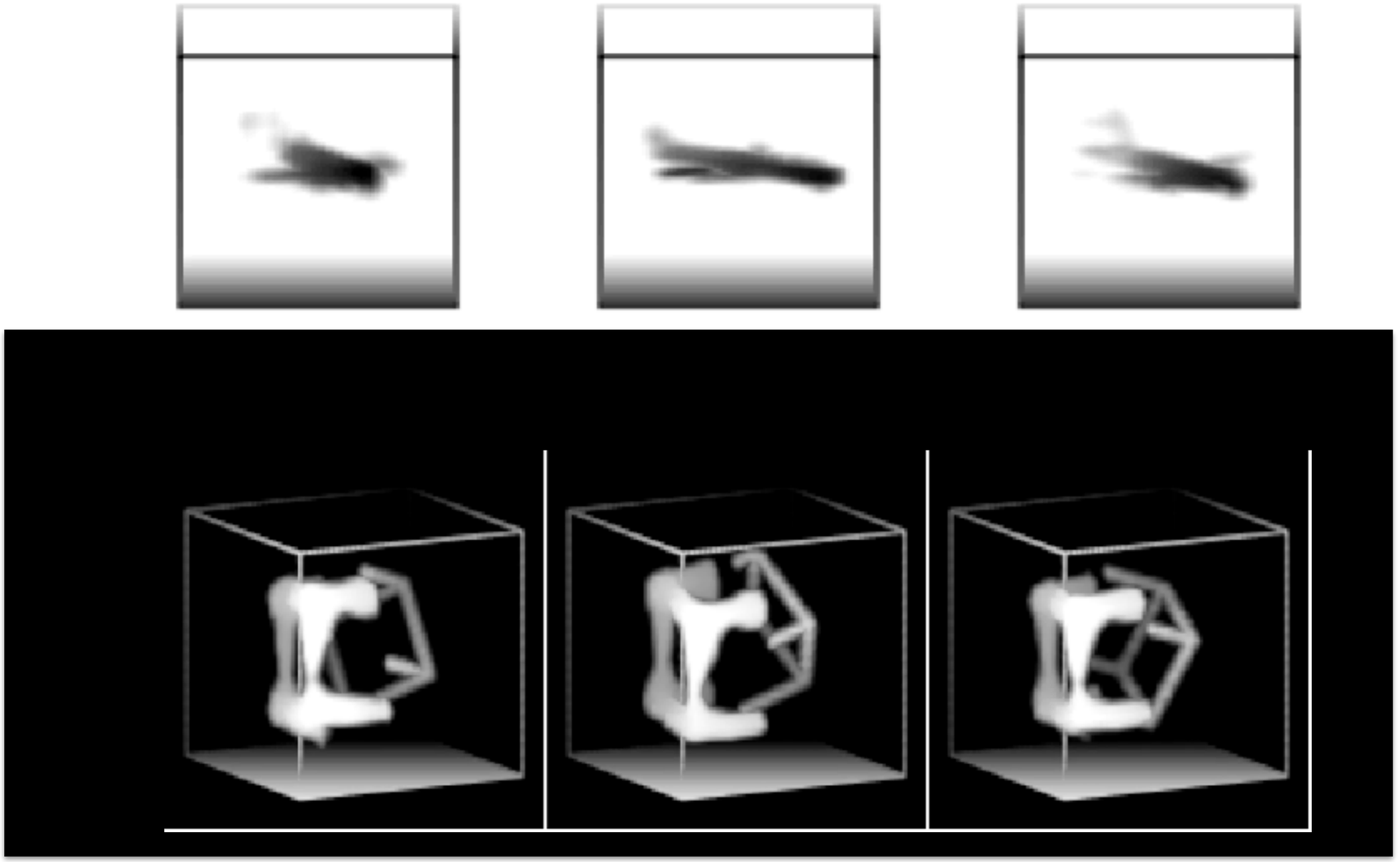
generative

mean

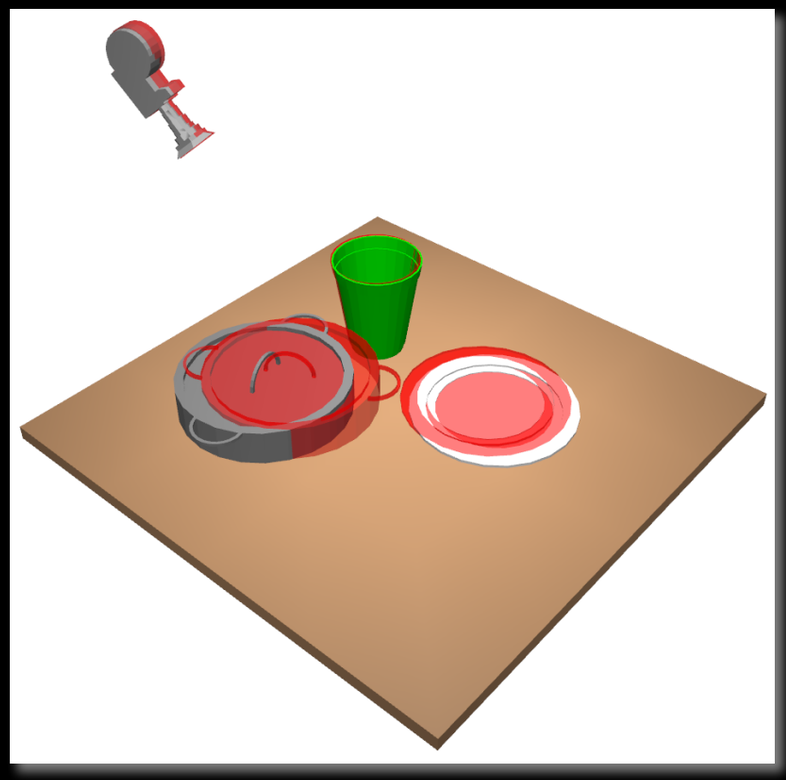
0.2 bits/pixel

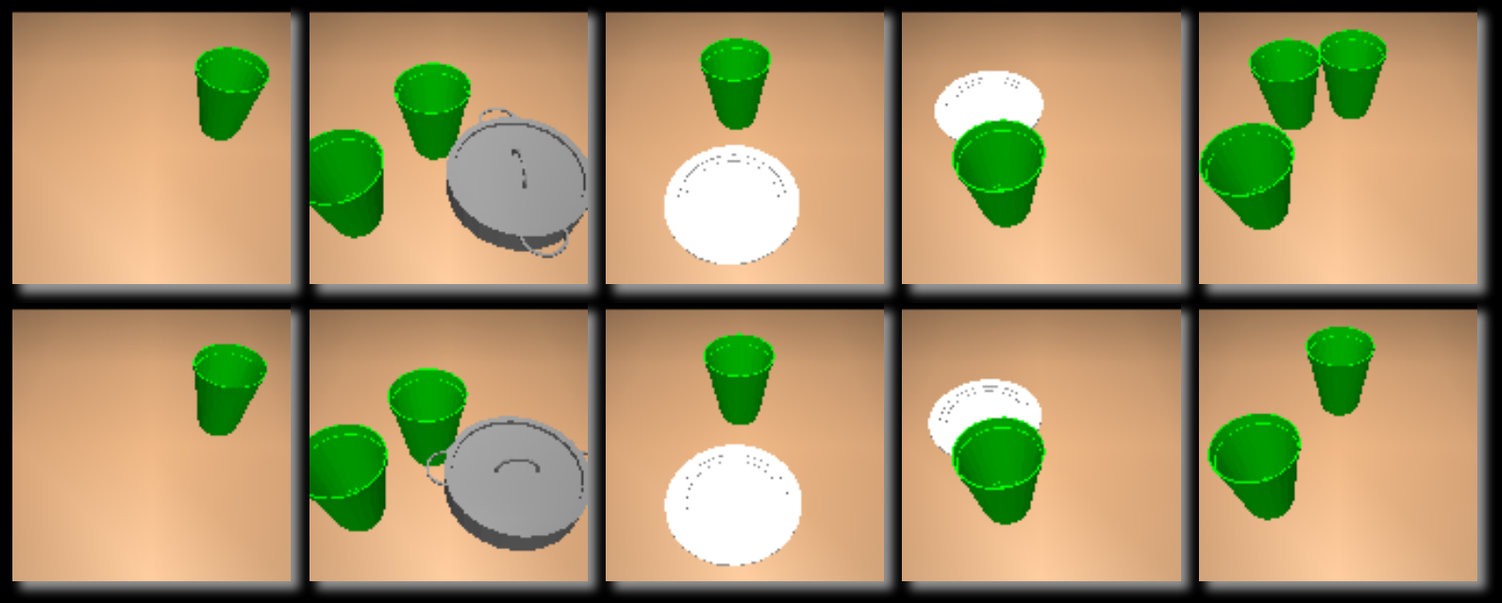
0.8 bits/pixel

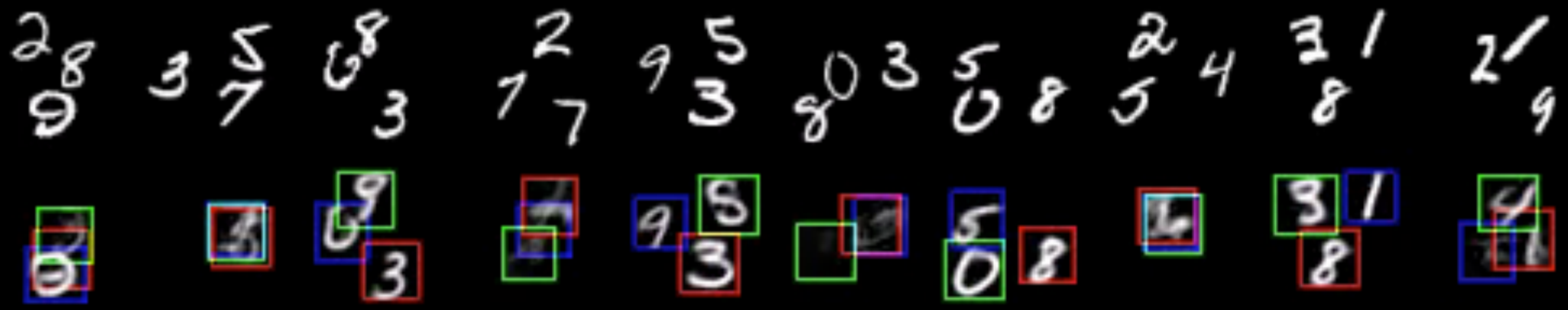
3D Scene Generation



Rapid Scene Understanding





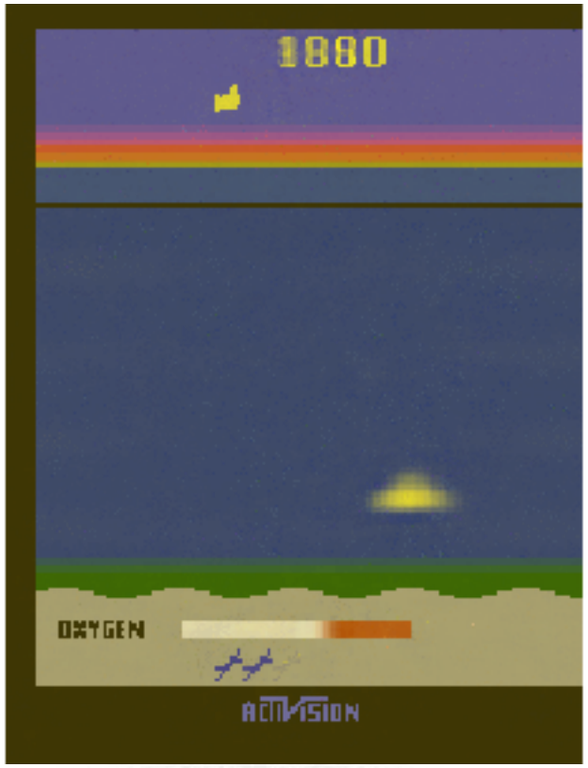


One-shot Generalisation

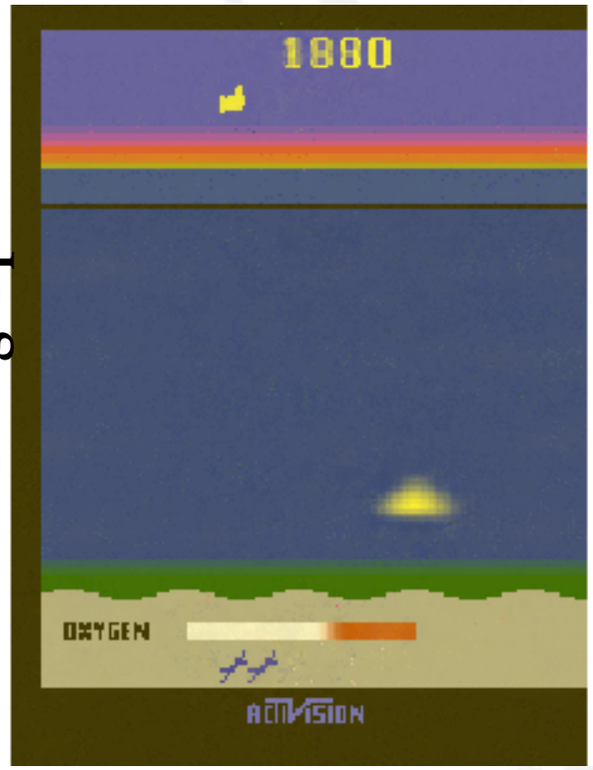
Visual Concept Learning

Original

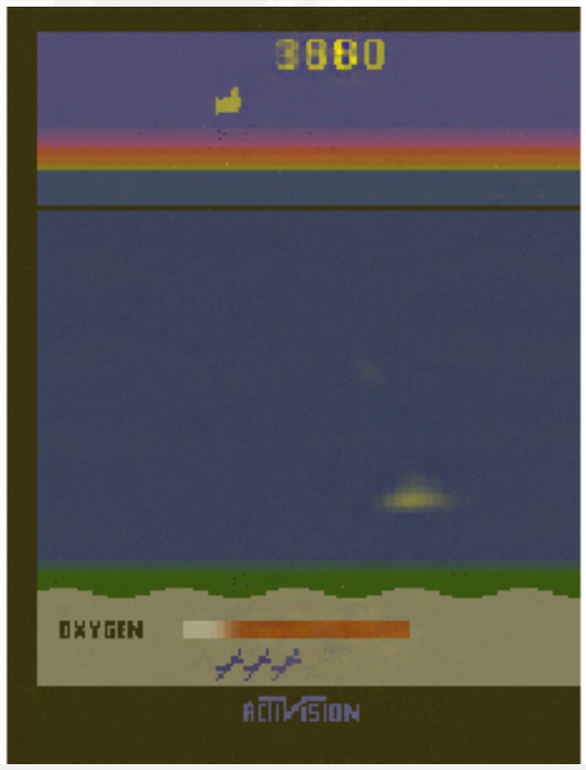
Score



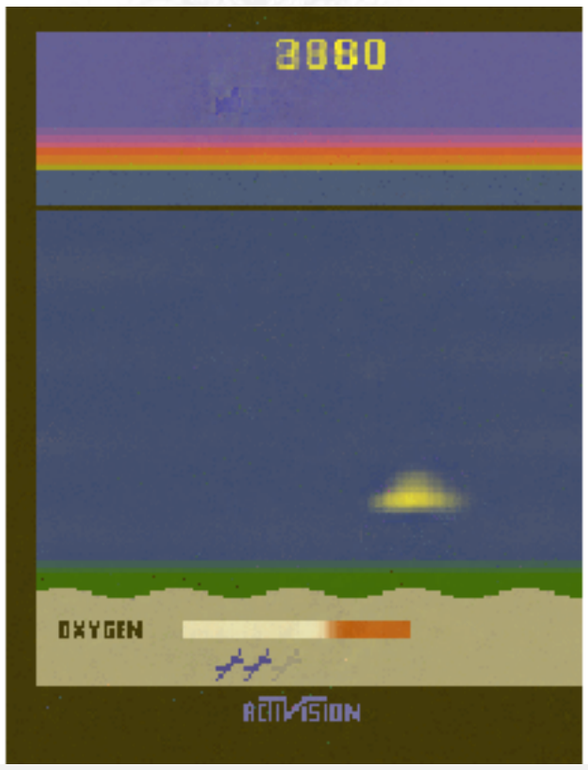
Moving Up



Oxygen/Swimmers



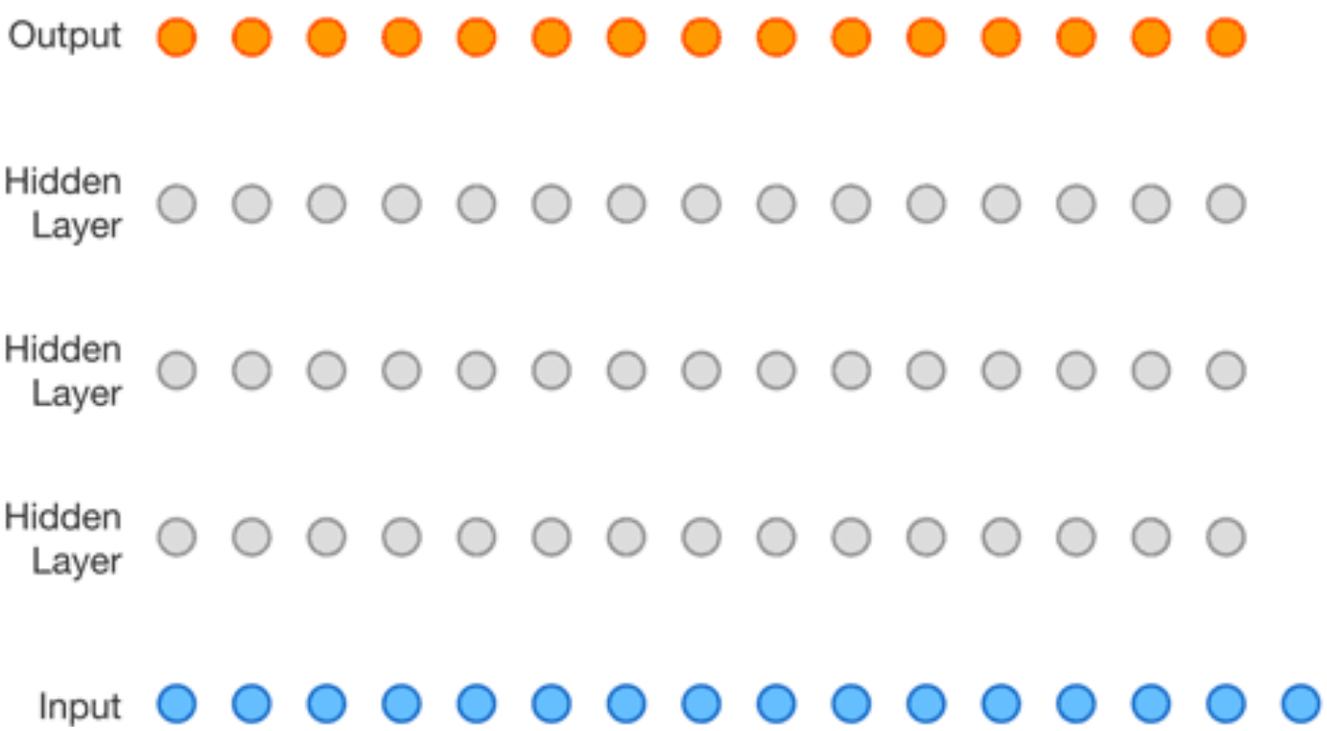
Score/Lives

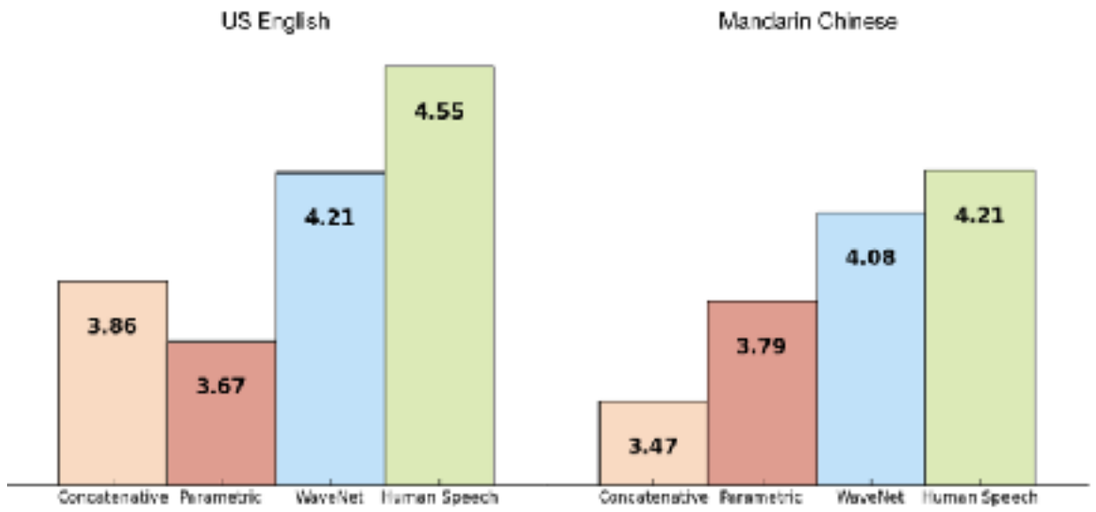


Moving Left

Text-to-Speech Synthesis

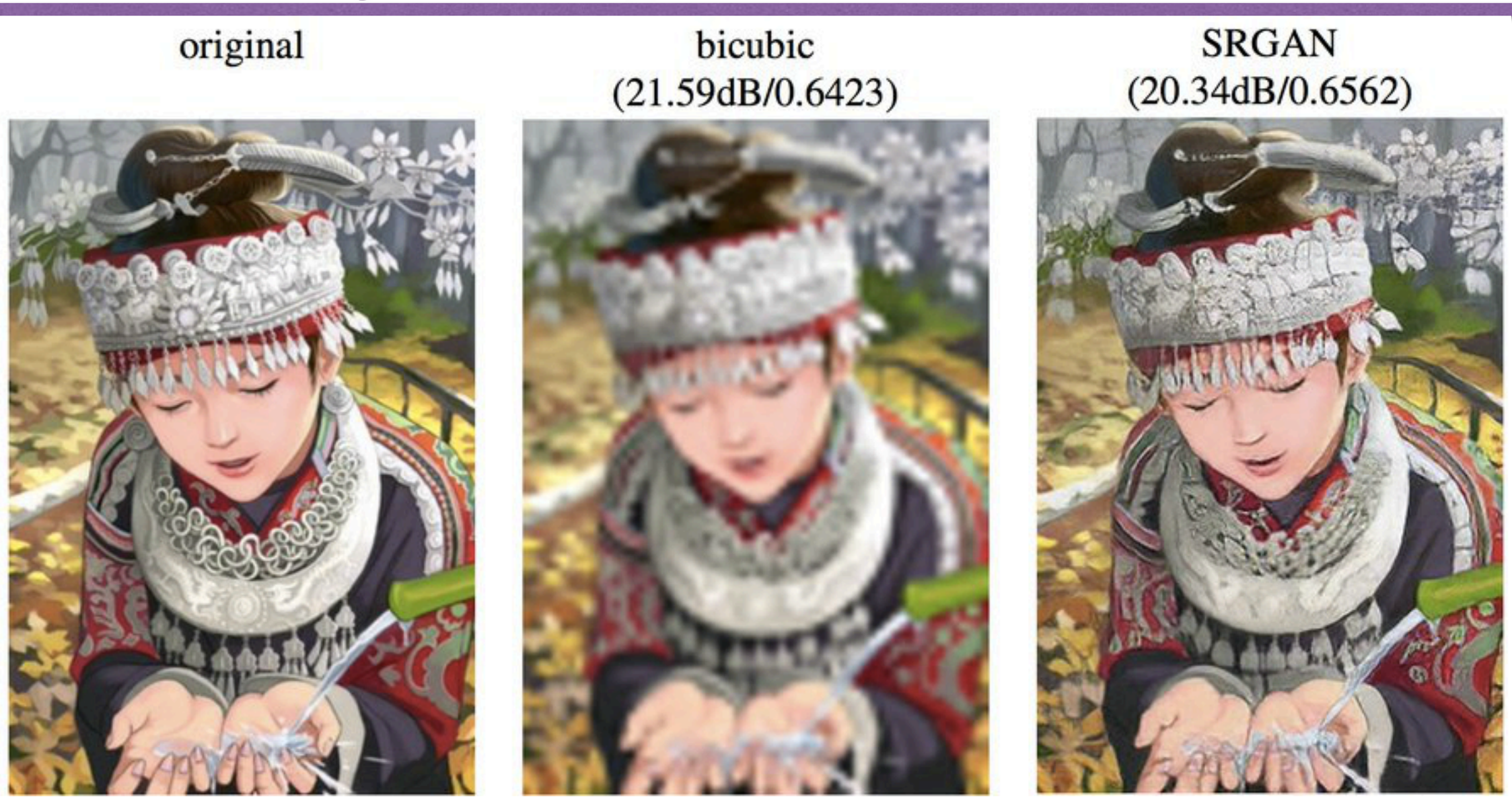
Auto-regressive for text-to-speech synthesis from raw waveforms.

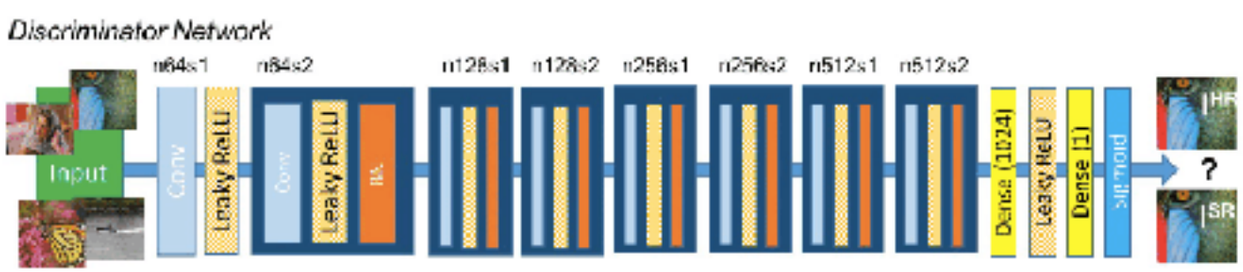




https://deepmind.com/blog/wavenet-generative-model-raw-audio/

Image Super-resolution





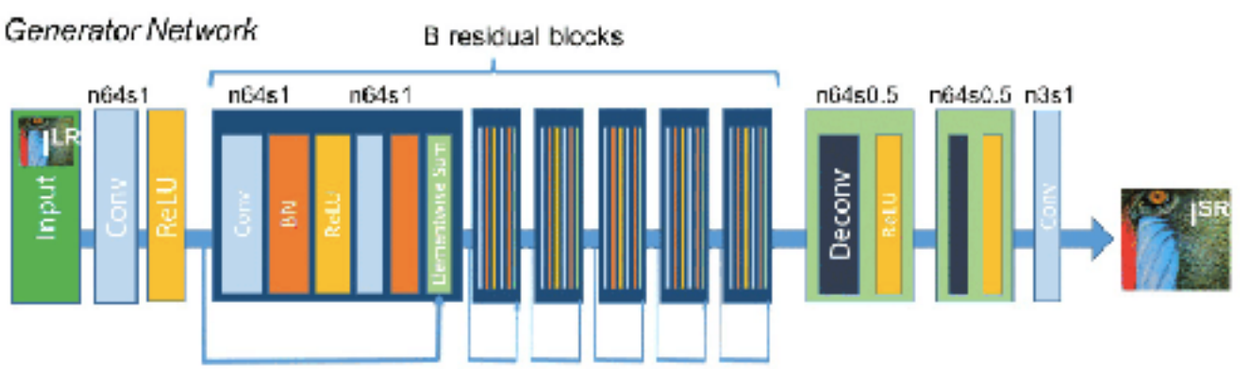


Photo-Realistic Single Image Super-Resolution Using a GAN

Summary

Some References

- 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).
- Bellemare, Marc G., Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. "Unifying Count-Based Exploration and Intrinsic Motivation." arXiv preprint arXiv:1606.01868 (2016).

Some References

- Alexander (Sasha) Vezhnevets, Mnih, Volodymyr, John Agapiou, Simon Osindero, Alex Graves, Oriol Vinyals, and Koray Kavukcuoglu. "Strategic Attentive Writer for Learning Macro-Actions." arXiv preprint arXiv:1606.04695 (2016).
- Gregor, Karol, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. "DRAW: A recurrent neural network for image generation." arXiv preprint arXiv:1502.04623 (2015).

- **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.

Some References

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.

Some References

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 in deep generative models." arXiv preprint arXiv:1401.4082 (2014).

Heess, Nicolas, Daniel Tarlow, and John Winn. "Learning to pass expectation propagation messages." In Advances in Neural Information Processing Systems, pp. 3219-3227. 2013.

Jitkrittum, Wittawat, Arthur Gretton, Nicolas Heess, S. M. Eslami, Balaji Lakshminarayanan, Dino Sejdinovic, and Zoltán Szabó. "Kernel-based just-in-time learning for passing expectation propagation messages." arXiv preprint arXiv:1503.02551 (2015).

Korattikara, Anoop, Vivek Rathod, Kevin Murphy, and Max Welling. "Bayesian dark knowledge." arXiv preprint arXiv:1506.04416 (2015).

Some References

Stochastic Optimisation

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).