Computer Science

CSC535: Probabilistic Graphical Models

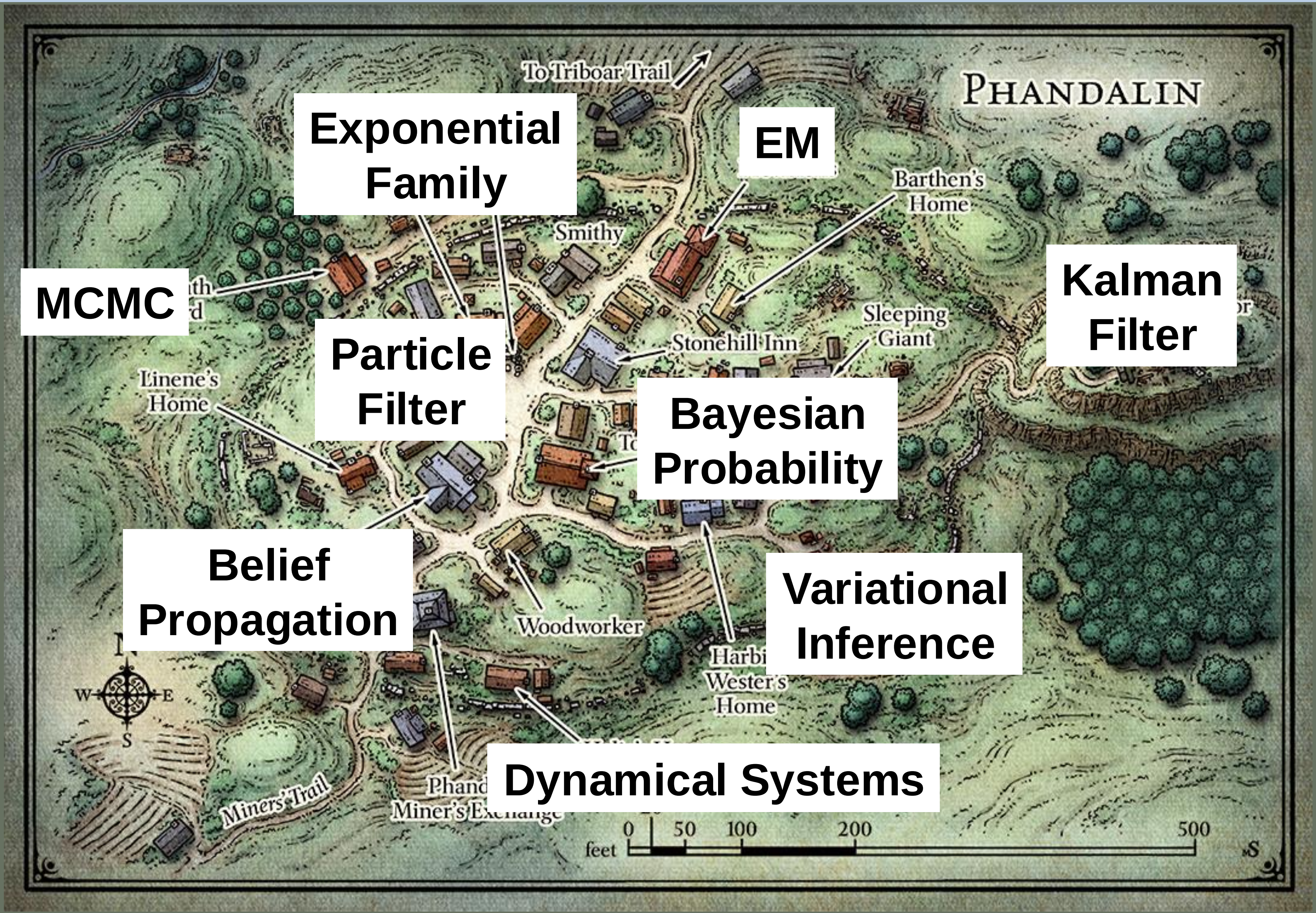
Course Wrap-Up

Prof. Jason Pacheco

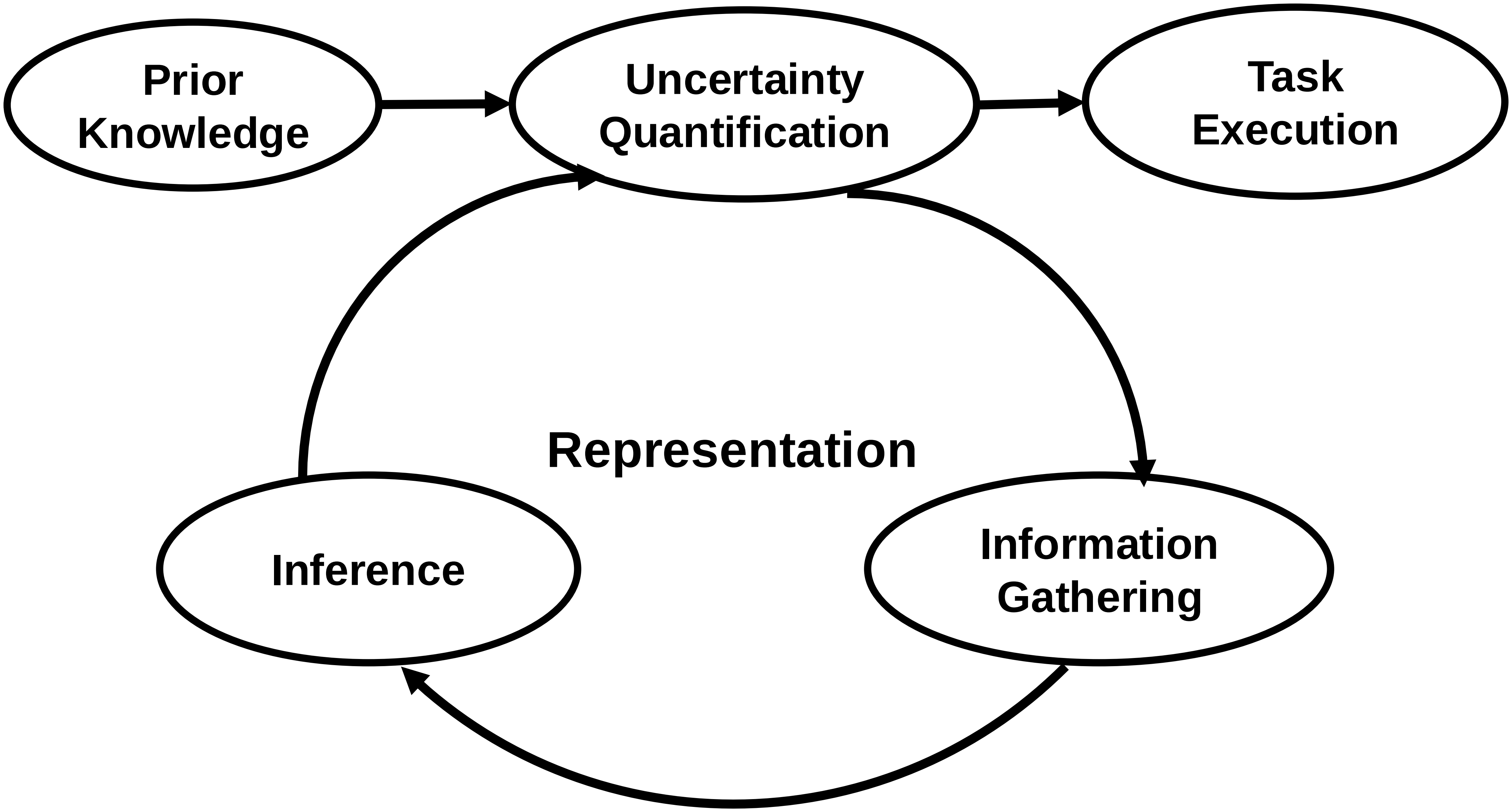
What we covered…

Probability and Statistics	Message Passing Algorithms	Monte Carlo Methods	Sequence Models and Dynamical Systems	Variational Inference	Bayesian Deep Learning
Probability primer, Bayesian statistics, PGMs, Exponential families	Elimination, Junction tree, Sum-product / max-product, Belief propagation, Viterbi decoding	Rejection sampling, Importance sampling, Metropolis-Hastings, Gibbs	Linear and switching state-space models, Kalman filter, Particle filter	Mean field, Stochastic variational, Bethe energy methods	Combining probability and deep learning models

Roadmap for ML Practice & Research

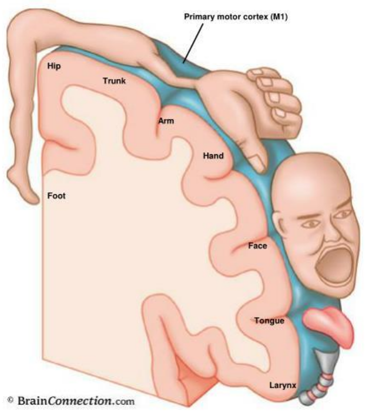


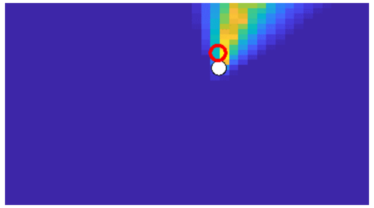
Probabilistic Reasoning

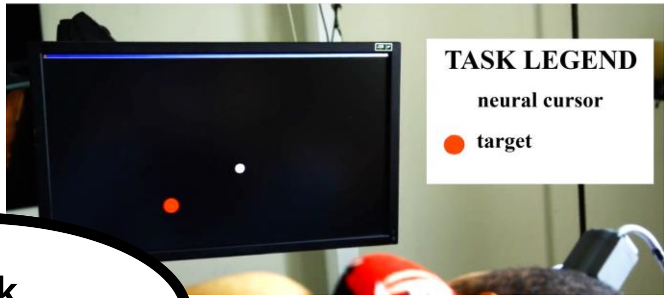


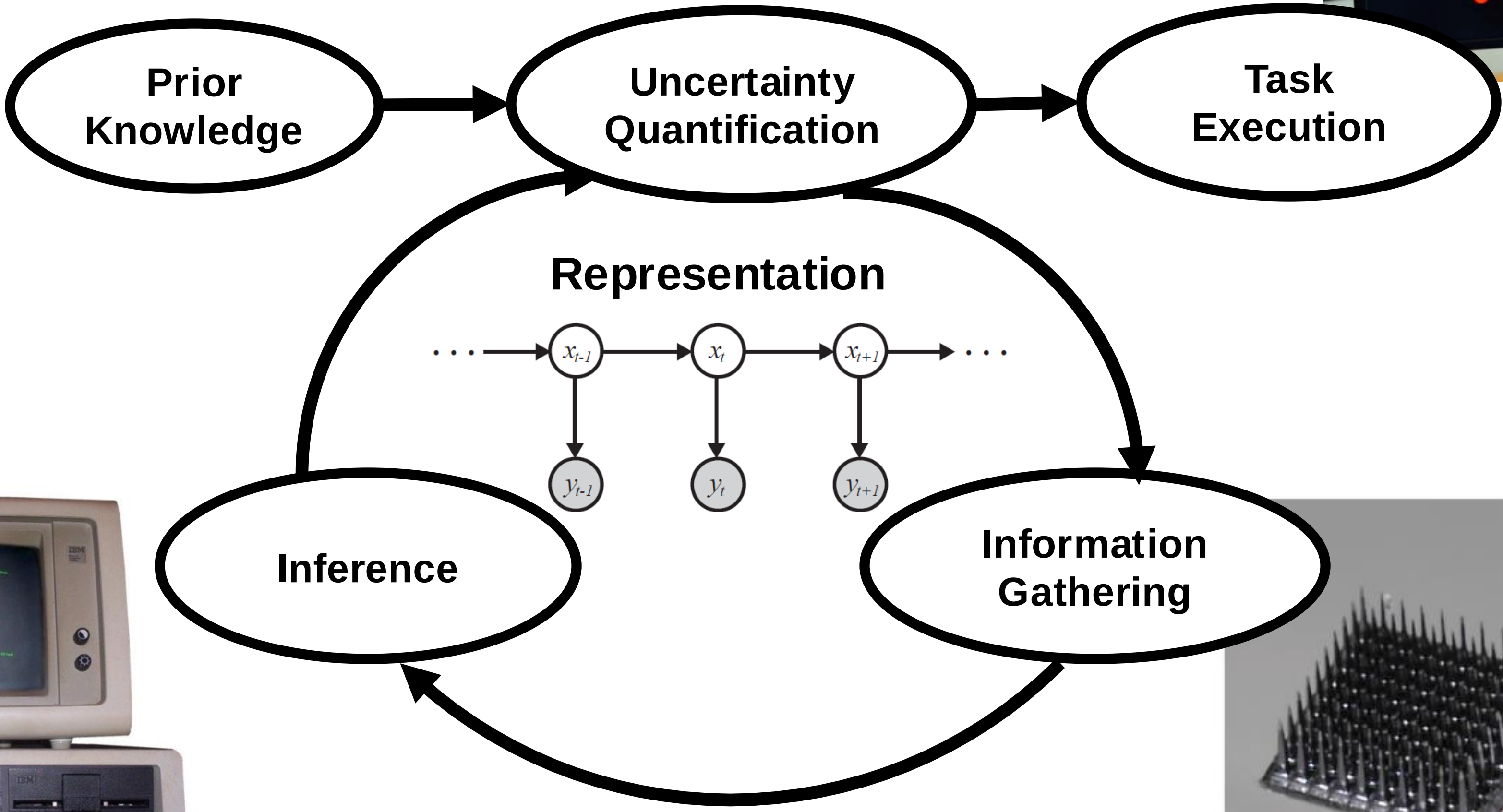
Block 12: "Multiscale Semi-Markov Model"

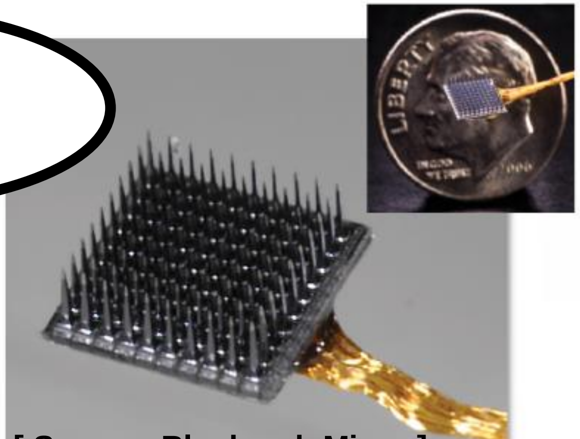
Probabilistic Reasoning





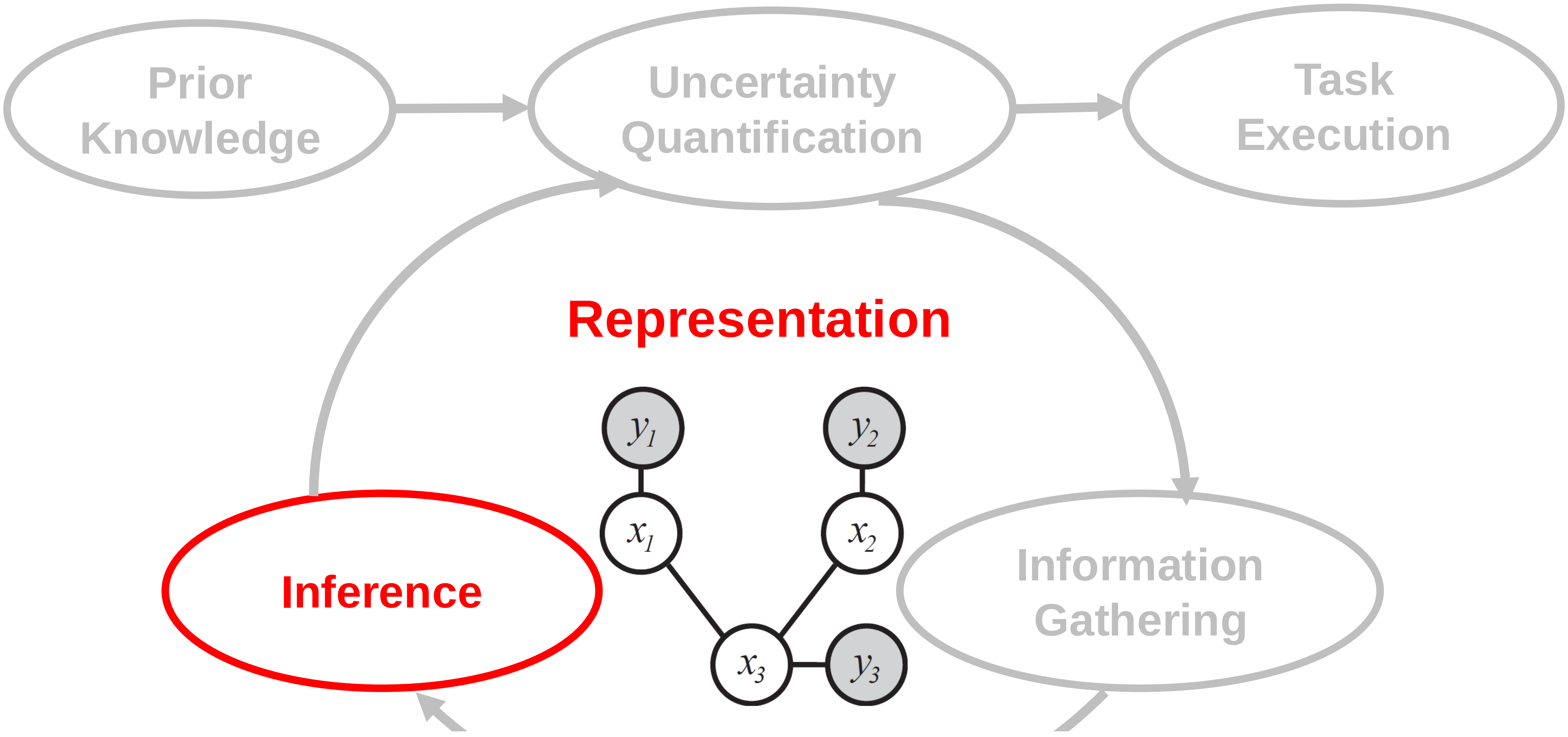






[Source: Blackrock Micro]

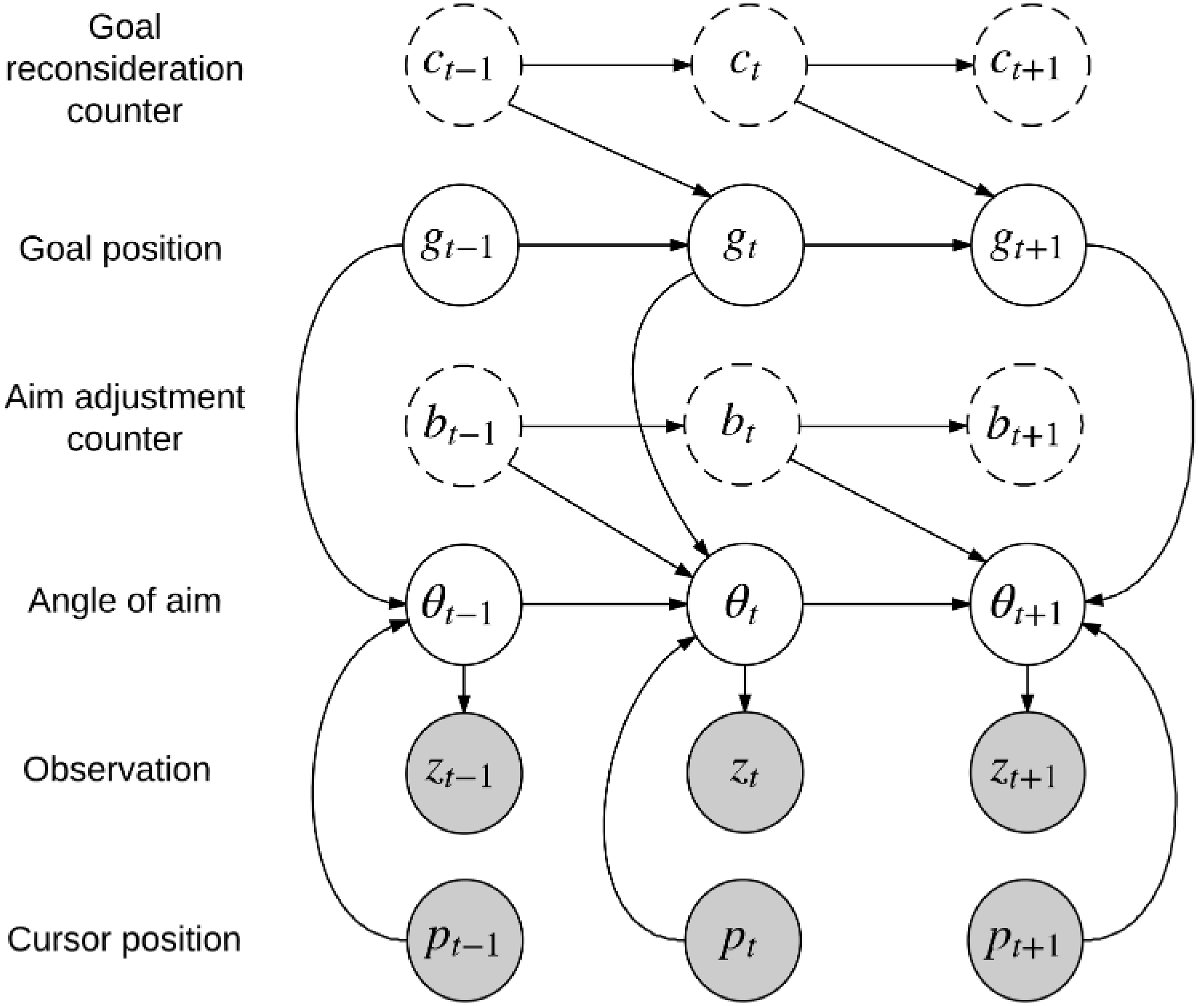
Probabilistic Reasoning

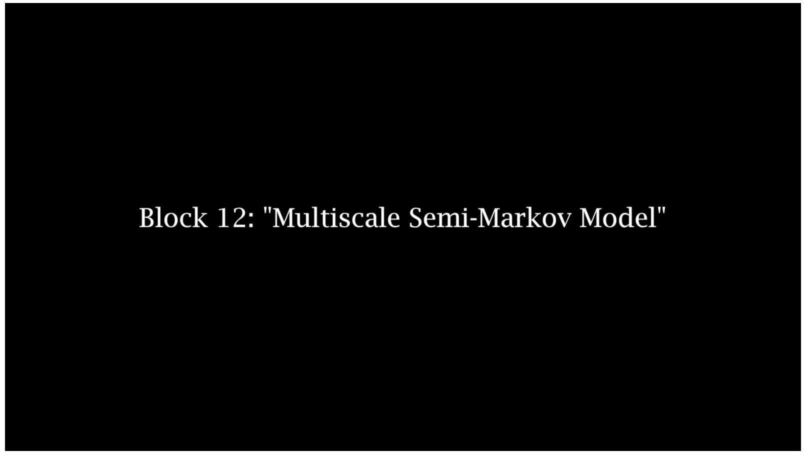


Facilitates development of efficient inference algorithms.

Intracortical Brain-Computer Interfaces (iBCI)

Multiscale Semi-Markov Model





Goal Given <u>observed</u> neural activity and cursor position <u>infer</u> user intended motion of cursor and goal location.

$$p(\theta_t, g_t \mid z_1^t, p_1^t)$$

Marginal Posterior

Observe Each model component sufficiently low-dimensional to discretize.

Approach Compute discrete BP messages

D. Milstein, J. Pacheco, et al. "Multiscale Semi-Markov Dynamics for Intracortical BCIs", NIPS, 2017.

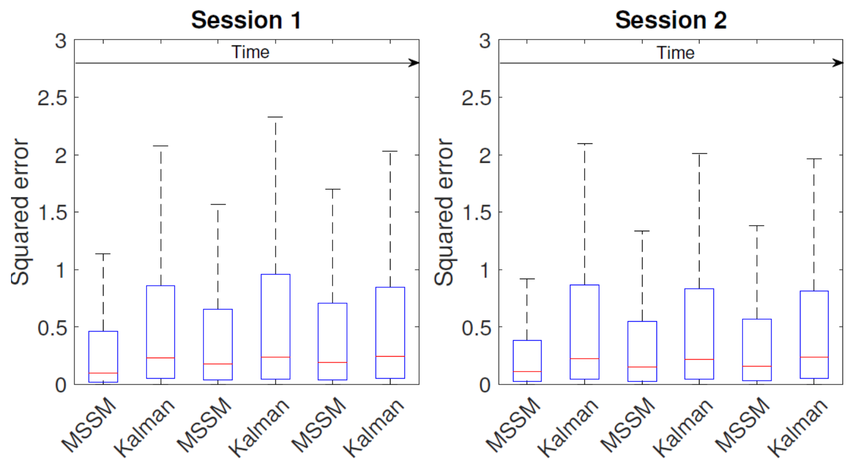
iBCI : MSSM vs. Kalman Filter

Multiscale Semi-Markov Model (MSSM)

Block 12: "Multiscale Semi-Markov Model"

Linear Gaussian (Kalman Filter)

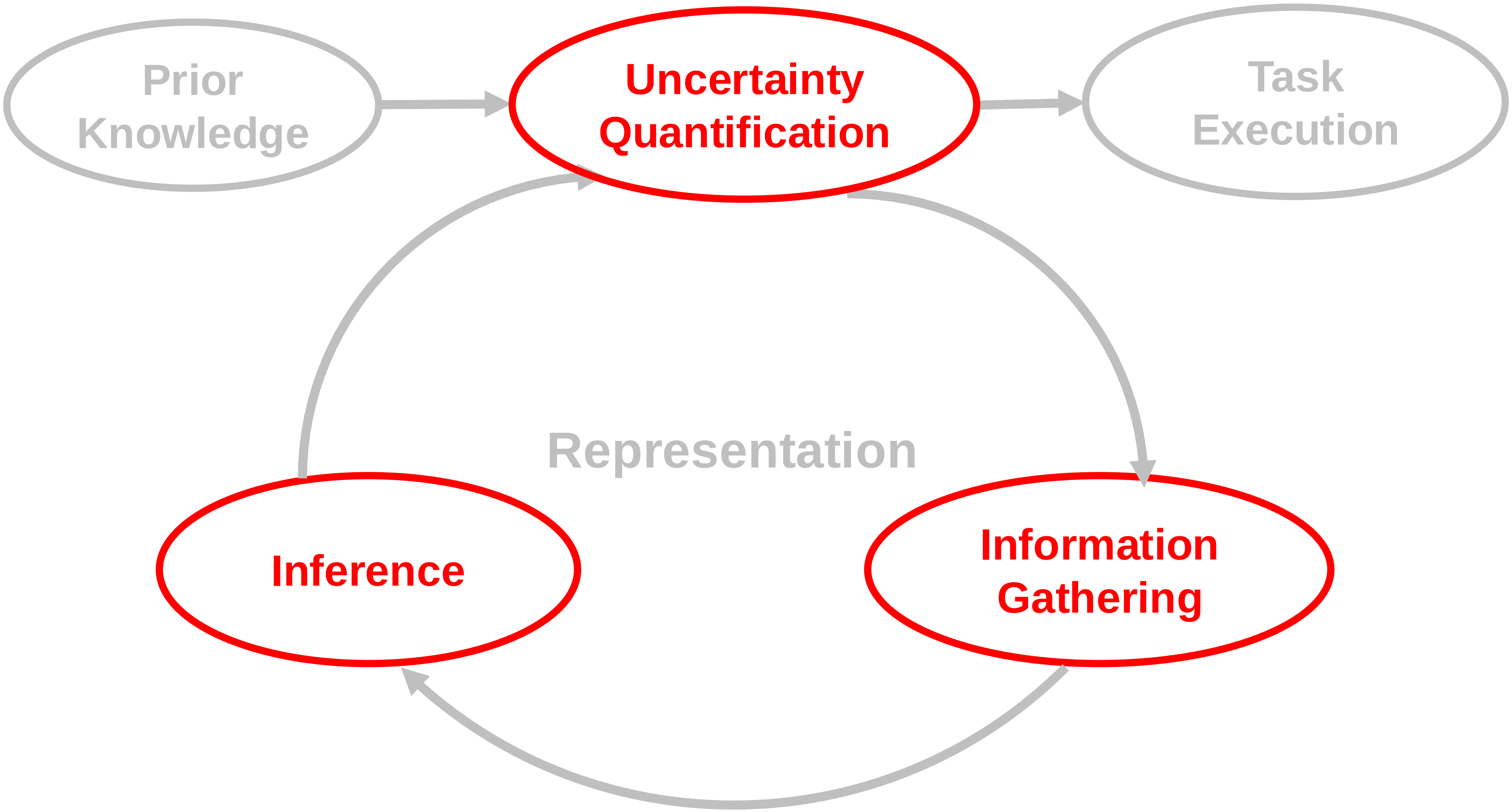
Block 13: "Kalman filter"



- Allows modelling of nonlinear / non-Gaussian dynamics
- **Dependency structure allows for discretization of each random variable**
- Discrete approximation belief propagation meets real-time constraints

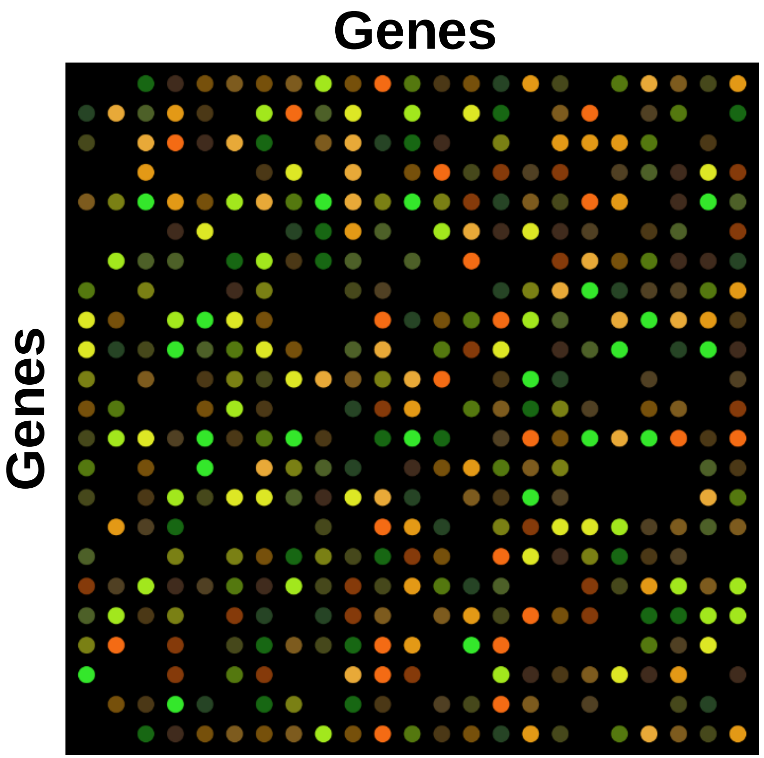
D. Milstein, J. Pacheco, et al. "Multiscale Semi-Markov Dynamics for Intracortical BCIs", NIPS, 2017.

Probabilistic Reasoning

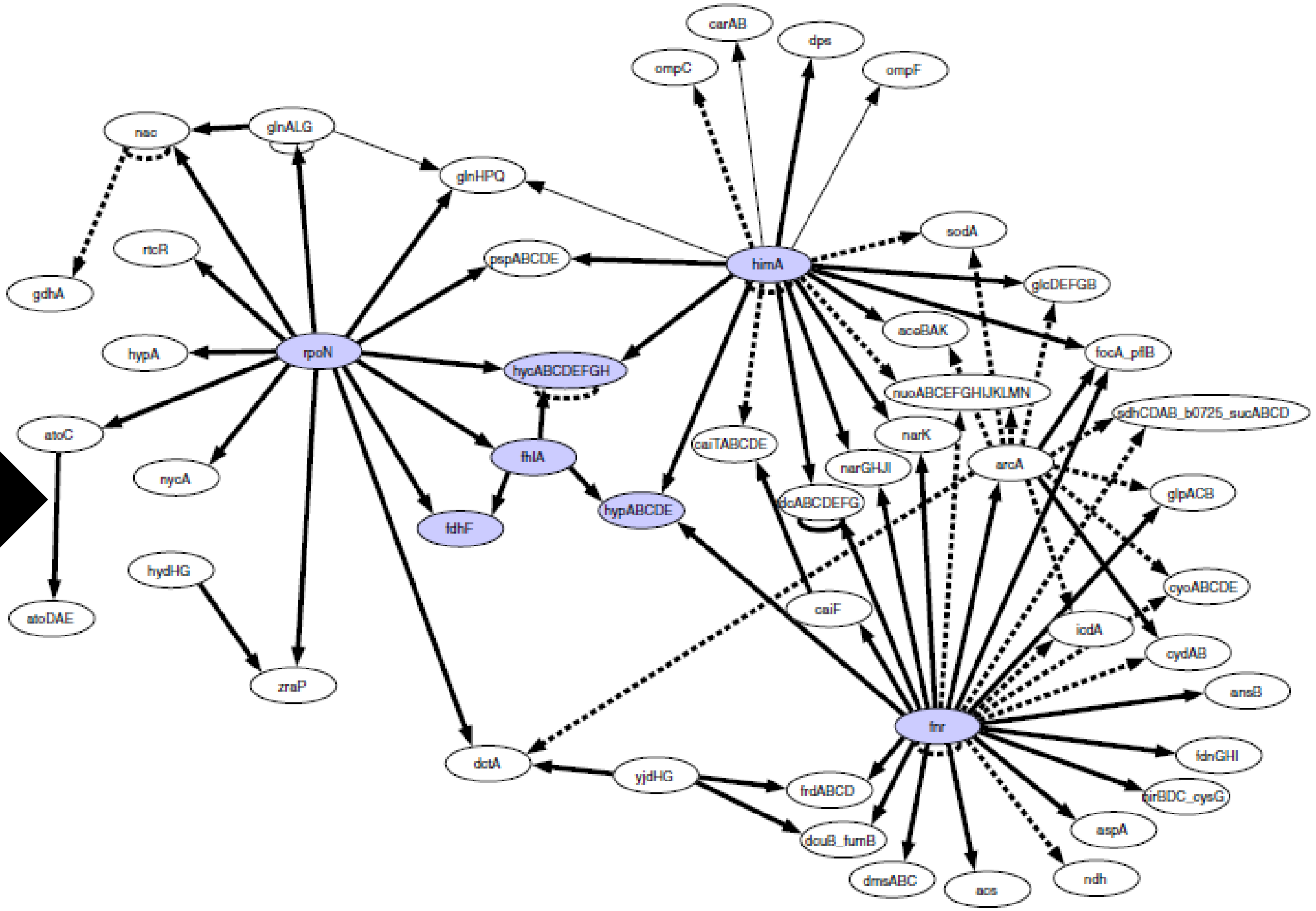


Example: Gene Regulatory Network

Gene Expression



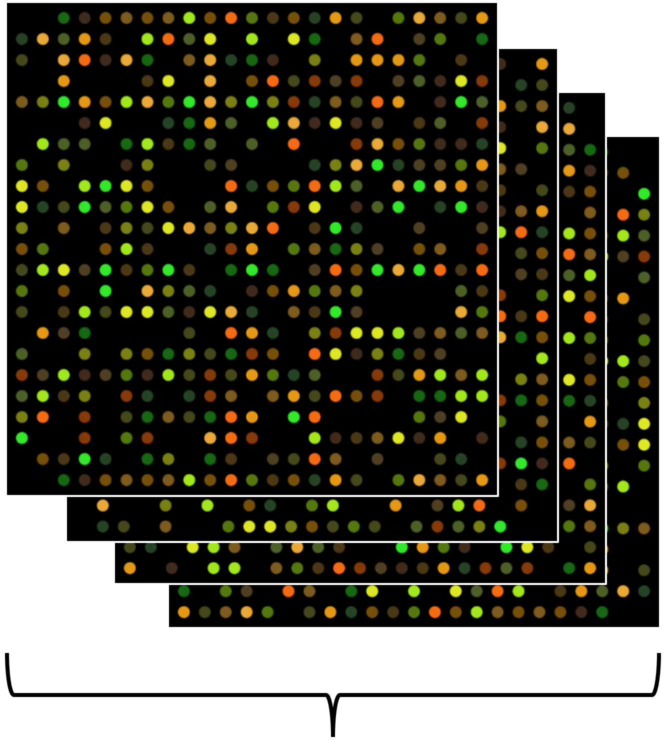
Regulatory Network



[Image: Bulcke et al., 2006]

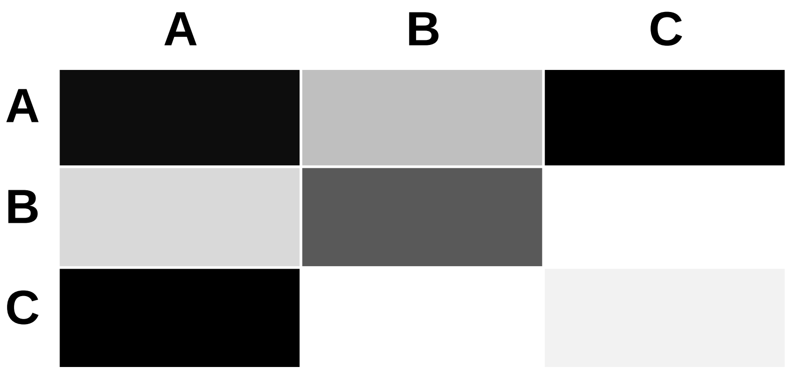
Goal: Estimate causal interaction network from expression data.

Identifying Causality

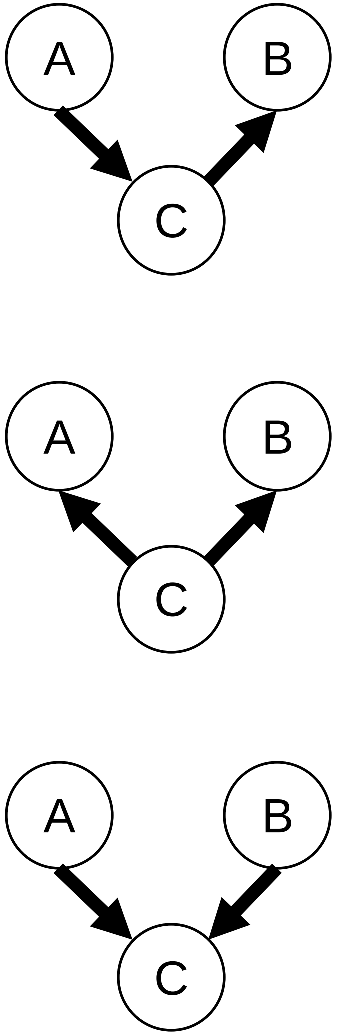


Dataset

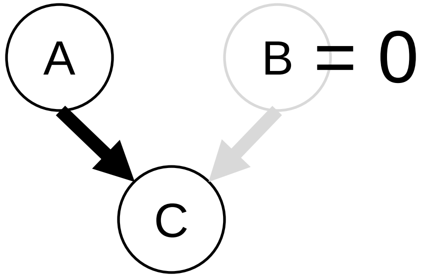
Covariance Matrix



Possible Graphs



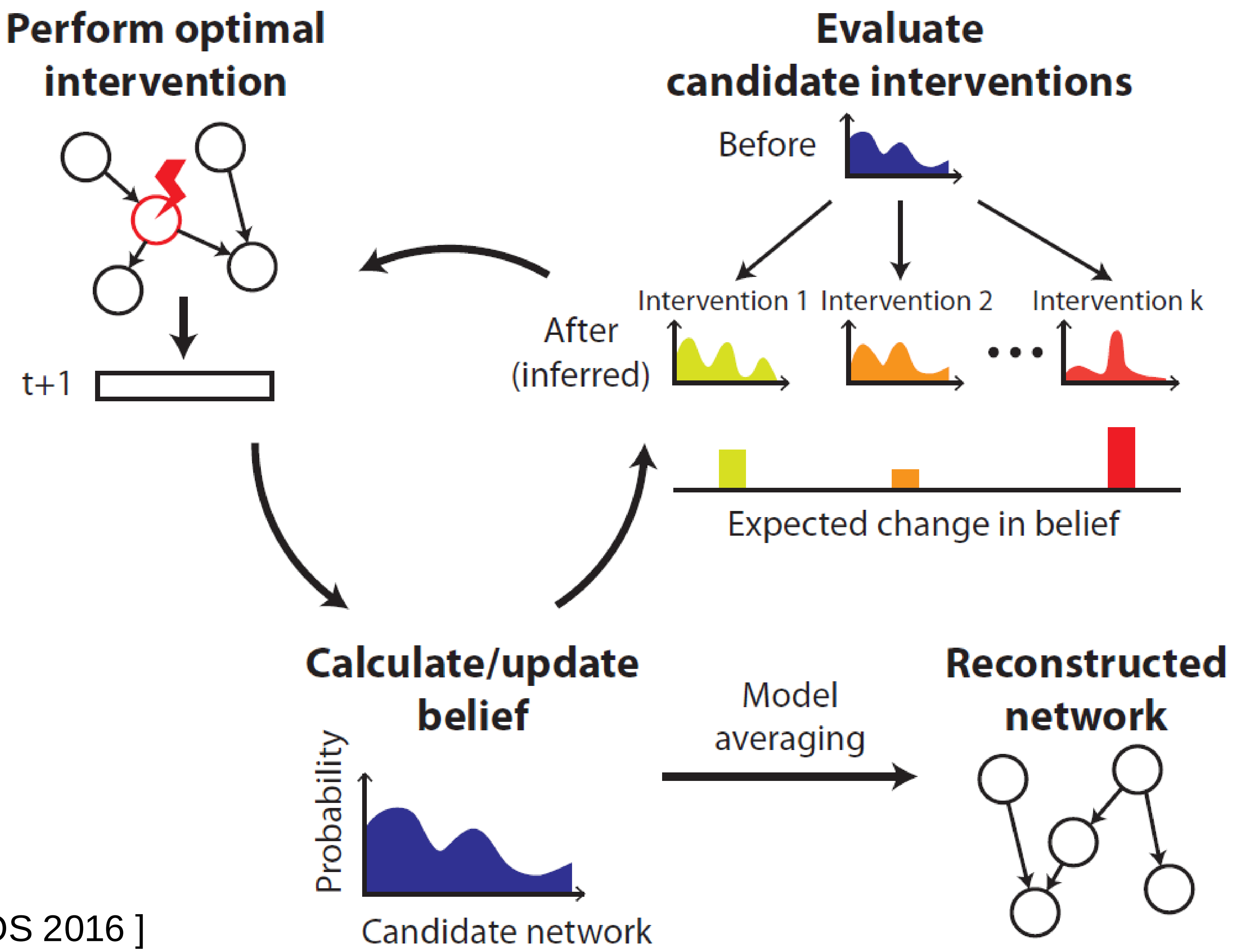
Cannot determine causality from correlations, need to perform <u>active interventions</u> …



Clamp node to fixed value.

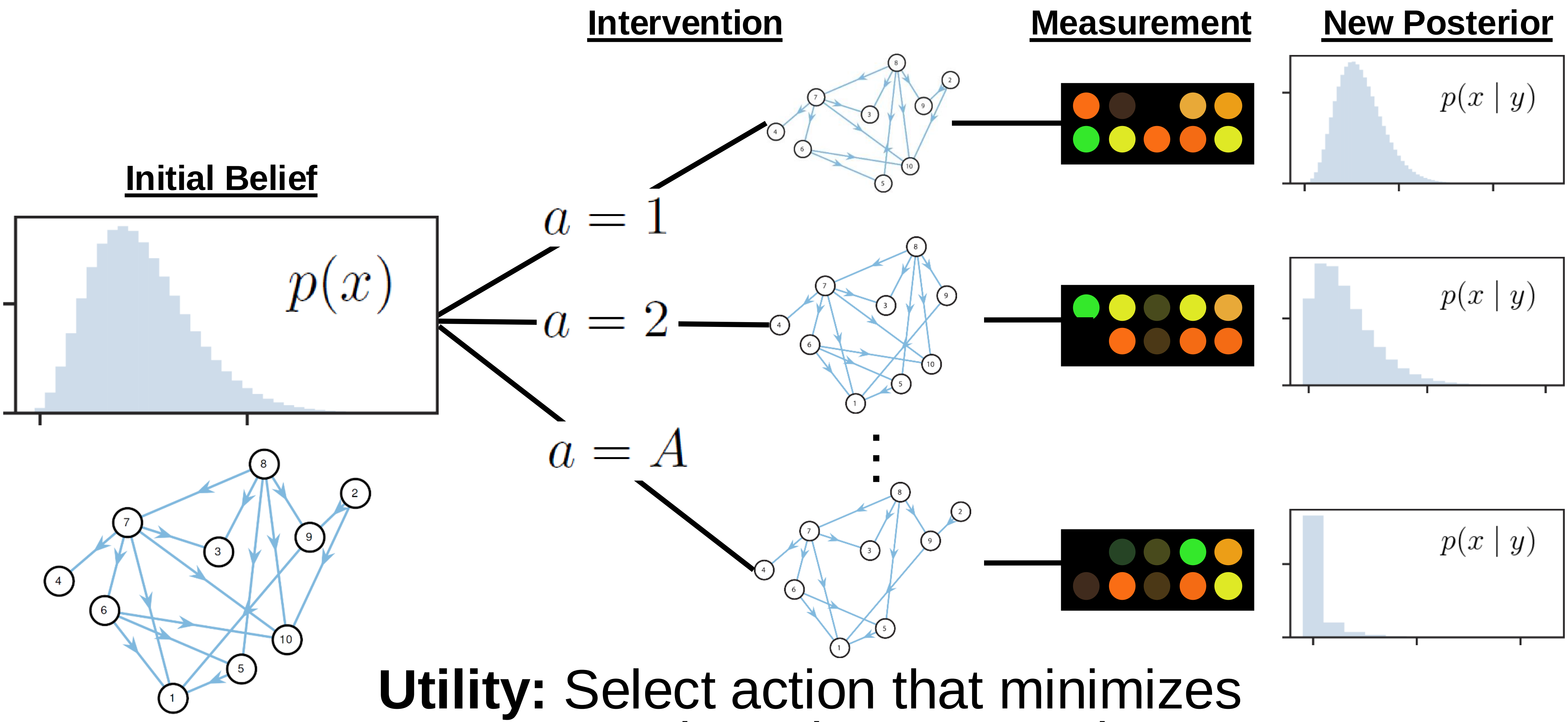
Gene Knockout

Bayesian Optimal Experiment Design



[Source: Cho et al., PLOS 2016]

Choosing Actions

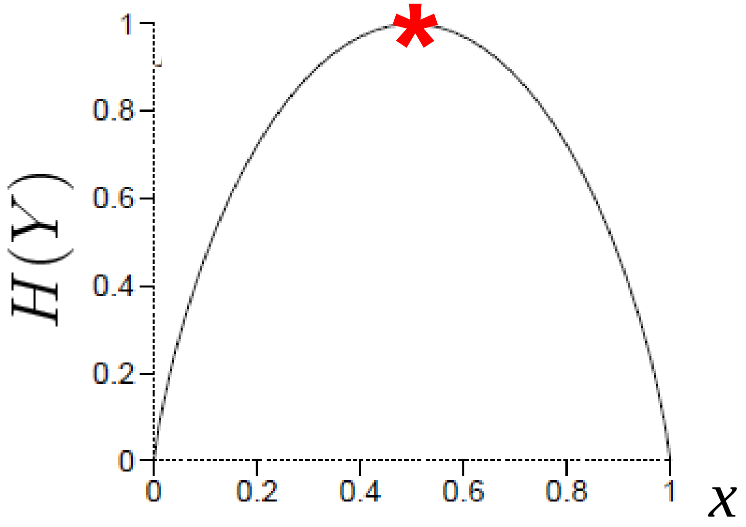


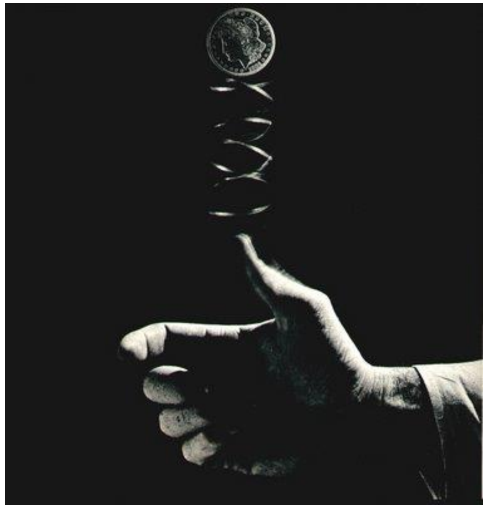
Utility: Select action that minimizes <u>uncertainty</u> about network

Uncertainty and Information

$$H(Y) = \mathbb{E}[-\log p(Y)]$$

Coin Flip Example: $Y \sim \mathrm{Bernoulli}(x)$



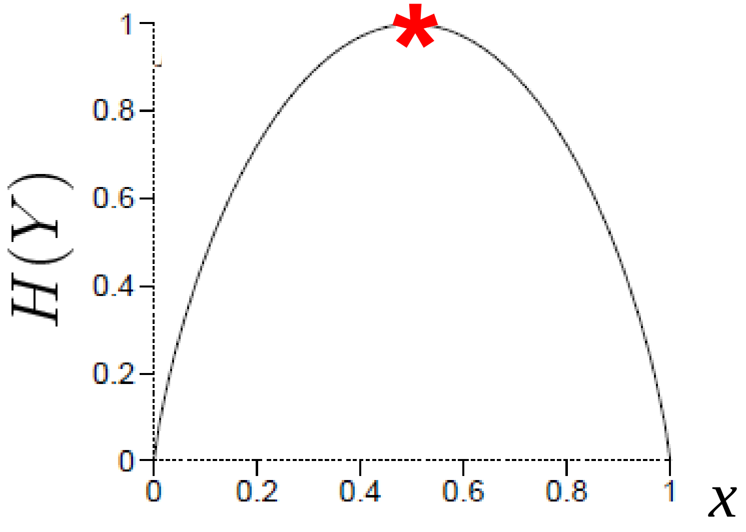


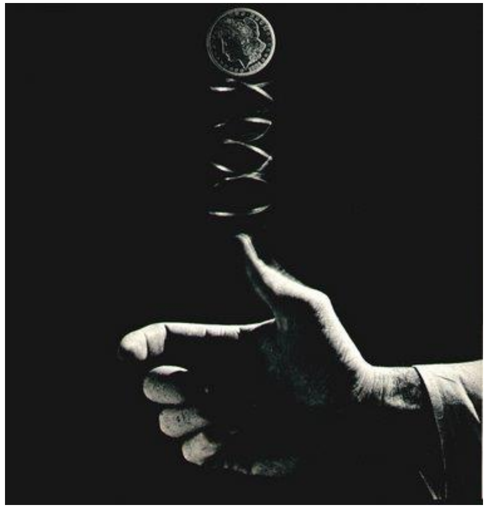
Maximum uncertainty when coin is <u>fair</u>.

Uncertainty and Information

$$H(Y) = \mathbb{E}[-\log p(Y)]$$

Coin Flip Example: $Y \sim \text{Bernoulli}(x)$





Maximum uncertainty when coin is fair.

Mutual Information

$$I(X;Y) = H(X) - H(X \mid Y)$$

- Measures entropy reduction after observing Y
- How much information does Y carry about X?

Computing MI as hard as doing inference.

Information-Theoretic Decision Making

At time t given history $\mathcal{Y}_{t-1} = \{y_1, a_1, \ldots, y_{t-1}, a_{t-1}\}$ choose *most informative* action $a = \{1, \ldots, A\}$:

$$a_t^* = \text{argmax}_a \, I_a(X; Y_t \mid \mathcal{Y}_{t-1})$$

<u>Problem 1</u> MI calculation requires posterior expectation:

$$I_a(X; Y_t \mid \mathcal{Y}_{t-1}) = \mathbb{E}\left[\log \frac{p_a(Y_t \mid X, \mathcal{Y}_{t-1})}{p_a(Y_t \mid \mathcal{Y}_{t-1})}\right]$$

Estimate over posterior samples $\{(x^i, y_t^i)\}_{i=1}^N \sim p(x \mid \mathcal{Y}_{t-1})p(y_t \mid x, a)$

Information-Theoretic Decision Making

$$I_a(X; Y_t \mid \mathcal{Y}_{t-1}) = \mathbb{E}\left[\log \frac{p_a(Y_t \mid X, \mathcal{Y}_{t-1})}{p_a(Y_t \mid \mathcal{Y}_{t-1})}\right]$$

Problem 2 Requires pointwise evaluation of log-predictive dist'n:

$$\log p_a(y_t \mid \mathcal{Y}_{t-1}) = \log \int p(x \mid \mathcal{Y}_{t-1}) p(y_t \mid x, a)\, dx$$

Plug-in estimate over *more* samples $\{x^{ij}\}_{j=1}^M \sim p(x \mid y_t^i, \mathcal{Y}_{t-1}, a)$:

$$\hat{I}_a = \frac{1}{N} \sum_{i=1}^N \log \frac{p_a(y_t^i \mid x^i, a)}{\frac{1}{M} \sum_{j=1}^M p(y_t^i \mid x^{ij}, a)}$$

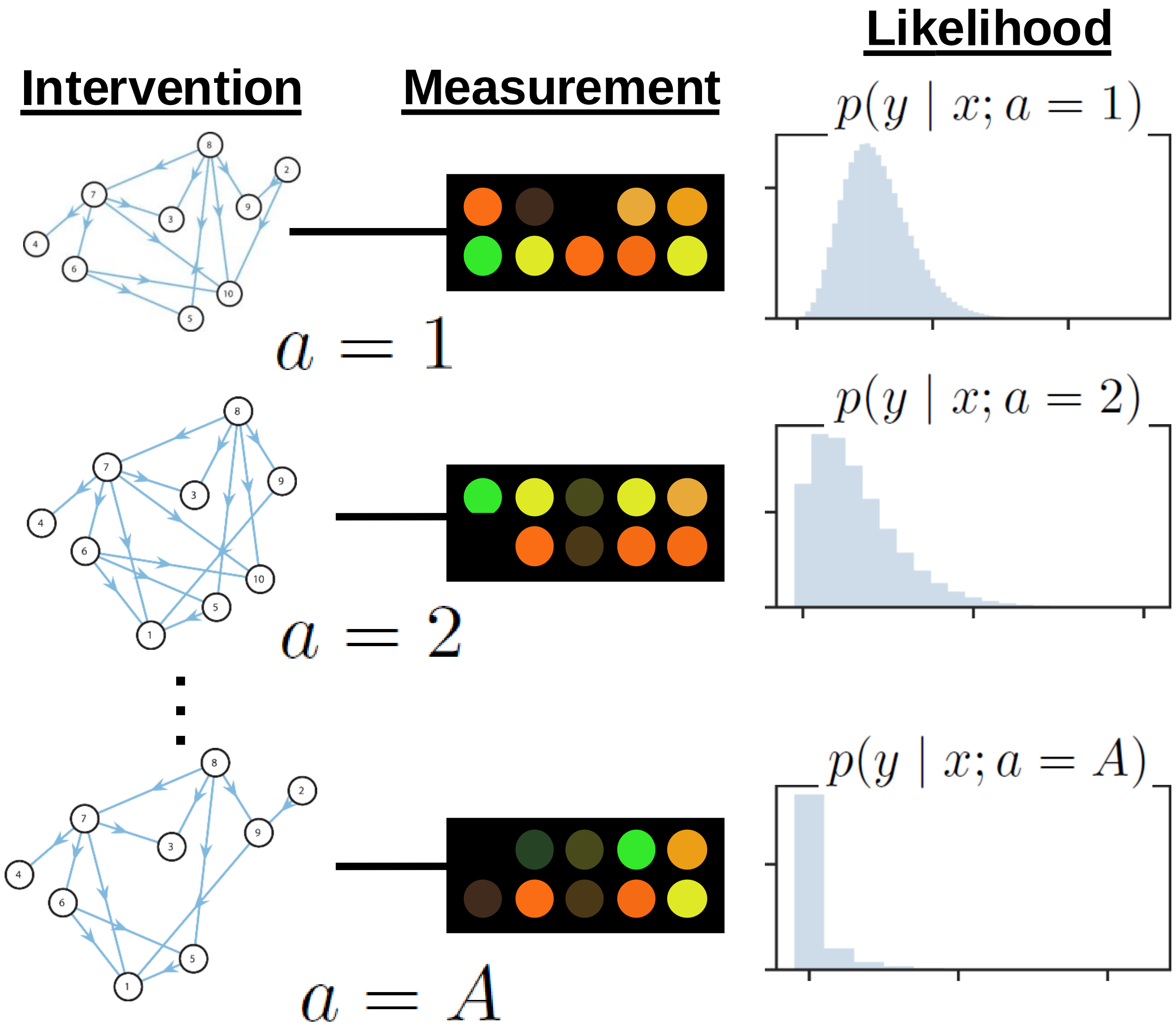
Empirical Information Planning

Given joint samples under each action compute Monte carlo empirical mean estimate:

$$\hat{I}_a = \frac{1}{N}\sum_{i=1}^{N} \log \frac{p_a(y^i \mid x^i, a)}{\frac{1}{M}\sum_{j=1}^{M} p(y^i \mid x^{ij}, a)}$$

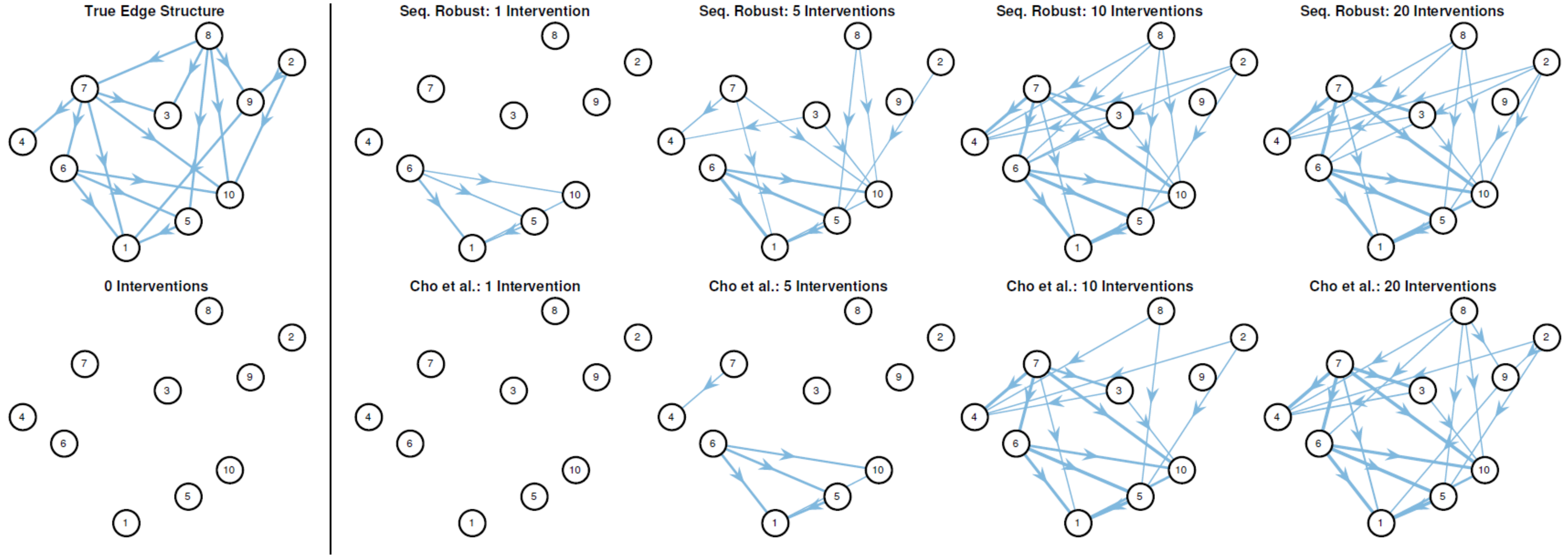
Planning based on estimated MI:

$$a^* = \arg\max_a \hat{I}_a(X;Y)$$



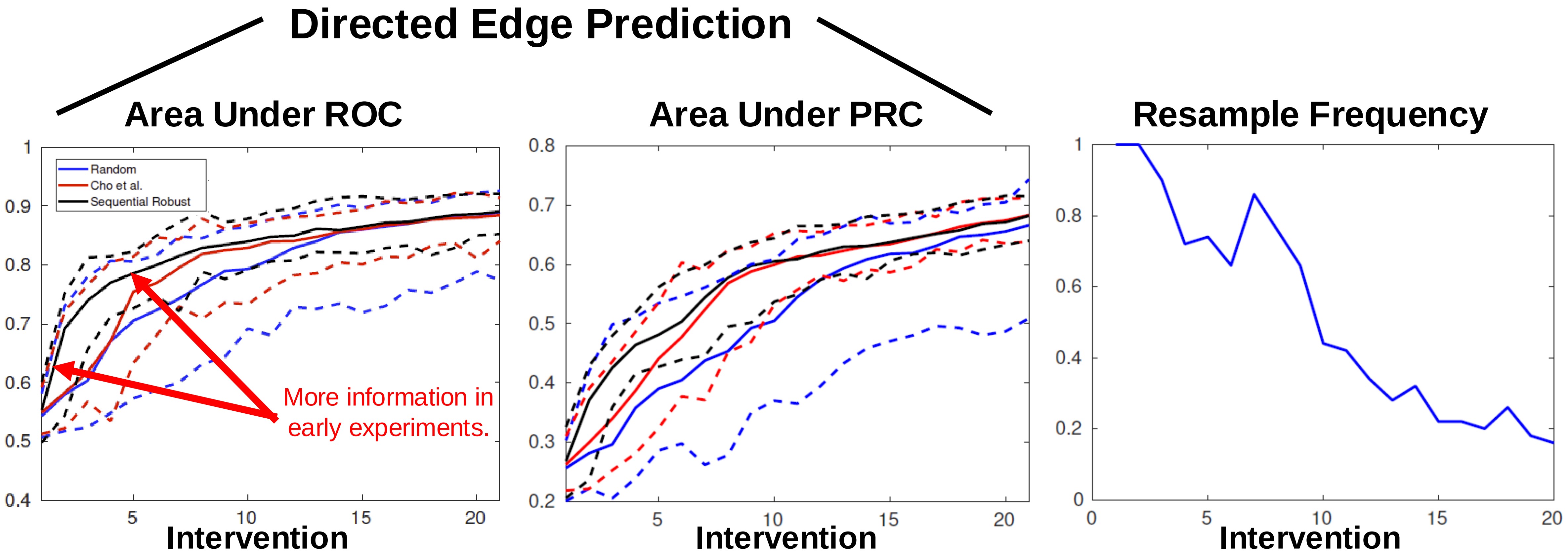
Problem: MI estimates are biased for finite samples

Gene Regulatory Network Inference



Edge width proportional to posterior probability

Gene Regulatory Network Inference



- Higher precision/recall w/ fewer interventions than non-robust method
- Resampling frequencies reduces as potential information gain is less

There's so much more to cover…

Models & Applications	AI Libraries	Representation Learning	Bayesian Nonparametrics	Advanced MCMC	Still more…
Course was mostly focused on algorithms, limited attention to modelling	Collection of tools for larger-scale AI	Unsupervised representation learning from structured data	A class of probability models where model complexity is inferred from the data	Avoiding random walk dynamics and allowing parallel computation	

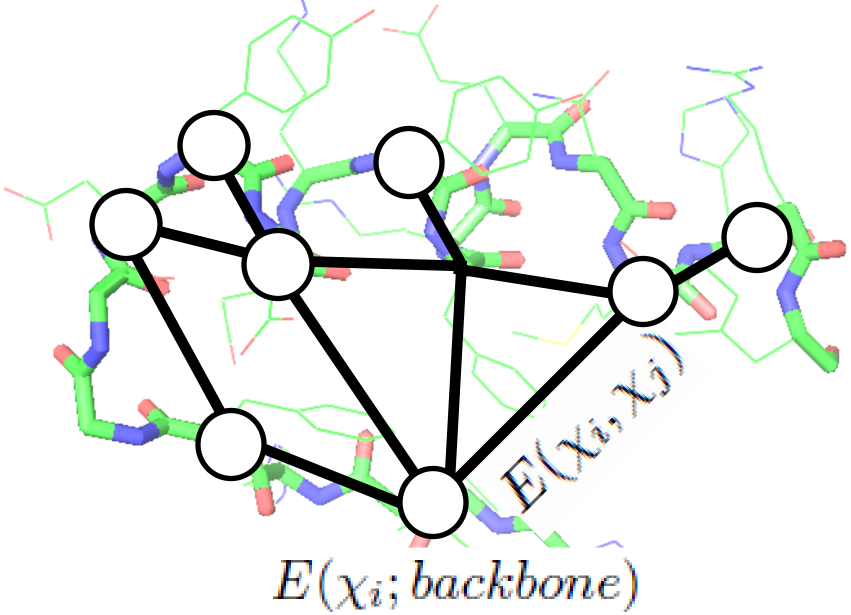
There’s so much more to cover…

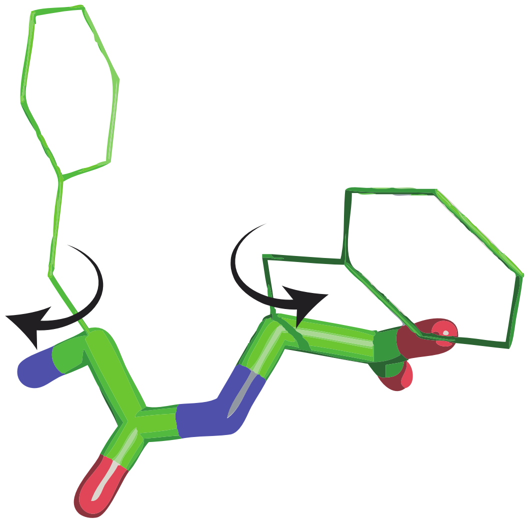
Models & Applications	**AI Libraries**	**Representation Learning**	**Bayesian Nonparametrics**	**Advanced MCMC**	**Still more…**
Course was mostly focused on algorithms, limited attention to modelling	Collection of tools for larger-scale AI	Unsupervised representation learning from structured data	A class of probability models where model complexity is inferred from the data	Avoiding random walk dynamics and allowing parallel computation	

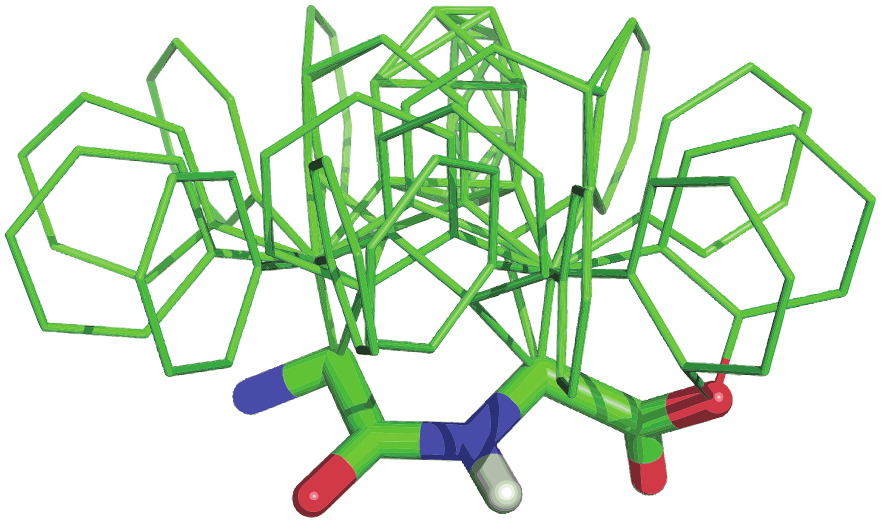
Protein Side-Chain Structure Prediction

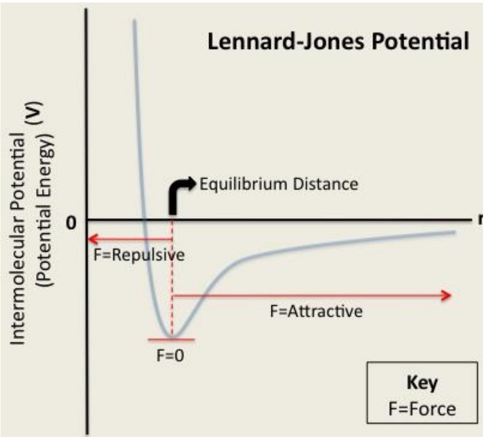
- A protein is a sequence of *amino acids*, each with a *side-chain*
- Side-chain structure prediction is MAP in pairwise MRF:

Pacheco et al., ICML 2015



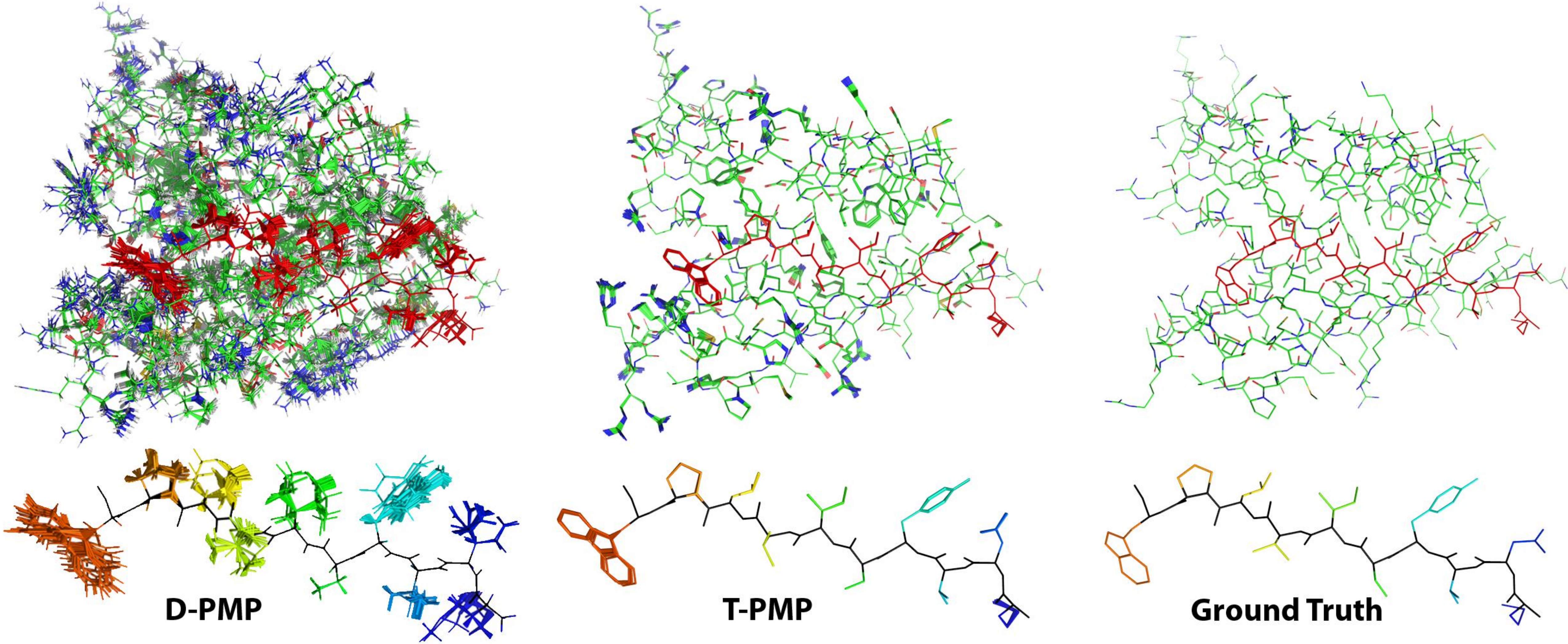






- Pairwise potentials describe repulsive (Pauli exclusion) and attractive (van der Waals force) energetic interactions
- Predicting structure lets biochemists better understand and predict function

Protein Side-Chain Structure Prediction



- Qualitative example of side-chain predictions for one protein.
- Energy evaluated via state-of-the-art Rosetta package.

Pacheco et al., ICML 2015

There's so much more to cover…

Models & Applications	AI Libraries	Representation Learning	Bayesian Nonparametrics	Advanced MCMC	Still more…
Course was mostly focused on algorithms, limited attention to modelling	Collection of tools for larger-scale AI	Unsupervised representation learning from structured data	A class of probability models where model complexity is inferred from the data	Avoiding random walk dynamics and allowing parallel computation	

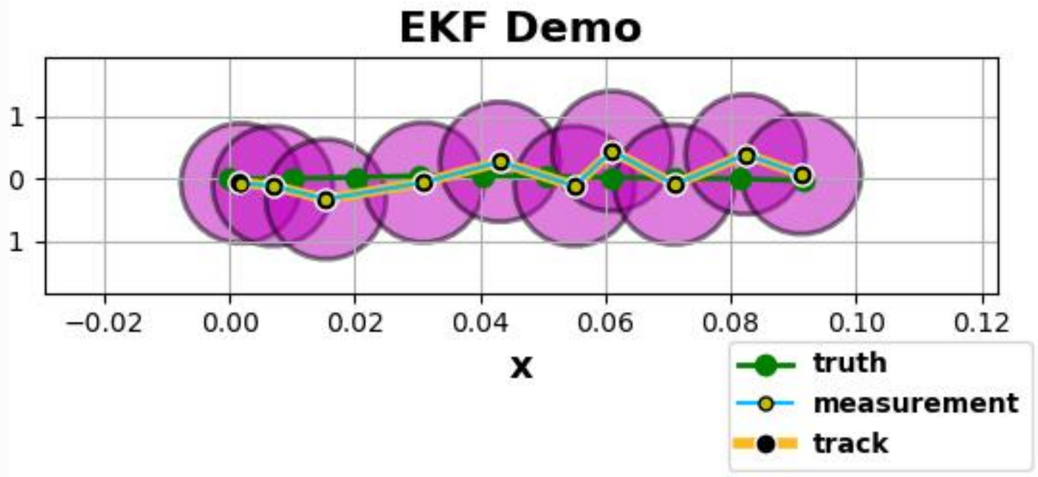
Libraries

Pyro.AI

- Python library for probabilistic programming
- Open-sourced by Uber AI Labs
- Implements many algorithms we have discussed (+more)
- Built on Pytorch / efficient inference via GPU

```
[ ]: def model(data):
         # a HalfNormal can be used here as well
         R = pyro.sample('pv_cov', dist.HalfCauchy(2e-6)) * torch.eye(4)
         Q = pyro.sample('measurement_cov', dist.HalfCauchy(1e-6)) * torch.eye(2)
         # observe the measurements
         pyro.sample('track_{}'.format(i), EKFDistribution(xs_truth[0], R, ncv,
                                                            Q, time_steps=num_frames),
                     obs=data)

     guide = AutoDelta(model)  # MAP estimation
```

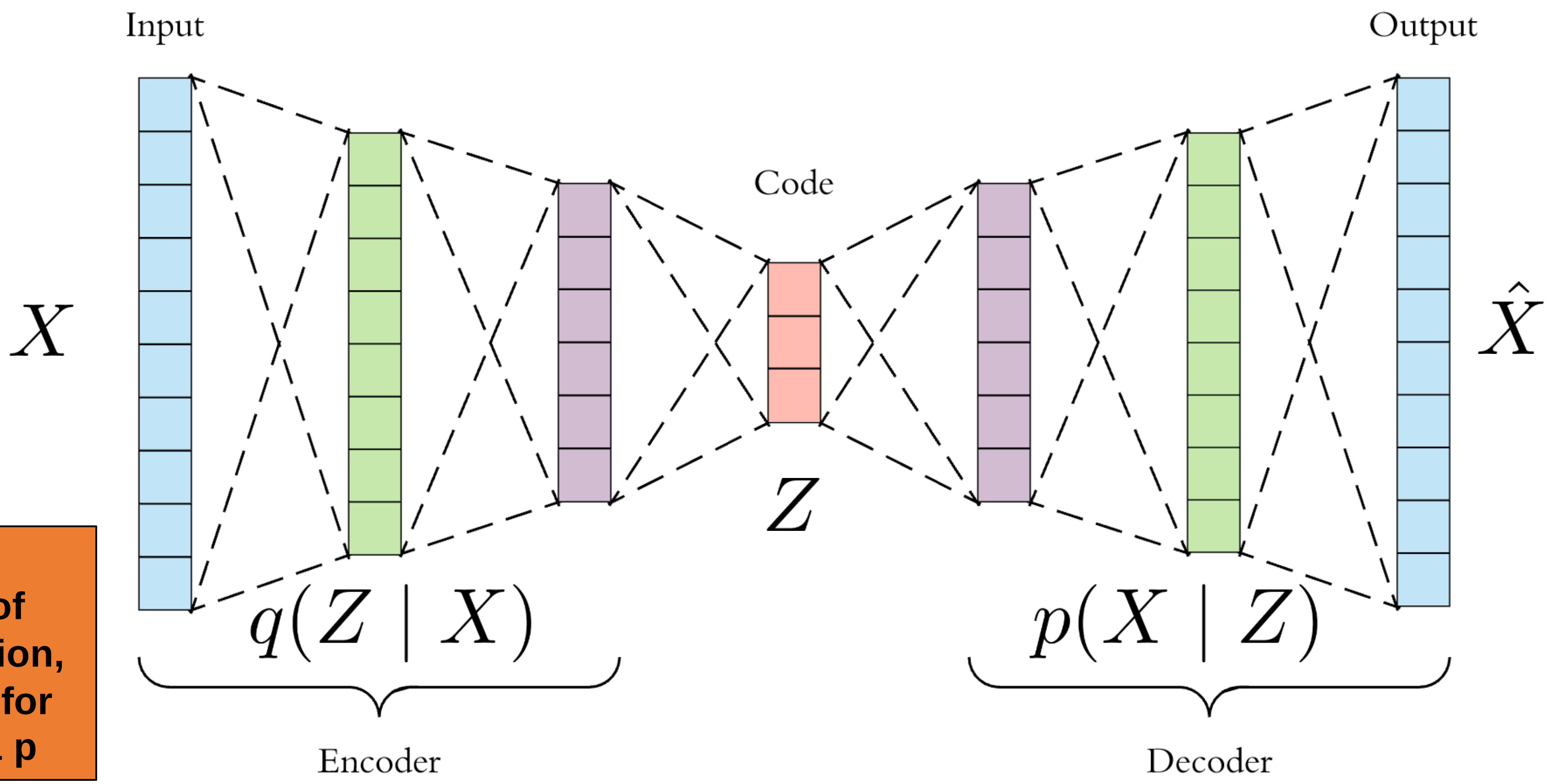
```
[ ]: optim = pyro.optim.Adam({'lr': 2e-2})
     svi = SVI(model, guide, optim, loss=Trace_ELBO(retain_graph=True))

     pyro.set_rng_seed(0)
     pyro.clear_param_store()

     for i in range(250 if not smoke_test else 2):
         loss = svi.step(zs)
         if not i % 10:
             print('loss: ', loss)
```



There's so much more to cover…

Models & Applications	AI Libraries	Representation Learning	Bayesian Nonparametrics	Advanced MCMC	Still more…
Course was mostly focused on algorithms, limited attention to modelling	Collection of tools for larger-scale AI	Unsupervised representation learning from structured data	A class of probability models where model complexity is inferred from the data	Avoiding random walk dynamics and allowing parallel computation	

Variational Autoencoder



NN learns parameters of some distribution, e.g. mean/var for Gaussian q & p

NOTE: This is completely *unsupervised*

Train by minimizing reconstruction loss and fit to marginal:

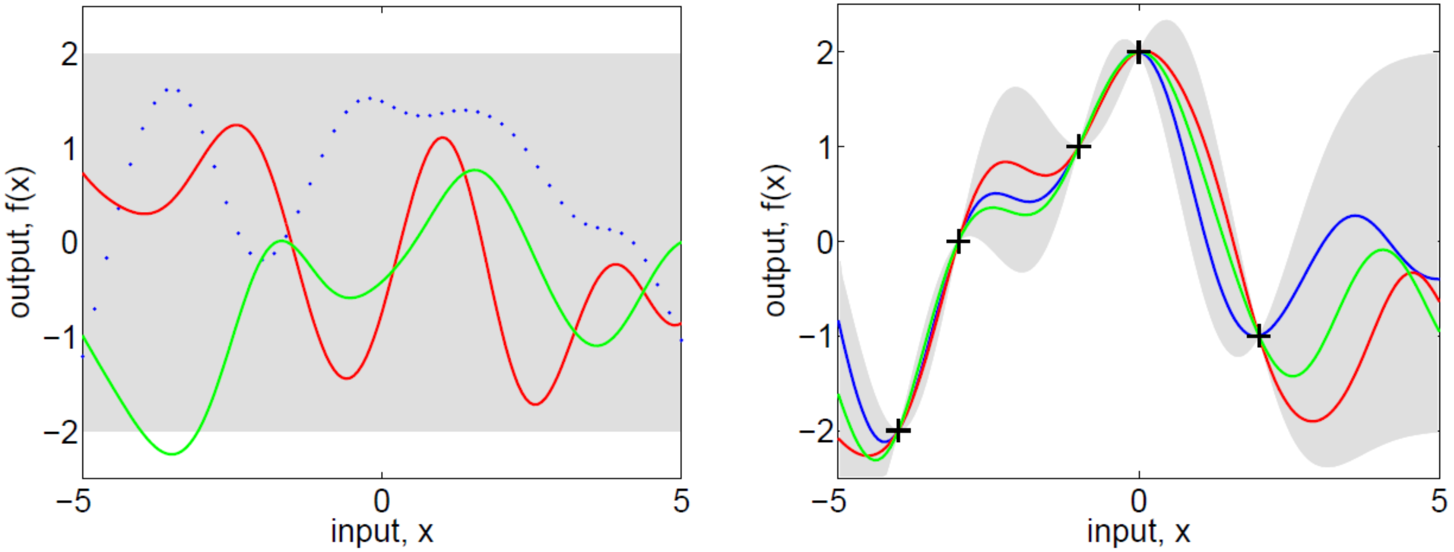
$$\min \mathcal{L}(x, \hat{x}) + KL(q(z \mid x) \| p(z))$$

There's so much more to cover…

Models & Applications	AI Libraries	Representation Learning	Bayesian Nonparametrics	Advanced MCMC	Still more…
Course was mostly focused on algorithms, limited attention to modelling	Collection of tools for larger-scale AI	Unsupervised representation learning from structured data	A class of probability models where model complexity is inferred from the data	Avoiding random walk dynamics and allowing parallel computation	

Gaussian Processes (GPs)

Distribution over random continuous functions…



(a), prior (b), posterior

$$\mathbf{f}_* \sim \mathcal{N}(\mathbf{0}, \underbrace{K(X_*, X_*)})$$

Kernel function encodes correlation between evaluation points in the domain

GPs are generative models…

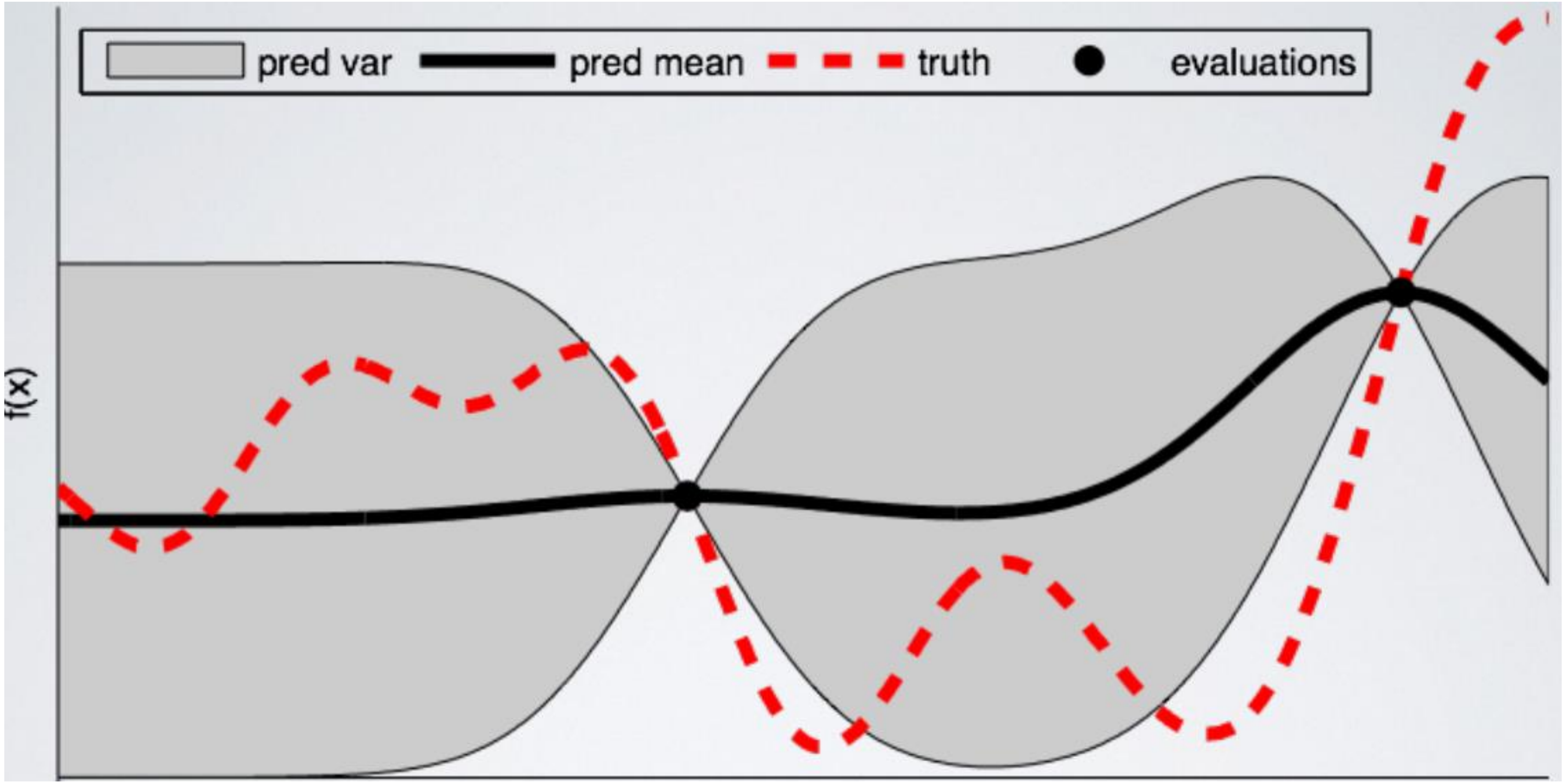
- Can sample function from prior
- Tractable posterior
- Posterior predictive

…equivalent to Bayesian linear regression in function space

[Source: C. Rassmussen]

Bayesian Optimization

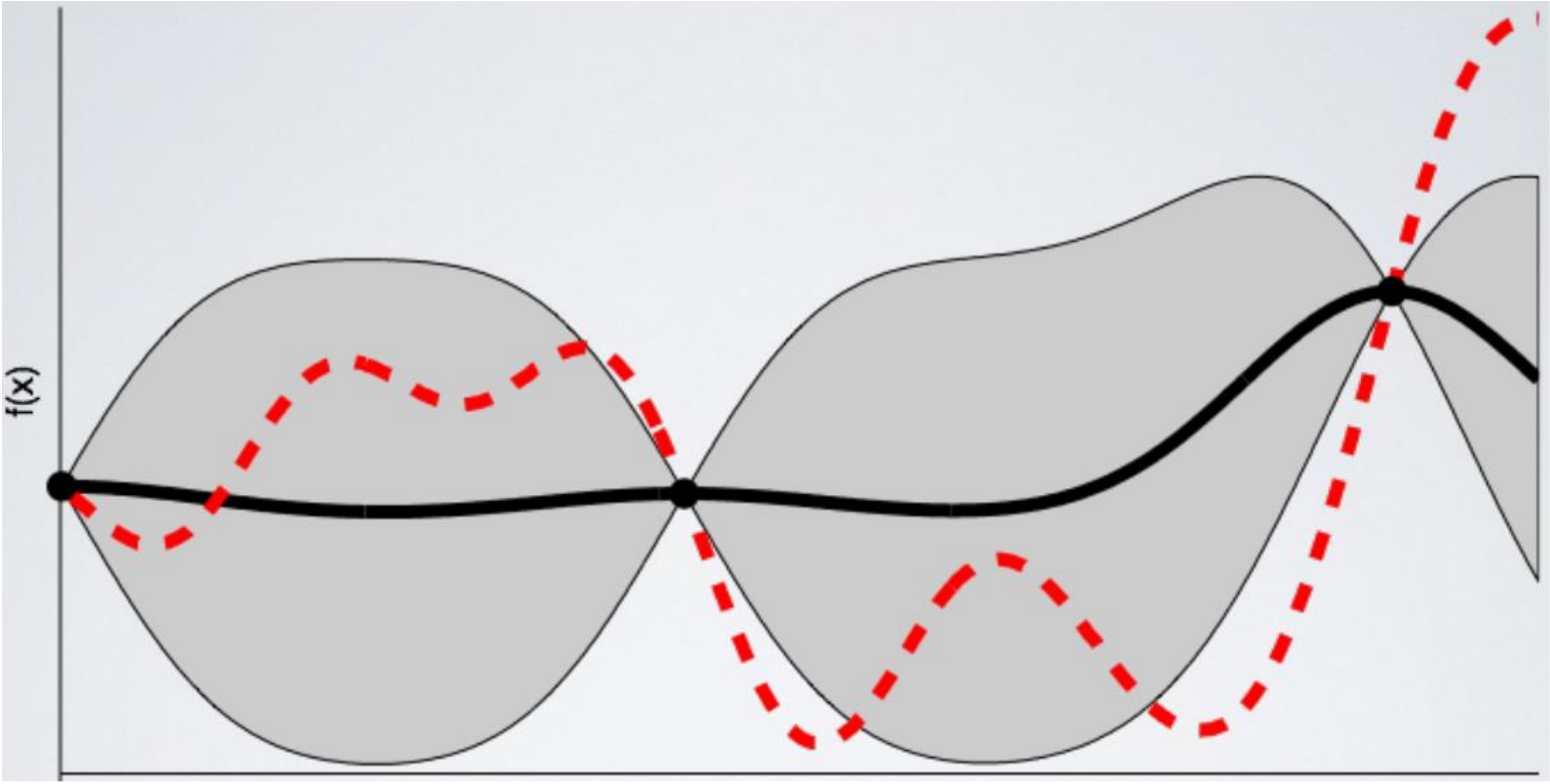
Global optimization of *random functions*: $\min_x f(x)$



[Source: Ryan Adams]

Bayesian Optimization

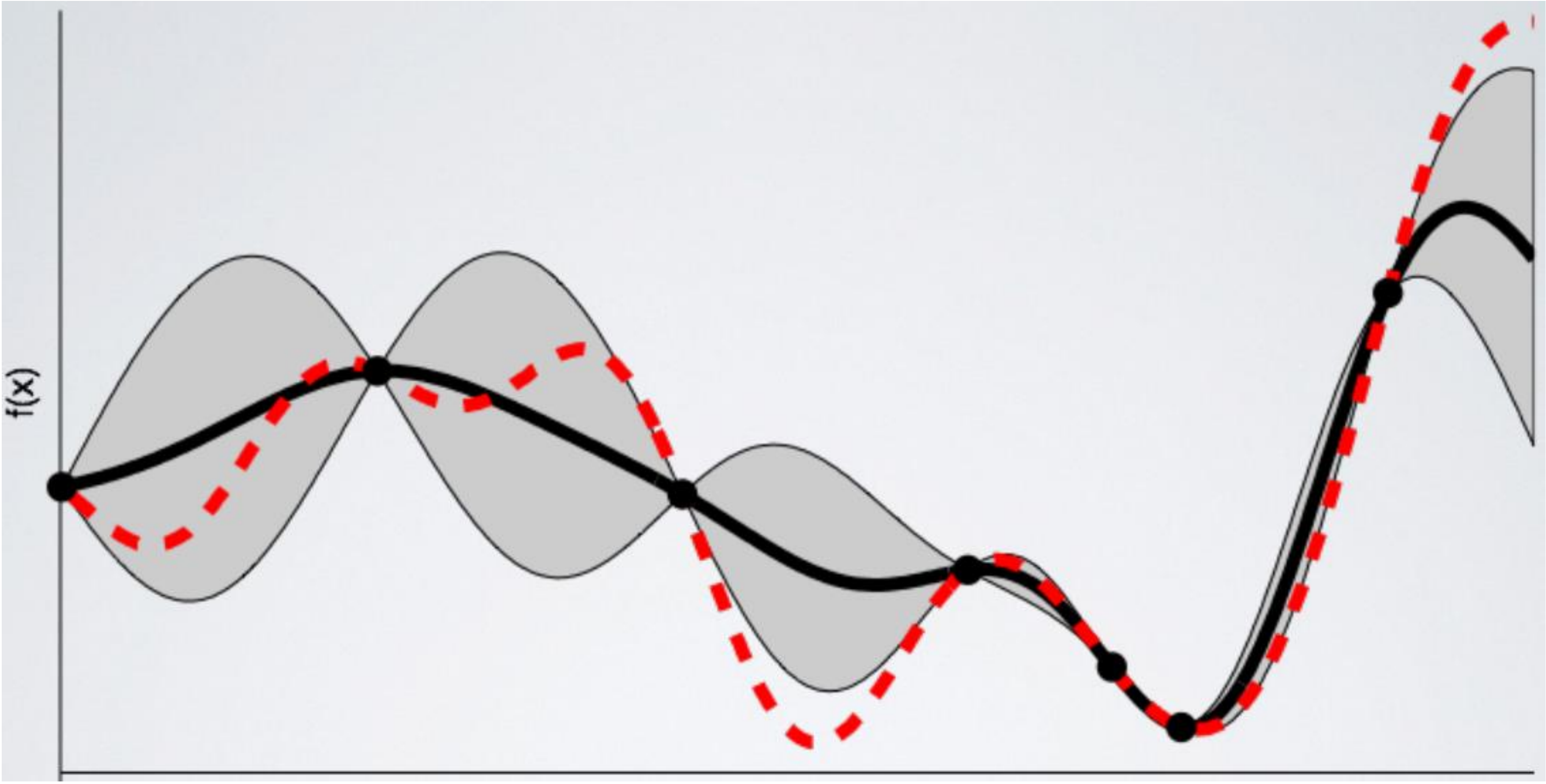
Iteratively updates distribution over function value (regression)



[Source: Ryan Adams]

Bayesian Optimization

The function is well-approximated around the minimizer



[Source: Ryan Adams]

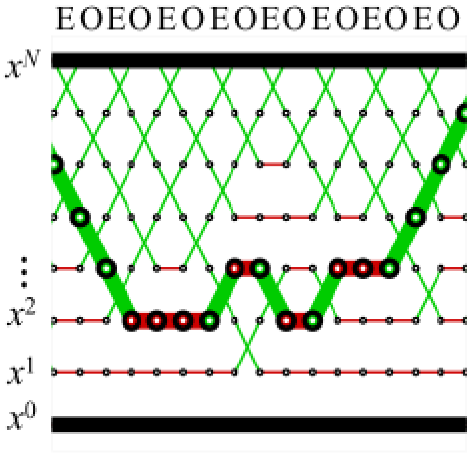
There's so much more to cover…

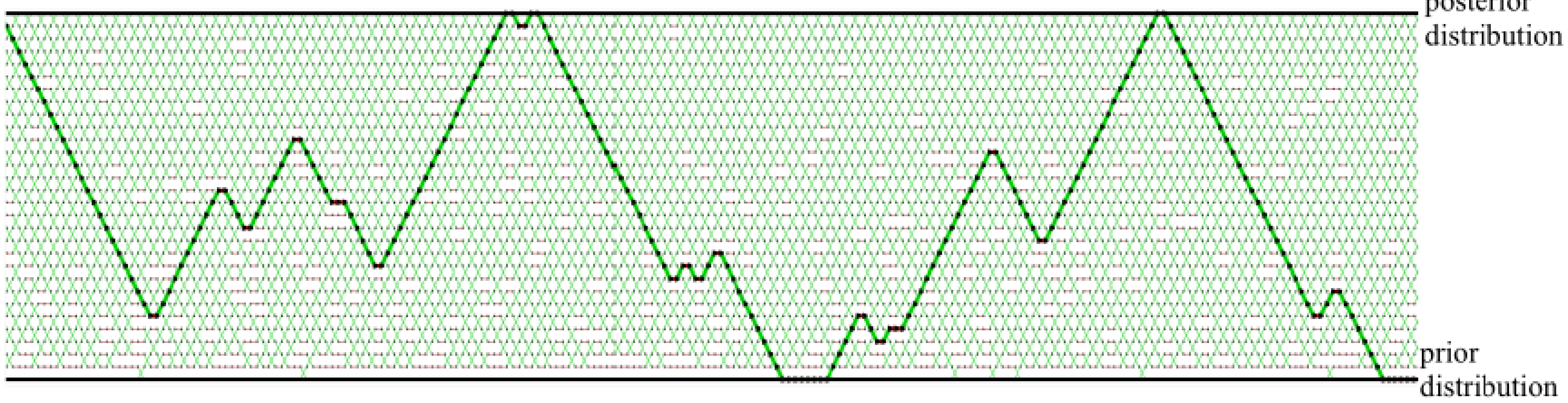
Models & Applications	AI Libraries	Representation Learning	Bayesian Nonparametrics	Advanced MCMC	Still more…
Course was mostly focused on algorithms, limited attention to modelling	Collection of tools for larger-scale AI	Unsupervised representation learning from structured data	A class of probability models where model complexity is inferred from the data	Avoiding random walk dynamics and allowing parallel computation	

Advanced Markov Chain Monte Carlo

Advanced MCMC techniques reduce sample complexity and avoid getting stuck in local energy minima

[Source: Syed et al, 2019]

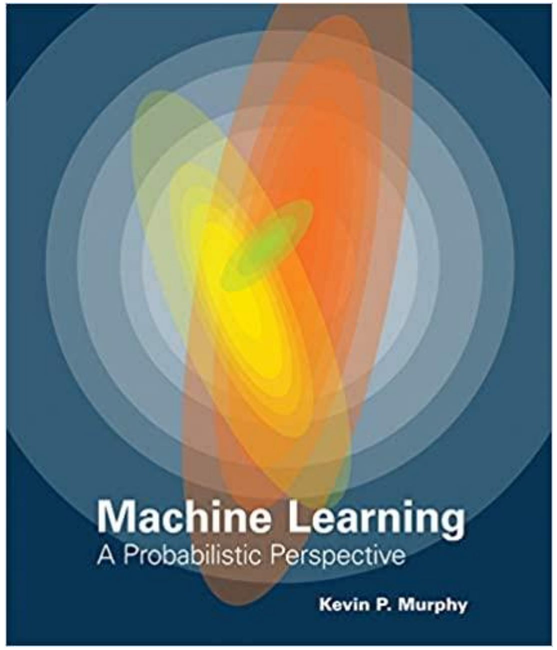


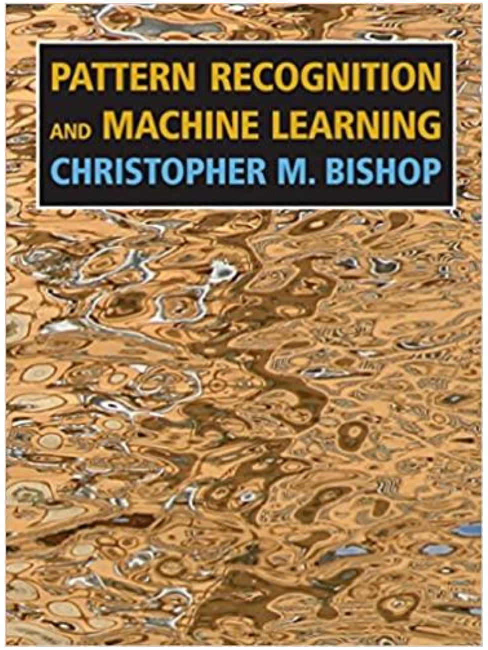


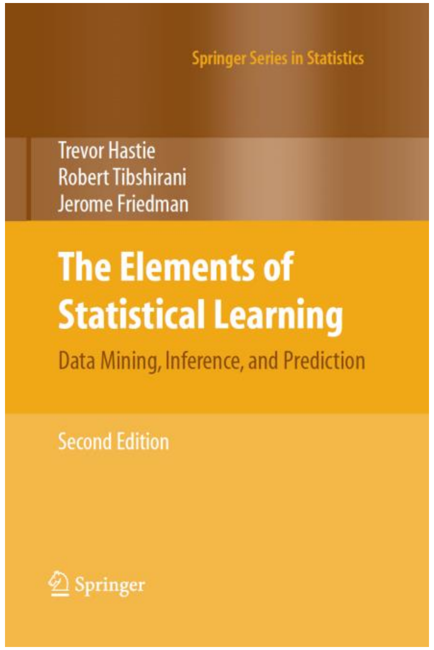
Example: Parallel tempering exchange replicates across multiple MCMC chains running in (embarrassingly) parallel

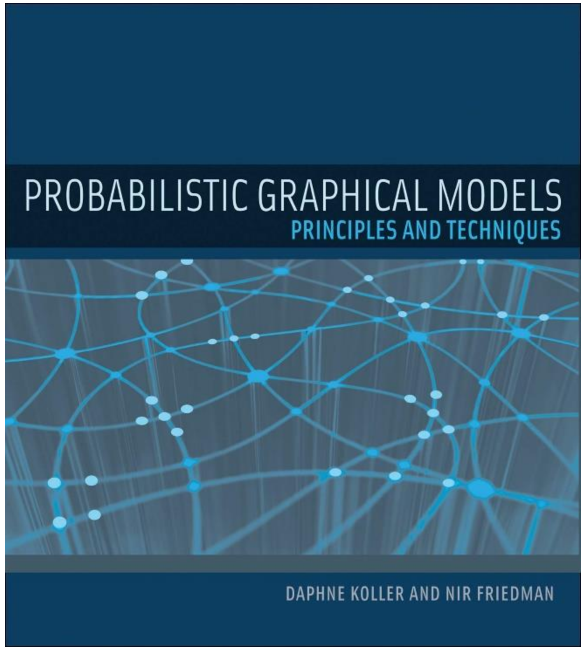
Resources

We covered a lot of ground…but there is a lot more to cover!









Important conferences to follow…

- **NeurIPS**
- **ICML**
- **AISTATS**
- **ICLR**

- ICRA
- IROS
- COLT

- IJCAI
- AAAI
- UAI