Augmentation and Amortization for Generative Models

Max Welling

University of Amsterdam

Qualcomm Technologies

Canadian Institute for Advanced Research

Qualcomm

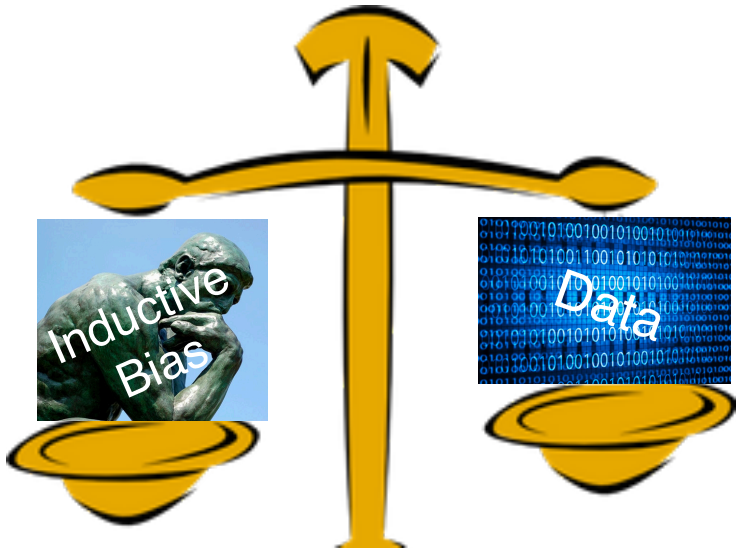
CIFAR
CANADIAN INSTITUTE FOR ADVANCED RESEARCH

Inductive Bias versus Data

Generative models
(e.g. Bayesian networks)
+ lots of inductive bias
+ out of domain generalization

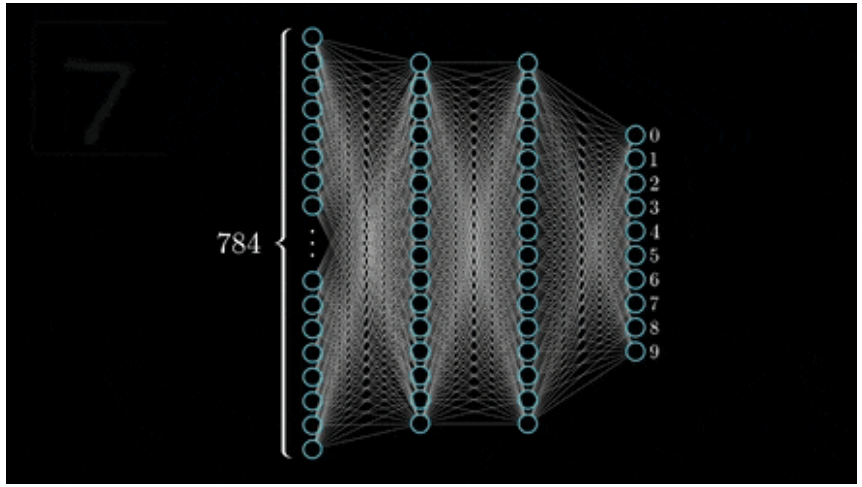
Inductive bias + data → predictions

Discriminative Models
(e.g. Deep learning)
+ lots of data
+ limited domain generalizatio



Discriminative Models

Raw data



Properties / class labels / semantic features

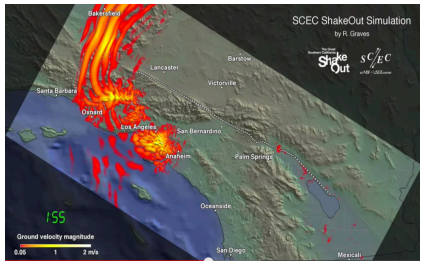
In ML we train discriminative models from raw data to class labels / system properties

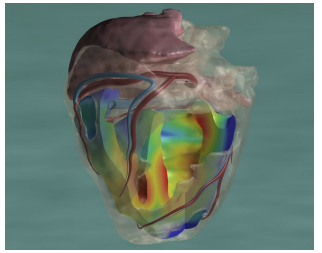
Generative / Forward Models

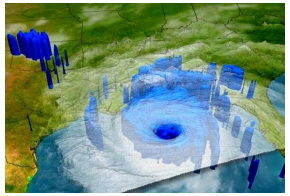
Definition: *A generative model simulates the data generation process.*

Examples:

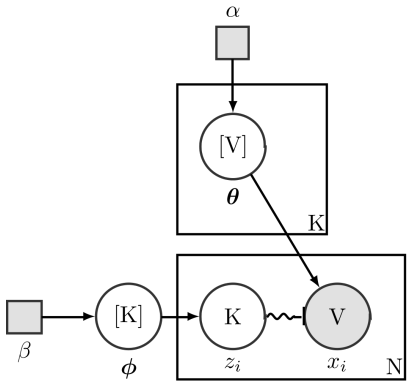
- Simulators
- Graphical models
- Probabilistic programs
- Ordinary/Partial differential equations

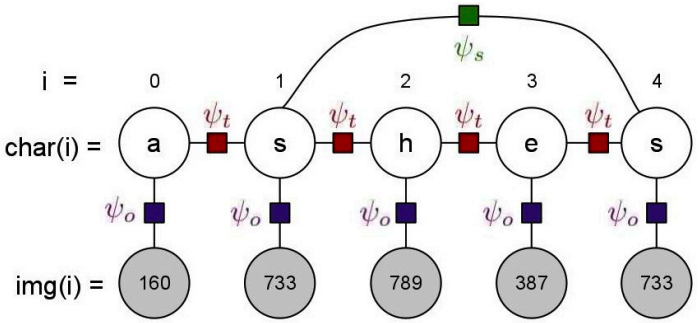


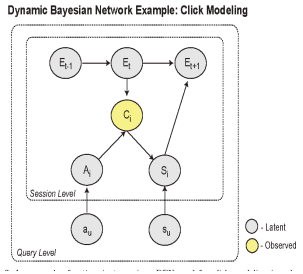




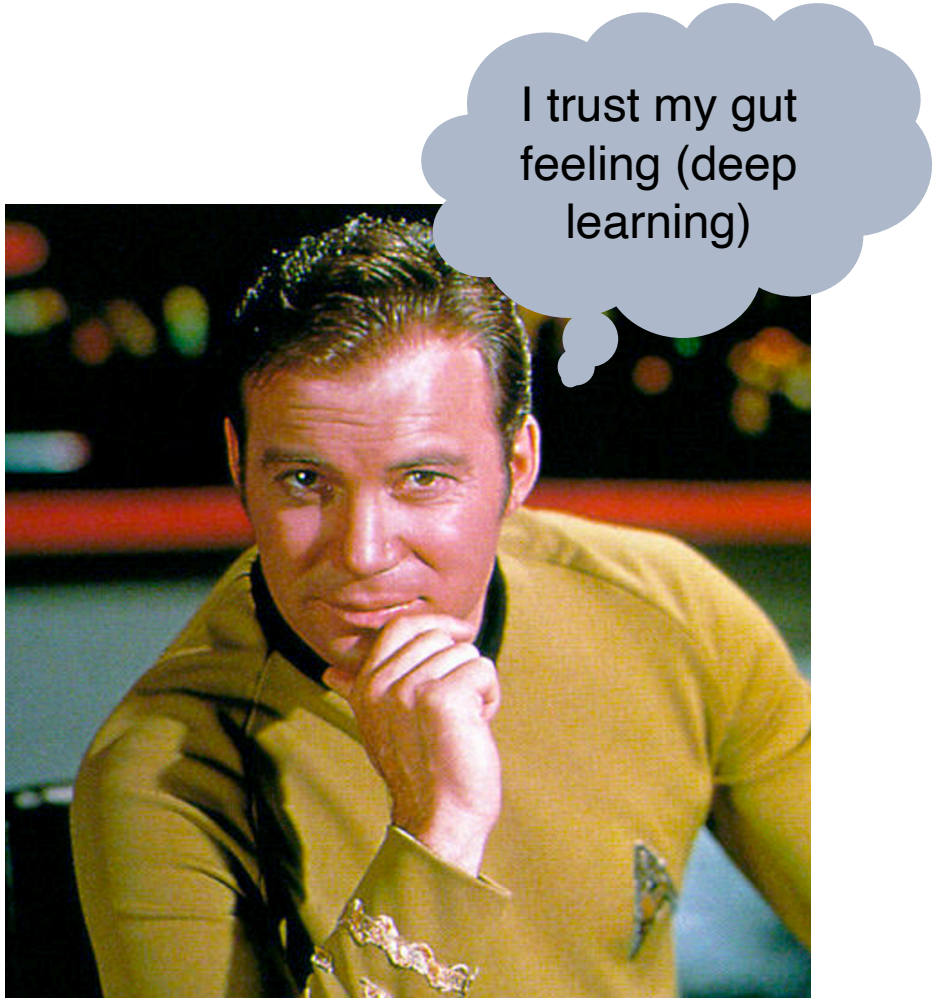
Also known as **forward models**



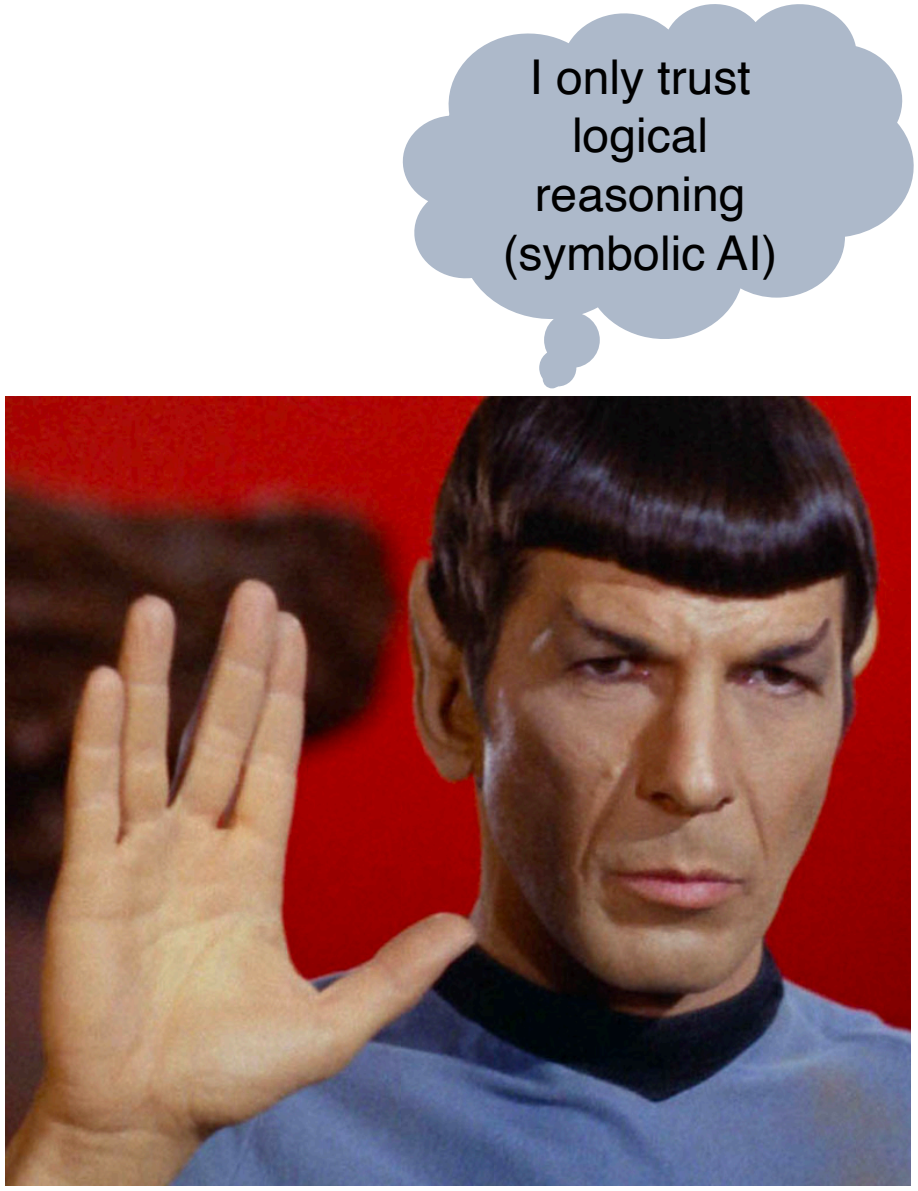




Intuition versus Logic

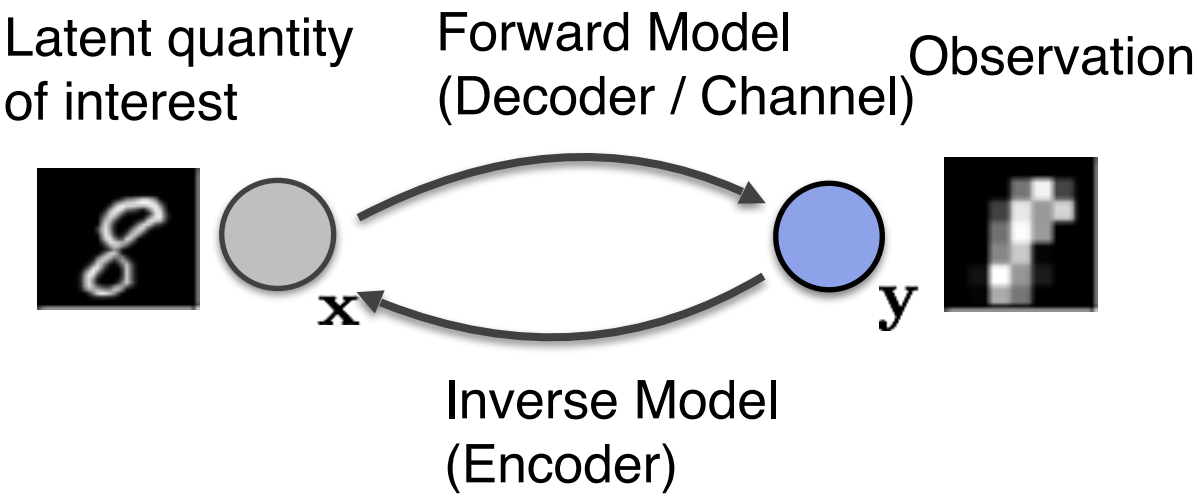


Kirk

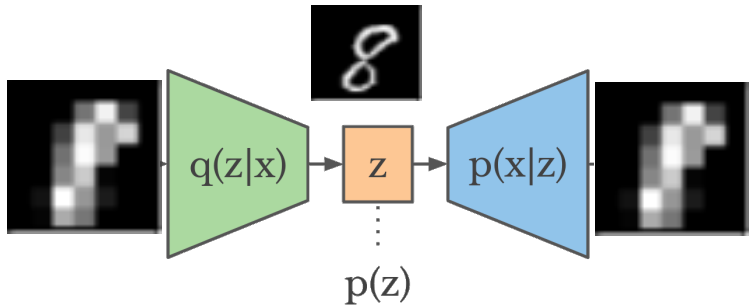


Spock

Forward Decoder & Inverse Encoder Models



VAE: Variational Autoencoder



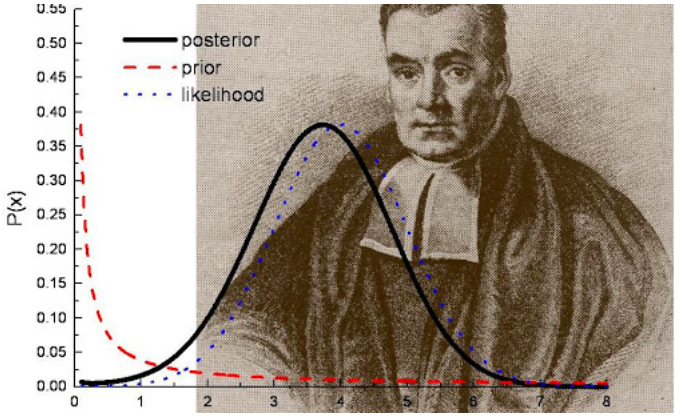
$$y = g(x) + \epsilon$$

$$\hat{x} = h(y)$$

Forward (Decoder) Model $y \sim p(y|x)$

Inverse (Encoder) Model $\hat{x} \sim q(x|y)$

How are they related?



Thomas Bayes

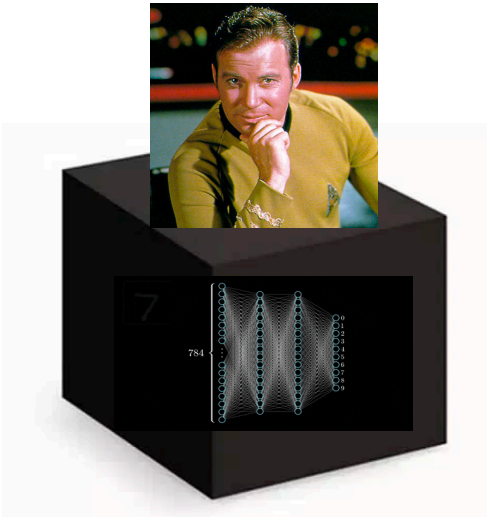
Bayes Rule:

Inverse model / Encoder

Forward model / Decoder

$$P_{discr}(\text{class}|\text{data}) = \frac{P_{gen}(\text{data}|\text{class})P_{prior}(\text{class})}{P(\text{data})}$$

Pros and Cons



White Box:

- **Data efficient**: model uses expert knowledge (e.g. laws of physics)
- **Interpretable**: variables mean something in the real world
- **Better generalization**: model uses causal structure

Black Box:

- **More flexible**: less biased to human imagination
- **Accurate**: you model your predictions directly
- **Fast** predictions: you don't need Bayes rule

We want to combine these strengths into one modeling paradigm

Amortization & Augmentation

Amortization

"Finding or generating related problems to learn a model or policy to do X"

- *Learning to Infer (VAE encoder)*: Instead of performing inference per instance (e.g. MF) learn a model to predict the outcome of inference.
- *Learning to optimize*: Instead of running a classical optimization tool (e.g. GA) learn a policy on similar optimization problems to build a better optimizer.
- *Learning to Learn (Meta-Learning):* Instead of learning a model using e.g. gradient descend, train an optimizer on related learning problems.

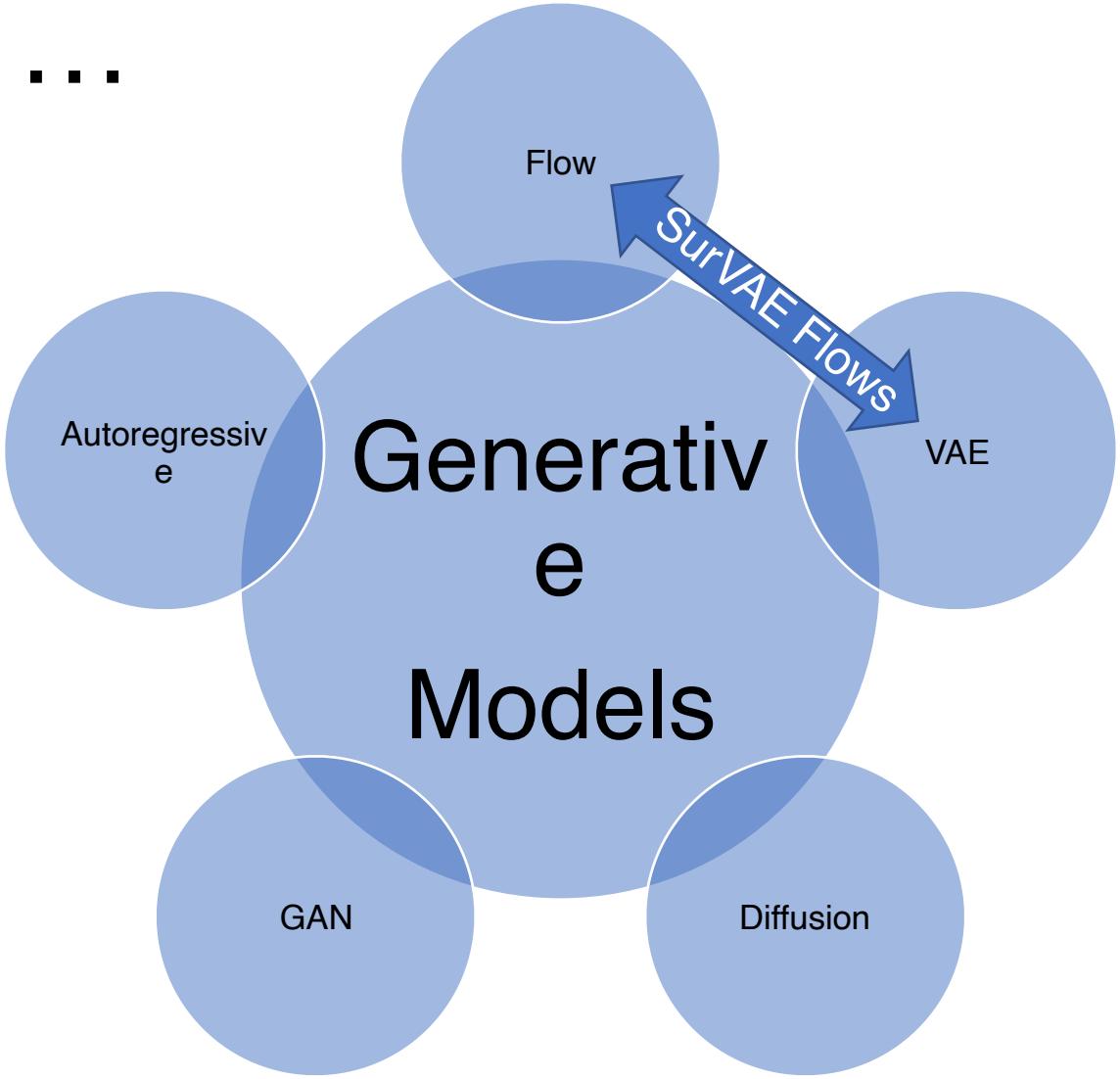
Augmentation

"A Hybrid system that combines classical, hand-engineered methods with learned ML-based models or policies"

- Probabilistic programs / simulators + neural networl
- Data forcing terms in encoders.
- Hybrid inference that combines classical inference in graphical models (e.g. BP, MF) with neural netwo (e.g. graph NN)s

Flows, GANs, VAEs, …

The state of world in terms of generative modeling



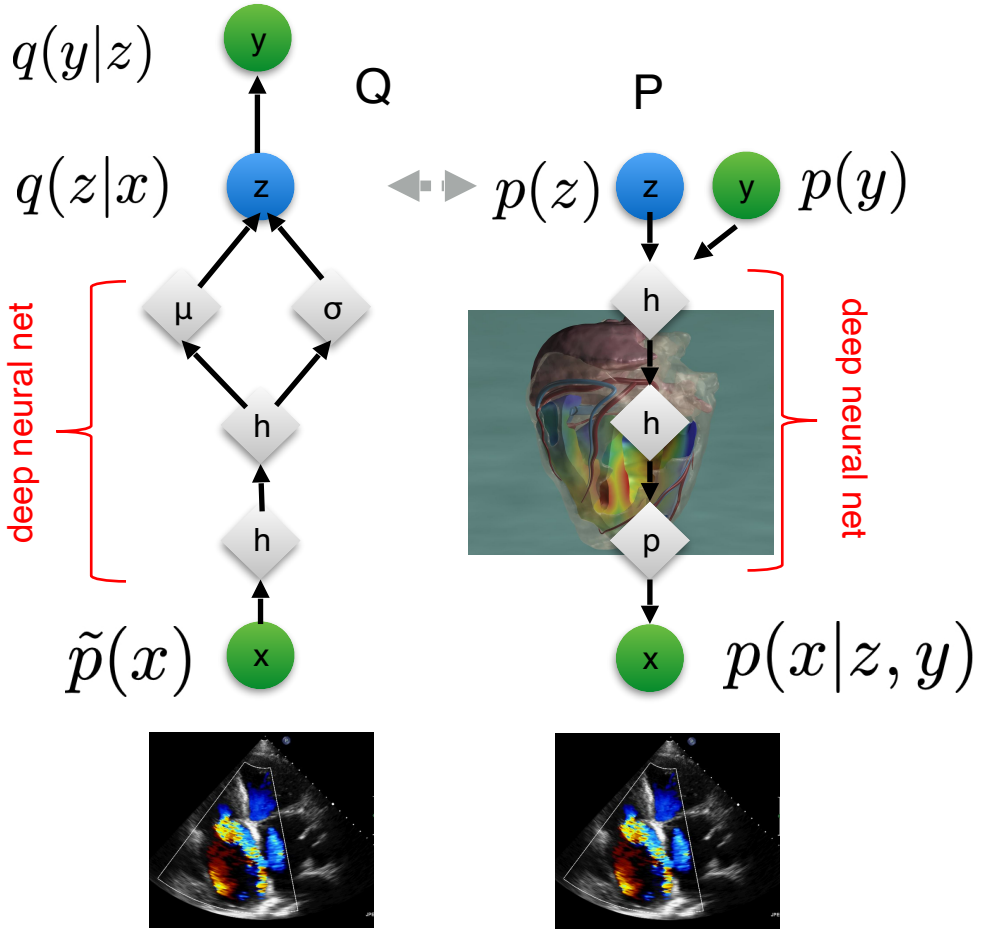
The Variational Auto-Encoder as Approximate Bayes Rule

(Kingma & W. 2014, Rezende et al, 2014)

deterministic NN node

unobserved stochastic node

observed stochastic node



Amortized encoder Augmented Decoder

VAE objective-function = ELBO:

$$\min_{\phi,\theta} KL[Q_\phi(Z|X)\hat{P}(X)||P(Z)P_\theta(X|Z)]$$

$$\min_{\phi,\theta} KL[\hat{P}(X)||P(X)] + E_{\hat{P}(X)}KL[Q(Z|X)||P(Z|X)]$$

$$\Rightarrow P(X) = \hat{P}(X) \quad \& \quad Q_\phi(Z|X) = \frac{P(Z)P_\theta(X|Z)}{P(X)}$$

Normalizing Flows Rezende & Mohamed 2015

- Deterministic (invertible) reparameterization of P(X):

$$\log P(X) = \log P(Z) + \log \left| \det \left(\frac{dZ(X)}{dX} \right) \right|$$

- Sequence of invertible transformations:

$$\log P(X = Z_T) = \log P(Z_0) + \sum_{t=1}^{T} \log \left| \det \left(\frac{dZ_t}{dZ_{t-1}} \right) \right|$$

Composable Transformations

- Normalizing flow: deterministic invertible transformation f:X→Z,

$$\log P(X) = \log P(Z) + \log \left| \det \left(\frac{dZ(X)}{dX} \right) \right|$$

- ELBO: Bound on stochastic transformation Q(Z|X),

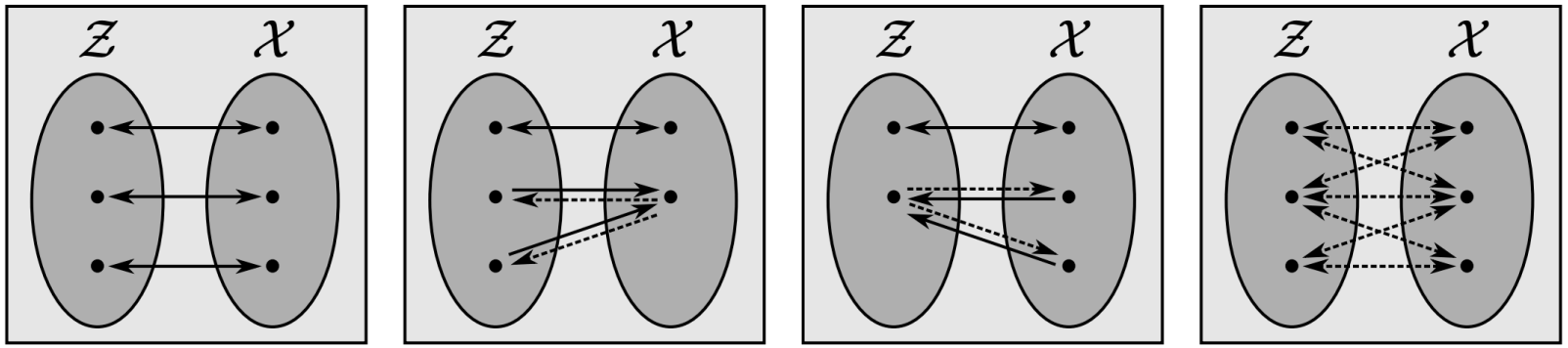
$$\log P(X) \geq E_Q \left[\log P(Z) \right] + E_Q \left[\log \left(\frac{P(X|Z)}{Q(Z|X)} \right) \right]$$

Algorithm 1: $\log - \text{likelihood}(\boldsymbol{x})$

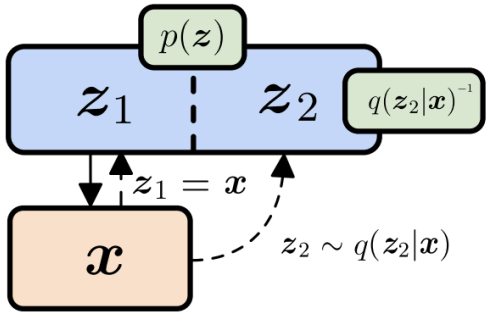
Data: $\boldsymbol{x}$, $p(\boldsymbol{z})$ & $\{f_t\}_{t=1}^{T}$
Result: $\mathcal{L}(\boldsymbol{x})$
for t *in* $\text{range}(T)$, **do**
  if f_t *is bijective* **then**
    $\boldsymbol{z} = f_t^{-1}(\boldsymbol{x})$;
    $\mathcal{V}_t = \log \left| \det \frac{\partial \boldsymbol{z}}{\partial \boldsymbol{x}} \right|$;
  else if f_t *is stochastic* **then**
    $\boldsymbol{z} \sim q_t(\boldsymbol{z}|\boldsymbol{x})$;
    $\mathcal{V}_t = \log \frac{p_t(\boldsymbol{x}|\boldsymbol{z})}{q_t(\boldsymbol{z}|\boldsymbol{x})}$;
  $\boldsymbol{x} = \boldsymbol{z}$;
end
return $\log p(\boldsymbol{z}) + \sum_{t=1}^{T} \mathcal{V}_t$

Surjections & SurVAE Flows

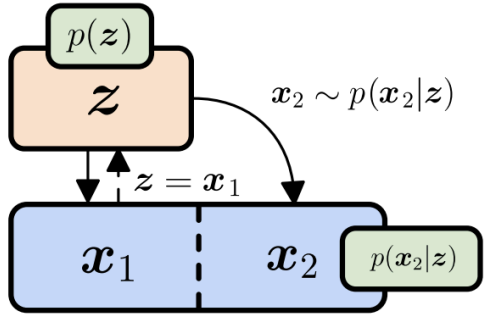
(Didrik Nielsen et al 2020)



(a) Bijective (b) Surjective (Gen.) (c) Surjective (Inf.) (d) Stochastic



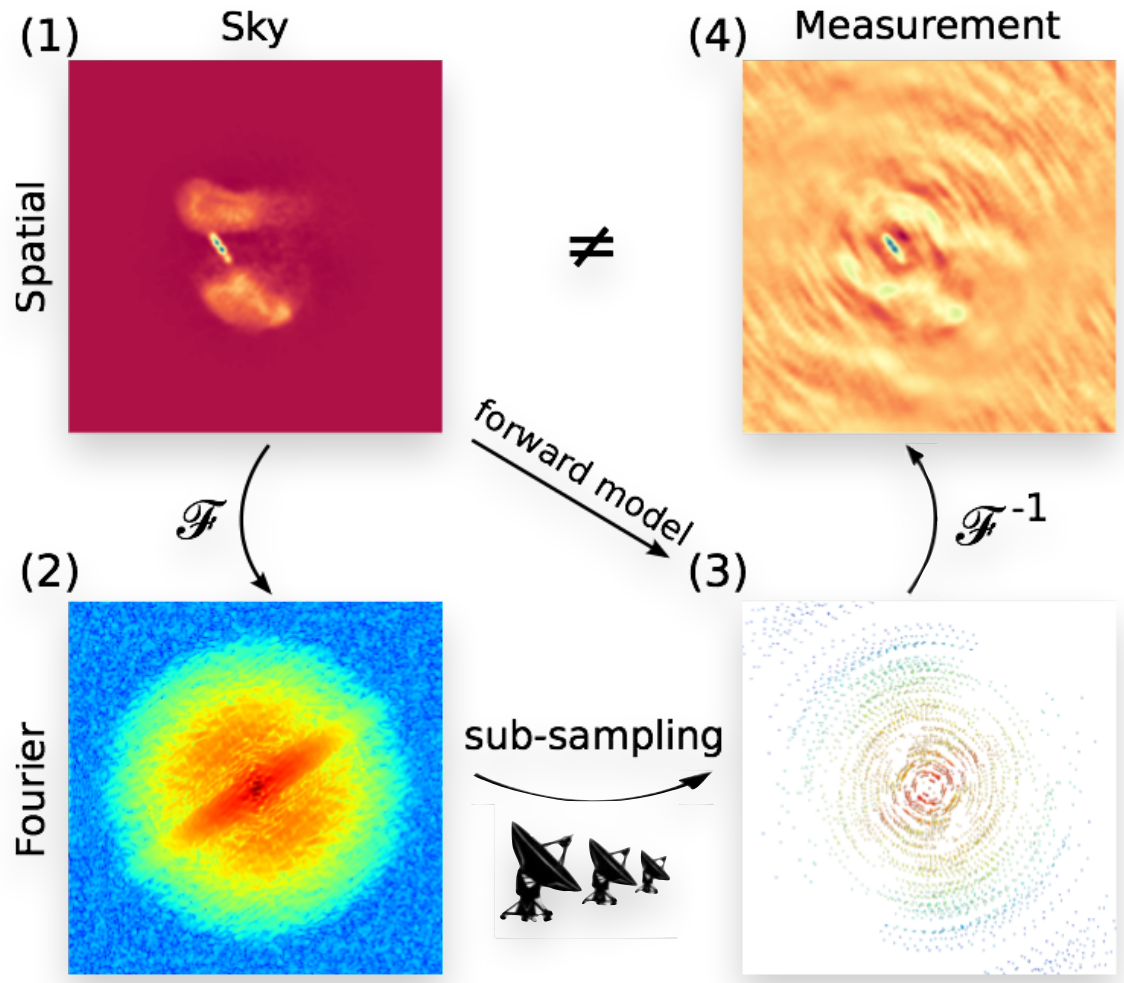
(a) Gen. slicing



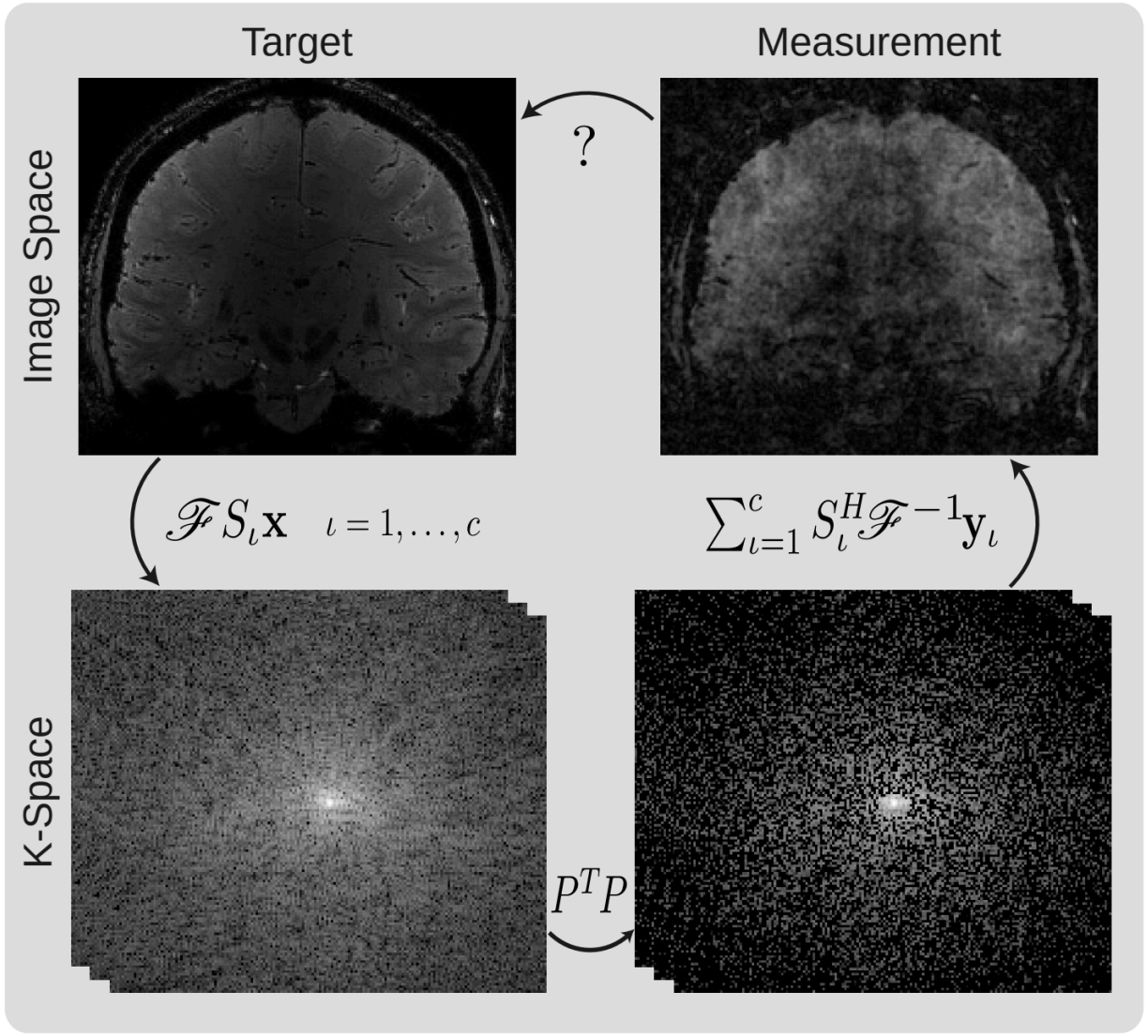
(b) Inf. slicing

ELLIS exchange

Inverse Problems: Radio Astronomy



Inverse Problems: MRI Reconstruction

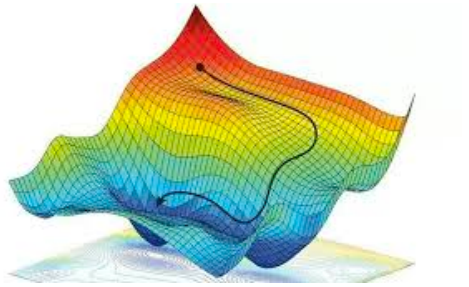


Classical Solutions

We will add neural components to a classical iterative optimization scheme

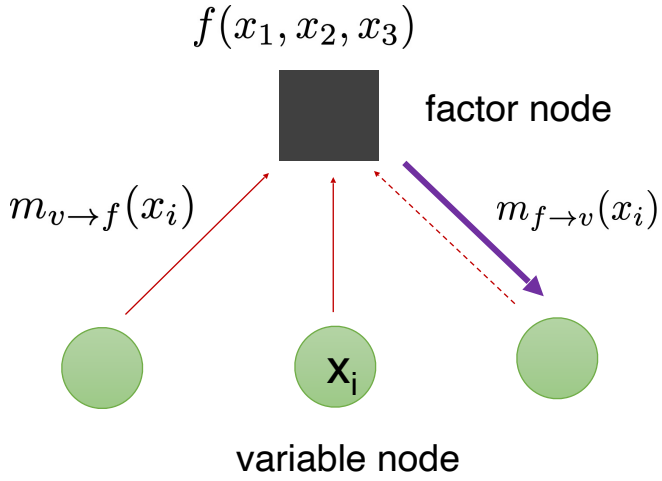
- Gradient ascent on objective $L(x) = \underbrace{\log P(y|x)}_{\text{observation model}} + \underbrace{\log P(x)}_{\text{learned prior}}$

$$x_{t+1} = x_t + \nabla_x L(x)|_{x=x_t}$$



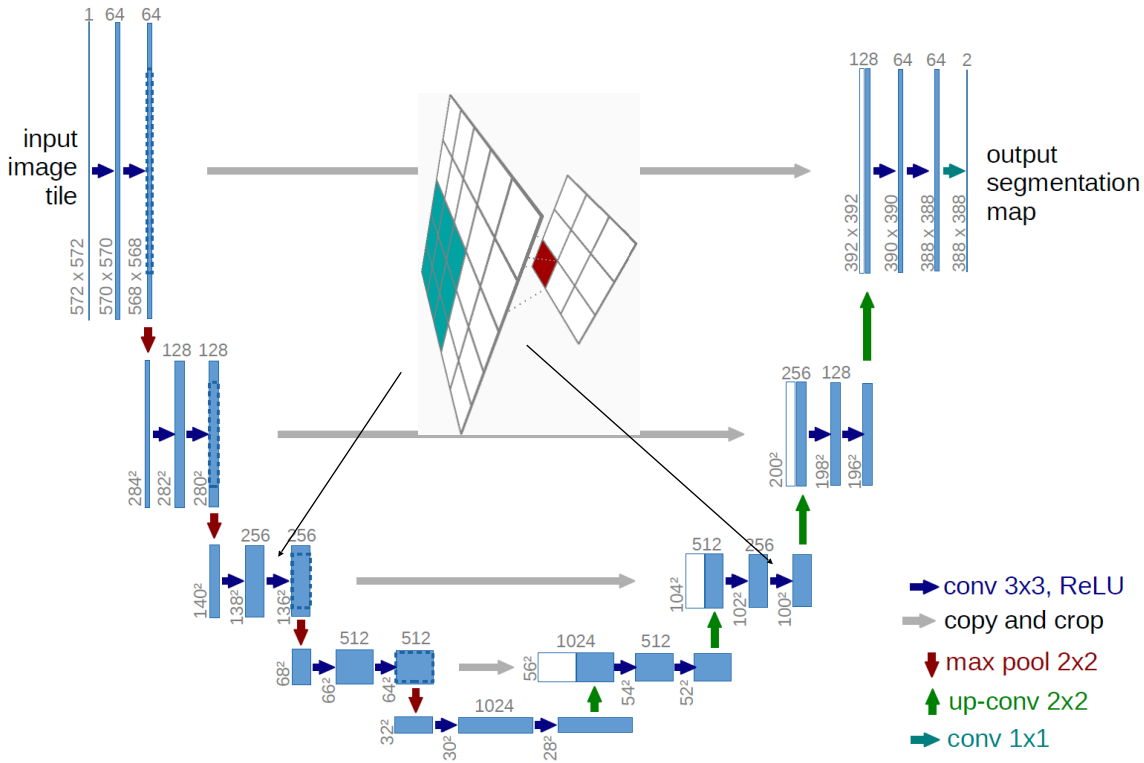
- Belief propagation or other inference algorithm

$$\mu_{f_s \to x_n}(x_n) = \sum_{\mathbf{x}_s \setminus x_n} f_s(\mathbf{x}_s) \prod_{m \in \mathcal{N}(f_s) \setminus n} \mu_{x_m \to f_s}(x_m)$$



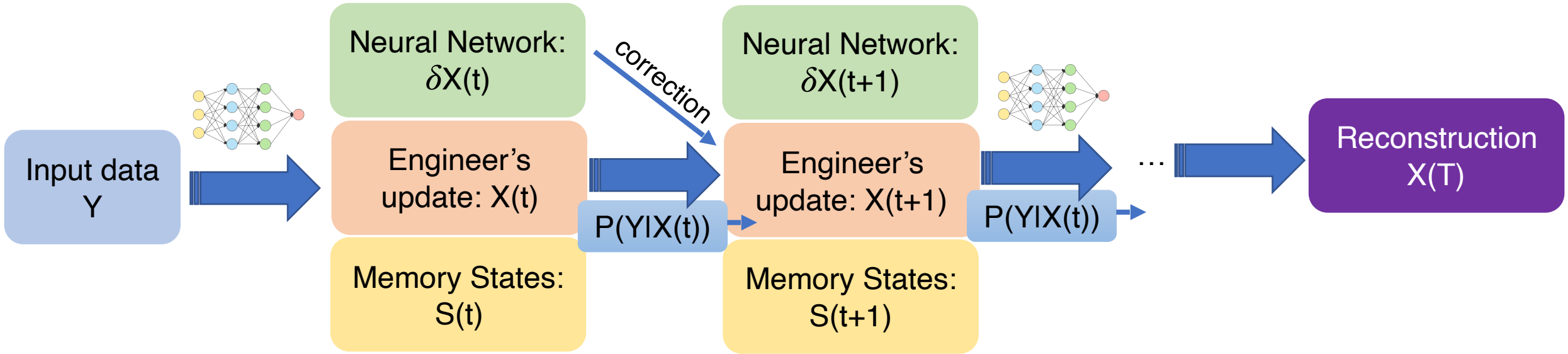
Deep Learning (Full Neural) Solution

- Obtain data pairs: $\{x_n, y_n\},\ n = 1..N$
- Train big neural net $X = f_\theta(Y)$



Amortized, Augmented Inverse (Encoder) Model

- Unroll classical iterative optimization scheme and interpret as RNN
- Add memory states S
- Train deep neural network to *correct* iterative engineering solution (neural augmentation)
- Use forward model P(Y|X) to check if current estimate agrees with data

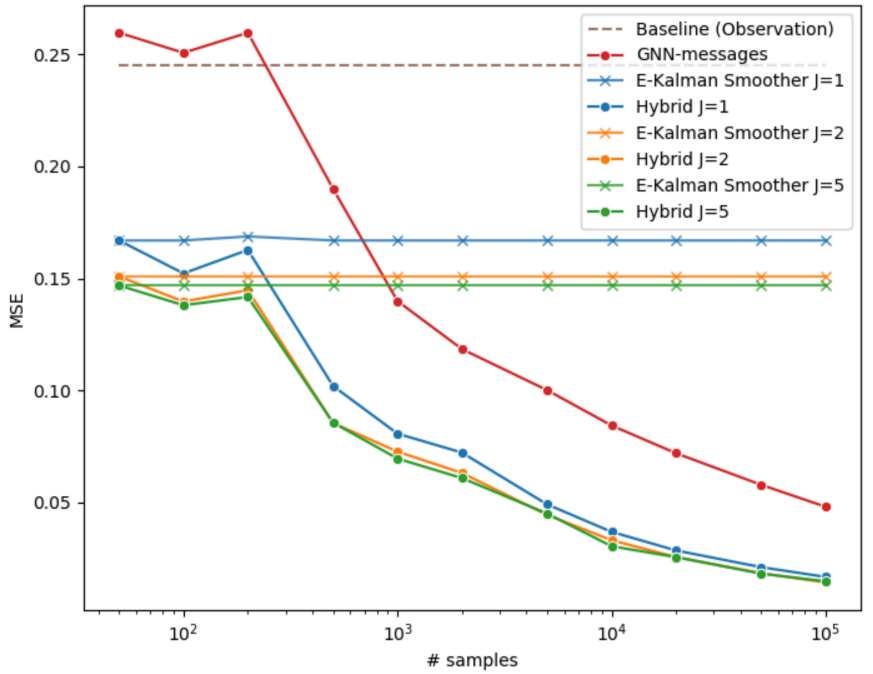


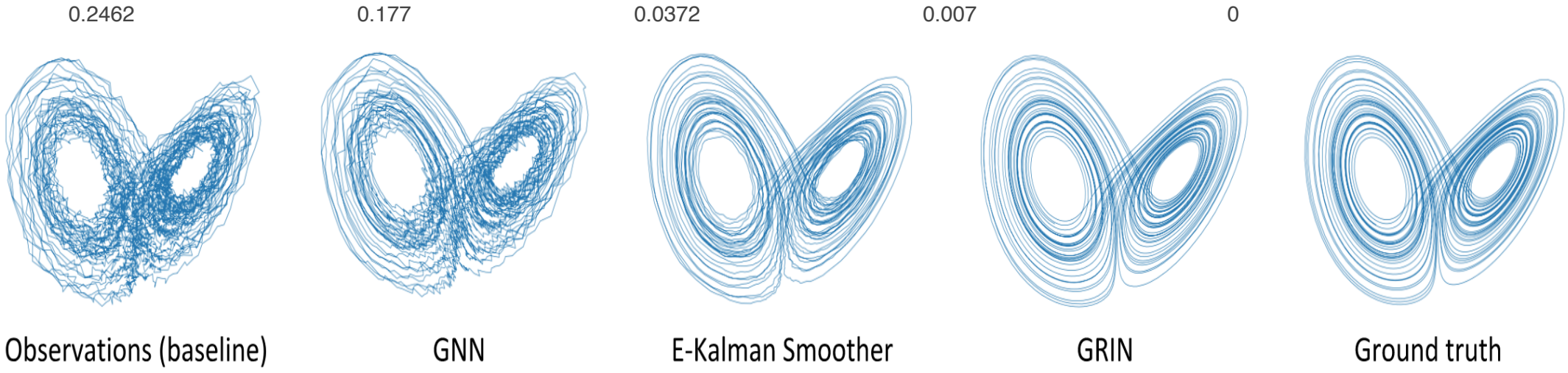
Nonlinear Kalman Filter

(Garcia et al 2019)

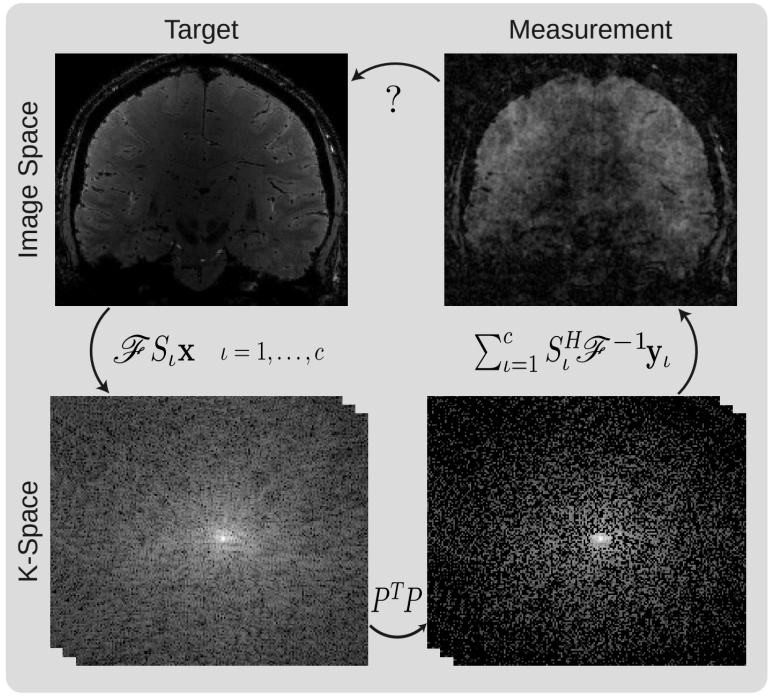
- Lorenz attractor dynamics (chaotic)

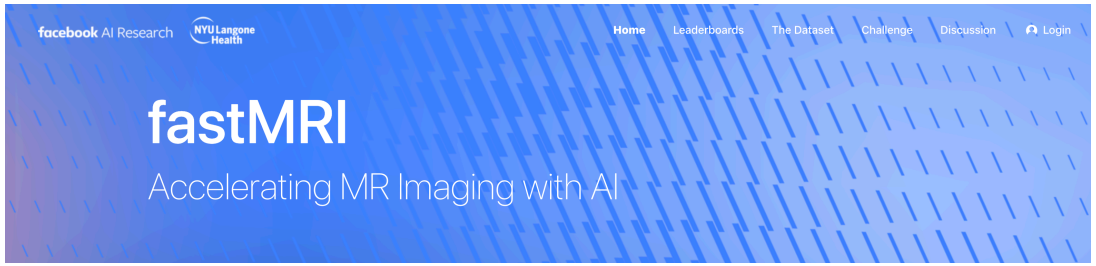
$$\frac{\partial z_1}{\partial t} = 10(z_2 - z_1)$$
$$\frac{\partial z_2}{\partial t} = z_1(28 - z_3) - z_2$$
$$\frac{\partial z_3}{\partial t} = z_1 z_2 - \frac{8}{3} z_3$$



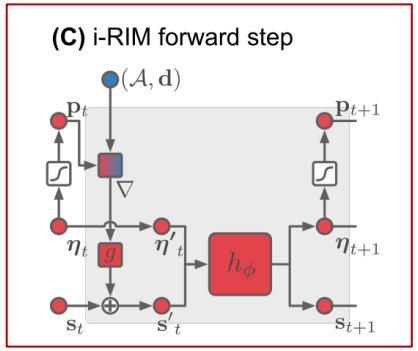


Example Problems: MRI Reconstruction

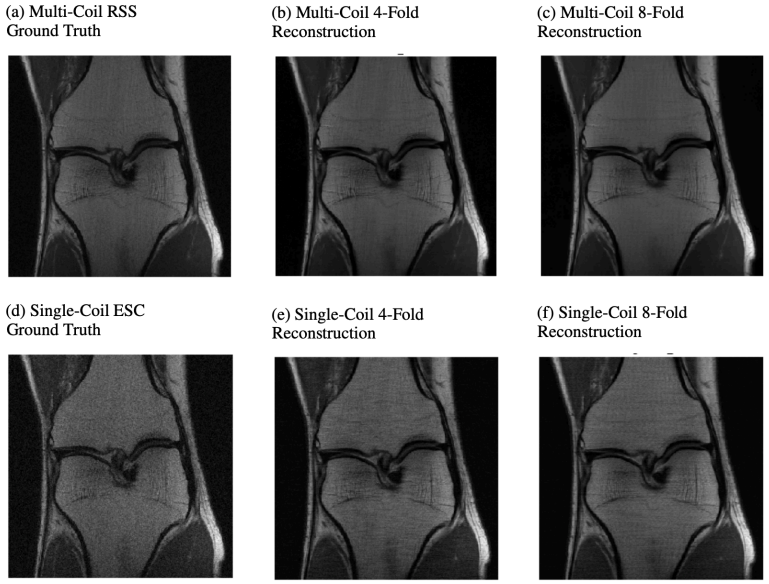




Patrick Putzky
(U. Amsterdam)



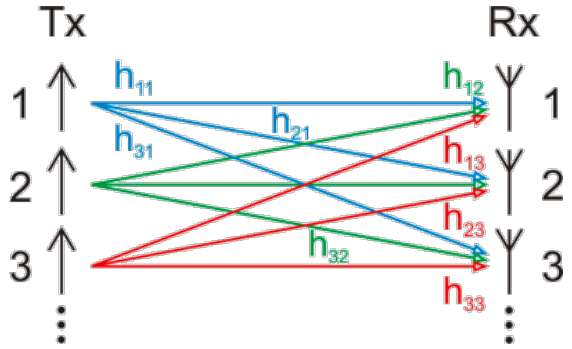
Our neural augmentation implementation (the invertible Recurrent Inference Machine) won the single coil track.



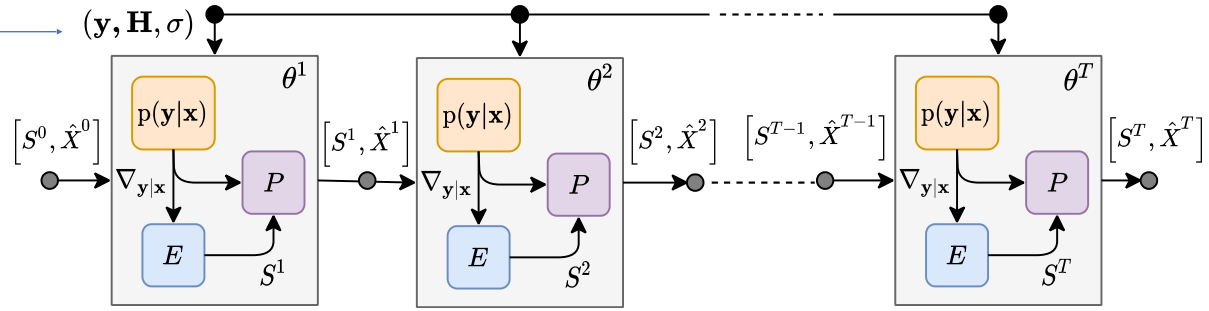
MIMO Detection

Pratik Kumar
(UvA/Qualcomm)

Bhaskar Rao
(UCSD)



MIMO forward model

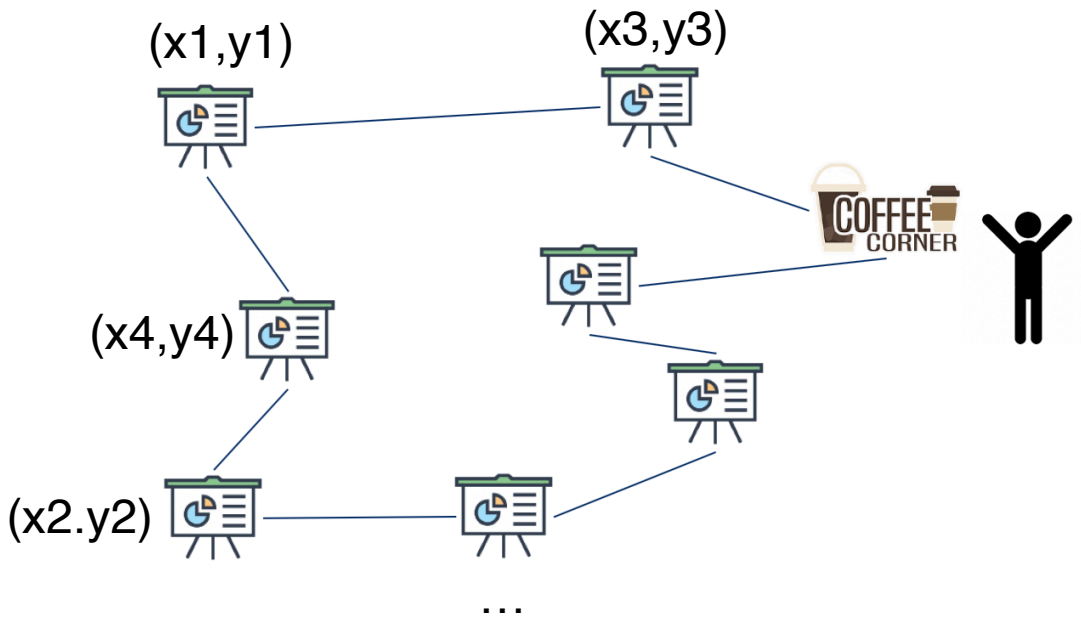


RE-MIMO inverse (inference) model

Amortized Optimization: Traveling Scientist Problem

(Kool et al , ICLR 2019)

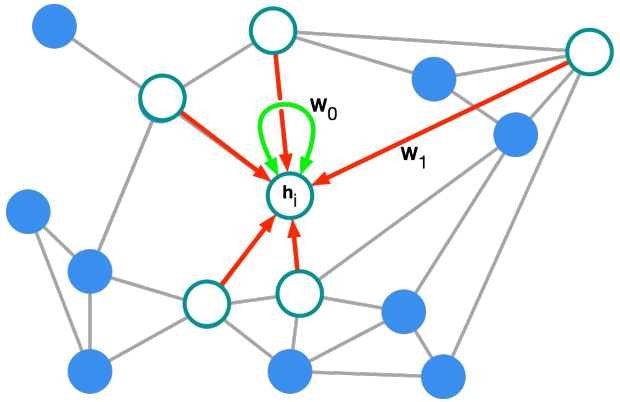
Wouter Kool



Learn Policy $\pi(a|s) \Rightarrow P(\text{next node is } i|\text{previous nodes})$

by generating lots of example trajectories

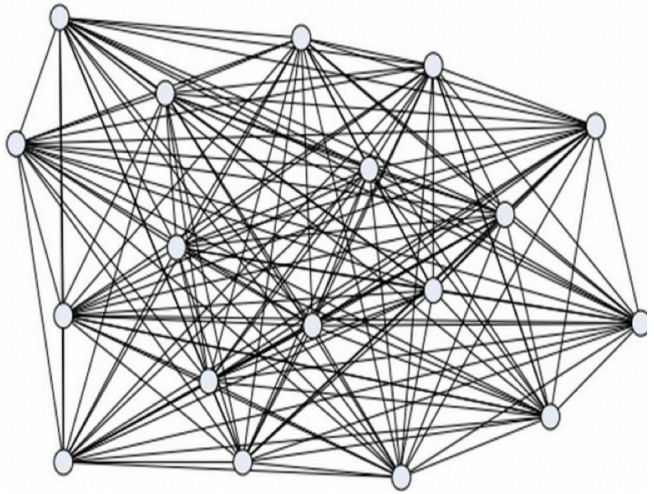
Graph NN



Input as (fully connected) graph

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\mathbf{h}_i^{(l)}\mathbf{W}_0^{(l)} + \sum_{j\in\mathcal{N}_i}\frac{1}{c_{ij}}\mathbf{h}_j^{(l)}\mathbf{W}_1^{(l)}\right)$$

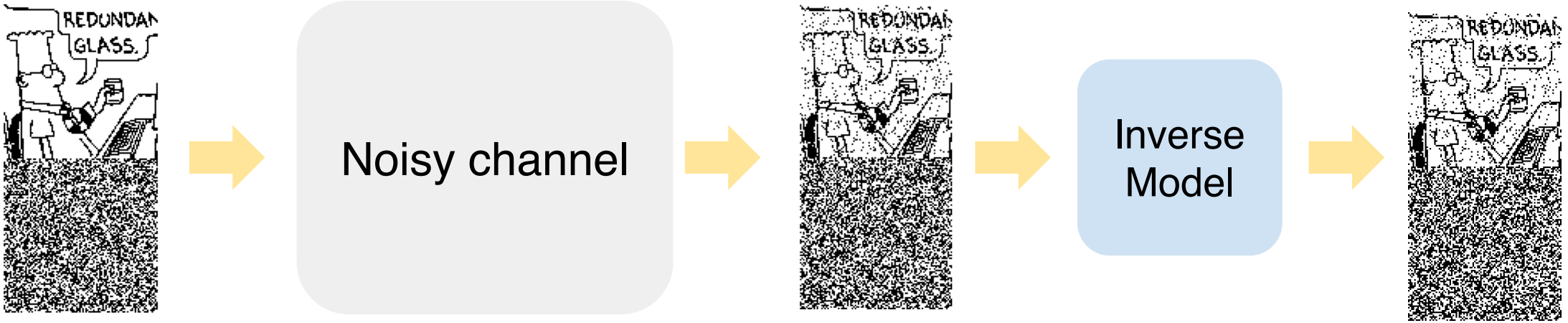
=



Error Correction Decoding

Victor Garcia
(Bosh-UvA Delta Lab

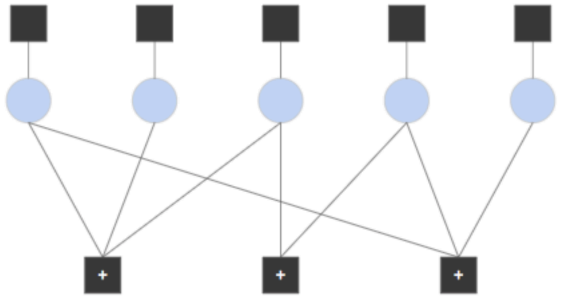
Low Density Parity Check Codes (LDPC)



(Image source: David MacKay)

Neural Augmented Belief Propagation

Factor Graph:



Roll BP out as an RNN

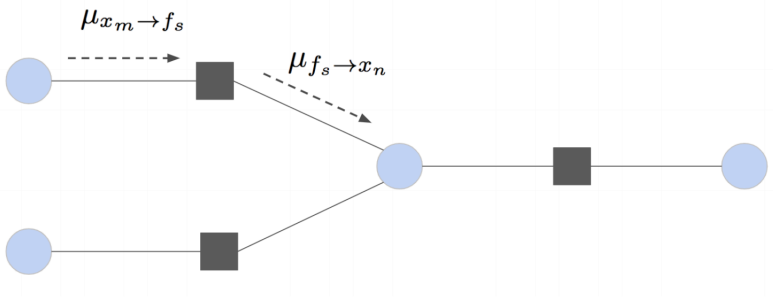


Figure 1. Belief Propagation on a Factor Graph

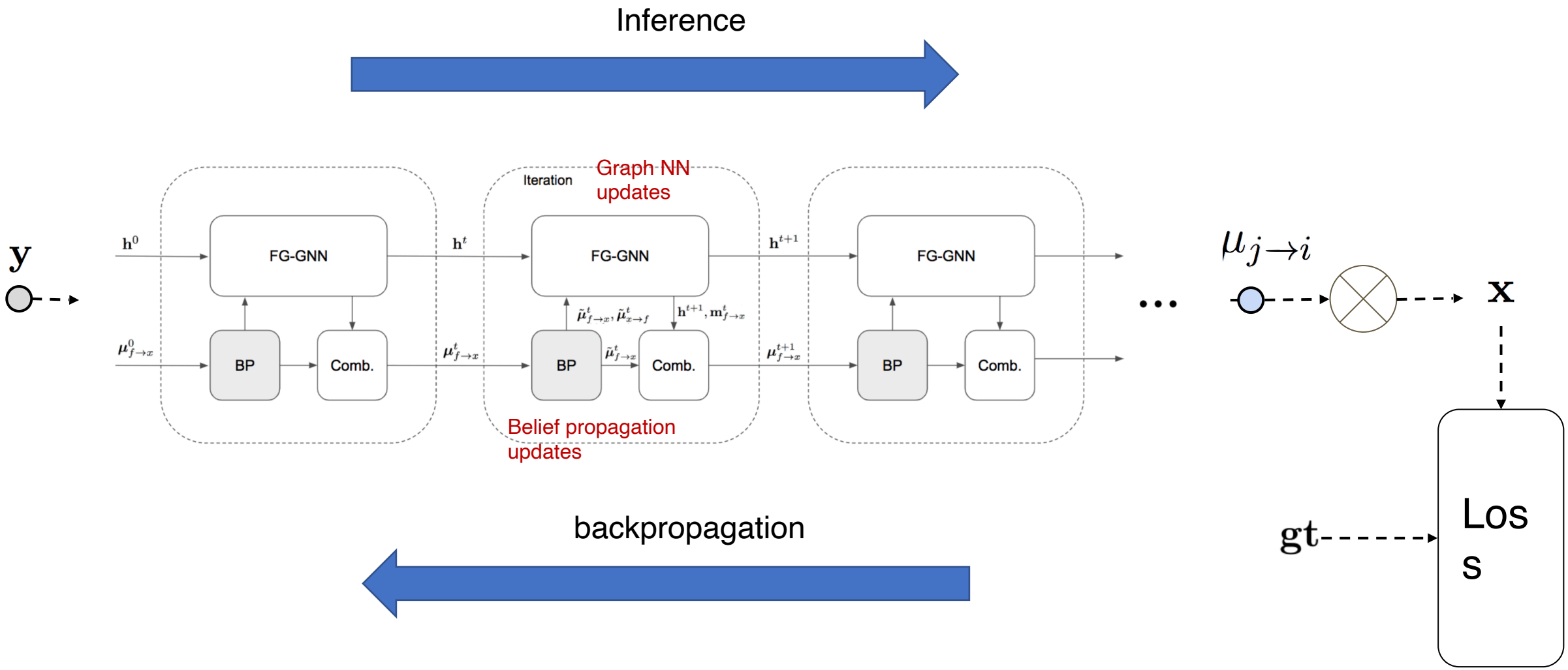
$$\mu_{x_m \to f_s}(x_m) = \prod_{l \in \mathcal{N}(x_m) \setminus f_s} \mu_{f_l \to x_m}(x_m)$$

$$\mu_{f_s \to x_n}(x_n) = \sum_{\mathbf{x}_s \setminus x_n} f_s(\mathbf{x}_s) \prod_{m \in \mathcal{N}(f_s) \setminus n} \mu_{x_m \to f_s}(x_m)$$

$$p(x_n) = \prod_{s \in \mathcal{N}(x_n)} \mu_{f_s \to x_n}(x_n)$$

Graphical Recurrent Inference Network

(Garcia & W. 2020)

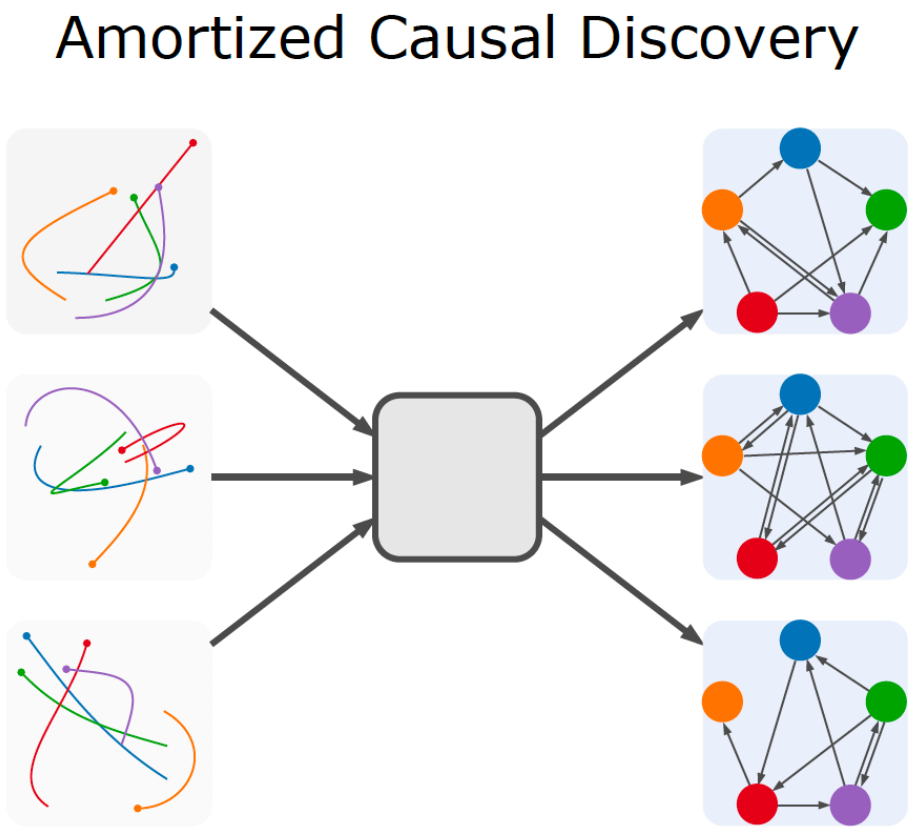


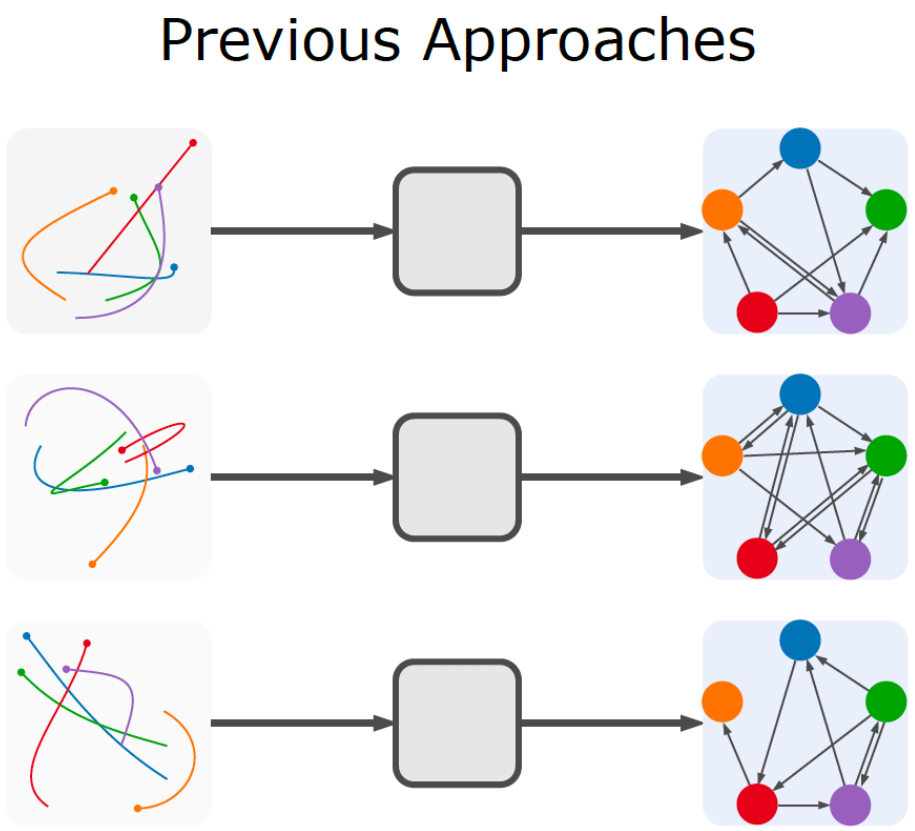
Amortized Causal Discover


Sindy Löwe


David Madras

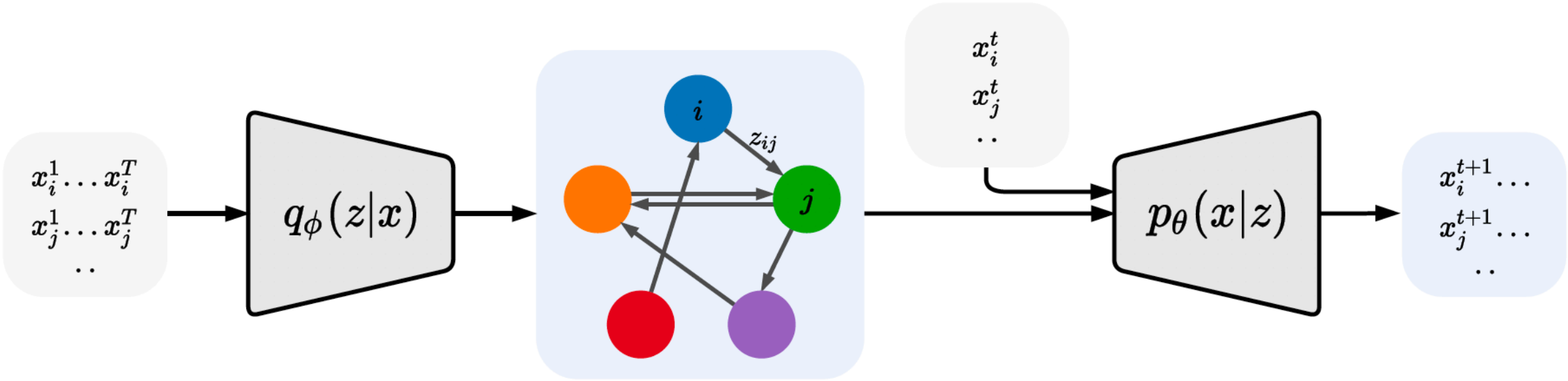

Rich Zemel





Amortized Causal Discovery

ELBO: $\mathcal{L} = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - \mathrm{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})]$



Encoder: $q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \prod q_\phi(\boldsymbol{z}_{ij}|\boldsymbol{x})$

$$z_{ij,e} = GNN(\{x_i^t\}, i = 1..N, t = 1..T)$$

Decoder: $p_\theta(\boldsymbol{x}|\boldsymbol{z}) = \prod_{t=1}^{T} p_\theta(\boldsymbol{x}^{t+1}|\boldsymbol{x}^{\leq t}, \boldsymbol{z})$

$$h_{ij}^t = \begin{cases} 0 & \text{if } z_{ij,0} = 1 \\ \sum_e z_{ij,e} f_e([x_i^t, x_j^t]) & \text{else} \end{cases}.$$

$$\boldsymbol{\mu}_j^{t+1} = \boldsymbol{x}_j^t + f_v\left(\left[\sum_{i \neq j} h_{ij}^t, \boldsymbol{x}_j^t\right]\right)$$

$$p_\theta(\boldsymbol{x}_j^{t+1}|\boldsymbol{x}^t, \boldsymbol{z}) = \mathcal{N}(\boldsymbol{\mu}_j^{t+1}, \sigma^2 \mathbb{I})$$

Conclusions

- Humans seem to continuously predict the future.
- We are surprised when we fail to predict the future correctly: this provides a learning signal
- Are we learning an encoder – decoder pair?
- The encoder is **amortized** for fast prediction and leverages previous inferences
- The encoder is **augmented** generative decoder that helps it correct mistakes real time.
- Two design principles: amortization and augmentation

TL;DR

ThisPersonDoesNotExist.com uses AI generate endless fake faces

Hit refresh to lock eyes with another imaginary stranger

By James Vincent | Feb 15, 2019, 7:38am EST

SHARE

A few sample faces — all completely fake — created by ThisPersonDoesNotExist.com