Probabilistic modeling and inference meets Deep Learning (part 2)

Iain Murray

School of Informatics,

University of Edinburgh

Last time

There are multiple unsupervised learning principles:

GANs, auto-encoding, energy-based models, probabilistic models

Sometimes I want to fit $p(\mathbf{x})$ from data

Ideally no approximations to evaluate $p(\mathbf{x})$

I'd like to lean on advances in deep learning

Density estimation methods

Autoregressive models

Unnormalized models

Flows

Not this talk: models requiring inference, VAEs, graphical models

Autoregressive models

$$p(x_1)$$

$$p(x_2 \,|\, x_1)$$

$$p(x_3 \,|\, x_1, x_2)$$

$$p(x_4 \,|\, x_1, x_2, x_3)$$

$$p(\mathbf{x}) = p(x_1) \prod_{d=2}^{D} p(x_d \,|\, \mathbf{x}_{<d})$$

Making it scale

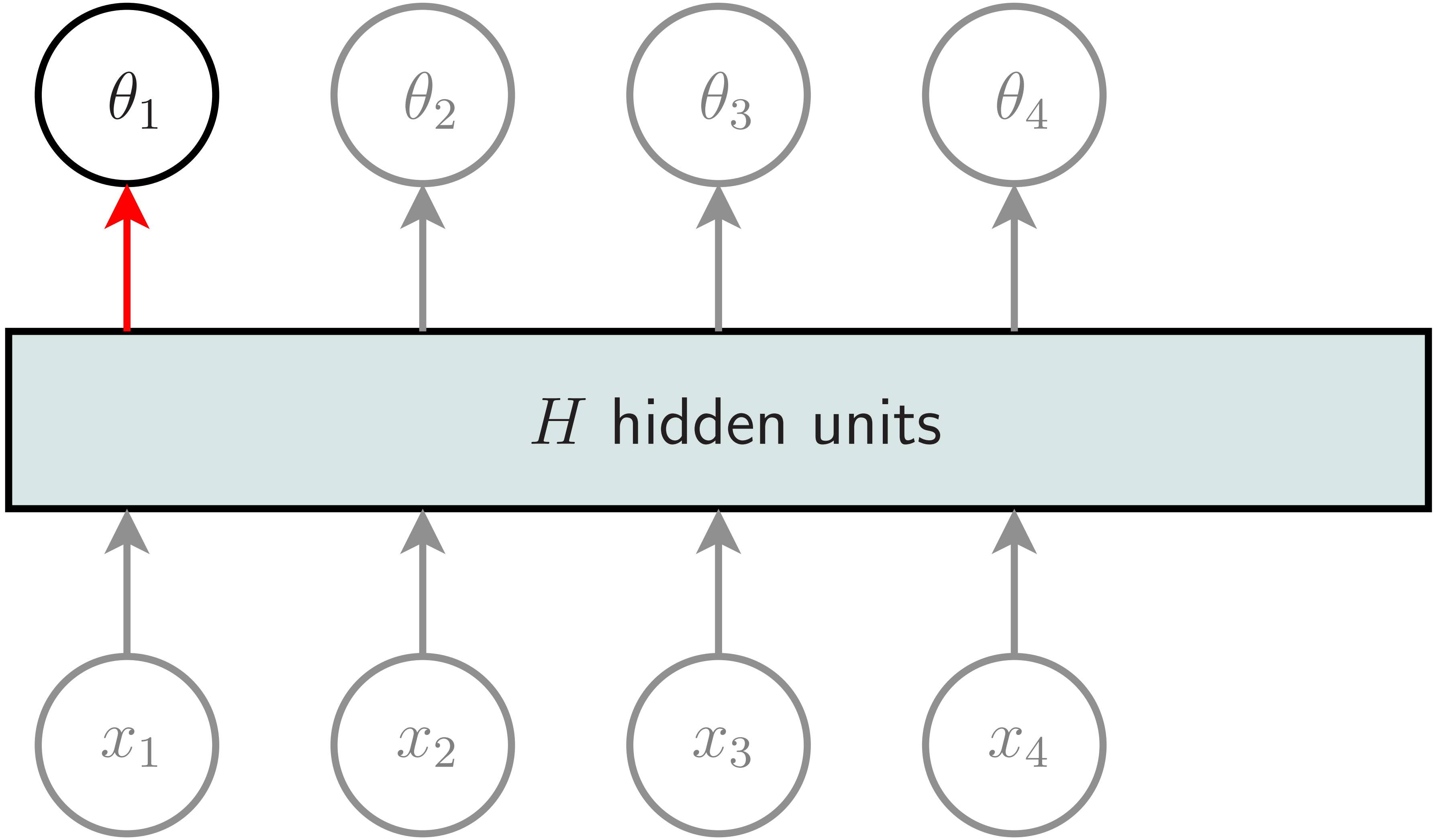
$$p(\mathbf{x}) \;=\; p(x_1)\prod_{d=2}^{D} p(x_d \,|\, \mathbf{x}_{<d})$$

D networks, one for each term

$O(DH)$ parameters each (H hiddens, D features)

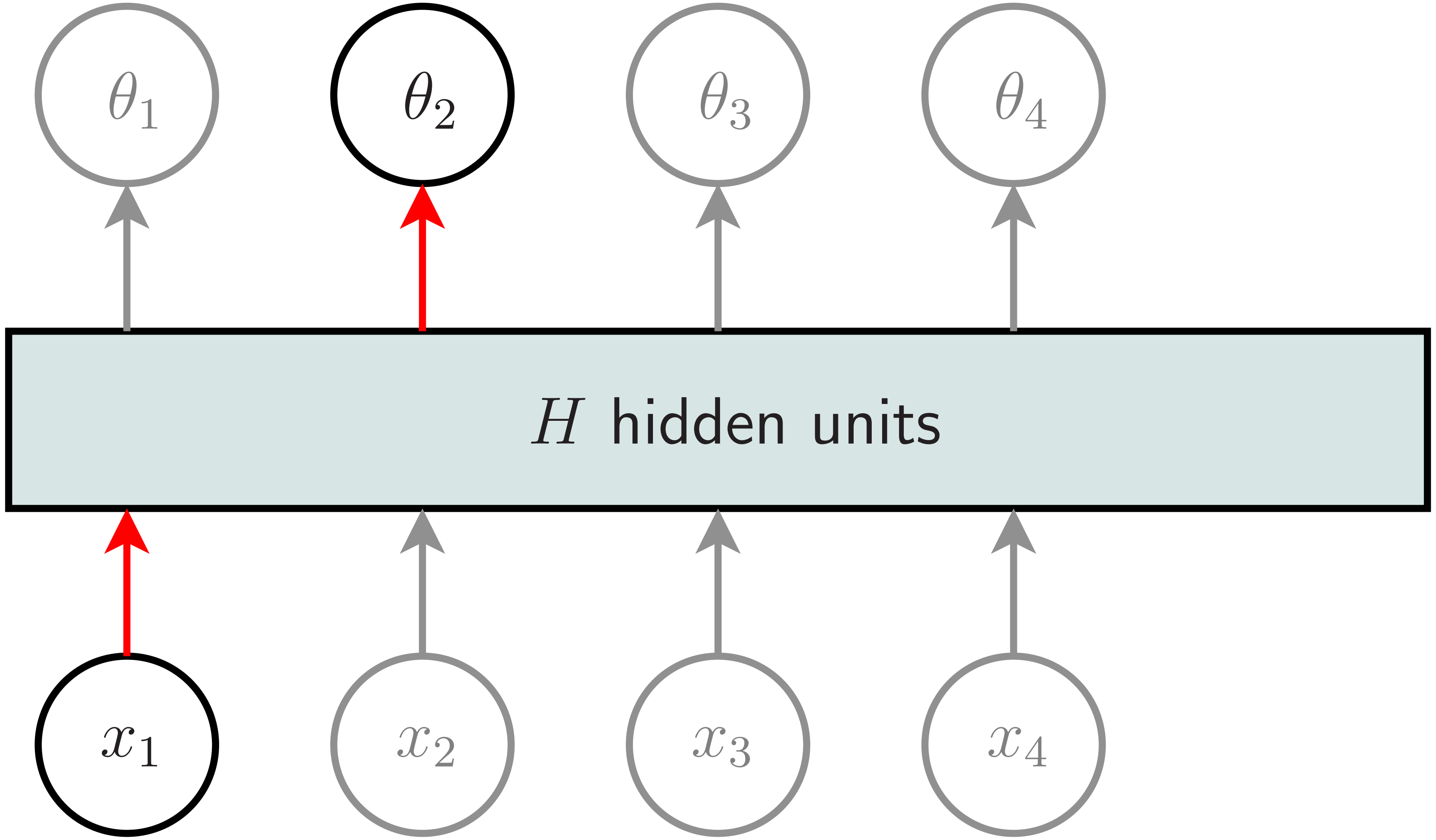
Total cost: $\mathcal{O}(D^2H)$

NADE: Sequential activation



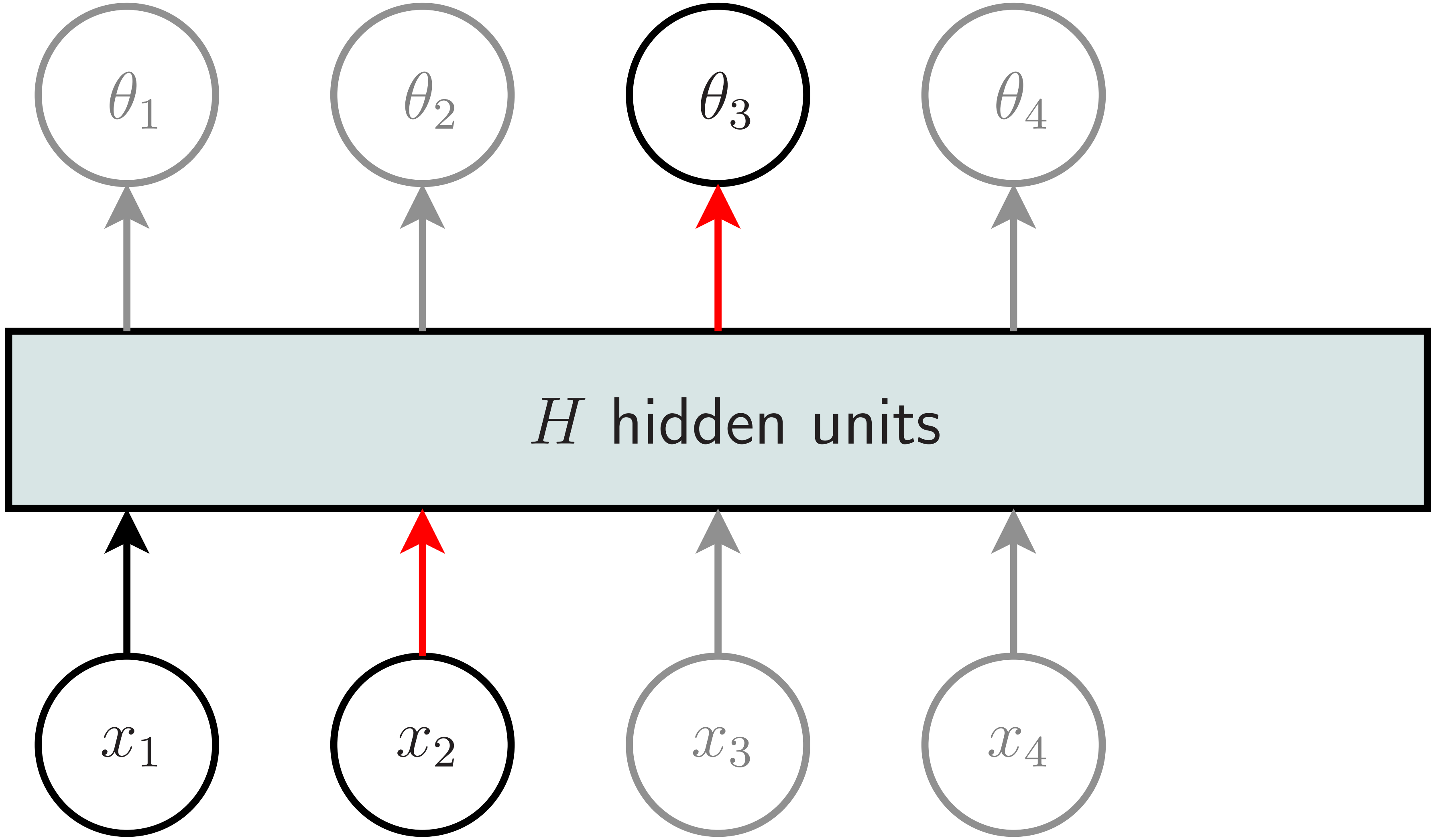
$$p(\mathbf{x}) = p(x_1 \,|\, \theta_1) \ldots$$

NADE: Sequential activation



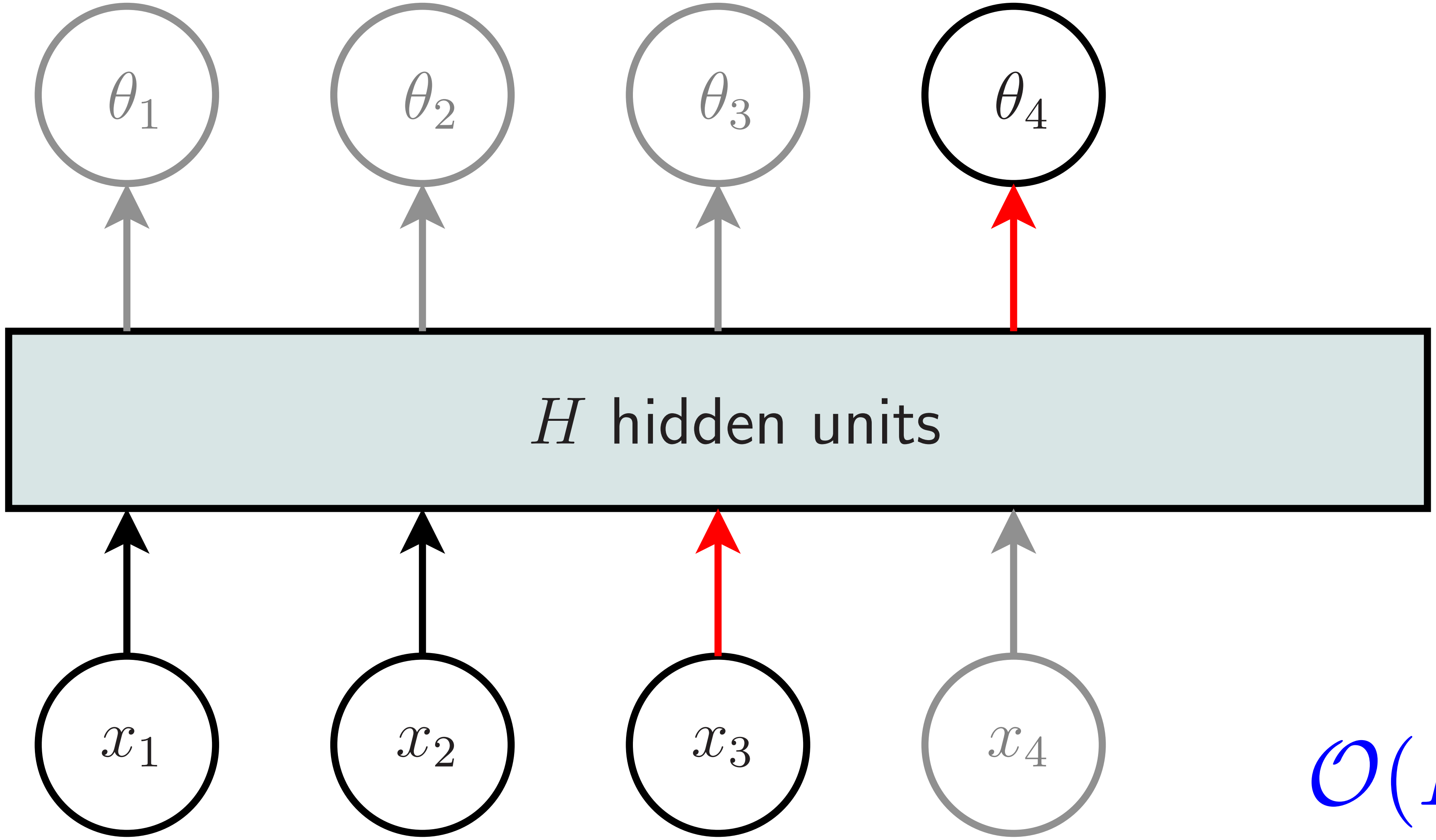
$$p(\mathbf{x}) = p(x_1 \,|\, \theta_1)\; p(x_2 \,|\, \theta_2(x_1)) \ldots$$

NADE: Sequential activation



$$p(\mathbf{x}) = p(x_1 \mid \theta_1)\, p(x_2 \mid \theta_2(x_1))\, p(x_3 \mid \theta_3(x_1, x_2)) \ldots$$

NADE: Sequential activation



$\mathcal{O}(DH)$

$$p(\mathbf{x}) = p(x_1 \,|\, \theta_1)\, p(x_2 \,|\, \theta_{2(x_1)})\, p(x_3 \,|\, \theta_{3(x_1, x_2)})\, p(x_4 \,|\, \theta_{4(x_1, x_2, x_3)}) \ldots$$

Mixture Density Networks (Bishop, 1994)

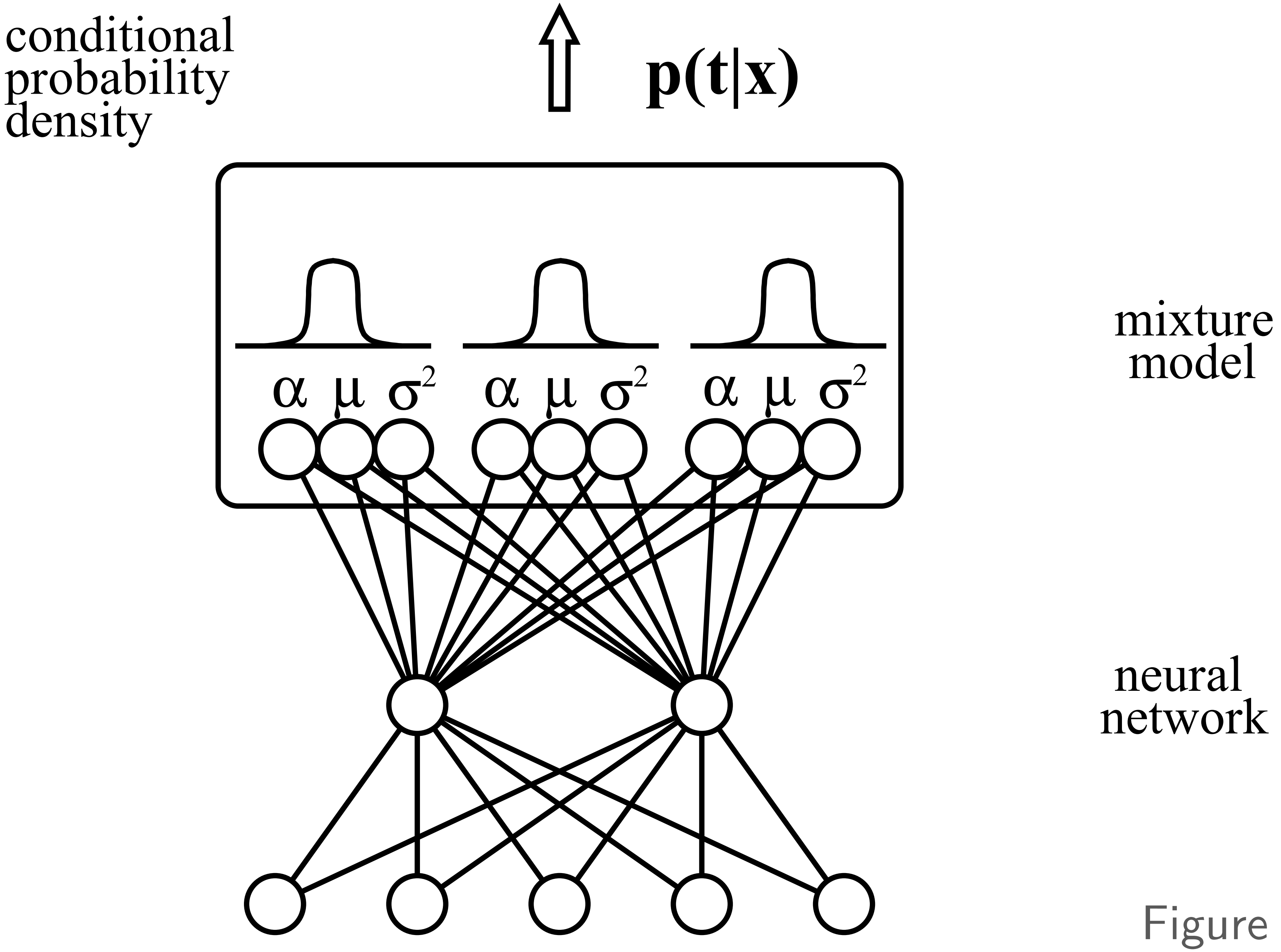


Figure stolen from Korin Richmond

Modelling Acoustic-Feature Dependencies with Artificial Neural-Networks: Trajectory-RNADE

Benigno Uría, Iain Murray, Steve Renals, Cassia Valentini-Botinhao and John Bridle.

Given a transcription, sampling from a good model of acoustic feature trajectories should result in plausible realizations of an utterance. However, samples from current probabilistic speech synthesis systems result in low quality synthetic speech. Henter et al. have demonstrated the need to capture the dependencies between acoustic features conditioned on the phonetic labels in order to obtain high quality synthetic speech. These dependencies are often ignored in neural network based acoustic models. We tackle this deficiency by introducing a probabilistic neural network model of acoustic trajectories, trajectory RNADE, able to capture these dependencies.

IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) pp4465–4469, 2015.

Superseded by WaveNet:

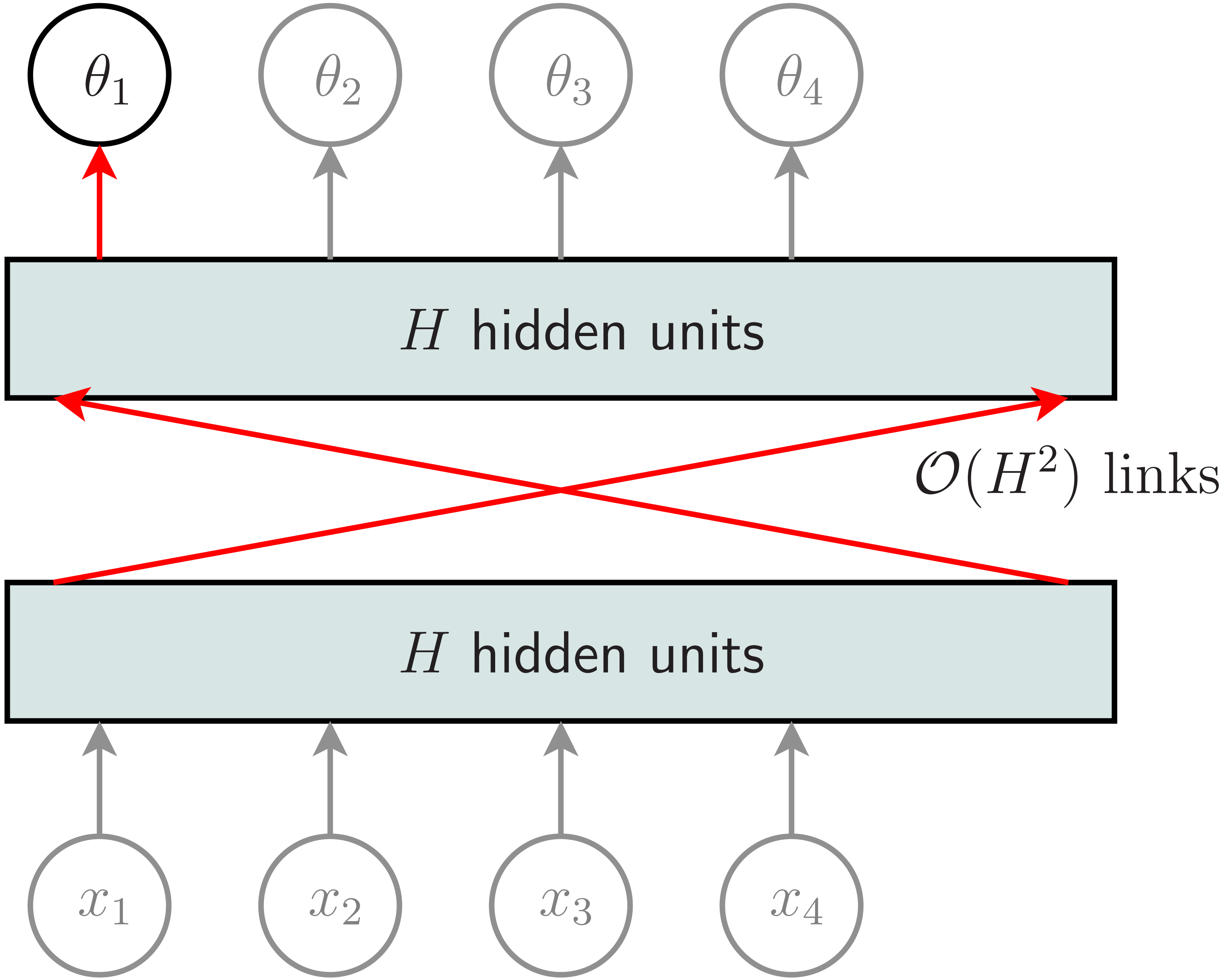
`deepmind.com/blog/wavenet-generative-model-raw-audio/`

How can we make it more expensive?

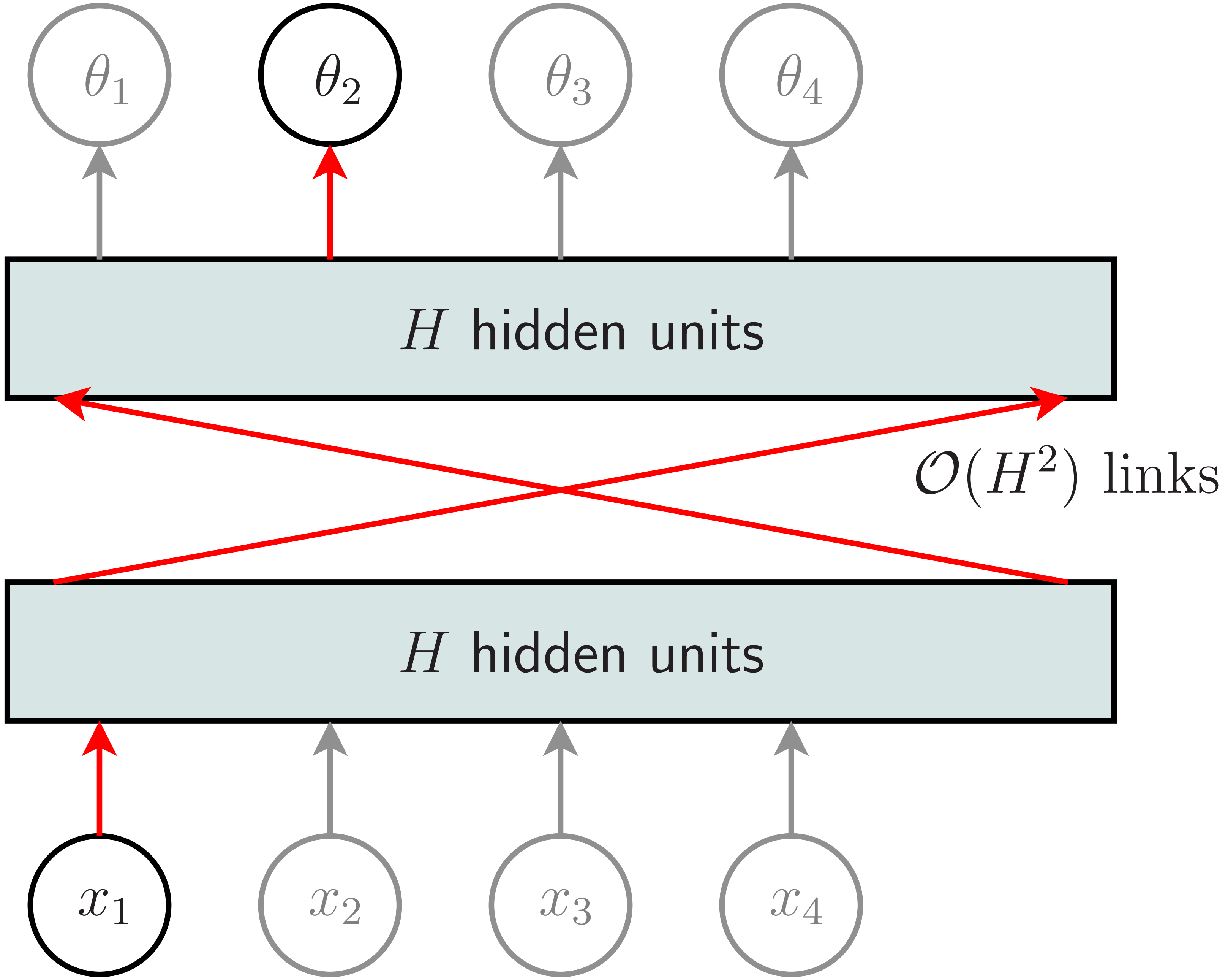
Recurrent neural networks (RNNs, LSTMs, pixel RNN, ...)

Deeper networks?

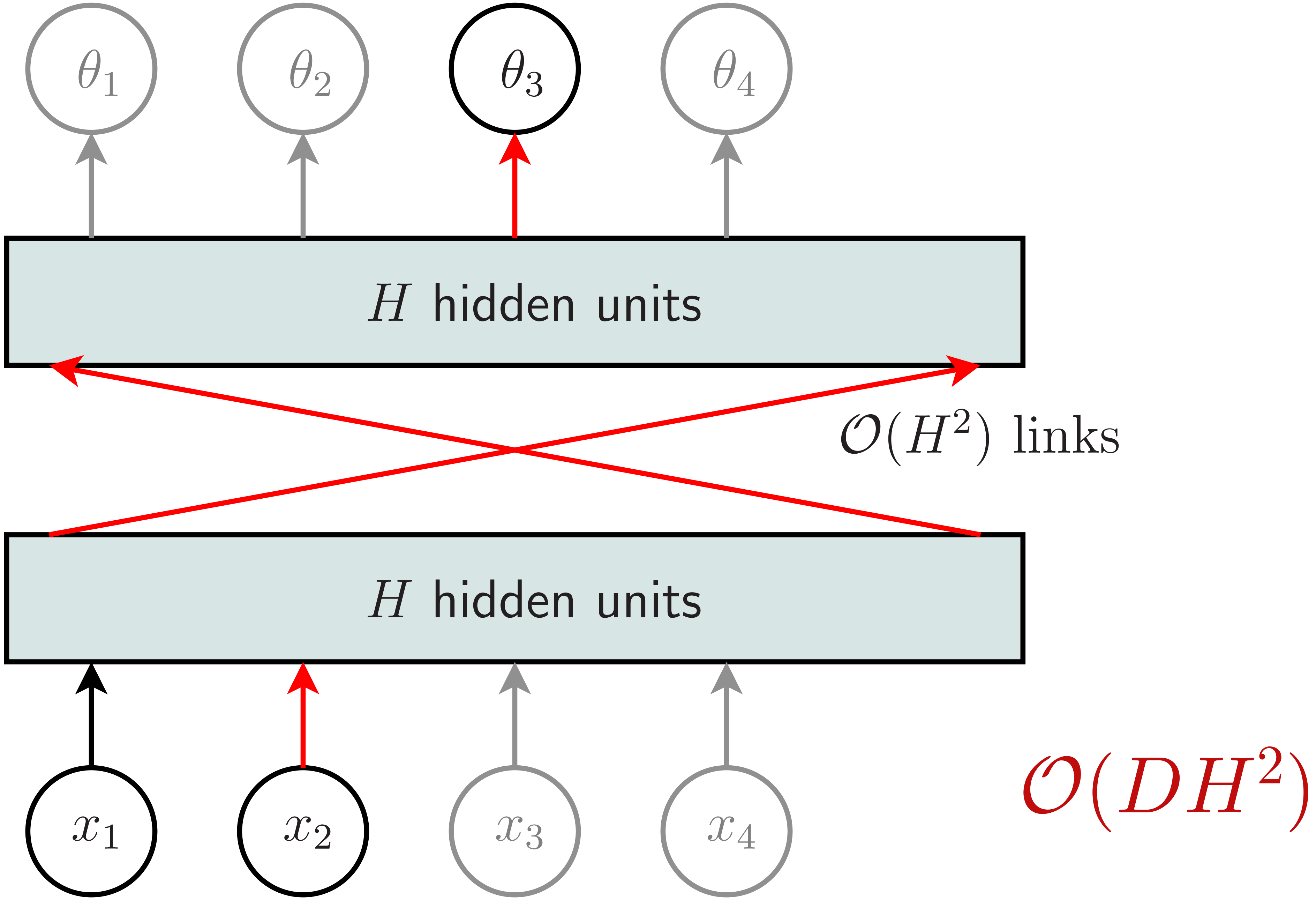
Sequential deep activation



Sequential deep activation



Sequential deep activation



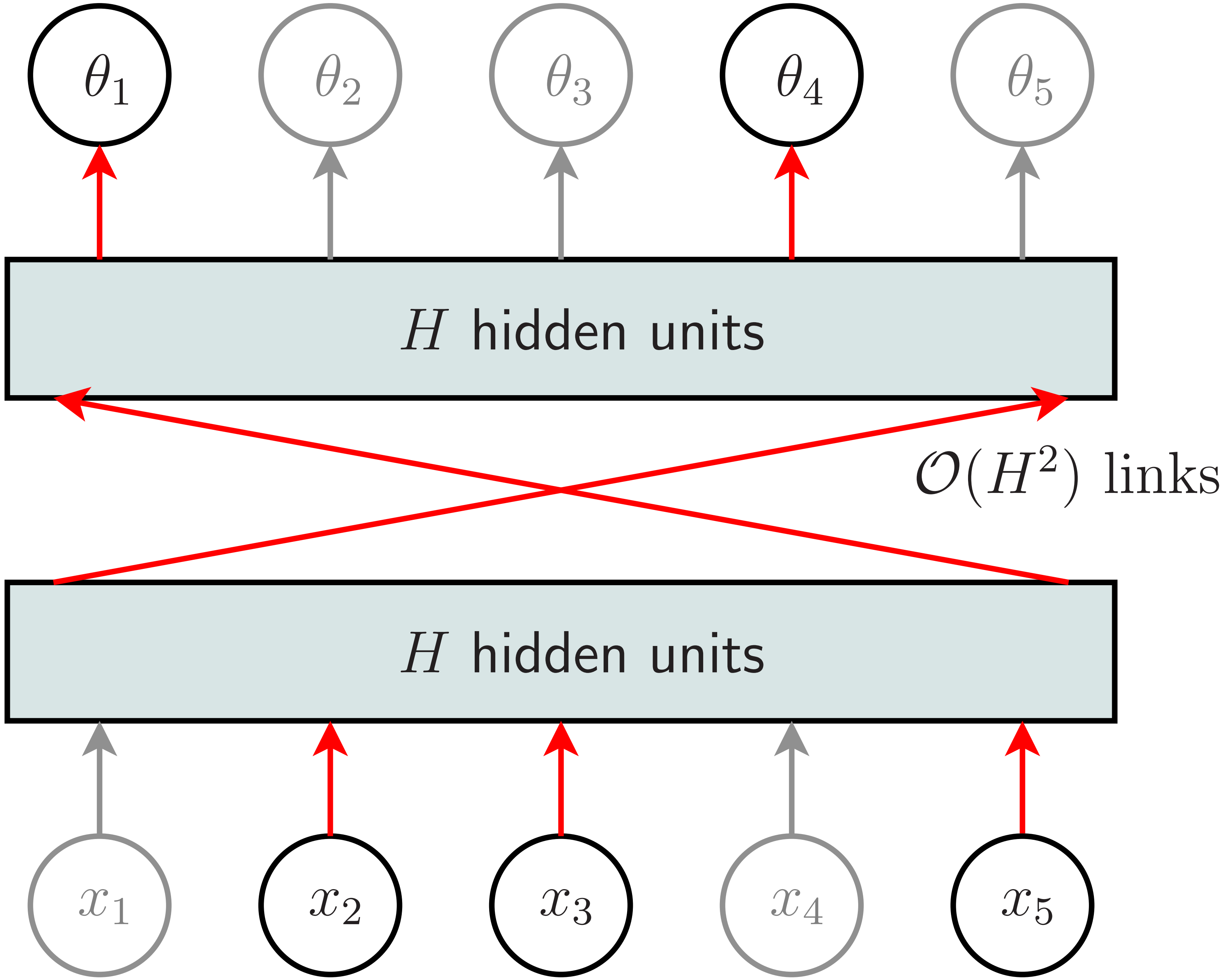
Autoencoders

One $\mathcal{O}(DH)$ pass

No density

Two ways of getting a direct density function

Deep NADE: a completing machine



A deep and tractable density estimator (ICML, 2014)

See also,
BERT arXiv:1810.04805

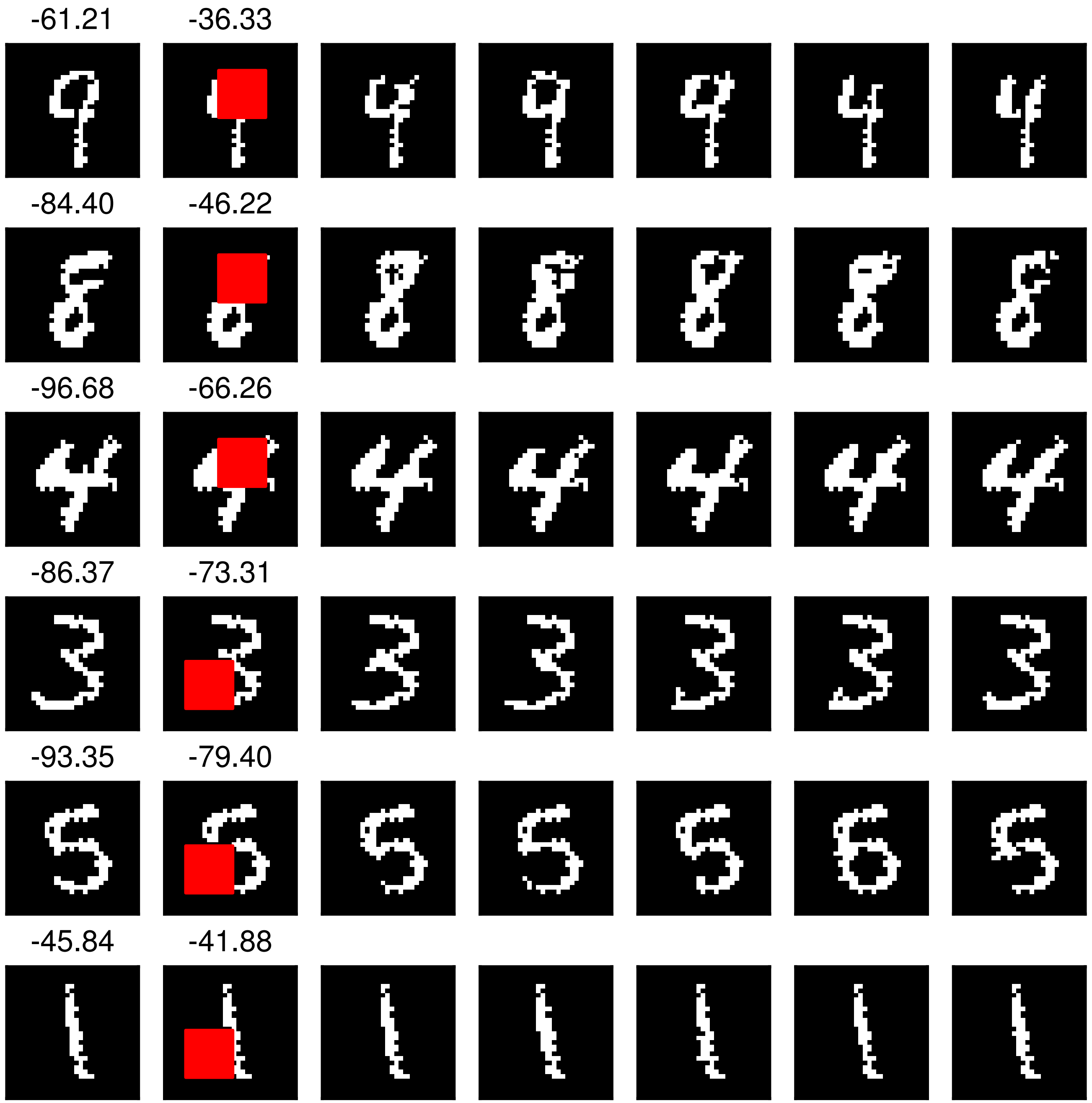
Missing inputs

Missing inputs are not the same as zero

Add binary mask, indicating what's missing

For one-hot encoded inputs, same as 'not present' symbol

Arbitrary ordering: inpainting



Deep NADE

Train time: $\mathcal{O}(DH + H^2)$ per update

Test time: predict features in any order

(can condition on observations)

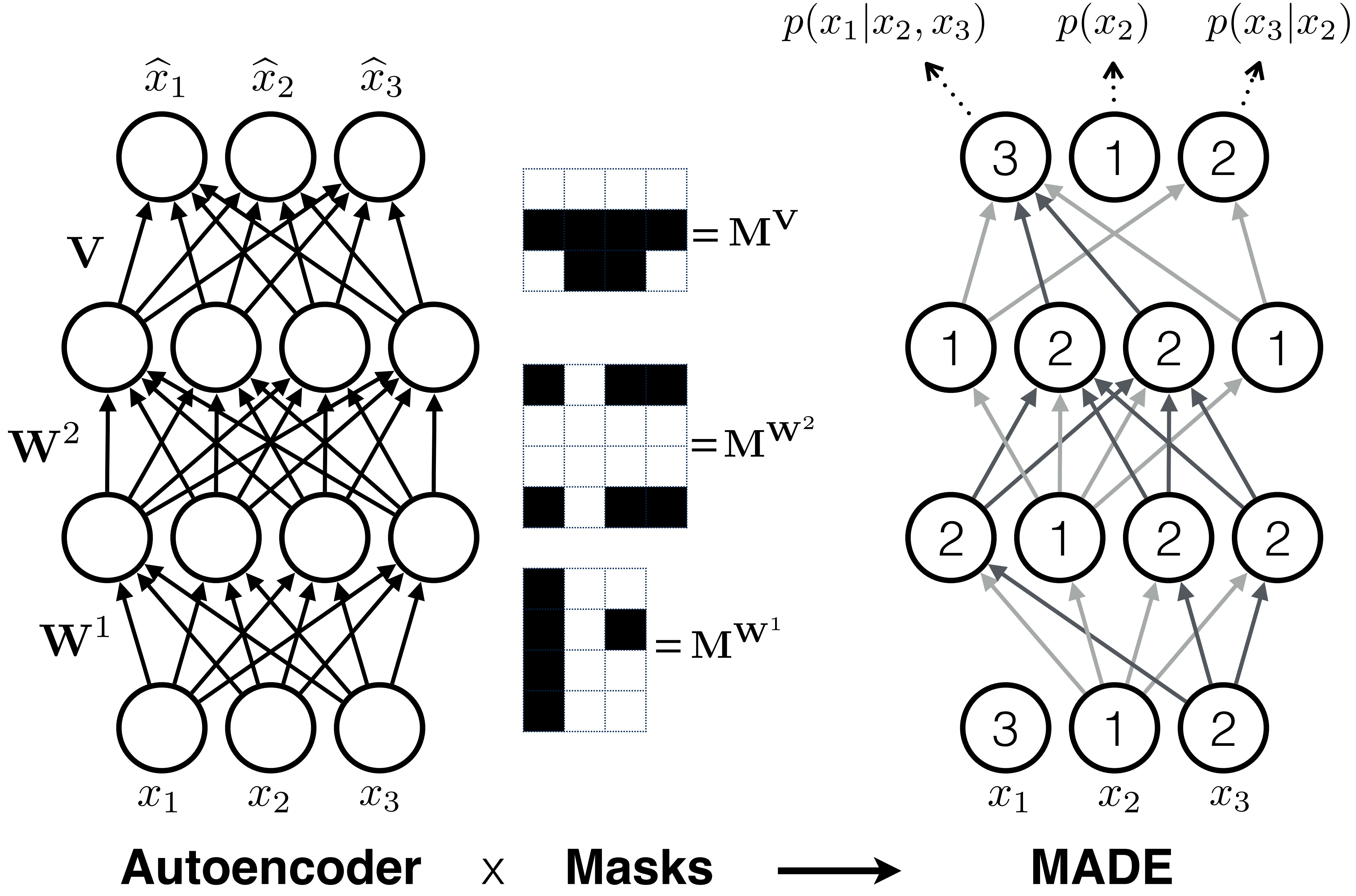
Different orderings not consistent:

— Seems bad, but. . .

— have trained large ensemble

— combining different orderings works better

MADE Masked Autoencoder Distribution Estimator



MADE vs Deep NADE

MADE: probability in one pass

(but sampling still needs multiple passes)

MADE's parameter masking $\rightarrow$ bigger models

Deep NADE trains an ensemble

MADE can too: randomize the masks

Which is better depends on data and details

Sequences and Images

Makes sense to use RNN cells and convolutions

WaveNet: causal convolutions use masking

Like MADE ‘quick’ to train, slow to sample.

Dilated causal convolutions long-term dependencies.

Image/sequence models often look/sound better.

More than densities might suggest.

Density estimation methods

Autoregressive models

NADE, Deep NADE, MADE, Pixel CNN/RNN, WaveNet, . . .

Unnormalized models

Flows

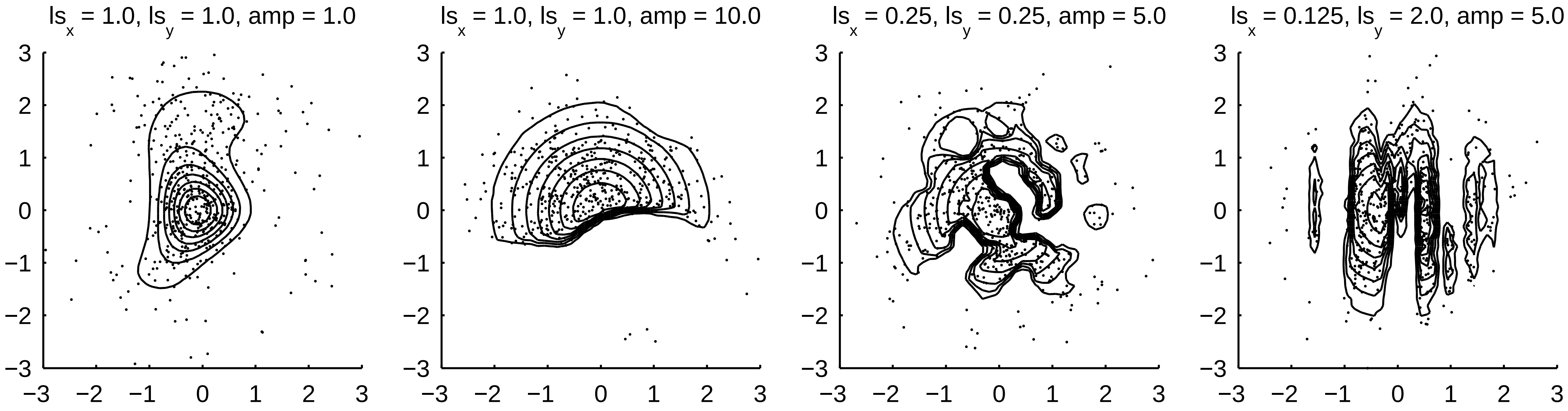
GP Density estimation

$$p(x \mid f) = \frac{1}{\mathcal{Z}(f)} \Phi(f(x))\, \pi(x)$$

$f \sim \mathcal{GP}$

$\Phi =$ sigmoidal function

$\pi =$ base measure



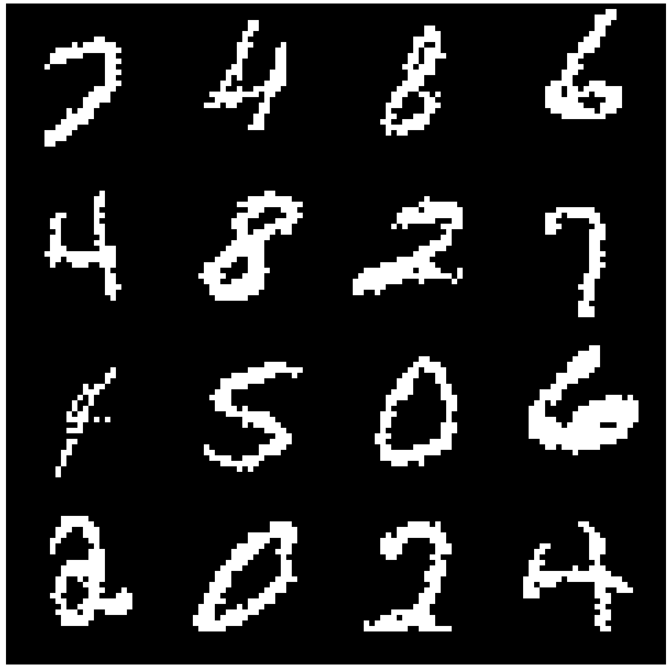
Gaussian Process Density Sampler, (Adams, Murray and MacKay, 2009).

Restricted Boltzmann Machines

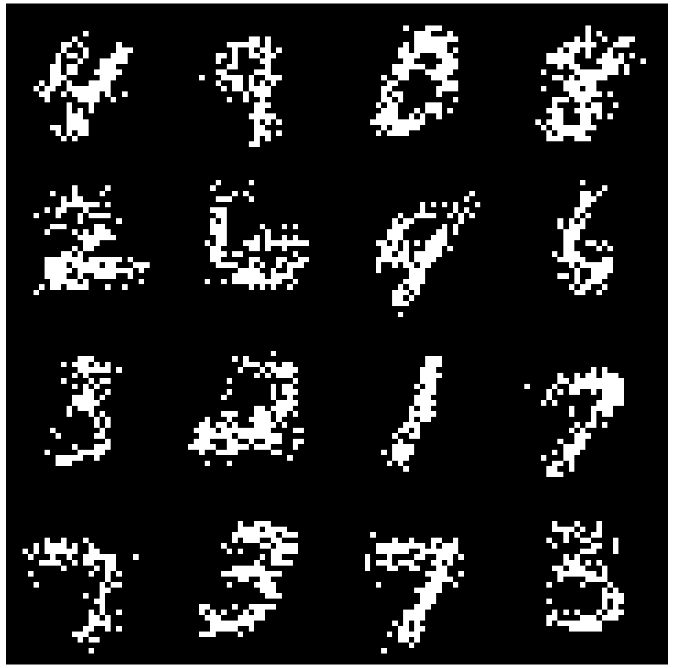
Fit to 60,000 binarized MNIST digits

Cf Mixture of multivariate Bernoullis (MoB), fit with EM

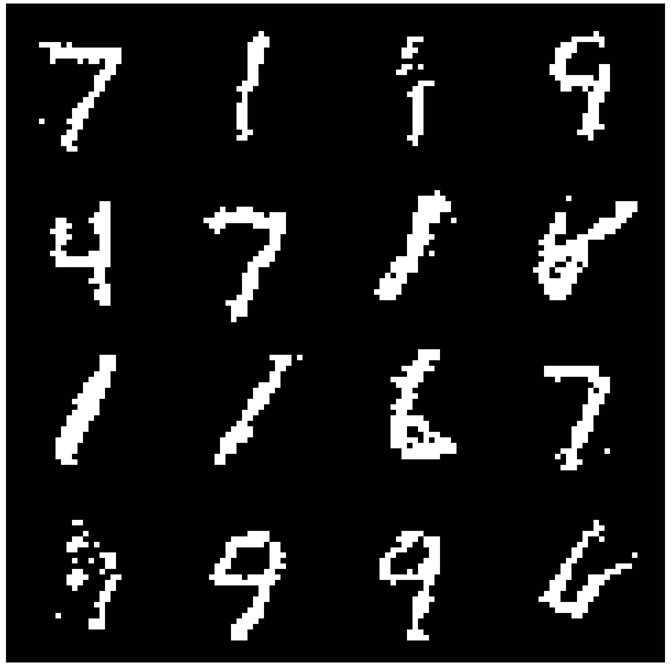
Data



MoB



RBM (CD25)

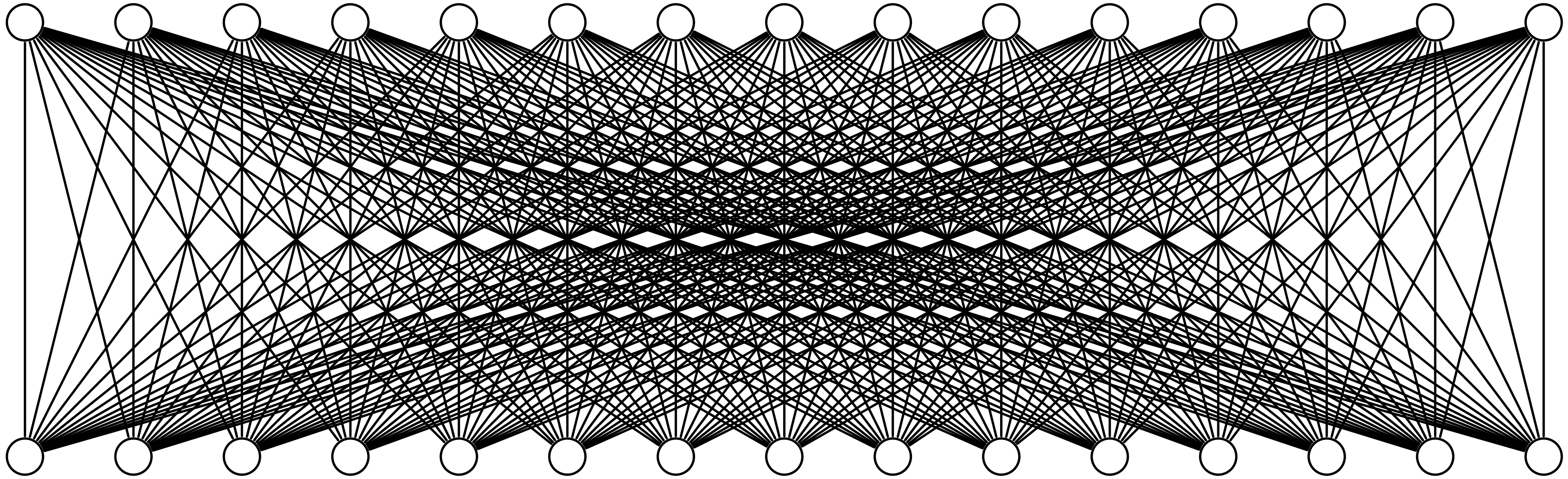


Test-set log-prob/digit: -143 nats -86 nats*

$*$ estimated

Restricted Boltzmann Machines

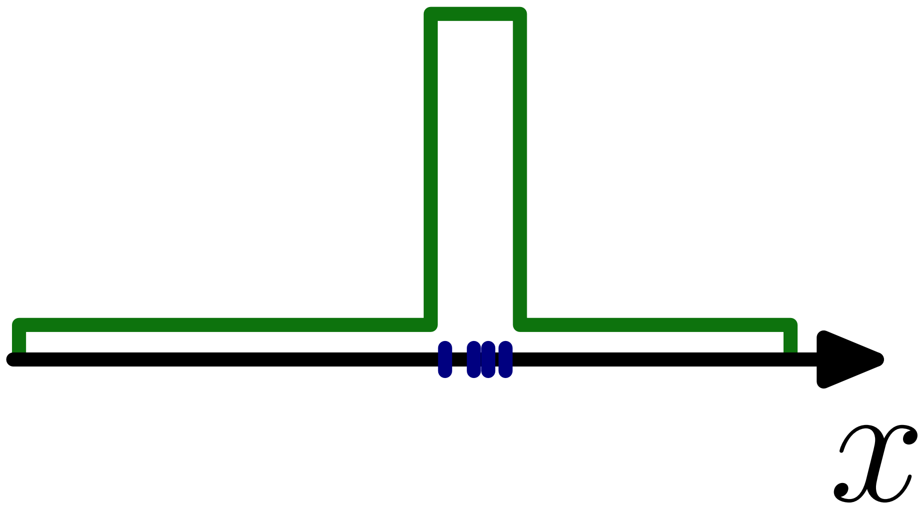
$$P(\mathbf{x}, \mathbf{h} \,|\, \theta) = \frac{1}{\mathcal{Z}(\theta)} \exp\left[\textstyle\sum_{ij} W_{ij} x_i h_j + \sum_i b_i^x x_i + \sum_j b_j^h h_j\right]$$

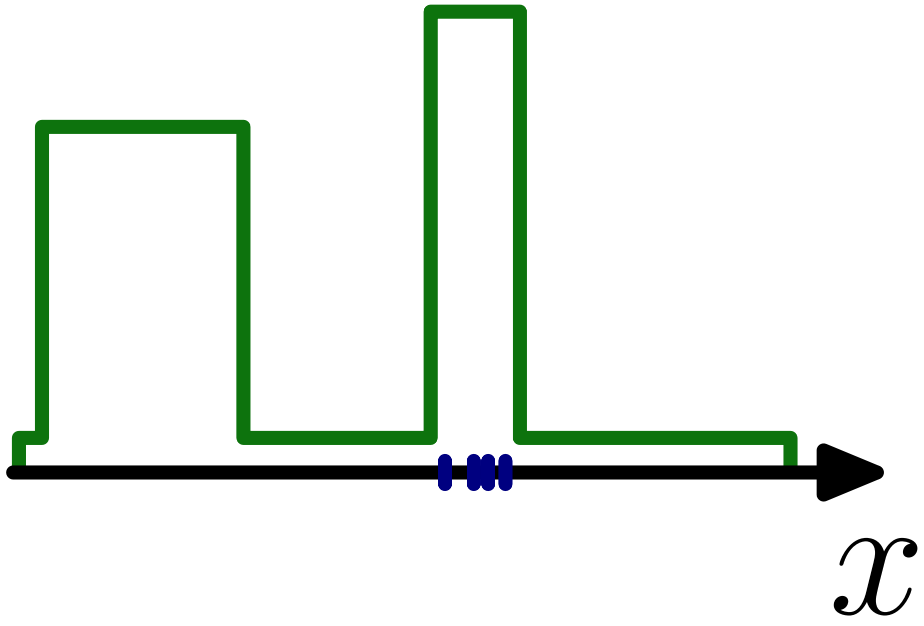


$$P(\mathbf{x} \,|\, \theta) = \sum_{\mathbf{h}} P(\mathbf{x}, \mathbf{h} \,|\, \theta) = \frac{1}{\mathcal{Z}(\theta)} \underbrace{\sum_{\mathbf{h}} \exp\left[\cdots\right]}_{f(\mathbf{x};\theta),\ \text{tractable}}$$

Why $\mathcal{Z}(\theta)$ matters

$\theta \quad \rightarrow \quad \theta'$





$$p(x \mid \theta) = \frac{f(x; \theta)}{\mathcal{Z}(\theta)}$$

Many learning principles

Negative examples

Contrastive Divergence, Stochastic approximation, Noise contrastive estimation

Variational inference $\rightsquigarrow$ NADE!

Score matching

Non-probabilistic energy-based models:

e.g., `http://yann.lecun.com/exdb/publis/pdf/lecun-06.pdf`

Density estimation methods

Autoregressive models

NADE, Deep NADE, MADE, Pixel CNN/RNN, WaveNet, . . .

Unnormalized models

GPDS, RBMs, energy-based models, undirected graphs, . . .

Flows

Transforming Noise

$$\mathbf{u} \sim \mathcal{N}(\mathbf{0},\ I), \qquad \mathbf{x} = f(\mathbf{u})$$

MacKay (1995, 1997)

Rippel and Adams (2013)

NICE, Real-NVP (Dinh, et al., 2014, 2016)

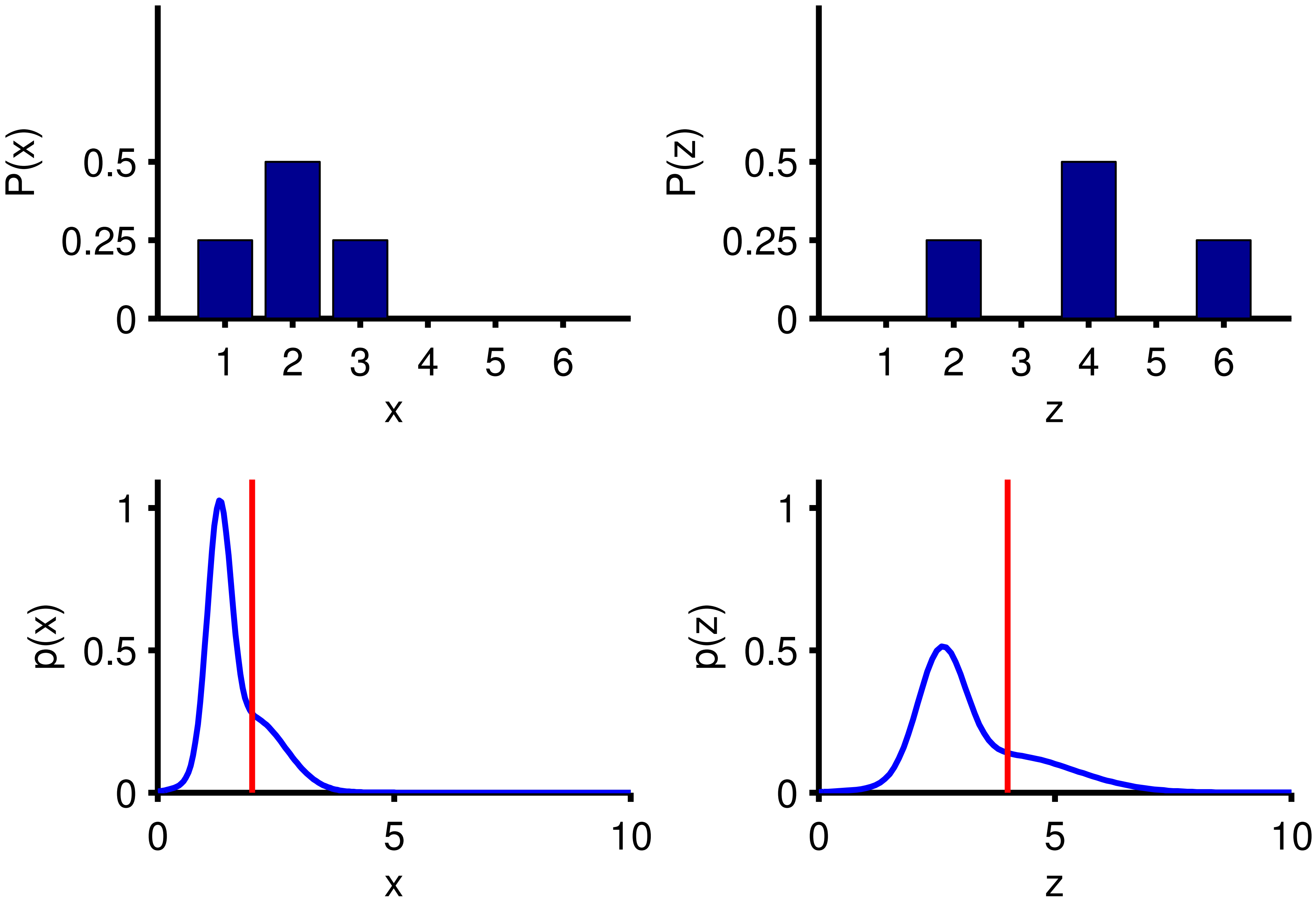
GANs

Reminder about densities

$$P(a < X < b) = \int_a^b p(x)\,\mathrm{d}x$$

$$P(x - {}^{\delta}\!/_{2} < X < x + {}^{\delta}\!/_{2}) \approx p(x)\delta$$

Transformations



$$z = 2x$$

$$P(z\!=\!4) = P(x\!=\!2)$$

$$p(z\!=\!4) \neq p(x\!=\!2)$$

$$p(z\!=\!4) = \tfrac{p(x=2)}{2}$$

For densities: $\int p(x)\,\mathrm{d}x = 1$

Nonlinear transformations

For 1–1 mappings between small elements δx and δz:

$$p(x)\,\delta x = p(z)\,\delta z$$

Notation heavily overloaded: different densities on LHS and RHS!

Taking limits:

$$p(z) \;=\; p(x(z))\left|\frac{\mathrm{d}x}{\mathrm{d}z}\right| \;=\; p(x(z))\Big/\left|\frac{\mathrm{d}z}{\mathrm{d}x}\right|$$

Example:

$$p(\sigma^2) \;=\; \frac{p(\log\sigma^2)}{\sigma^2}$$

Multivariate version with Jacobian:

$$p(\mathbf{z}) \;=\; p(\mathbf{x})\left|\begin{matrix} \frac{\partial x_1}{\partial z_1} & \frac{\partial x_1}{\partial z_2} & \cdots & \frac{\partial x_1}{\partial z_D} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x_D}{\partial z_1} & \frac{\partial x_D}{\partial z_2} & \cdots & \frac{\partial x_D}{\partial z_D} \end{matrix}\right|$$

Flows, invertible function

Model:

$$\mathbf{u} \sim \mathcal{N}(\mathbf{0},\ I), \quad \mathbf{x} = f(\mathbf{u})$$

Inference:

$$\mathbf{u} = f^{-1}(\mathbf{x})$$

Density:

$$p(\mathbf{x}) \ = \ \mathcal{N}(\mathbf{u}; \mathbf{0},\ I) \begin{vmatrix} \frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} & \cdots & \frac{\partial u_1}{\partial x_D} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial u_D}{\partial x_1} & \frac{\partial u_D}{\partial x_2} & \cdots & \frac{\partial u_D}{\partial x_D} \end{vmatrix}$$

Papers often swap the meaning of f and f^{-1}. We fit function from data $\mathbf{x}$ to random sources $\mathbf{u}$, so they call that the main function f.

Multiple ways to form a flow

Number of hiddens in all layers = Number of features

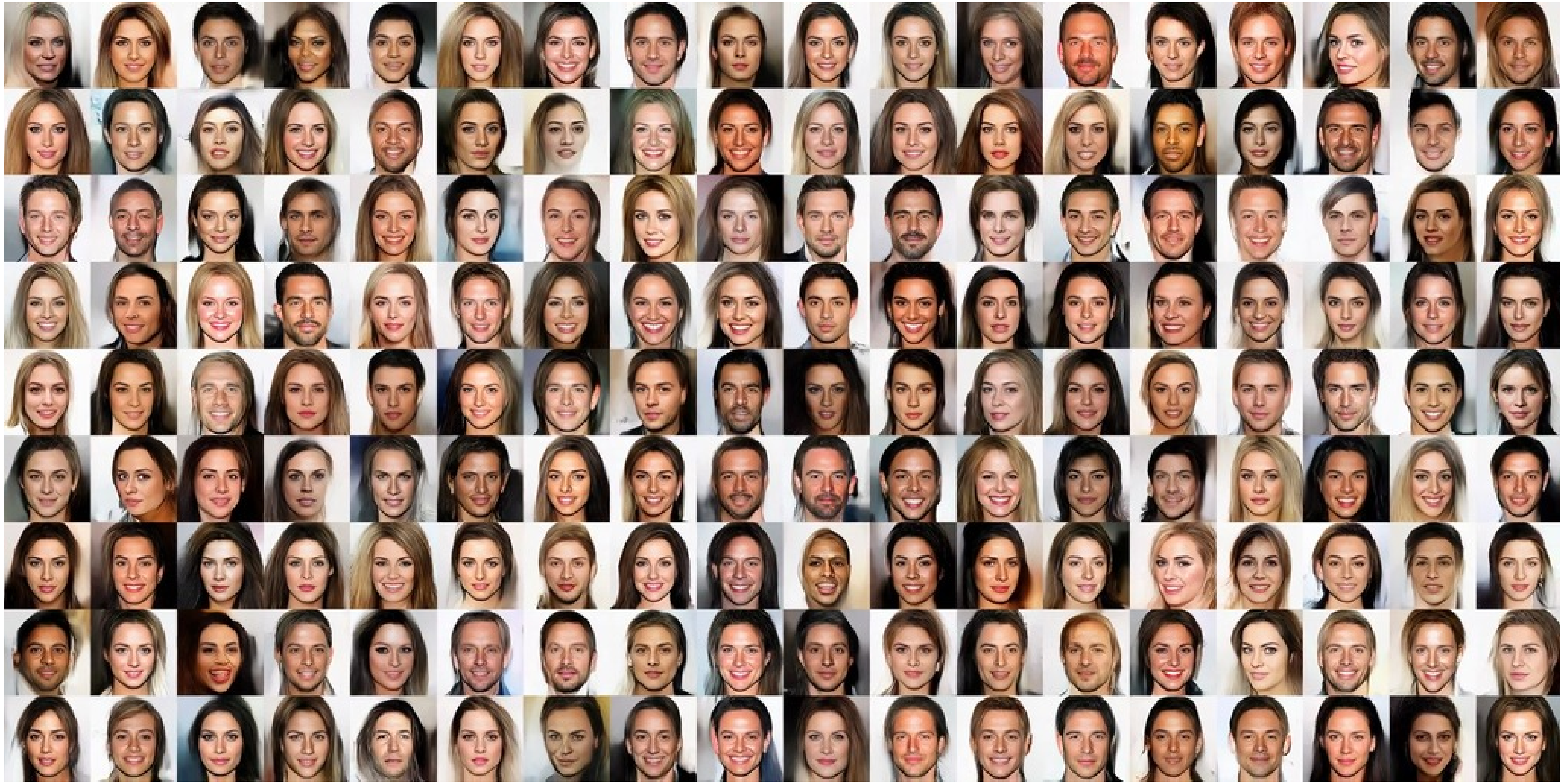
Real NVP (Dinh et al., ICLR 2017, arXiv:1605.08803)

real-valued non-volume preserving transformations

careful choice layers, batch-norm

Densities and sampling in only one feed-forward pass

Glow



https://blog.openai.com/glow/

MAF Masked Autoregressive Flow

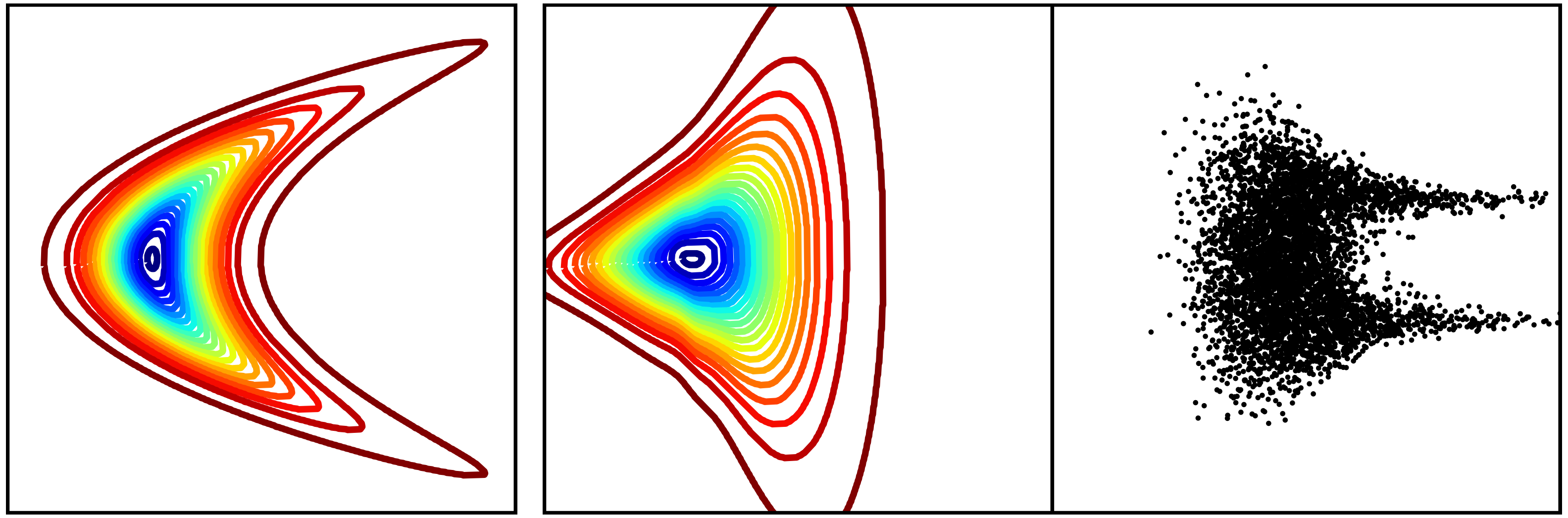
View MADE as a transformation: (real-valued version only)

$$\mathbf{u} \sim \mathcal{N}(0,\ I), \qquad \mathbf{x} = f(\mathbf{u})$$

$$x_d \ = \ \sigma_d(\mathbf{x}_{<d})\, u_d \ + \ \mu_d(\mathbf{x}_{<d})$$

`arXiv:1705.07057` (Papamakarios, Pavlakou, M, 2017)

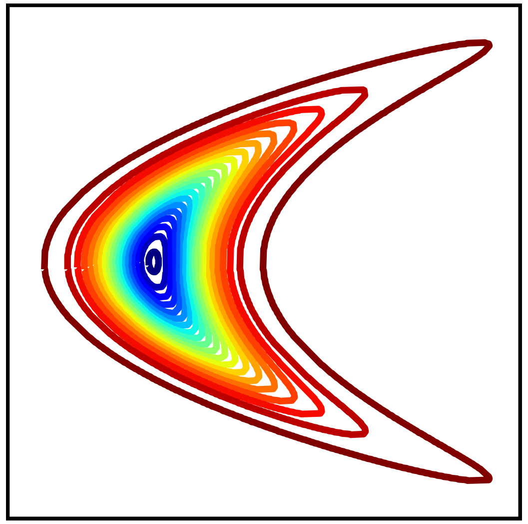
MAF Masked Autoregressive Flow



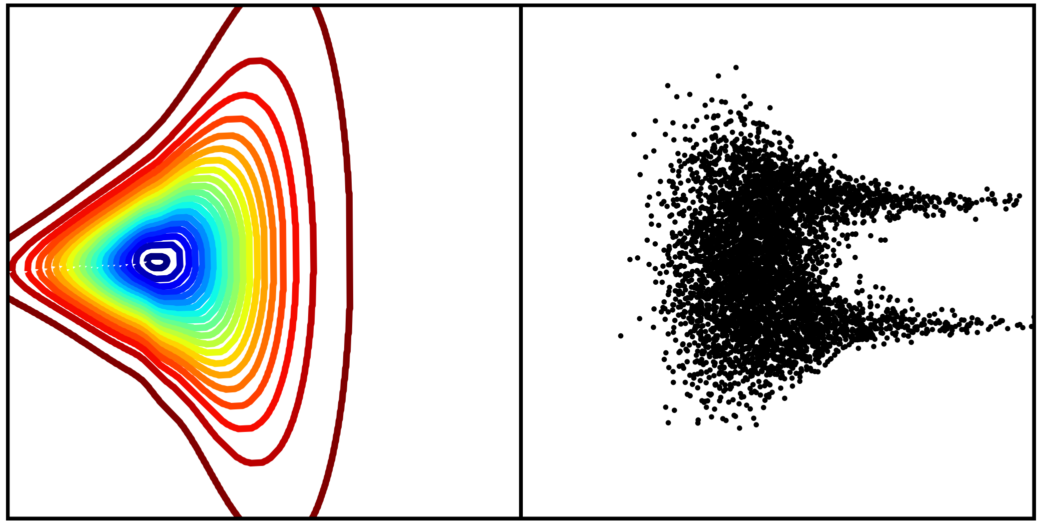
(a) Target density (b) MADE with Gaussian conditionals

Model checking: $\mathbf{u}$ implied by data

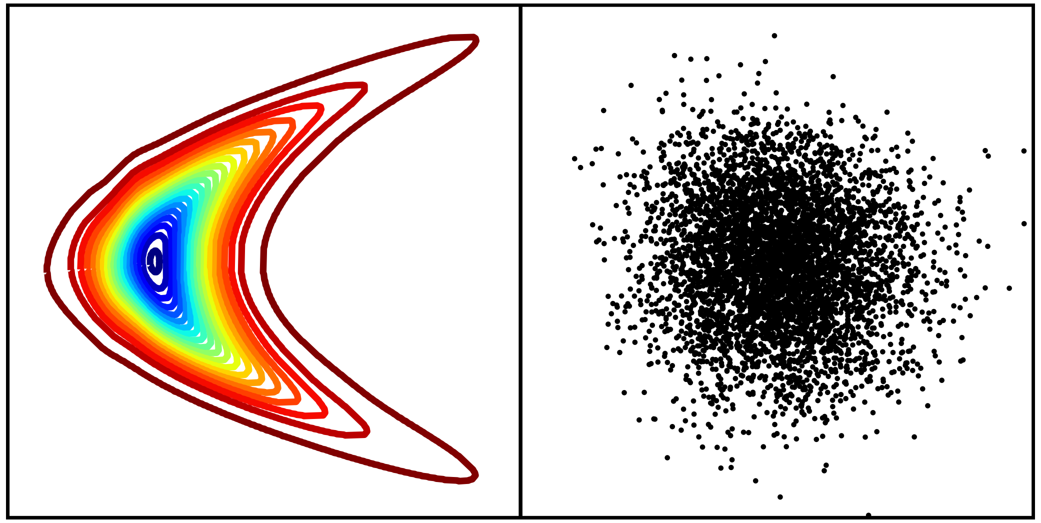
MAF Masked Autoregressive Flow



(a) Target density



(b) MADE with Gaussian conditionals



(c) MAF with 5 layers

MAF vs MADE

1 layer:

MAF is MADE with one Gaussian for each output

Bigger models:

MADE: choose whether to add layers or components

MAF: just join more layers

NAF: universal approximation (Huang et al., arXiv:1804.00779)

Inverse Autoregressive Flow (IAF)

Kingma et al. (2016) arXiv:1606.04934

Pointed out MADE is a flow.

Used inverse of function used by MAF

Sampling now one pass, but test-set densities slow

But know densities of points we sampled

Ideal for variational inference

Flows vs NADE/MADE

Both work well (WaveNet, WaveGlow, ...)

As usual which is better depends on data and details

Some flows give fast densities and samples

NADE/MADE can model discrete and mixed data

Take home message

Direct probabilistic models of large rich features sets

(including images, audio, text, posterior distributions, . . .)

simpler than general probabilistic models

Can use all the neural net tricks

convolutions, attention, ...

some constraints over auto-encoders $\rightarrow$ $p(\mathbf{x})$ over-kill for pre-training?

‘Likelihood free’ inference problems (or ‘ABC’)

Simulation based model: $\theta \rightarrow \mathbf{x}$

Likelihood $p(\mathbf{x} \,|\, \theta)$ exists, but we can’t evaluate it

Too many latent variables for MCMC

And/or $\mathbf{x}$ is a summary of a huge dataset

What to believe given data: $p(\theta \,|\, \mathbf{x})$

‘ABC’ Approximate Bayesian Computation

A surrogate likelihood:

$$P(\tilde{\mathbf{x}} \in \|\tilde{\mathbf{x}} - \mathbf{x}\| < \epsilon \mid \theta), \quad \tilde{\mathbf{x}} \sim P(\mathbf{x} \,|\, \theta)$$

Is simulation inside ϵ-ball?

Unbiased approximation:

$$\mathbb{I}(\tilde{\mathbf{x}} \in \|\tilde{\mathbf{x}} - \mathbf{x}\| < \epsilon), \quad \tilde{\mathbf{x}} \sim P(\mathbf{x} \,|\, \theta)$$

Recognition networks

$$\theta^{(s)} \sim p(\theta)$$

$$\mathbf{x}^{(s)} \sim p(\mathbf{x} \,|\, \theta^{(s)})$$

Training set: $\left\{\theta^{(s)}, \mathbf{x}^{(s)}\right\}_{s=1}^{S}$

Some of the relevant work

Hinton et al. (1995, Science) — Wake Sleep, Helmholtz machine

Morris (2001, UAI) — Recognition Networks

Blum & Francois (2010, S&C) — Conditional Gaussian, neural nets

Fan, Nott, Sisson (2012, Stat) — Mixture of experts

Mitrović, Dino Sejdinović, Teh (2016, ICML) — Kernel regression

...

Fast ϵ-free Inference of Simulation Models with Bayesian Conditional Density Estimation

Papamakarios and Murray (NeurIPS, 2016)
Lueckmann et al. (NeurIPS, 2017)

— Fit $\hat{p}(\theta \,|\, \mathbf{x})$ maximize $\sum_s \log \hat{p}(\theta^{(s)} \,|\, \mathbf{x}^{(s)})$ (or use SVI)

Fast ϵ-free Inference of Simulation Models with Bayesian Conditional Density Estimation

Papamakarios and Murray (NeurIPS, 2016)
Lueckmann et al. (NeurIPS, 2017)

— Fit $\hat{p}(\theta \,|\, \mathbf{x})$ maximize $\sum_s \log \hat{p}(\theta^{(s)} \,|\, \mathbf{x}^{(s)})$ (or use SVI)

— $\hat{p}(\theta \,|\, \mathbf{x}_{\text{observed}}) \;\rightarrow\;$ approx posterior

Fast ϵ-free Inference of Simulation Models with Bayesian Conditional Density Estimation

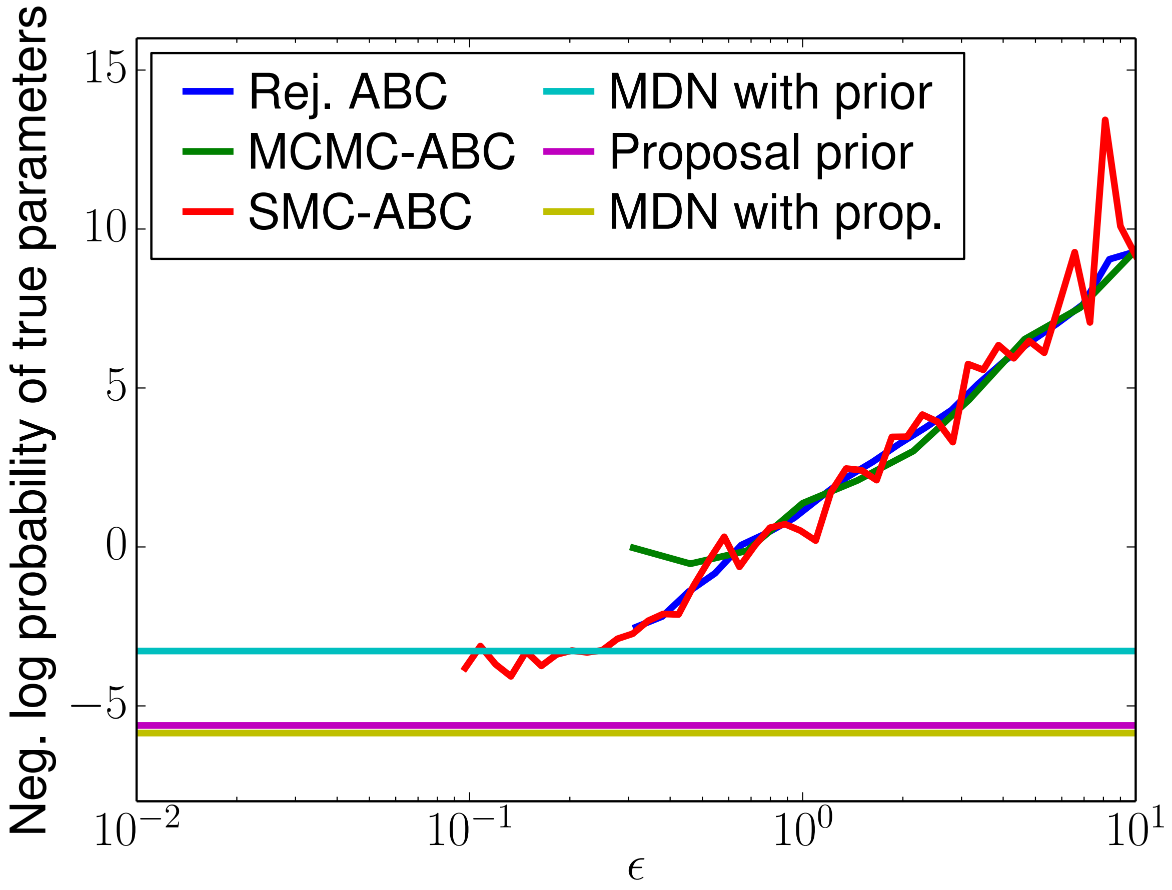
Papamakarios and Murray (NeurIPS, 2016)
Lueckmann et al. (NeurIPS, 2017)

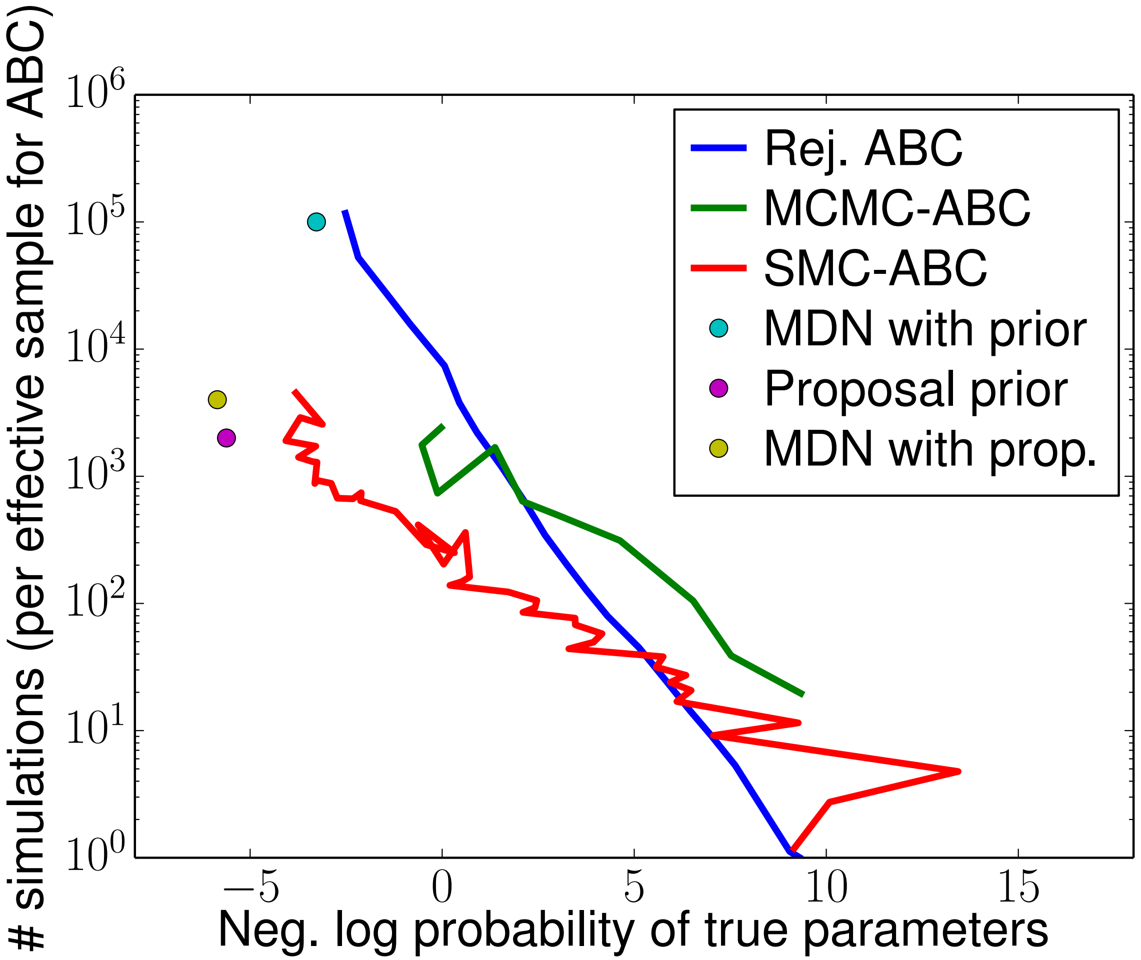
— Fit $\hat{p}(\theta \,|\, \mathbf{x})$ maximize $\sum_s \log \hat{p}(\theta^{(s)} \,|\, \mathbf{x}^{(s)})$ (or use SVI)

— $\hat{p}(\theta \,|\, \mathbf{x}_{\text{observed}}) \;\rightarrow\;$ approx posterior

— Refine fit: more simulations

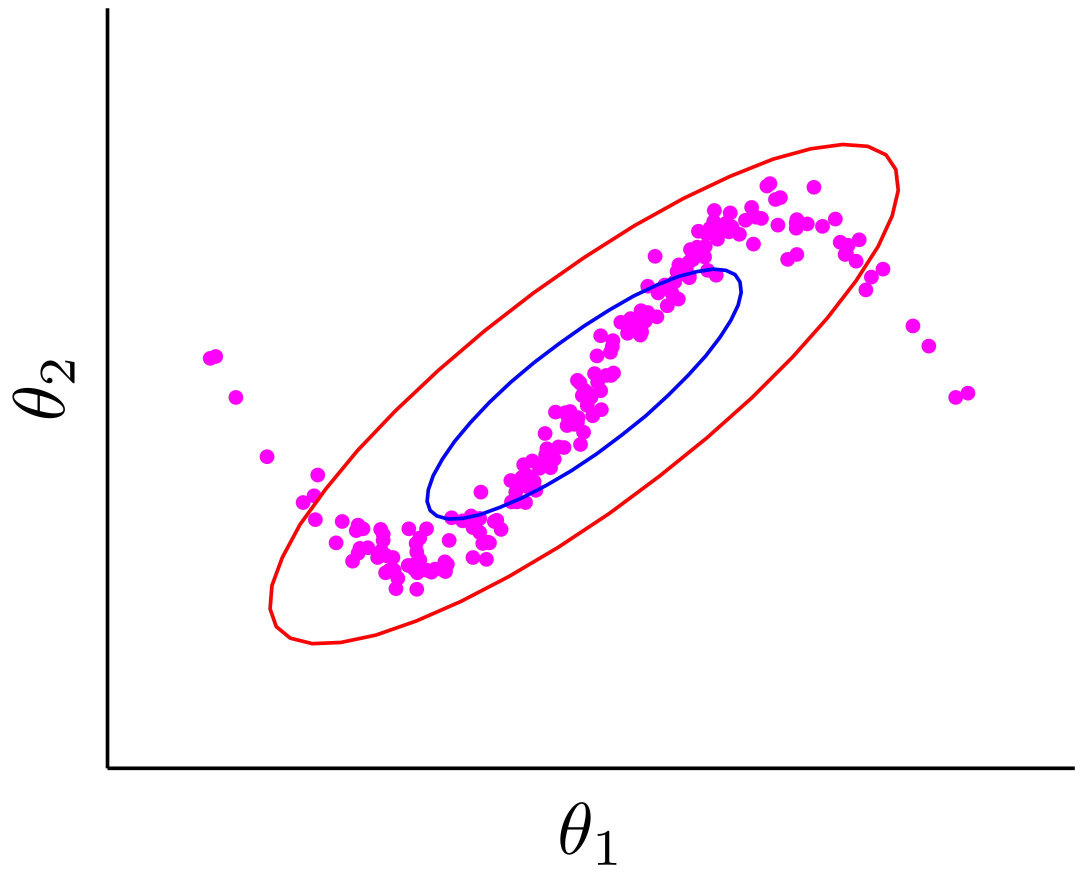
"ϵ-free ABC" (Papamakarios & Murray, NeurIPS 2016)



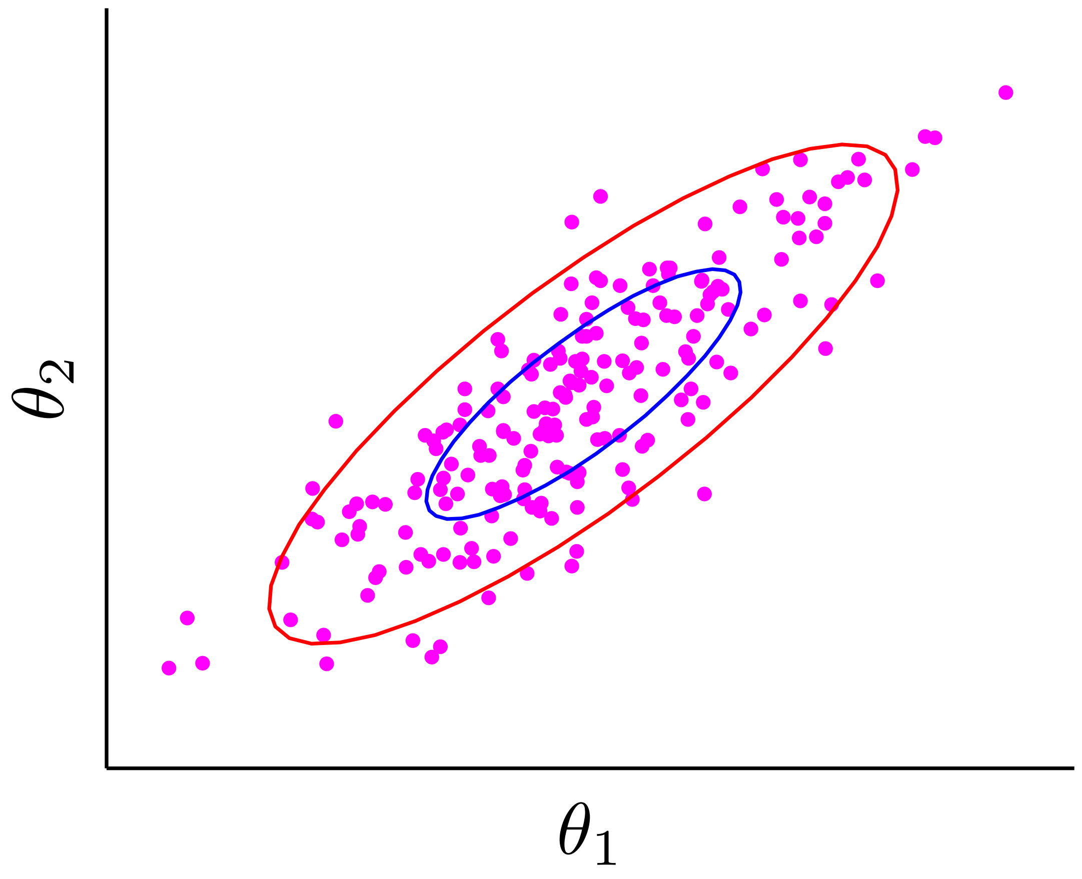


Results on Lotka–Volterra test case

Underfitting



True posterior samples



samples from Gaussian fit

Challenges of recognition networks

Need to consider how chose simulations (hard!)

First exploration algorithm tied to MDNs with one component

Stability and training difficulty (hard to quantify)

Still using lots of simulations

Sequential Neural Likelihood

(Papamakarios, Sterratt, & M, AISTATS 2019)

0. Simulate the model a bit

1. Fit $\hat{p}(\mathbf{x} \,|\, \theta)$ using any (conditional) density (we used MAF)

2. Run MCMC on $p(\theta \,|\, \mathbf{x}_{\text{observed}}) \;\propto\; \hat{p}(\mathbf{x}_{\text{observed}} \,|\, \theta)\, p(\theta)$

3. Add to simulation data. GoTo 1.

SNL (Papamakarios, Sterratt, & M, AISTATS 2019)

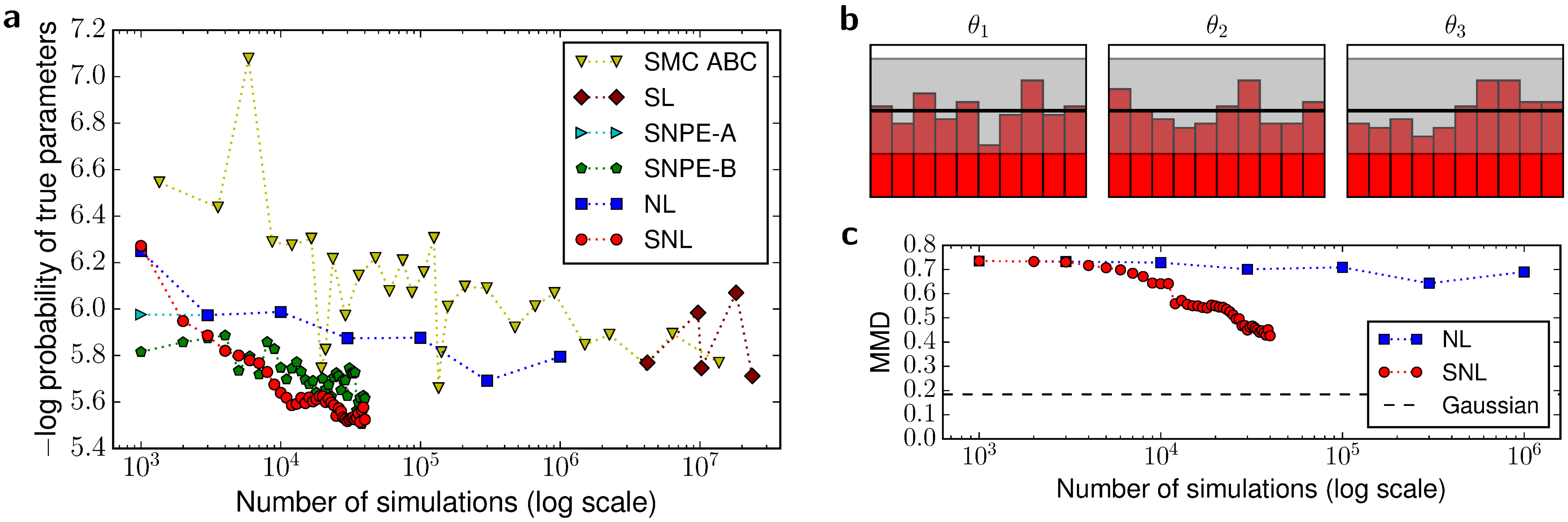


Figure 4: Hodgkin–Huxley model. **a**: Accuracy vs simulation cost: bottom left is best. **b**: Calibration test for SNL, histogram outside gray band indicates poor calibration. **c**: Likelihood goodness-of-fit vs simulation cost, calculated at true parameters.

Neural nets and inference

Model priors (image denoising, Milky Way)

Model posterior (SNPE, NeurIPS 2016/2017)

Model likelihood (SNL, AISTATS 2019, arXiv:1805.07226)

Comparison at NeurIPS BDL 2018, arXiv:1811.08723

Wider view

When you can, start with classification

or regression, simple "supervised learning"

Autoencoding and density estimation

starting points for representing large objects

Flows convenient/fast; autoregression more flexible

Bonus topic

Model compression

Bucilă et al. (KDD, 2006)

"Often the best performing supervised learning models are ensembles of hundreds or thousands of base-level classifiers."

Understanding capacity

Ba and Caruana, NeurIPS 2014, arXiv:1312.6184

"we show that shallow feed-forward networks can learn the complex functions previously learned by deep nets and achieve accuracies previously only achievable with deep models"

. . . but see paper and follow-up work for caveats!

Matching densities

Distilling Model Knowledge

George Papamakarios (MSc, 2015), arXiv:1510.02437

Bayesian Dark Knowledge

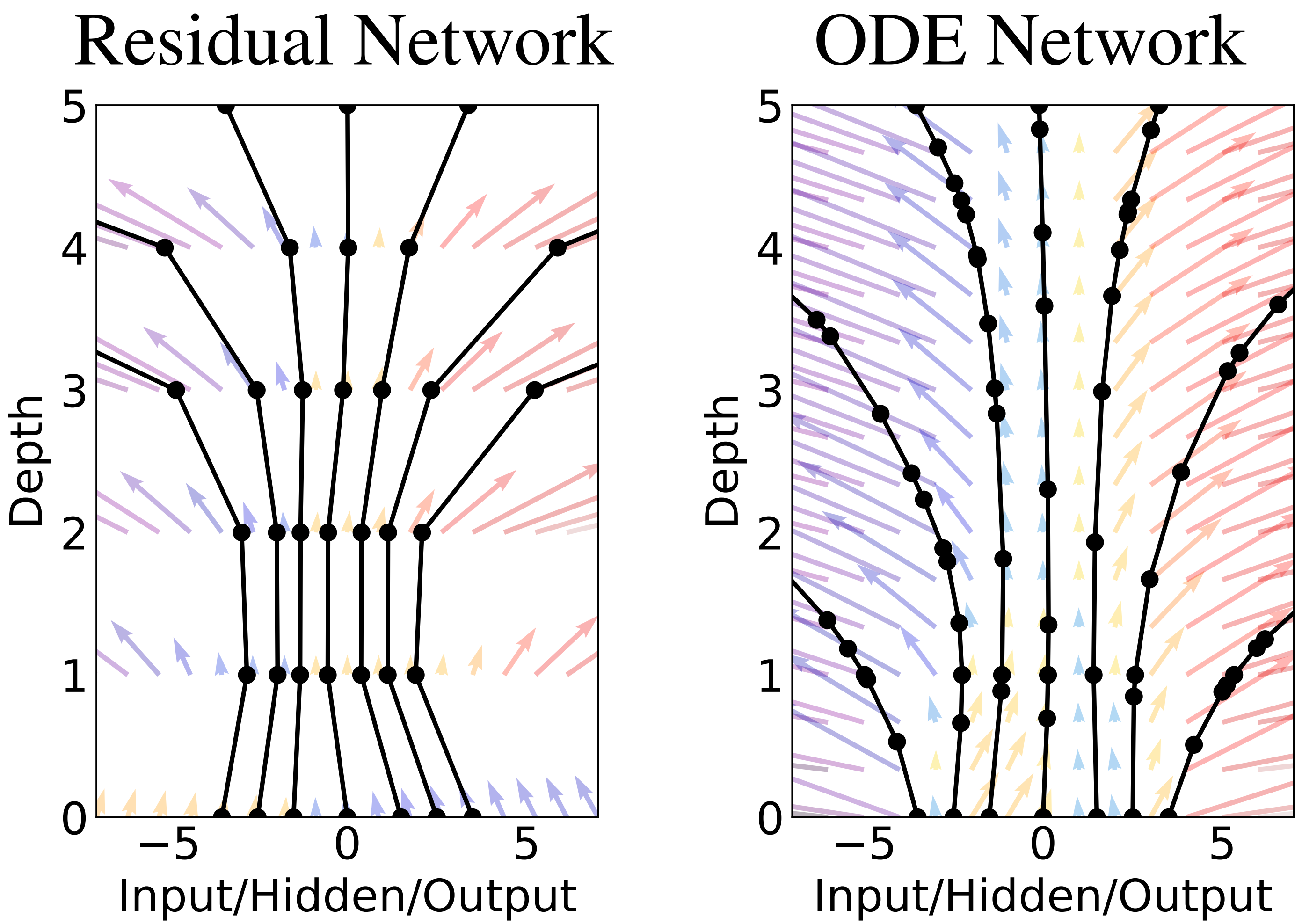
Korattikara et al., (NeurIPS, 2015), arXiv:1506.04416

Parallel WaveNet

Oord et al., (ICML, 2018), arXiv:1711.10433

Neural Ordinary Differential Equations

Chen et al. (NeurIPS, 2018)



Continous time MCMC

Flurry of recent papers. Example:

The zig-zag process and super-efficient sampling for Bayesian analysis of big data

Bierkens et al. (2016), arXiv:1607.03188