Machine Learning for Probabilistic Modelling

Iain Murray

School of Informatics, University of Edinburgh

Hello! These slides were visual aids for a talk, and weren't designed to be read. I've inserted some notes here to summarize what the points were supposed to be, and to give further references.

1. Hierarchical modelling is essential. Many models contain large numbers of 'nuisance variables'. We need to *learn* how these are distributed, because if we make assumptions (including vague or so-called 'uninformative' ones), we'll simply get wrong answers. The model I discussed was referring to: "Inferring the force law in the solar-system from a snapshot", Bovy et al., 2010.
`http://arxiv.org/abs/0903.5308`

More thoughts and references on hierarchical modelling are in my discussion `http://homepages.inf.ed.ac.uk/imurray2/pub/11catchup/catchup.pdf` of `http://dx.doi.org/10.1111/j.1467-9868.2011.01025.x`

Hierarchical models can be hard to infer. Example:
`http://homepages.inf.ed.ac.uk/imurray2/pub/10hypers/`

2. Real-world models have a lot of messy detail:
1) complicated instrument-error distributions we may not care about;
2) theory encoded in expensive-to-run simulations.

Machine Learning can help.
If we don't want wrong models (point 1.) we need to learn from large amounts of data about our instruments, and from simulation data describing our theories. I've been involved in a series of papers on flexible black-box probabilistic models that could be used here:
`http://homepages.inf.ed.ac.uk/imurray2/pub/11nade/`
`http://homepages.inf.ed.ac.uk/imurray2/pub/13rnade/`
`http://homepages.inf.ed.ac.uk/imurray2/pub/14dnade/`

4. Probabilistic inference methods need extending.

Approximate inference is a heavily-mined and active area. Getting up-to-speed and finding a niche is challenging. However, work in this area is important. In deep and wide graph structures, with billions of observations in some of the plates, it's hard to do fully Bayesian inference.

Some of my work has been on identifying common small inference problems, which are usually only part of an analysis, and developing easier-to-use inference methods for them. E.g. `http://homepages.inf.ed.ac.uk/imurray2/pub/10ess/`

I'm now also interested in developing easier-to-use methods to summarize and communicate the results of local inferences across large models. I believe the way forward is fitting flexible representations of beliefs, by combining machine learning methods and approximate inference algorithms.

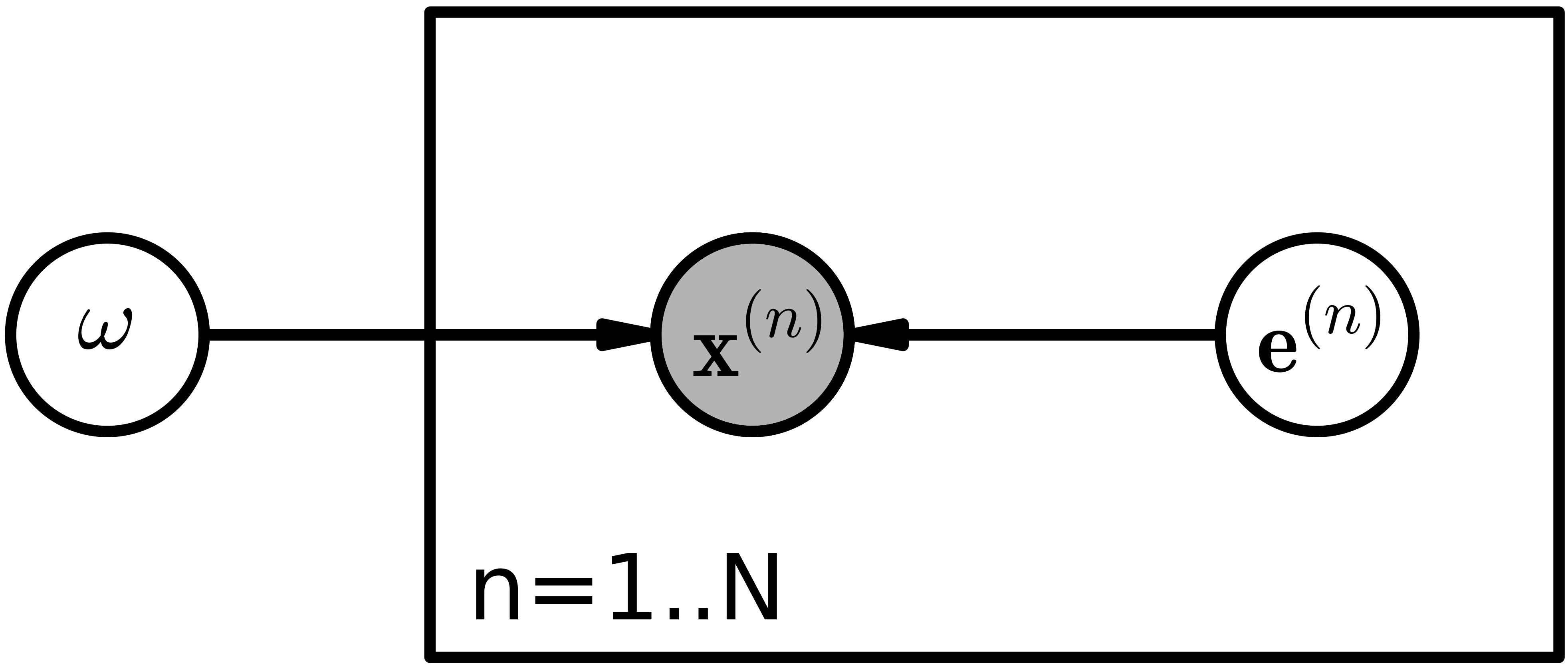
Acceleration law around the sun

$$a(r) = -A\left(\frac{r}{r_0}\right)^{-\alpha}$$

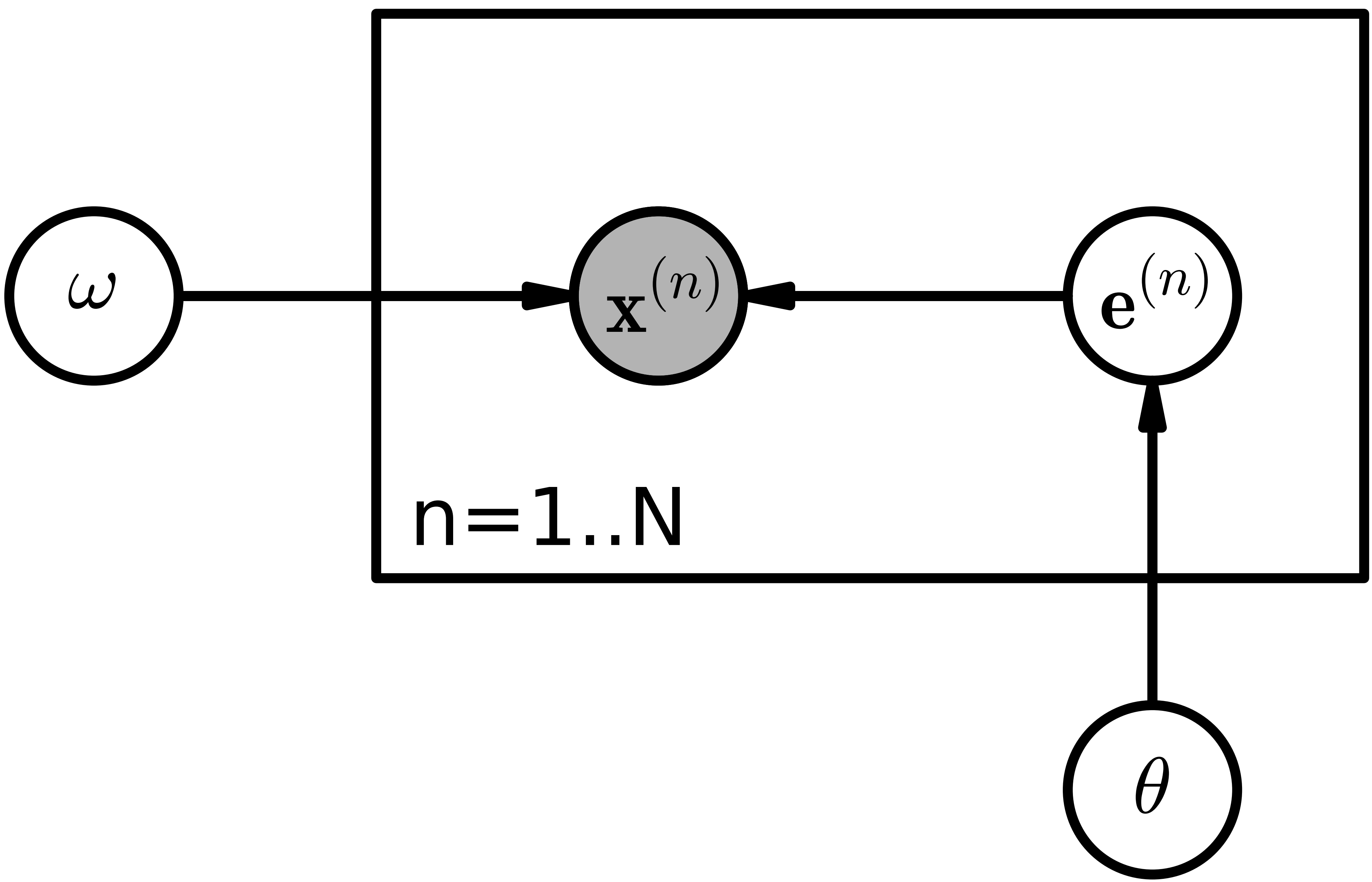
From a snapshot:

8 planet positions and velocities

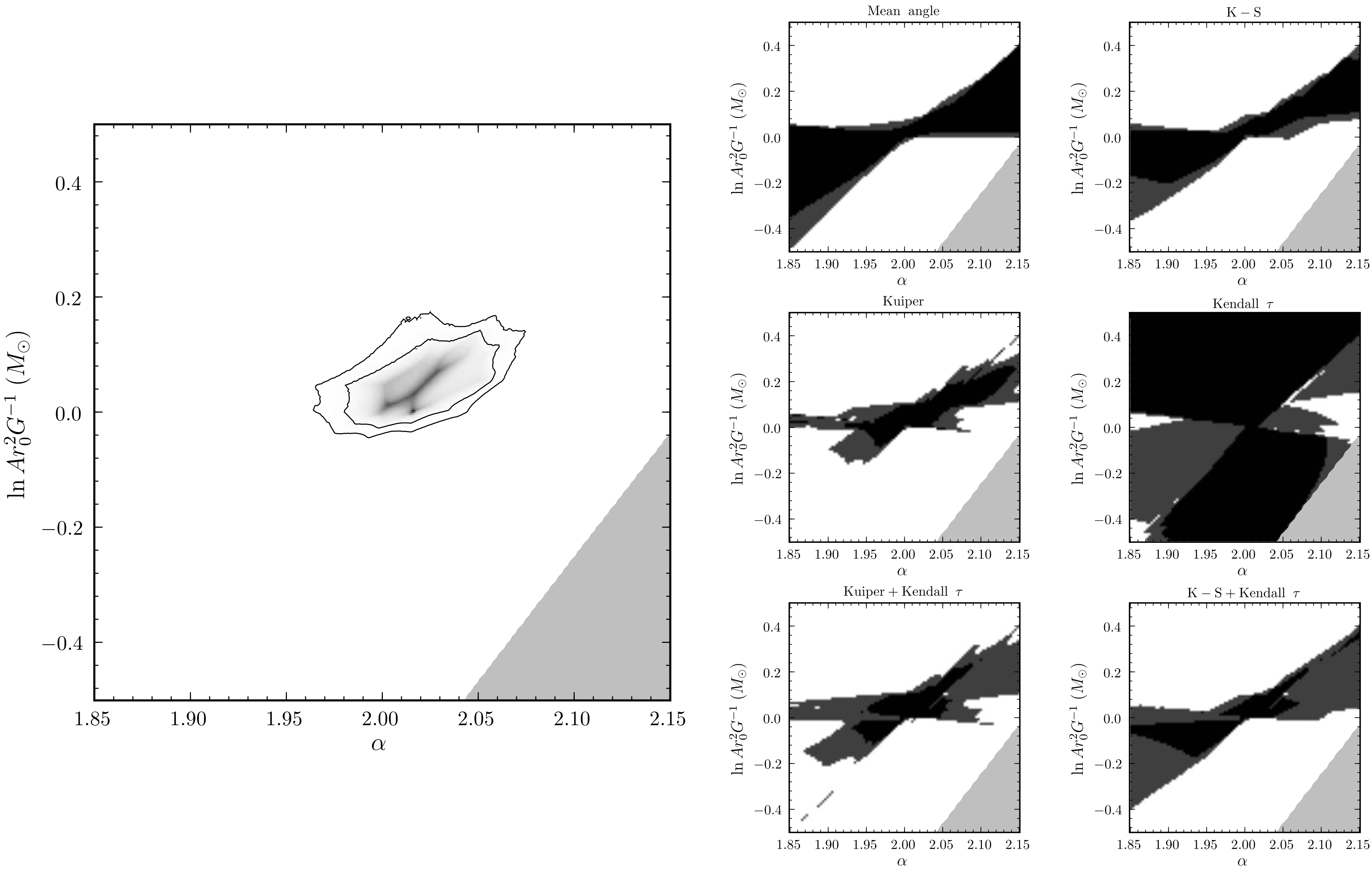
Graphical model



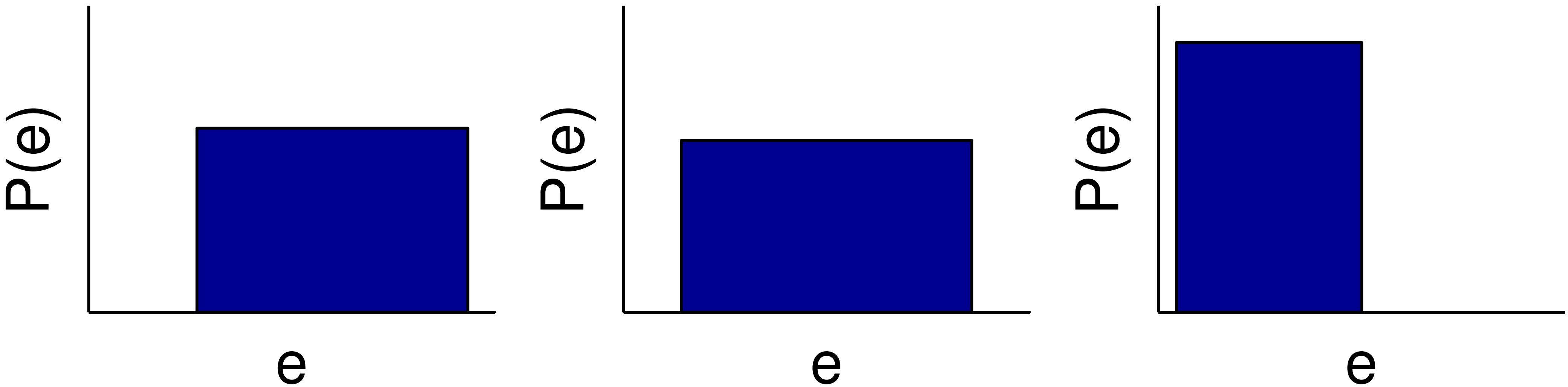
Hierarchical graphical model



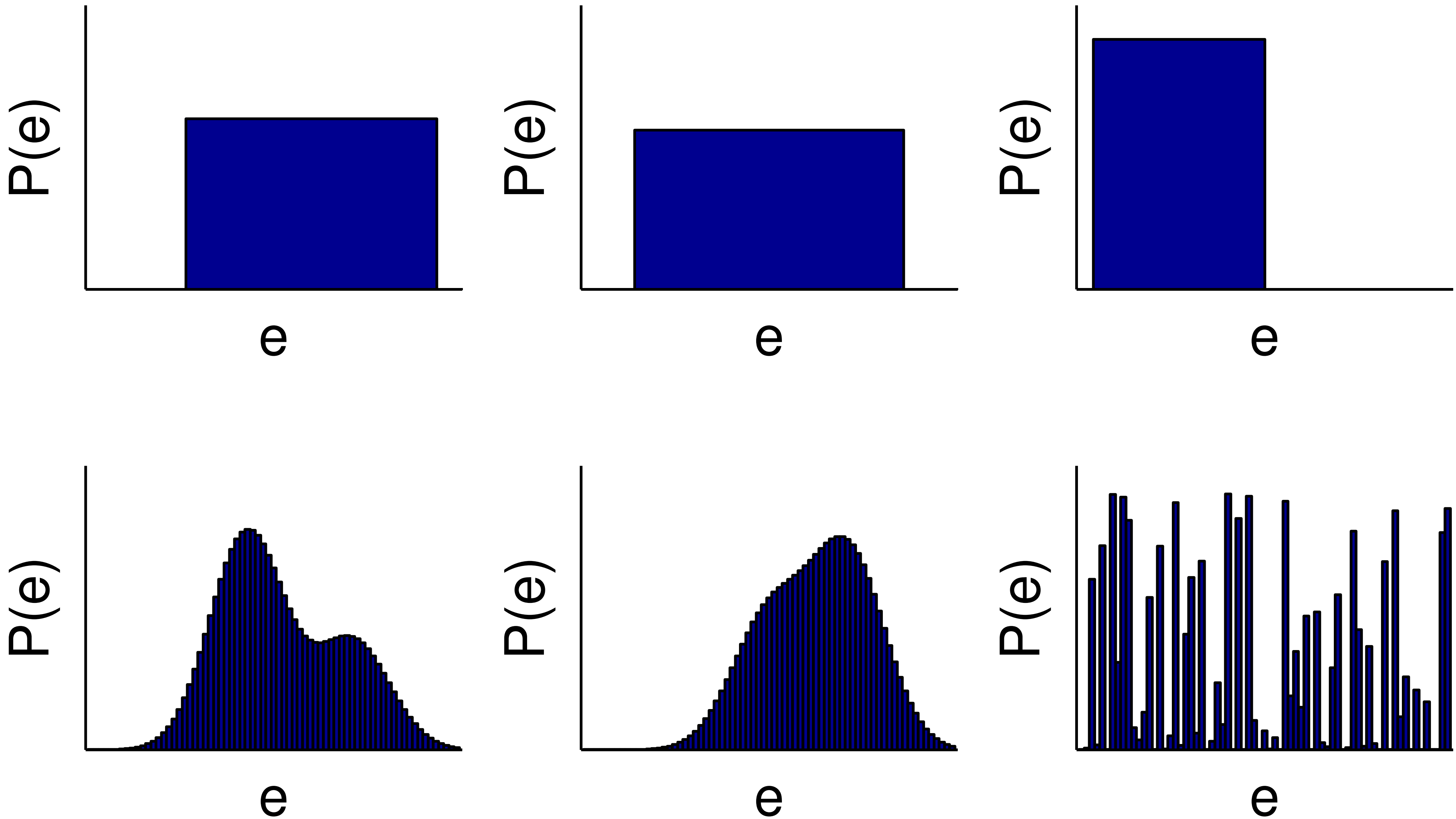
Inferences about the Sun



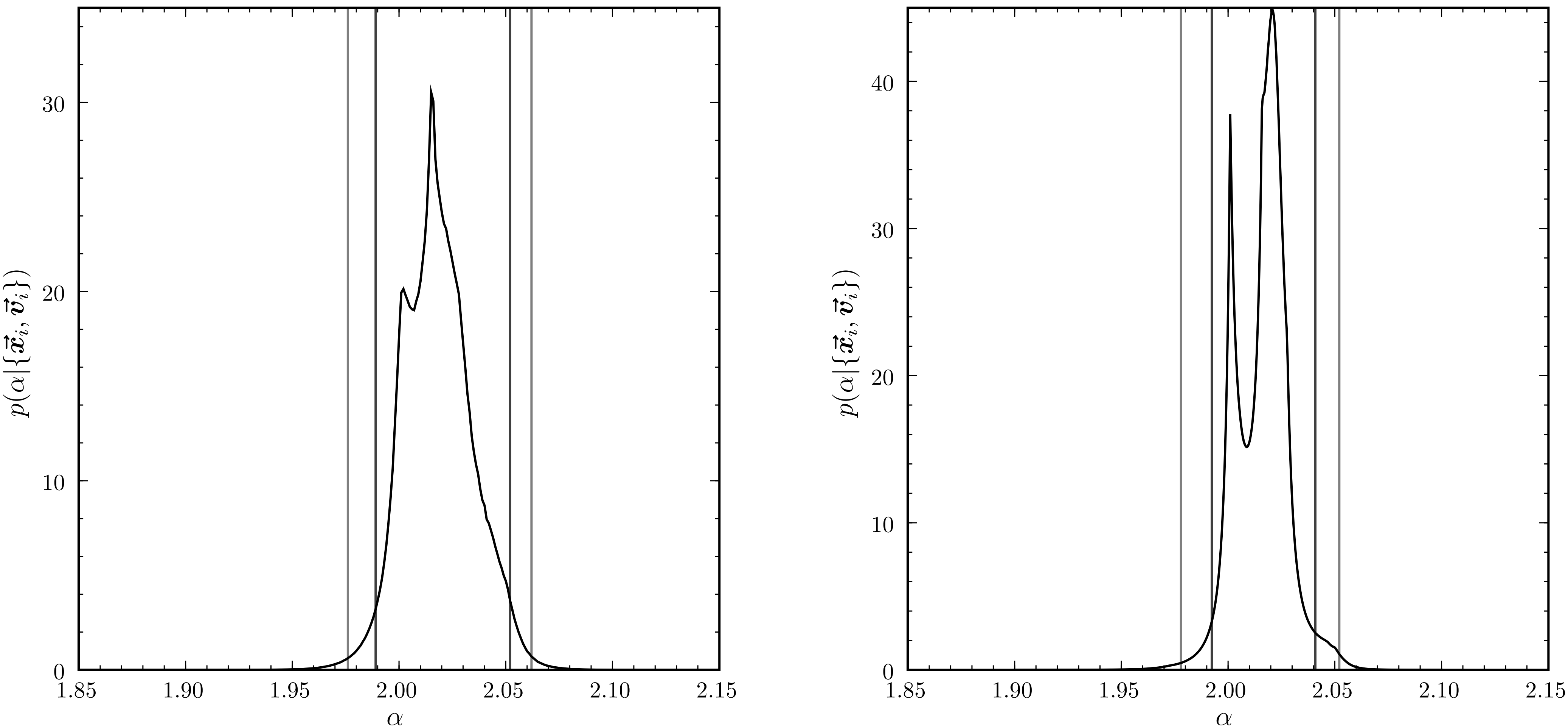
Priors on nuisance distributions

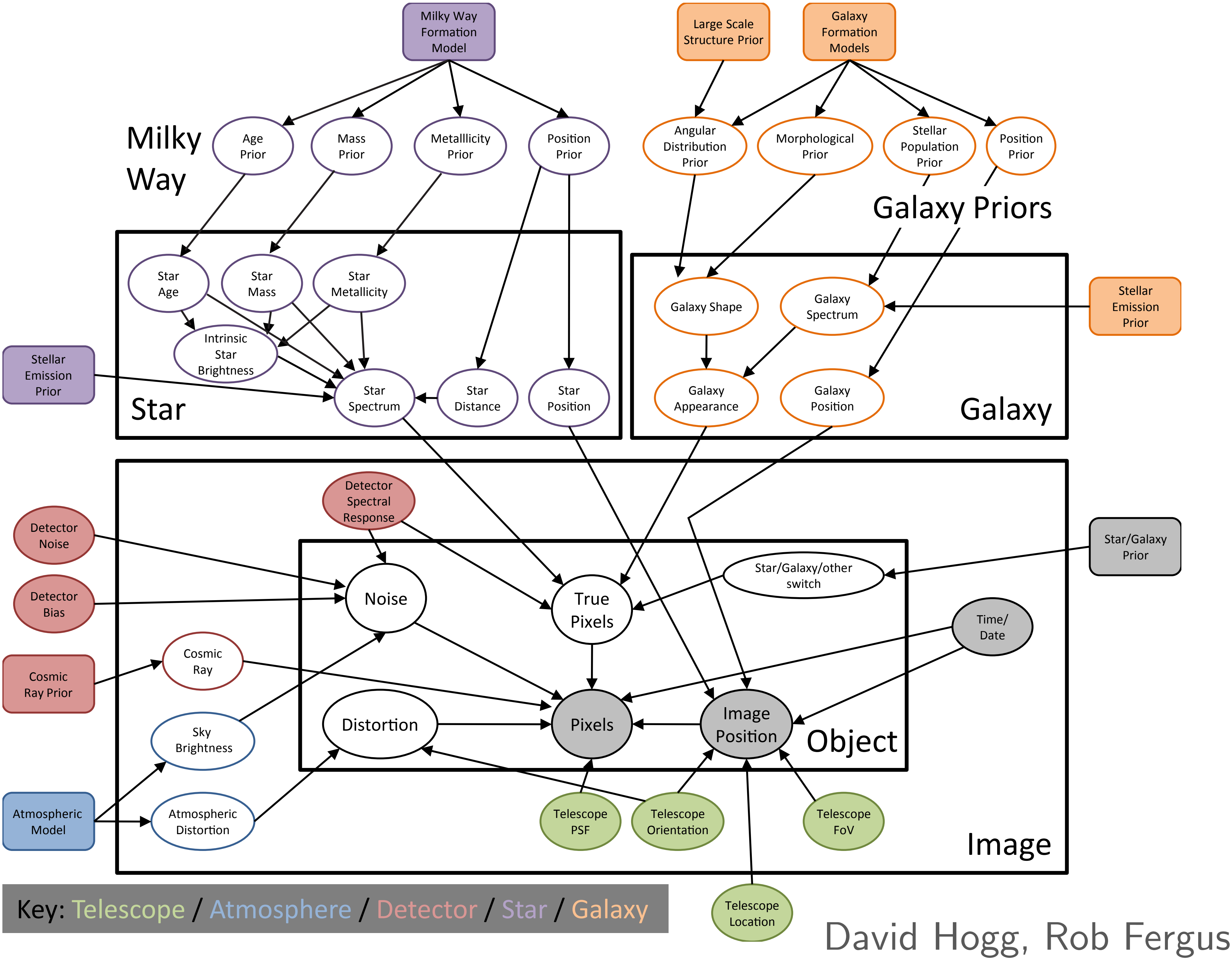


Priors on nuisance distributions



Gravitational exponent




Milky Way Formation Model
Large Scale Structure Prior
Galaxy Formation Models
Milky Way
Age Prior
Mass Prior
Metalllicity Prior
Position Prior
Angular Distribution Prior
Morphological Prior
Stellar Population Prior
Position Prior
Galaxy Priors
Star Age
Star Mass
Star Metallicity
Intrinsic Star Brightness
Stellar Emission Prior
Star Spectrum
Star Distance
Star Position
Star
Galaxy Shape
Galaxy Spectrum
Stellar Emission Prior
Galaxy Appearance
Galaxy Position
Galaxy
Detector Spectral Response
Detector Noise
Detector Bias
Noise
True Pixels
Star/Galaxy/other switch
Star/Galaxy Prior
Time/ Date
Cosmic Ray
Cosmic Ray Prior
Sky Brightness
Distortion
Pixels
Image Position
Object
Atmospheric Model
Atmospheric Distortion
Telescope PSF
Telescope Orientation
Telescope FoV
Image
Telescope Location
Key: Telescope / Atmosphere / Detector / Star / Galaxy
David Hogg, Rob Fergus

Machine Learning?

— **Density estimation**

Neural networks and Gaussian processes

— **Inference methods**

Statistical methods: MCMC, etc.

Learning: recognition networks

Representations: communicating results

Appendix Slides

Snapshot of the solar-system

Model for the sun: $\boldsymbol{\omega} = \{\log A, \alpha\}$

Acceleration law, $a(r) = -A\,[r/r_0]^{-\alpha}$

Model for each planet:

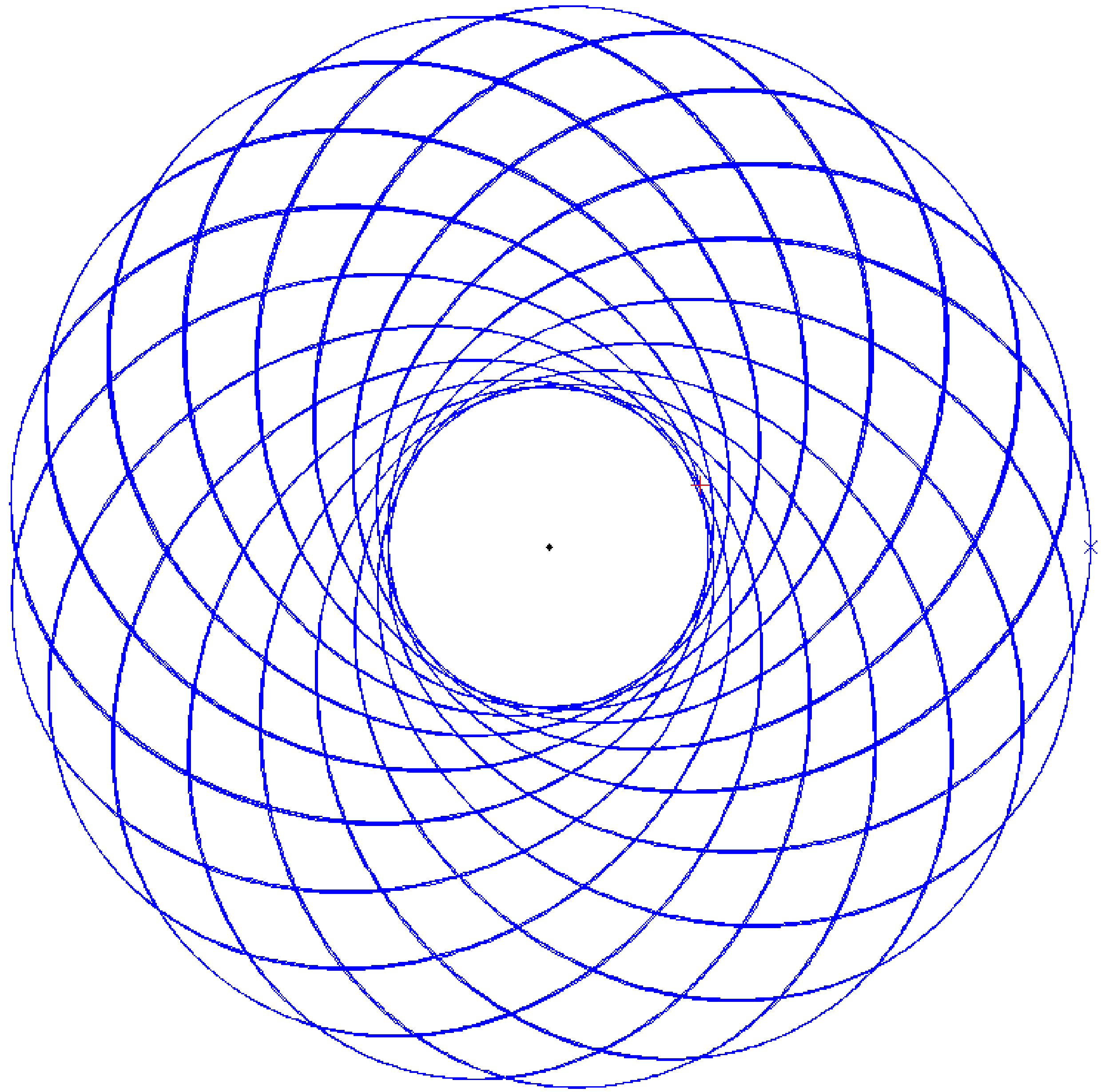
$\log \epsilon_n \sim p_\epsilon(\cdot|\theta_\epsilon)$ binding energy

$e_n \sim p(\cdot|\theta_e)$ radial asymmetry

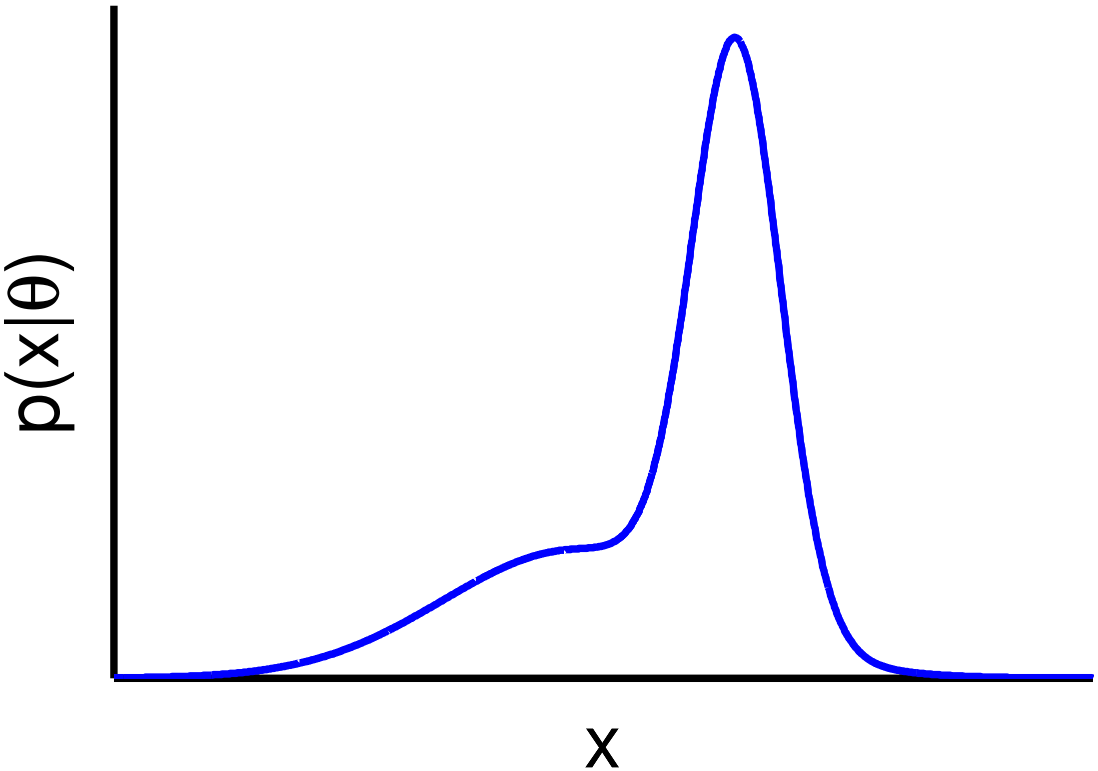
$f_n \sim \mathrm{Uniform}[0,1]$ fraction of time through orbit from aphelion (say), t/T_{orbit}

Observations in sky relate to:

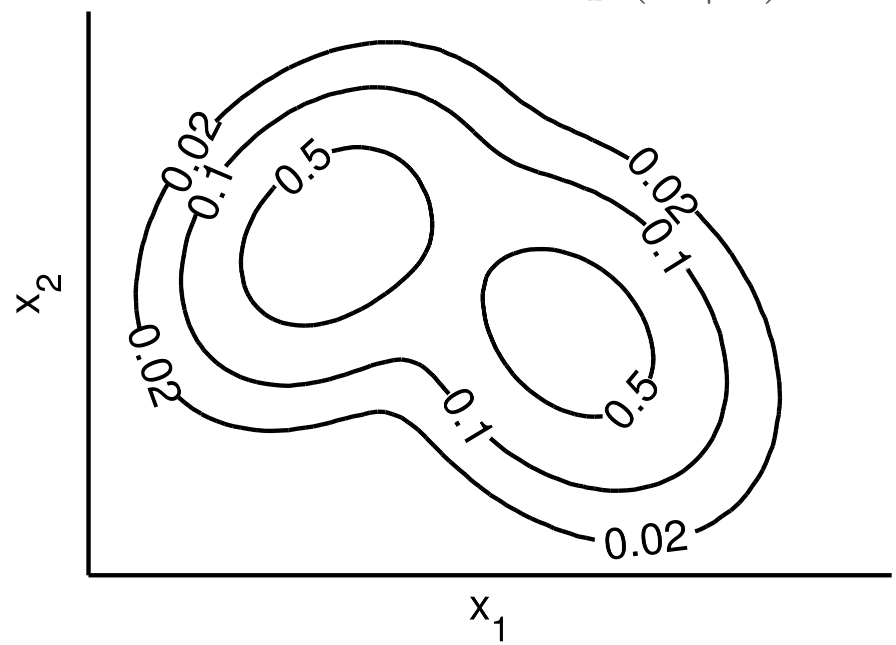
r, v_r, v_t: radial distance, radial velocity, transverse velocity

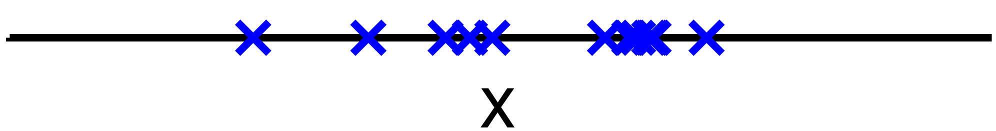


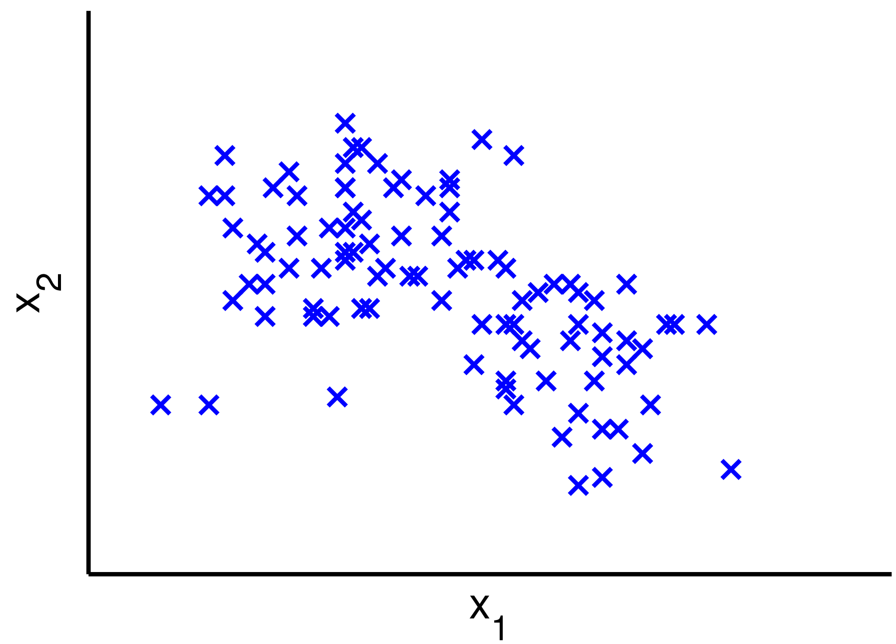
Density estimation



Quantiles of $p(\mathbf{x}\,|\,\theta)$







Task: $\{\mathbf{x}^{(n)}\} \rightarrow \theta$

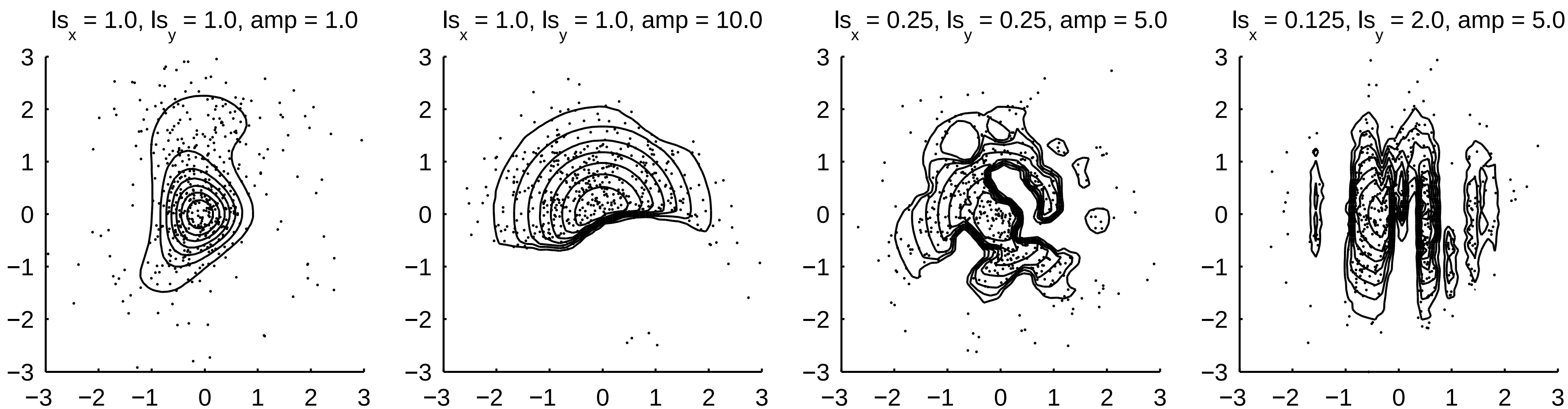
GP Density estimation

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

$\mathbf{f} \sim \mathcal{GP}$

$\Phi =$ sigmoidal function

$\pi =$ base measure



Gaussian Process Density Sampler

Adams, Murray and MacKay (2009).

Modelling via the Chain Rule

$$P(x_1)$$

$$P(x_2 \mid x_1)$$

$$P(x_3 \mid x_1, x_2)$$

$$P(x_4 \mid x_1, x_2, x_3)$$

$$P(\mathbf{x}) = P(x_1) \prod_{k=2}^{K} P(x_k \mid \mathbf{x}_{<k})$$

Mixture Density Networks

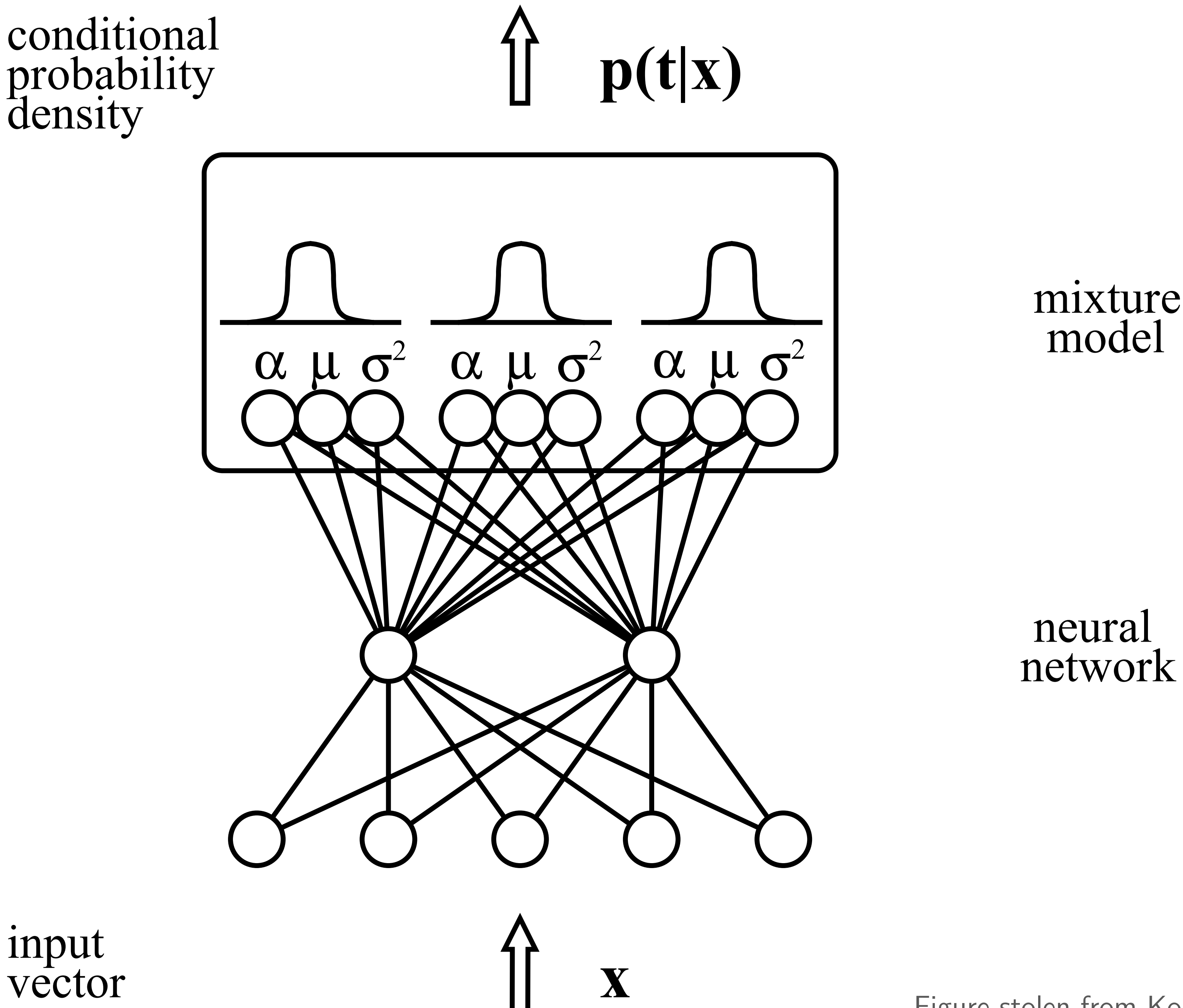
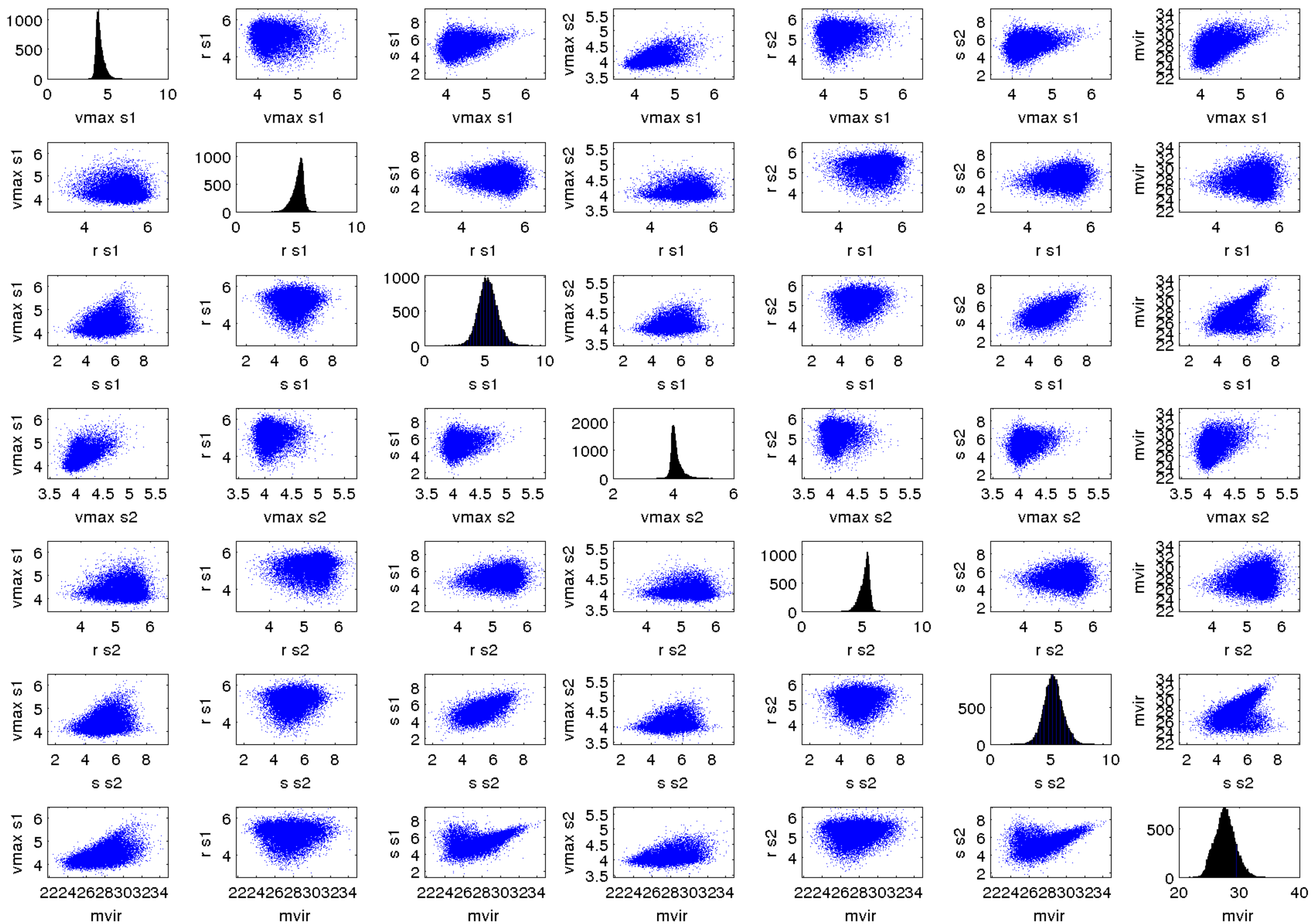
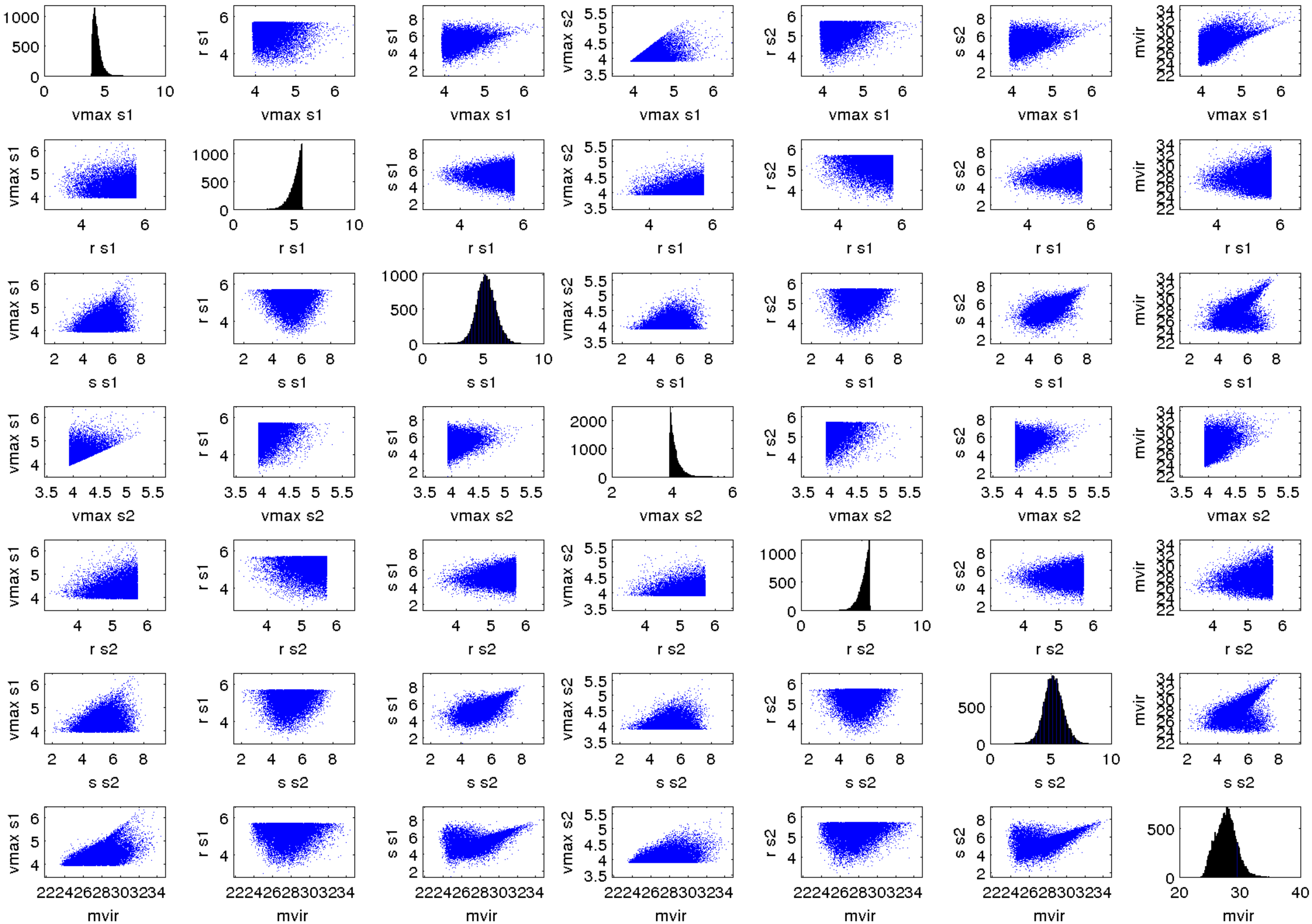


Figure stolen from Korin Richmond

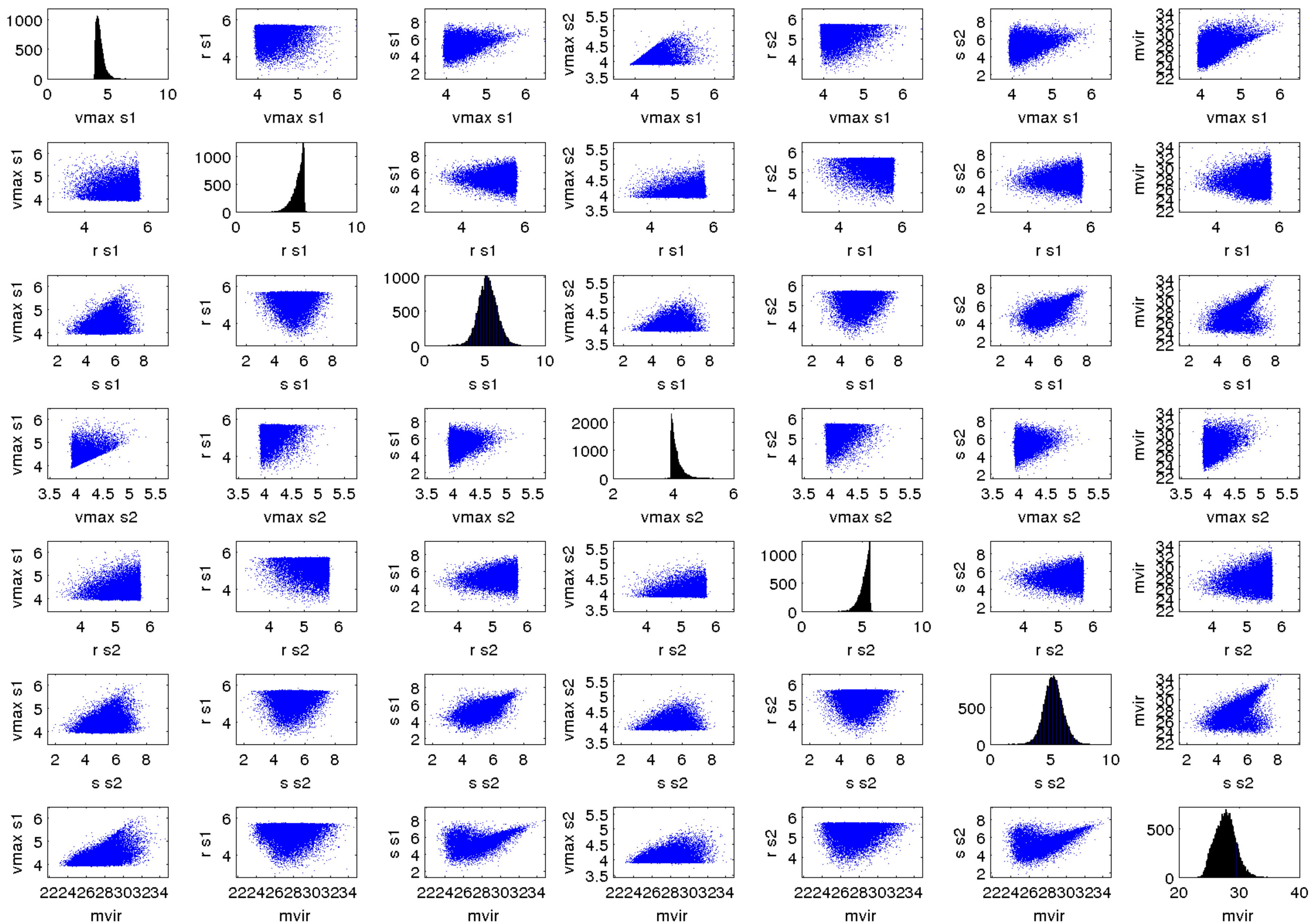
Mixture of Gaussian samples



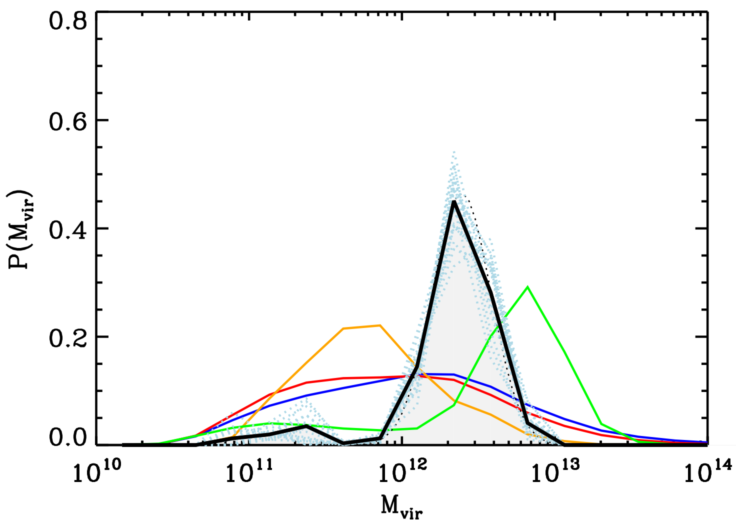
Simulation samples

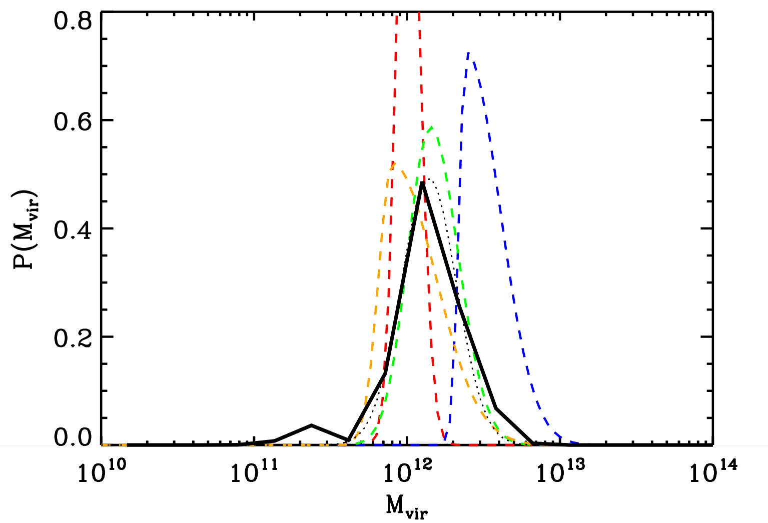


AMDN samples

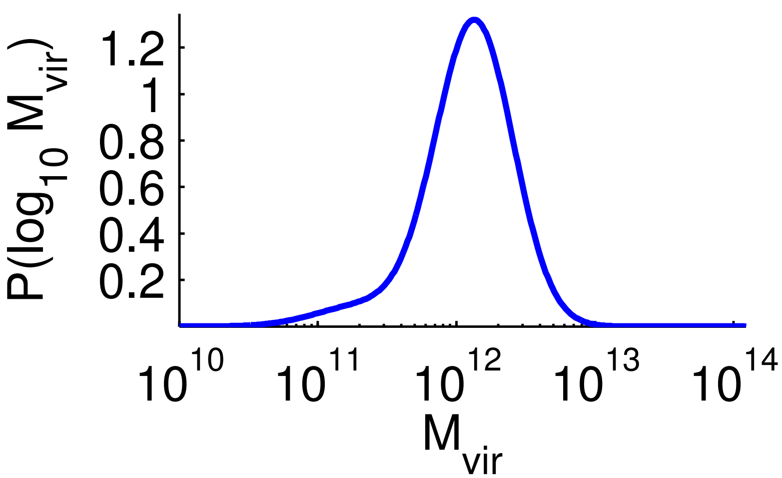


Results of inference

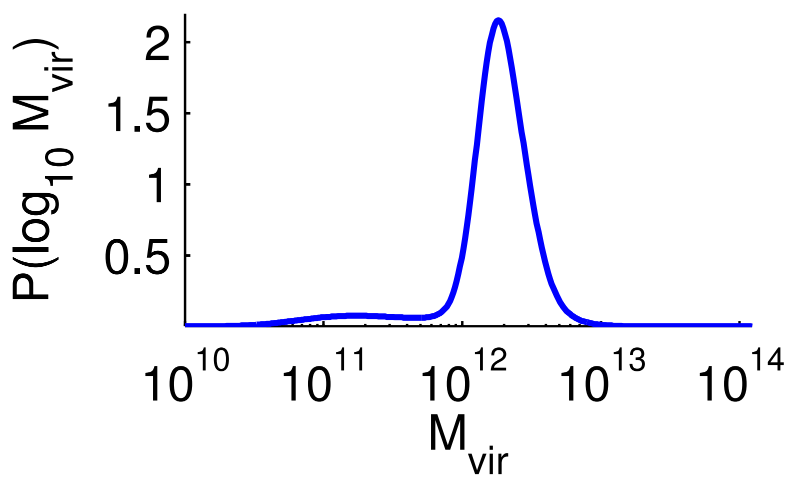




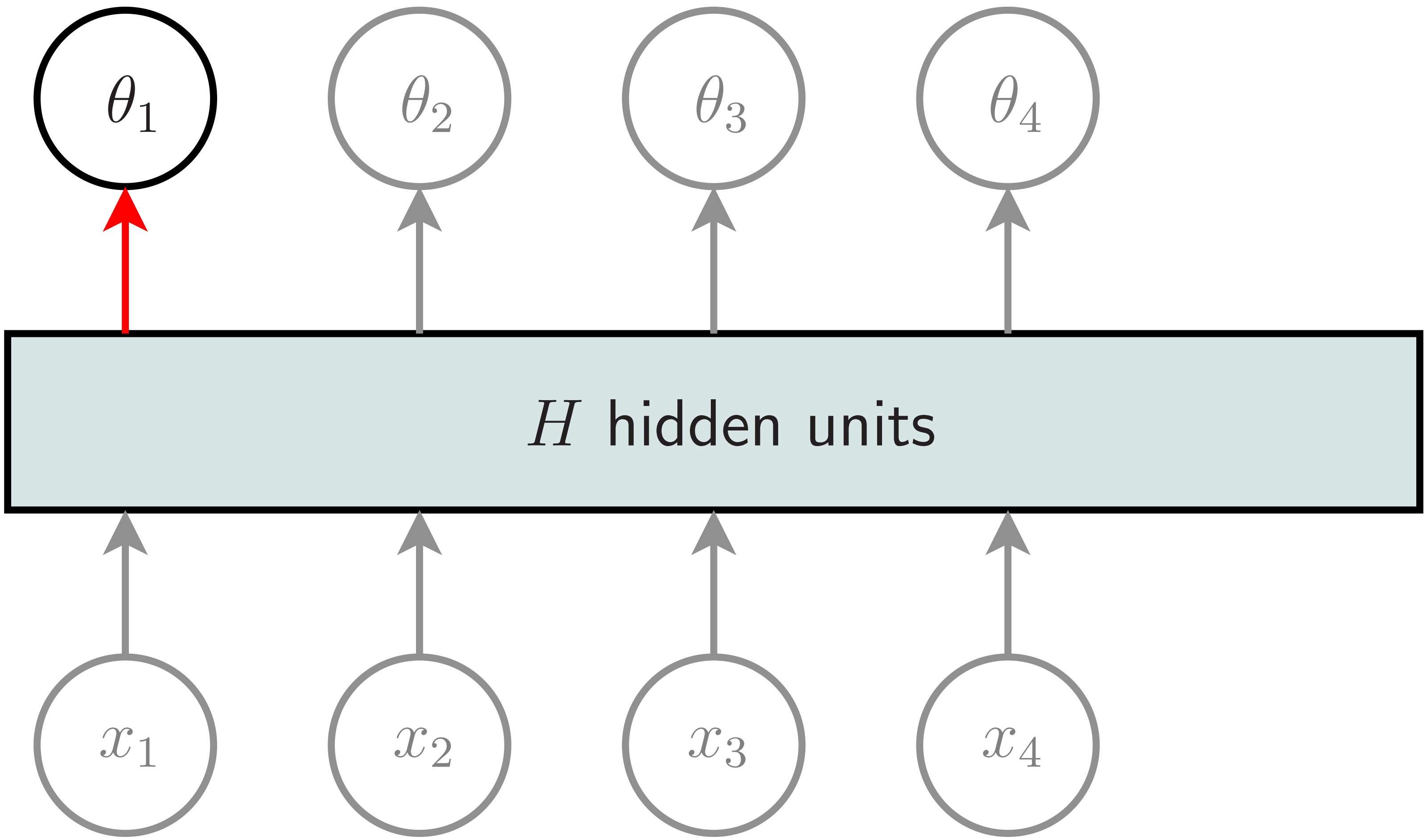
Prior modeled with MoG



Prior modeled with AMDN

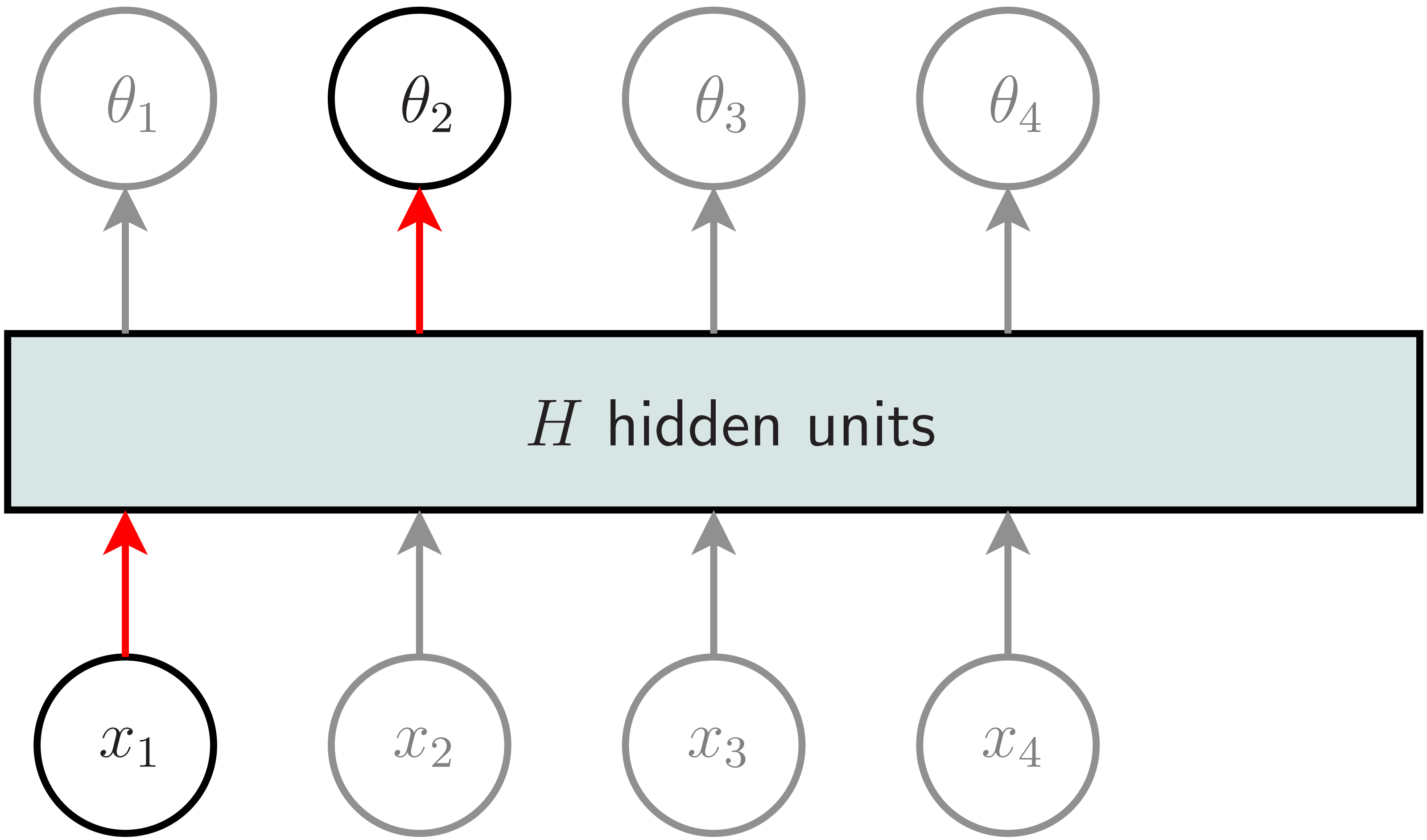


Sequential activation



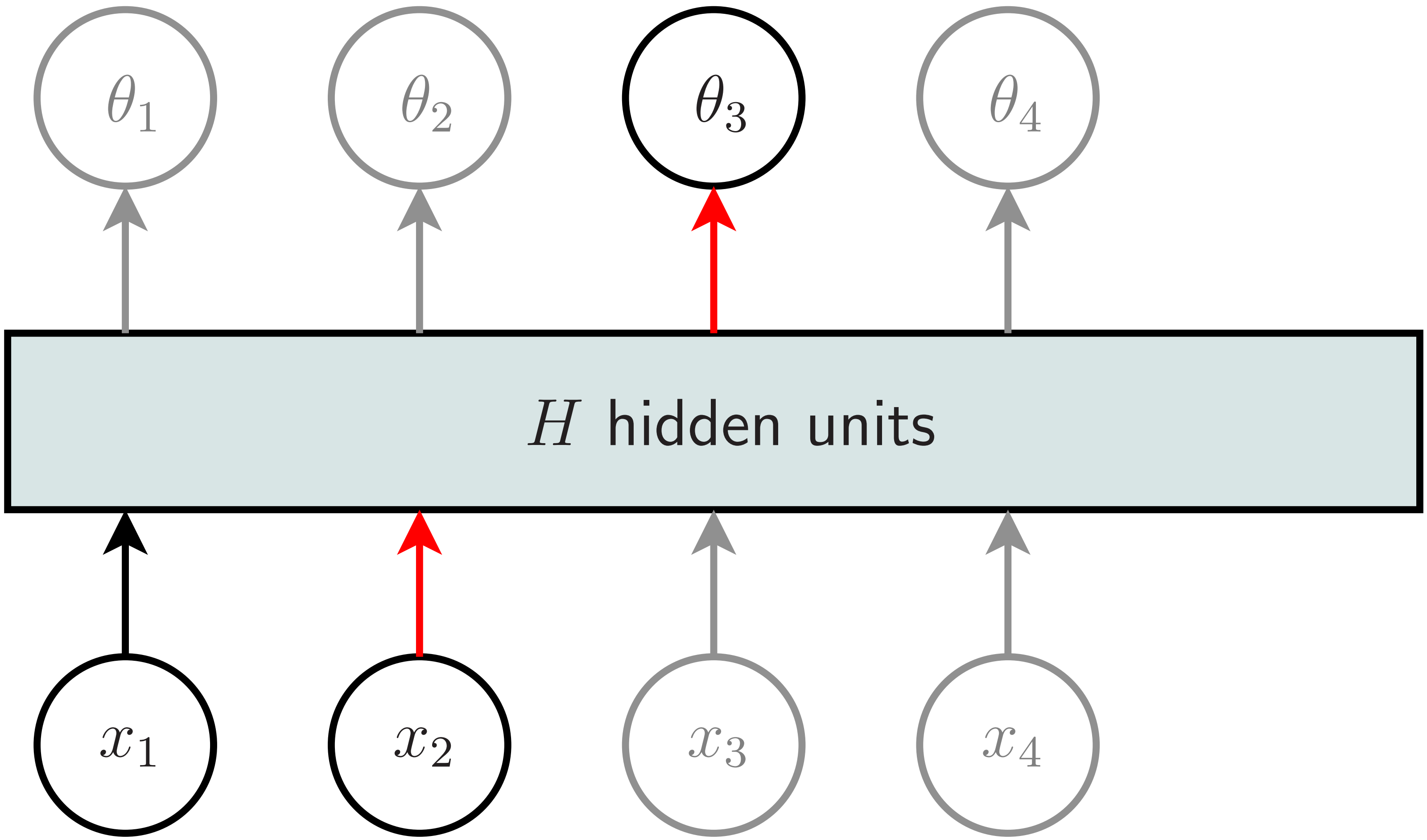
$$P(\mathbf{x}) = P(x_1 \,|\, \theta_1) \ldots$$

Sequential activation



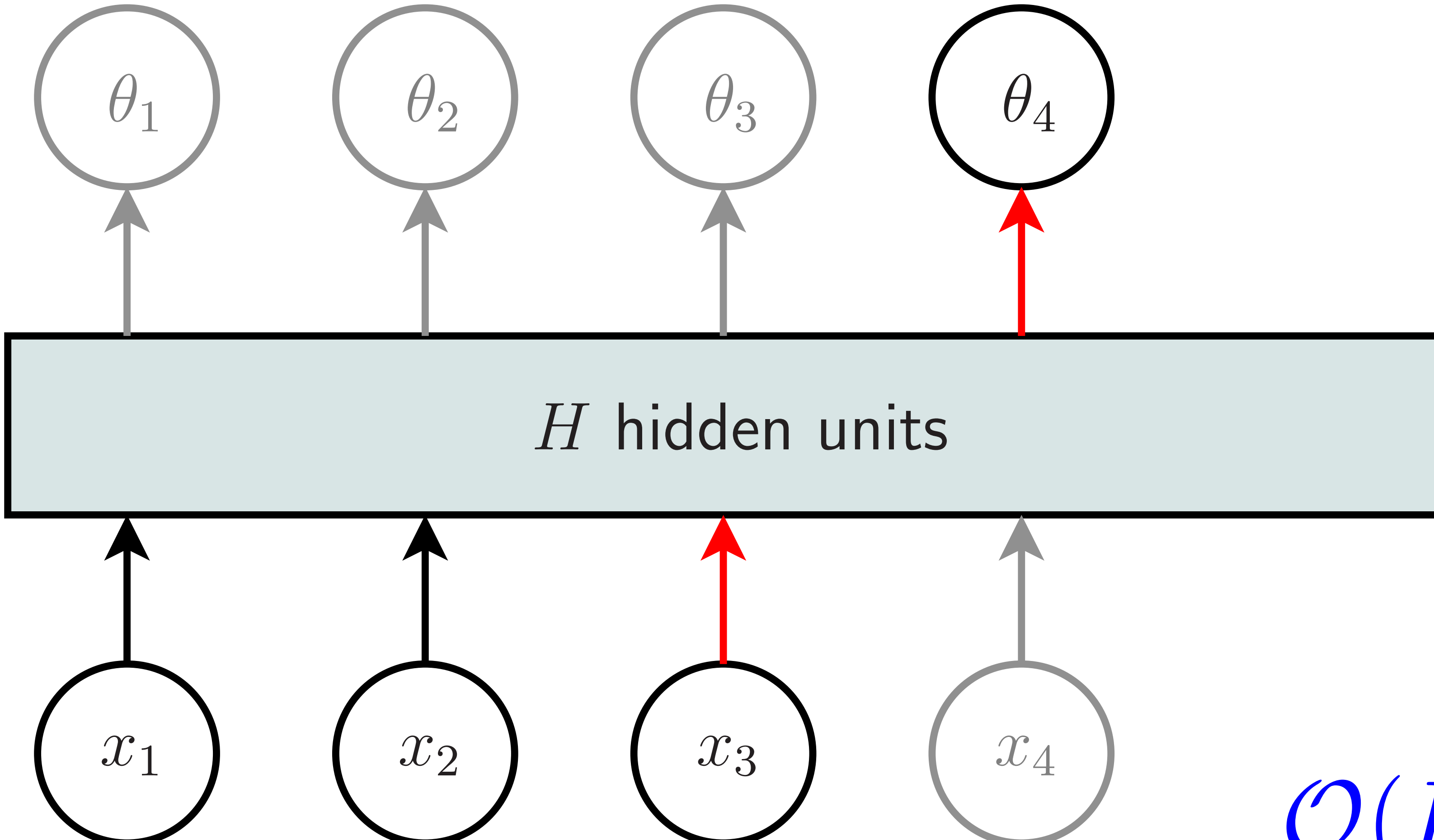
$$P(\mathbf{x}) = P(x_1 \,|\, \theta_1)\, P(x_2 \,|\, \theta_2(x_1)) \ldots$$

Sequential activation



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

Sequential activation



$\mathcal{O}(DH)$

$$P(\mathbf{x}) = P(x_1 \,|\, \theta_1)\, P(x_2 \,|\, \theta_2(x_1))\, P(x_3 \,|\, \theta_3(x_1, x_2))\, P(x_4 \,|\, \theta_4(x_1, x_2, x_3)) \ldots$$

NADE results

Model	ADULT	CONNECT-4	DNA	MUSHROOMS	NIPS-0-12	OCR-LETTERS	RCV1	WEB
MoB	0.00 ± 0.10	0.00 ± 0.04	0.00 ± 0.53	0.00 ± 0.10	0.00 ± 1.12	0.00 ± 0.32	0.00 ± 0.11	0.00 ± 0.23
RBM	4.18 ± 0.06	0.75 ± 0.02	1.29 ± 0.48	-0.69 ± 0.09	12.65 ± 1.07	-2.49 ± 0.30	-1.29 ± 0.11	0.78 ± 0.20
RBM mult.	4.15 ± 0.06	-1.72 ± 0.03	1.45 ± 0.40	-0.69 ± 0.05	11.25 ± 1.06	0.99 ± 0.29	-0.04 ± 0.11	0.02 ± 0.21
RBForest	4.12 ± 0.06	0.59 ± 0.02	1.39 ± 0.49	0.04 ± 0.07	12.61 ± 1.07	3.78 ± 0.28	0.56 ± 0.11	-0.15 ± 0.21
FVSBN	**7.27 ± 0.04**	11.02 ± 0.01	**14.55 ± 0.50**	4.19 ± 0.05	13.14 ± 0.98	1.26 ± 0.23	-2.24 ± 0.11	0.81 ± 0.20
NADE	**7.25 ± 0.05**	**11.42 ± 0.01**	13.38 ± 0.57	**4.65 ± 0.04**	**16.94 ± 1.11**	**13.34 ± 0.21**	**0.93 ± 0.11**	**1.77 ± 0.20**
Normalization	-20.44	-23.41	-98.19	-14.46	-290.02	-40.56	-47.59	-30.16

★ Little variation when changing input ordering:

DNA = **+/- 0.05**

MUSHROOMS = **+/- 0.045**

NIPS-0-12 = **+/- 0.15**

NADE results

Intractable

Model	Log. Like.
MoB*	-137.64
RBM (CD1)*	≈-125.53
RBM (CD3)*	≈-105.50
RBM (CD25)*	≈-86.34
FVSBN	-97.45
NADE	-88.86

Samples

RNADE results

Dataset	dim	size	Gaussian	MFA	FVBN	RNADE-MoG	RNADE-MoL
Red wine	11	1599	−13.18	−10.19	−11.03	**−9.36**	**−9.46**
White wine	11	4898	−13.20	−10.73	−10.52	**−10.23**	−10.38
Parkinsons	15	5875	−10.85	−1.99	**−0.71**	**−0.90**	−2.63
Ionosphere	32	351	−41.24	−17.55	−26.55	**−2.50**	**−5.87**
Boston housing	10	506	−11.37	−4.54	**−3.41**	**−0.64**	−4.04

RNADE results

Model	Training LogL	Test LogL
MoG $N=50$	111.6	110.4
MoG $N=100$	113.4	112.0
MoG $N=200$	113.9	112.5
MoG $N=300$	114.1	112.5
RNADE-MoG $K=10$	125.9	123.9
RNADE-MoG $K=20$	126.7	**124.5**
RNADE-MoL $K=10$	120.3	118.0
RNADE-MoL $K=20$	122.2	119.8

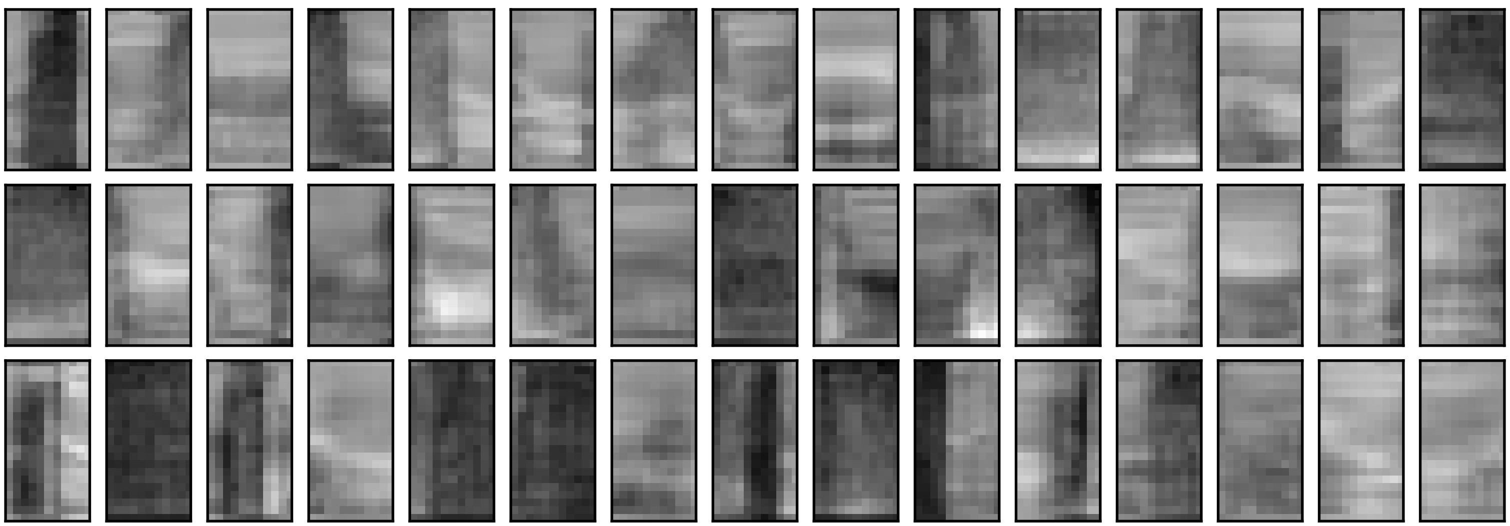


Figure 2: **Top:** 15 datapoints from the TIMIT core-test set. **Center:** 15 samples from a MoG model with 200 components. **Bottom:** 15 samples from an RNADE with 1024 hidden units and output components per dimension. On each plot, time is shown on the horizontal axis, the bottom row displays the energy feature, while the others display the filter bank features (in ascending frequency order from the bottom). All data and samples were drawn randomly.

Deep learning?

One motivation

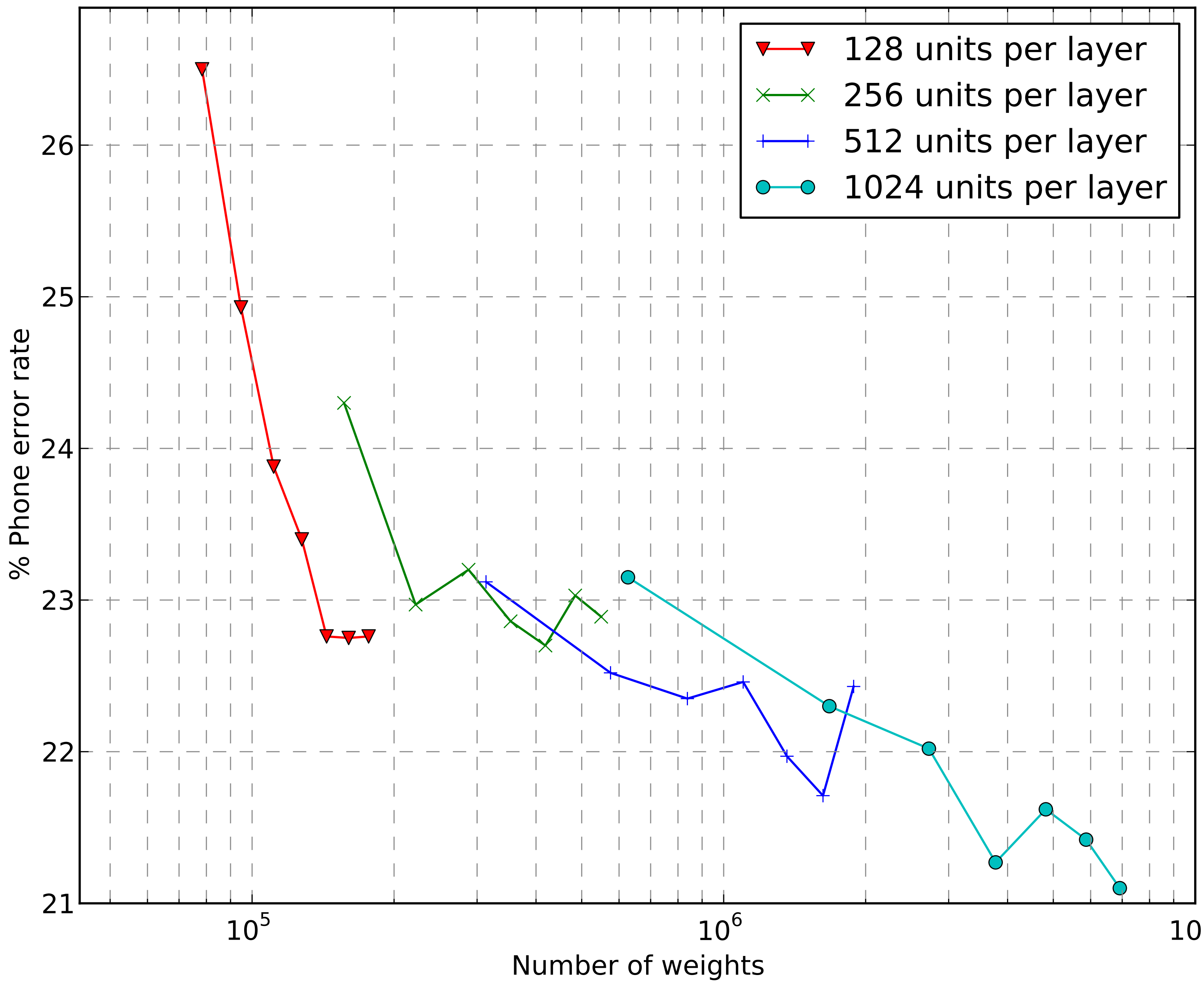
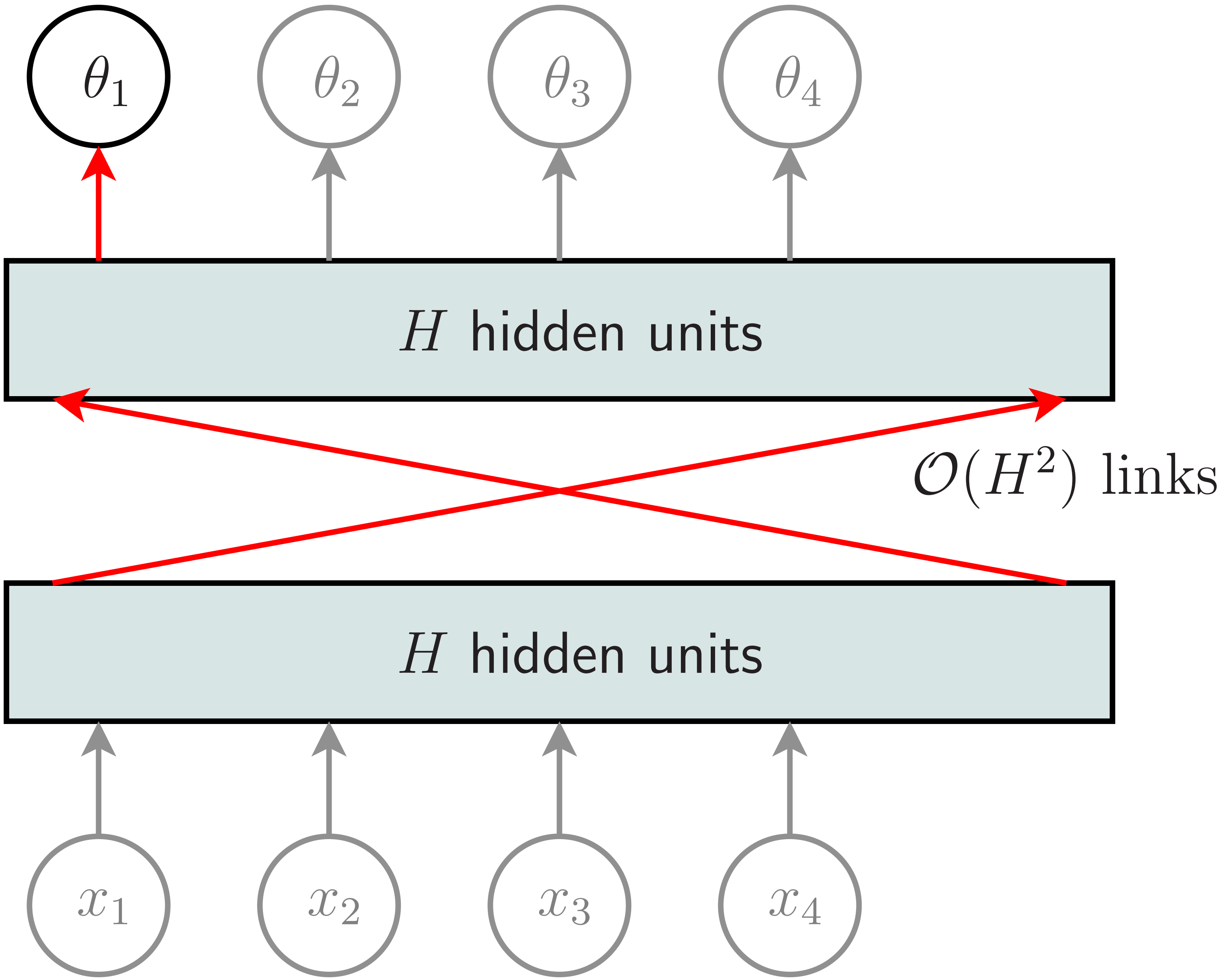
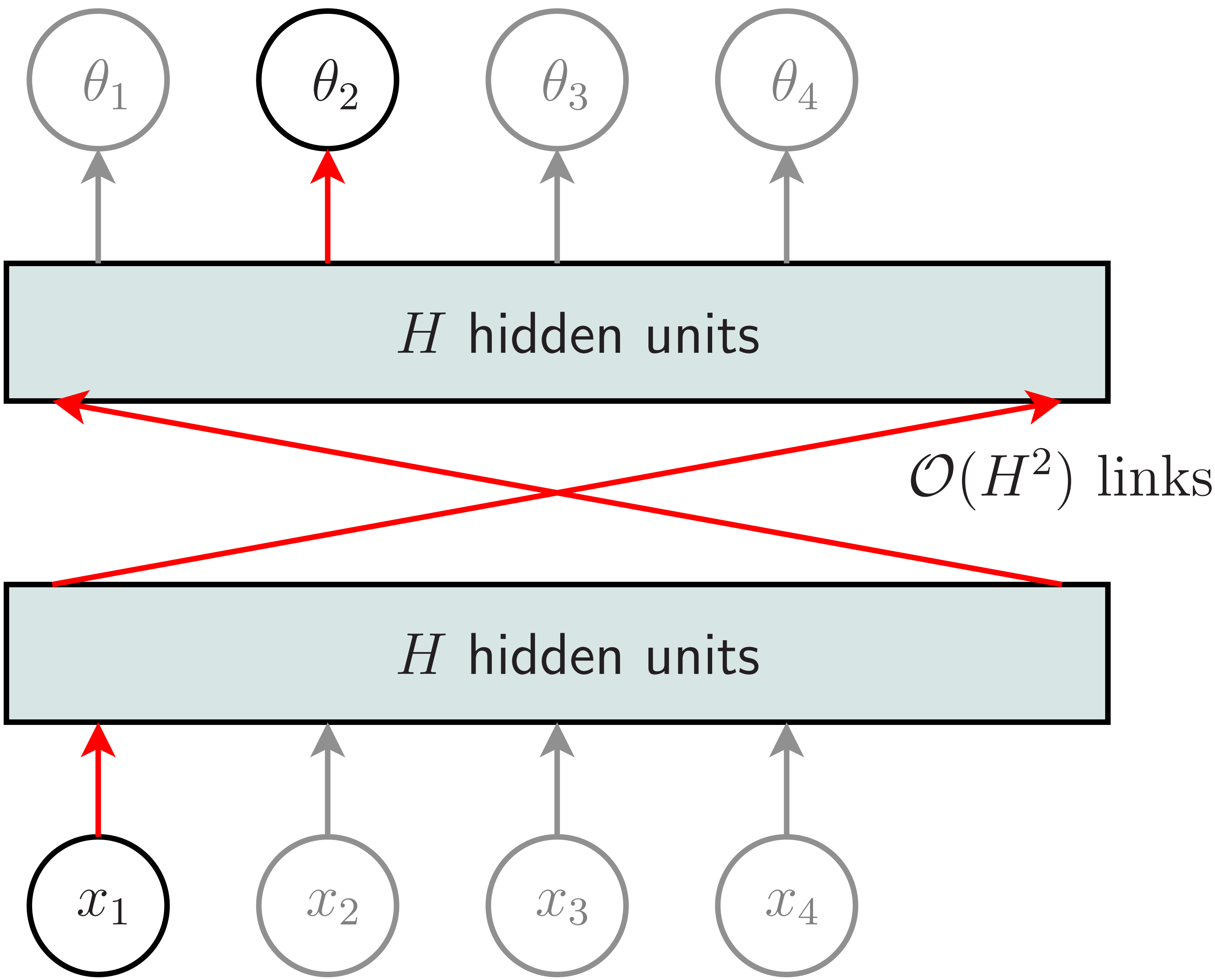


Figure by Benigno Uria

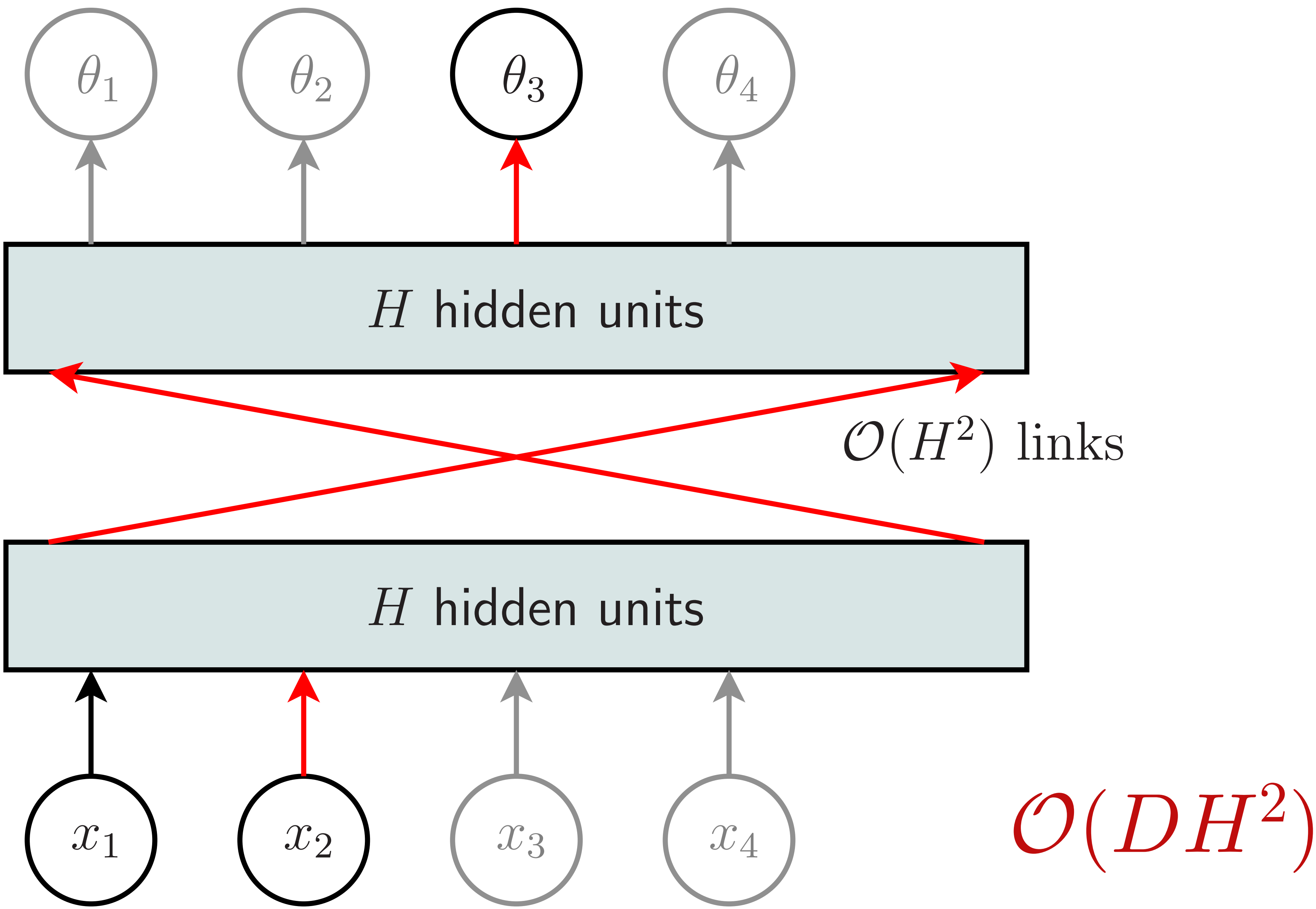
Sequential deep activation



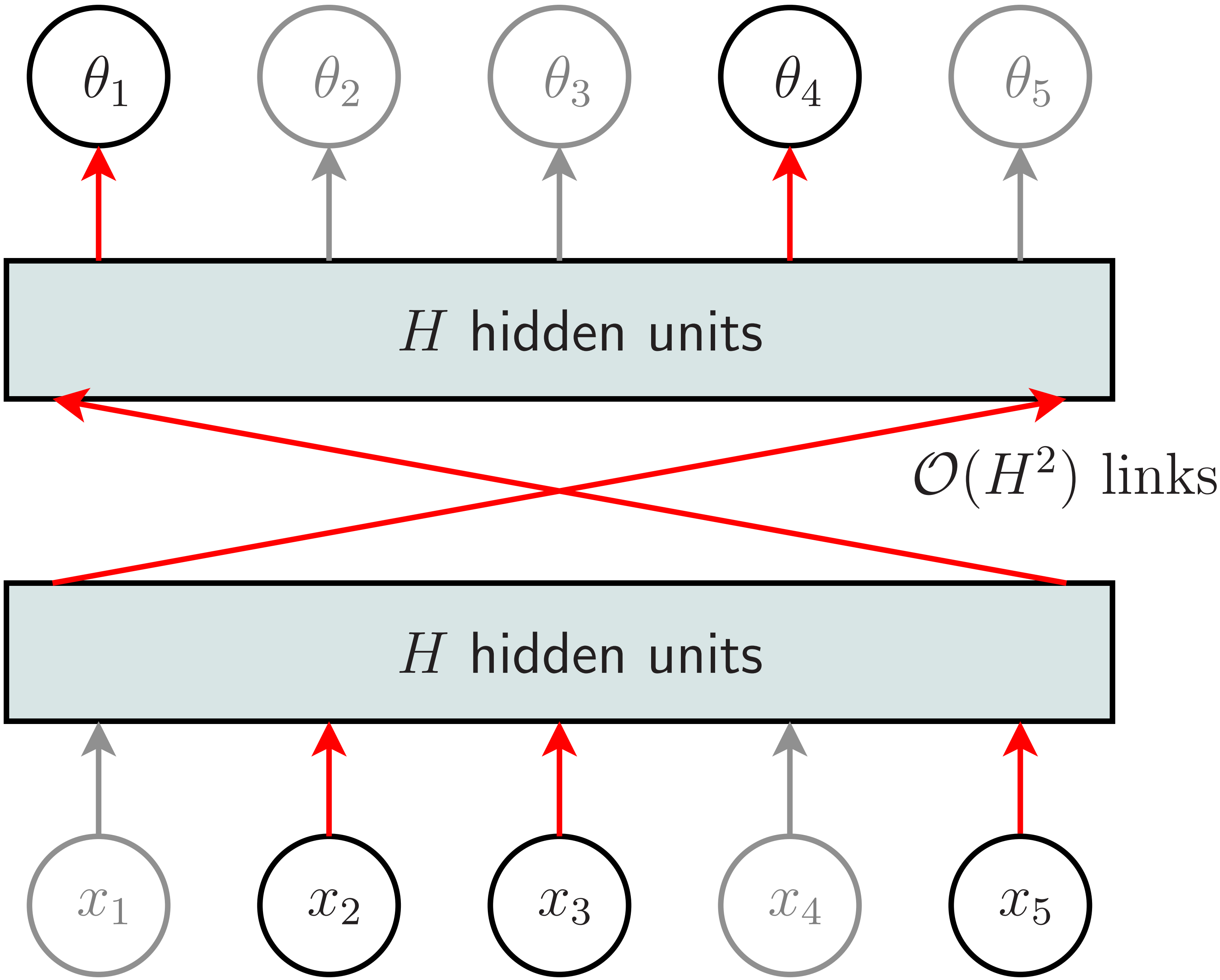
Sequential deep activation



Sequential deep activation



A completing machine



Deep NADE

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

Test time: predict features in any order

(can condition on observations)

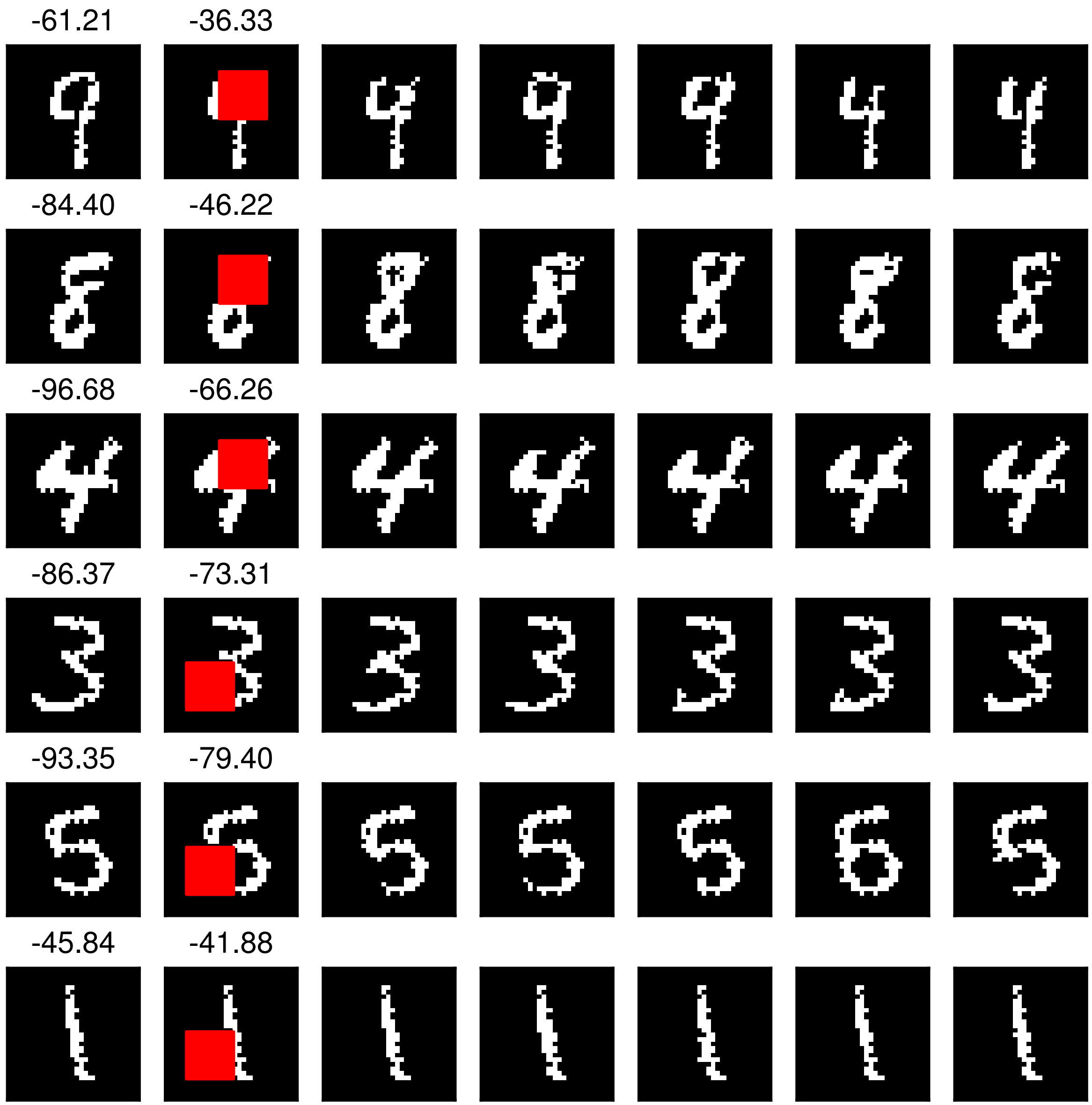
Different orderings not consistent:

— Seems bad, but. . .

— have trained large ensemble

— combining different orderings works better

Arbitrary ordering: inpainting



Deep ensembles — results

Small improvements across most UCI datasets

Finally beat MoG on image patches:

Table 4. Average test-set log-likelihood for several models trained on 8 by 8 pixel patches of natural images taken from the BSDS300 dataset. Note that because these are log probability densities they are positive, higher is better.

Model	Test LogL
MoG $K = 200$ (Zoran & Weiss, 2012)	152.8
RNADE 1hl (fixed order)	152.1
RNADE 1hl	143.2
RNADE 2hl	149.2
RNADE 3hl	152.0
RNADE 4hl	153.6
RNADE 5hl	154.7
RNADE 6hl	**155.2**
EoRNADE 6hl 2 ord.	156.0
EoRNADE 6hl 32 ord.	**157.0**

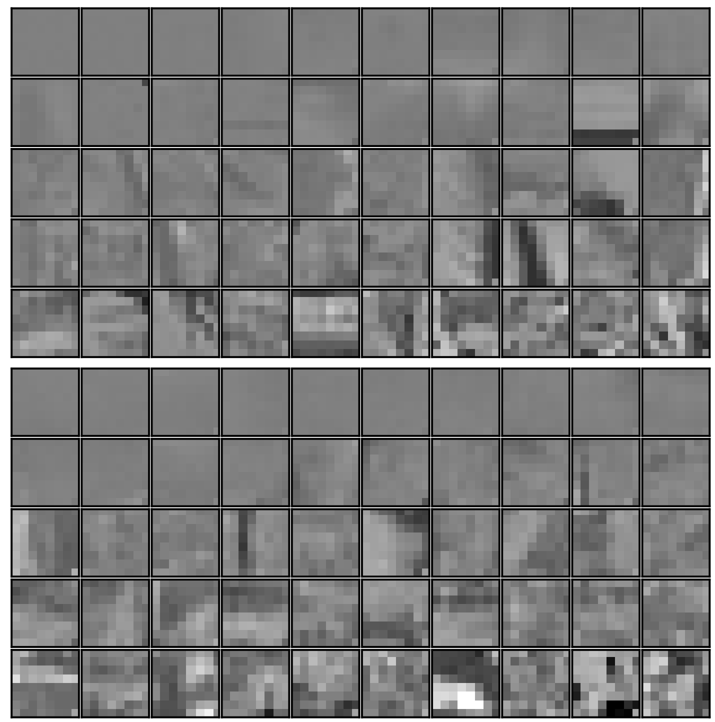


Figure 5. **Top:** 50 examples of 8×8 patches in the BSDS300 dataset ordered by decreasing likelihood under a 6-hidden-layer NADE. **Bottom:** 50 samples from a 6-hidden-layer NADE.

ever reported on this task. Ensembles of RNADEs also show