Density estimation

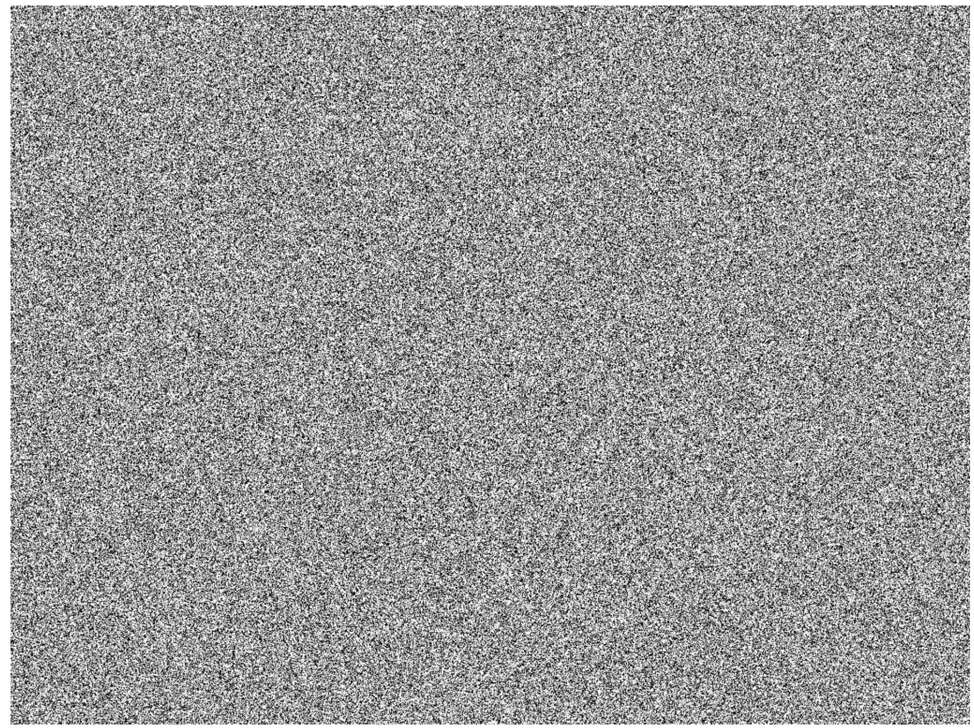
Computing, and avoiding, partition functions

Roadmap:
— Motivation: density estimation
— Understanding annealing/tempering
— NADE

Iain Murray
School of Informatics, University of Edinburgh
Includes work with **Ruslan Salakhutdinov** and **Hugo Larochelle**

Probabilistic model $\mathcal{H}$

Predict new images: $P(\mathbf{x} \,|\, \mathcal{H})$



High density can be boring

$$P(\mathbf{x} \,|\, \mathcal{H})$$

Image reconstruction

Observation model: $P(\mathbf{y} \,|\, \mathbf{x})$

Underlying image:

$$P(\mathbf{x} \,|\, \mathbf{y}) \propto P(\mathbf{y} \,|\, \mathbf{x}) \, P(\mathbf{x})$$

(e.g., Zoran and Weiss, 2011; Lucas Theis's work)

Roadmap

— Unsupervised learning and $P(\mathbf{x} \,|\, \mathcal{H})$

— **Evaluating** $P(\mathbf{x} \,|\, \mathcal{H})$

Salakhutdinov and Murray (2008)

Murray and Salakhutdinov (2009)

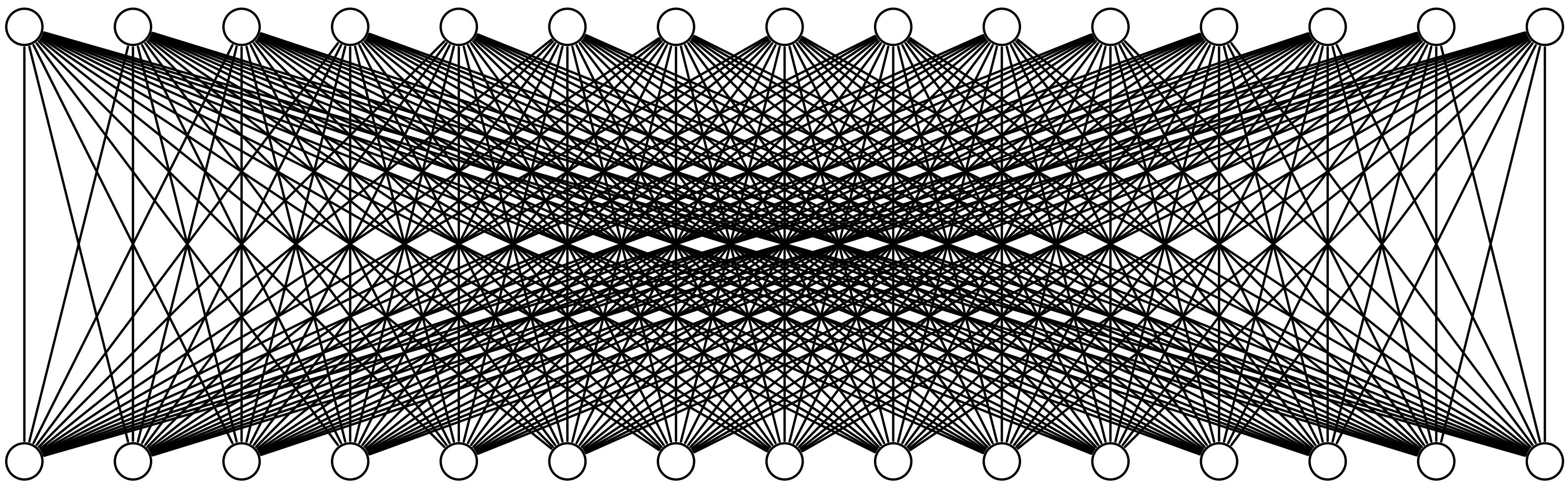
Wallach, Murray, Salakhutdinov & Mimno (2009)

— NADE: "density estimation put first"

Larochelle and Murray (2011)

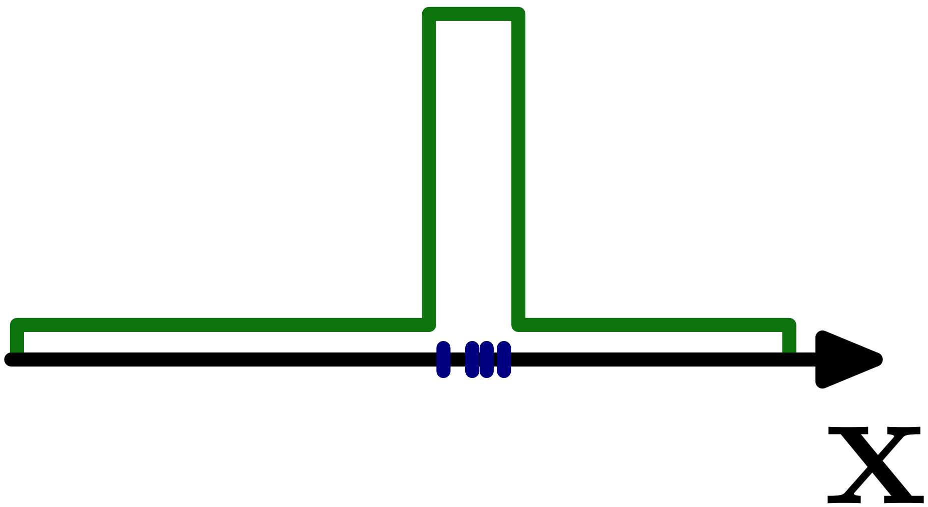
Restricted Boltzmann Machines

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



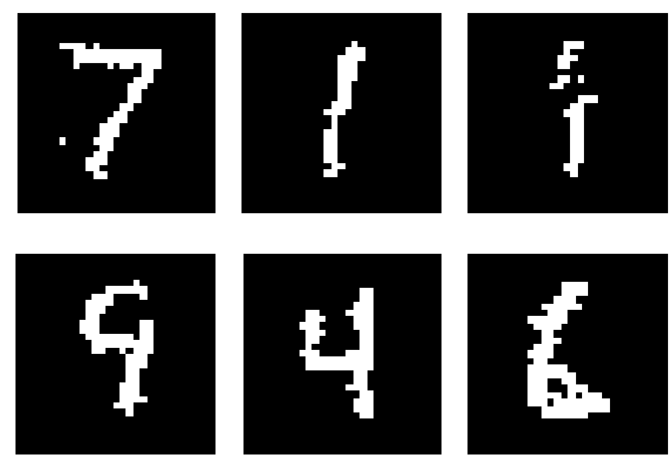
$$P(\mathbf{x} \,|\, \theta) = \sum_{\mathbf{h}} P(\mathbf{x}, \mathbf{h} \,|\, \theta) = \frac{1}{\mathcal{Z}(\theta)} \underbrace{\sum_{\mathbf{h}} \exp\left[\cdots\right]}_{f(\mathbf{x};\theta),\ \text{tractable}}$$

$\mathcal{Z}$ a normalizer



$\mathbf{x} \sim$ Uniform

$\mathbf{x} \sim$ Model

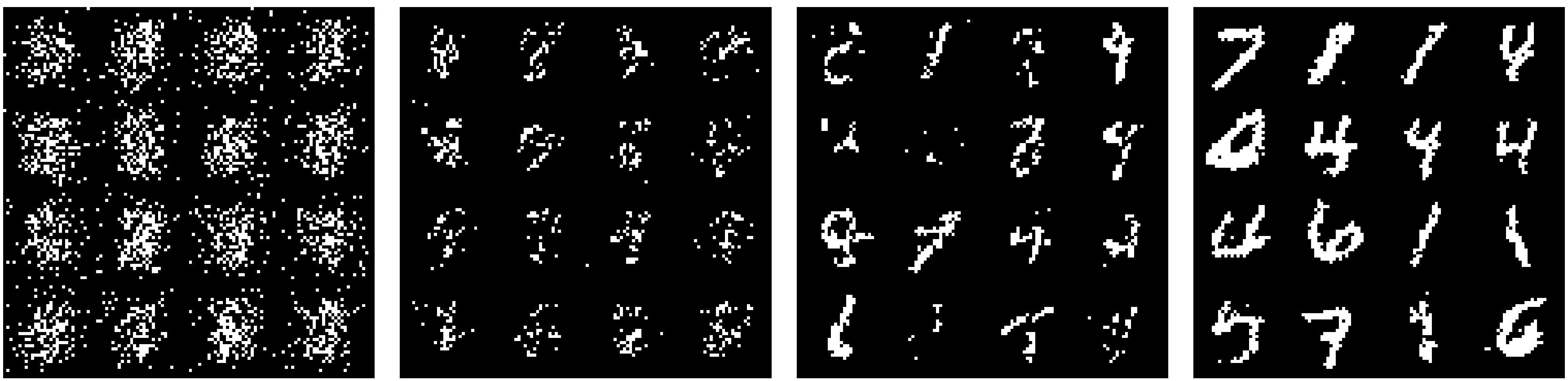


$$p(\mathbf{x}) = \frac{f(\mathbf{x})}{\mathcal{Z}}$$

Annealing / Tempering

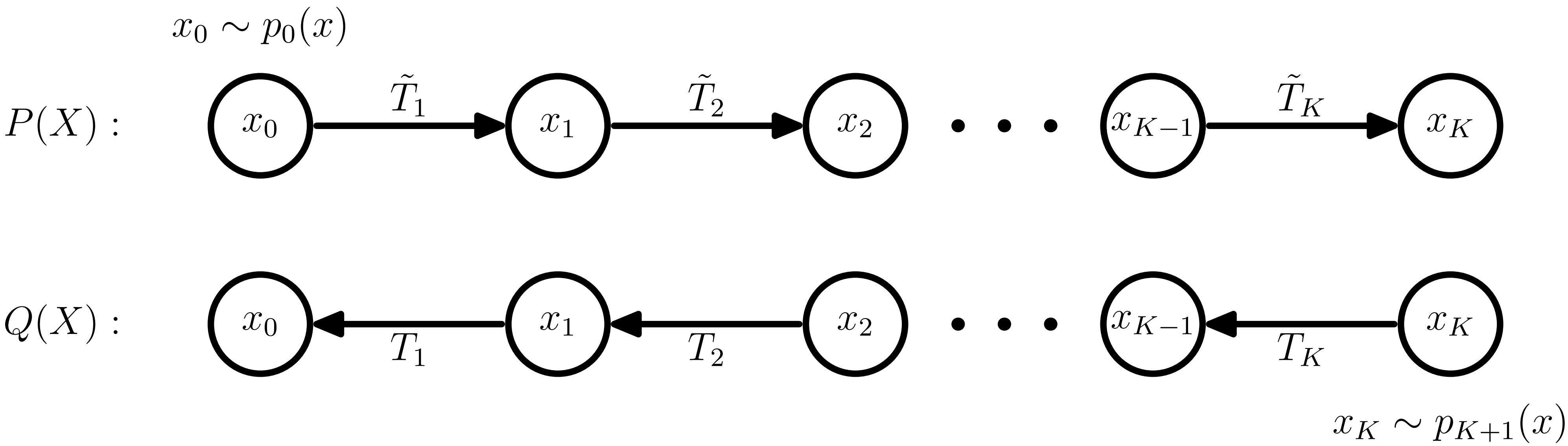
e.g. $P(\mathbf{x}; \beta) \;\propto\; P^*(\mathbf{x})^{\beta}\, \pi(\mathbf{x})^{(1-\beta)}$

$\beta = 0$ $\beta = 0.01$ $\beta = 0.1$ $\beta = 0.25$ $\beta = 0.5$ $\beta = 1$



$1/\beta \;=\;$ "temperature"

Annealed Importance Sampling

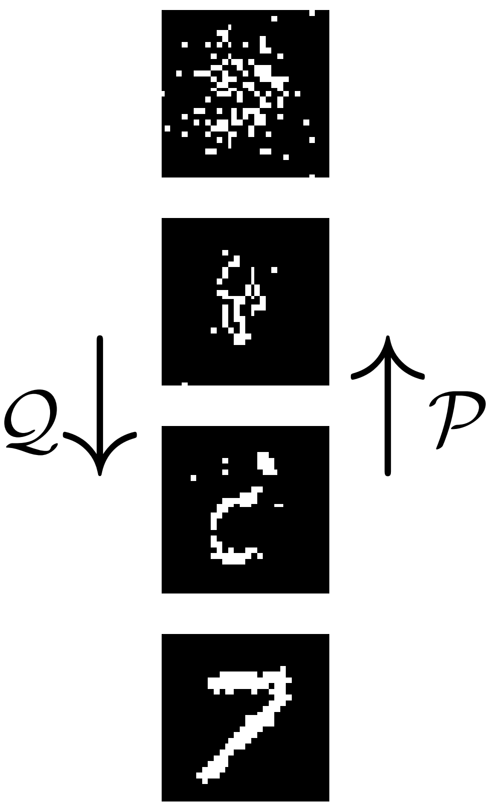


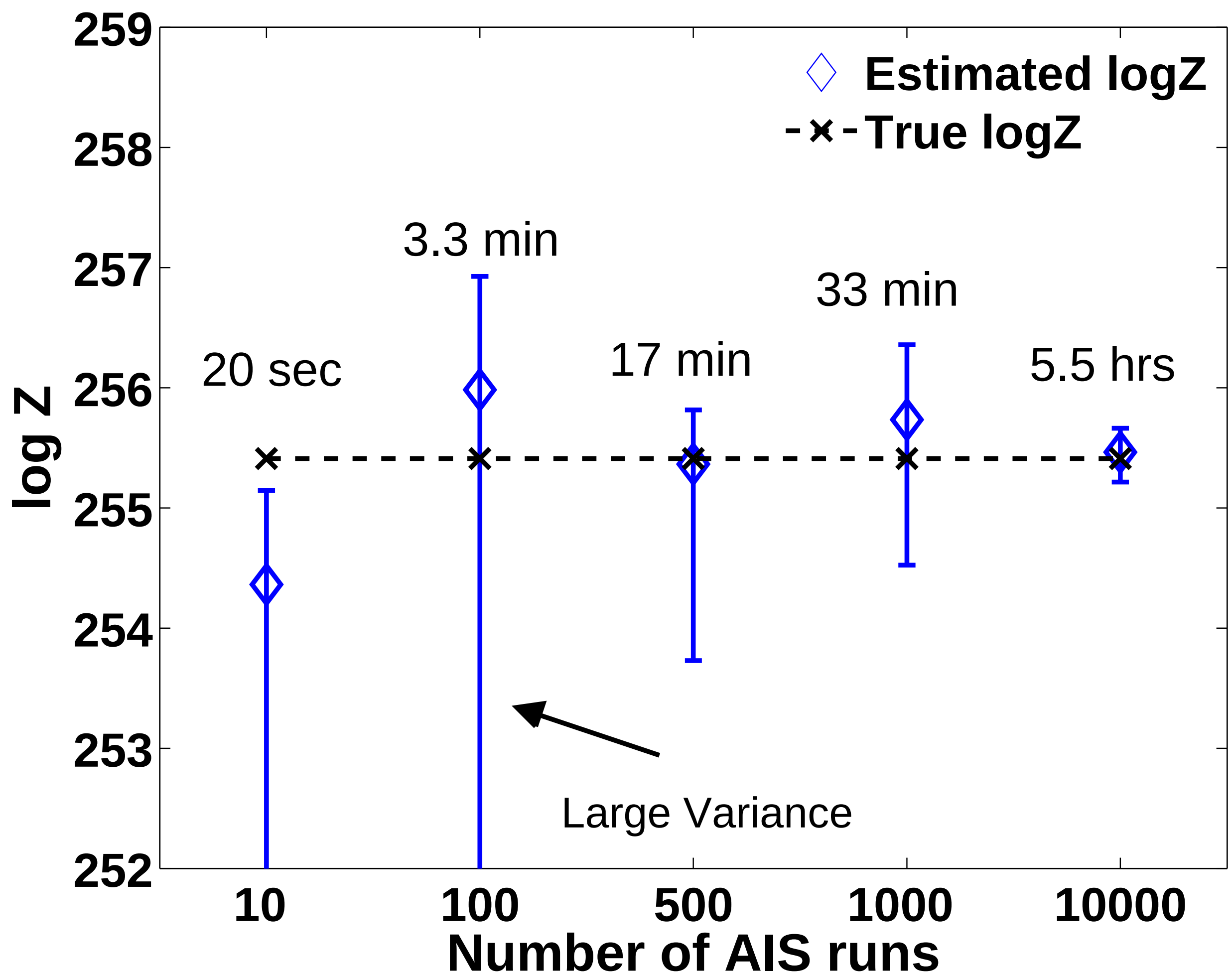
$$\mathcal{P}(X) = \frac{P^*(\mathbf{x}_K)}{\mathcal{Z}} \prod_{k=1}^{K} \widetilde{T}_k(\mathbf{x}_{k-1}; \mathbf{x}_k), \qquad \mathcal{Q}(X) = \pi(\mathbf{x}_0) \prod_{k=1}^{K} T_k(\mathbf{x}_k; \mathbf{x}_{k-1})$$

Standard importance sampling of $\mathcal{P}(X) = \frac{\mathcal{P}^*(X)}{\mathcal{Z}}$ with $\mathcal{Q}(X)$

Annealed Importance Sampling

$$\mathcal{Z} \approx \frac{1}{S}\sum_{s=1}^{S}\frac{\mathcal{P}^*(X)}{\mathcal{Q}(X)}$$

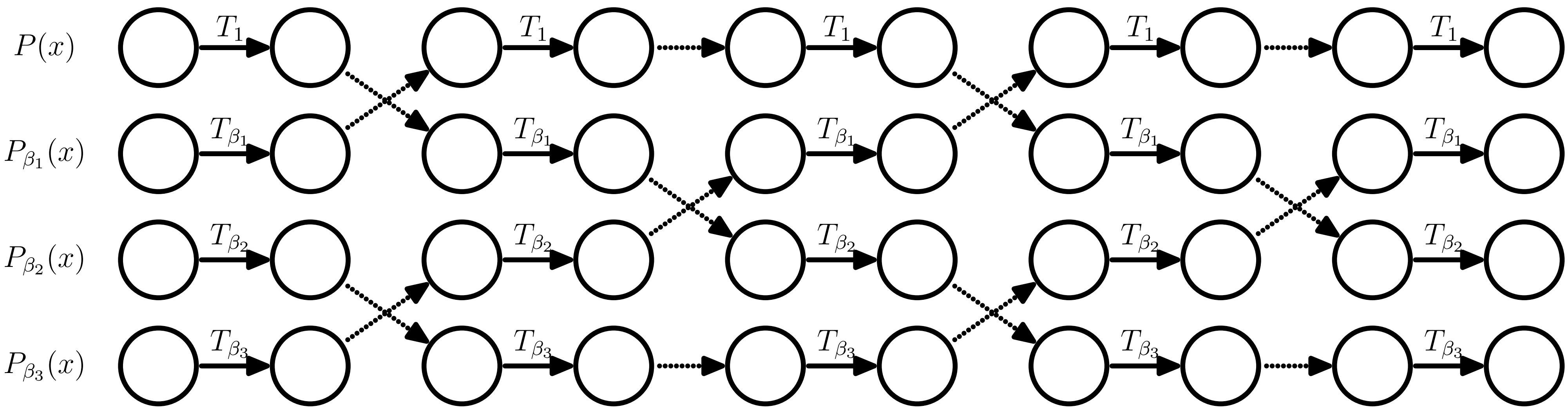




Parallel tempering

Standard MCMC: Transitions + swap proposals on joint:

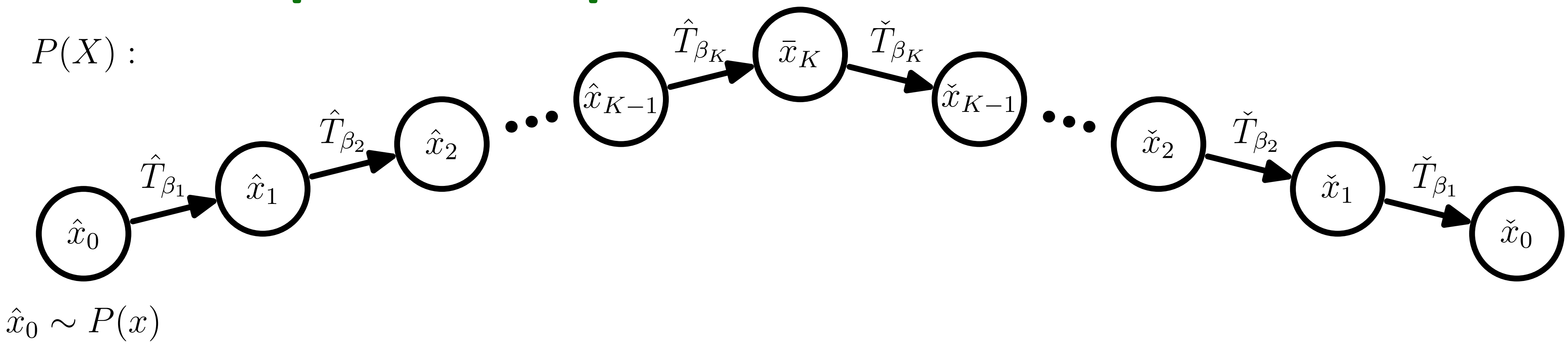
$$P(X) = \prod_{\beta} P(X;\beta)$$



- larger system
- information from low β diffuses up by slow random walk

Tempered transitions

Drive temperature up. . .



. . . and back down

Proposal: swap order, final point $\check{x}_0$ putatively $\sim P(x)$

Acceptance probability:

$$\min\left[1,\ \frac{P_{\beta_1}(\hat{x}_0)}{P(\hat{x}_0)} \cdots \frac{P_{\beta_K}(\hat{x}_{K-1})}{P_{\beta_{K-1}}(\hat{x}_0)} \frac{P_{\beta_{K-1}}(\check{x}_{K-1})}{P_{\beta_K}(\check{x}_{K-1})} \cdots \frac{P(\check{x}_0)}{P_{\beta_1}(\check{x}_0)}\right]$$

Summary on $\mathcal{Z}$

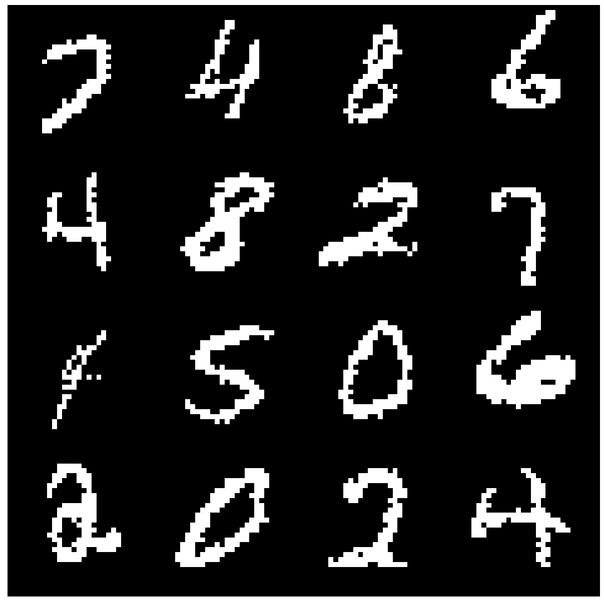
Whirlwind tour of annealing / tempering

Must be able to get anywhere in distribution

Methods to use generally for hardest problems.

An experiment

Take 60,000 binarized MNIST digits, like these:



— Train an RBM using CD (and then find $\mathcal{Z}$)

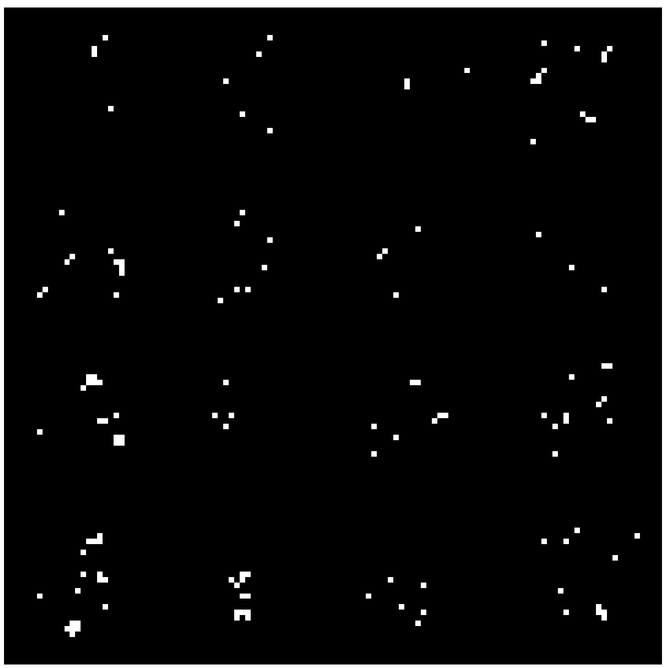
— Train a mixture of multivariate Bernoullis with EM

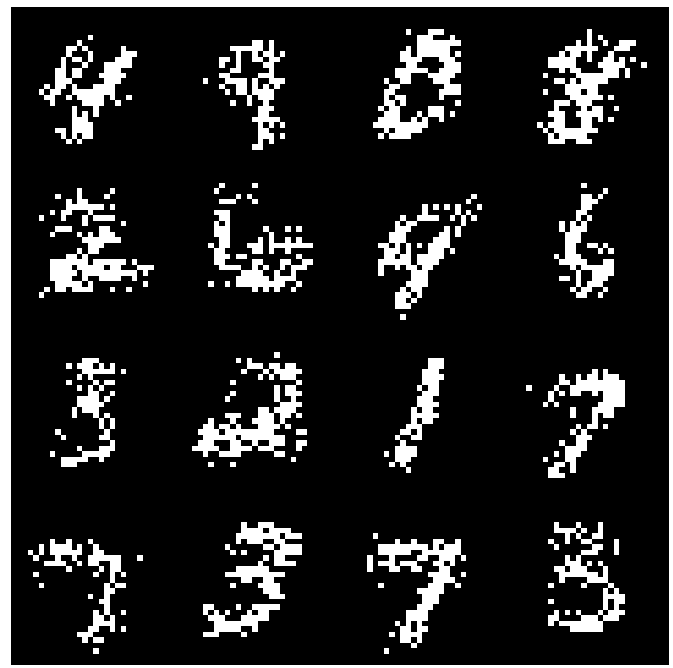
Compare samples and test-set log-probs

A comparison

Samples from:

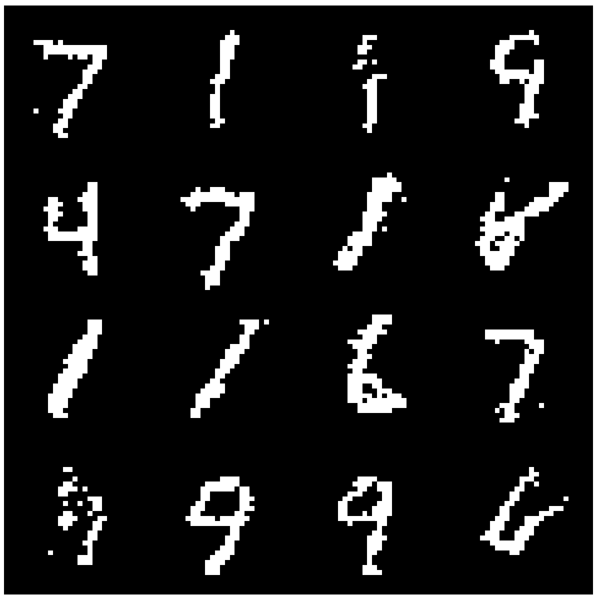
- mixture of Bernoullis, -143 nats/test digit
- RBM, -106 nats/test digit



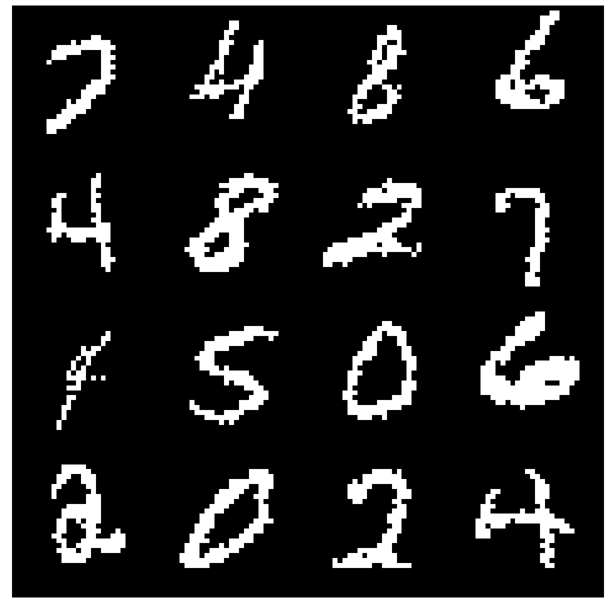


Which is which?

A better fitted RBM



RBM samples

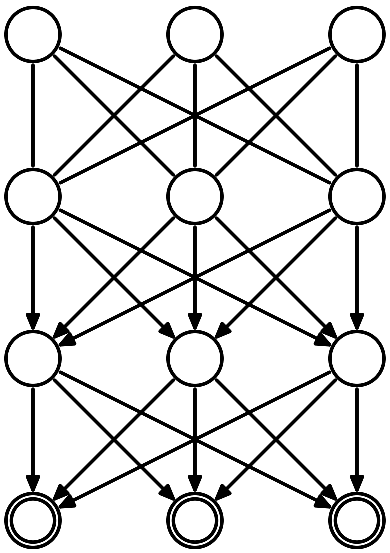


Training set examples

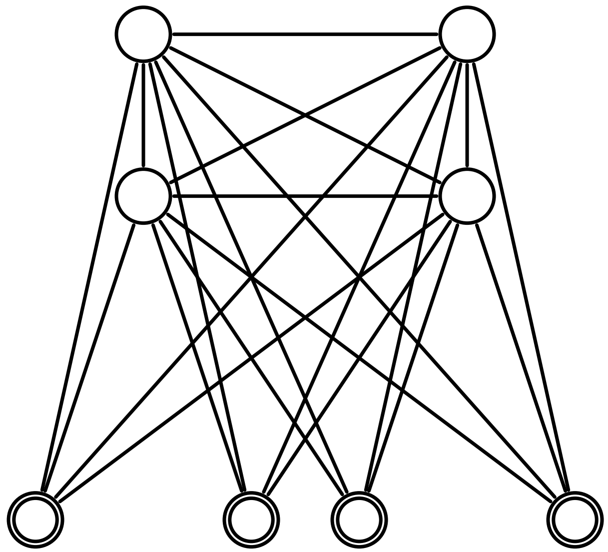
Test log-prob now 20 nats better (-86 nats/digit)

Dependent latent variables

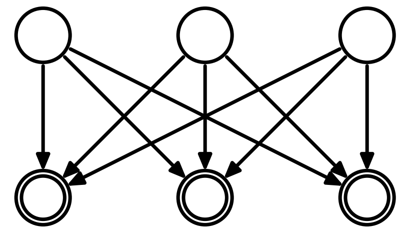
“Deep Belief Net”



Lateral connections



Directed model



$P(\mathbf{x}) = \dfrac{1}{\mathcal{Z}} \sum_{\mathbf{h}} P^*(\mathbf{x}, \mathbf{h})$, not available

Chib-style estimates

Bayes Rule:

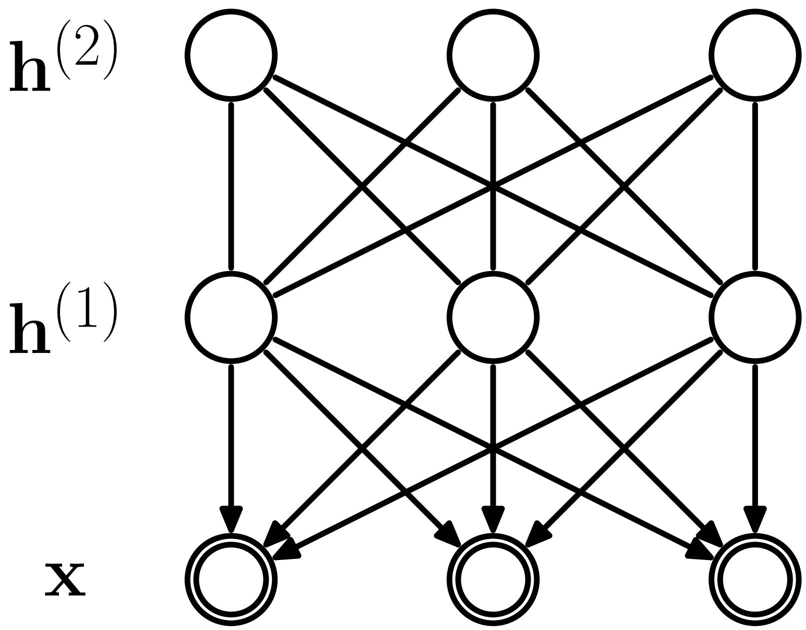
$$P(\mathbf{h} \,|\, \mathbf{x}) = \frac{P(\mathbf{h}, \mathbf{x})}{P(\mathbf{x})}$$

For *any* special state $\mathbf{h}^*$:

$$P(\mathbf{x}) = \frac{P(\mathbf{h}^*, \mathbf{x})}{P(\mathbf{h}^* \,|\, \mathbf{x})} \;\leftarrow \text{Estimate}$$

Murray and Salakhutdinov (2009)

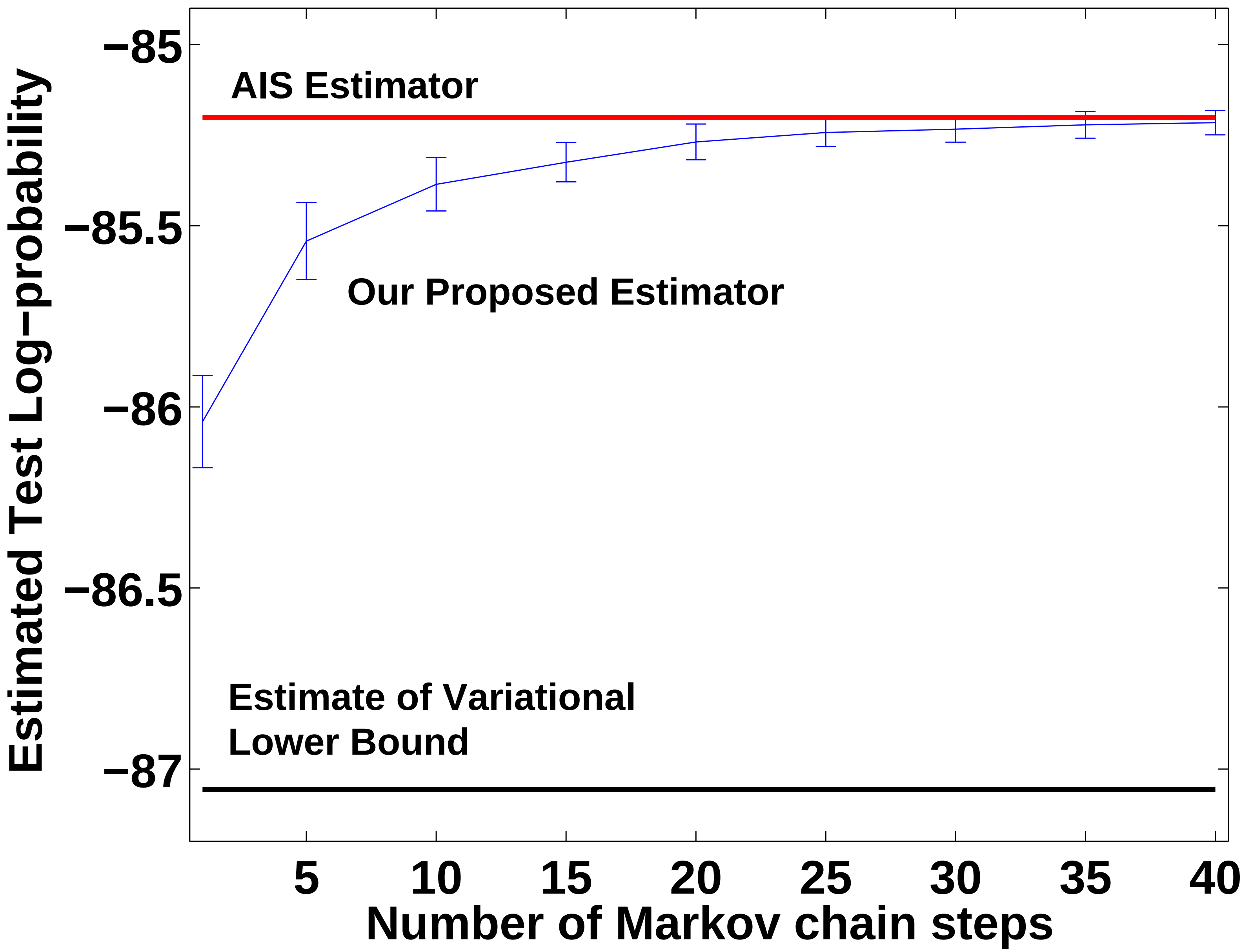
Variational approach



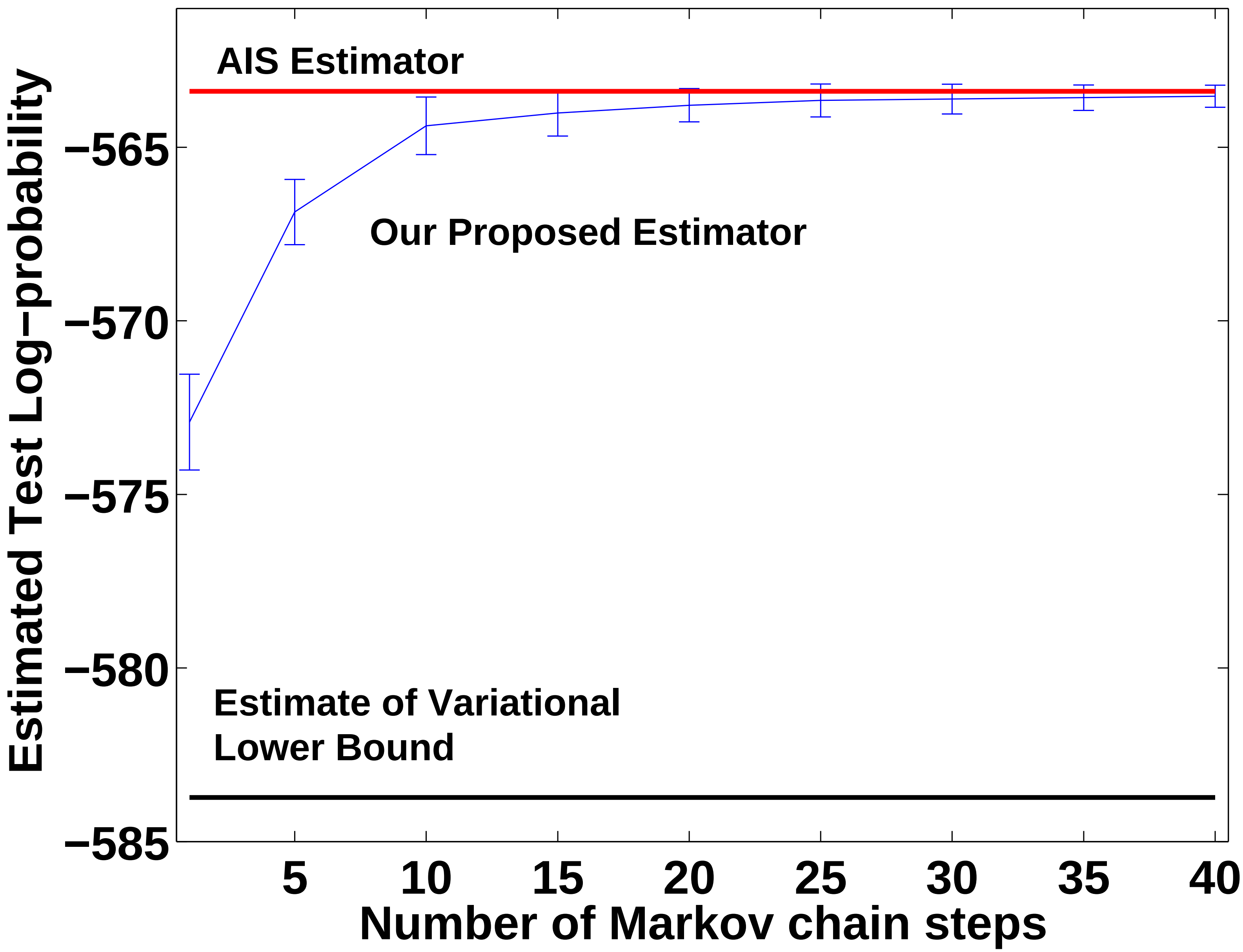
$$\log P(\mathbf{x}) = \log \sum_{\mathbf{h}} \frac{1}{\mathcal{Z}} P^*(\mathbf{x}, \mathbf{h})$$

$$\geq \sum_{\mathbf{h}} Q(\mathbf{h}) \log P^*(\mathbf{x}, \mathbf{h}) - \log \mathcal{Z} + \mathcal{H}[Q(\mathbf{h})]$$

Results MNIST



Results Natural Scenes



$P(\mathbf{x} \,|\, \mathcal{H})$ taught me

— RBM: state-of-the-art for binary dists

— Deep nets only very slightly better on MNIST

— Some Gaussian RBMs are really bad...
...and going deep won't help

— Most topic model $P(\mathbf{x} \,|\, \mathcal{H})$ ests wrong

Roadmap

— Unsupervised learning and $P(\mathbf{x} \,|\, \mathcal{H})$

— Evaluating $P(\mathbf{x} \,|\, \mathcal{H})$

Salakhutdinov and Murray (2008)

Murray and Salakhutdinov (2009)

Wallach, Murray, Salakhutdinov & Mimno (2009)

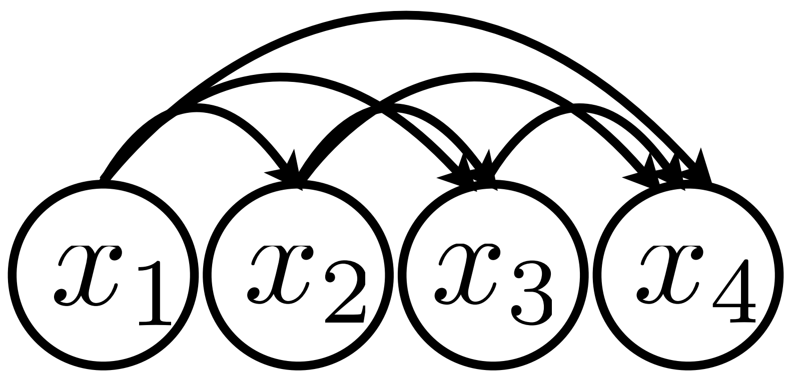
— NADE: “density estimation put first”

Larochelle and Murray (2011)

Decompose into scalars

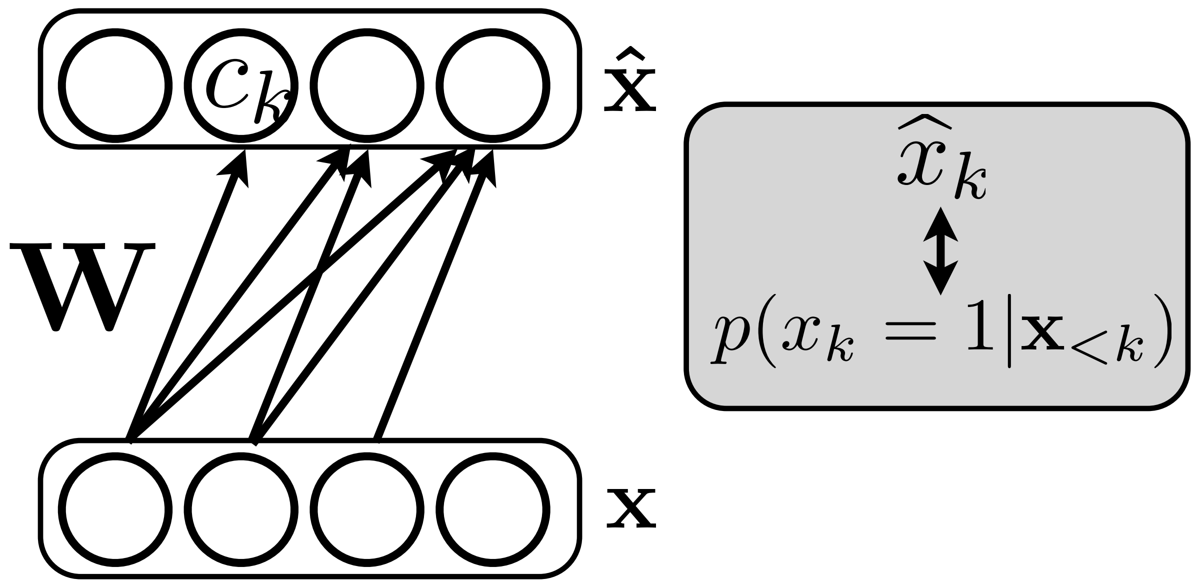
$$P(\mathbf{x}) = P(x_1)\,P(x_2 \,|\, x_1)\,P(x_3 \,|\, x_1, x_2)\ldots$$

$$= \prod_k P(x_k \,|\, \mathbf{x}_{<k})$$



FVSBN: Fully Visible Sigmoid Belief Net

Logistic regression for conditionals



FVSBNs beat mixtures, but not RBMs

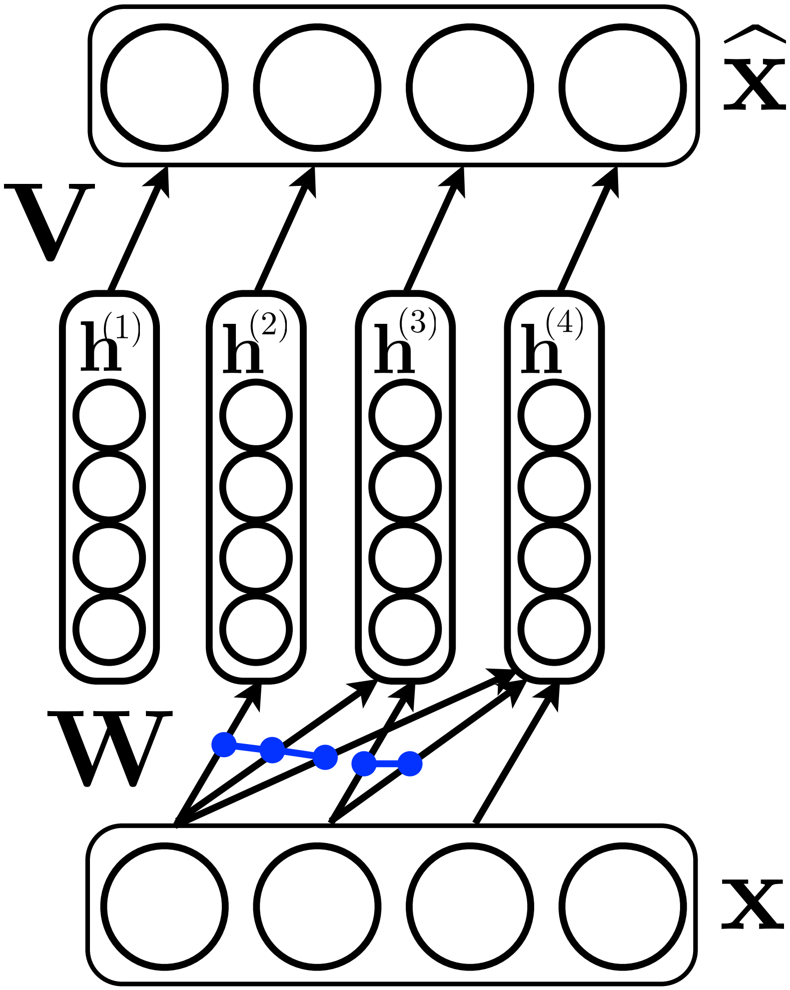
Approximate RBM

$P(x_k \,|\, \mathbf{x}_{<k})$ from MCMC

or mean field

(Requires fitted RBM. Creates new model.)

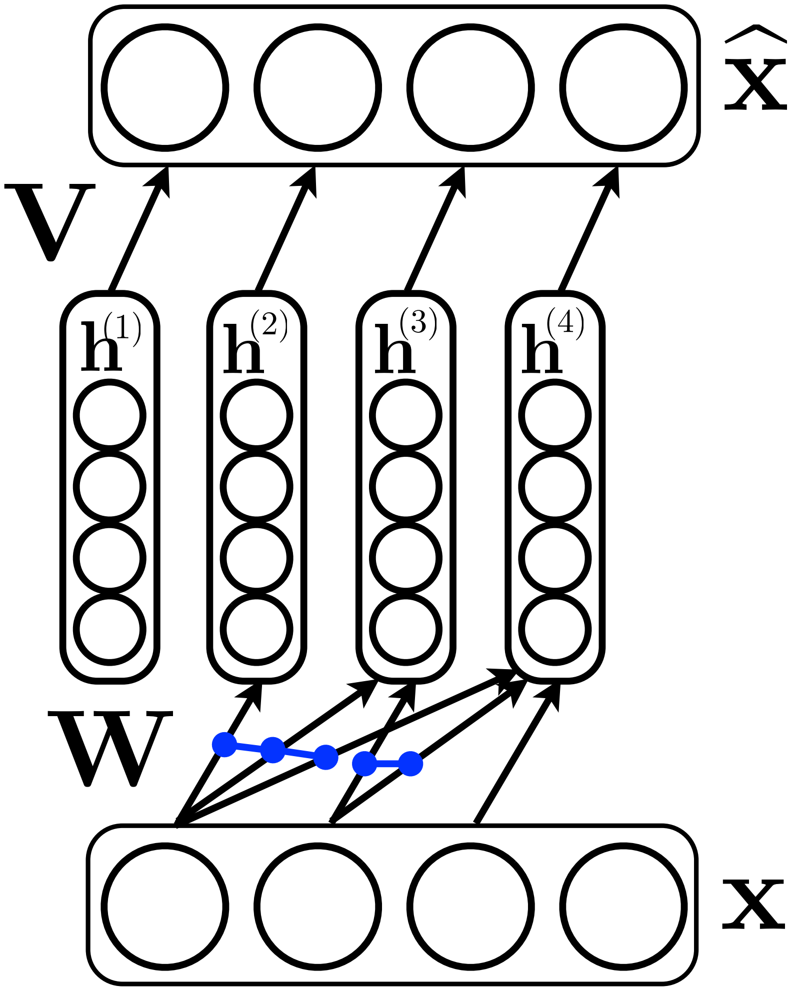
One Mean Field step



$$\hat{x}_k = \sigma\left(b^x_k + W_{k,\cdot}\mathbf{h}^{(k)}\right)$$

$$\mathbf{h}^{(k)} = \sigma\left(\mathbf{b}^h + W^\top_{\cdot,<k}\mathbf{x}_{<k}\right)$$

NADE Neural Autoregressive Distribution Estimator



Fit as new model

$$P(\mathbf{x}\,|\,\mathcal{H}) = \prod_k \hat{p}(x_k)$$

Tractable, $\mathcal{O}(DH)$

NADE results

Model	ADULT	CONNECT-4	DNA	MUSHROOMS	NIPS-0-12	OCR-LETTERS	RCV1	WEB
MoB	0.00 $\pm$ 0.10	0.00 $\pm$ 0.04	0.00 $\pm$ 0.53	0.00 $\pm$ 0.10	0.00 $\pm$ 1.12	0.00 $\pm$ 0.32	0.00 $\pm$ 0.11	0.00 $\pm$ 0.23
RBM	4.18 $\pm$ 0.06	0.75 $\pm$ 0.02	1.29 $\pm$ 0.48	-0.69 $\pm$ 0.09	12.65 $\pm$ 1.07	-2.49 $\pm$ 0.30	-1.29 $\pm$ 0.11	0.78 $\pm$ 0.20
RBM mult.	4.15 $\pm$ 0.06	-1.72 $\pm$ 0.03	1.45 $\pm$ 0.40	-0.69 $\pm$ 0.05	11.25 $\pm$ 1.06	0.99 $\pm$ 0.29	-0.04 $\pm$ 0.11	0.02 $\pm$ 0.21
RBForest	4.12 $\pm$ 0.06	0.59 $\pm$ 0.02	1.39 $\pm$ 0.49	0.04 $\pm$ 0.07	12.61 $\pm$ 1.07	3.78 $\pm$ 0.28	0.56 $\pm$ 0.11	-0.15 $\pm$ 0.21
FVSBN	**7.27 $\pm$ 0.04**	11.02 $\pm$ 0.01	**14.55 $\pm$ 0.50**	4.19 $\pm$ 0.05	13.14 $\pm$ 0.98	1.26 $\pm$ 0.23	-2.24 $\pm$ 0.11	0.81 $\pm$ 0.20
NADE	**7.25 $\pm$ 0.05**	**11.42 $\pm$ 0.01**	13.38 $\pm$ 0.57	**4.65 $\pm$ 0.04**	**16.94 $\pm$ 1.11**	**13.34 $\pm$ 0.21**	**0.93 $\pm$ 0.11**	**1.77 $\pm$ 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)*	$\approx$-125.53
RBM (CD3)*	$\approx$-105.50
RBM (CD25)*	$\approx$-86.34
FVSBN	-97.45
NADE	-88.86

Samples

Talking points

When should we learn $P(\mathbf{x} \mid \mathcal{H})$?

Monte Carlo methods

Autoregressive models (see also Lucas Theis)

A longer talk on NADE:

`http://videolectures.net/aistats2011_larochelle_neural/`

Appendix slides

Markov chain estimation

Stationary condition for Markov chain:

$$P(\mathbf{h}^*|\mathbf{x}) = \sum_{\mathbf{h}} T(\mathbf{h}^* \!\leftarrow\! \mathbf{h}) P(\mathbf{h}|\mathbf{x})$$

$$\approx \left[\frac{1}{S}\sum_{s=1}^{S} T(\mathbf{h}^* \!\leftarrow\! \mathbf{h}^{(s)}), \quad \mathbf{h}^{(s)} \sim \mathcal{P}(H)\right] = \hat{p}$$

$\mathcal{P}(H)$ draws a sequence from an equilibrium Markov chain:

$$\mathbf{h}^{(1)} \xrightarrow[T]{} \mathbf{h}^{(2)} \xrightarrow[T]{} \mathbf{h}^{(3)} \xrightarrow[T]{} \mathbf{h}^{(4)} \xrightarrow[T]{} \mathbf{h}^{(5)}$$

$\mathbf{h}^{(1)} \sim P(\mathbf{h}|\mathbf{v})$

Bias in answer

$$P(\mathbf{x}) = \frac{P(\mathbf{h}^*, \mathbf{x})}{P(\mathbf{h}^*|\mathbf{x})} = \frac{P(\mathbf{h}^*, \mathbf{x})}{\mathbb{E}[\hat{p}]} \leq \mathbb{E}\left[\frac{P(\mathbf{h}^*, \mathbf{x})}{\hat{p}}\right]$$

Idea: bias Markov chain by starting at $\mathbf{h}^*$

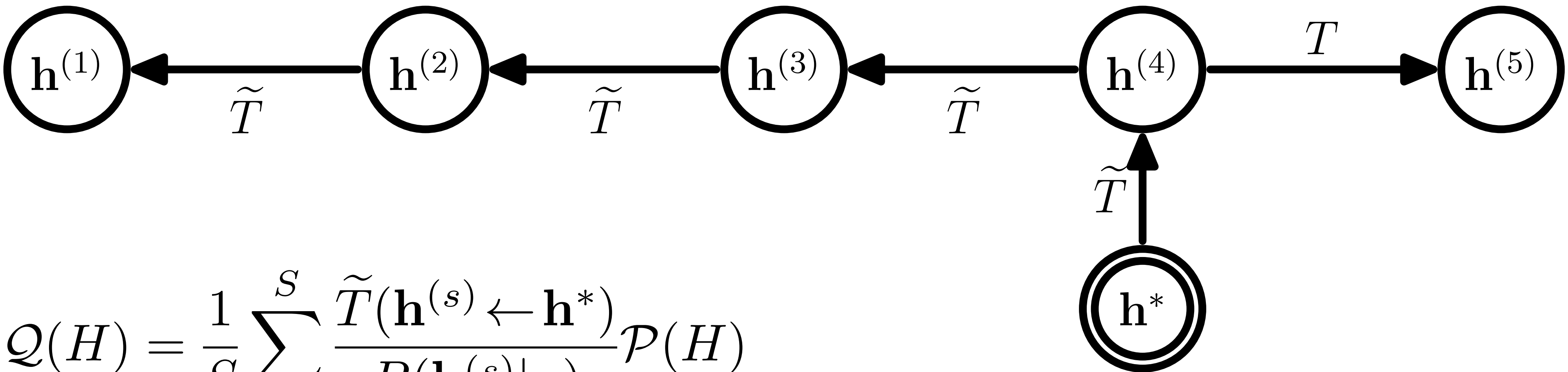
$\frac{1}{S}\sum_{s=1}^{S} T(\mathbf{h}^* \!\leftarrow\! \mathbf{h}^{(s)})$ will often overestimate $P(\mathbf{h}^*|\mathbf{x})$

$$\begin{tikzcd}
\mathbf{h}^{(1)} \arrow[r, "T"'] & \mathbf{h}^{(2)} \arrow[r, "T"'] & \mathbf{h}^{(3)} \arrow[r, "T"'] & \mathbf{h}^{(4)} \arrow[r, "T"'] & \mathbf{h}^{(5)} \\
\mathbf{h}^* \arrow[u, "T"] & & & &
\end{tikzcd}$$

$$\mathcal{Q}(H) = \frac{T(\mathbf{h}^{(1)} \!\leftarrow\! \mathbf{h}^*)}{P(\mathbf{h}^{(1)}|\mathbf{x})}\mathcal{P}(H)$$

New estimator

We actually need a slightly more complicated $\mathcal{Q}$:



$$\mathcal{Q}(H) = \frac{1}{S}\sum_{s=1}^{S} \frac{\widetilde{T}(\mathbf{h}^{(s)} \leftarrow \mathbf{h}^{*})}{P(\mathbf{h}^{(s)}|\mathbf{x})}\mathcal{P}(H)$$

$$\mathbb{E}_{\mathcal{Q}(H)}\left[1 \Big/ \frac{1}{S}\sum_{s=1}^{S} T(\mathbf{h}^{*} \leftarrow \mathbf{h}^{(s)})\right] = \frac{1}{P(\mathbf{h}^{*}|\mathbf{x})}$$

$\hat{P}(\mathbf{x}) = \dfrac{P(\mathbf{x}, \mathbf{h}^{*})}{\hat{P}(\mathbf{h}^{*}|\mathbf{x})}$ unbiased $\Rightarrow$ stochastic lower bound on $\log P(\mathbf{x})$