

Generative Stochastic Networks: how to reduce the difficulties arising from marginalization across many major modes

Yoshua Bengio

U. Montreal

Thanks to: Eric Laufer, Li Yao, Guillaume Alain,
Pascal Vincent, Jason Yosinski

October 23nd, 2013

Masterclass lecture, UCL Center for Computational
Statistics and Machine Learning

Machine Learning as Estimating the Underlying Data Distribution (or aspects of it)

- Learning as the estimation of a probability function
- Generalization: guessing **where** probability mass concentrates
- Challenge: the curse of dimensionality (exponentially many configurations of the variables to consider)

Basic Challenge with Probabilistic Models: marginalization

- Joint and marginal likelihoods involve intractable sums over configurations of random variables (inputs x , latent h , outputs y) e.g.

$$P(x) = \sum_h P(x,h)$$

$$P(x,h) = e^{-\text{energy}(x,h)} / Z$$

$$Z = \sum_{x,h} e^{-\text{energy}(x,h)}$$

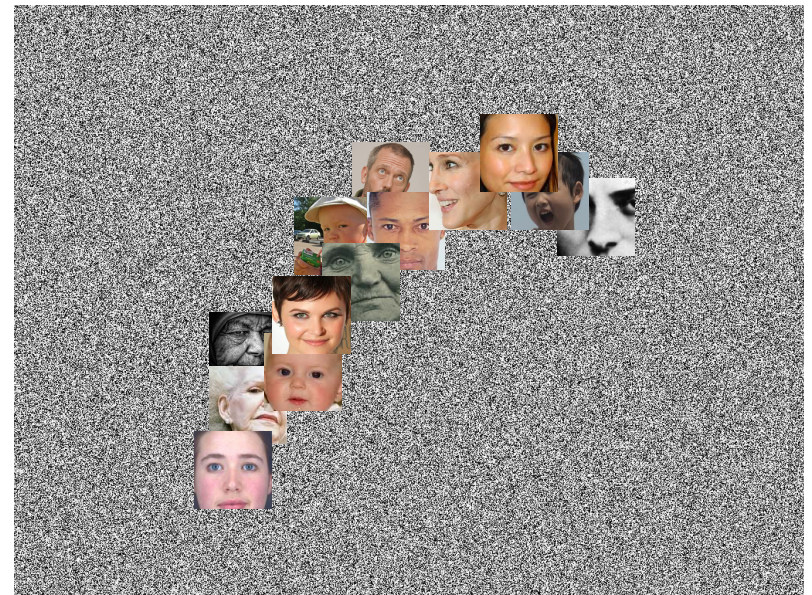
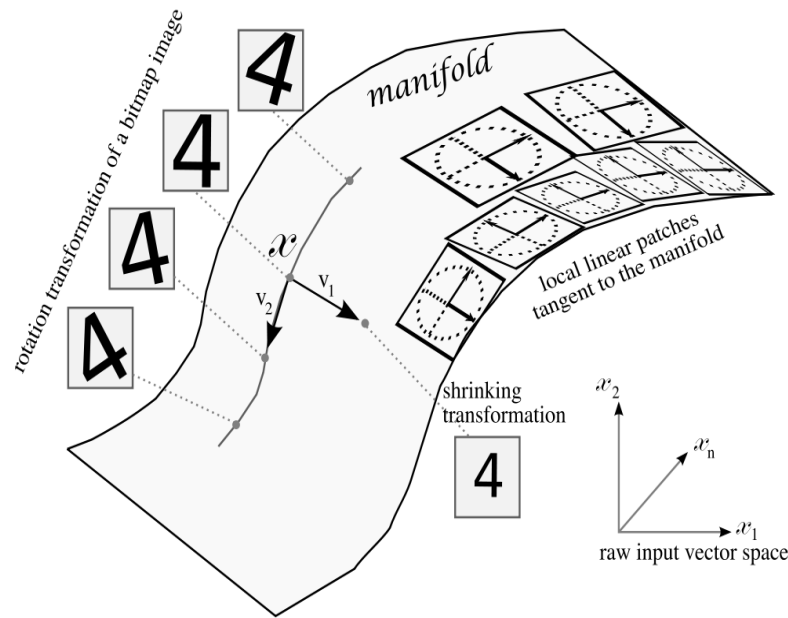
- MCMC methods can be used for these sums, by sampling from a chain of x 's (or of (x,h) pairs) approximately from $P(x,h)$

Two Fundamental Problems with Probabilistic Models with Many Random Variables

1. MCMC mixing between modes
(manifold hypothesis)
2. Many non-negligible modes
(both in posterior & joint distributions)

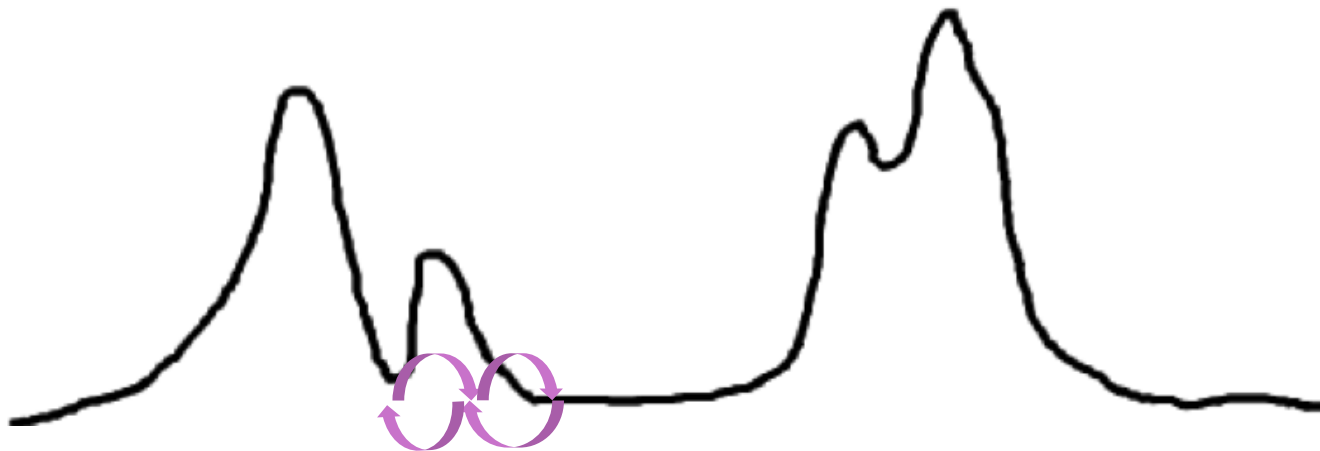
For AI Tasks: Manifold structure

- examples **concentrate** near a lower dimensional “manifold” (region of high density with only few operations allowed which allow small changes while staying on the manifold)
- **Evidence: most input configurations are unlikely**



Mixing Between Well-Separated Modes is Fundamentally Hard

- MCMC steps are typically local (otherwise, curse of dimensionality means most proposals would be rejected)
- If the data has a manifold structure, the chances of going from manifold A to manifold B = prob. accepting a long string of improbable moves = exponentially small

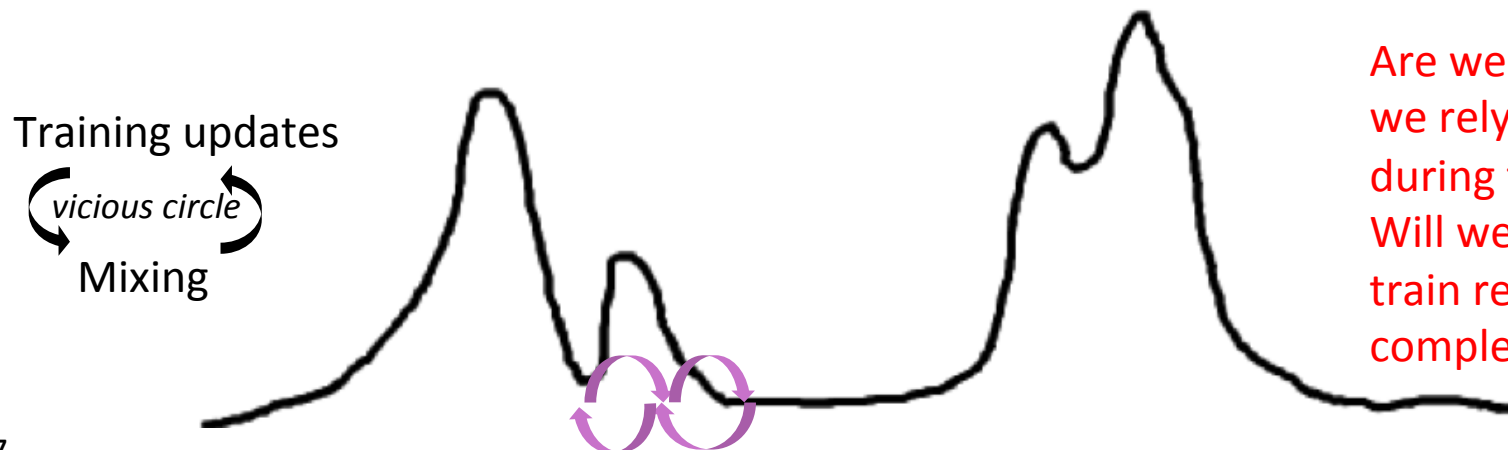


Mixing Between Modes: Vicious Circle Between Learning and MCMC Sampling

- Early during training, density smeared out, mode bumps overlap



- Later on, hard to cross empty voids between modes



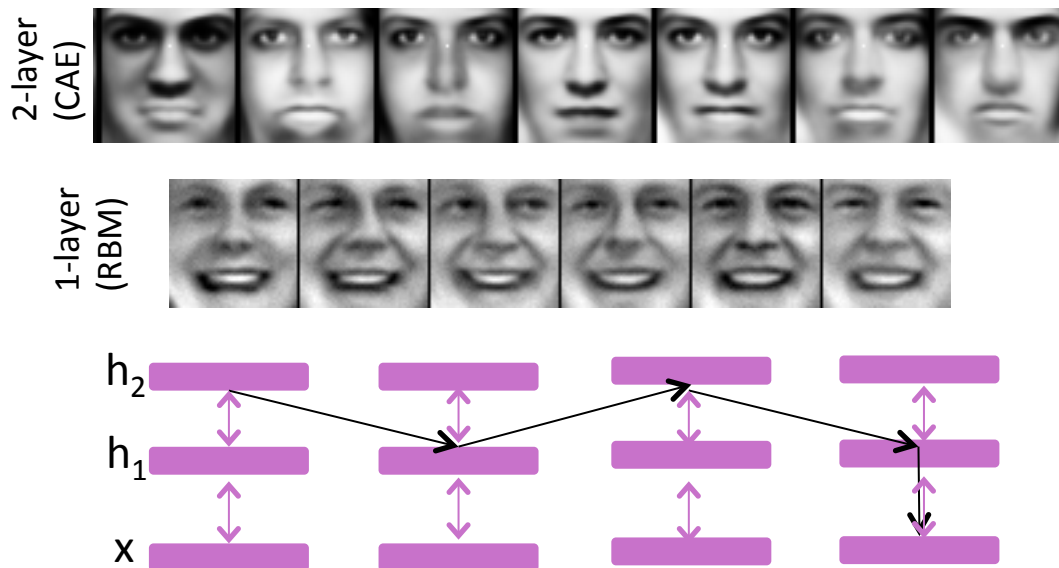
Fixing the Mixing Problem?

- If there were few important modes, we could just run many chains from different random starts and collect the results
- We have tried that and it did not work
- Another option is tempering and related variants
- Appealing but very expensive, has not fixed the problem yet
- Deep representations seem to be a promising avenue

Poor Mixing: Depth to the Rescue

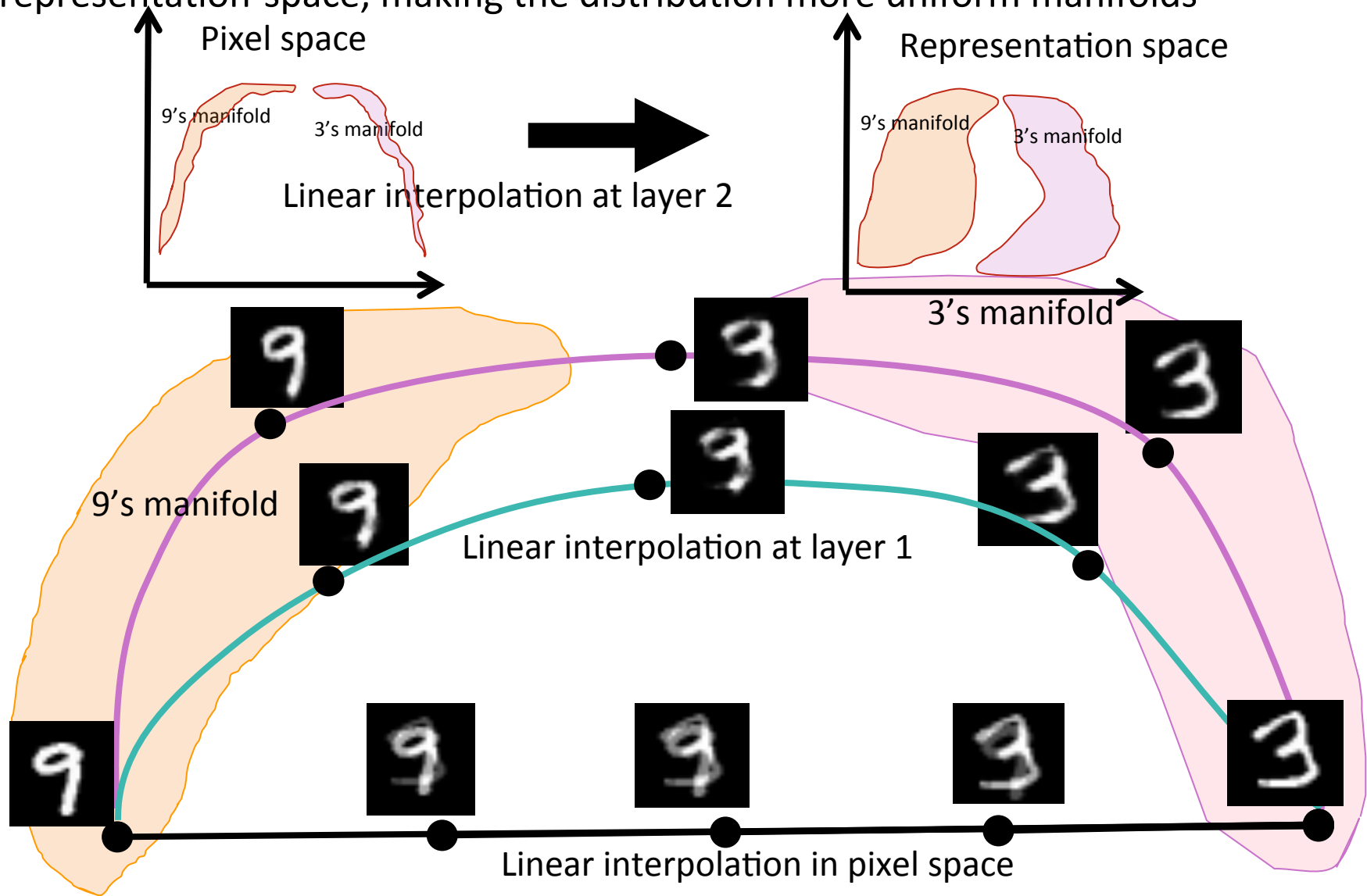
(Bengio et al ICML 2013)

- Sampling from DBNs and stacked Contractive Auto-Encoders:
 1. MCMC sampling from top layer model
 2. Propagate top-level representations to input-level repr.
- Deeper nets visit more modes (classes) faster! **WHY?**



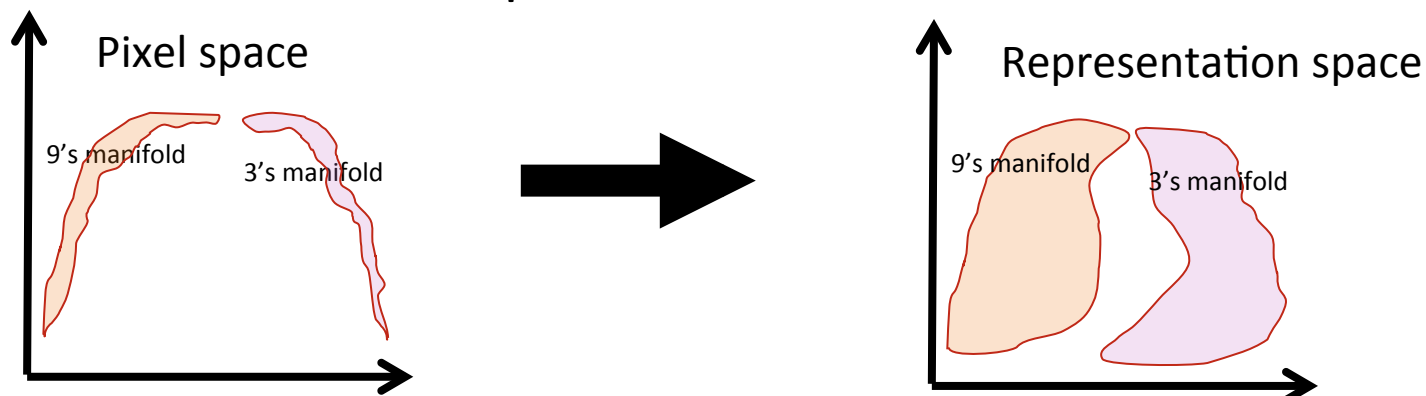
Space-Filling in Representation-Space

High-probability samples fill space between them when viewed in the learned representation-space, making the distribution more uniform manifolds



Poor Mixing: Depth to the Rescue

- Deeper representations \rightarrow abstractions \rightarrow disentangling
- E.g. reverse video bit, class bits in learned representations: easy to Gibbs sample between modes at abstract level
- Hypotheses tested and not rejected:
 - more abstract/disentangled representations unfold manifolds and fill more the space



- can be exploited for better mixing between modes

The Main Problem
that Remains:
**MANY IMPORTANT
MODES**

Many Important Modes

- Issue arises typically in two places with probabilistic models:
 - Inference: need to consider the major modes of $P(h|x)$ or $P(y,h|x)$
 - Learning (estimating the log-likelihood gradient): need to consider the major modes of $P(h,x)$ when computing the gradient of the normalization constant
- Important for:
 - Unsupervised (and semi-supervised) learning
 - Structured output learning

Potentially **Huge** Number of Modes in the Posterior $P(h|x)$

- Human hears foreign speech, y =answer to question:
 - 10 word segments
 - 100 plausible candidates per word
 - 10^6 possible segmentations
 - Most configurations (999999/1000000) implausible
 - $\rightarrow 10^{20}$ high-probability modes
- Humans probably don't consider all these in their mind
- **All known approximate inference scheme break down if the posterior has a huge number of modes** (fails MAP & MCMC) and not respecting a variational approximation (fails variational)

PROPOSED SOLUTION

~~• Approximate inference?~~

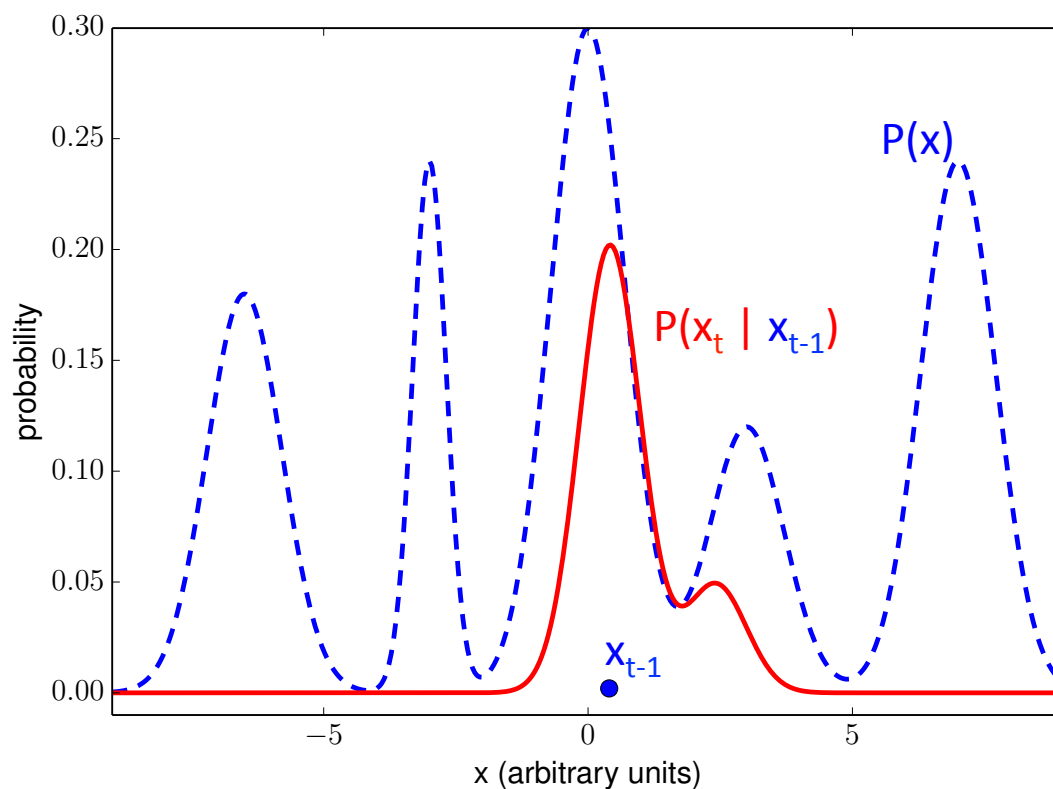
• Function approximation

Hint

- Deep neural nets learn good $P(y|\mathbf{x})$ classifiers even if there are potentially many true latent variables involved
- Exploits structure in $P(y|\mathbf{x})$ that persist even after summing h
- But how do we generalize this idea to full joint-distribution learning and answering any question about these variables, not just one?

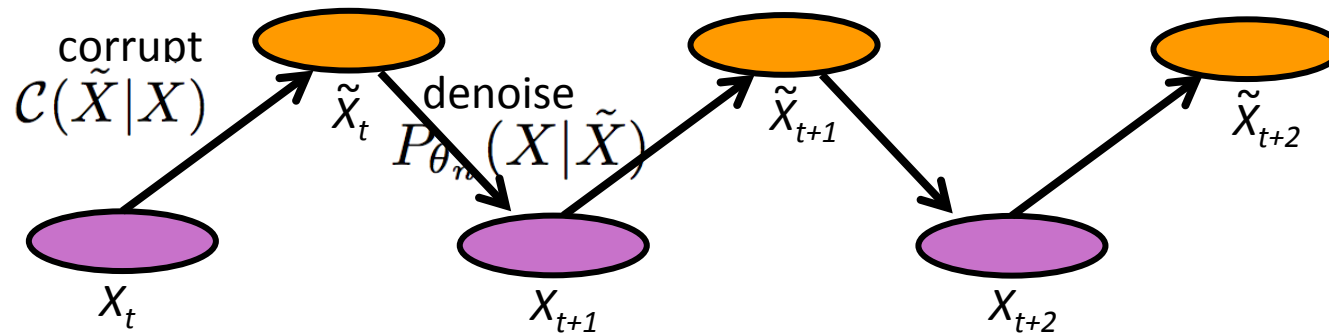
Instead of Learning $P(x)$ directly, Learn Markov chain operator $P(x_t | x_{t-1})$

- $P(x)$ may have many modes, making the normalization constant intractable, and MCMC approximations poor
- $P(x_t | x_{t-1})$ could be much simpler because most of the time a local move, might even be well approximated by unimodal

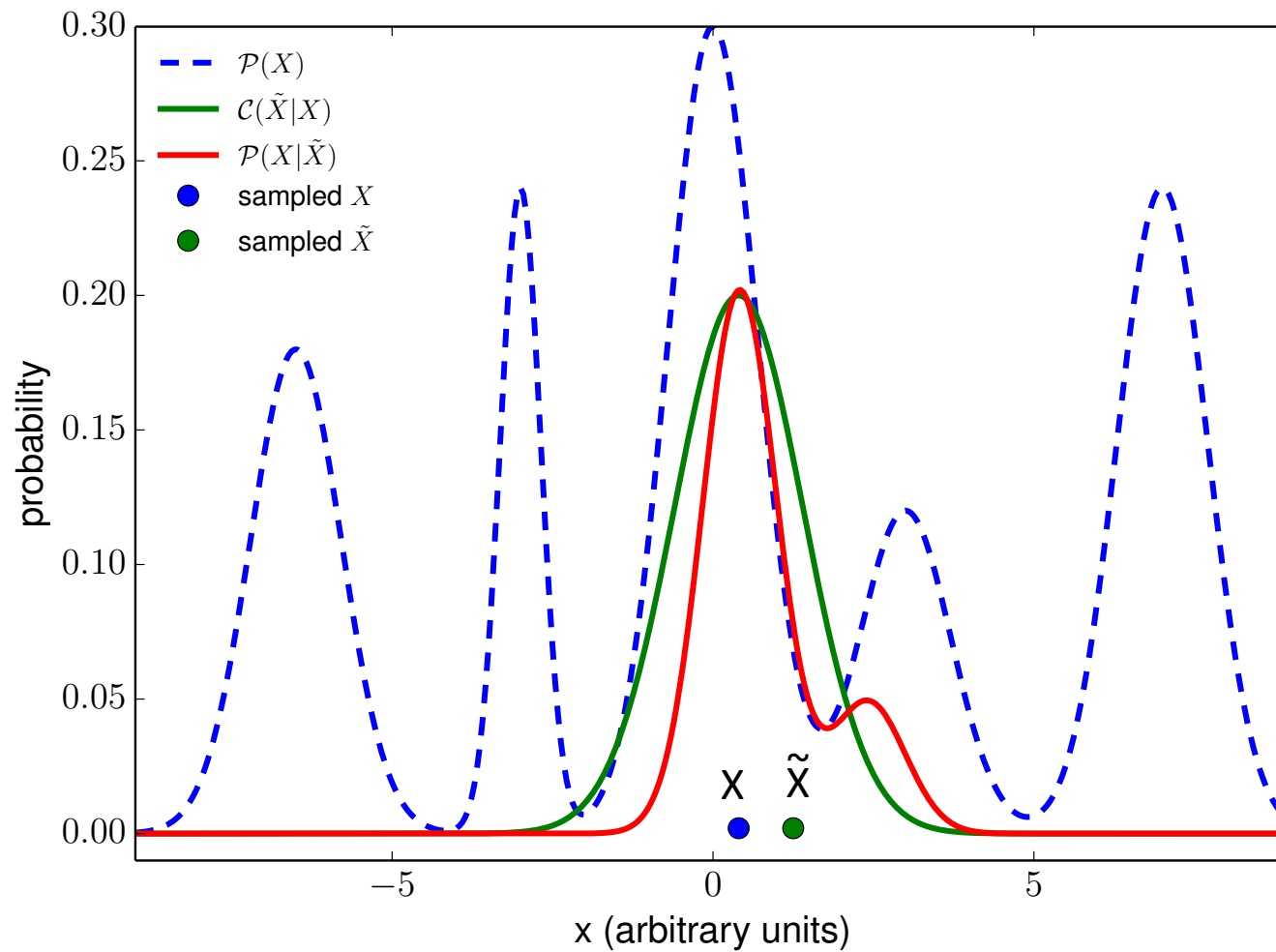


How to train the transition operator?

- One solution was recently discovered, based on the denoising auto-encoder research
- The transition operator is decomposed in two steps:
 - Corruption process $\mathcal{C}(\tilde{X}|X)$
 - Reconstruction (denoising) distribution $P_{\theta_n}(X|\tilde{X})$
- The parameters can be trained by maximum likelihood over the pairs \tilde{X}, X

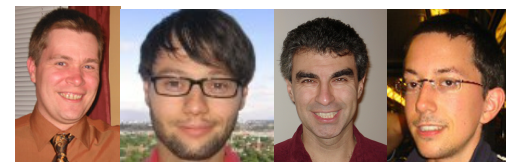


Decomposing the Transition Operator

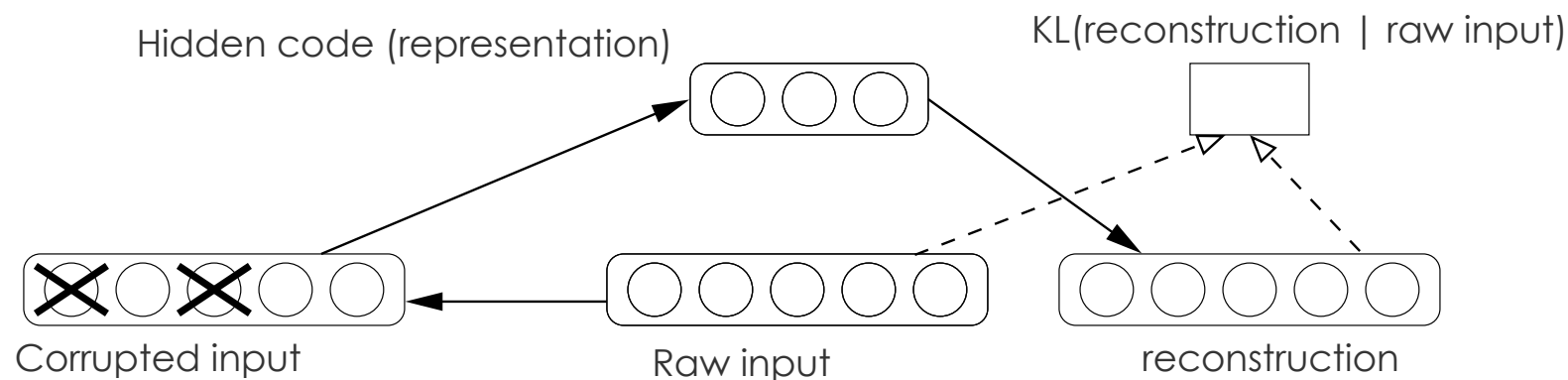


Denoising Auto-Encoder

(Vincent et al 2008)



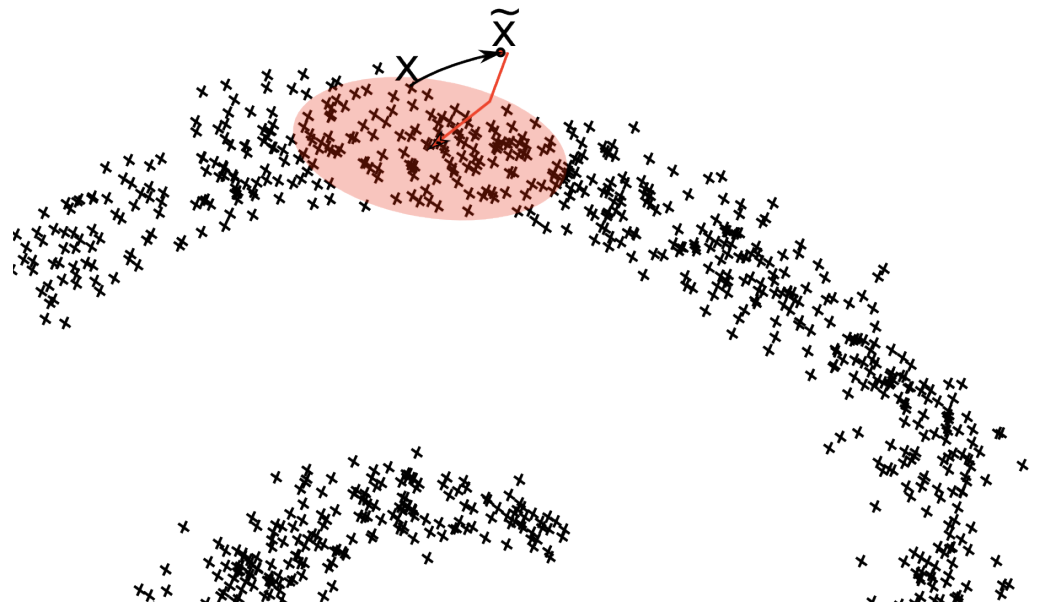
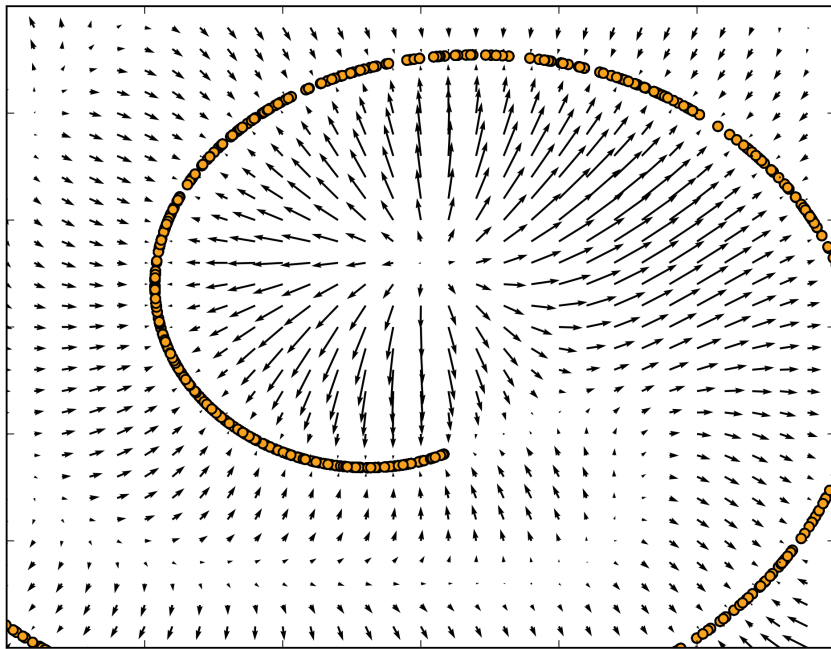
- Corrupt the input during training only
- Train to reconstruct the uncorrupted input



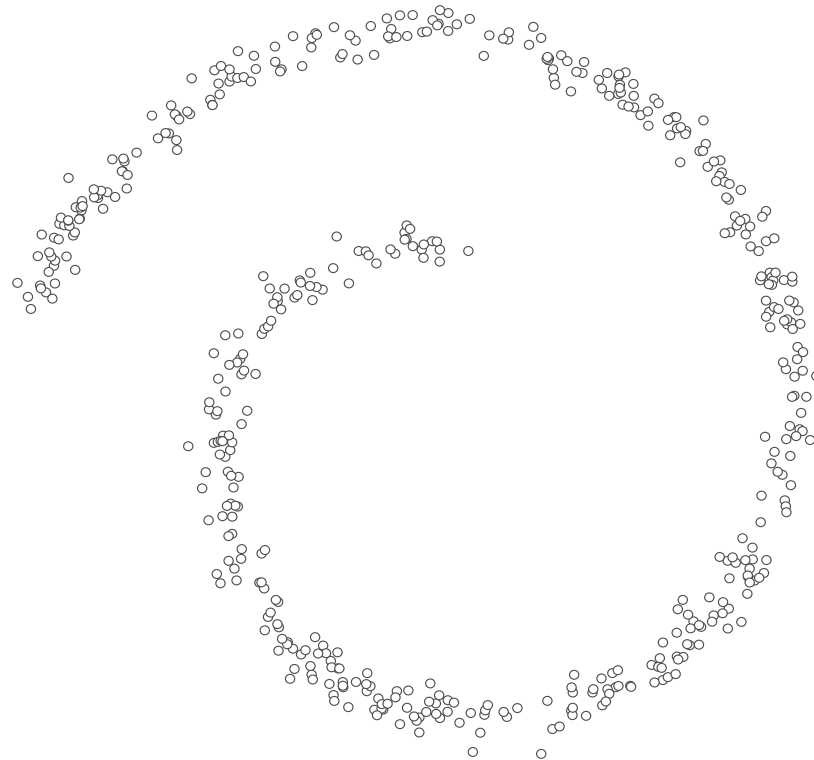
- Encoder & decoder: any parametrization
- As good or better than RBMs for unsupervised pre-training

Regularized Auto-Encoders Learn a Vector Field or a Markov Chain Transition Distribution

- (Bengio, Vincent & Courville, TPAMI 2013) review paper
- (Alain & Bengio ICLR 2013; Bengio et al, NIPS 2013)

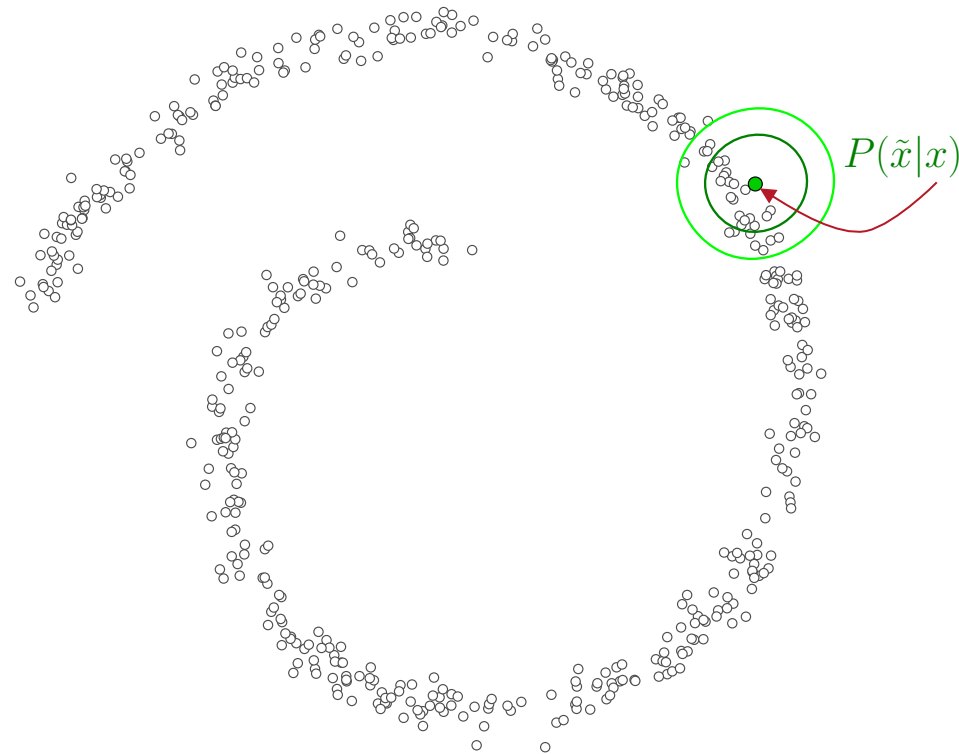


Learning with a simpler normalization constant, a nearly unimodal conditional distribution instead of a complicated multimodal one



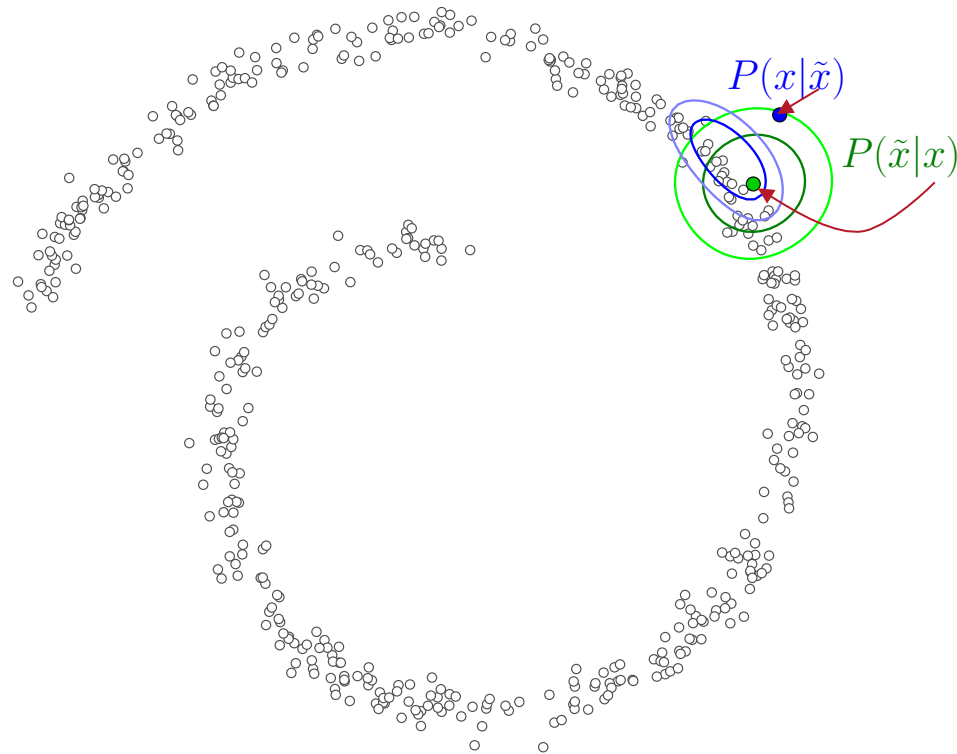
Thanks:
Jason Yosinski

Learning with a simpler normalization constant, a nearly unimodal conditional distribution instead of a complicated multimodal one



Thanks:
Jason Yosinski

Learning with a simpler normalization constant, a nearly unimodal conditional distribution instead of a complicated multimodal one

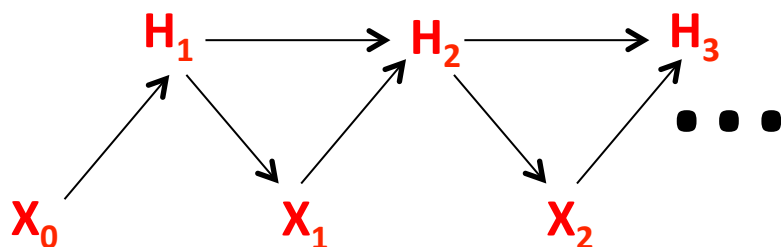


Thanks:
Jason Yosinski

Generative Stochastic Networks

- Generalizes the denoising auto-encoder training scheme
 - Introduce latent variables in the Markov chain (over X,H)
 - Instead of a fixed corruption process, have a deterministic function with parameters θ_1 and a noise source Z as input

$$H_{t+1} = f_{\theta_1}(X_t, Z_t, H_t)$$



$$H_{t+1} \sim P_{\theta_1}(H|H_t, X_t)$$

$$X_{t+1} \sim P_{\theta_2}(X|H_{t+1})$$

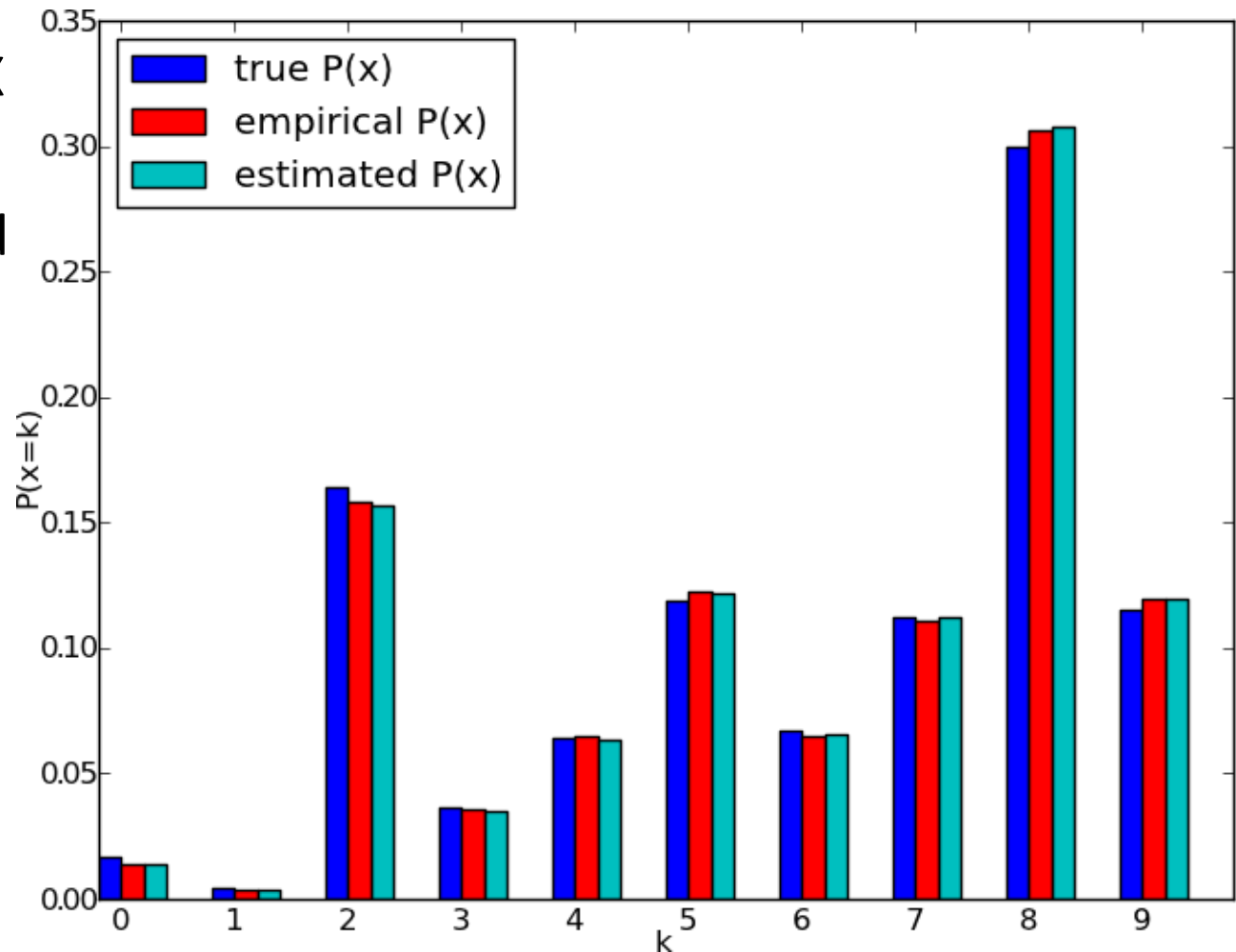
Consistent Estimator Theorem

Theorem:

If the parametrization is rich enough to have $P(X|H)$ a consistent estimator and the Markov chain is ergodic, then maximizing the expected log of $P_{\theta_2}(X|f_{\theta_1}(X, Z_{t-1}, H_{t-1}))$ makes the stationary distribution of the Markov chain a consistent estimator of the true data generating distribution.

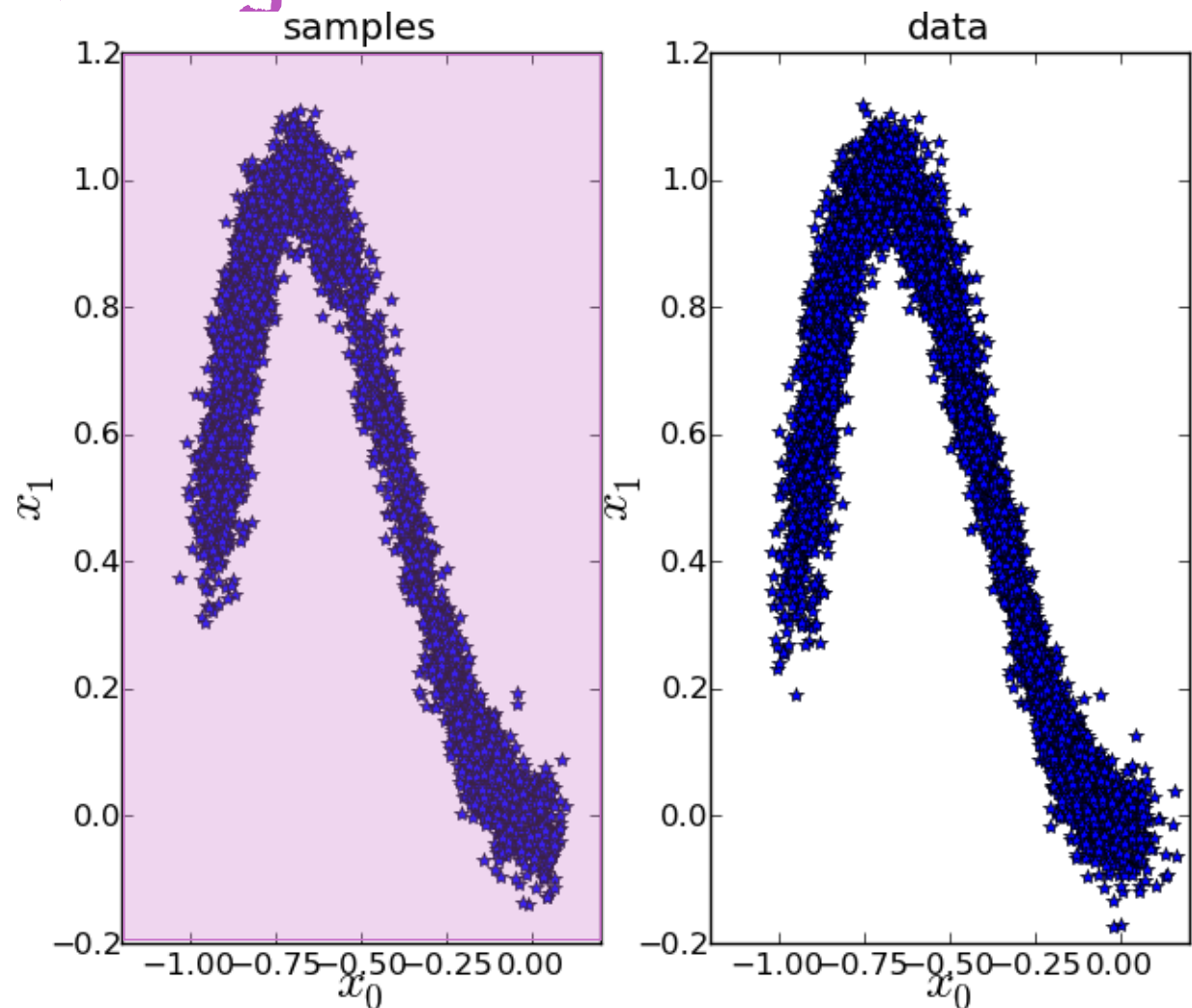
GSN Experiments: validating the theorem in a discrete non-parametric setting

- Discrete data, X in $\{0, \dots, 9\}$
- Corruption: add \pm small int.
- Reconstruction distribution = maximum likelihood estimator (counting)

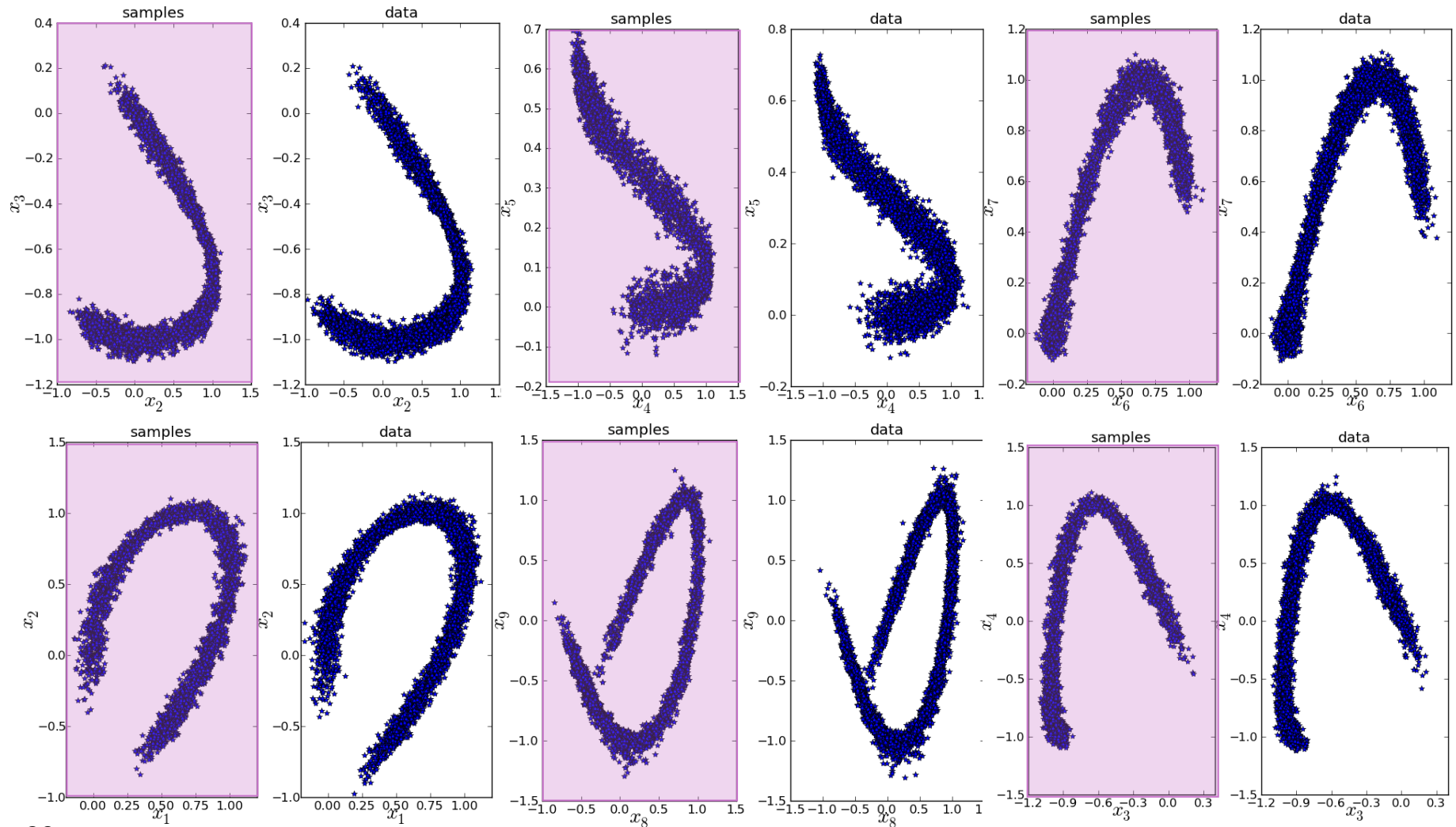


GSN Experiments: validating the theorem in a continuous non-parametric setting

- Continuous data, X in R^{10} , Gaussian corruption
- Reconstruction distribution = Parzen (mixture of Gaussians) estimator
- 5000 training examples, 5000 samples
- Visualize a pair of dimensions

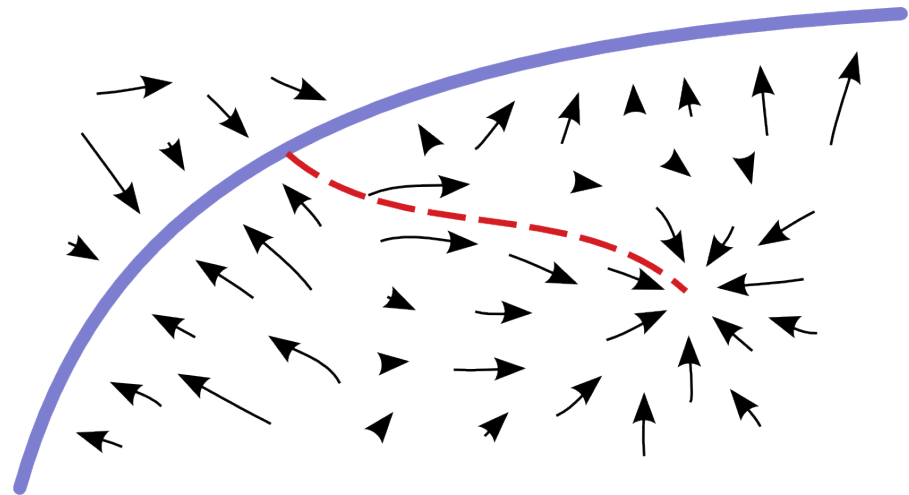


GSN Experiments: validating the theorem in a continuous non-parametric setting



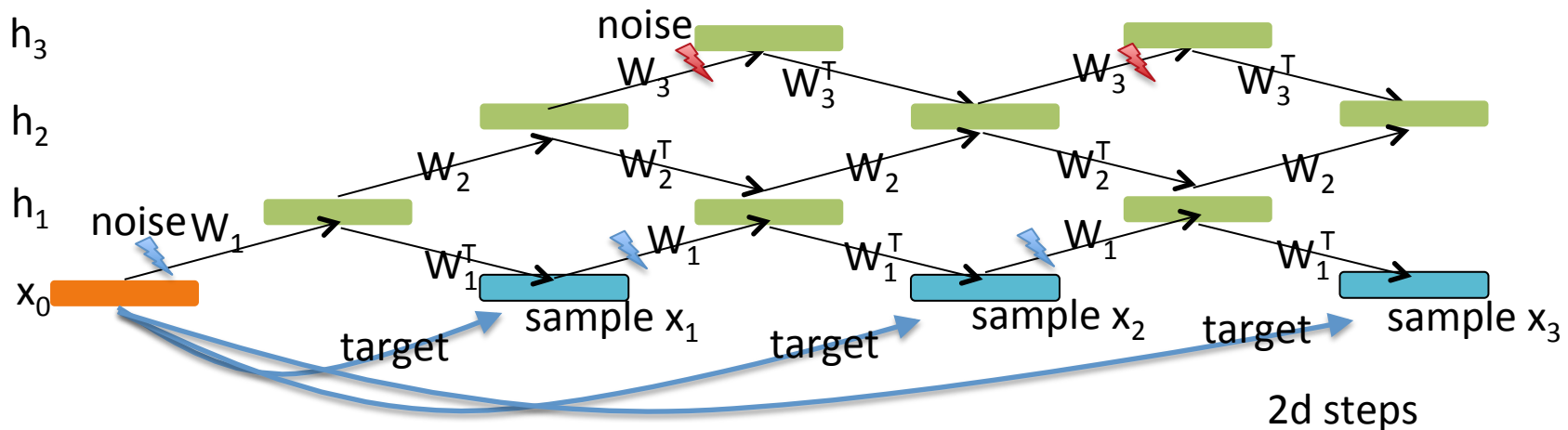
The Walkback Training Procedure

- Analogous to Contrastive Divergence, but NOT an approximation
- For any given operator T , create operator $T' = T^k$
- Maximize the probability of reconstructing data example x after applying $T' = T^k$
- Provably same solution
- Seeks out spurious modes and destroys them!



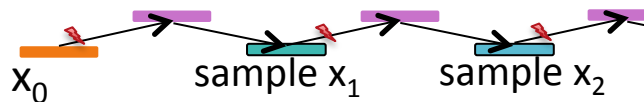
GSN Emulating a Deep Boltzmann Machine

- Noise injected in input and hidden layers
- Trained to max. reconstruction prob. of example at each step
- Depth $d \rightarrow k=2d$ walkback steps
- **Example** structure inspired from the DBM Gibbs chain:

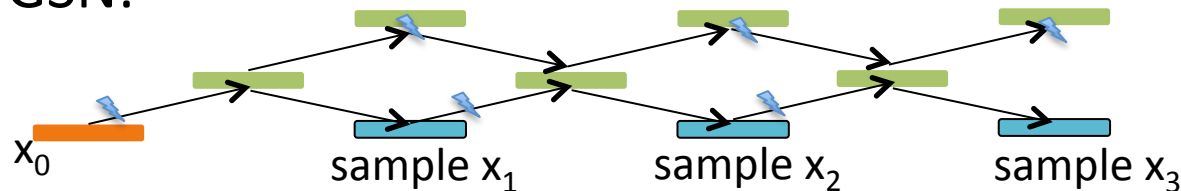


Experiments: Shallow vs Deep

- Shallow (DAE), no recurrent path at higher levels, state=X only




- Deep GSN:



Quantitative Evaluation of Samples

- Previous procedure for evaluating samples (Breuleux et al 2011, Rifai et al 2012, Bengio et al 2013):
 - Generate 10000 samples from model
 - Use them as training examples for Parzen density estimator
 - Evaluate its log-likelihood on MNIST test data



	GSN-2	DAE	RBM	DBM-3	DBN-2	MNIST
LOG-LIKELIHOOD	214	-152	-244	32	138	24
STANDARD ERROR	1.1	2.2	54	1.9	2.0	1.6

Question Answering, Missing Inputs and Structured Output

- Once trained, a GSN can sample from any conditional over subsets of its inputs, so long as we use the conditional associated with the reconstruction distribution and clamp the right-hand side variables.

Proposition 1. *If a subset $x^{(s)}$ of the elements of X is kept fixed (not resampled) while the remainder $X^{(-s)}$ is updated stochastically during the Markov chain of corollary 2, but using $P(X_{t+1}|f(X_t, Z_t), X_{t+1}^{(s)} = x^{(s)})$, then the asymptotic distribution π_n produces samples of $X^{(-s)}$ from the conditional distribution $\pi_n(X^{(-s)}|X^{(s)} = x^{(s)})$.*

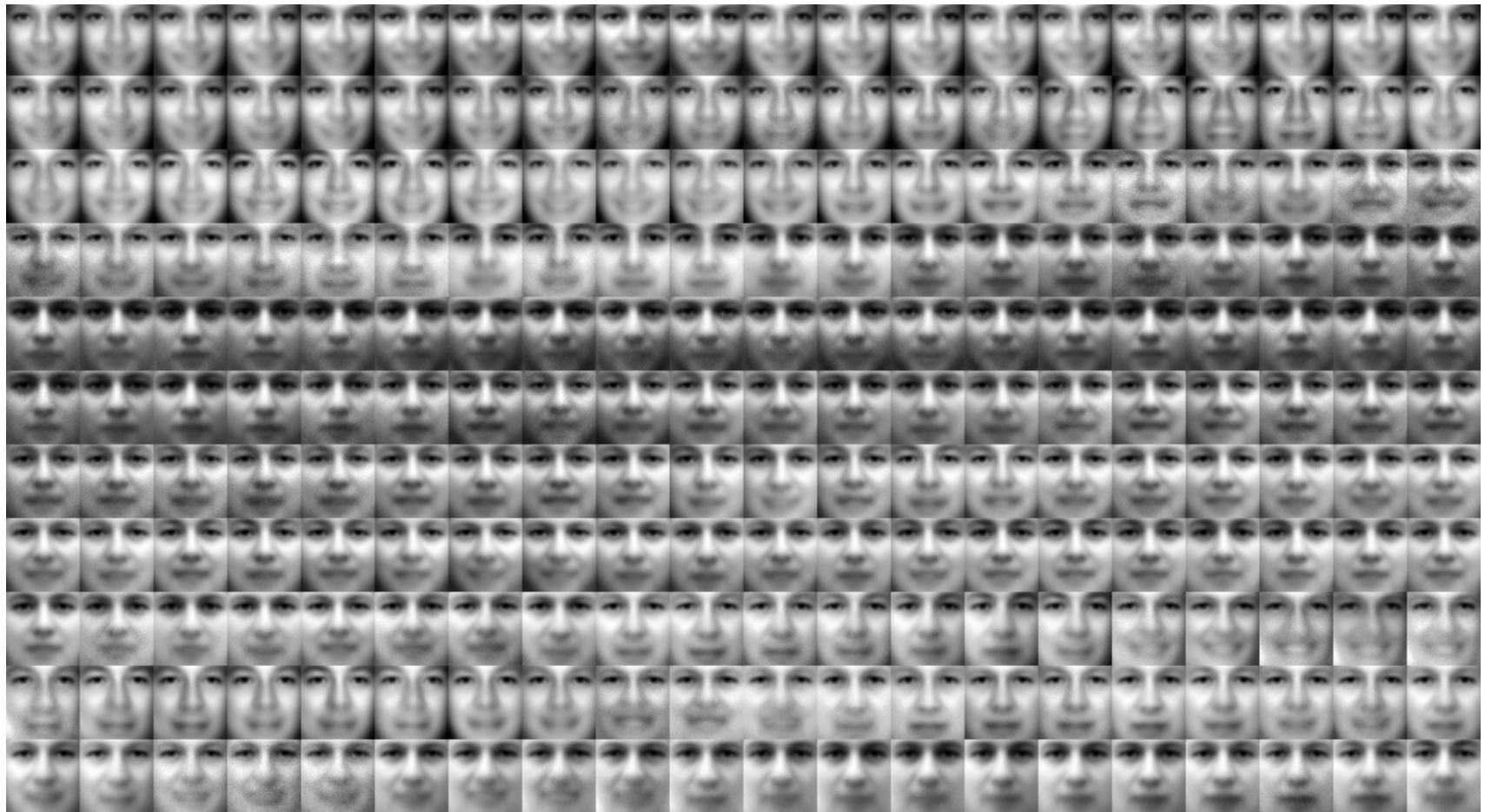
Experiments: Structured Conditionals

- Stochastically fill-in missing inputs, sampling from the chain that generates the conditional distribution of the missing inputs given the observed ones (notice the fast burn-in!)



Not Just MNIST: experiments on TFD

- 3 hidden layer model, consecutive samples:

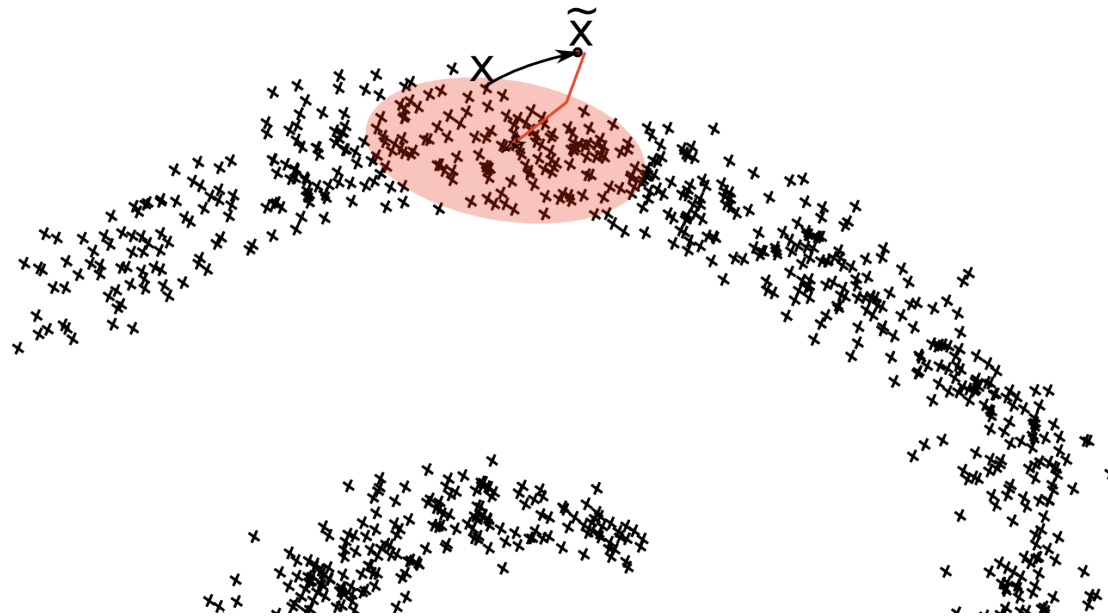


A Proper Generative Model for Dependency Networks

- Dependency networks (Heckerman et al 2000) estimate separate $P_{\theta_i}(X_i \mid X_{-i})$ not guaranteed to be conditionals of a unique joint
- Heckerman et al's sampling procedure iterates over i , but that is a GSN that is not guaranteed to be ergodic (it is periodic), with latent variable = (i, X)
- Randomly choosing which i to resample makes a proper GSN where the noise source chooses i independently
- Defines a unique joint distribution = stationary distr. of chain (which averages out over resampling orders)

Future Work: Multi-Modal Reconstruction Distributions

- All experiments: unimodal (factorial) reconstruction distribution
- Theorems require potentially multimodal one
- In the limit of small noise, unimodal is enough (Alain & Bengio 2013)



Conclusions

- **Radically different approach to probabilistic unsupervised learning of generative models through learning a transition operator**
 - **Address mode mixing with depth (latent variable)**
 - **Avoid marginalization during training**
- Consistent estimator
- Can be used to handle missing inputs or structured outputs
- Easy to train and sample from, hard to compute $P(x)$

LISA team: **Merci! Questions?**



LISA team: **Merci! Questions?**

