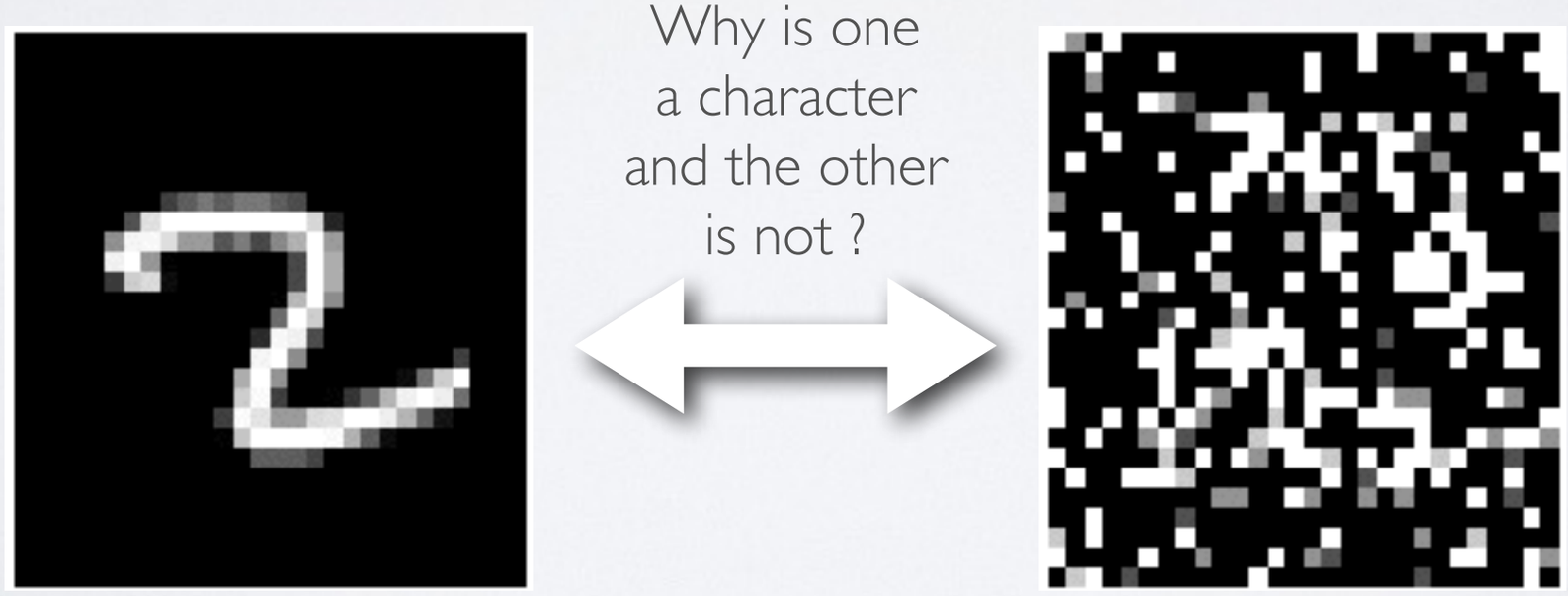
Deep Learning for Distribution Estimation

Hugo Larochelle (@hugo_larochelle)
Twitter / Université de Sherbrooke

Joint work with Iain Murray, Benigno Uria, Yin Zheng, Stanislas Lauly, Mathieu GermainKarol Gregor

DISTRIBUTION ESTIMATION

- **Task:** produce a estimator of $p(\mathbf{x})$ based on samples from it
  - hard problem as dimensionality of observations $\mathbf{x}$ increases
  - very general problem
  - often used as a proxy for feature learning
  - can provide regularization cues for another model



character image

random image

OUTLINE

- Neural Autoregressive Distribution Estimator (NADE)
  - binary observations (with Iain Murray)
  - real-valued observations (with Benigno Uria and Iain Murray)
  - multinomial observations (with Stanislas Lauly and Yin Zheng)
  - deep NADE (with Benigno Uria and Iain Murray)

- Masked Autoencoder Distribution Estimator (MADE)
  - binary observations (with Mathieu Germain, Karol Gregor and Iain Murray)

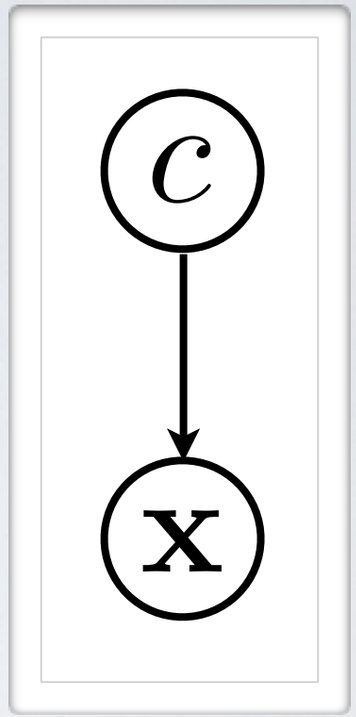
OUTLINE

- Neural Autoregressive Distribution Estimator (NADE)
  - **binary observations (with Iain Murray)**
  - real-valued observations (with Benigno Uria and Iain Murray)
  - multinomial observations (with Stanislas Lauly and Yin Zheng)
  - deep NADE (with Benigno Uria and Iain Murray)

- Masked Autoencoder Distribution Estimator (MADE)
  - binary observations (with Mathieu Germain, Karol Gregor and Iain Murray)

MIXTURE MODEL

Distribution : $$p(\mathbf{x}) = \sum_{i=1}^{C} p(\mathbf{x}|c=i)p(c=i)$$

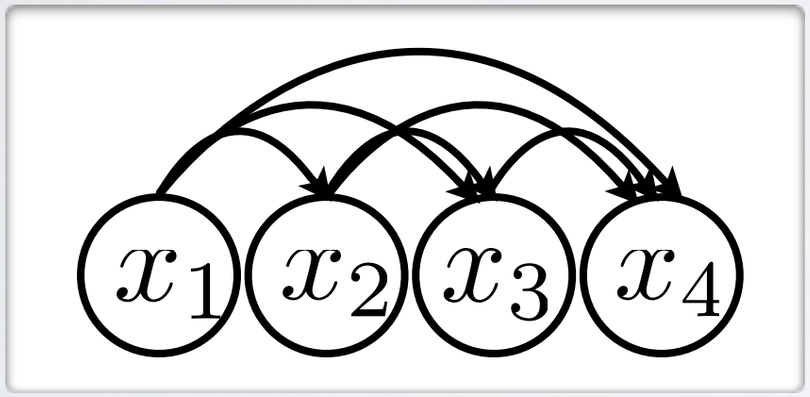


- Simple, well understood model
- Performance can be disappointing

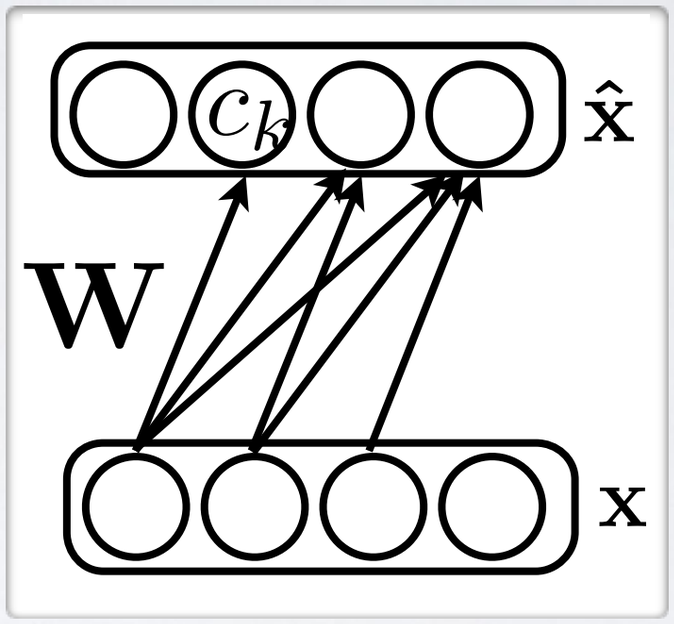
FULLY VISIBLE BAYESIAN NETWORKS

General graphical model

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



Fully Visible Sigmoid Belief Net (FVSBN)



$\widehat{x}_k$

$\updownarrow$

$p(x_k = 1 | \mathbf{x}_{<k})$

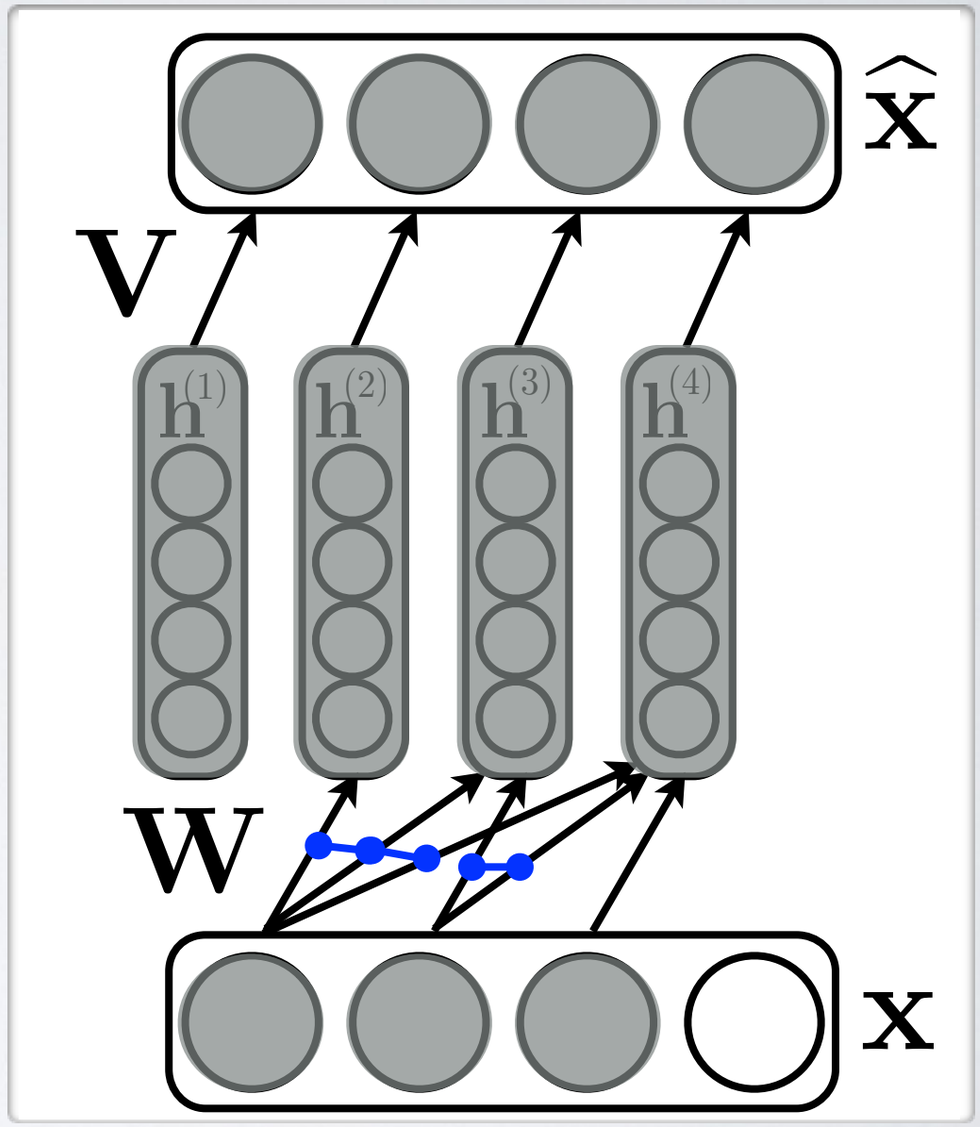
- Models $p(x_k = 1 | \mathbf{x}_{<k})$ as logistic regression

NEURAL AUTOREGRESSIVE DISTRIBUTION ESTIMATOR

- **NADE:** autoencoder-like neural network wired to learn $p(x_k|\mathbf{x}_{<k})$

$$\mathbf{h}^{(k)} = \text{sigm}\left(\mathbf{b} + \mathbf{W}_{.,<k}\mathbf{x}_{<k}\right)$$
$$\widehat{x}_k = \text{sigm}\left(c_k + \mathbf{V}_{k,.}\mathbf{h}^{(k)}\right)$$

  - in words, each conditional $p(x_k|\mathbf{x}_{<k})$ is modeled by the **same** neural network



- We can leverage the fact that:

$$(\mathbf{b} + \mathbf{W}_{.,<k+1}\mathbf{x}_{<k+1}) - (\mathbf{b} + \mathbf{W}_{.,<k}\mathbf{x}_{<k}) = \mathbf{W}_{.,k+1}x_{k+1}$$

NEURAL AUTOREGRESSIVE DISTRIBUTION ESTIMATOR

- **NADE:** training by maximizing the average log-likelihood

$$\frac{1}{T}\sum_{t=1}^{T}\log p(\mathbf{x}^{(t)}) = \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{D}\log p(x_k^{(t)}|\mathbf{x}_{<k}^{(t)})$$

- Advantages
  - efficient: computations are in $O(HD)$
  - could make use of second-order optimizers
  - easily extendable to other types of observations (reals, multinomials)
- What order for the inputs
  - random order works fine!

EXPERIMENTS

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**

EXPERIMENTS

- On a binarized version of MNIST:

	Model	Log. Like.
	MoB*	-137.64
Intractable	RBM (CD1)*	≈ -125.53
Intractable	RBM (CD3)*	≈ -105.5
Intractable	RBM (CD25)*	≈ -86.34
	FVSBN	-97.45
	NADE	-88.86

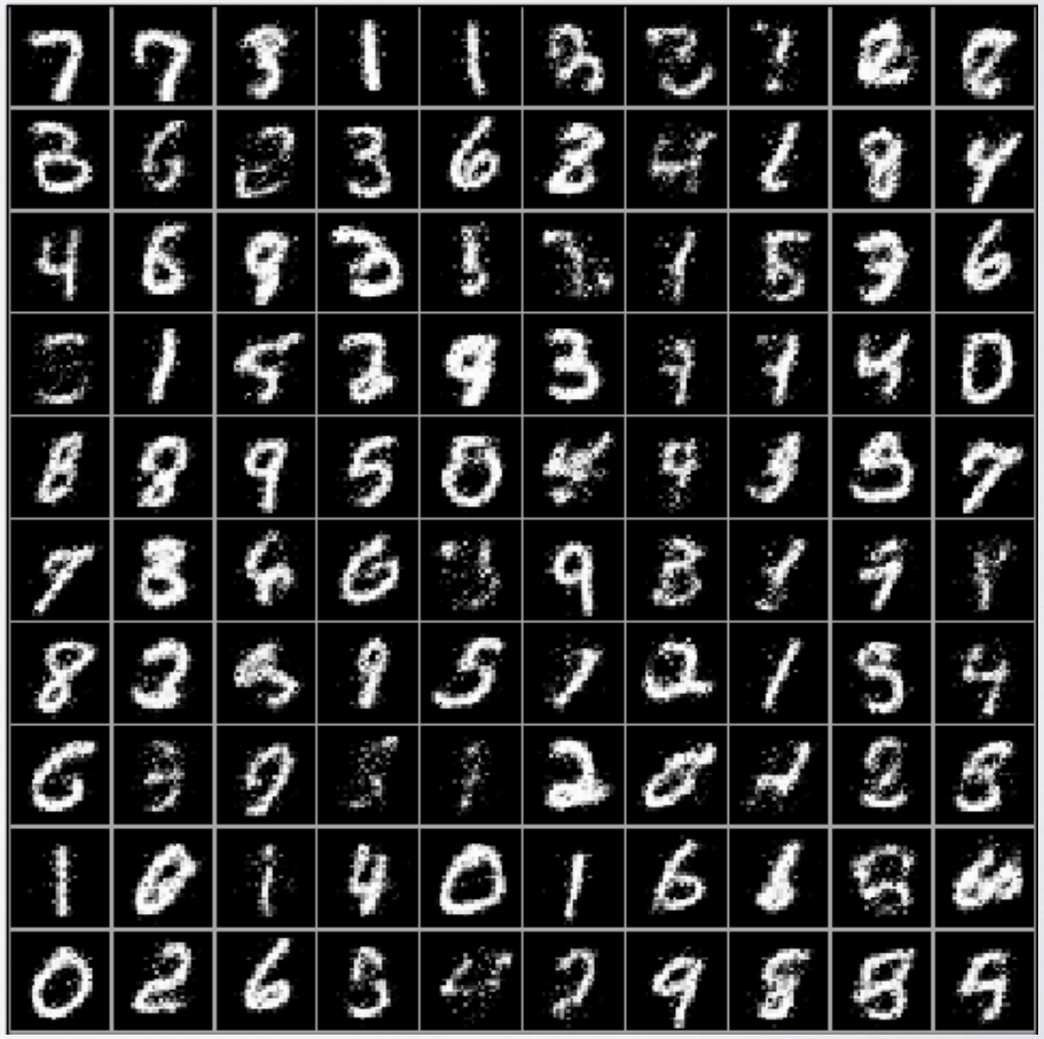
Samples

* : taken from Salakhutdinov and Murray (2008)

EXPERIMENTS

- On a binarized version of MNIST:

	Model	Log. Like.
Intractable	MoB*	-137.64
	RBM (CD1)*	≈ -125.53
	RBM (CD3)*	≈ -105.5
	RBM (CD25)*	≈ -86.34
	FVSBN	-97.45
	NADE	-88.86



Probabilities

* : taken from Salakhutdinov and Murray (2008)

OUTLINE

- Neural Autoregressive Distribution Estimator (NADE)
  - binary observations (with Iain Murray)
  - **real-valued observations (with Benigno Uria and Iain Murray)**
  - multinomial observations (with Stanislas Lauly and Yin Zheng)
  - deep NADE (with Benigno Uria and Iain Murray)

- Masked Autoencoder Distribution Estimator (MADE)
  - binary observations (with Mathieu Germain, Karol Gregor and Iain Murray)

REAL-VALUED NADE

(Uria, Murray, Larochelle)

- **RNADE:** models real-valued observations by
  - outputting the parameters of a mixture model for $p(x_k|\mathbf{x}_{<k})$

Means

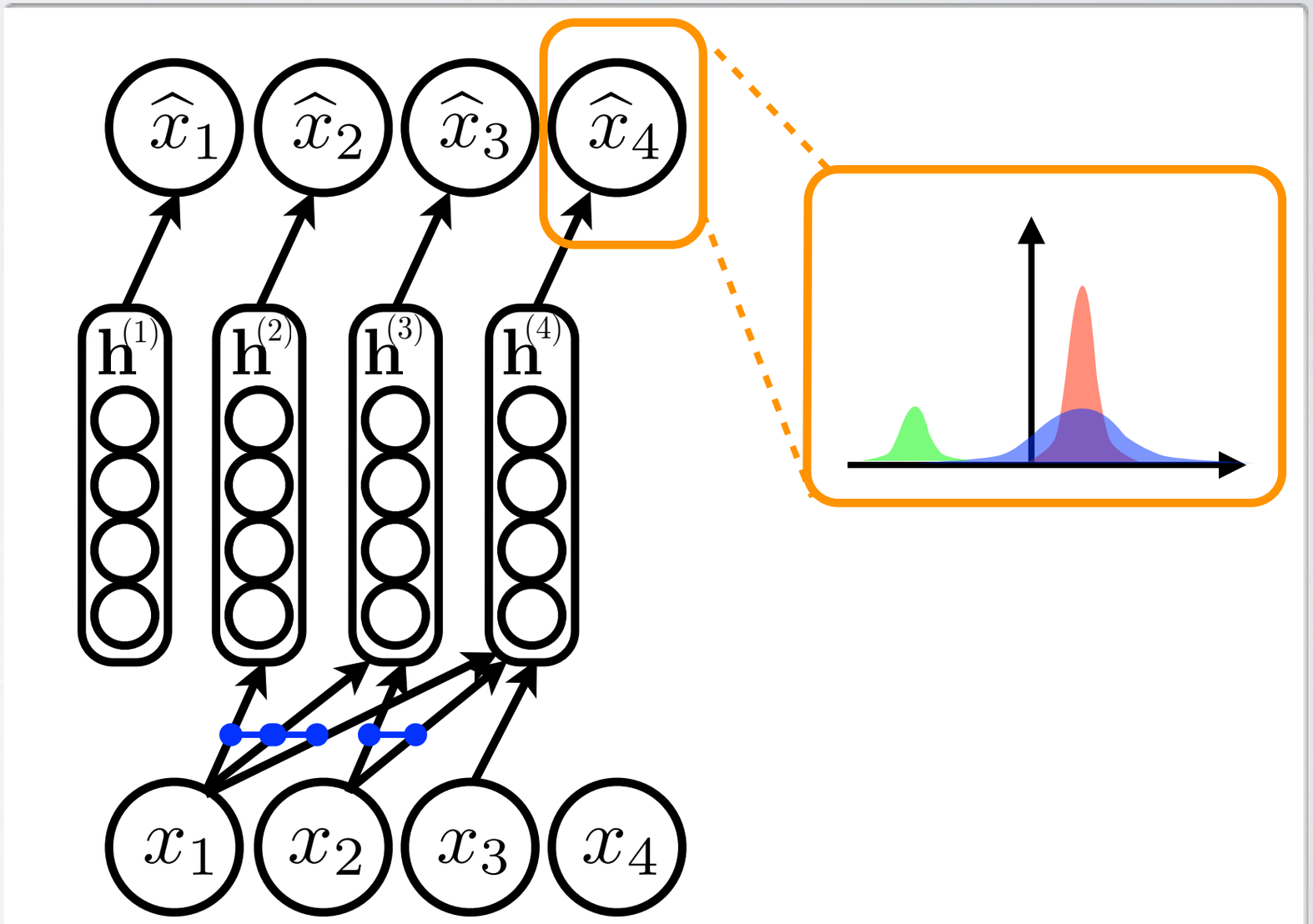
$$\mu_{ik} = b_k^{\mu_i} + \mathbf{V}_{k,.}^{\mu_i}\mathbf{h}^{(k)}$$

Std. deviations

$$\sigma_{ik} = \exp(b_k^{\sigma_i} + \mathbf{V}_{k,.}^{\sigma_i}\mathbf{h}^{(k)})$$

Mixing weights

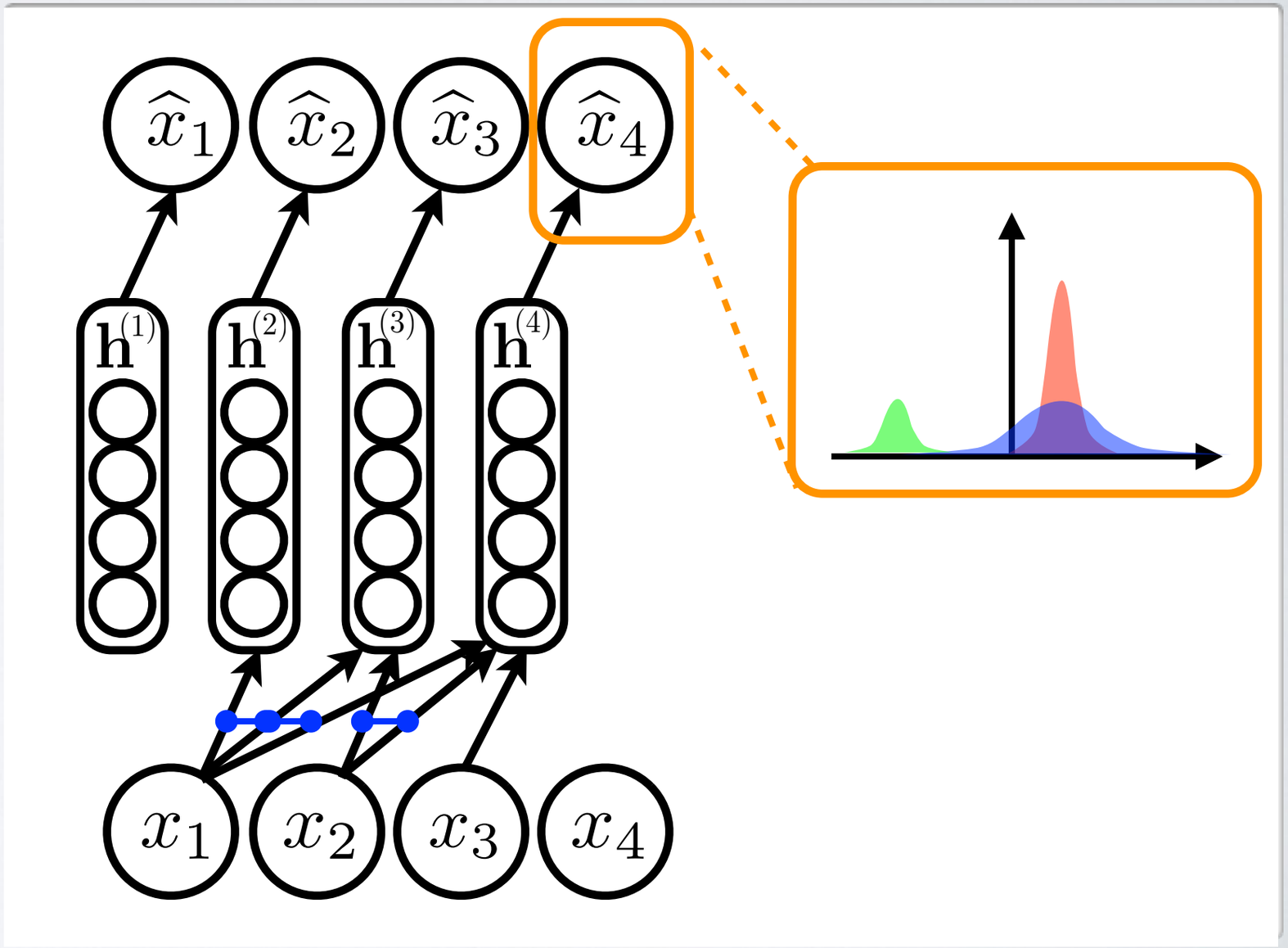
$$\pi_{ik} = \mathrm{softmax}(b_k^{\pi_i} + \mathbf{V}_{k,.}^{\pi_i}\mathbf{h}^{(k)})$$



REAL-VALUED NADE

(Uria, Murray, Larochelle)

- **RNADE:** models real-valued observations by
  - gradient descent training
  - rectified linear units
  - learned a position-specific scaling factor, applied before the activation function
  - we also experimented with mixture of Laplacians



EXPERIMENTS

(Uria, Murray, Larochelle)

- **UCI datasets:** comparison with
  - mixture of Gaussians
  - mixture of factor analyzers

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

EXPERIMENTS

(Uria, Murray, Larochelle)

- **Natural image patches:** comparison with
  - mixture of Gaussians

Model	Training LogL	Test LogL
MoG $K = 200$ (Z&W)	161.9	**152.8**
MoG $K = 100$	152.8	144.7
MoG $K = 200$	159.3	150.4
MoG $K = 300$	159.3	150.4
RNADE-MoG $K = 5$	158.0	149.1
RNADE-MoG $K = 10$	160.0	151.0
RNADE-MoG $K = 20$	158.6	149.7
RNADE-MoL $K = 5$	150.2	141.5
RNADE-MoL $K = 10$	149.7	141.1
RNADE-MoL $K = 20$	150.1	141.5
RNADE-MoG $K = 10$ (sigmoid h. units)	155.1	146.4
RNADE-MoL $K = 10$ (1024 units, 400 epochs)	161.1	152.1

EXPERIMENTS

(Uria, Murray, Larochelle)

- **TIMIT spectrograms:** comparison with
  - mixture of Gaussians

Model	Training LogL	Test LogL
MoG $N=50$	111.6	110.4 (SE: 0.3)
MoG $N=100$	113.4	112.0 (SE: 0.3)
MoG $N=200$	113.9	112.5 (SE: 0.3)
MoG $N=300$	114.1	112.5 (SE: 0.3)
RNADE-MoG $K=10$	125.9	123.9 (SE: 0.3)
RNADE-MoG $K=20$	126.7	**124.5** (SE: 0.3)
RNADE-MoL $K=10$	120.3	118.0 (SE: 0.3)
RNADE-MoL $K=20$	122.2	119.8 (SE: 0.3)

OUTLINE

- Neural Autoregressive Distribution Estimator (NADE)
  - binary observations (with Iain Murray)
  - real-valued observations (with Benigno Uria and Iain Murray)
  - **multinomial observations (with Stanislas Lauly and Yin Zheng)**
  - deep NADE (with Benigno Uria and Iain Murray)

- Masked Autoencoder Distribution Estimator (MADE)
  - binary observations (with Mathieu Germain, Karol Gregor and Iain Murray)

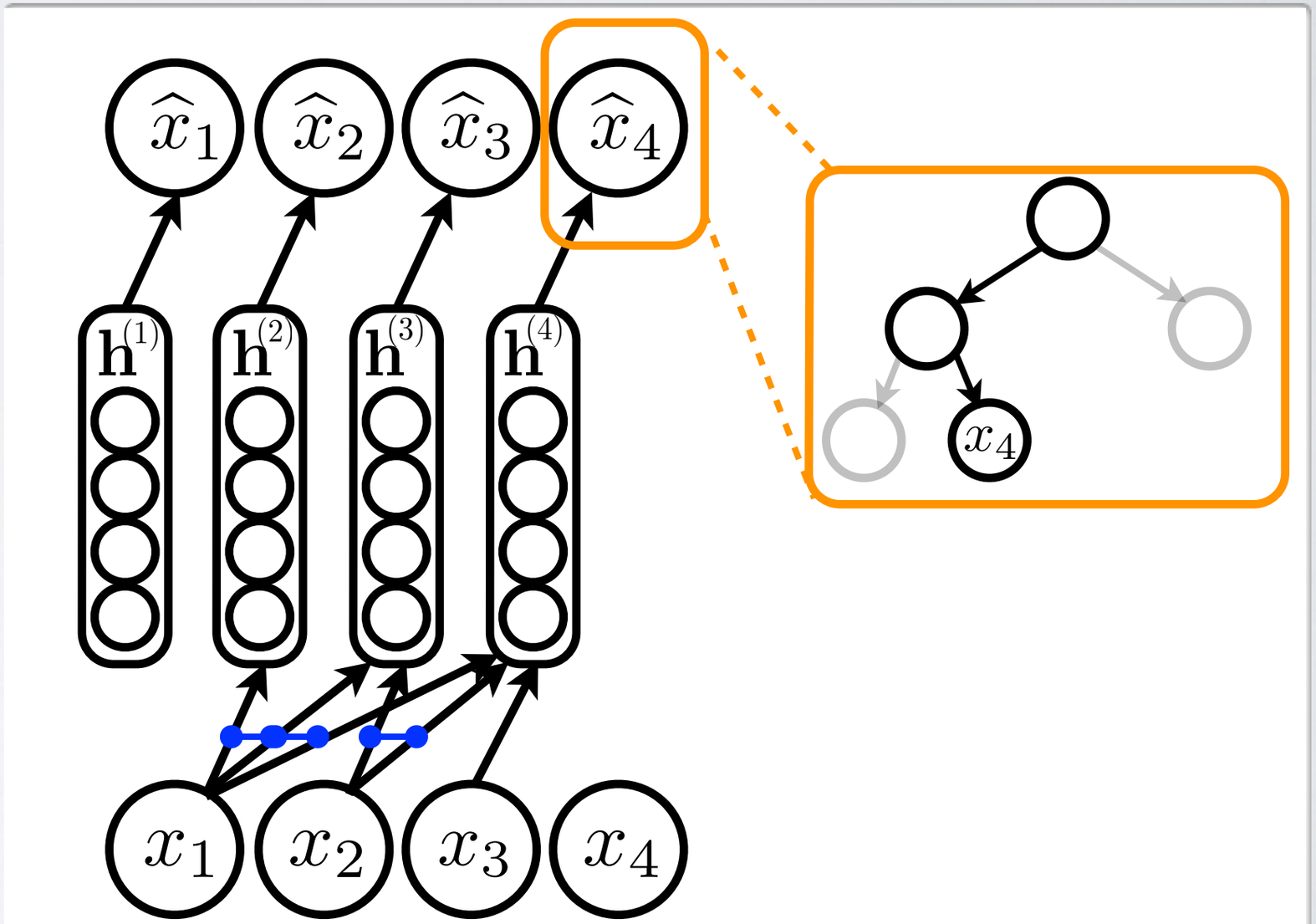
DOCUMENT NADE

(Larochelle, Lauly)

- **DocNADE:** models multinomial observations by
  - outputting a multinomial distribution
  - for efficiency, we use a tree structured multinomial

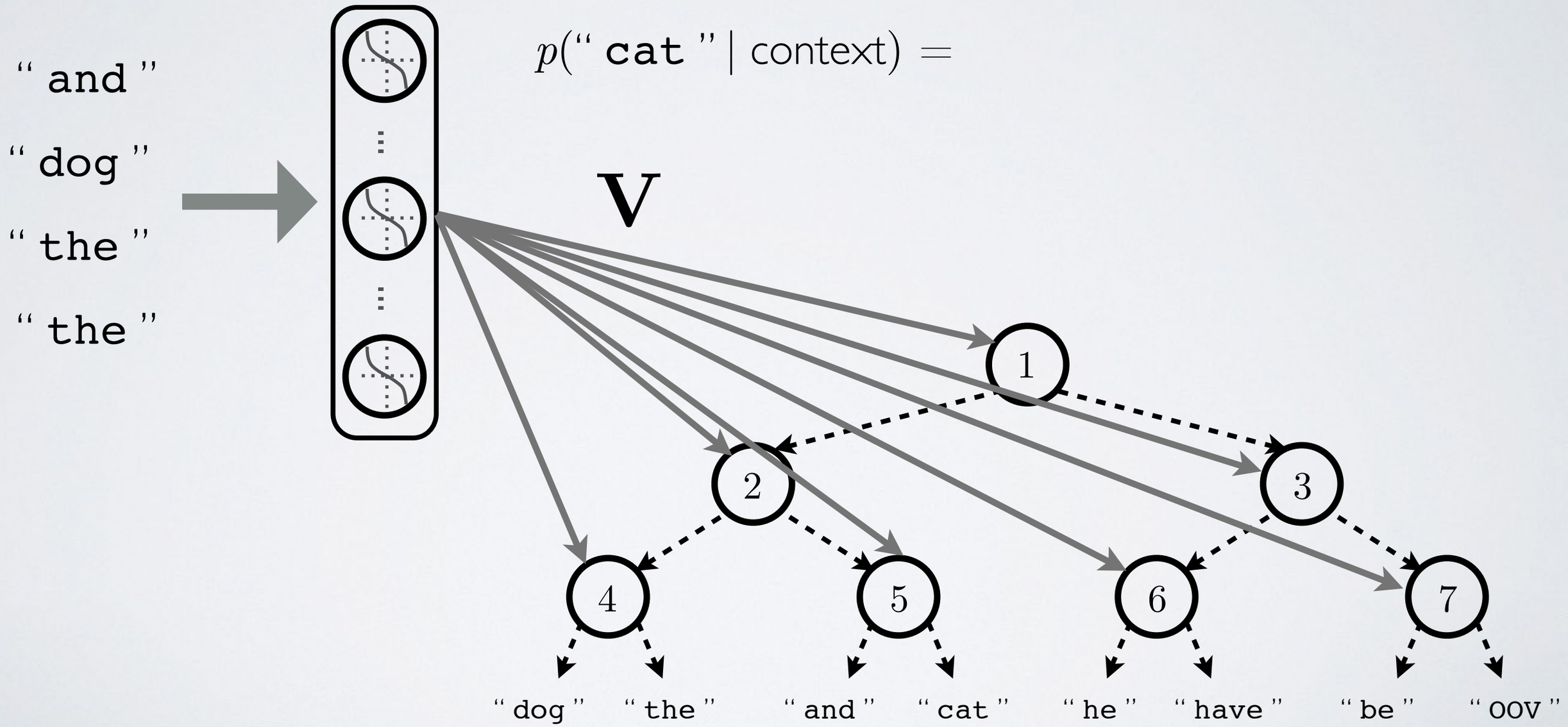
$$p(x_k = w|\mathbf{x}_{<k}) = \prod_{m=1}^{|\pi(x_k)|} p(\pi(x_k)_m|\mathbf{x}_{<k})$$

$$p(\pi(x_k)_m = 1|\mathbf{x}_{<k}) = \text{sigm}(c_{l(x_k)_m} + \mathbf{V}_{l(x_k)_m,.}\mathbf{h}^{(k)})$$



TREE STRUCTURED MULTINOMIAL

- Example: [“ **and** ”,“ **dog** ”,“ **the** ”,“ **the** ”,“ **cat** ”]



TREE STRUCTURED MULTINOMIAL

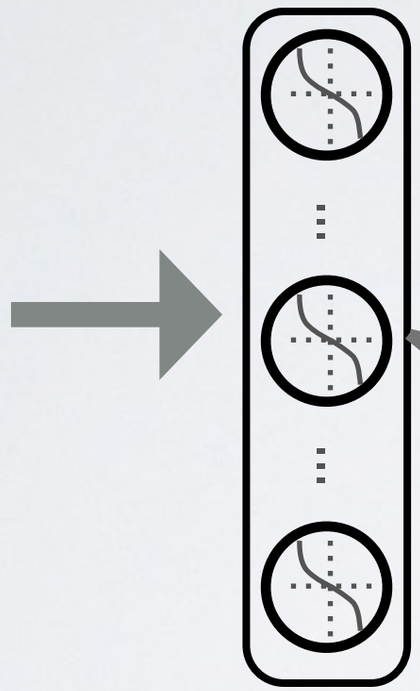
- Example: [“ `and` ”,“ `dog` ”,“ `the` ”,“ `the` ”,“ `cat` ”]

“ `and` ”

“ `dog` ”

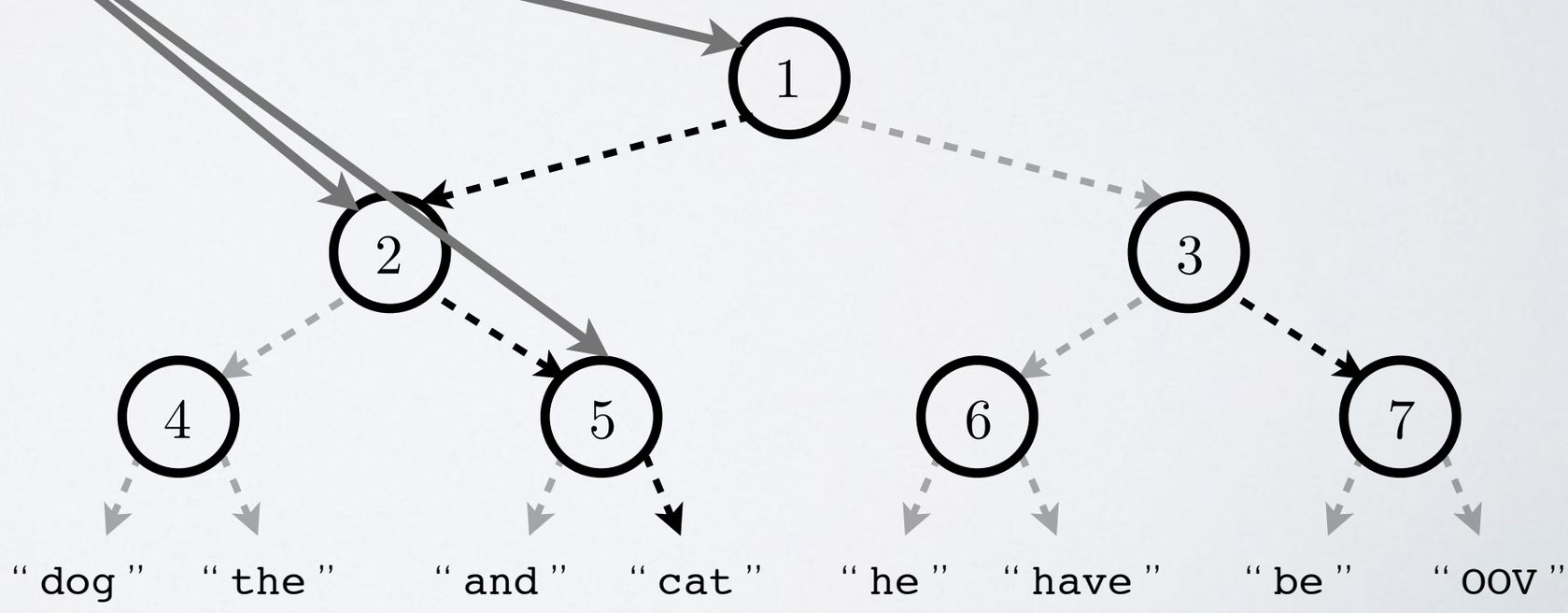
“ `the` ”

“ `the` ”



$\mathbf{V}$

$$p(\text{“ cat ”} \mid \text{context}) = p(\text{branch left at } 1 \mid \text{context}) \times p(\text{branch right at } 2 \mid \text{context}) \times p(\text{branch right at } 5 \mid \text{context})$$



TREE STRUCTURED MULTINOMIAL

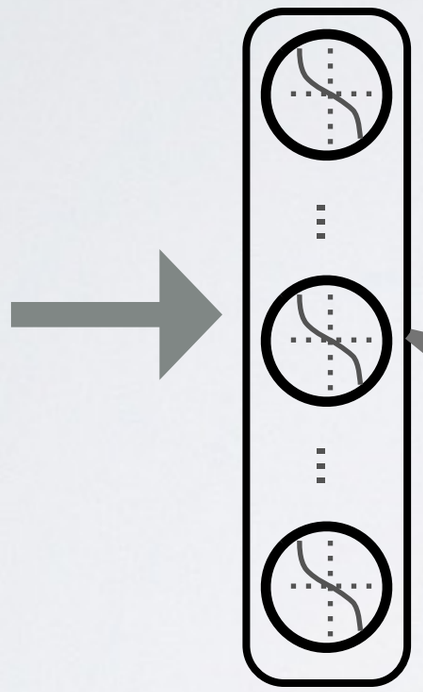
- Example: [“ `and` ”,“ `dog` ”,“ `the` ”,“ `the` ”,“ `cat` ”]

“ `and` ”

“ `dog` ”

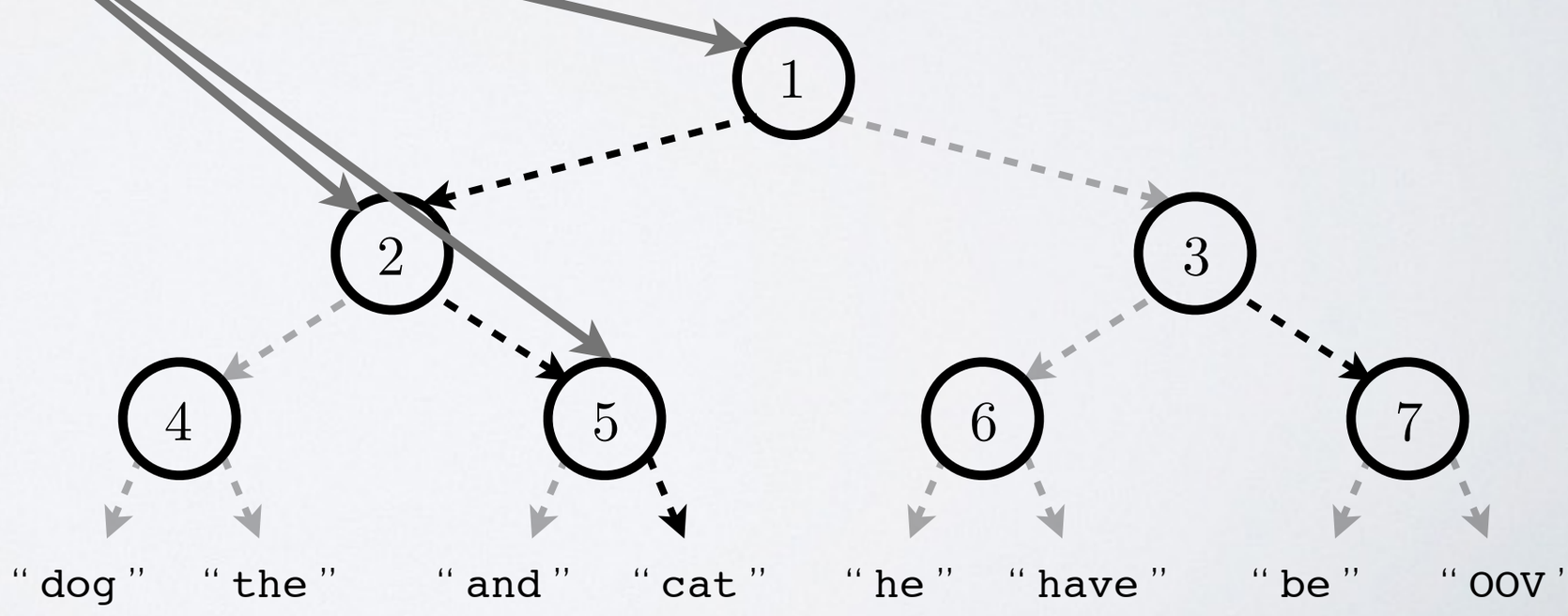
“ `the` ”

“ `the` ”



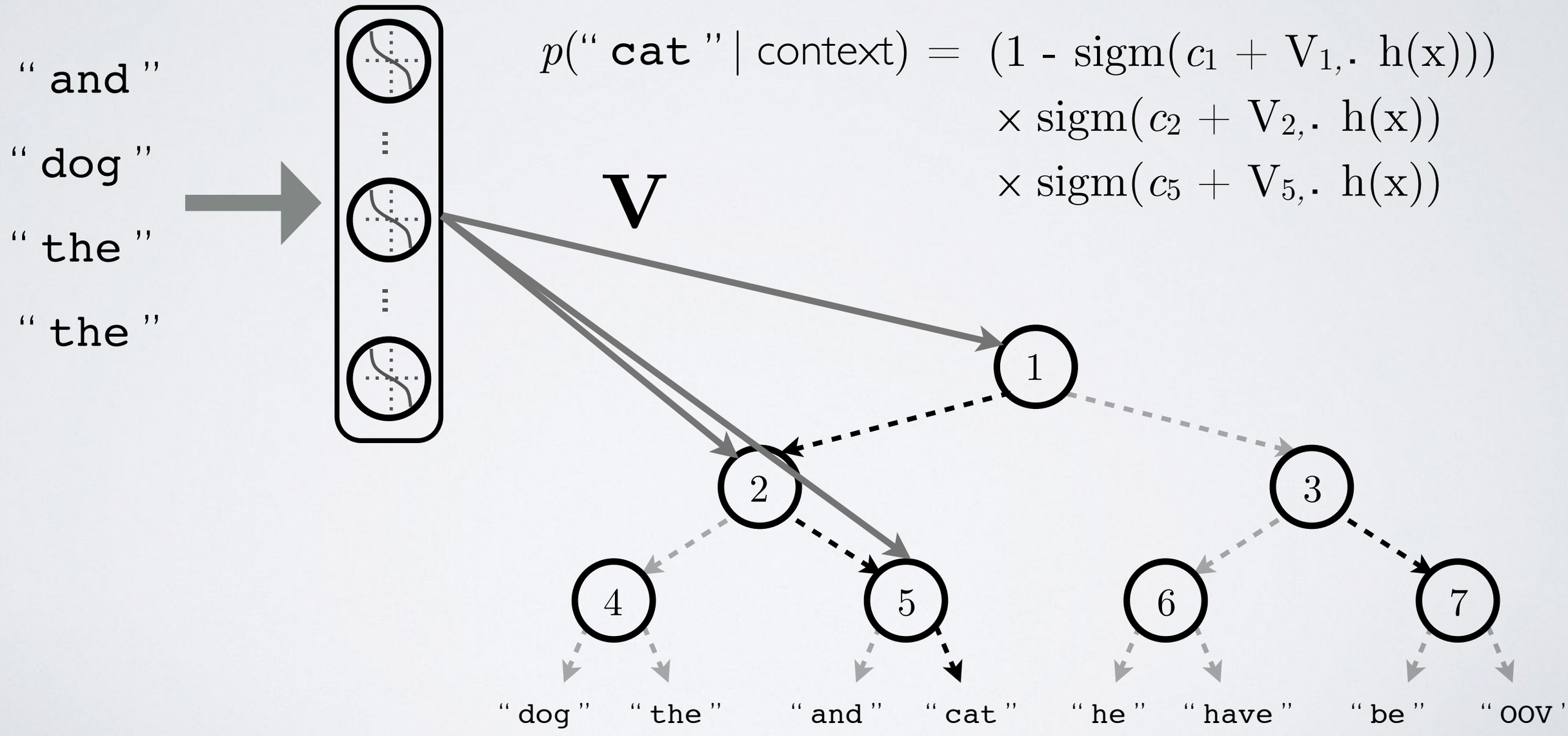
$\mathbf{V}$

$$p(\text{“ cat ”} \mid \text{context}) = (1\text{-}p(\text{branch right at } 1 \mid \text{ context})) \times p(\text{branch right at } 2 \mid \text{ context}) \times p(\text{branch right at } 5 \mid \text{ context})$$



TREE STRUCTURED MULTINOMIAL

- Example: [“ and ”,“ dog ”,“ the ”,“ the ”,“ cat ”]



$$p(\text{“ cat ”} \mid \text{context}) = (1 - \text{sigm}(c_1 + \text{V}_{1,}.\ \text{h(x)})) \times \text{sigm}(c_2 + \text{V}_{2,}.\ \text{h(x)}) \times \text{sigm}(c_5 + \text{V}_{5,}.\ \text{h(x)})$$

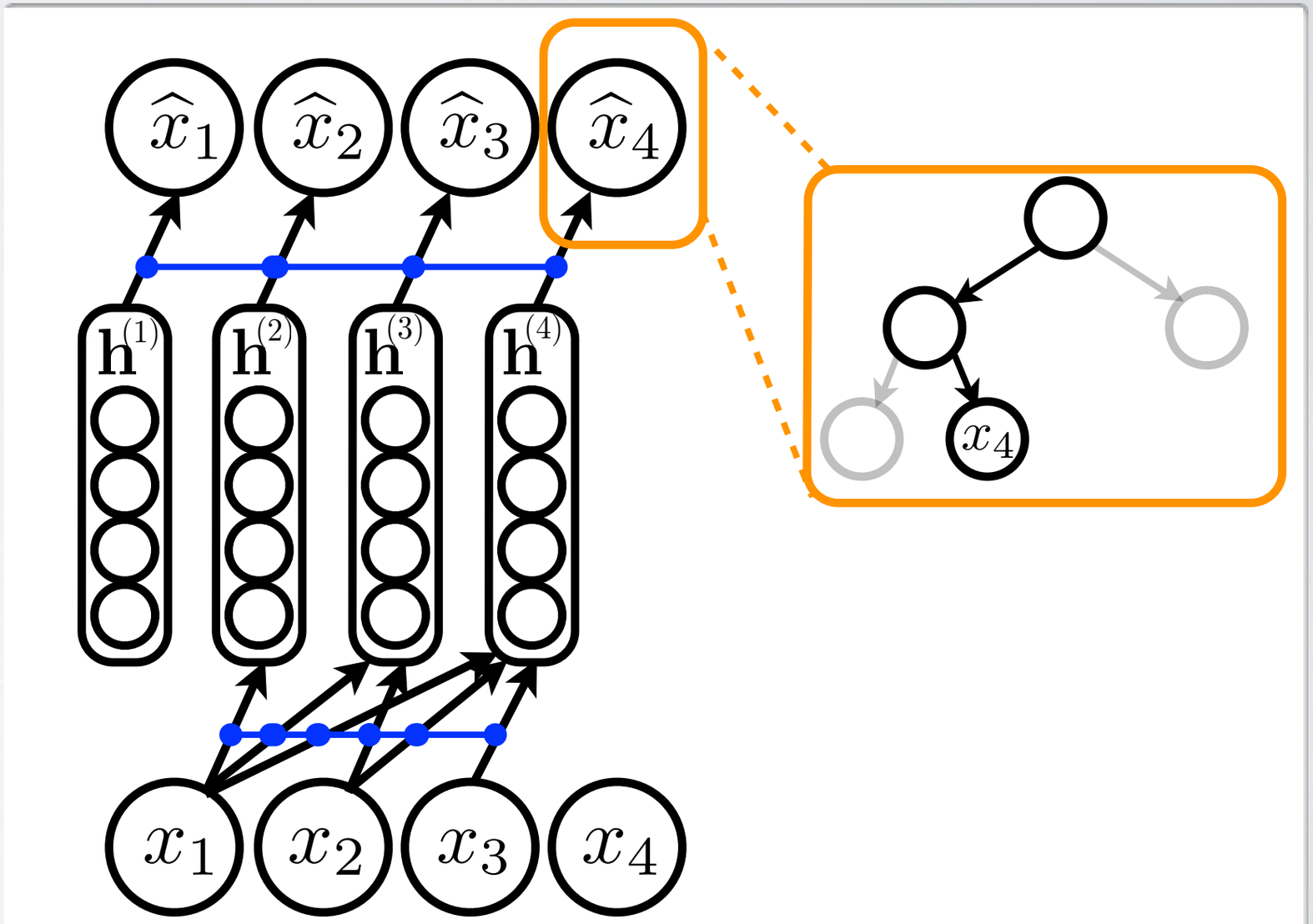
DOCUMENT NADE

(Larochelle, Lauly)

- **DocNADE:** models multinomial observations by
  - for bags of words, we have
    - unordered words
    - variable number of words
  - we share all parameters across conditionals (not position dependent)

$$\mathbf{h}^{(k)} = \text{sigm}\left(\mathbf{b} + \sum_{i<k} \mathbf{W}_{\cdot,x_i}\right)$$

word ID



DOCUMENT NADE

(Larochelle, Lauly)

- **Text modeling**
  - we shuffle the word order at every update
    - learning not to insert «intruder words» at any position in the document
    - useful if word order is not available

  - we evaluate the model on two measures
    - perplexity (normalized by document length)
    - information retrieval

EXPERIMENTS

(Larochelle, Lauly)

- **Text modeling:** comparison with
  - LDA
  - Replicated Softmax

Perplexity Results

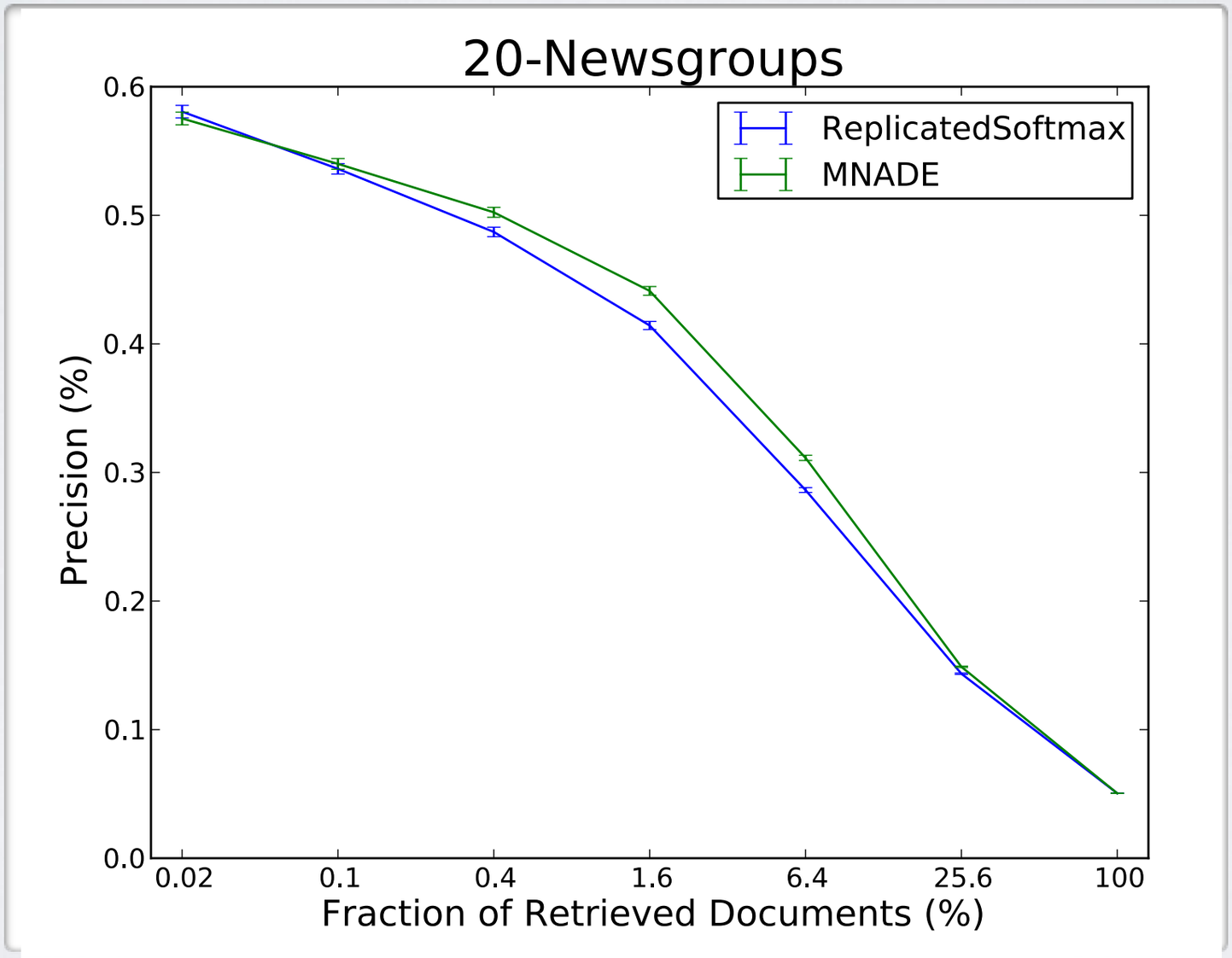
Data Set	LDA (50)	LDA (200)	Replicated Softmax (50)	DocNADE (50)	DocNADE St. Dev
20 Newsgroups	1091	1058	953	**896**	6.9
RCV1-v2	1437	1142	988	**742**	4.5

EXPERIMENTS

(Larochelle, Lauly)

- **Text modeling:** comparison with
  - Replicated Softmax

Information Retrieval



EXPERIMENTS

(Larochelle, Lauly)

- **Text modeling:** learned representation

Hidden unit topics			
jesus	shuttle	season	encryption
atheism	orbit	players	escrow
christianity	lunar	nhl	pgp
christ	spacecraft	league	crypto
athos	nasa	braves	nsa
atheists	space	playoffs	rutgers
bible	launch	rangers	clipper
christians	saturn	hockey	secure
sin	billion	pitching	encrypted
atheist	satellite	team	keys

EXPERIMENTS

(Larochelle, Lauly)

- **Text modeling:** learned representation

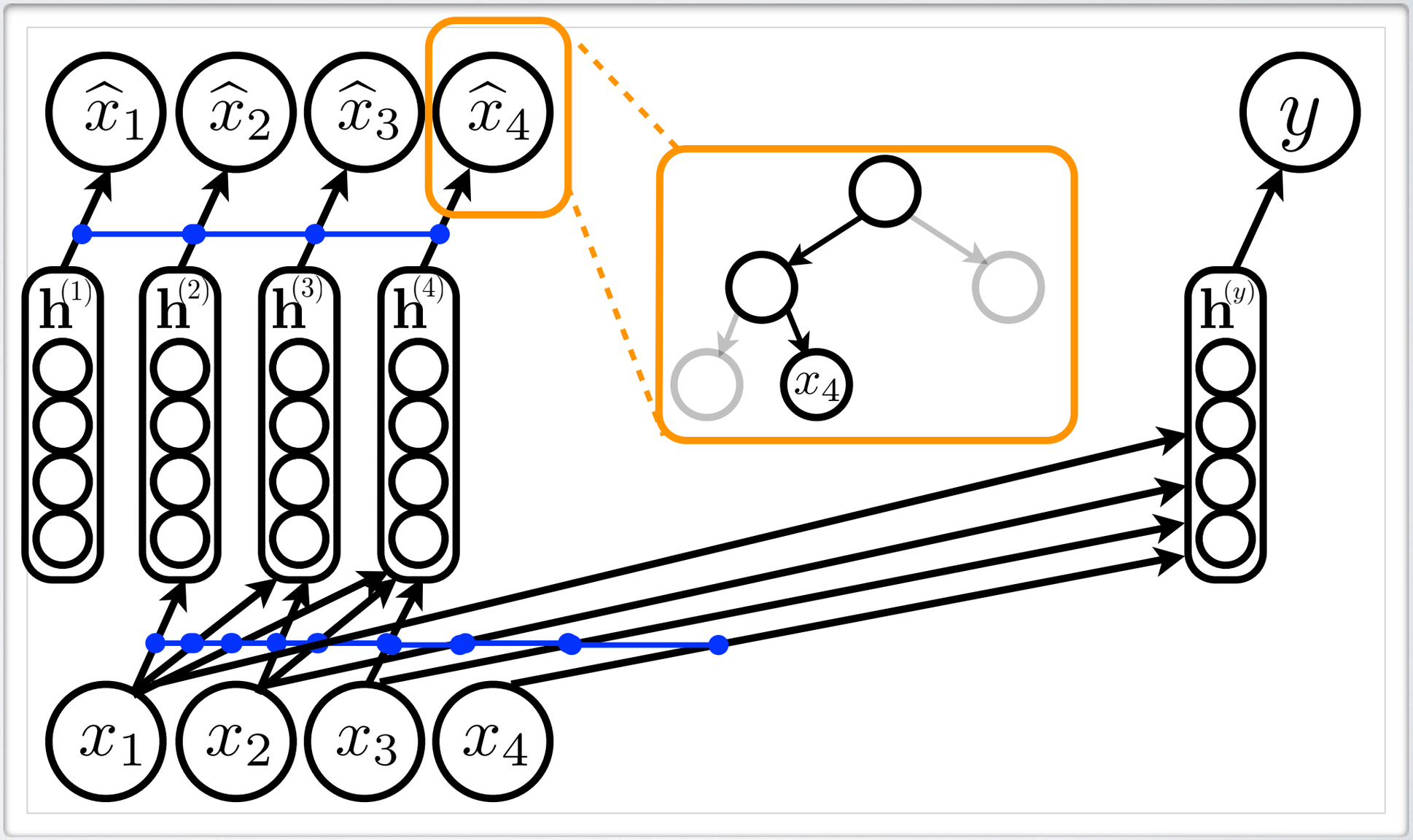
weapons	**medical**	**companies**	**define**	**israel**	**book**	**windows**
weapon	treatment	demand	defined	israeli	reading	dos
shooting	medecine	commercial	definition	israelis	read	microsoft
firearms	patients	agency	refer	arab	books	version
assault	process	company	make	palestinian	relevent	ms
armed	studies	credit	examples	arabs	collection	pc

- Timing for 1 training iteration (20 newsgroups / RCV1-v2)
  - DocNADE: 13 sec. / 726 sec.
  - Replicated Softmax CD1: 28 sec. / 4945 sec.
  - Replicated Softmax CD5: 60 sec. / 11000 sec.

SUPERVISED DOCNADE

(Zheng, Zhang, Larochelle)

- **Supervised DocNADE:** incorporate label by
  - adding a final, supervised layer



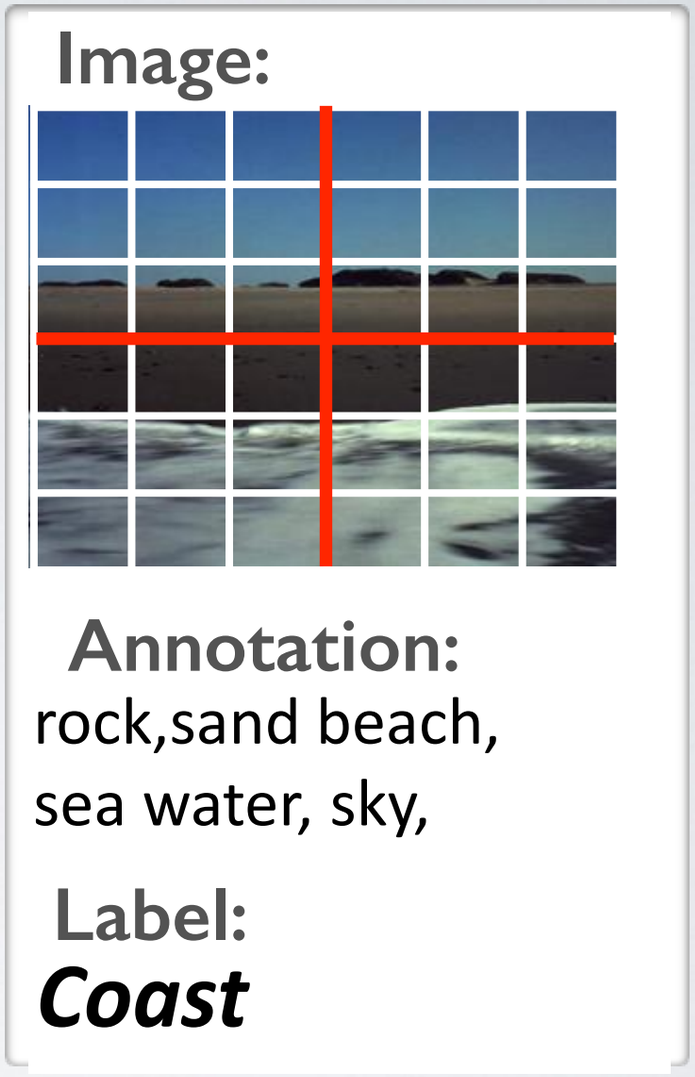
SUPERVISED DOCNADE

(Zheng, Zhang, Larochelle)

- **Scene classif/annotation**
  - images are converted into a bag of «visual» words
    - extracted by k-means on dense SIFT features
    - grid pooling: assign different word IDs for visual words in different regions
  - add the annotation words
    - word tree contains both visual and annotation words
  - use hybrid training loss:

$$-\log p(\mathbf{x}, y) = -\log p(y|\mathbf{x}) + \lambda \sum_{k=1}^{D} -\log p(x_k|\mathbf{x}_{<k})$$

  - used as features to an SVM
    - annotations are not used at test time

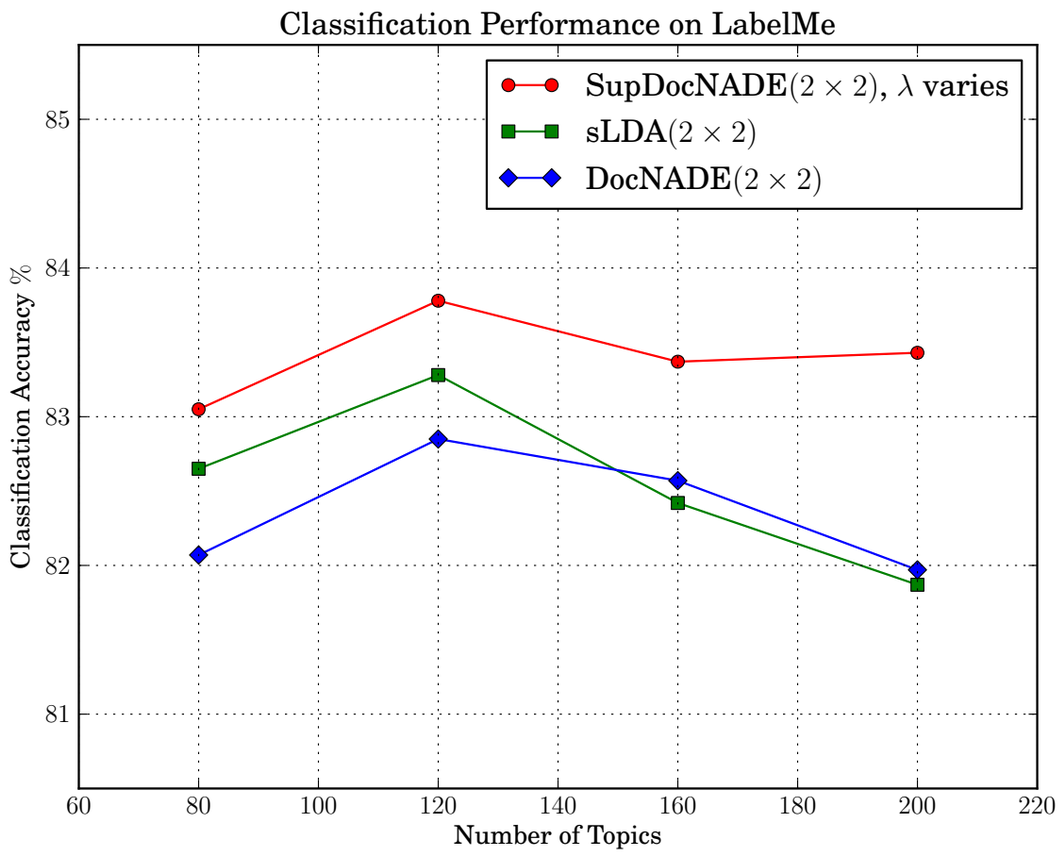


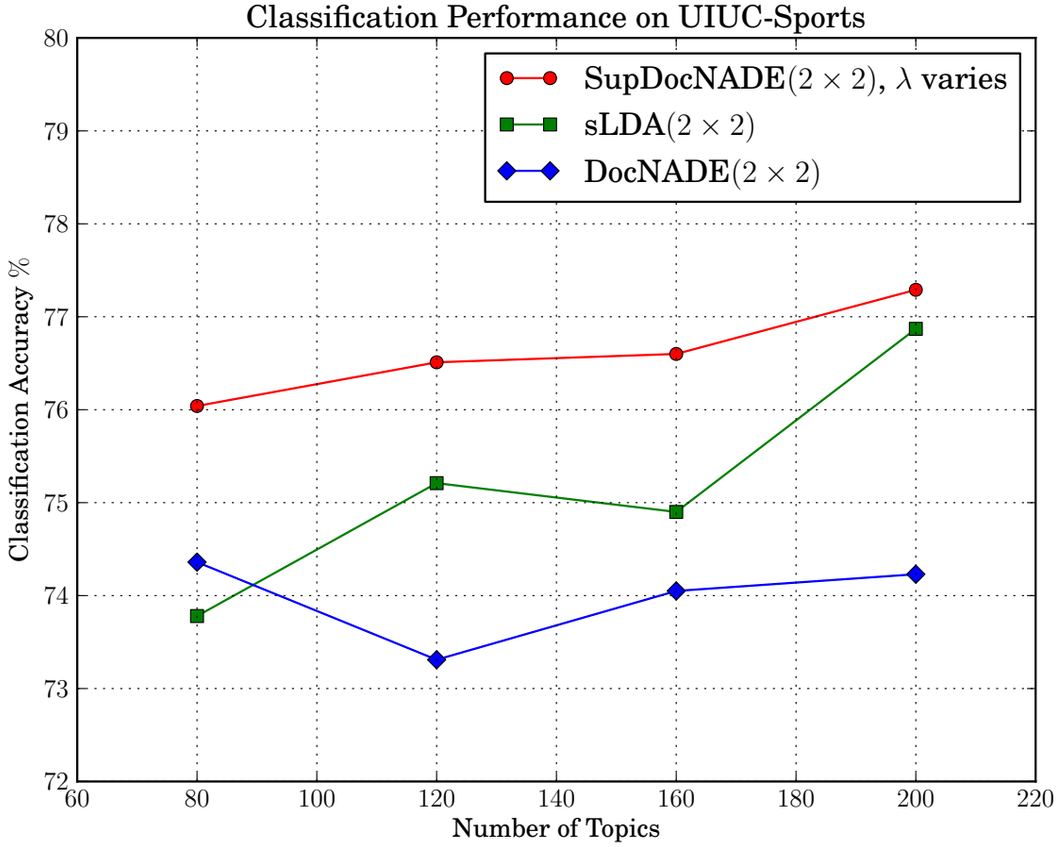
EXPERIMENTS

(Zheng, Zhang, Larochelle)

- **Scene classification:** comparison with
  - Supervised LDA
  - DocNADE

Classification Results



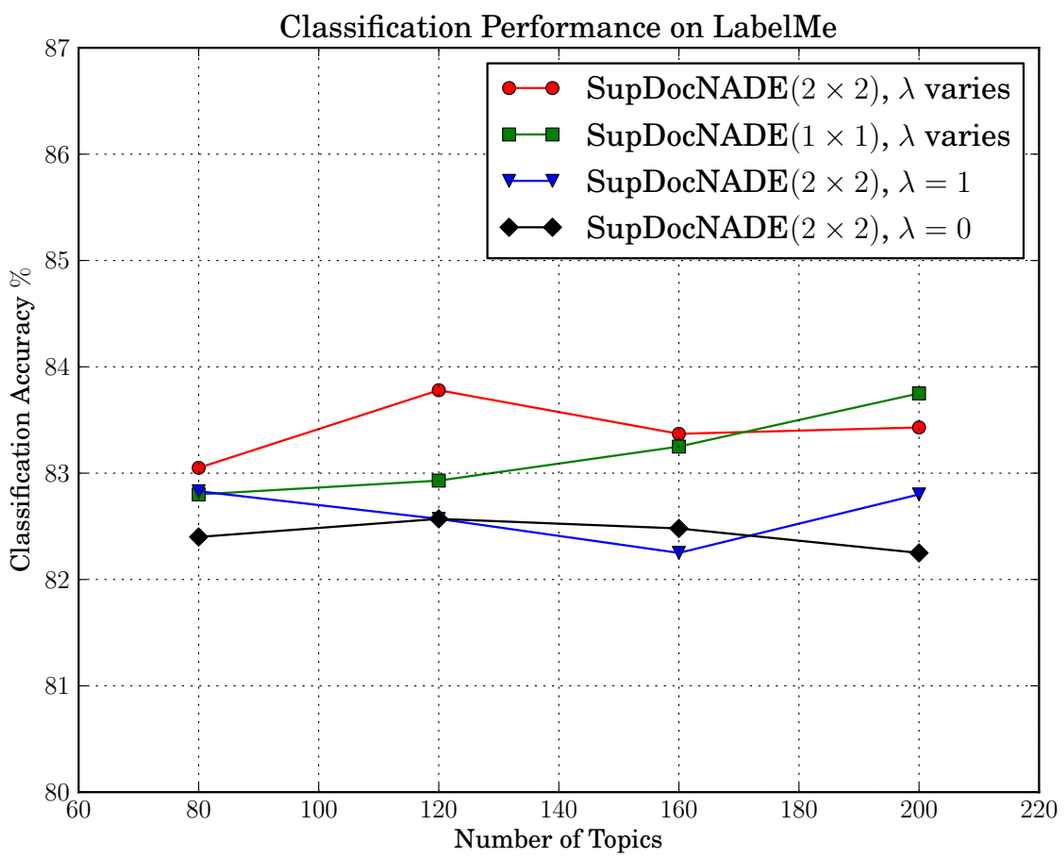


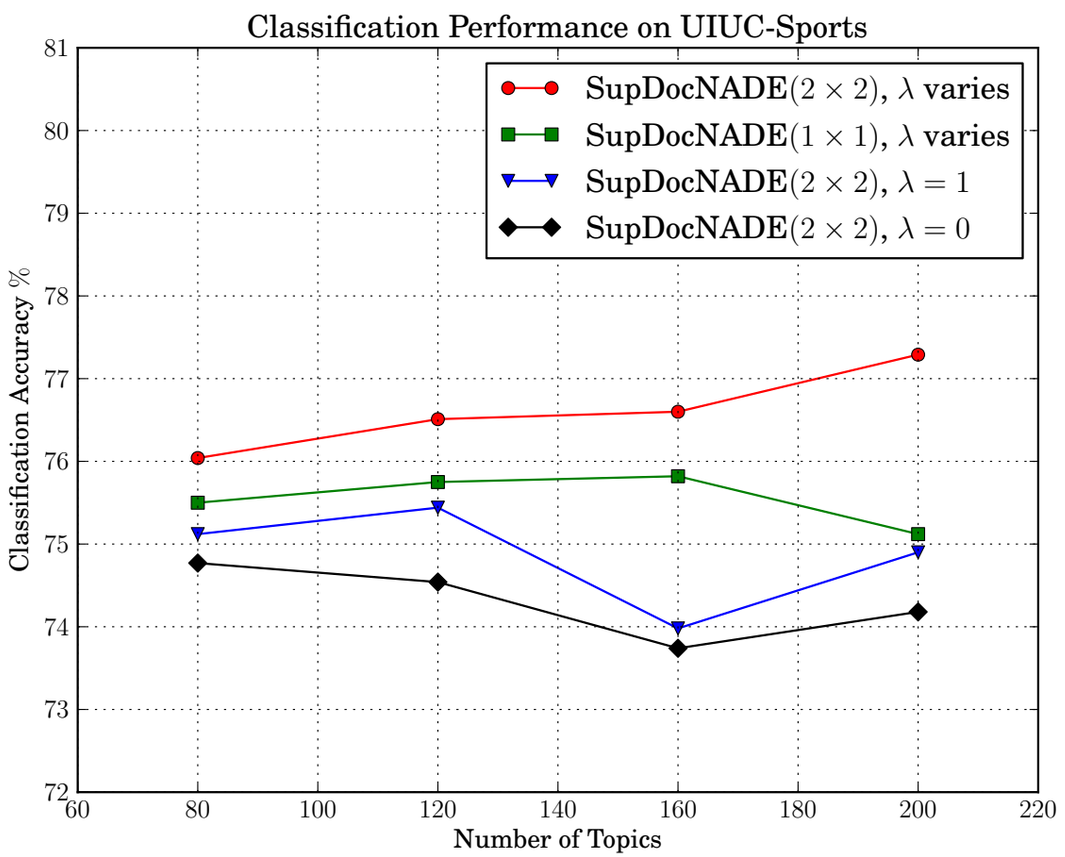
EXPERIMENTS

(Zheng, Zhang, Larochelle)

- **Scene classification:** comparison with
  - variants of Supervised DocNADE

Classification Results





EXPERIMENTS

(Zheng, Zhang, Larochelle)

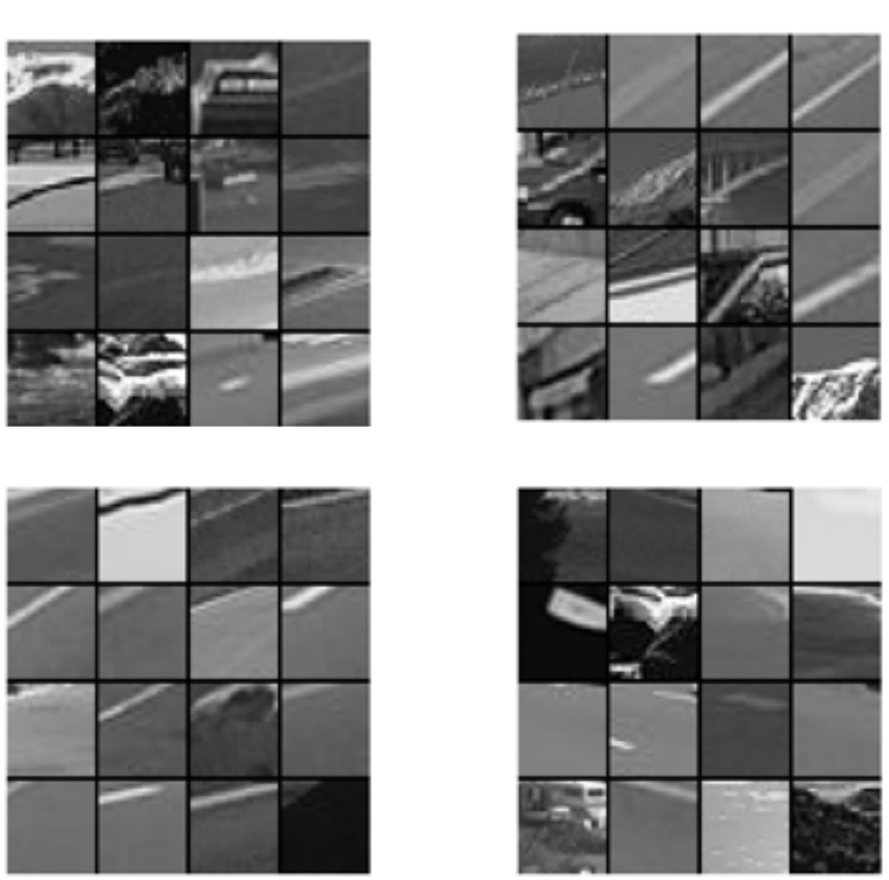
- **Scene annotation:** learned representation

Class: Highway

Annotation words

car
car occluded,
road,
fence,
trees

Visual words



EXPERIMENTS

(Zheng, Zhang, Larochelle)

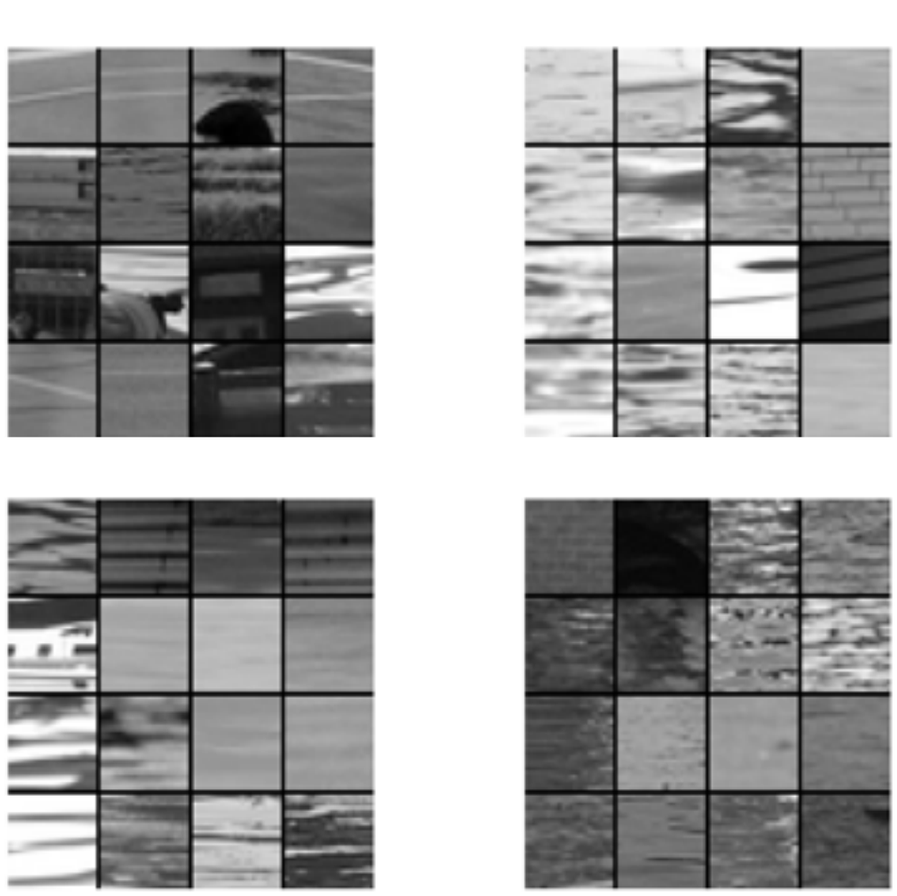
- **Scene annotation:** learned representation

Class: Sailing

Annotation words

athlete,
sky,
boat,
oar,
floater

Visual words



EXPERIMENTS

(Zheng, Zhang, Larochelle)

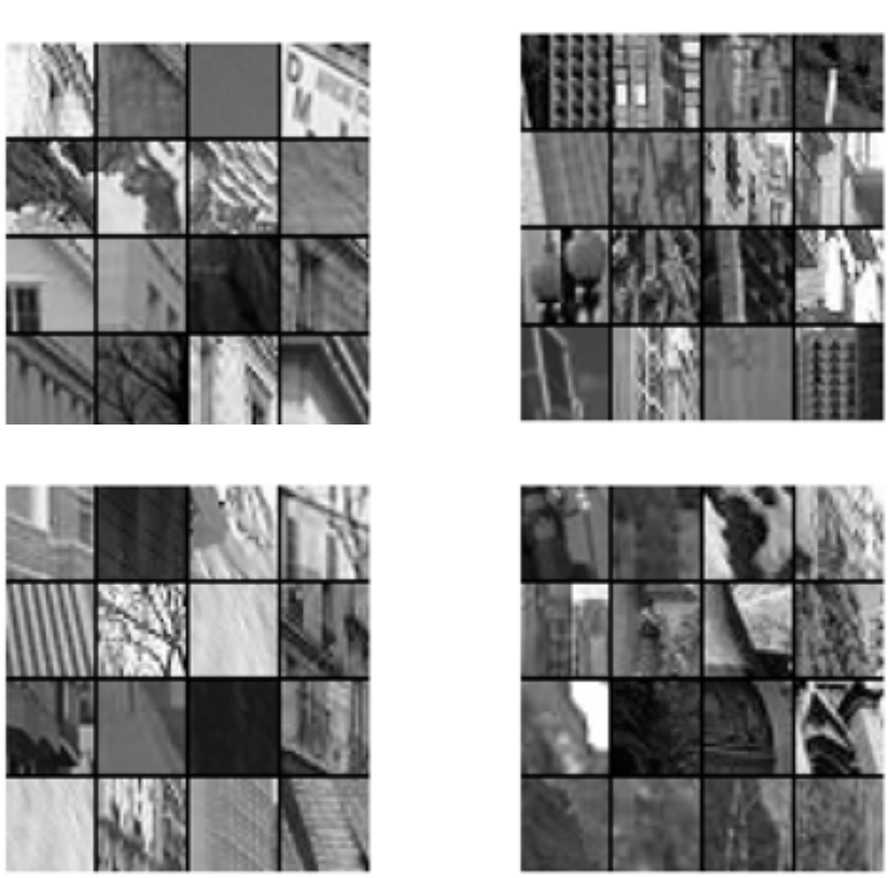
- **Scene annotation:** learned representation

Class: Street

Annotation words

building,
buildings,
window,
person walking,
sky

Visual words



OUTLINE

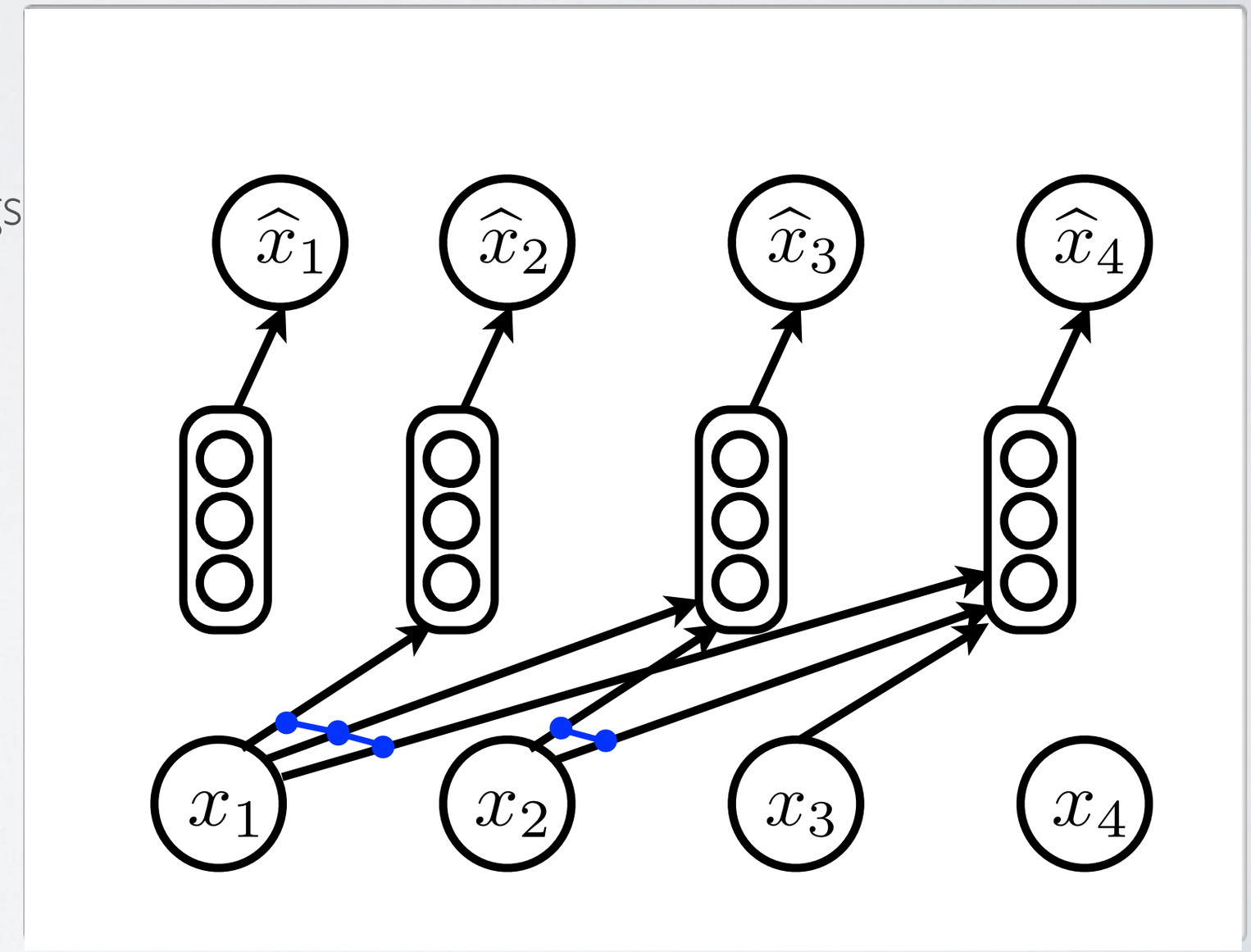
- Neural Autoregressive Distribution Estimator (NADE)
  - binary observations (with Iain Murray)
  - real-valued observations (with Benigno Uria and Iain Murray)
  - multinomial observations (with Stanislas Lauly and Yin Zheng)
  - **deep NADE (with Benigno Uria and Iain Murray)**

- Masked Autoencoder Distribution Estimator (MADE)
  - binary observations (with Mathieu Germain, Karol Gregor and Iain Murray)

DEEP NADE

(Uria, Murray, Larochelle)

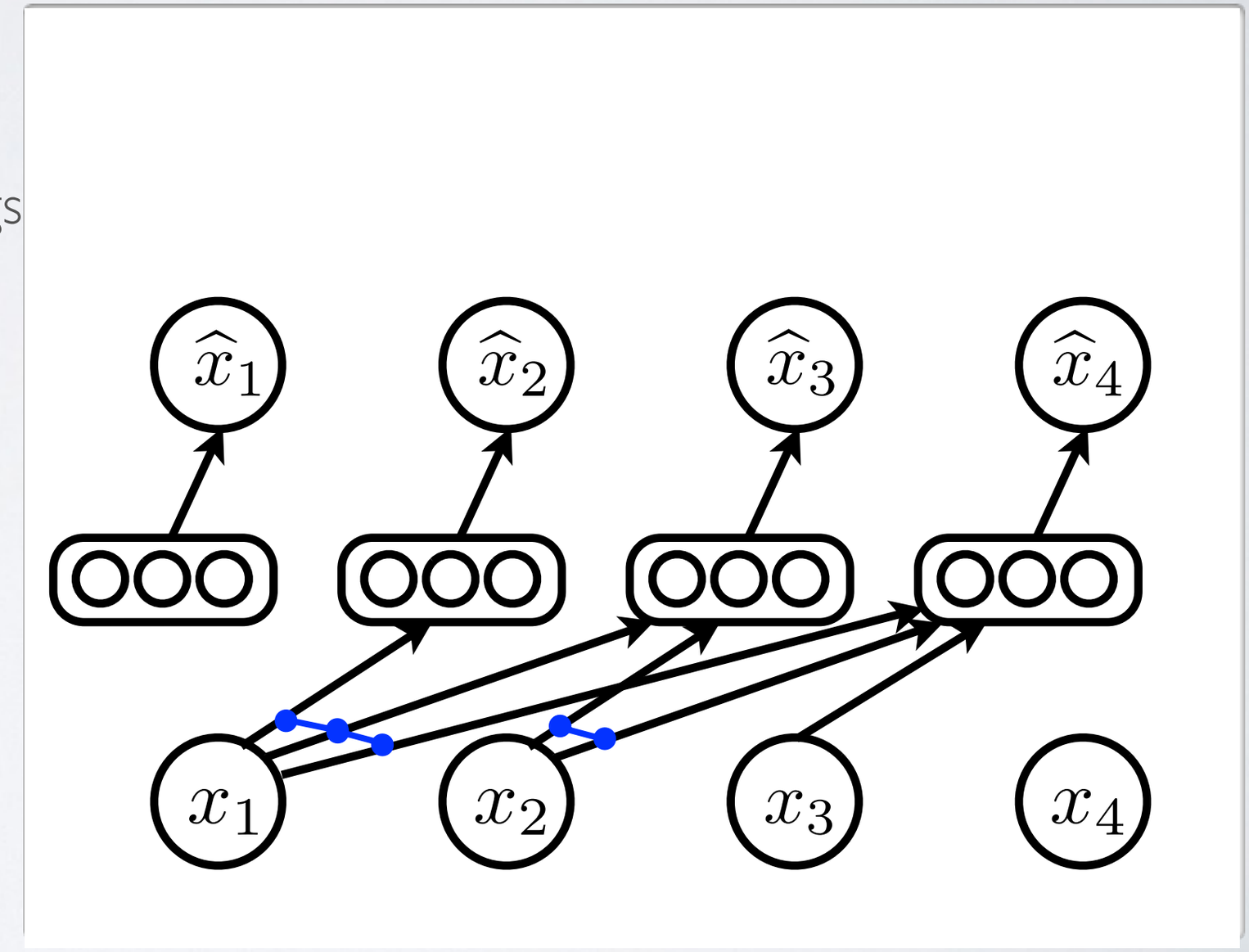
- **Deep NADE:** use a deep neural network
  - use more > 1 hidden layer
    - makes update in $O(DH^2)$
  - train on all possible orderings of the inputs
    - stochastic update by sampling the conditioning subset to get $O(DH + H^2)$
    - predict all future inputs to train all conditionals
  - condition on whether the input is observed using «conditioning weights»



DEEP NADE

(Uria, Murray, Larochelle)

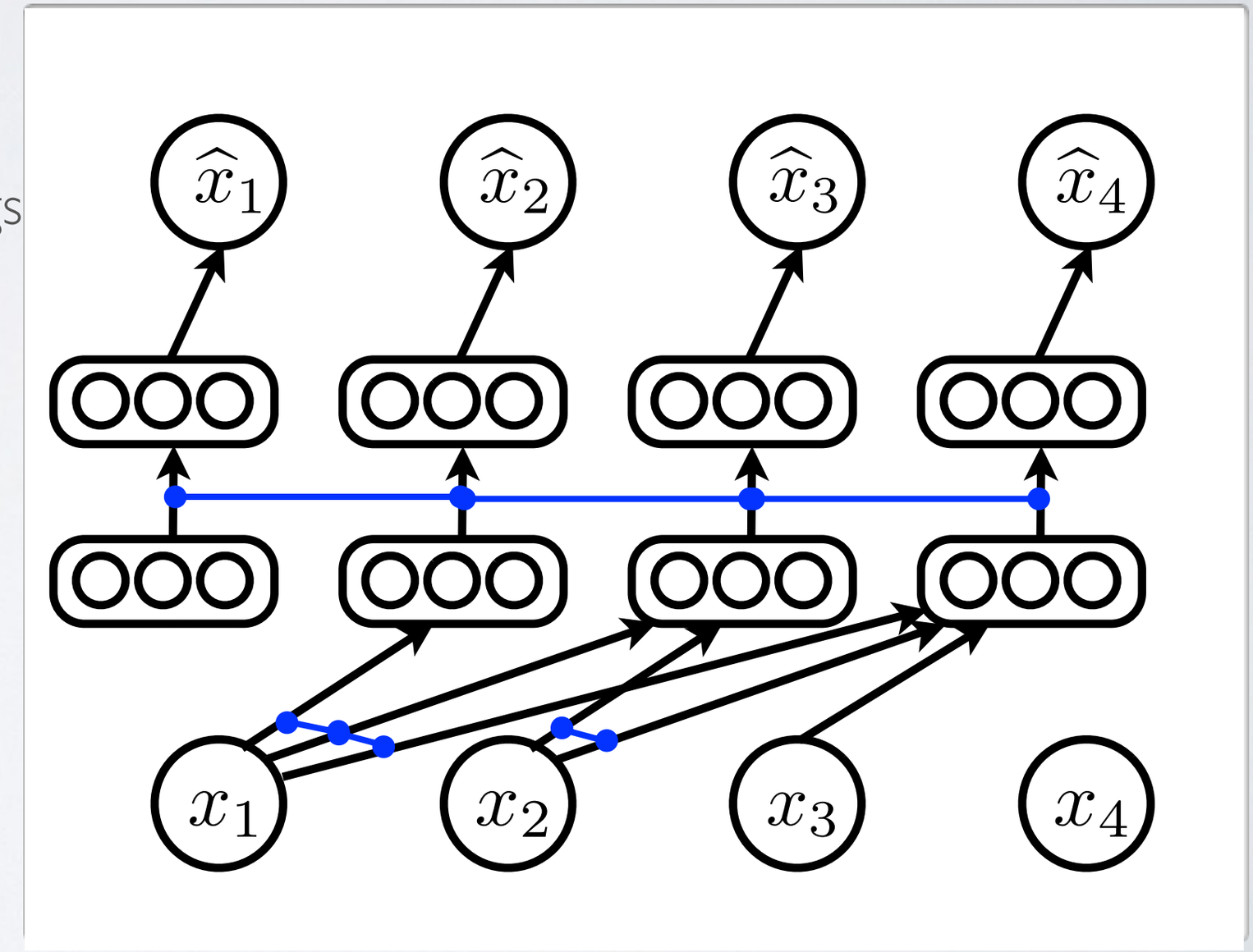
- **Deep NADE:** use a deep neural network
  - use more > 1 hidden layer
    - makes update in $O(DH^2)$
  - train on all possible orderings of the inputs
    - stochastic update by sampling the conditioning subset to get $O(DH + H^2)$
    - predict all future inputs to train all conditionals
  - condition on whether the input is observed using «conditioning weights»



DEEP NADE

(Uria, Murray, Larochelle)

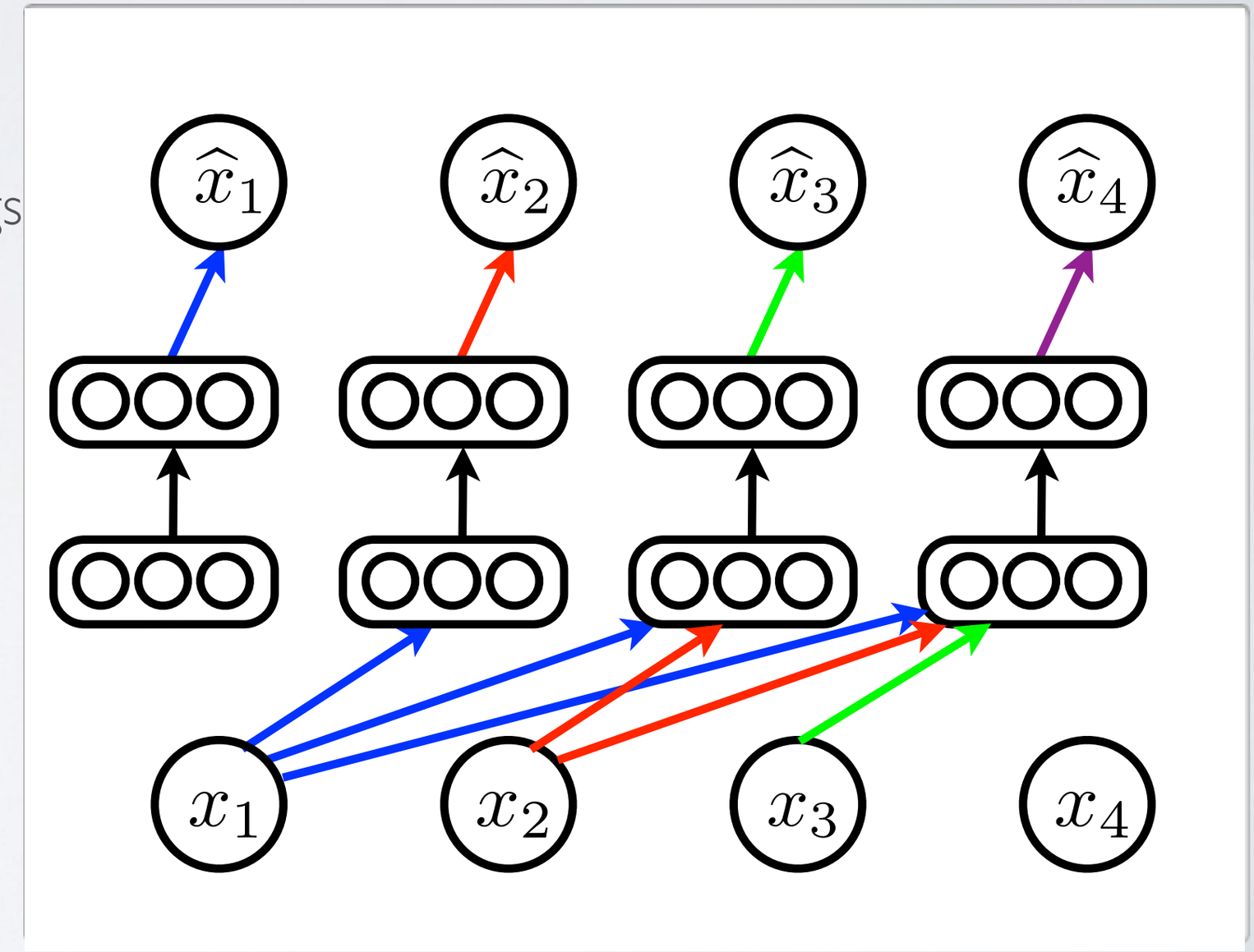
- **Deep NADE:** use a deep neural network
  - use more > 1 hidden layer
    - makes update in $O(DH^2)$
  - train on all possible orderings of the inputs
    - stochastic update by sampling the conditioning subset to get $O(DH + H^2)$
    - predict all future inputs to train all conditionals
  - condition on whether the input is observed using «conditioning weights»



DEEP NADE

(Uria, Murray, Larochelle)

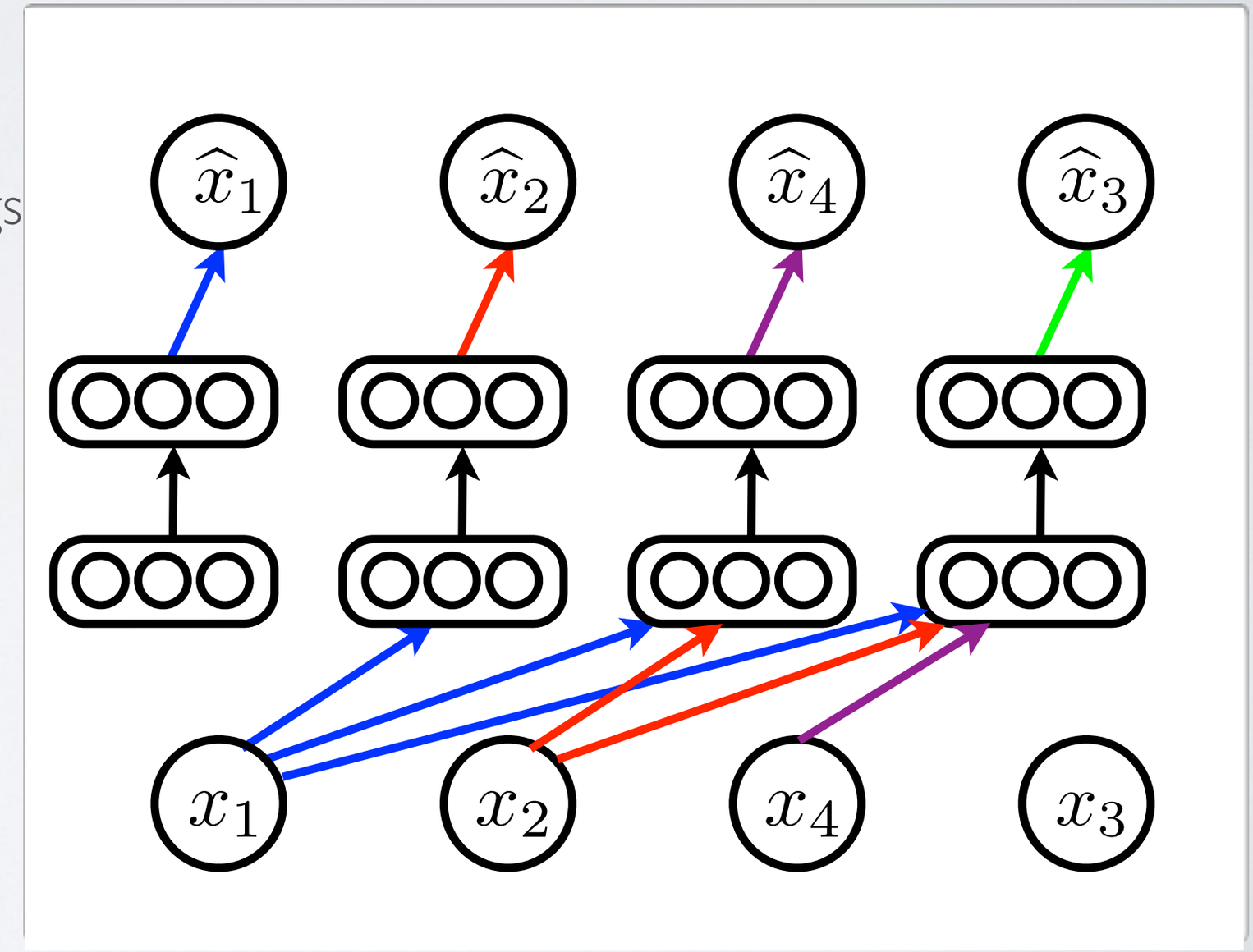
- **Deep NADE:** use a deep neural network
  - use more > 1 hidden layer
    - makes update in $O(DH^2)$
  - train on all possible orderings of the inputs
    - stochastic update by sampling the conditioning subset to get $O(DH + H^2)$
    - predict all future inputs to train all conditionals
  - condition on whether the input is observed using «conditioning weights»



DEEP NADE

(Uria, Murray, Larochelle)

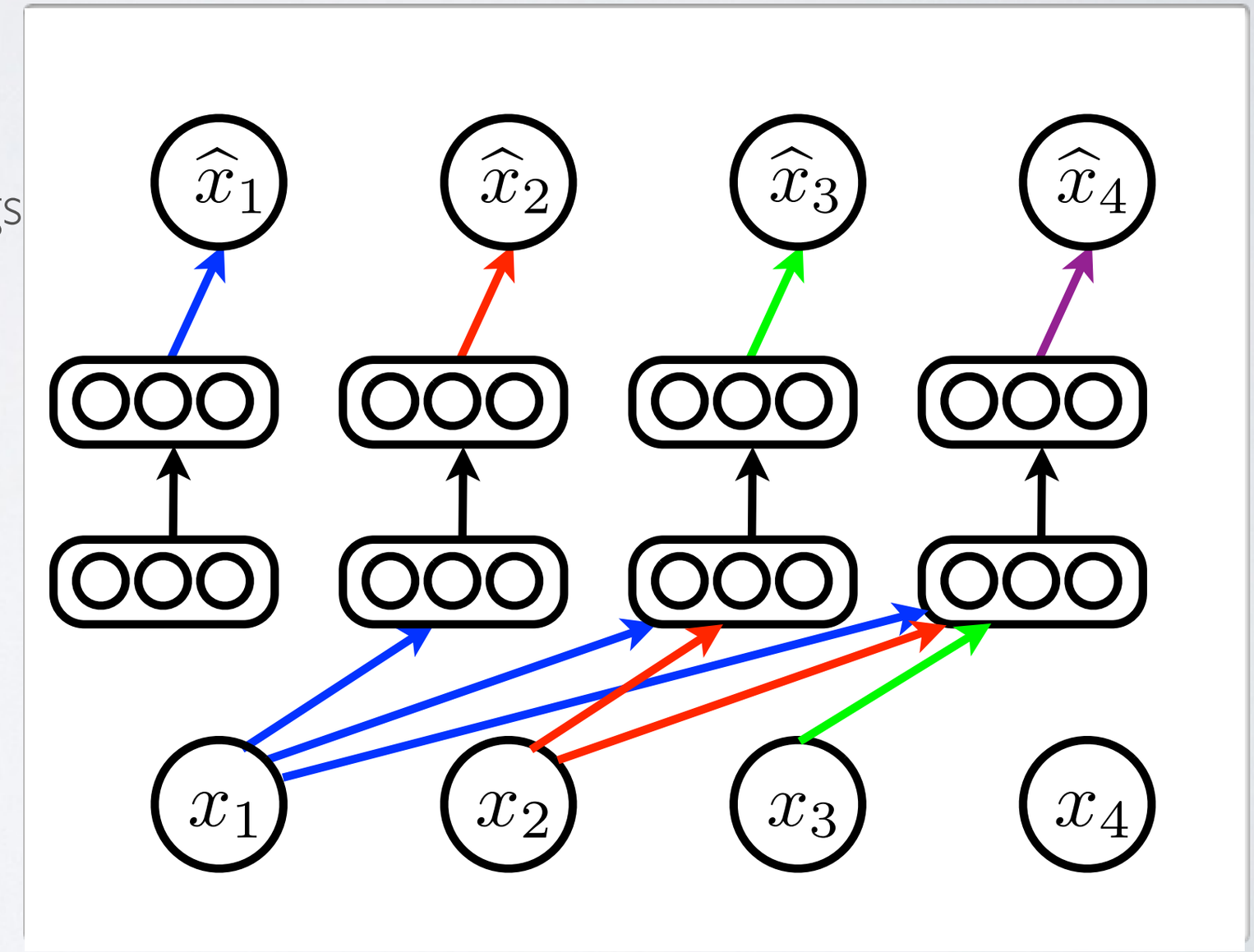
- **Deep NADE:** use a deep neural network
  - use more > 1 hidden layer
    - makes update in $O(DH^2)$
  - train on all possible orderings of the inputs
    - stochastic update by sampling the conditioning subset to get $O(DH + H^2)$
    - predict all future inputs to train all conditionals
  - condition on whether the input is observed using «conditioning weights»



DEEP NADE

(Uria, Murray, Larochelle)

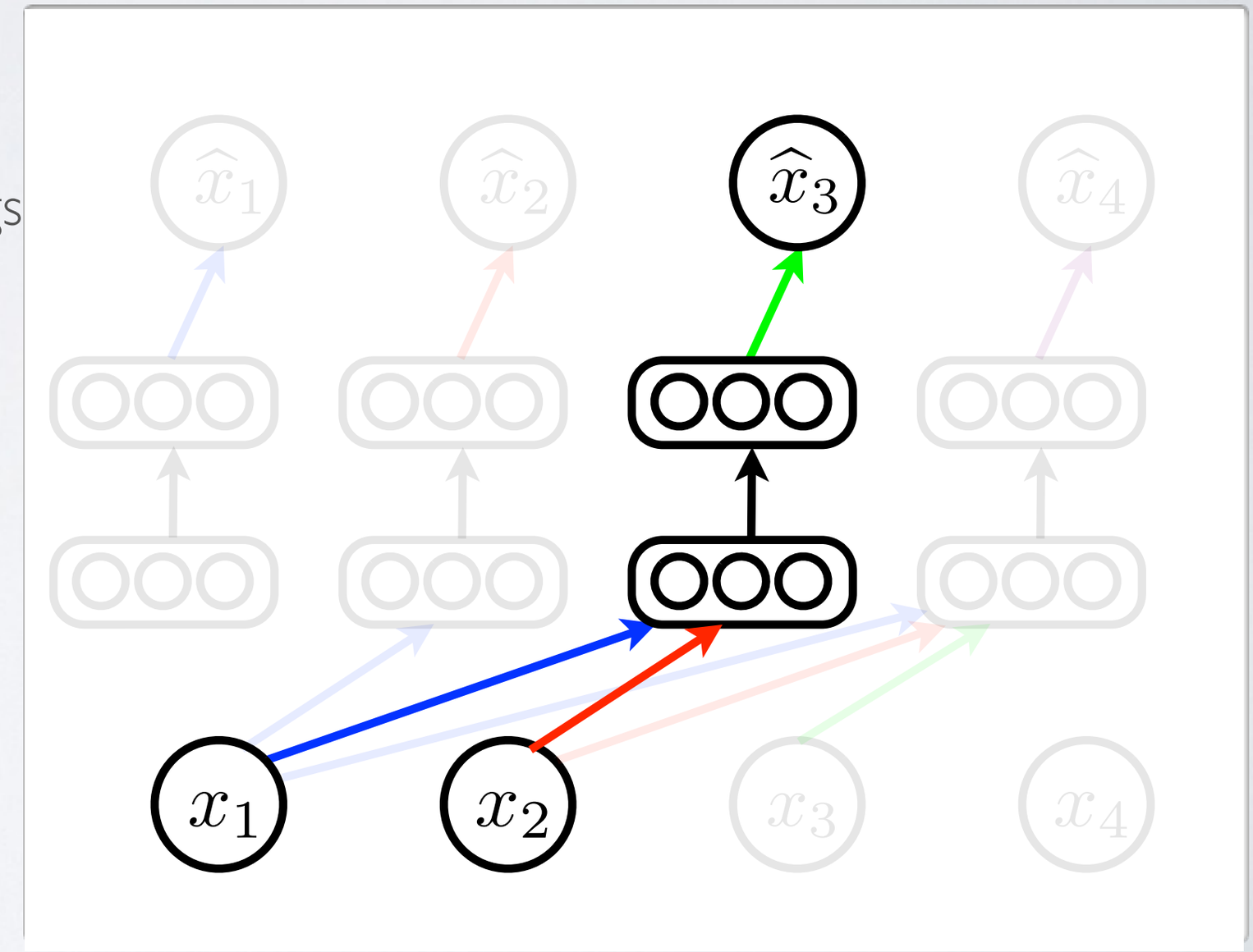
- **Deep NADE:** use a deep neural network
  - use more > 1 hidden layer
    - makes update in $O(DH^2)$
  - train on all possible orderings of the inputs
    - stochastic update by sampling the conditioning subset to get $O(DH + H^2)$
    - predict all future inputs to train all conditionals
  - condition on whether the input is observed using «conditioning weights»



DEEP NADE

(Uria, Murray, Larochelle)

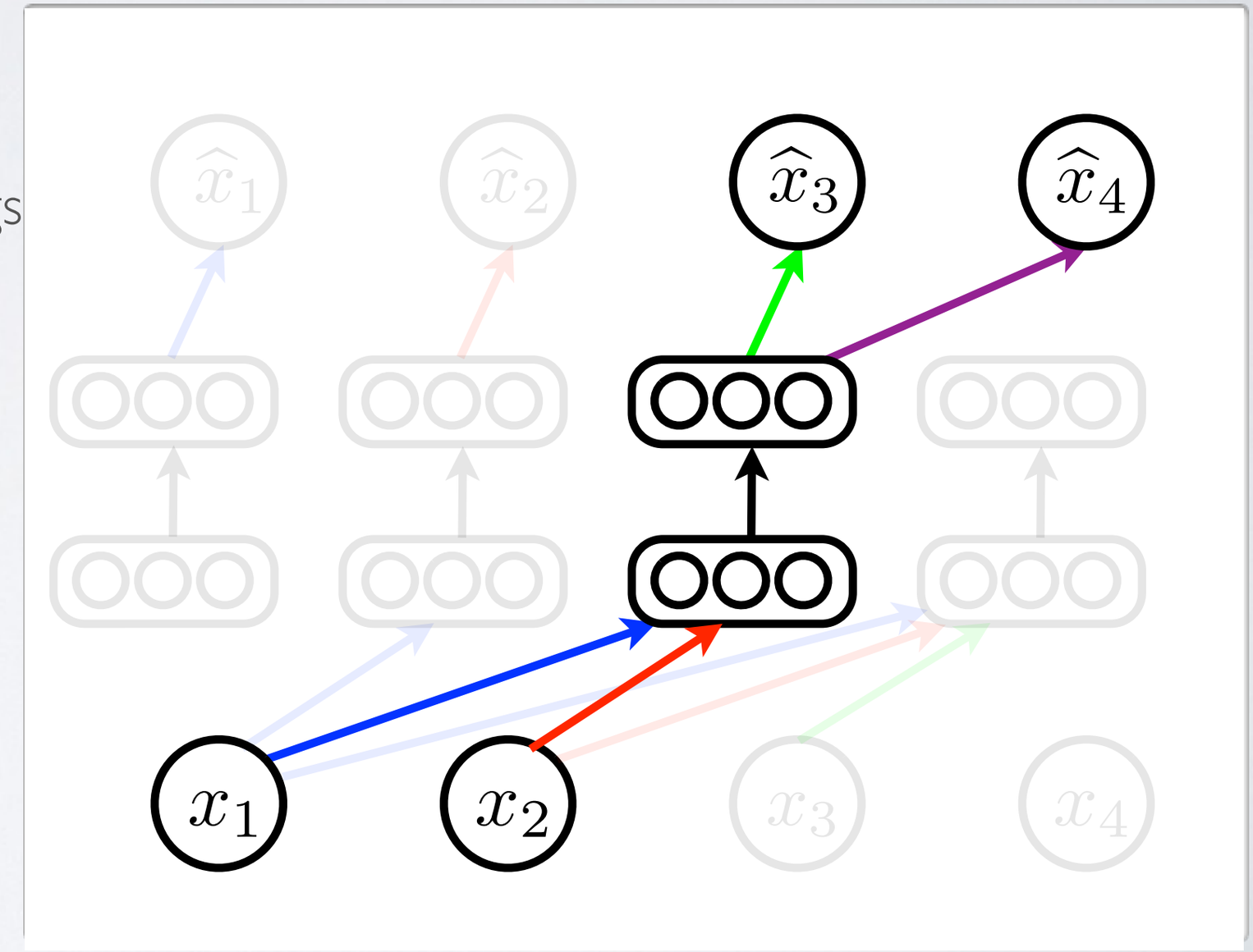
- **Deep NADE:** use a deep neural network
  - use more > 1 hidden layer
    - makes update in $O(DH^2)$
  - train on all possible orderings of the inputs
    - stochastic update by sampling the conditioning subset to get $O(DH + H^2)$
    - predict all future inputs to train all conditionals
  - condition on whether the input is observed using «conditioning weights»



DEEP NADE

(Uria, Murray, Larochelle)

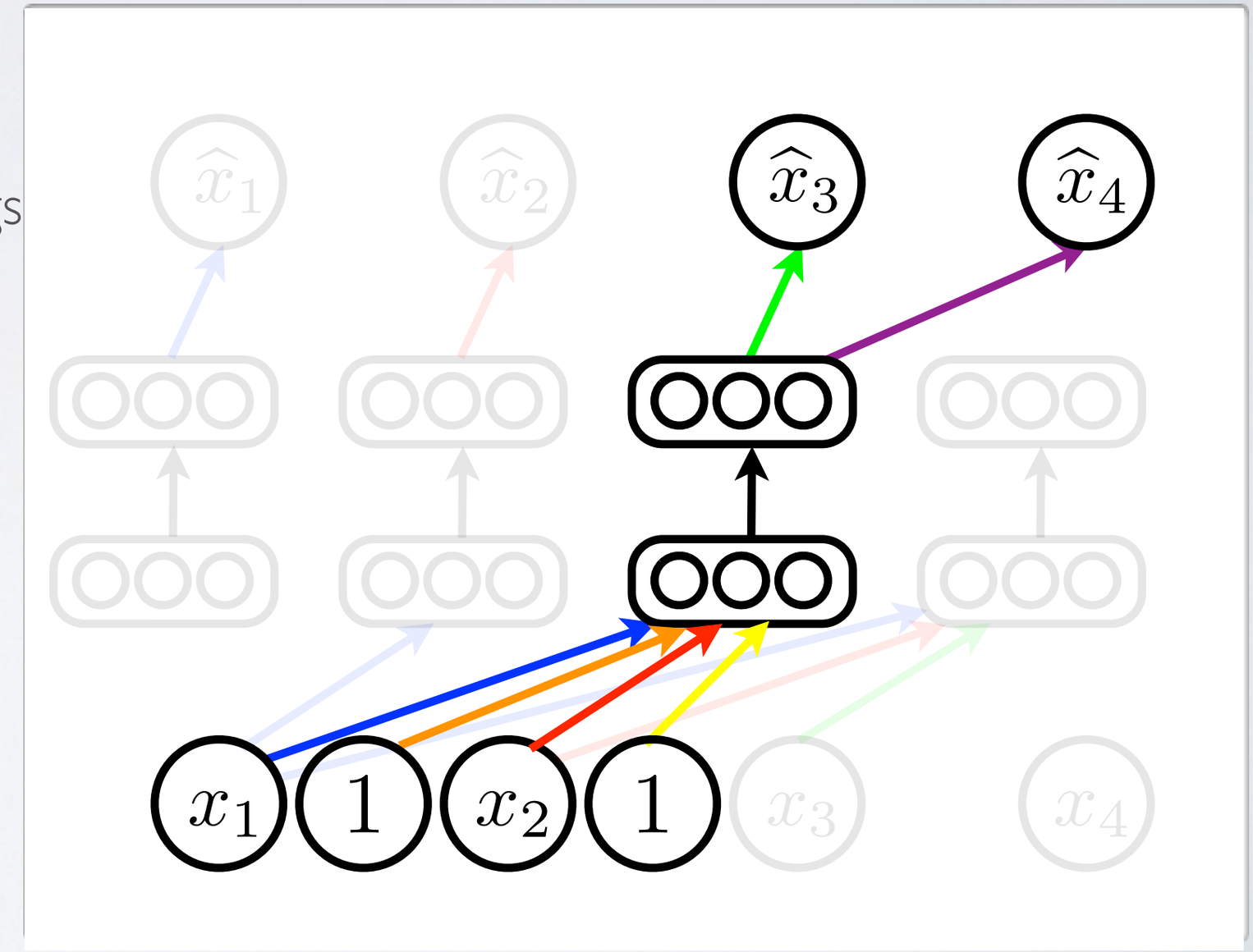
- **Deep NADE:** use a deep neural network
  - use more > 1 hidden layer
    - makes update in $O(DH^2)$
  - train on all possible orderings of the inputs
    - stochastic update by sampling the conditioning subset to get $O(DH + H^2)$
    - predict all future inputs to train all conditionals
  - condition on whether the input is observed using «conditioning weights»



DEEP NADE

(Uria, Murray, Larochelle)

- **Deep NADE:** use a deep neural network
  - use more > 1 hidden layer
    - makes update in $O(DH^2)$
  - train on all possible orderings of the inputs
    - stochastic update by sampling the conditioning subset to get $O(DH + H^2)$
    - predict all future inputs to train all conditionals
  - condition on whether the input is observed using «conditioning weights»



DEEP NADE

(Uria, Murray, Larochelle)

- **Deep NADE:** use a deep neural network
  - computing the full $p(\mathbf{x}|\boldsymbol{o})$ is in $O(DH^2)$
    - this reasonable and tractable
  - at test time, we can sample ensembles of deep NADE models on the fly
    - exploit the fact that $p(\mathbf{x}|\boldsymbol{o})$ varies across orderings $\boldsymbol{o}$
    - can reduce the variance of our estimate without retraining
    - can adapt to varying computational budget
  - can solve arbitrary inference problem $p(x_k|\mathbf{x}_{\boldsymbol{o}<d})$ by invoking the corresponding neural network

EXPERIMENTS

(Uria, Murray, Larochelle)

- **MNIST:** comparison with
  - MoB
  - RBM (approx.)
  - DBN (approx.)
  - NADE

Model	Test LogL
MoBernoullis K=10	−168.95
MoBernoullis K=500	−137.64
RBM (500 h, 25 CD steps) approx.	−86.34
DBN 2hl approx.	**−84.55**
NADE 1hl (fixed order)	−88.86
NADE 1hl (fixed order, RLU, minibatch)	−88.33
NADE 1hl (fixed order, sigm, minibatch)	−88.35
NADE 1hl (no input masks)	−99.37
NADE 2hl (no input masks)	−95.33
NADE 1hl	−92.17
NADE 2hl	−89.17
NADE 3hl	−89.38
NADE 4hl	−89.60
EoNADE 1hl (2 orderings)	−90.69
EoNADE 1hl (128 orderings)	−87.71
EoNADE 2hl (2 orderings)	−87.96
EoNADE 2hl (128 orderings)	−85.10

EXPERIMENTS

(Uria, Murray, Larochelle)

- **MNIST:** samples

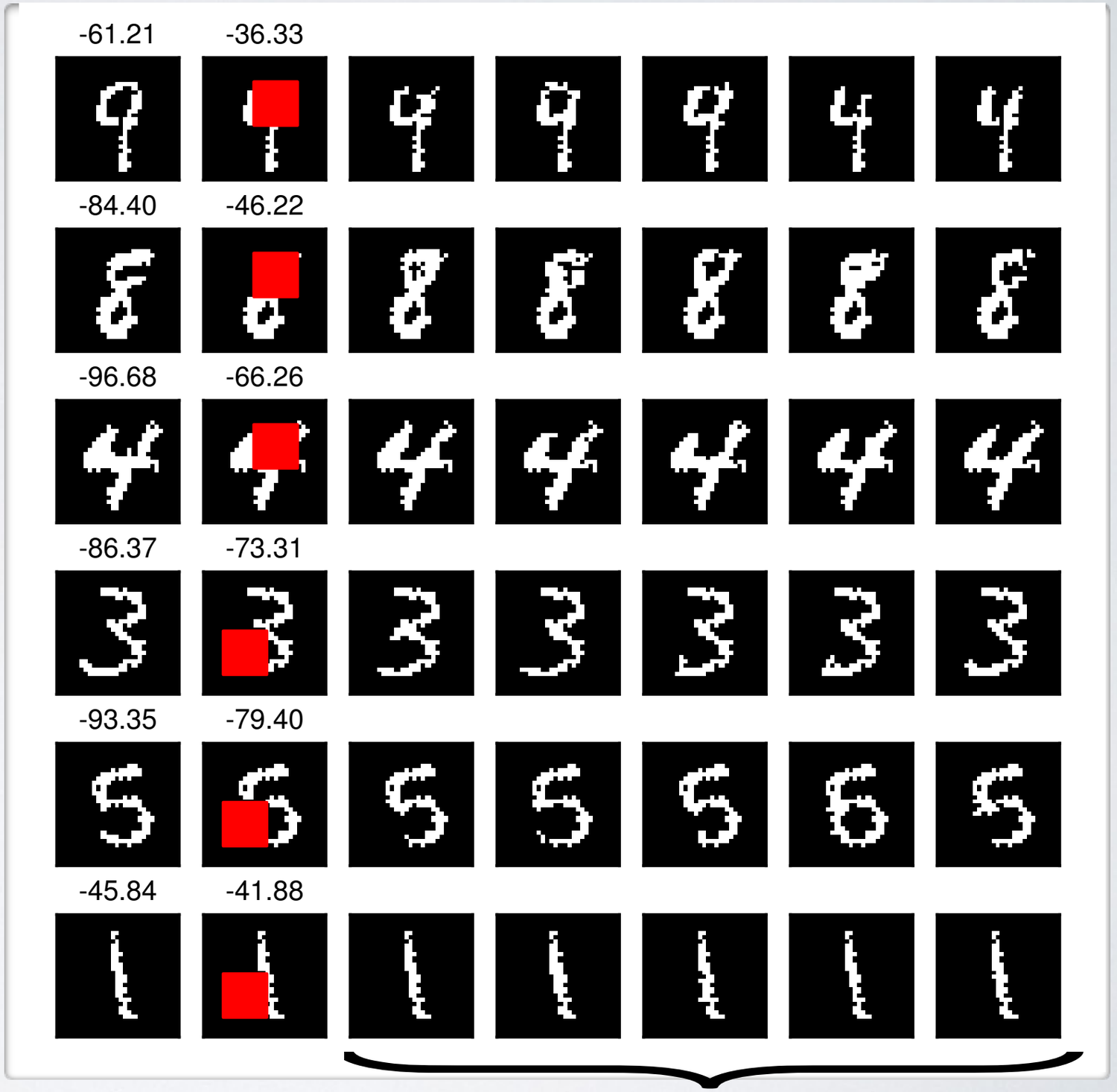
Examples from dataset

Samples from deep NADE

EXPERIMENTS

(Uria, Murray, Larochelle)

- **MNIST:** inference



completion samples

EXPERIMENTS

(Uria, Murray, Larochelle)

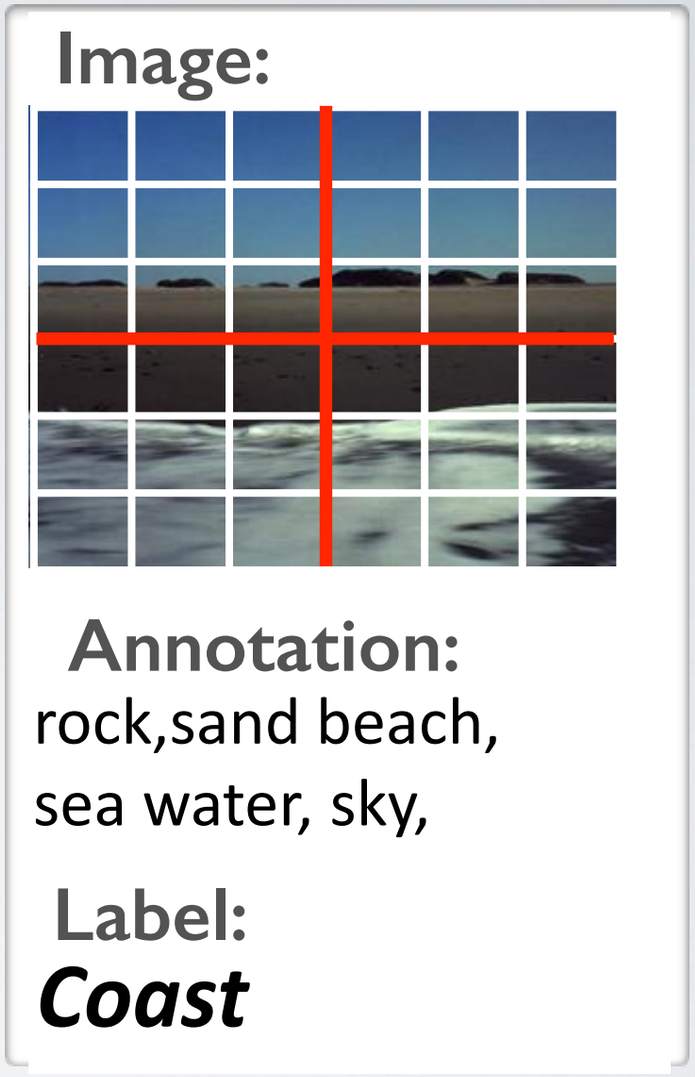
- **Natural image patches:** comparison with
  - mixture of Gaussians
  - RNADE

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**

MODELING IMAGES AND TEXT

(Zheng, Zhang, Larochelle)

- Back to modeling pairs of images and words
  - visual and annotation words are mixed
    - no need for the tree-structured multinomial
  - start by training a deep DocNADE model on unlabeled data
    - Flickr data set (≈1M image / annotation pairs)
    - train for several days
    - GPGPU implementation
  - finalize training with pure supervised learning (fine-tuning)



MODELING IMAGES AND TEXT

(Zheng, Zhang, Larochelle)

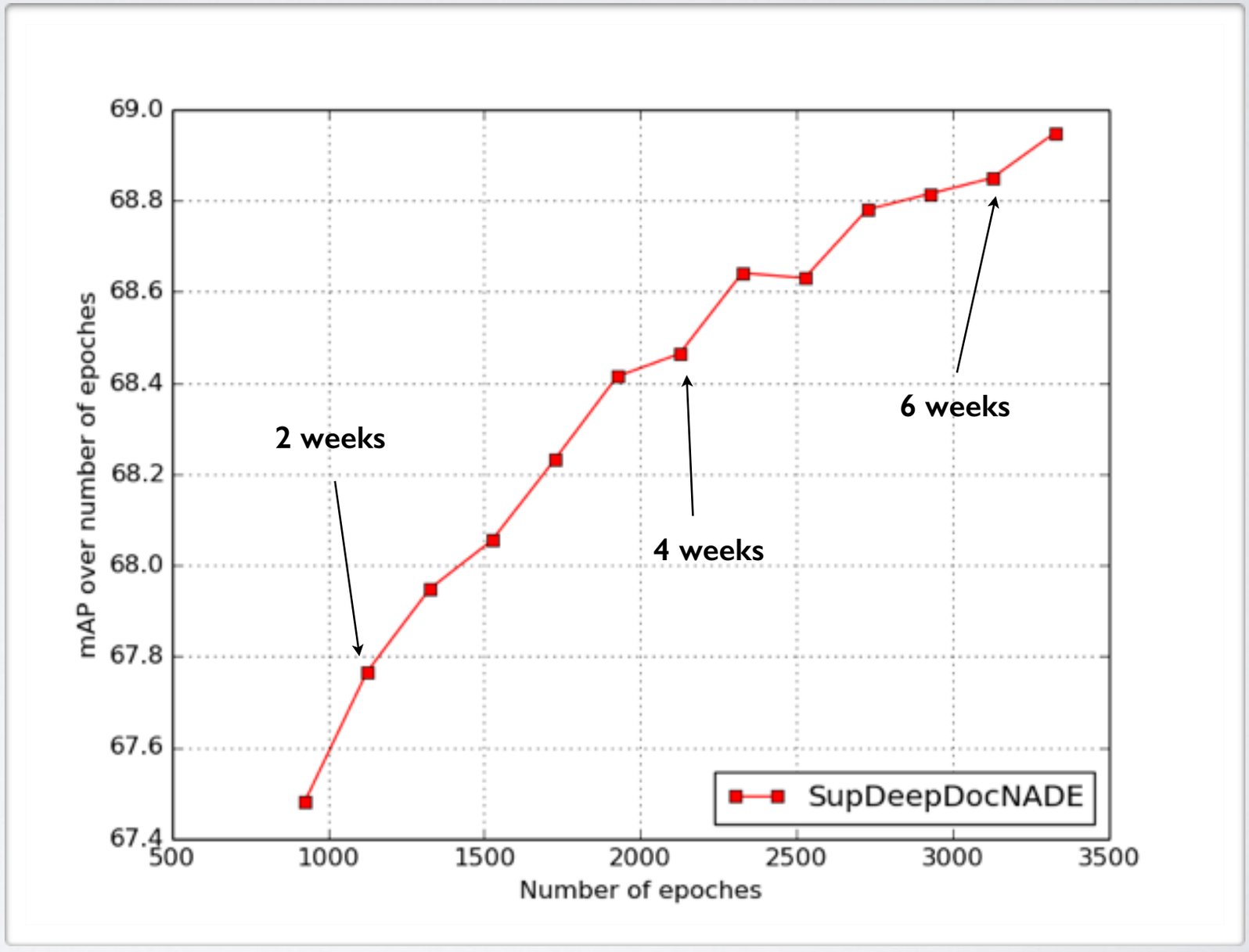
- Back to modeling pairs of images and words
  - Quantitative evaluation: **Classification**

Model	MAP
TF-IDF	0.384 ± 0.004
Multiple Kernel Learning SVMs [6]	0.623
TagProp [31]	0.640
Multimodal DBM [13]	0.651 ± 0.005
MDRNN [15]	0.686 ± 0.003
SupDeepDocNADE (1 hidden layer, 625 epochs pretraining)	0.654 ± 0.004
SupDeepDocNADE (2 hidden layers, 625 epochs pretraining)	0.671 ± 0.006
SupDeepDocNADE (3 hidden layers, 625 epochs pretraining)	0.670 ± 0.005
SupDeepDocNADE (2 hidden layers, 2325 epochs pretraining)	0.682 ± 0.005
SupDeepDocNADE (3 hidden layers, 2325 epochs pretraining)	0.686 ± 0.005
SupDeepDocNADE (2 hidden layers, 4125 epochs pretraining)	0.684 ± 0.005
SupDeepDocNADE (3 hidden layers, 4125 epochs pretraining)	$\mathbf{0.691} \pm 0.005$

MODELING IMAGES AND TEXT

(Zheng, Zhang, Larochelle)

- Back to modeling pairs of images and words
  - Quantitative evaluation: **Classification**



MODELING IMAGES AND TEXT

(Zheng, Zhang, Larochelle)

- Back to modeling pairs of images and words
  - Quantitative evaluation: **Annotating images**

Image Input	Ground Truth Annotations	Generated Texts
	cheese	food,vegan,recipe, cheese,breakfast,cake, dessert,dinner
	explore,italy,europe,italia, roma,rome,public,i500,eu	sign,red,white,circle,letter, heart,sticker,typography
	2007,boat,michigan,raw, ship,d70,harbor,nikond70	ship,boat,sea,chicago,sanfra ncisco,beach,water,harbour
	france	wall,graffiti,streetart,bricks, tiles,pattern,london,mosaic

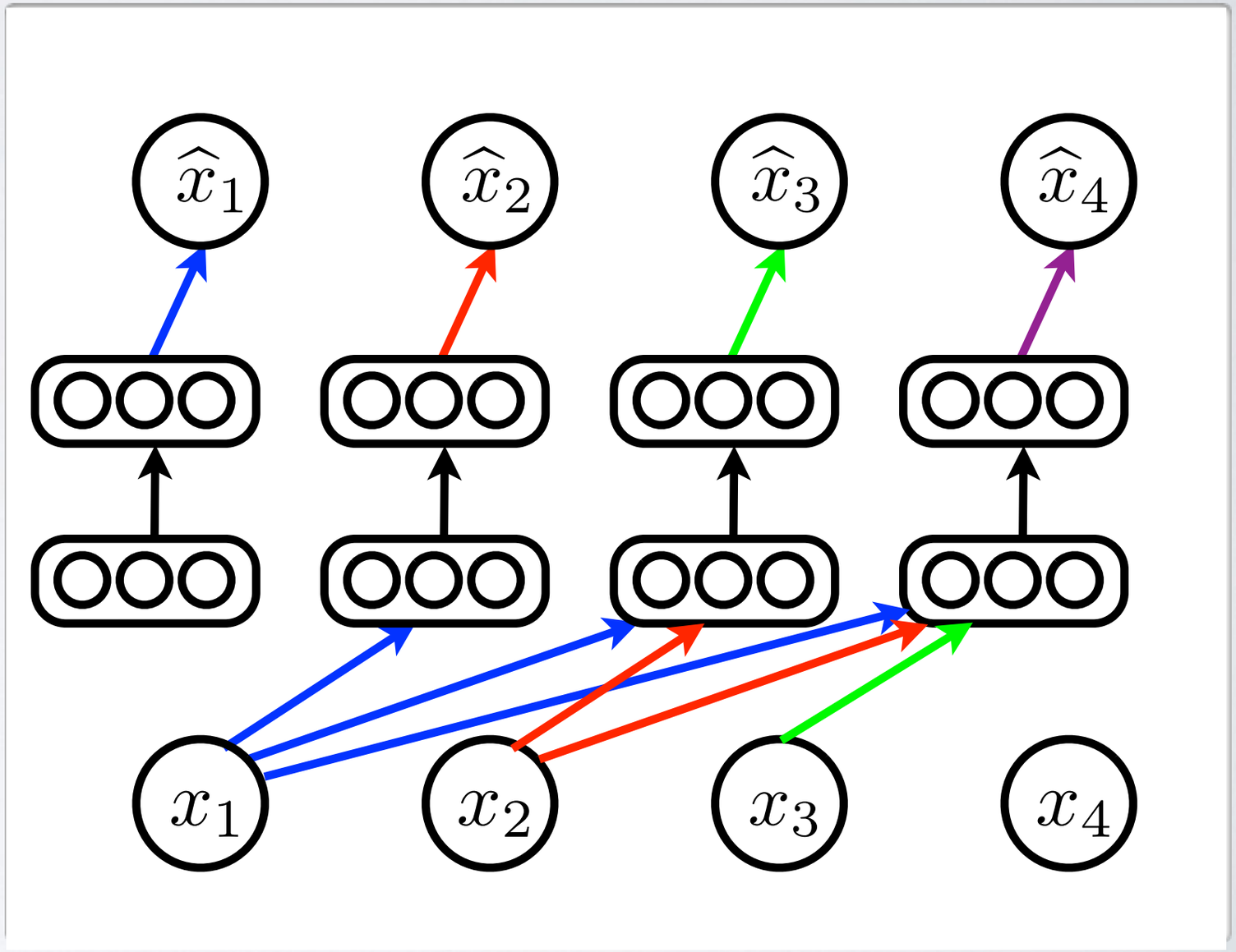
OUTLINE

- Neural Autoregressive Distribution Estimator (NADE)
  - binary observations (with Iain Murray)
  - real-valued observations (with Benigno Uria and Iain Murray)
  - multinomial observations (with Stanislas Lauly and Yin Zheng)
  - deep NADE (with Benigno Uria and Iain Murray)

- **Masked Autoencoder Distribution Estimator (MADE)**
  - **binary observations (with Mathieu Germain, Karol Gregor and Iain Murray)**

AUTOREGRESSIVE PROPERTY

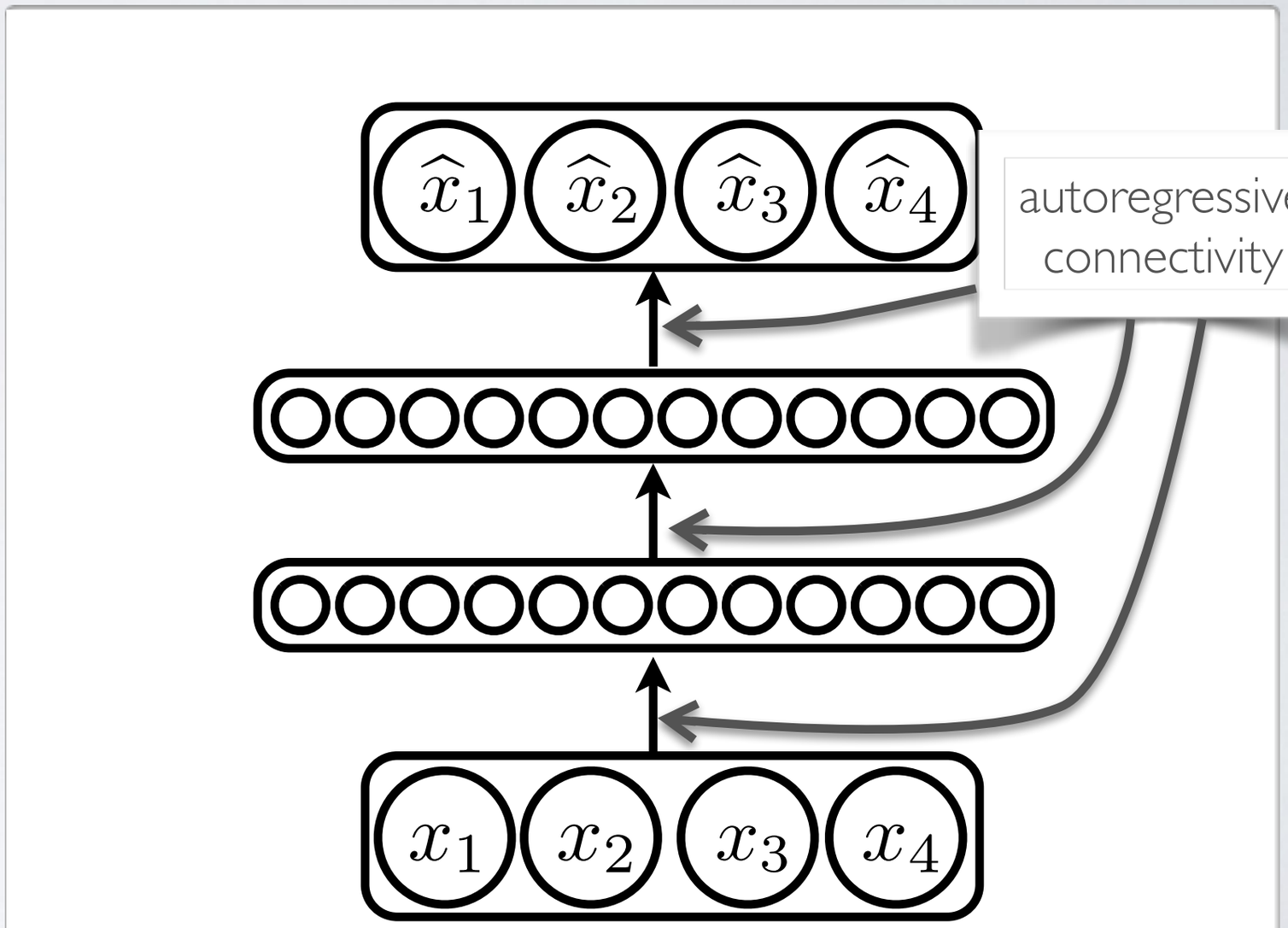
- NADE is a distribution estimator because it respects the autoregressive property
  - each output depends only en previous inputs (in some ordering)
- Autoencoder doesn't respect that property
  - is there a simple way to adapt the autoencoder so that it does



AUTOREGRESSIVE PROPERTY

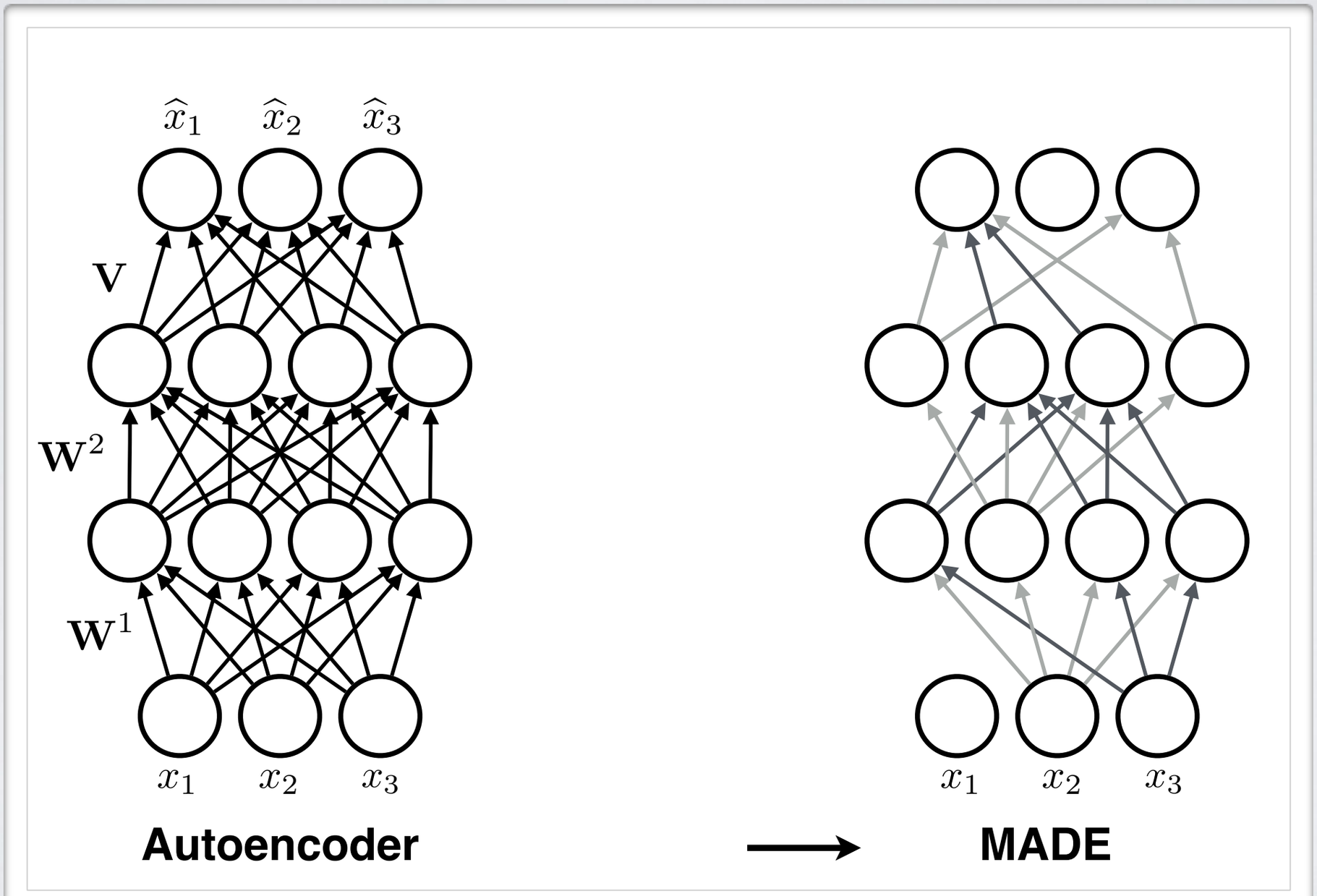
- NADE is a distribution estimator because it respects the autoregressive property
  - each output depends only en previous inputs (in some ordering)

- Autoencoder doesn't respect that property
  - is there a simple way to adapt the autoencoder so that it does



MADE IN PICTURE

- Masked Autoencoder Distribution Estimator (MADE)



MADE IN WORDS

- Masked Autoencoder Distribution Estimator (MADE)
  - masks can be generated once and for all or for every update
  - training is similar to dropout, except the dropping out is on the connections, not the units (just like DropConnect)
  - at test time, can create an ensemble if used multiple masks
  - can easily generalize to other connections between layers (e.g. direct input to output)

MADE IN WORDS

- Related work
  - FVSBN is MADE without hidden units
  - Bengio and Bengio (2000) proposed the first version of MADE (single hidden layer, no mask sampling)
  - For the same number of hidden units, MADE is **much** faster to train than NADE
    - matrix multiplies are faster + fewer non-linearities to compute
  - At test time, MADE is much faster to evaluate than deep NADE
    - can compute the log-likelihood in one pass in MADE, instead of D

RESULTS

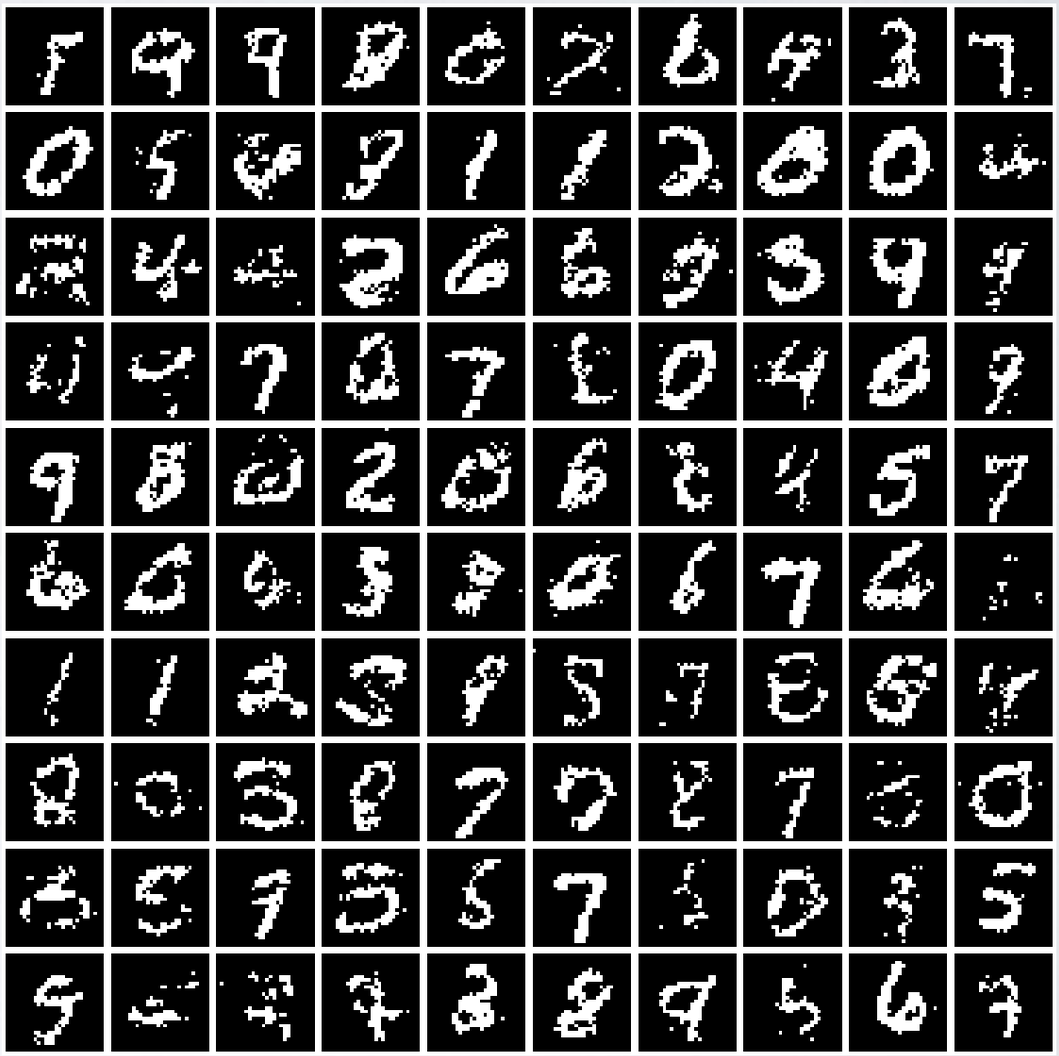
- **UCI benchmark**

Model	Adult	Connect4	DNA	Mushrooms	NIPS-0-12	OCR-letters	RCV1	Web
MoBernoullis	20.44	23.41	98.19	14.46	290.02	40.56	47.59	30.16
RBM	16.26	22.66	96.74	15.15	277.37	43.05	48.88	29.38
FVSBN	**13.17**	12.39	83.64	10.27	276.88	39.30	49.84	29.35
NADE (fixed order)	**13.19**	11.99	84.81	9.81	**273.08**	**27.22**	46.66	28.39
EoNADE 1hl (16 ord.)	**13.19**	12.58	82.31	9.69	**272.39**	**27.32**	**46.12**	**27.87**
DARN	13.19	11.91	81.04	**9.55**	274.68	≈28.17	≈**46.10**	≈28.83
MADE	**13.12**	**11.90**	83.63	9.68	280.25	28.34	47.10	28.53
MADE mask sampling	**13.13**	**11.90**	**79.66**	9.69	277.28	30.04	46.74	**28.25**

RESULTS

- **Binarize MNIST**

Model	$-\log p$	
RBM (500 h, 25 CD steps)	≈ 86.34	Intractable
DBM 2hl	≈ 84.62	
DBN 2hl	≈ 84.55	
DARN n_h=500	≈ 84.71	
DARN n_h=500, adaNoise	≈ 84.13	
MoBernoullis K=10	168.95	Tractable
MoBernoullis K=500	137.64	
NADE 1hl (fixed order)	88.33	
EoNADE 1hl (128 orderings)	87.71	
EoNADE 2hl (128 orderings)	85.10	
MADE 1hl (1 mask)	88.40	
MADE 2hl (1 mask)	89.59	
MADE 1hl (32 masks)	88.04	
MADE 2hl (32 masks)	86.64	



CONCLUSION

- Distribution estimation / generative modeling is a very active topic right now
  - **VAE**: Variational Autoencoder (Kingma et al., 2014)
  - **DARN**: Deep Autoregressive Networks (Gregor et al. 2014)
  - **DRAW**: Deep Recurrent Attentive Writer (Gregor et al. 2015)
  - **GAN**: Generative Adversarial Networks (Goodfellow et al. 2014)
  - **GSN**: Generative Stochastic Networks (Bengio et al. 2014)
  - **DBM**: deep Boltzmann machines (Salakhutdinov and Hinton, 2009)

- Lots of research still needed to yield models with practical applications

Merci !