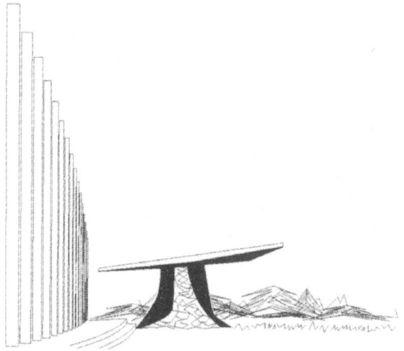
IBM T. J. Watson Research Center

Model-based Speech Separation and Recognition



Steven J. Rennie

Chime 2013 Workshop
June 1, 2013

Motivation

- **Noise-robust Automatic Speech Recognition (ASR)**
- **Noise-robust *Multi-talker* ASR**
- **Signal Separation/Isolation/Analysis/Decomposition**

Some Applications

surveillance

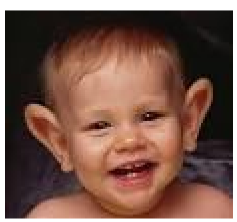
acoustic forensics

mobile computing

signal re-composition/editing

artificial perception

enhanced hearing



robust audio search

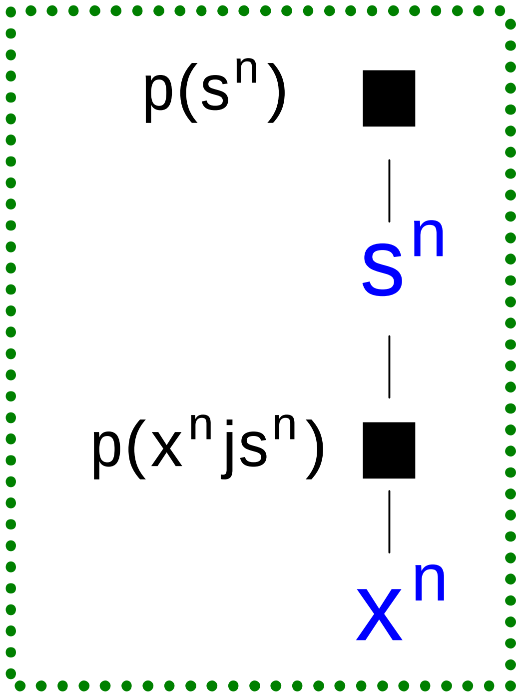
Why is Robust ASR hard?

PLACE WHITE AT D ZERO SOON

- **Multiple sources of interference, including speech**
  - Computational explosion in the number of possible "acoustic states" of the environment
  - This makes data acquisition difficult
  - This makes statistical data analysis difficult

Audio demos: http://researcher.watson.ibm.com/researcher/view_project.php?id=2819

Combinatorial Considerations



Source Models:

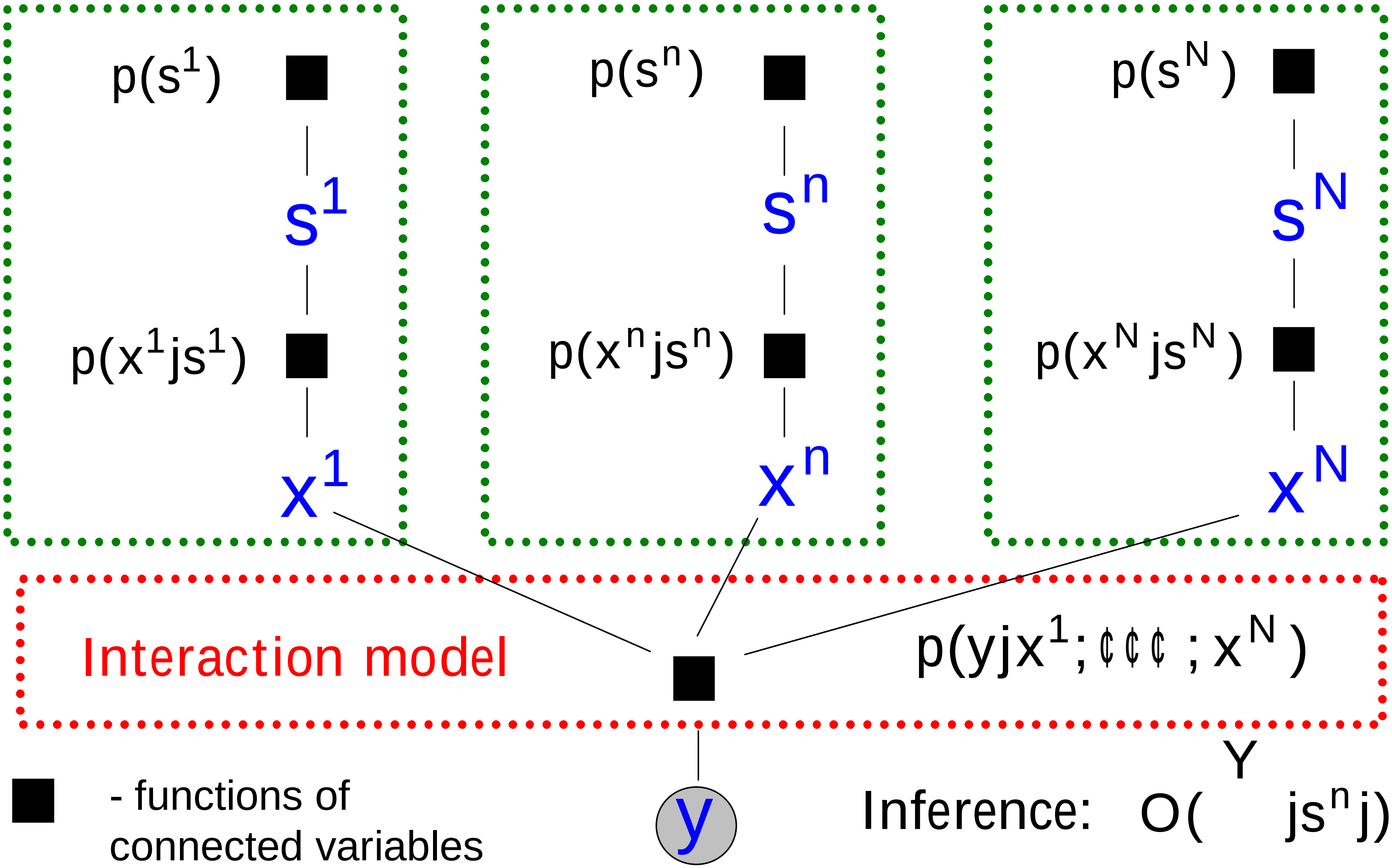
-features x^n

-states s^n

-number of states $j s^n j$

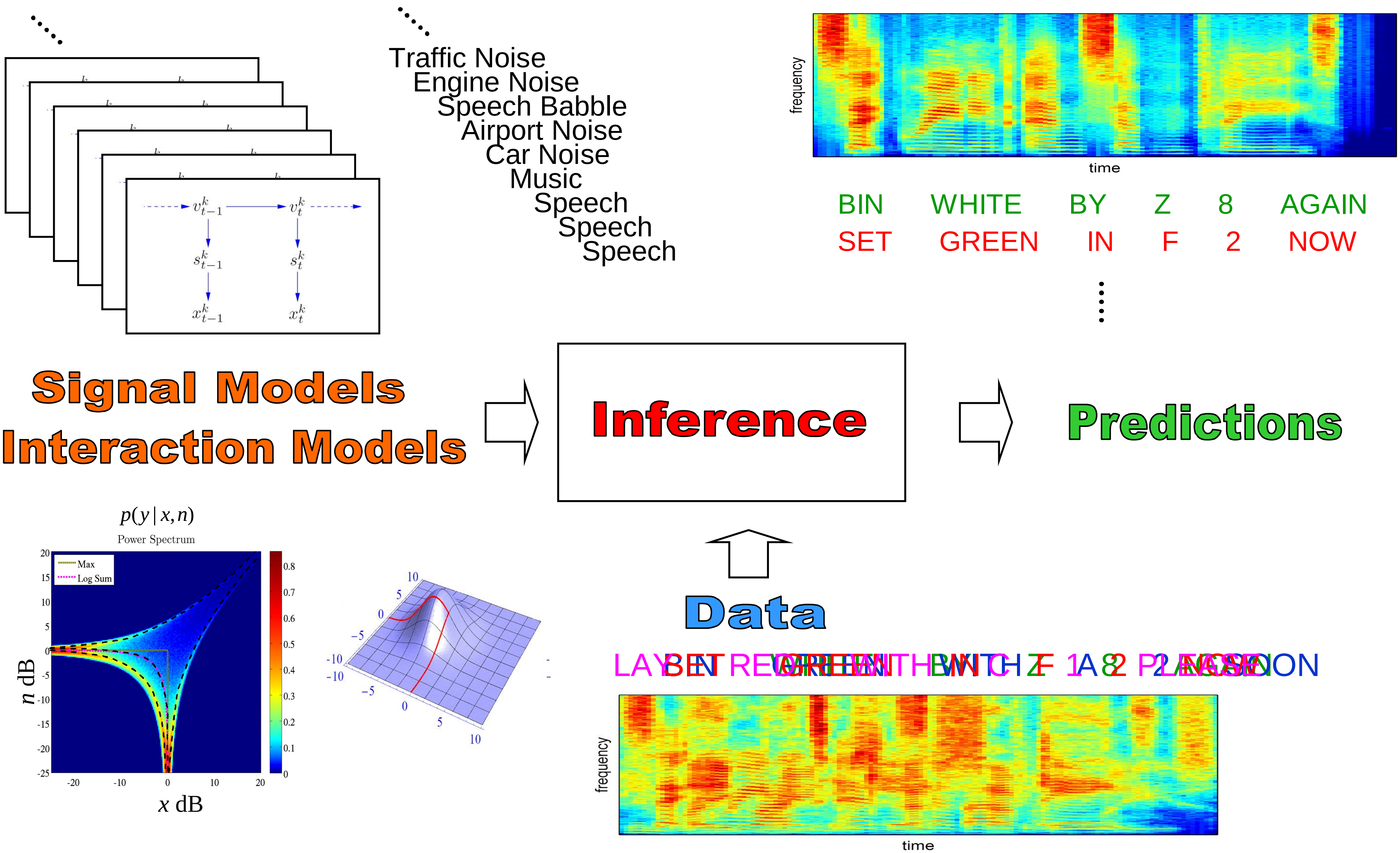
■ - functions of connected variables

Combinatorial Considerations



Inference: $O(\prod_n |s^n|)$

Factorial Models of Noisy Speech

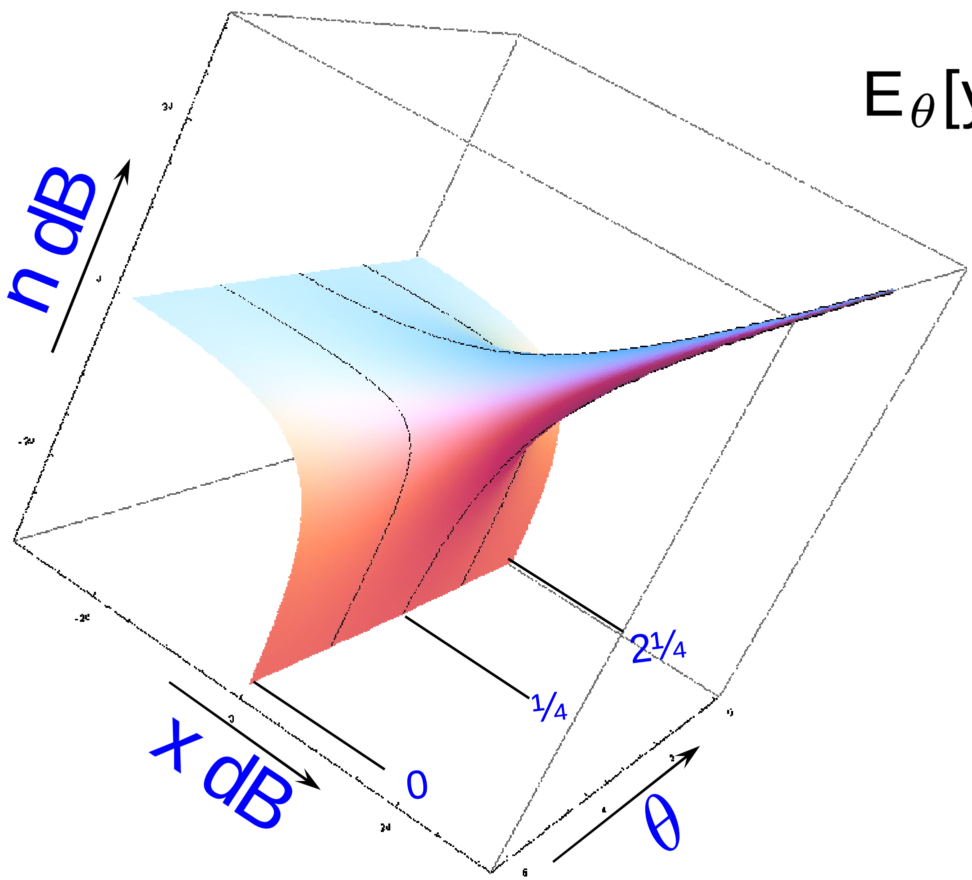


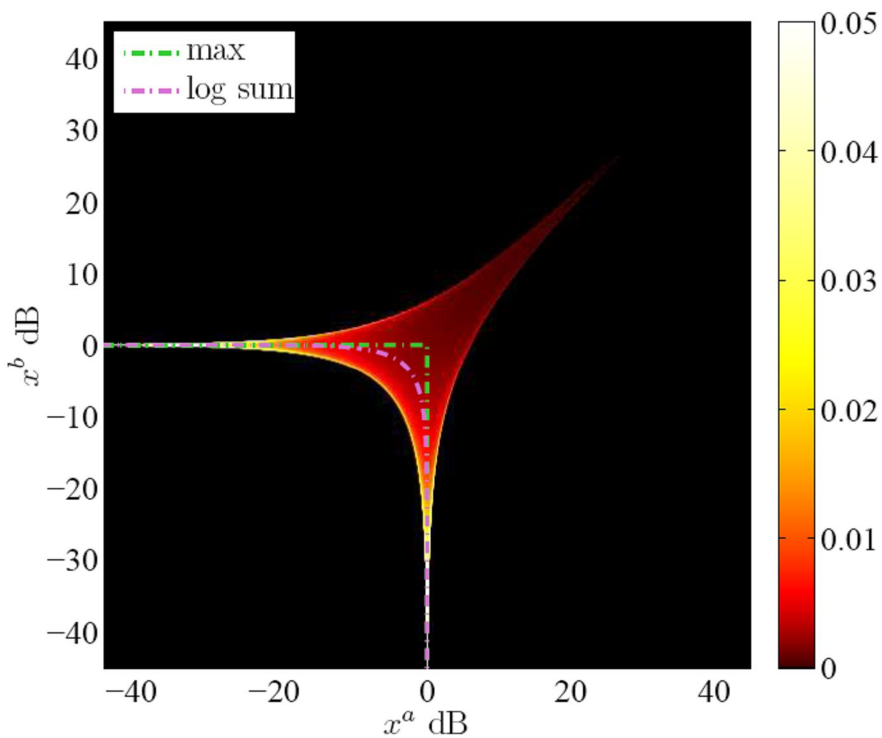
Exact Interaction: signal with additive noise, log domain

¶

$$y = \log(e^x + e^n) + \log\left(1 + \frac{2e^x e^n}{e^x + e^n}\cos(\theta)\right)$$

$$E_\theta[yjx; n] = \max(x; n)$$



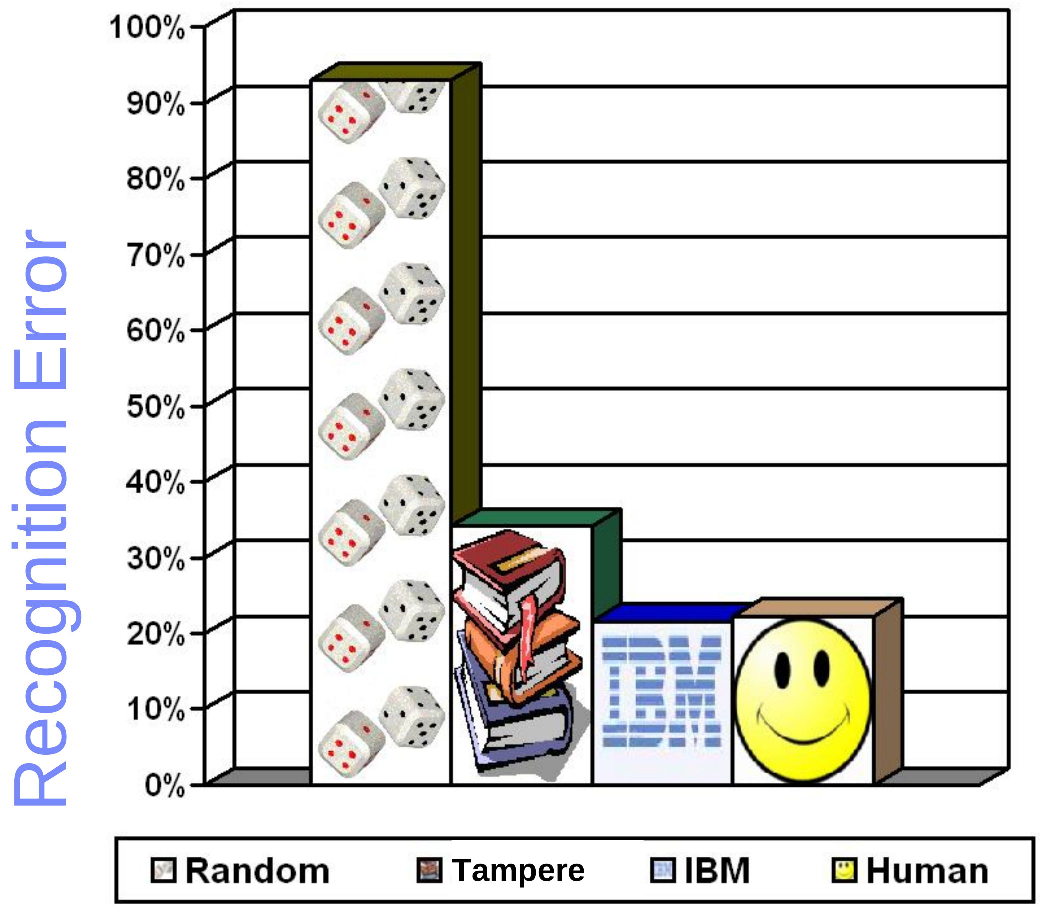


[1] Hershey, J. R., Rennie, S. J., & Le Roux, J. (2012). Factorial Models for Noise Robust Speech Recognition. *Techniques for Noise Robustness in Automatic Speech Recognition*, 311-345.

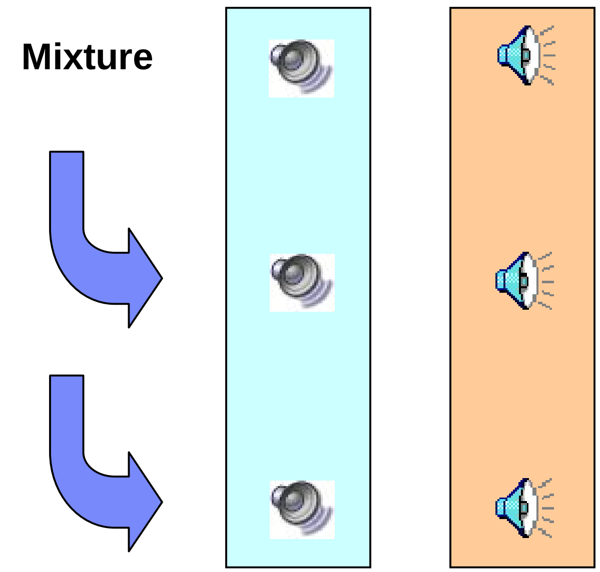
[2] Radfar, M. H. et al., (2012) “Nonlinear minimum mean square error estimator for mixture-maximisation approximation,” Electron. Lett., vol. 42, no. 12, pp. 724–725.

Pascal Speech Separation Challenge:

2006: Factorial HMMs achieve super-human performance on the SSC.

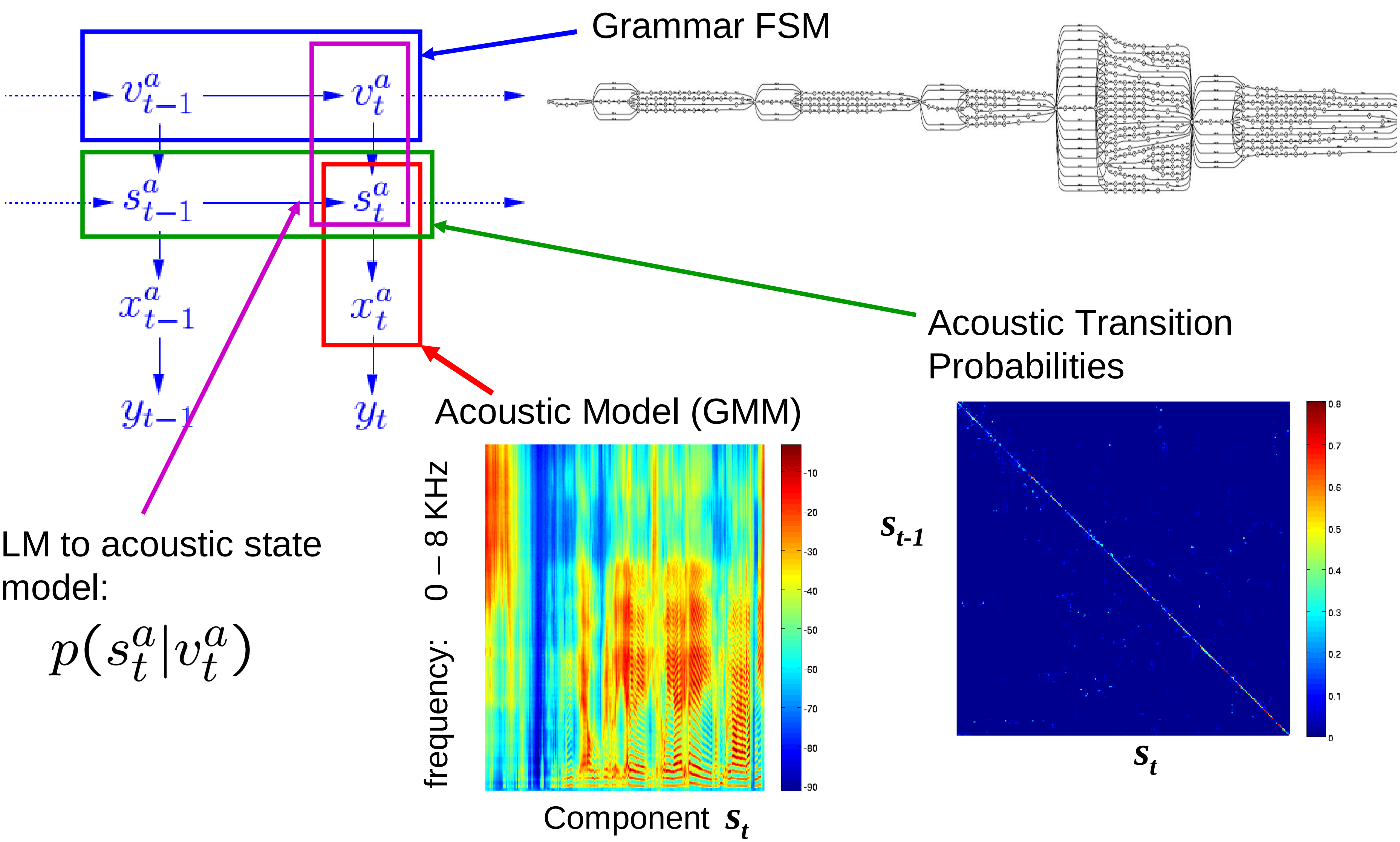


Examples:



[3] Hershey, J. R., Rennie, S. J., Olsen, P. A., & Kristjansson, T. T. (2010). Super-human multi-talker speech recognition: A graphical modeling approach. *Computer Speech & Language*, *24*(1), 45-66.

SSC Model



Grammar FSM
v_{t-1}^a
v_t^a
s_{t-1}^a
s_t^a
x_{t-1}^a
x_t^a
y_{t-1}
y_t
Acoustic Transition Probabilities
Acoustic Model (GMM)
LM to acoustic state model:
$p(s_t^a|v_t^a)$
frequency: 0 – 8 KHz
Component s_t
s_{t-1}
s_t

Beyond Two Sources

2009: New variational methods can separate data with trillions of states

PLACE WHITE AT D ZERO SOON
PLACE RED IN H 3 NOW
LAY BLUE AT P ZERO NOW
PLACE GREEN WITH B 8 SOON

- Excellent separation using a variational posterior with 1K masks/frame

PLACE WHITE AT D ZERO SOON	0 dB
PLACE RED IN H 3 NOW	-7 dB
LAY BLUE AT P ZERO NOW	-7 dB
PLACE GREEN WITH B 8 SOON	-7 dB

Audio demos: http://researcher.watson.ibm.com/researcher/view_project.php?id=2819

[4] Rennie, S., Hershey, J., Olsen, P., Single Channel Multi-talker Speech Recognition: Graphical Modeling Approaches. IEEE Signal Processing Magazine, Vol. 27:6, November 2010.

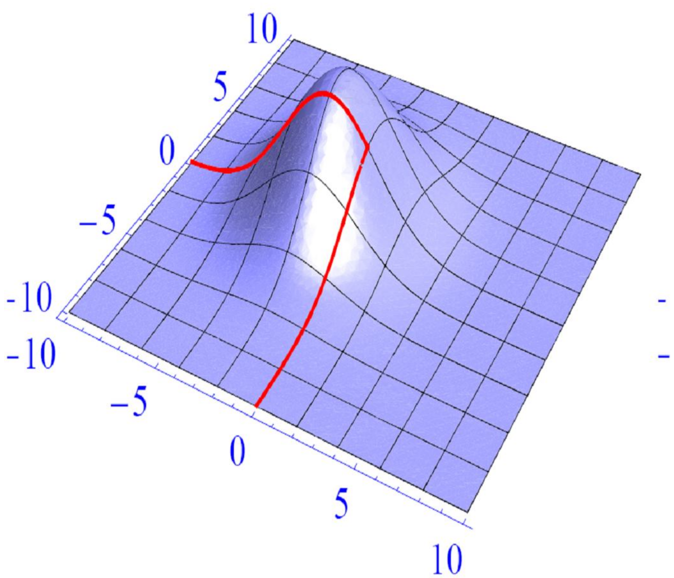
Max Interaction: More than two speakers

Given: $p_{x_f^k}(y_f|s^k);\ \Phi_{x_f^k}(y_f|s^k) \equiv p(x_f^k \le y_f)\ \forall k$

Then: $$p(y_f|\{s^k\}) = \frac{\delta}{\delta y_f}\prod_k \Phi_{x_f^k}(y_f|s^k)$$

$$= \sum_k \boxed{p_{x_f^k}(y_f|s^k)\prod_{j \neq k}\Phi_{x_f^j}(y_f|s^j)}^{*}$$

$$= \sum_{d_f} p(y_f, d_f|\{s^k\})$$



$$\log(p(y_f|\{s^k\})) \ge \sum_k q(d_f)\log\frac{p(y_f, d_f|\{s^k\})}{q(d_f)}$$

*If all masks are known, the sources can be *independently* inferred.

Max Interaction: New Variational Bound

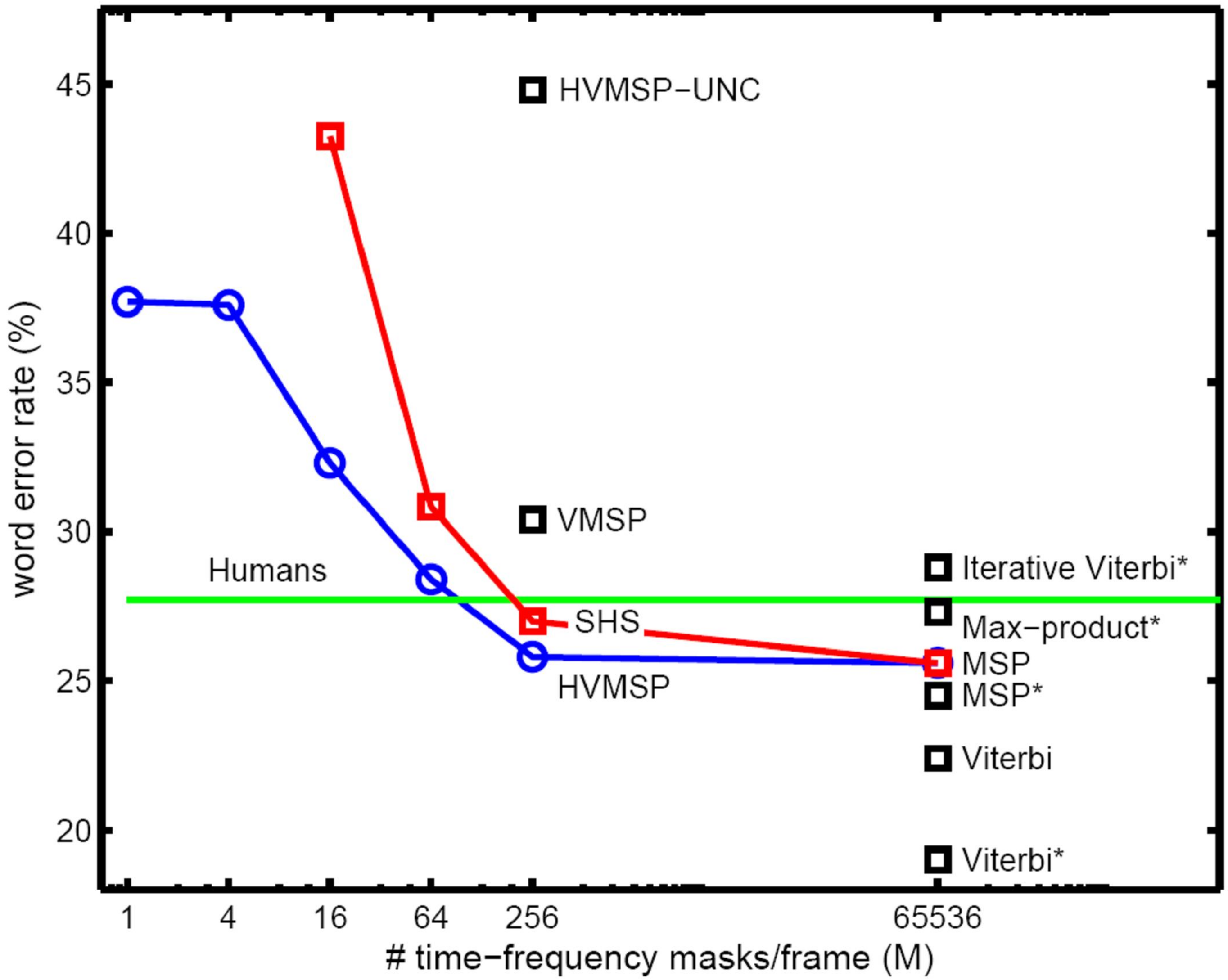
$$\log(p(y_f|\{s^k\})) = \log(\sum_{d_f} p(y_f, d_f|\{s^k\}))$$

$$\geq \sum_{k} q(d_f) \log \frac{p(y_f, d_f|\{s^k\})}{q(d_f)}$$

- If $q(d_f) = p(d_f|y_f, \{s^k\})$ the bound is *tight!*
- *Complexity of inference (i.e. #masks inferred) can be controlled*
- *Models of the sources are utilized to* ***jointly estimate the masks and decode the sources***
- ***Deep connections with CASA and MFT.***

[5] Rennie, S., Hershey, J., and Olsen P . "Hierarchical variational loopy belief propagation for multi-talker speech recognition." *ASRU, 2009.*

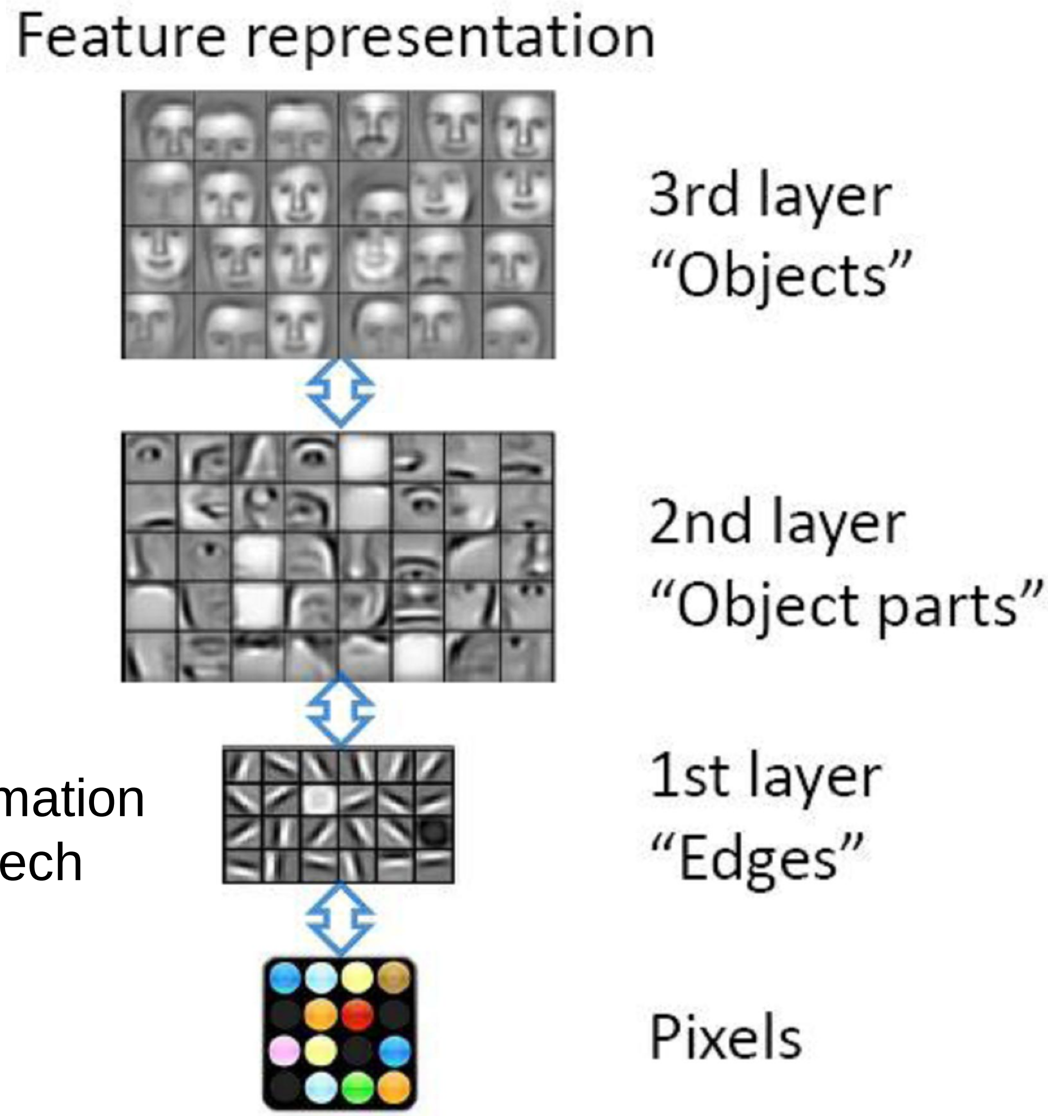
Two Speaker Results



Factorial RBMs for robust ASR

Motivation:

- Learn parts-based models
  - Distributed states
    - Compositional model
    - Better generalization
- Leverage known interactions
  - Instead of learning the transformation from noisy speech to clean speech again and again



Review: Restricted Boltzmann Machines

- **A Markov Random Field (MRF)**
  - Two layers, no connections between hidden layer nodes
  - For binary hidden, Gaussian visible units:

$$\log p(v; h) = -\sum_{i=1}^{V} \frac{(v_i - b_i)^2}{2\sigma_i^2} + \sum_{j=1}^{H} a_j h_j + \sum_{i=1}^{V} \sum_{j=1}^{H} \omega_{ij} v_i h_j - Z$$

  - Form of conditional posterior of hidden units

$$p(h_j = 1 | v) = \frac{\exp(a_j + \sum_{i=1}^{V} \omega_{ij} v_i)}{1 + \exp(a_j + \sum_{i=1}^{V} \omega_{ij} v_i)}$$

$$= \mathrm{sig}(a_j + \sum_{i=1}^{V} \omega_{ij} v_i)$$

Review: Restricted Boltzmann Machines (cont'd)

– Form of conditional prior of a visible unit

$$p(v_i|h) = \frac{\exp\left(\frac{-(v_i - b_i)^2}{2\sigma_i^2} + \sum_{j=1}^{H} \omega_{ij} v_i h_j\right)}{\int_{v_i} \exp\left(-\frac{(v_i - b_i)^2}{2\sigma_i^2} + \sum_{j=1}^{H} \omega_{ij} v_i h_j\right)}$$

$$= \mathcal{N}\left(v_i; b_i + \sigma_i^2 \sum_{j=1}^{H} \omega_{ij} h_j; \sigma_i^2\right);$$

– Can be represented as a mixture of 2^H Gaussians

– Can be evaluated in time ***linear*** in the number of hidden units H since $p(h|v) = \prod_j p(h_j|v)$

Factorial Hidden Restricted Boltzmann Machines

– *Interaction Model* $p(y|v^x, v^n)$ describes how the visible units of multiple RBMs (two here) *generate* observed data

– Inference now ***intractable*** due to explaining away effects

– One solution: *variational methods*

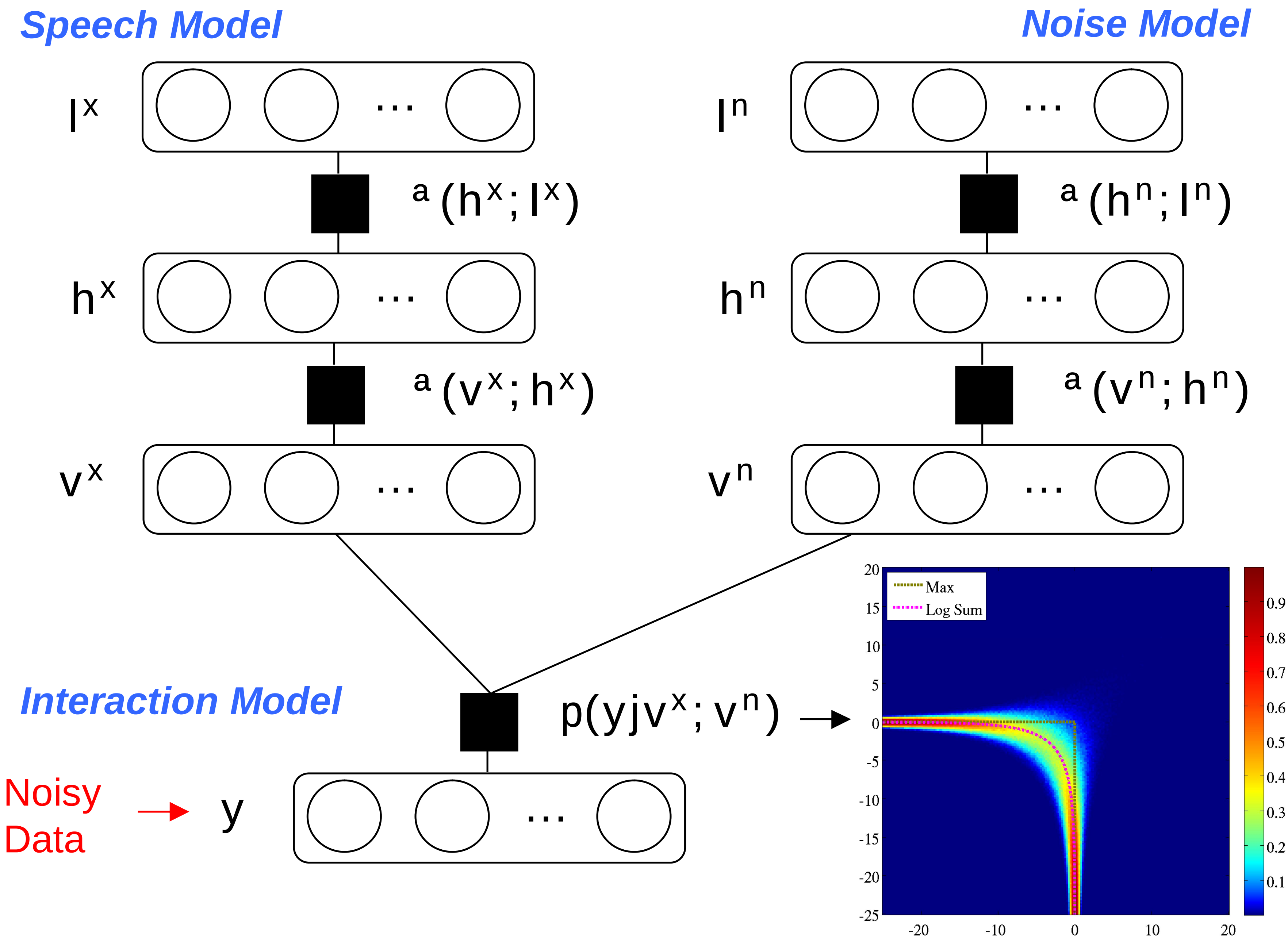
$$\log p(y) = \log \sum_{h,v} p(h^x, v^x) p(h^n, v^n) p(y|v^x, v^n)$$

$$\geq \sum_{h,v} q(h, v) \log \frac{p(h^x, v^x) p(h^n, v^n) p(y|v)}{q(h, v)}$$

$$= E_{q(v^x, v^n)}[\log p(y|v)] + \sum_{i=x,n} E_{q(h^i, v^i)}[\log \frac{p(h^i, v^i)}{q(h^i, v^i)}] \equiv L$$

– Choose *surrogate posterior q* that makes inference tractable (bound tight without structural assumptions on q)

FHRBM Model : Factor Graph



FHRBMs for Robust ASR

- Speech RBM $p(v^x, h^x)$
- Noise RBM $p(v^n, h^n)$
- Interaction Model (log Mel power spectrum)

$$p(y|v^x, v^n) = \prod_f \mathcal{N}(y_f; g(v_f), \psi_f^2), \qquad v_f = [v_f^x \; v_f^n]^T$$

$$g(v_f) = \log(\exp(v_f^x) + \exp(v_f^n))$$

[this choice ignores phase interactions]

- Assumed form of surrogate posterior q

$$q(h^x, v^x, h^n, v^n) = \prod_f q(v_f^x, v_f^n) \prod_{j=1}^{H_x} q(h_j^x) \prod_{k=1}^{H_n} q(h_k^n)$$

$$= \prod_f \mathcal{N}(v_f; \mu_f, \Sigma_f) \prod_{s=x,n} \prod_{j=1}^{H_s} (\gamma_{h_j^s})^{h_j^s} (1 - \gamma_{h_j^s})^{1 - h_j^s}$$

FHRBMs for Robust ASR

– Iteration:

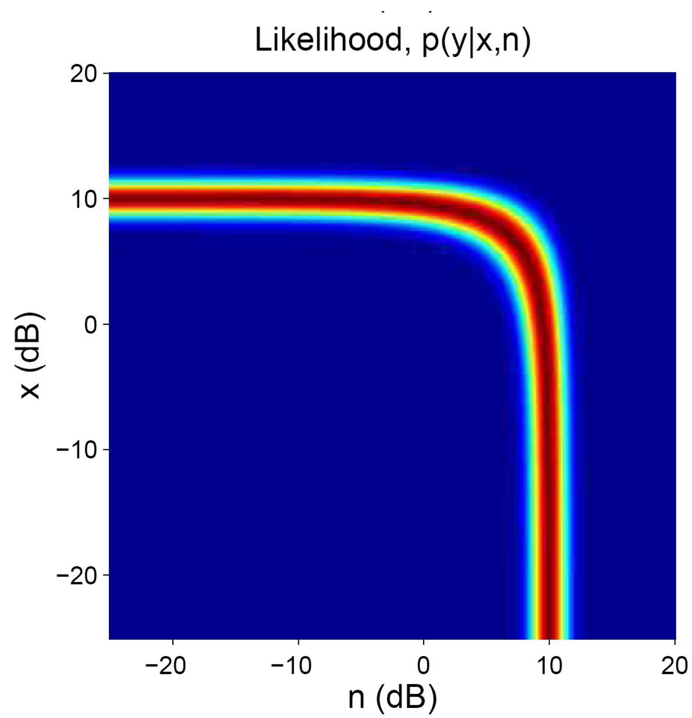
1. Update context-dependent linear approx. of interaction

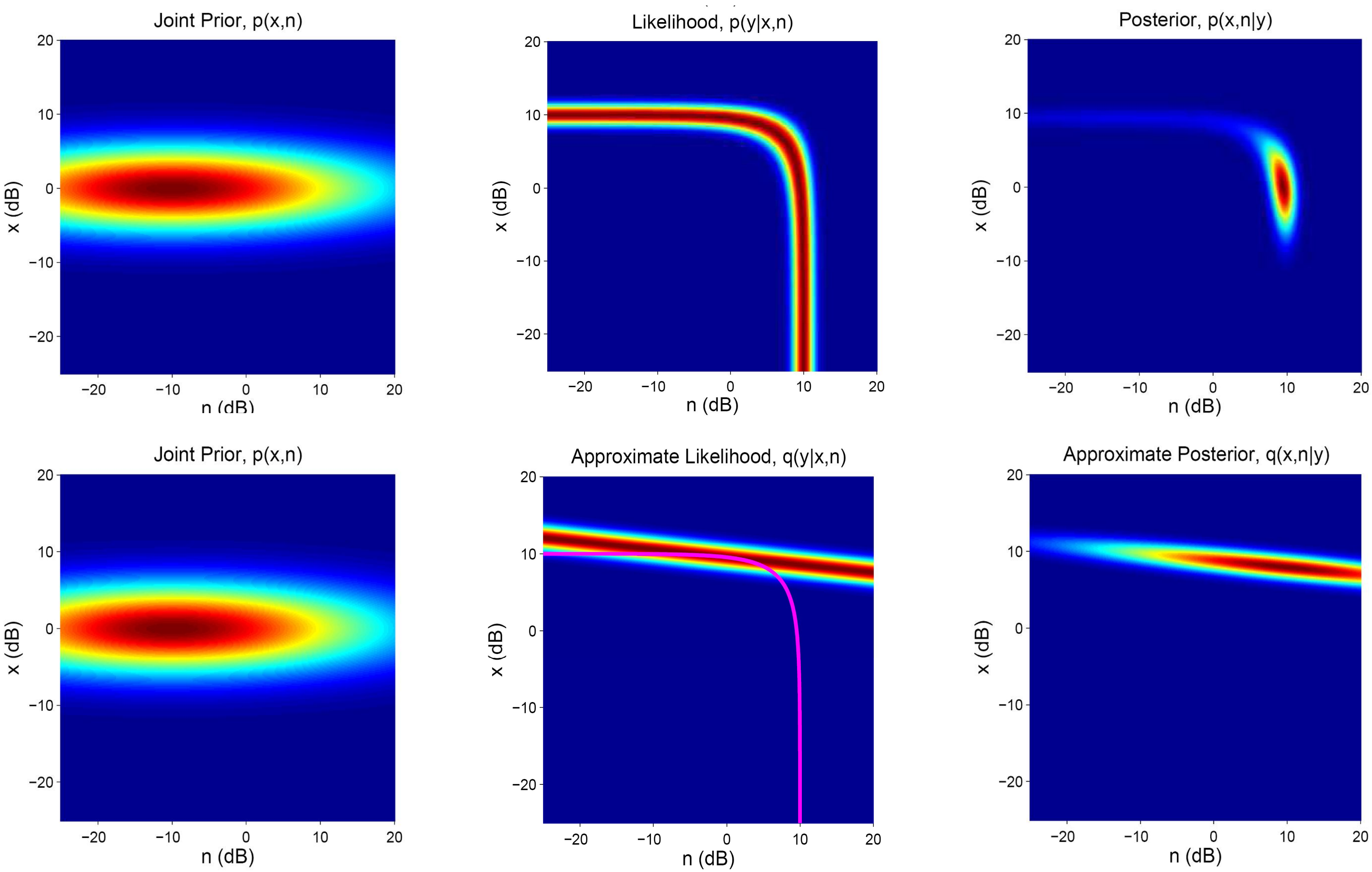
$$p(y|v^x; v^n) \approx \prod_f \mathcal{N}(y_f; g(\mu_f) + (v_f - \mu_f)^T d_f; \sigma_f^2);$$

$$d_f = [d_{v_f^x} \; d_{v_f^n}]^T = \left.\frac{\partial g}{\partial v_f}\right|_{v_f = \mu_f}$$

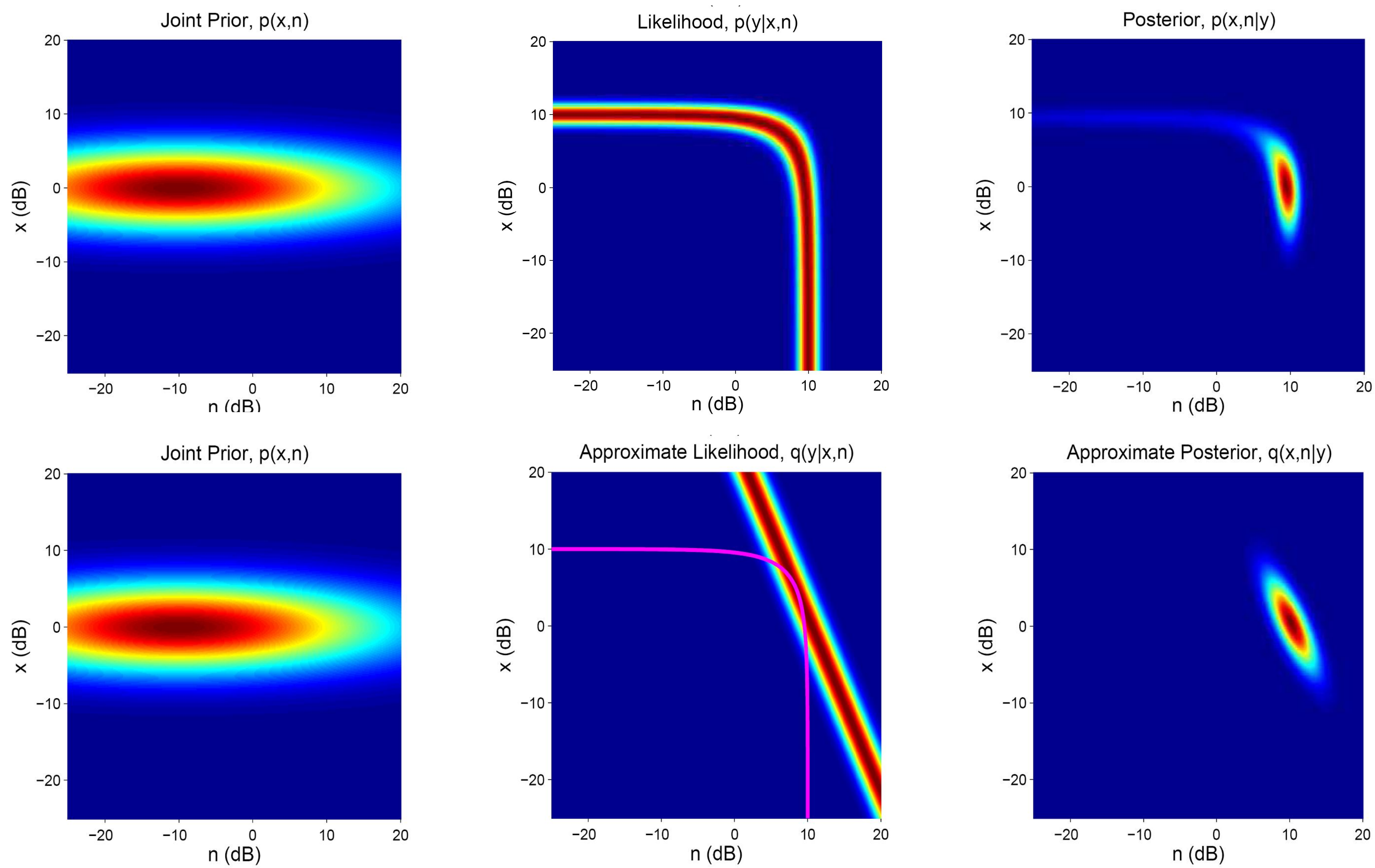
$$d_{v_f^x} = \mathrm{sig}(\mu_{v_f^n} - \mu_{v_f^x})$$

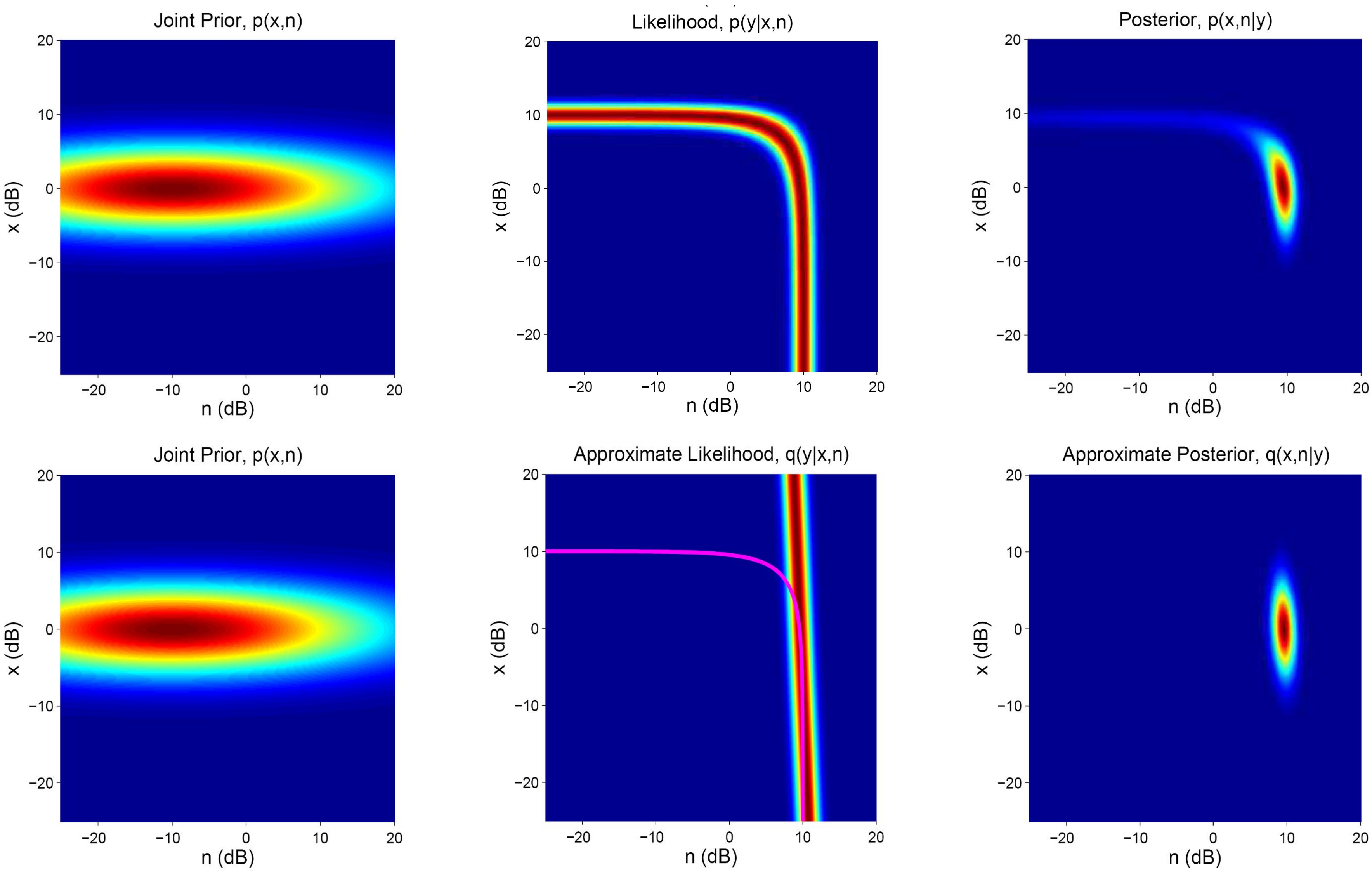
$$d_{v_f^n} = 1 - d_{v_f^x}$$





Joint Prior, p(x,n)
Likelihood, p(y|x,n)
Posterior, p(x,n|y)
Joint Prior, p(x,n)
Approximate Likelihood, q(y|x,n)
Approximate Posterior, q(x,n|y)
x (dB)
n (dB)
20
10
0
−10
−20


Joint Prior, p(x,n)
Likelihood, p(y|x,n)
Posterior, p(x,n|y)
Joint Prior, p(x,n)
Approximate Likelihood, q(y|x,n)
Approximate Posterior, q(x,n|y)
x (dB)
n (dB)


Joint Prior, p(x,n)
Likelihood, p(y|x,n)
Posterior, p(x,n|y)
Joint Prior, p(x,n)
Approximate Likelihood, q(y|x,n)
Approximate Posterior, q(x,n|y)
x (dB)
n (dB)
20
10
0
−10
−20

FHRBMs for Robust ASR

– Iteration:

2. Update the variational parameters of source s

a) $Á^2_{v^s_f} = (¾^{i\,2}_{v^s_f} + d^2_{v^s_f}(Ã^0_f)^{i\,2})^{i\,1}$

b) $^1 v^s_f = Á^2_{v^s_f}(¾^{i\,2}_{v^s_f}(b_{v^s_f} + ¾^2_{v^s_f} P^{H^s}_{j=1} !^s_{fj}\, ° h^s_j) + d_{v^s_f}(Ã^0_f)^{i\,2} y^0_f)$

Influence of source's network — Influence of data

$y^0_f = y_f \; i \; g_{v^s_f}\, ^1 v^s_f$ — $Ã^0_f = Ã_f + g_{v^s_f} ¾^2_f g_{v^s_f}$

c) $° h^s_j = sig(a^s_j + P^{V^s}_{f=1} !^s_{fj}\, ^1 v^s_f)$

3. Toggle s (between s=x and s=n)

[6] Rennie, S. J., Fousek, P., & Dognin, P. L, "Factorial Hidden Restricted Boltzmann Machines for noise robust speech recognition". ICASSP 2012.

Deep FHRBMs for Robust ASR

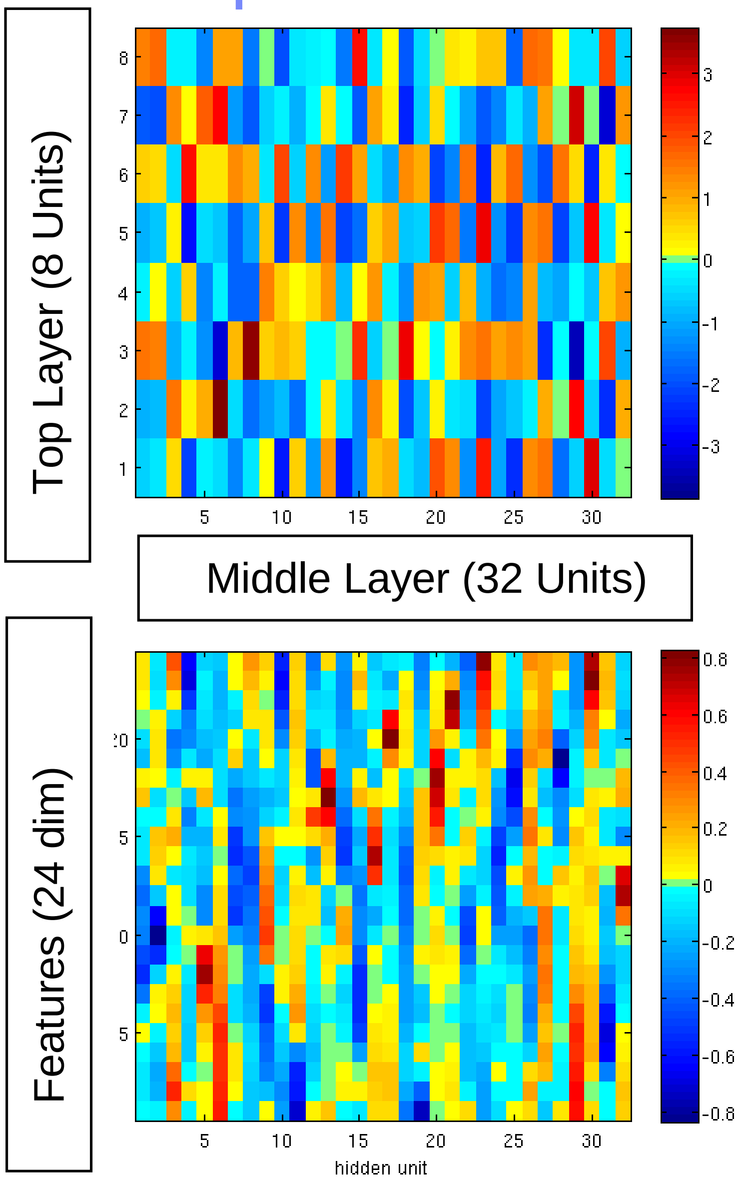
- Updates readily generalize to use of deep belief network (DBNs) of RBMs
- Example: Source RBMs with two hidden layers
  - Top Layer Variables $\quad l^s = \{l_1^s, l_k^s, \ldots, l_{L^s}^s\}$
  - Variational distribution $\quad q(l^s) = \prod_k q(l_k^s) = \prod_k \gamma_{l_k^s}$
  - New update for first hidden layer

$$\gamma_{h_j^s} = \mathrm{sig}\Big(a_j^s + \sum_{i=1}^{V^s} \omega_{ij}^s \mu_{v_i^s} + \alpha_j^s + \sum_{j=1}^{L^s} \varpi_{jk}^s \gamma_{l_k^s}\Big)$$

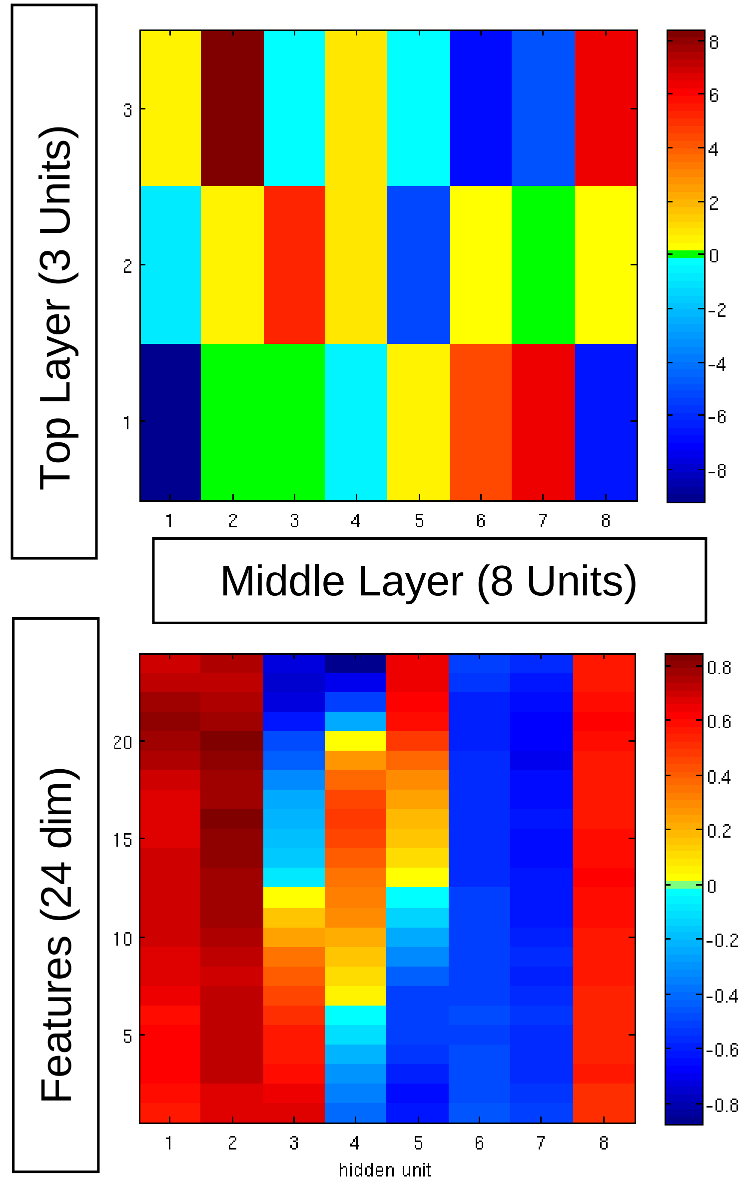
Influence of layer below   Influence of layer above

  - Extension to use of source RBMs with more than two hidden layers straightforward…

Speech RBM



Noise RBM



Experimental Results

- **Task: Test time only noise compensation, noisy in-car speech data**
- **Recognizer:** IBM embedded system (eVV)
- **AM:** 10K Gaussians, 865 CD states
- **LM:** task-specific grammars
- **Training data:** 786 hrs, ~10K speakers, C&C, dialing, navigation
- **Test data:** 206k words, *well matched*

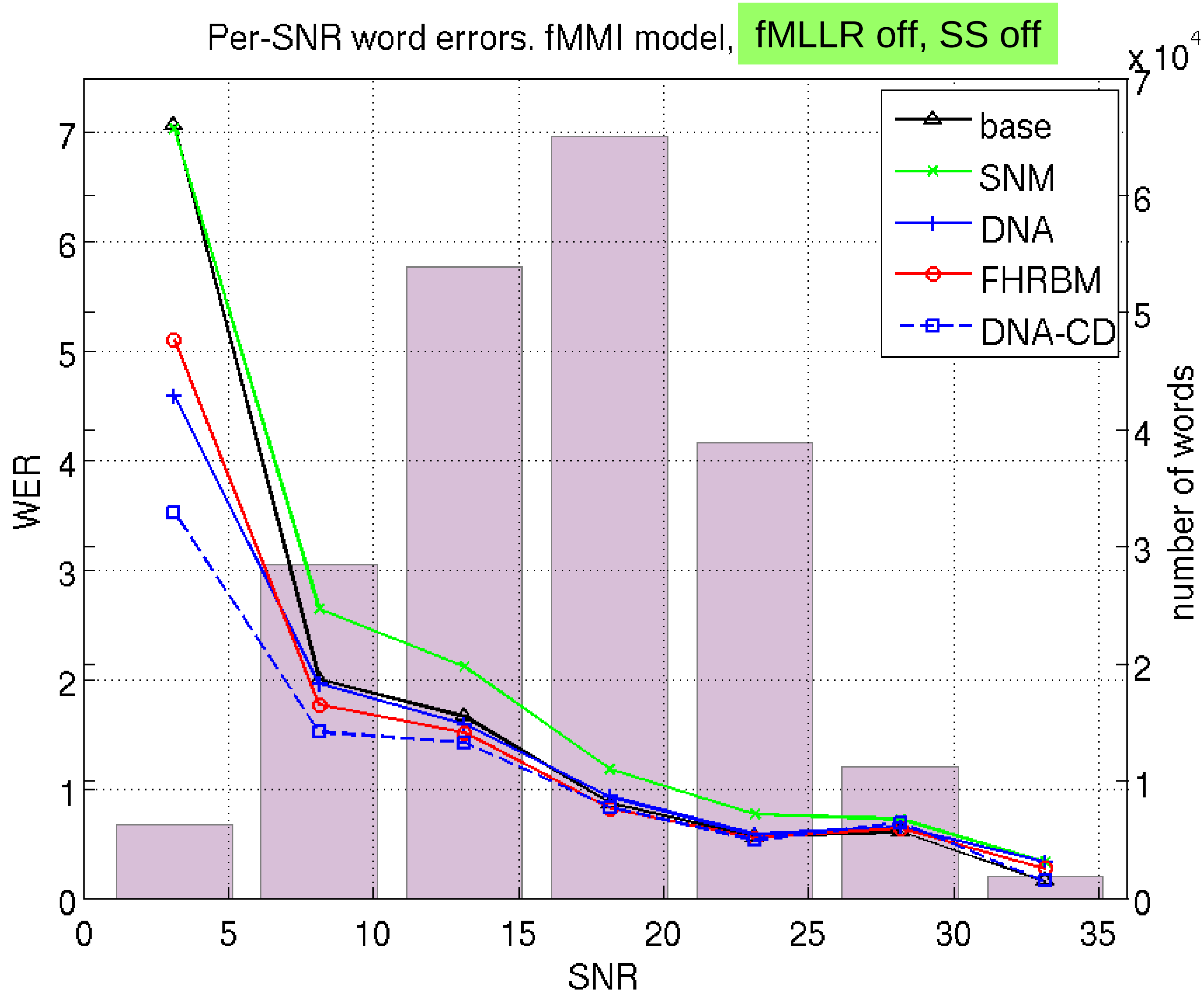
Results (cont'd.)

Algorithm	Speech Model	Noise Model
DNA (CD)	GMM	Gaussian Process
FHRBM	**RBM**	**RBM**
SNM	GMM	Fixed Gaussian

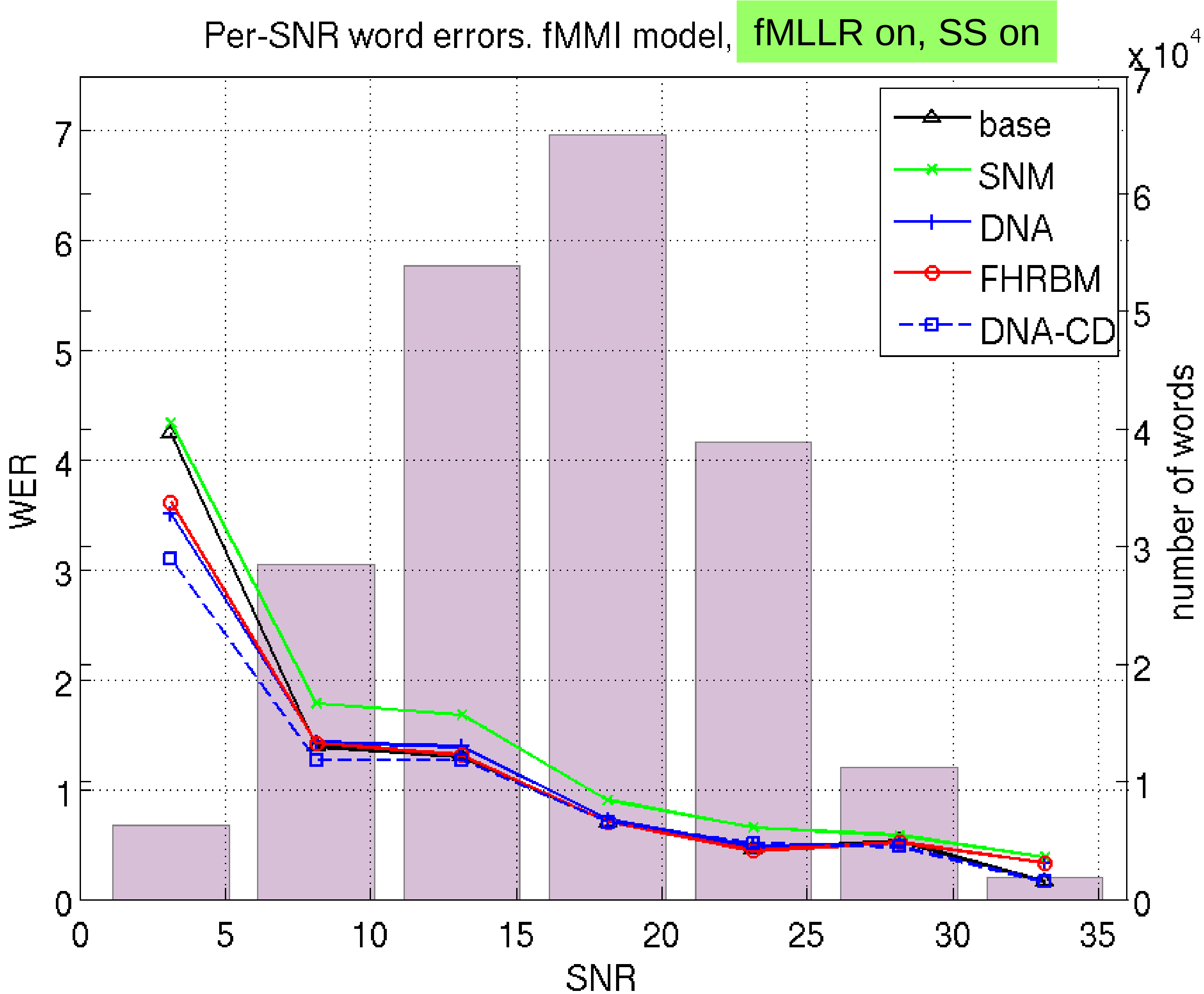
- **WER/SER Ranks** 1 2 3
- $|\theta_{RBM}^{\times}| = |\theta_{GMM}^{\times}|$
- **DNA outperforms use of noise GMM on this task (diffuse evolving noise)**
- **FHRBM outperforms DNA, but not DNA with Condition Detection (CD)**
- **CD could be used with FHRBMs…**

Algorithm	WER rank	WER/SER (%)	SER rank
fMMI (B1)		1.34/3.77	3
B1 + SNM		1.70/5.06	
B1 + DNA	3	1.27/4.04	
B1 + FHRBM	2	1.20/3.51	2
B1 + DNA-CD	1	1.09/3.19	1
fMMI+SS (B2)	2	1.18/3.41	2
B2 + SNM		1.76/5.27	
B2 + DNA		1.34/4.24	
B2 + FHRBM	2	1.18/3.48	3
B2 + DNA-CD	1	1.10/3.17	1
fMMI+fMLLR (B3)		1.08/3.00	3
B3 + SNM		1.25/3.59	
B3 + DNA	3	1.06/3.04	
B3 + FHRBM	2	1.03/2.95	2
B3 + DNA-CD	1	0.93/2.59	1
fMMI+fMLLR+SS (B4)	2	1.00/2.79	2
B4 + SNM		1.26/3.56	
B4 + DNA		1.02/3.03	
B4 + FHRBM	3	0.99/2.82	3
B4 + DNA-CD	1	0.95/2.67	1

Results (cont'd.) – WER vs. (biased) SNR



Results (cont'd.) – WER vs. (biased) SNR



Thoughts

- **Results/investigations quite preliminary**
  - **Models**
    - DNA: (matched) quasi-stationary noise model, speech GMM
    - FHRBM: no dynamics yet, tiny RBMs
  - **SNR estimates of each frequency band**
    - DNA: estimated uniquely for every speech state for each frame
    - FHRBM: single set of SNR estimates for each frame
  - **Initialization**
    - DNA: noise model initialized on first 10 frames
    - FHRBM: only state posterior (feature layer not yet adapted)
- **Need to evaluate FHRBMs on more general noise containing non-stationary & structured elements**
- **Need to explore model/inference procedures further: e.g. FHRBM a bootstrap for fast feed-forward system?**

Direct Product Based Deep Neural Networks

Motivation:

- **Resurgence of interest/success with DNNs for ML**
  - New algorithms, more data, better machines
- **Still time-consuming to train**
  - Restricts neurons/layer, #layers utilized

Idea:

- **Learn networks with connections that can be represented using sums of *direct products***
  - Make it feasible to learn networks with ***millions*** of neurons

Direct Product DNNs

- **Constrain the weight matrix W to be a sum of *direct products***

$$W = \sum_i A_i \otimes B_i;$$

  - Direct products: Kronecker, outer, “box” product
  - Low rank W a DPDNN, Input layers are naturally Kronecker-structured for spliced input data
  - A structured weight-tying strategy that
    - Facilitates efficient matrix multiplication, storage
    - Composes W from sets of “complete” bases: exact representation always possible

Review

- **The Kronecker Product (KP)**

$$W = A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{pmatrix}$$

Interesting Facts:

- FFT can be expressed as a recursive factorization of the DFT matrix using KPs
- So can several combinatorial algorithms
- Any circulant Matrix can be diagonalized by DFT

Kronecker Product DNNs

$$W = \sum_i A_i \otimes B_i; \qquad \begin{array}{l} W \in \mathbb{R}^{M \times N} \\ A_i \in \mathbb{R}^{M_i \times N_i}; B_i \in \mathbb{R}^{O_i \times P_i} \end{array}$$

- **Efficient Matrix Multiplication**

$$(A_i \otimes B_i)\mathrm{vec}(Z) = \mathrm{vec}(B_i Z A_i^T)$$

  – For M=N, A, B square $O(N^2) \to O(N^{3/2})$

  – E.g. Multiplying a vector by 10K x 10K matrix requires only 1 million rather than 100 million scalar multiplications.

- **Efficient Storage**

  – For M=N, all A,B square $O(N^2) \to O(2N)$

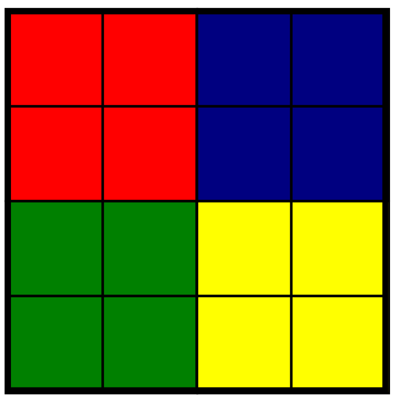
  – A 1M x 1M matrix has only 2 million parameters, rather than 1 trillion

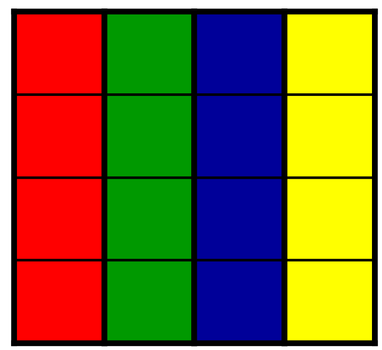
Factoring Existing DBNs

- **Recall**

$$W = \sum_i A_i \otimes B_i; \quad A_i \in R^{M_i \times N_i}; B_i \in R^{O_i \times P_i}$$

- **If all A,B dims are independent of i, reduces to an SVD problem (Van Loan, 1992)**

$W =$ 

$\tilde{W} =$  $= UDV^T$

- **Spliced features lead to Kron-structured W**

[7] Fousek, P., Rennie, S., Dognin, P., and Goel, V., “Direct Product Based Deep Belief Networks for Automatic Speech Recognition”. ICASSP 2013.

Learning/Inference

- **Forward Pass**

$$z_j = \sigma(x_j) = \sigma(W z_{j-1} + b_j)$$

$$x_j = \sum_i \mathrm{vec}(B_i Z_i^{j-1} A_i^T) + b_j$$

- **Error Back-propagation**

$$\delta_{j-1} = \sigma'(x_{j-1}) \cdot W^T \delta_j$$

$$= \sigma'(x_{j-1}) \cdot \sum_i \mathrm{vec}(B_i^T \Delta_i^j A_i)$$

$$\frac{\partial E}{\partial A_i} = \Delta_i^{(j)^T} B_i Z_i^{j-1}; \qquad \frac{\partial E}{\partial B_i} = \Delta_i^j A_i Z_i^{(j-1)^T}$$

Experiments

- **50 hr English Broadcast News (EBN) task**
- **Training: 50 hours 1996/1997 EBN (5/50 Dev.)**
- **Test: 3 hrs EARS dev-04f set**
- **Acoustic Model**
  - Hybrid (NN fully replaces GMM)
  - 2200 acoustic targets
  - Features: 13 dim. PLP -> VTLN -> CMS -> splice ±4 frames -> 117 dim. input features
  - Baseline:
    - NN topology: 117 -> 1K -> 1K -> 2200
    - NN training: Stochastic Gradient, CE (no pre-training)
    - WER: 23.0%

Training DPDBNs

- **Poor Man's Trainer:**
  - Enforce Kronecker structure via periodic SVD during training (first layer only)
  - Not effective

$$W = \sum_i A_i \otimes B_i;$$

$$W \in \mathbb{R}^{1024 \times 117}$$

$$A_i \in \mathbb{R}^{1 \times 9}$$

$$B_i \in \mathbb{R}^{1024 \times 13}$$

terms	%FAcc	%WER
1	34.1	24.3
2	33.7	25.0
3	34.0	24.5
all (baseline)	35.0	23.0

Training DPDBNs

- **By Projected Gradient**
  - Project full gradient onto representation
  - Pros: easy, correct, sub-routine of existing trainer
  - Cons: no training speedup, can't train large W

L1 Topology	FAcc	WER	L1 Param. Reduction
[1024,117] (base)	35.0	23.0	1
1x[32*9, 32*13]	32.8	25.0	170
2x[32*9, 32*13]	32.9	24.9	85
3x[32*9, 32*13]	33.2	24.6	57

IBM

Results

Standard DNN					DPDNN				
Topology L1/L2/L3	PR L1/L2/L3	PR DNN	FAcc	WER	Topology L1/L2/L3	PR L1/L2/L3	PR DNN	FAcc	WER
(740,117) (740,740) (2220,740)	1.4 1.9 1.4	1.5	32.7	26.4	5x(32*32, 9*13) 10x(32*32,32*32) (2220,1024)	27 49 1	1.5	33.7	24.8
(280,117) (280,280) (2220, 280)	3.7 13.4 3.7	4.7	31.2	27.7	5x(32*32, 9*13) 10x(32*32,32*32) 10x(2220*1,32*32)	27 49 3.2	4.7	31.9	26.9
(135,117) (135,135) (2220, 135)	7.6 57.3 7.6	10.3	28.8	31.2	5x(32*32, 9*13) 20x(32*32,32*32) 4x(2220*1, 32*32)	27 25 7.9	10.3	29.8	29.1
(2k, 117) (1k, 2k) (2220, 1k)	0.5 0.5 1.0	0.75	34.7	23.5	5x(64*64, 9*13) 10x(64*64,64*64) (2220,1024)	11 25 1	1.5	35.0	23.5

Trend: For fixed #params, DPDNNs outperform standard DNNs.

Current/Future Work

- **Native DPDBN Trainer**
  - Operational, experimentation in progress
    - Currently training 100K by 100K weight matrices
- **Generalization of estimation framework**
  - to non-uniform direct products of non-uniform size, and transformations thereof
- **Test interactions**
  - RELU, dropout, input noisification,…
- **Investigate on composite data**
  - Factorizations correspond to independence assumptions…

Scalar-Matrix Function Optimization

The Anatomy of the Hessian

Theorem: Any scalar-matrix function $f(\mathbf{X})$ formed using $\mathrm{trace}()$, $\log\det()$, $()^T$, and arithmetic operations ($+$, $-$, $*$ and $()^{-1}$) has a Hessian of the form:

$$f''(\mathbf{X}) = \sum_{i=1}^{k_1} \mathbf{A}_i \otimes \mathbf{B}_i + \sum_{i=k_1+1}^{k_2} \mathbf{A}_i \boxtimes \mathbf{B}_i + \sum_{i=k_2+1}^{k} \mathrm{vec}(\mathbf{A}_i)\mathrm{vec}^\top(\mathbf{B}_i),$$

This allows the Hessian to be efficiently utilized...

$$f''(\mathbf{X})\mathrm{vec}(\mathbf{C}) = \mathrm{vec}\left(\sum_{i=1}^{k_1} \mathbf{B}_i\mathbf{C}\mathbf{A}_i^\top + \sum_{i=k_1+1}^{k_2} \mathbf{B}_i\mathbf{C}^\top\mathbf{A}_i^\top + \sum_{i=k_2+1}^{k} \mathbf{A}_i\,\mathrm{trace}(\mathbf{C}^\top\mathbf{B}_i)\right).$$

[8] Chin, G., Nocedal, J., Olsen, P., Rennie, S., “Second Order Methods for Optimizing Convex Matrix Functions”, IEEE Transactions on Audio, Speech, and Language Processing, Special Issue on Large-scale Optimization, Vol. 20, No. 6, 2013.

Scalar Matrix Function Optimization

- **Efficiently Solving the Covariance Selection Problem**
  - Problem: infer a sparse (L1-regularized) inverse covariance matrix that maximizes the probability of a dataset

$$\mathbf{P}^* = \arg\max_{\mathbf{P}\succ 0} \log\det(\mathbf{P}) - \mathrm{trace}(\mathbf{SP}) - \lambda\|\mathrm{vec}(\mathbf{P})\|_1,$$

  - Approach: Iteratively apply Newton-like algorithms on locally quadratic approximations to the objective
    - Efficient inference by exploiting sparsity & structure of Hessian
  - Applicability: recently shown that covariance selection can be used to infer the structure of more general networks (e.g. discrete)

[9] Olsen, P., Oztoprak, F., Nocedal, J., Rennie, S., Newton-Like Methods for Sparse Inverse Covariance Estimation, NIPS 2012.
[10] Loh, P., Wainwright, M., “No voodoo here! Learning discrete graphical models via inverse covariance estimation”, NIPS 2012.

Closing Remarks

- **Questions to ponder**
  - The evolving role of models that can “explain away” phenomena
    - Are feed-forward representations sufficient?
  - The known and still poorly understood limitations of current neural networks
    - More teaching, less tuning
  - The increasingly important role of optimization methods in machine learning and signal processing
    - Help the machine help itself
  - The role of separation and robustness research in ASR
    - Commercial systems are now *very* good, but the NN revolution is blurring the distinction between core and robust ASR.

Thank-you.