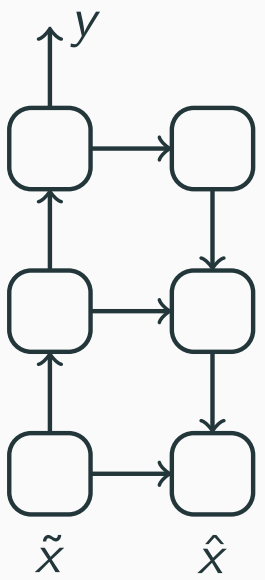
Recurrent Ladder Networks

Alexander Ilin and Isabeau Prémont-Schwarz

The Curious AI Company

Ladder Networks

- Neural network with
  - encoder (bottom-up pass)
  - decoder (top-down pass)
  - lateral connections
  - bottom task: denoising of input
  - top task: e.g. classification

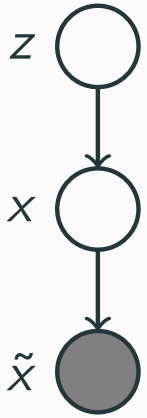


- *Rasmus et al. Semi-supervised learning with Ladder networks. NIPS-2015*
- State-of-the-art (2015) results on semi-supervised classification of MNIST

Denoising encourages learning a probabilistic model

- Denoising encourages learning a probabilistic model of the data

$$p(x|\tilde{x}) = \int p(x|z)p(z|\tilde{x})dz$$

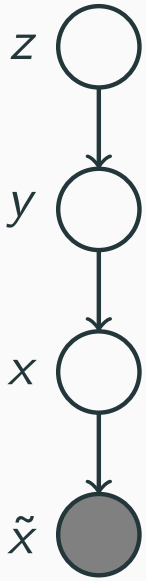


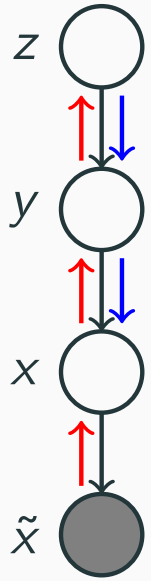
- Optimal denoising function (Curious blog):

$$g(\tilde{x}) = \tilde{x} + \sigma_n^2 \nabla_{\tilde{x}} \log p(\tilde{x})$$

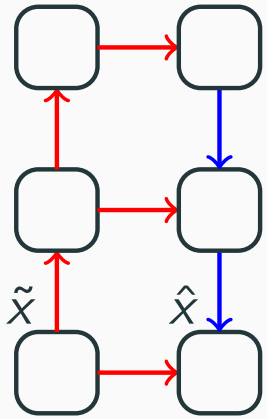
Ladder learns an inference procedure

- Ladder emulates (message-passing algorithm) an inference procedure in an implicit probabilistic model





Message passing

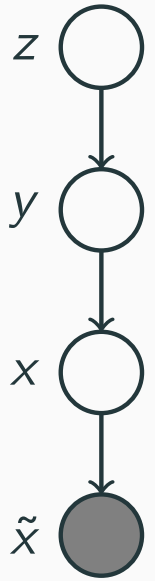


Ladder

Ladder: Gatings in the decoder

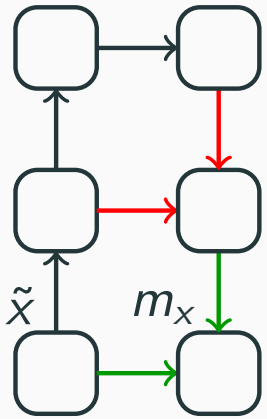
- Simple probabilistic model:

$$p(z) = N(z|\mu_z, \sigma_z^2)$$
$$p(y|z) = N(y|w_{yz}z, \sigma_y^2)$$
$$p(x|y) = N(x|w_x y, \sigma_x^2)$$
$$p(\tilde{x}|y) = N(\tilde{x}|w_x y, \sigma_x^2 + \sigma^2)$$



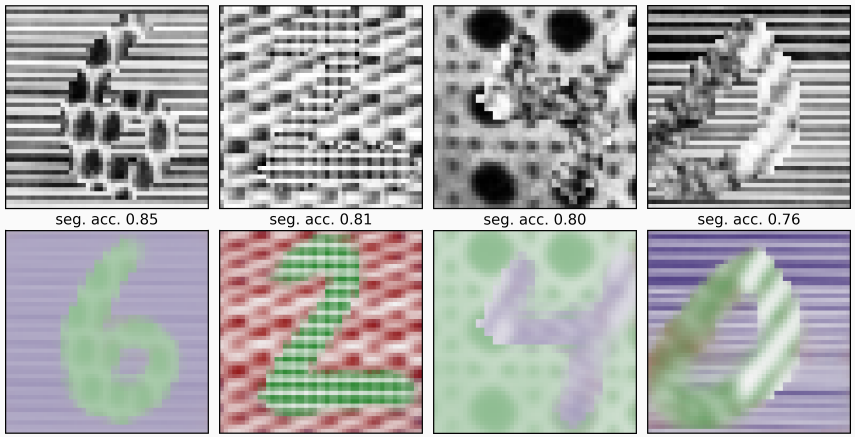
- Posterior approximation: $q(y) = N(y|m_y, v_y)$
- Derived message-passing updates:

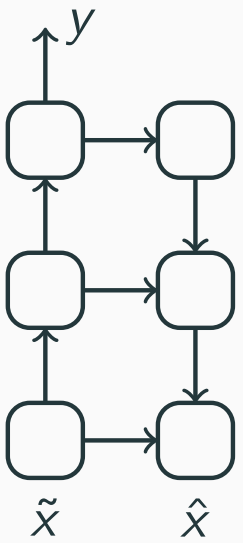
$$m_y = s_y \frac{m_x}{w_x} + (1 - s_y) w_{yz} m_z$$
$$s_y = \text{sigmoid}(\log \sigma_y^2 w_x^2 - \log \sigma_x^2))$$



Iterative inference with RLadder

- In complex cognitive tasks it can be extremely difficult to come up with the right solution in one iteration

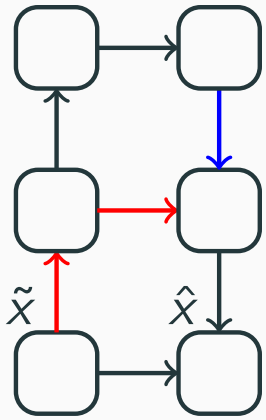




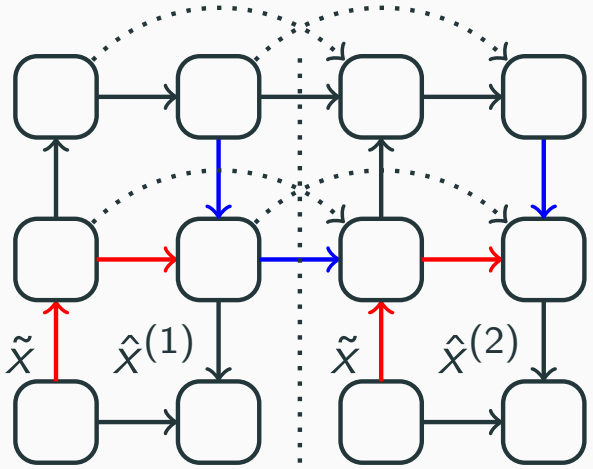
- Nonlinear probabilistic graphical models: derived inference is iterative

Iterative inference with RLadder

- From Ladder to recurrent Ladder:



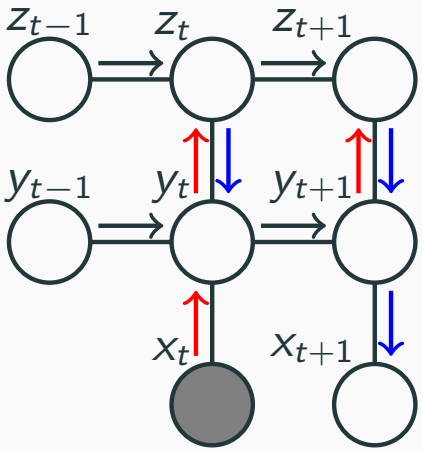
Ladder



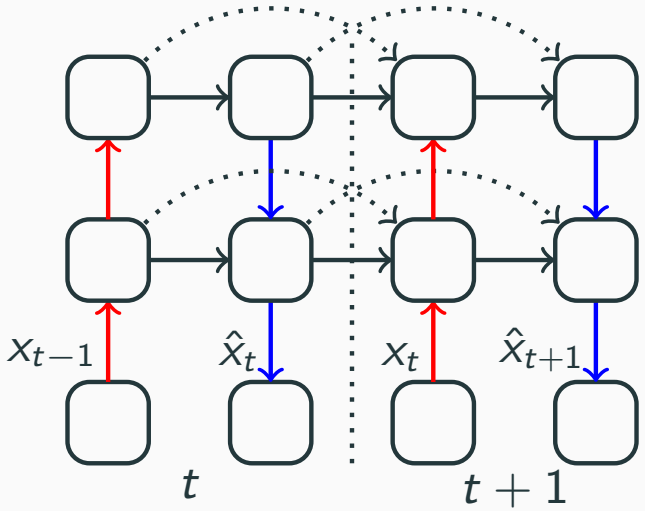
RLadder

Temporal modeling with RLadder

- RLadder can be used for temporal modeling
- Inference in temporal models: update the distribution of states at every time instance
- Combine messages from past, from below and from above



Graphical model

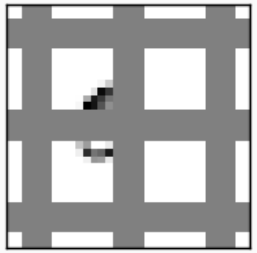


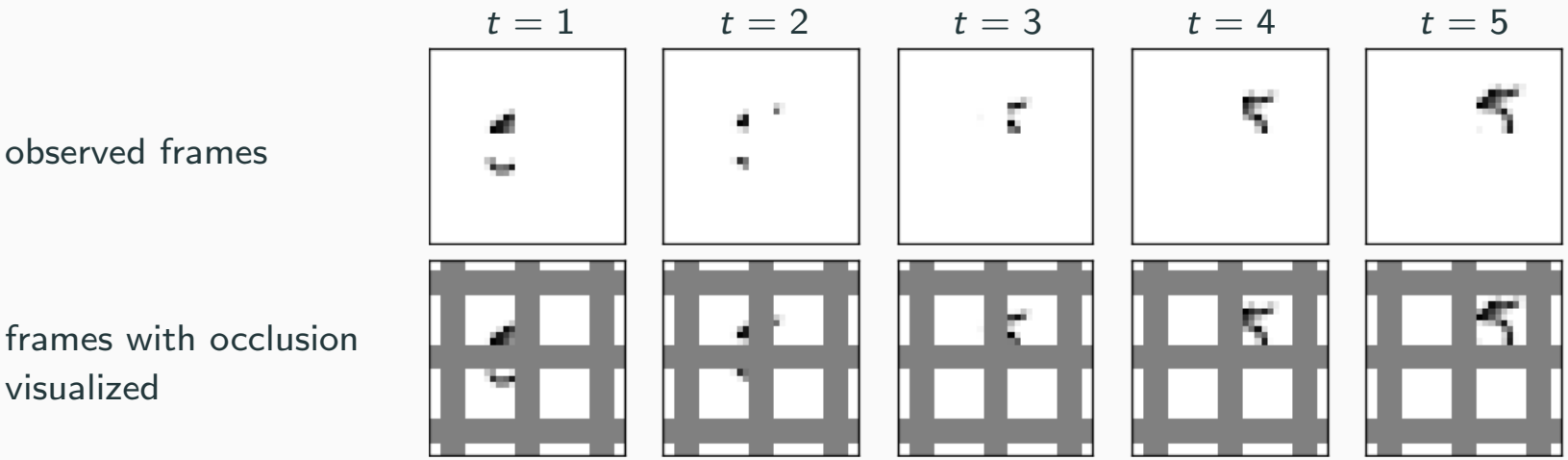
Recurrent Ladder (RLadder)

Experiments: Temporal modeling with RLadder

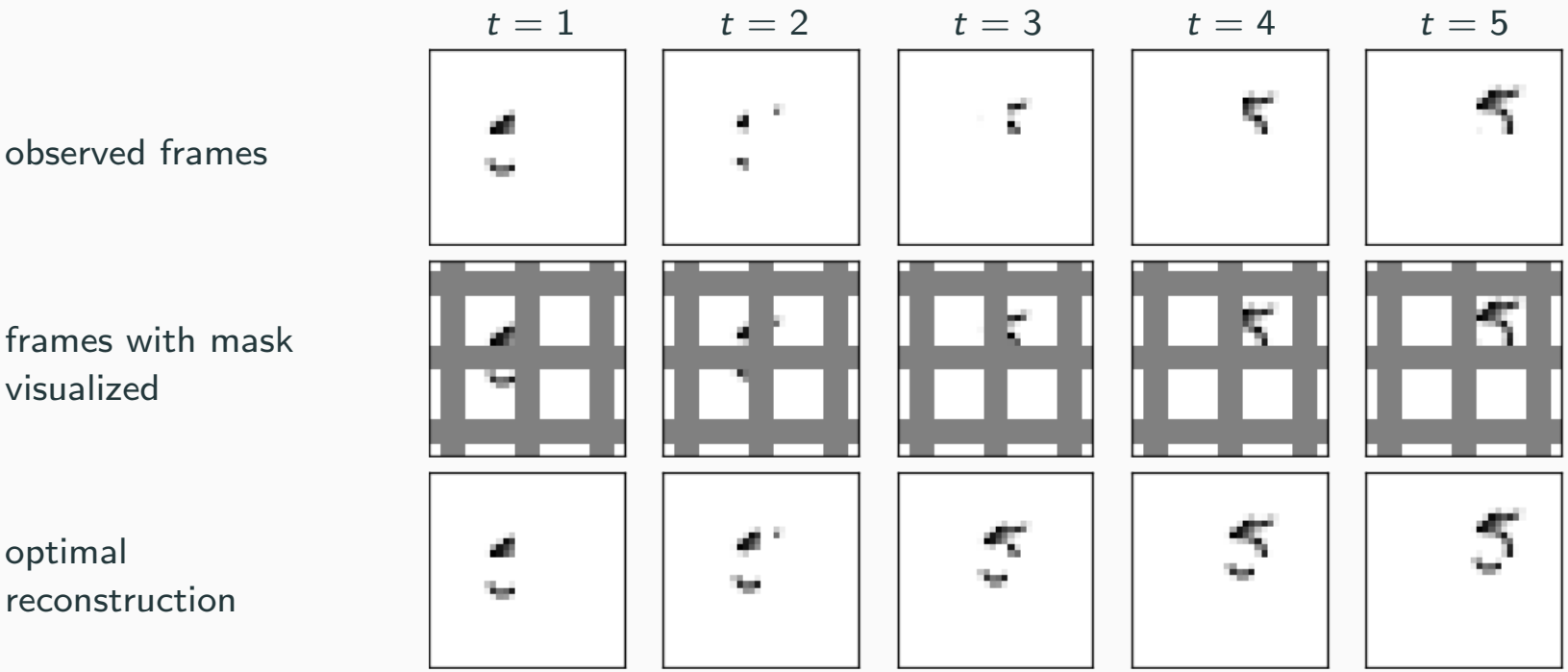
Occluded moving MNIST

- Digits moving on a canvas occluded by bars
- Top-level task (primary): Classify digit
- Low-level task (auxiliary): Next-frame prediction



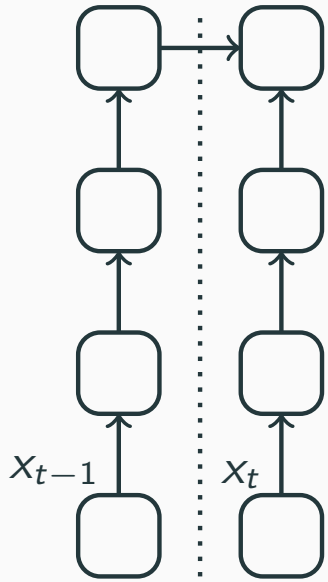


Comparison models: Optimal reconstruction

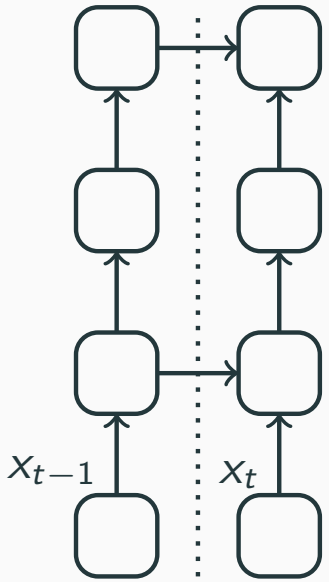


- Optimal reconstructions are fed to a static classifier

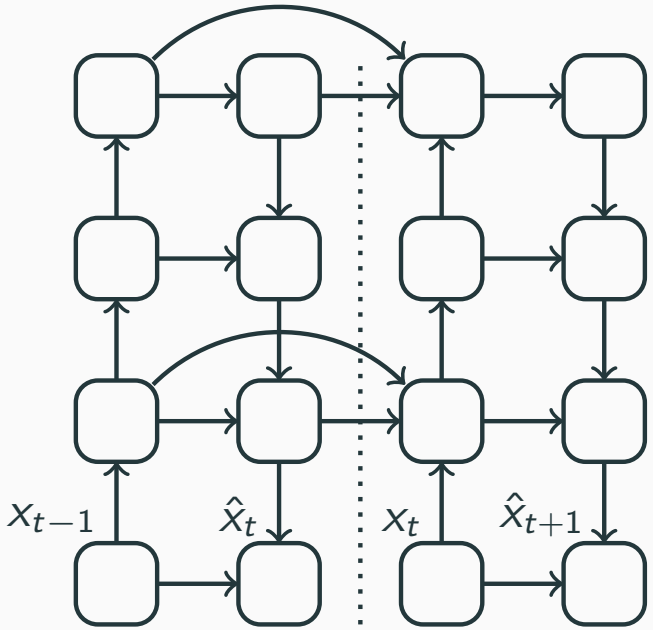
Comparison models



2) Temporal baseline



3) Hierarchical RNN

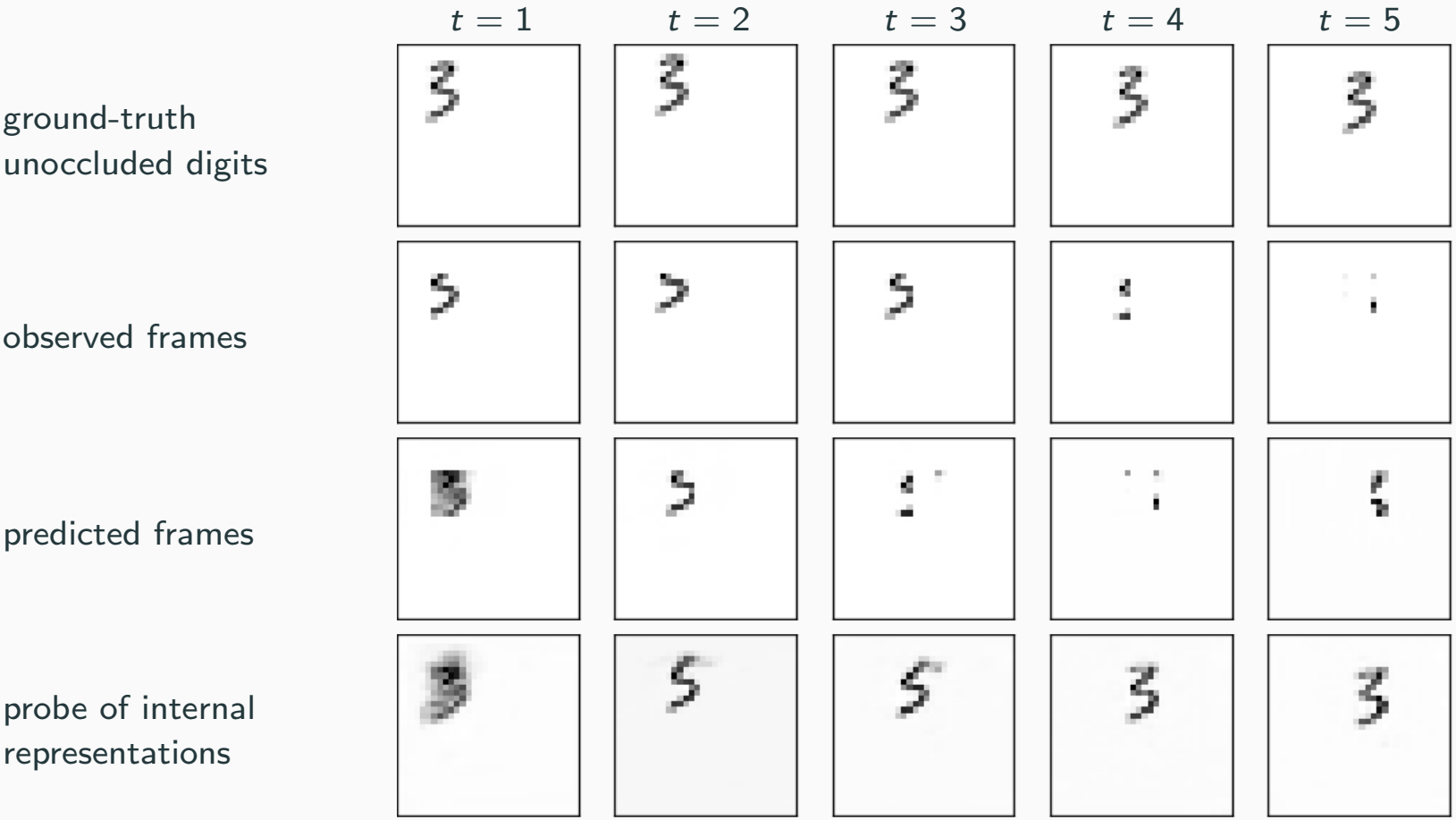


4) RLadder

Fully supervised learning results

	Classification error (%)	Prediction error, $\cdot 10^{-3}$
Optimal reconstruction and static classifier	0.71 ± 0.03	
Temporal baseline	2.02 ± 0.16	
Hierarchical RNN (encoder only)	1.60 ± 0.05	
RLadder w/o prediction task	1.51 ± 0.21	
RLadder w/o decoder-to-encoder conn.	1.24 ± 0.05	$\mathbf{1.567 \pm 0.004}$
RLadder w/o classification task		1.552 ± 0.025
RLadder	$\mathbf{0.74 \pm 0.09}$	$\mathbf{1.501 \pm 0.001}$

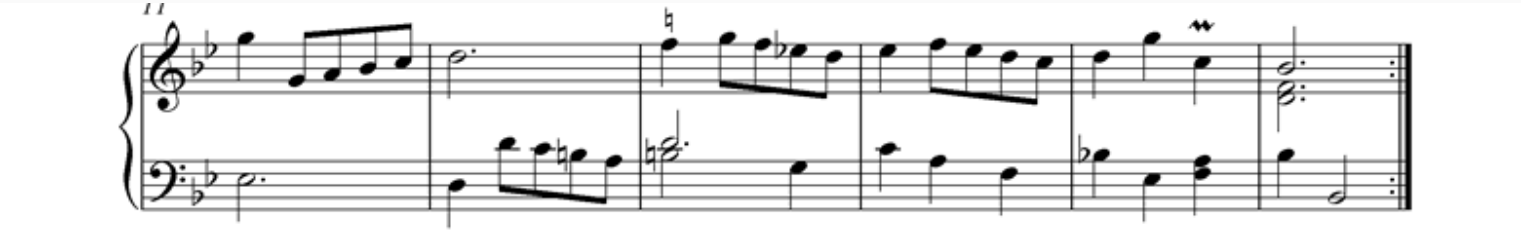
Probe of internal representations



Semi-supervised learning results

	1k labeled	1k labeled & 59k unlabeled	
		no WACT	WACT
Optimal reconstruction and static classifier	3.50 ± 0.28	3.50 ± 0.28	1.34 ± 0.04
Temporal baseline	10.86 ± 0.43	10.86 ± 0.43	3.14 ± 0.16
RLadder	10.49 ± 0.81	5.20 ± 0.77	1.69 ± 0.14

Polyphonic Music Dataset



- Piano rolls (the notes played at every time step) of various piano pieces by 19 different classical composers
- Time step is an eighth note
- Low-level task (primary): Output a distribution of notes for the next time step.
- Measure: negative log likelihood (NLL)

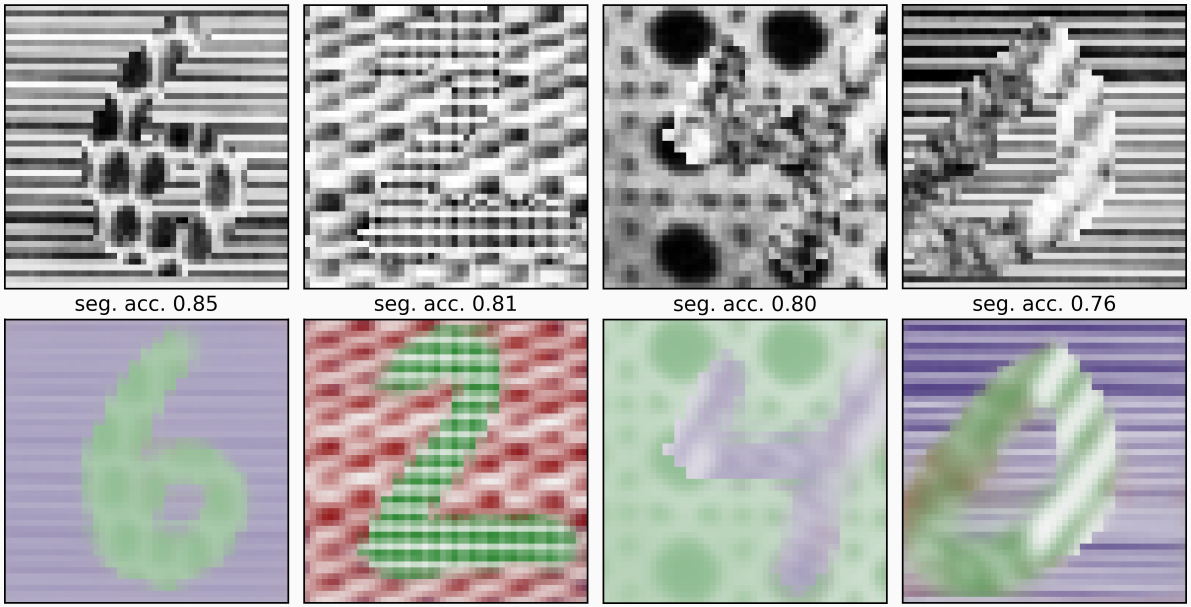
Polyphonic Music Dataset: Results

	Piano-midi.de	Nottingham	Muse	JSB Chorales
Models outputting a joint distribution of notes:				
NADE masked	7.42	3.32	6.48	8.51
NADE	7.05	2.89	5.54	7.59
RNN-RBM	7.09	2.39	6.01	6.27
RNN-NADE (HF)	7.05	2.31	5.60	**5.56**
LSTM-NADE	7.39	2.06	5.03	6.10
TP-LSTM-NADE	5.49	1.64	4.34	5.92
BALSTM	**5.00**	**1.62**	**3.90**	5.86
Models outputting marginal probabilities for each note:				
RNN	7.88	3.87	7.43	8.76
LSTM	6.866	3.492		
MUT1	6.792	3.254		
RLadder	**6.19 $\pm$ 0.02**	**2.42 $\pm$ 0.03**	**5.69 $\pm$ 0.02**	**5.64 $\pm$ 0.02**

Perceptual grouping with RLadder

Perceptual grouping with RLadder

- Process of identifying which parts of the sensory input belong to the same higher-level perceptual components (objects)

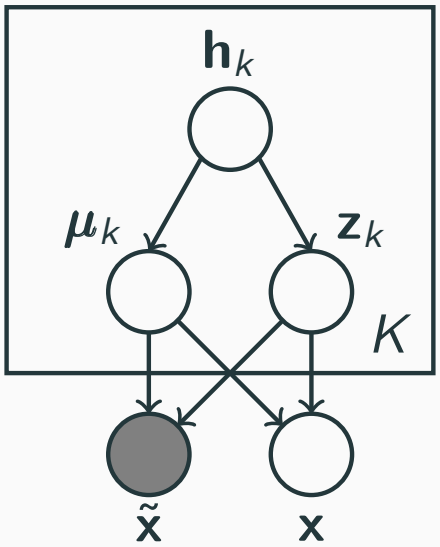


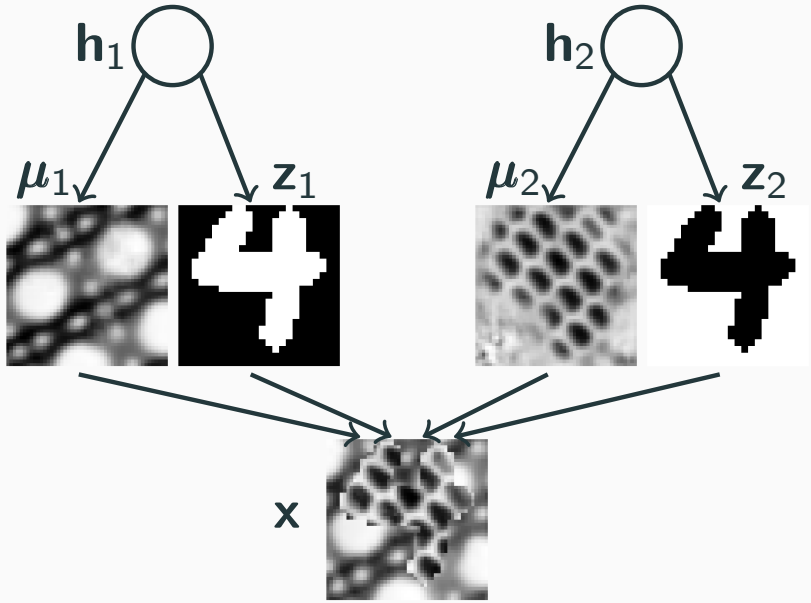
- *Greff et al. Tagger: Deep unsupervised perceptual grouping. NIPS-2016*

Perceptual grouping with RLadder

- Implicitly assumed probabilistic model:

$$p(\tilde{\mathbf{x}}, \boldsymbol{\mu}, \mathbf{z}, \mathbf{h}) = \prod_{i,k} N(\tilde{x}_i | \mu_{i,k}, \sigma_k^2 + \sigma^2)^{z_{i,k}} \prod_{k=1}^{K} p(\mathbf{z}_k, \boldsymbol{\mu}_k | \mathbf{h}_k) p(\mathbf{h}_k) \,.$$





Perceptual grouping with RLadder

- Posterior approximation:

$$p(\mathbf{z}, \boldsymbol{\mu}, \mathbf{h}|\tilde{\mathbf{x}}) \approx \prod_k q(\mathbf{z}_k, \boldsymbol{\mu}_k, \mathbf{h}_k) = \prod_k \prod_i \pi_{i,k}^{z_{i,k}} N(\mu_{i,k}|m_{i,k}, v_{i,k}) q(\mathbf{h}_k)$$

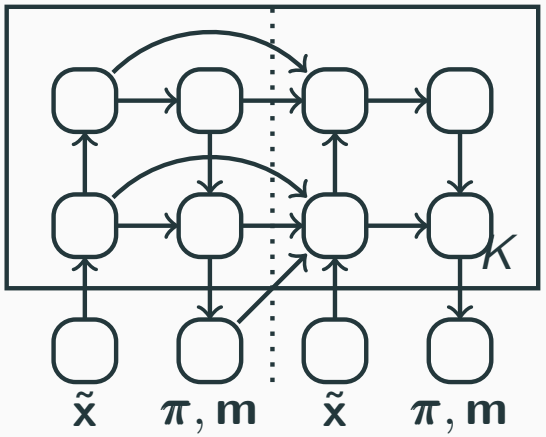
- $\pi_{i,k}$ – posterior probability that pixel i belongs to object k
- $m_{i,k}$ – expected value of object k in pixel i
- Cost function:

$$C \approx -\log p(\mathbf{x}|\tilde{\mathbf{x}}) = -\log \sum_k \pi_{i,k} N(x_i|m_{i,k}, \sigma_k^2 + v_{i,k})$$

Perceptual grouping with RLadder

- Iterative inference of each $q(\mathbf{z}_k, \boldsymbol{\mu}_k, \mathbf{h}_k)$ is done with RLadder
- Shared weights: $q(\mathbf{z}_k, \boldsymbol{\mu}_k, \mathbf{h}_k)$, $p(\mathbf{z}_k, \boldsymbol{\mu}_k, \mathbf{h}_k)$ are assumed to have the same parametric form

- We update each $q(\mathbf{z}_k, \boldsymbol{\mu}_k, \mathbf{h}_k)$ independently multiple times
- Inputs:
  - $\pi_{i,k}$, $m_{i,k}$ of all groups
  - cost function C
  - some functions of $\tilde{\mathbf{x}}$

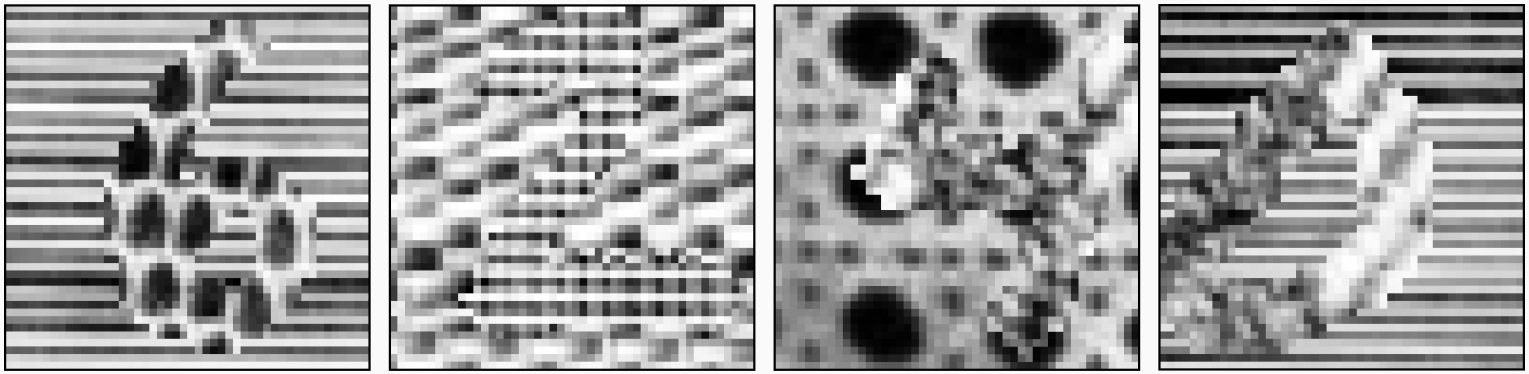


Recurrent Tagger (RTagger)

Experiments with perceptual grouping

Textured MNIST classification

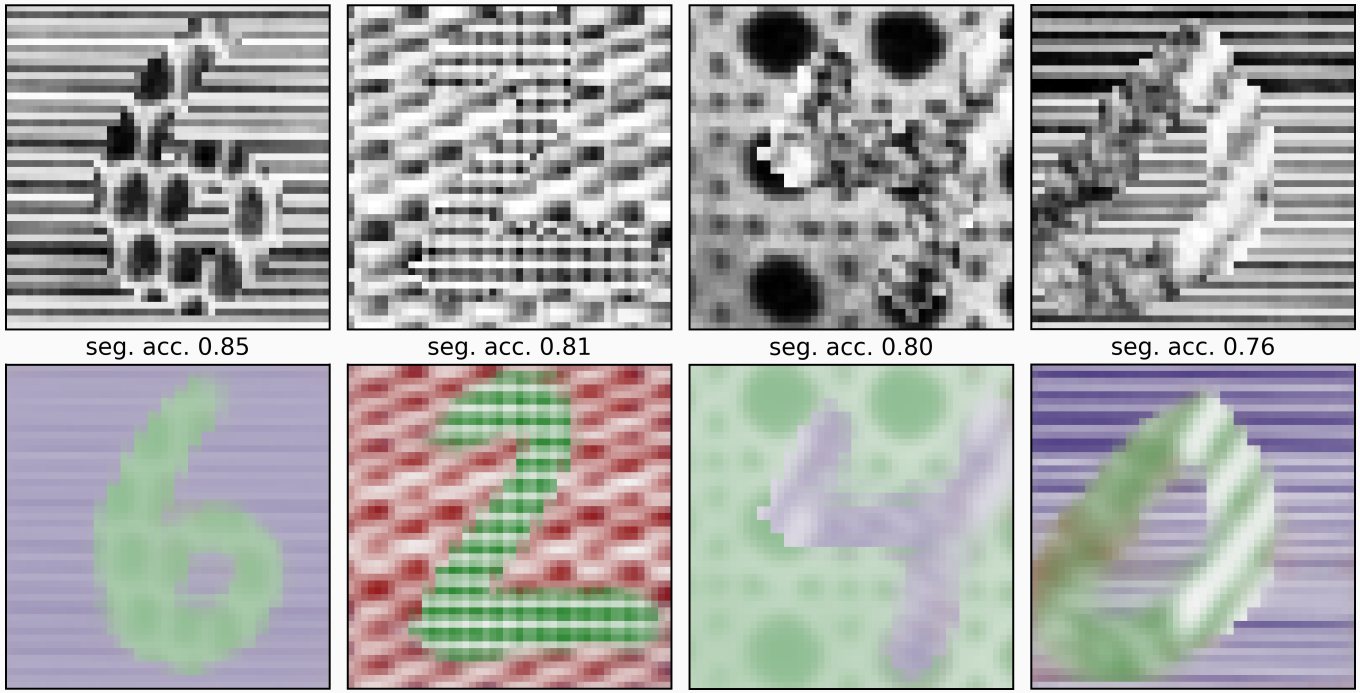
- Textured MNIST digit on textured background
- Top-level task (primary): Digit classification
- Bottom-level task (auxiliary): Denoising with a mixture model (RTagger)



Textured MNIST classification: Results

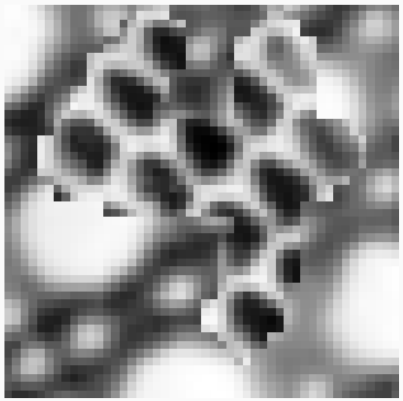
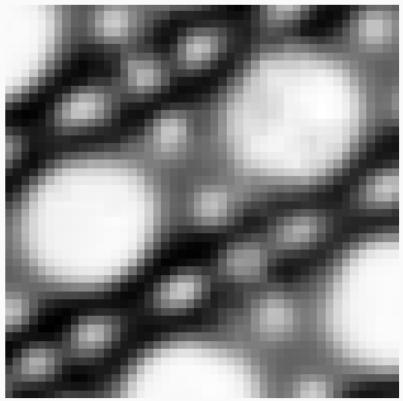
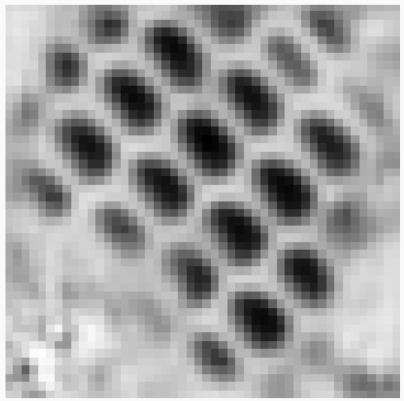
Segmentation accuracy:			
RTagger	0.55	0.75	$\mathbf{0.80 \pm 0.01}$
Tagger	0.31	0.45	0.51 ± 0.25
Classification error, %:			
RTagger	18.2	8.0	$\mathbf{5.9 \pm 0.2}$
Tagger	26.5	17.9	17.13 ± 8.9
ConvNet	–	–	14.3 ± 0.46

Textured MNIST: Segmentation results



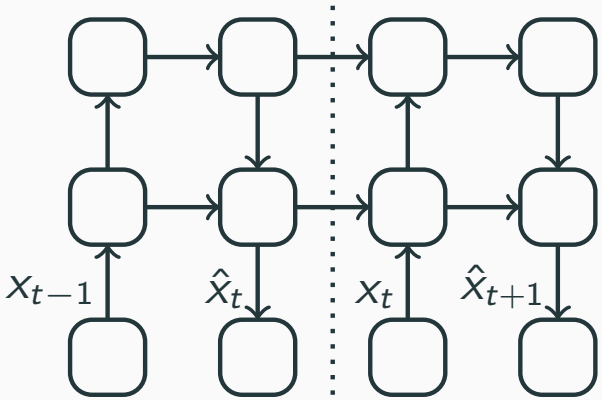
Textured MNIST: Segmentation results

- Filling invisible parts of objects:

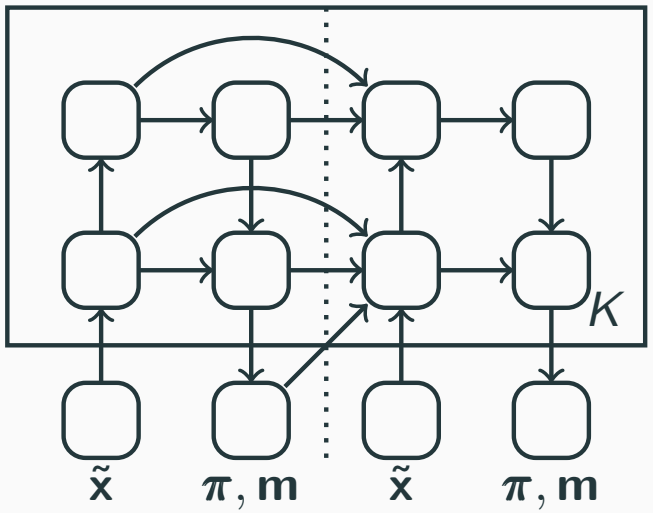




Summary

- Two proposed architectures:



RLadder (iterative inference & temporal modeling)



RTagger (perceptual grouping)

- Close-to-optimal results on temporal modeling of video data, competitive results on music modeling, and improved perceptual grouping based on higher order abstractions, such as stochastic textures