Language Technologies Institute

Carnegie Mellon University

Multimodal Machine Learning

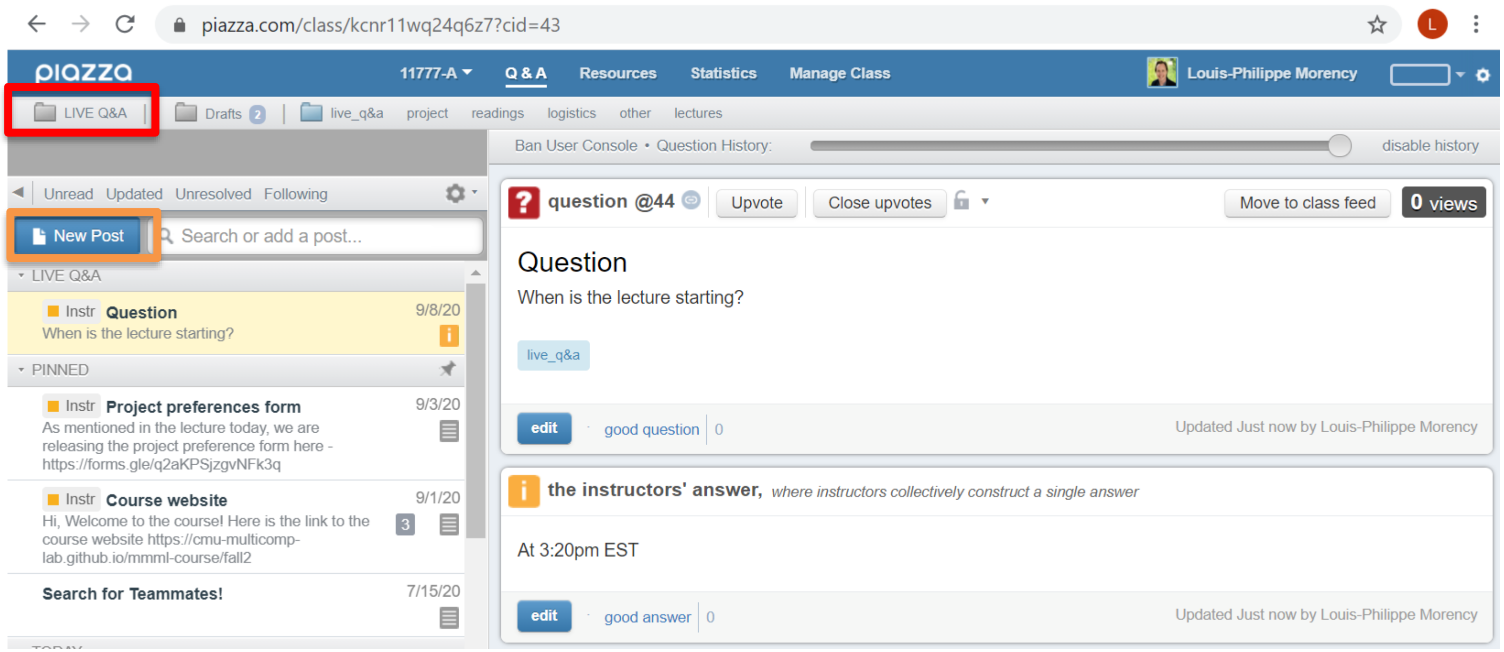
Lecture 8.1: Discriminative Graphical Models

Louis-Philippe Morency

* Original version co-developed with Tadas Baltrusaitis

Administrative Stuff

Piazza Live Q&A



Please share your questions and comments on Piazza Live Q&A

- Live responses by your TAs and follow-up by the instructor after the main lecture

Language Technologies Institute

Carnegie Mellon University

Multimodal Machine Learning

Lecture 8.1: Discriminative Graphical Models

Louis-Philippe Morency

* Original version co-developed with Tadas Baltrusaitis

Lecture Objectives

- Markov Random Fields
  - Boltzmann/Gibbs distribution
  - Factor graphs
- Conditional Random Fields
  - Multi-View Conditional Random Fields
- CRFs and Deep Learning
  - DeepConditional Neural Fields
  - CRF and Bilinear LSTM
- Continuous and Fully-Connected CRFs

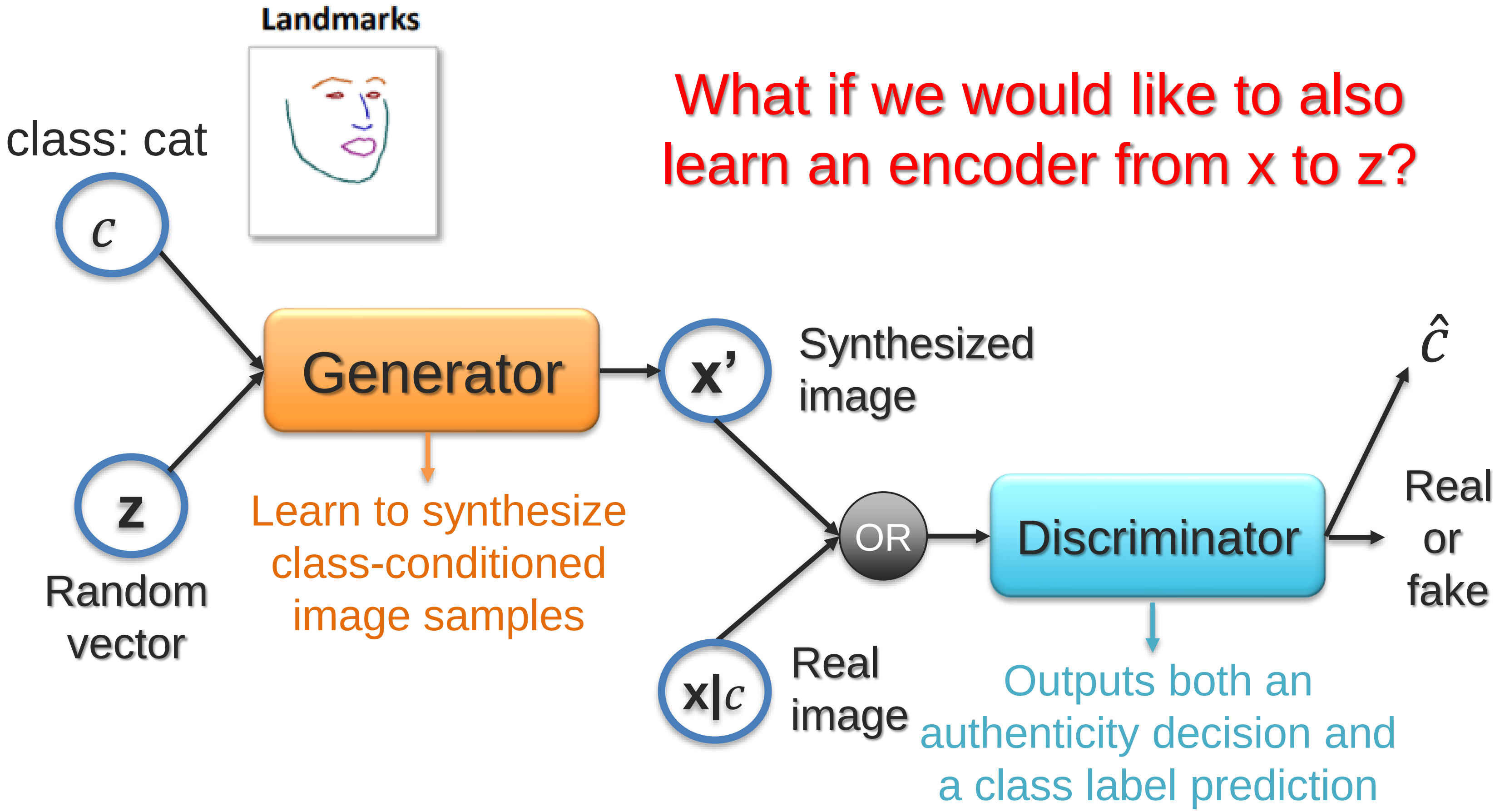
Bidirectional and Cycle GAN

Example: Talking Head

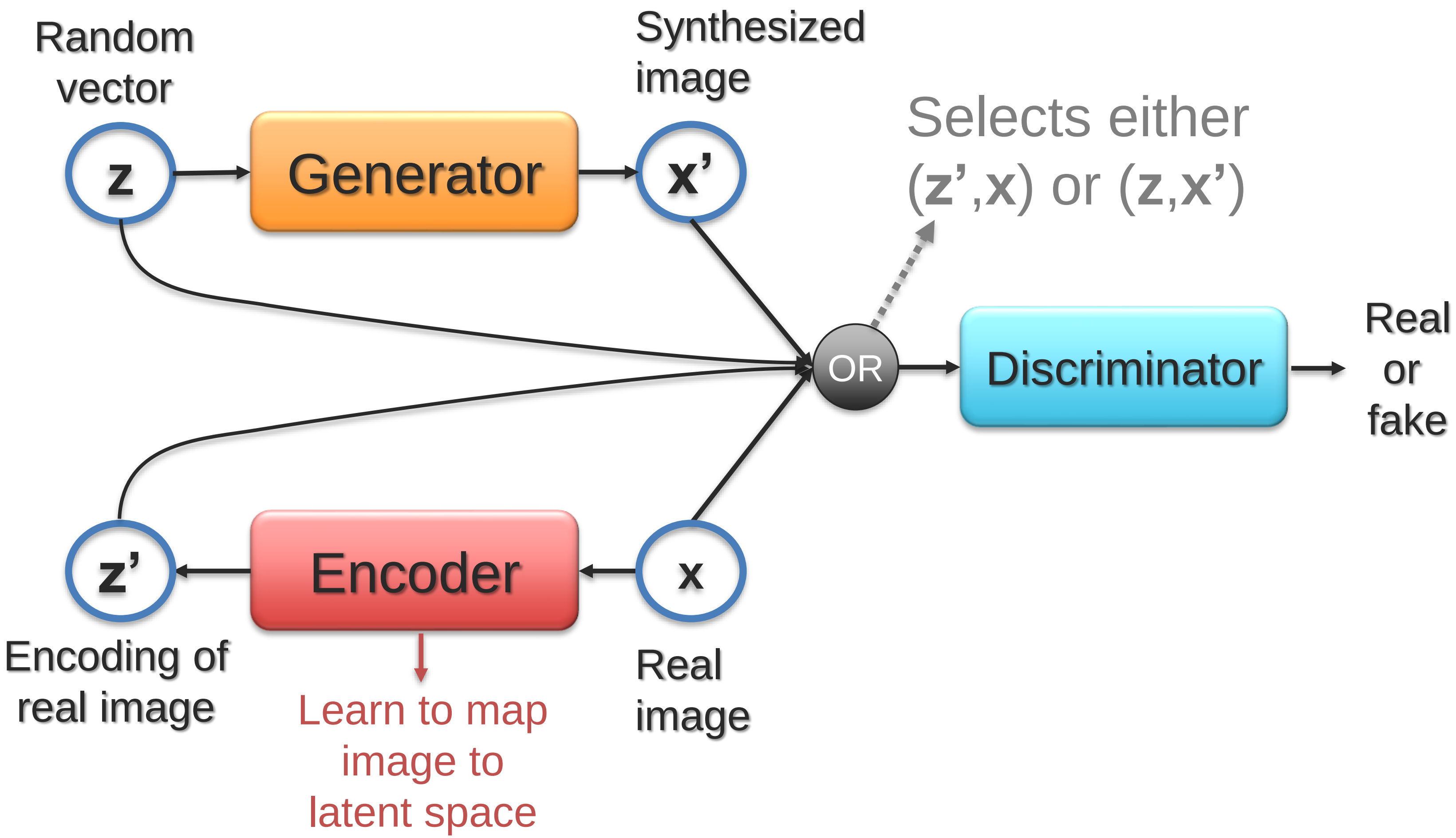
https://arxiv.org/abs/1905.08233

Info GAN

What if we would like to also learn an encoder from x to z?

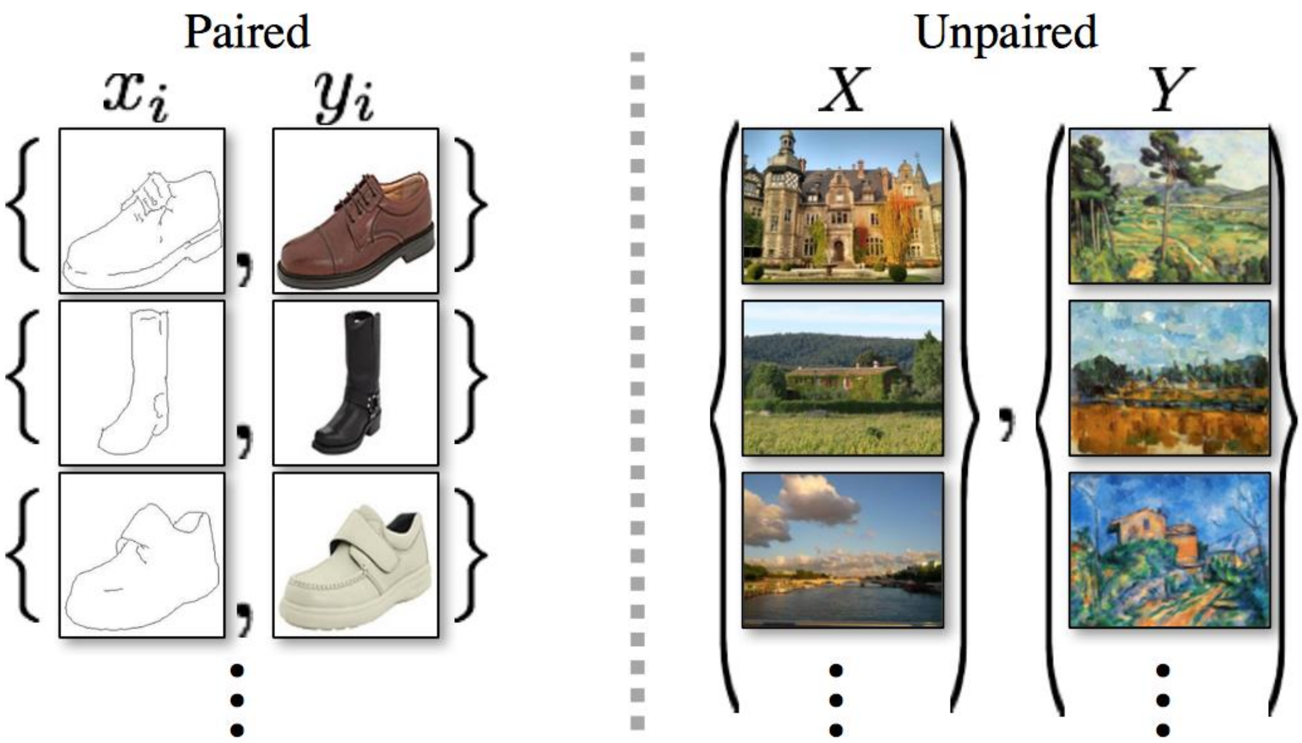


Bidirectional GAN



Paired and Unpaired Data

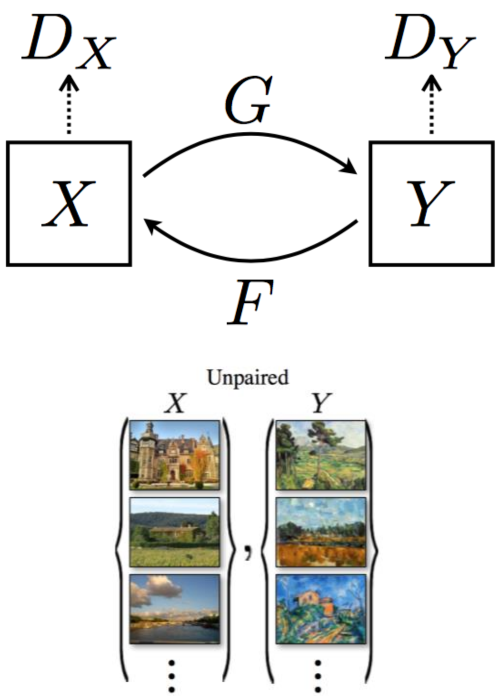
Many of these approaches use paired data…

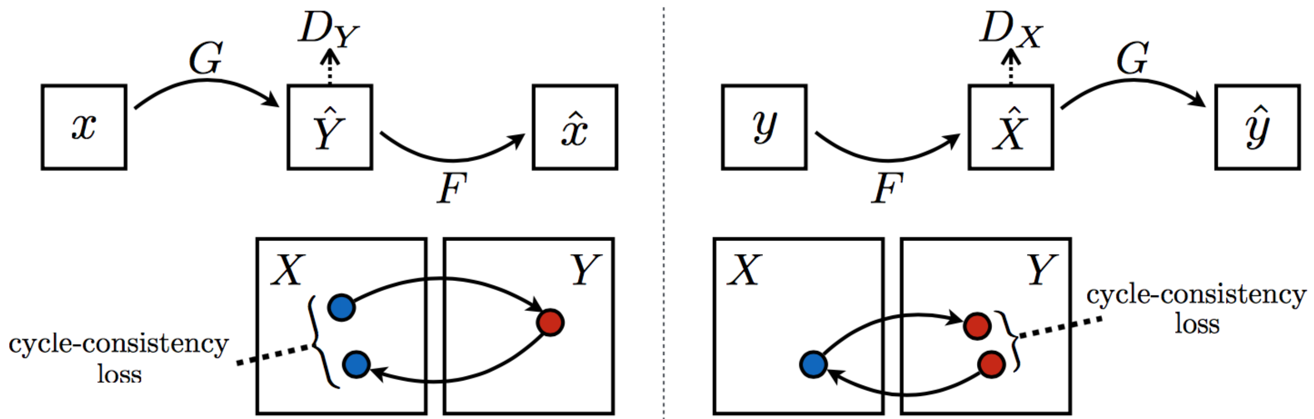


… but how to handle unpaired data?

Cycle GAN

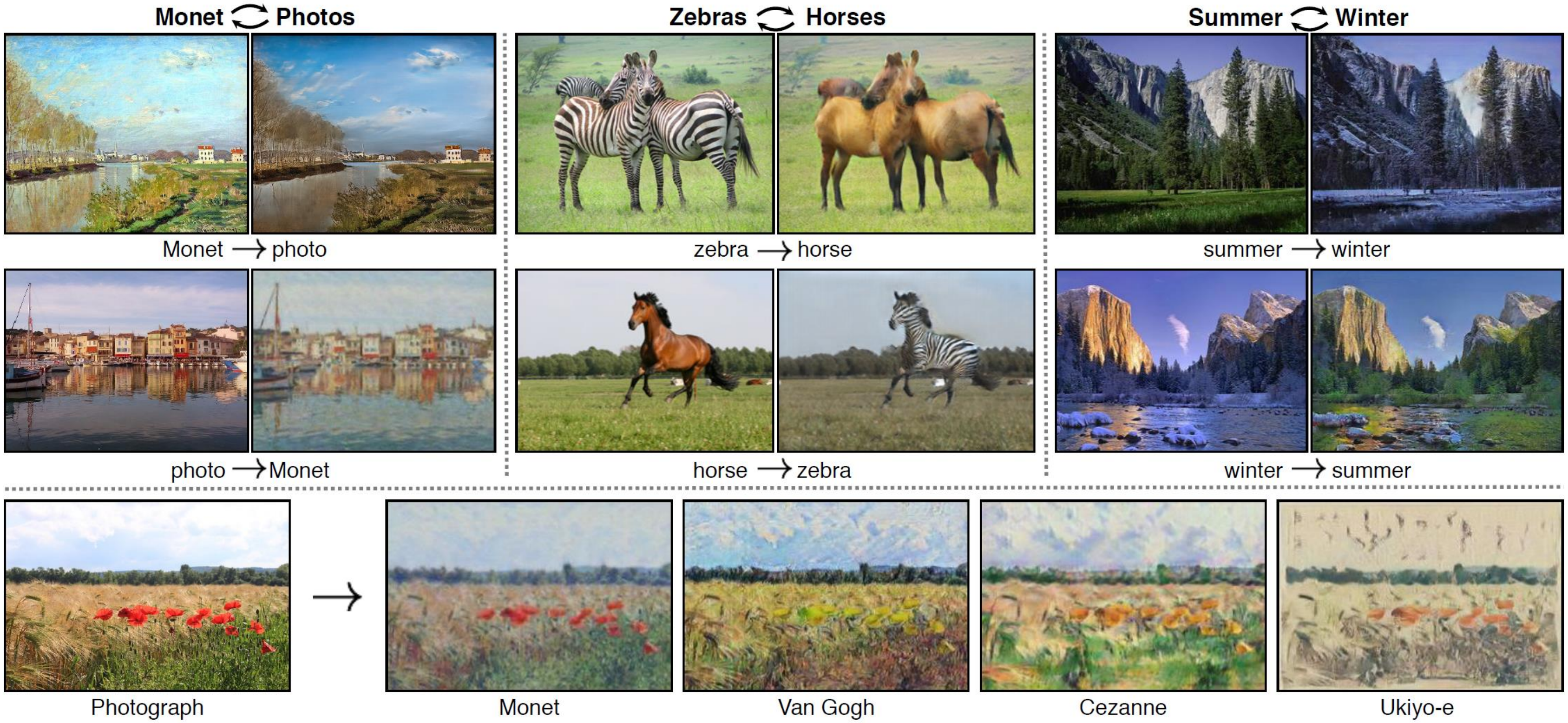
Idea 1: Let's have multiple discriminators and generators





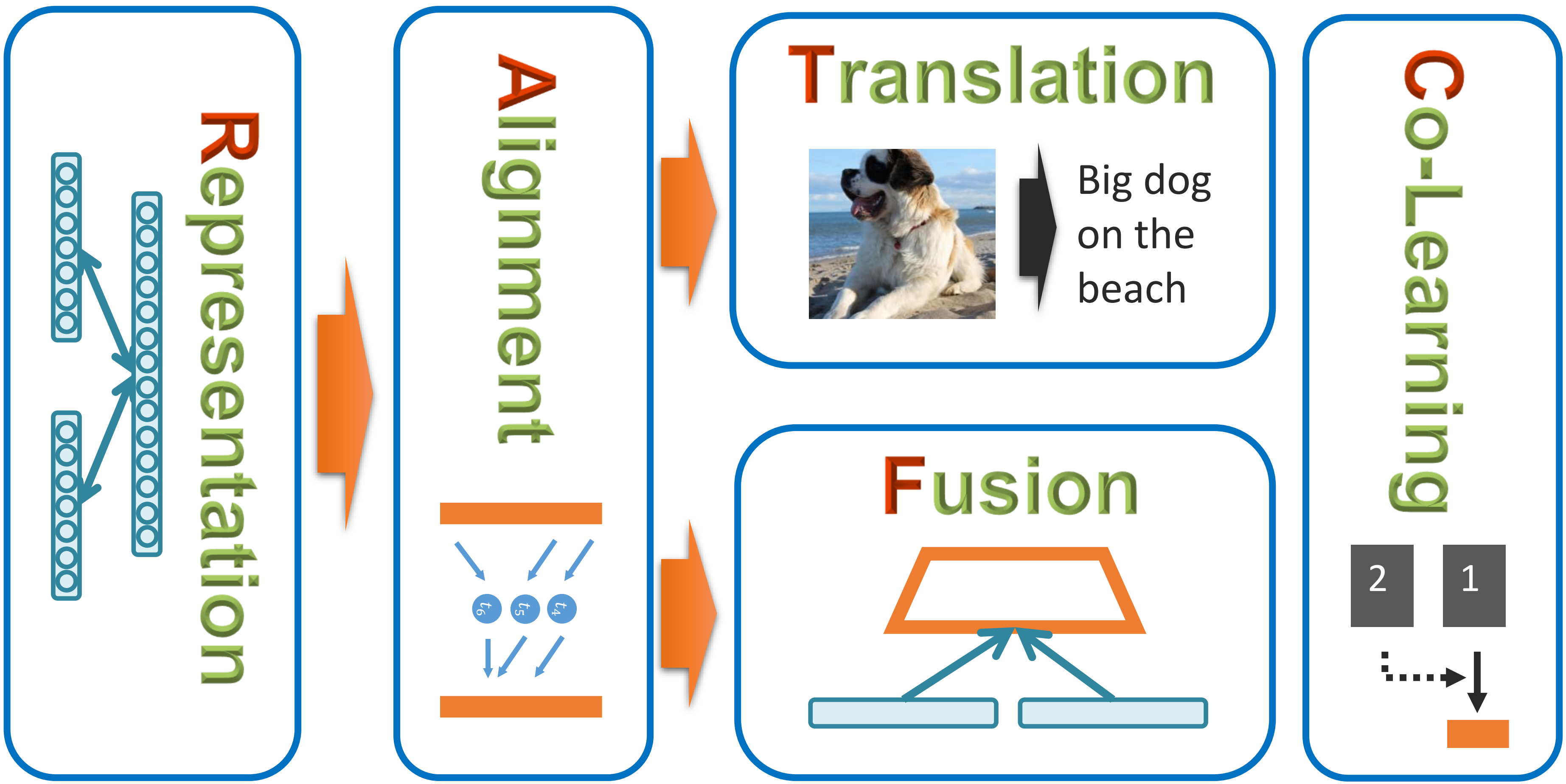
Idea 2: Use two cycle-consistency losses, one for each view

Cycle GAN


Monet ⇄ Photos
Zebras ⇄ Horses
Summer ⇄ Winter
Monet → photo
zebra → horse
summer → winter
photo → Monet
horse → zebra
winter → summer
Photograph
Monet
Van Gogh
Cezanne
Ukiyo-e

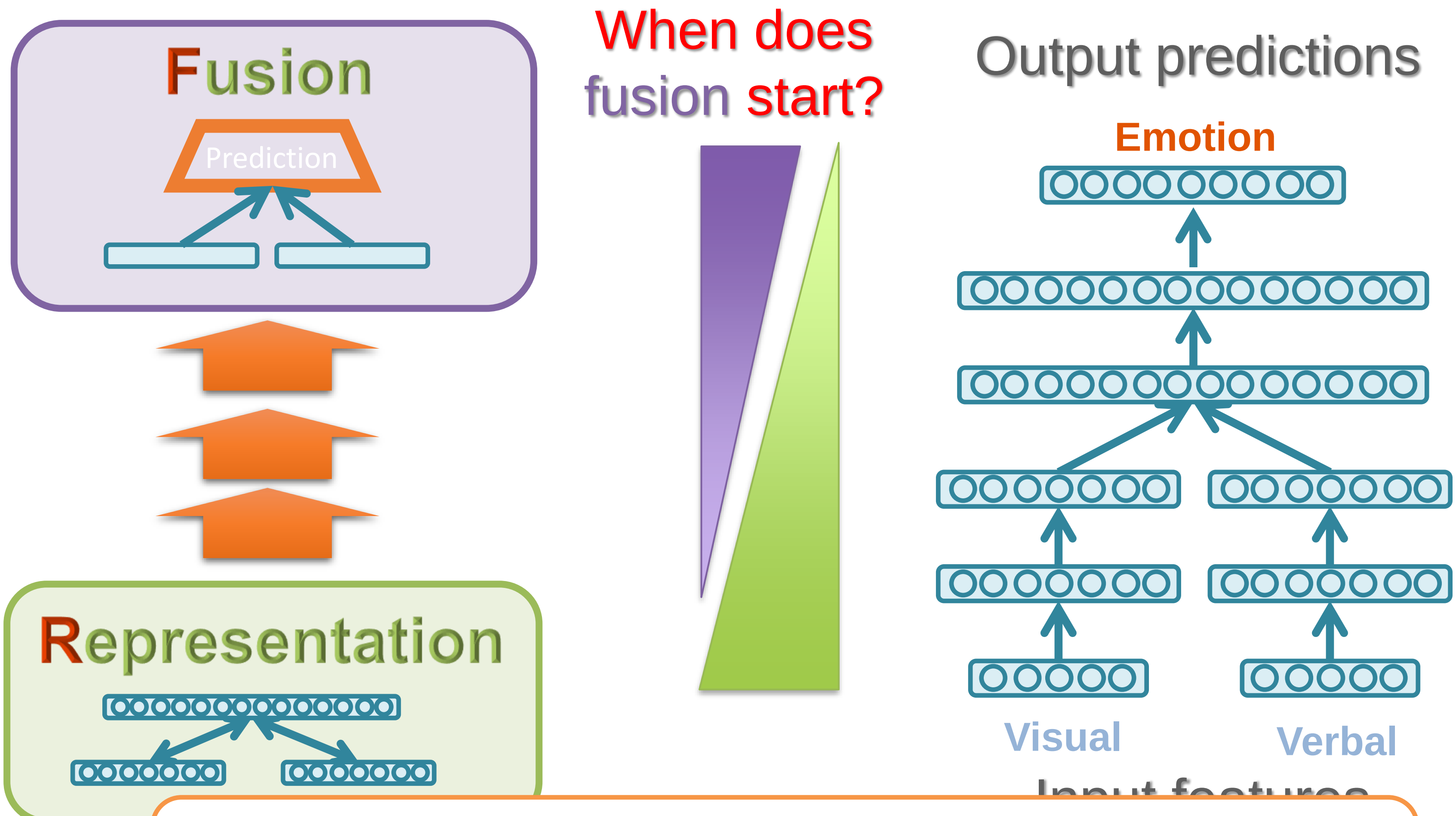
Quick Recap

Five Multimodal Core Challenges



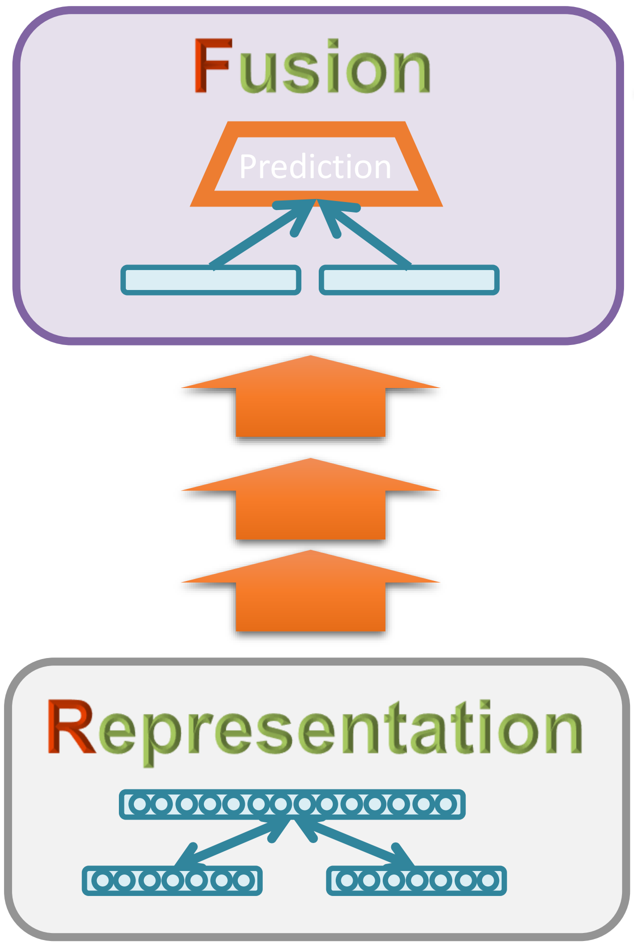
Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency, Multimodal Machine Learning: A Survey and Taxonomy

Fusion and Representation – Neural Networks



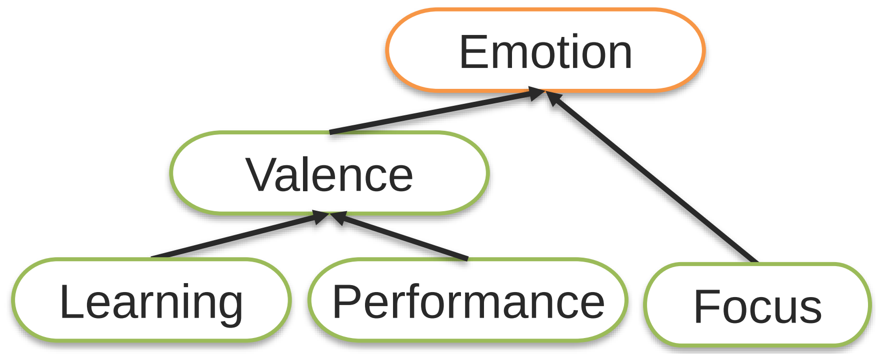
For many neural models, it is ambiguous when representation learning ends..

Fusion – Probabilistic Graphical Models

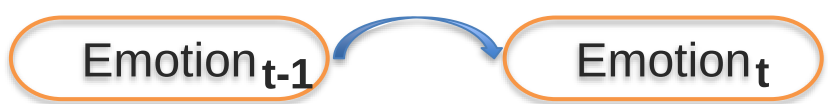


Domain knowledge

a) Latent sub-structure

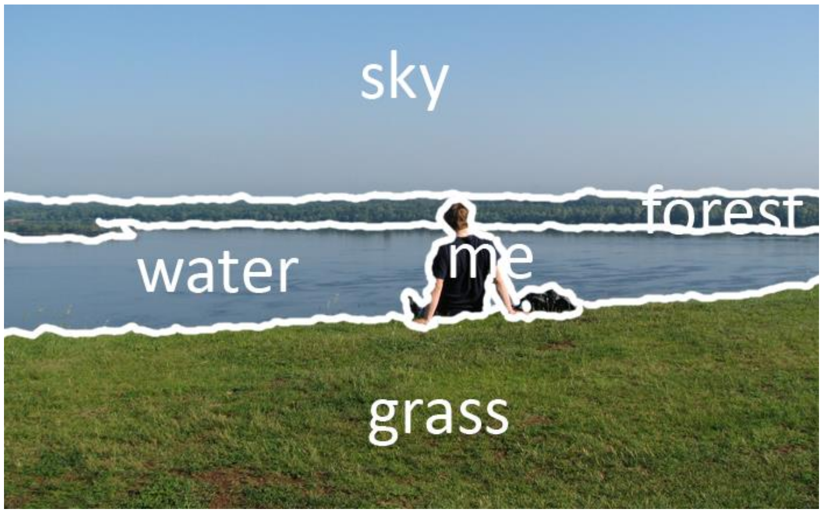


b) Structured output prediction



Structured Prediction - Examples

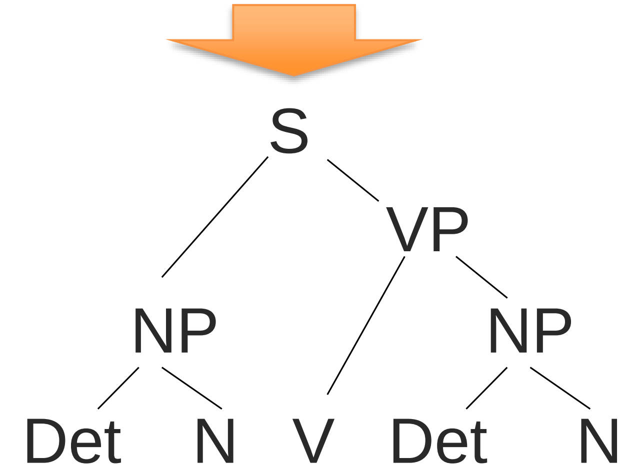
Image “semantic” segmentation



We do not want to predict each pixel separately!

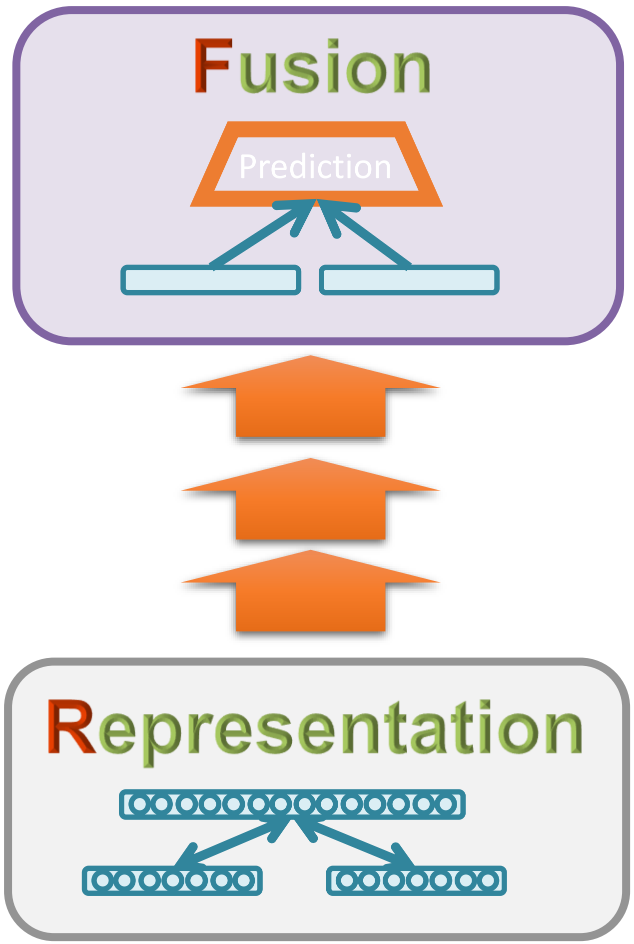
Dependency parsing

The boy saw the dog

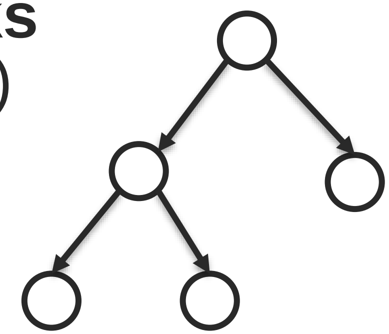


The output is a structured tree

Fusion – Probabilistic Graphical Models

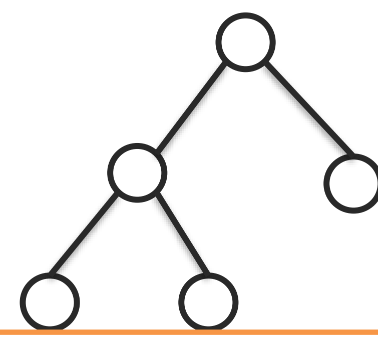


Bayesian networks
(last lecture)



But they can be challenging to optimize jointly with neural networks…

Markov Models
(this lecture)



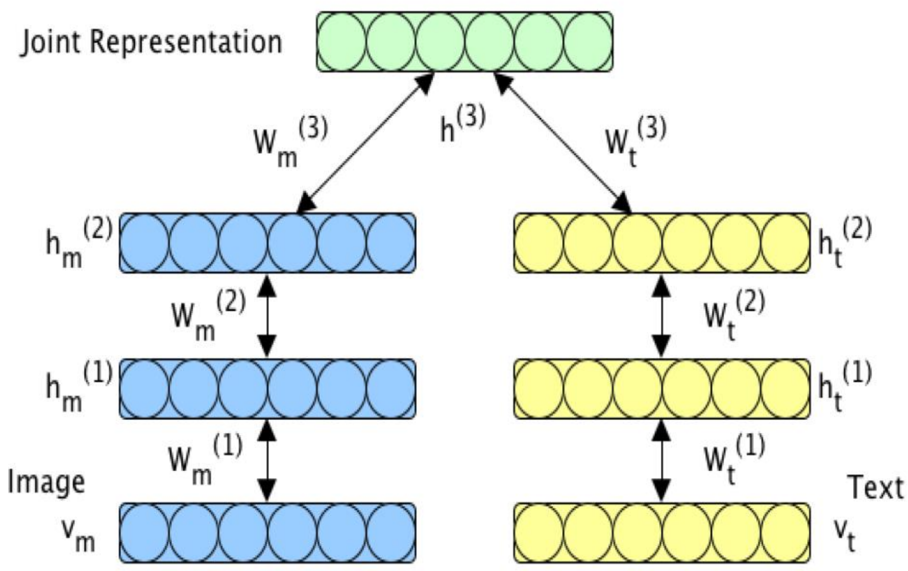
But let's first do a historical detour…

Restricted Boltzmann Machines

Deep Multimodal Boltzmann machines

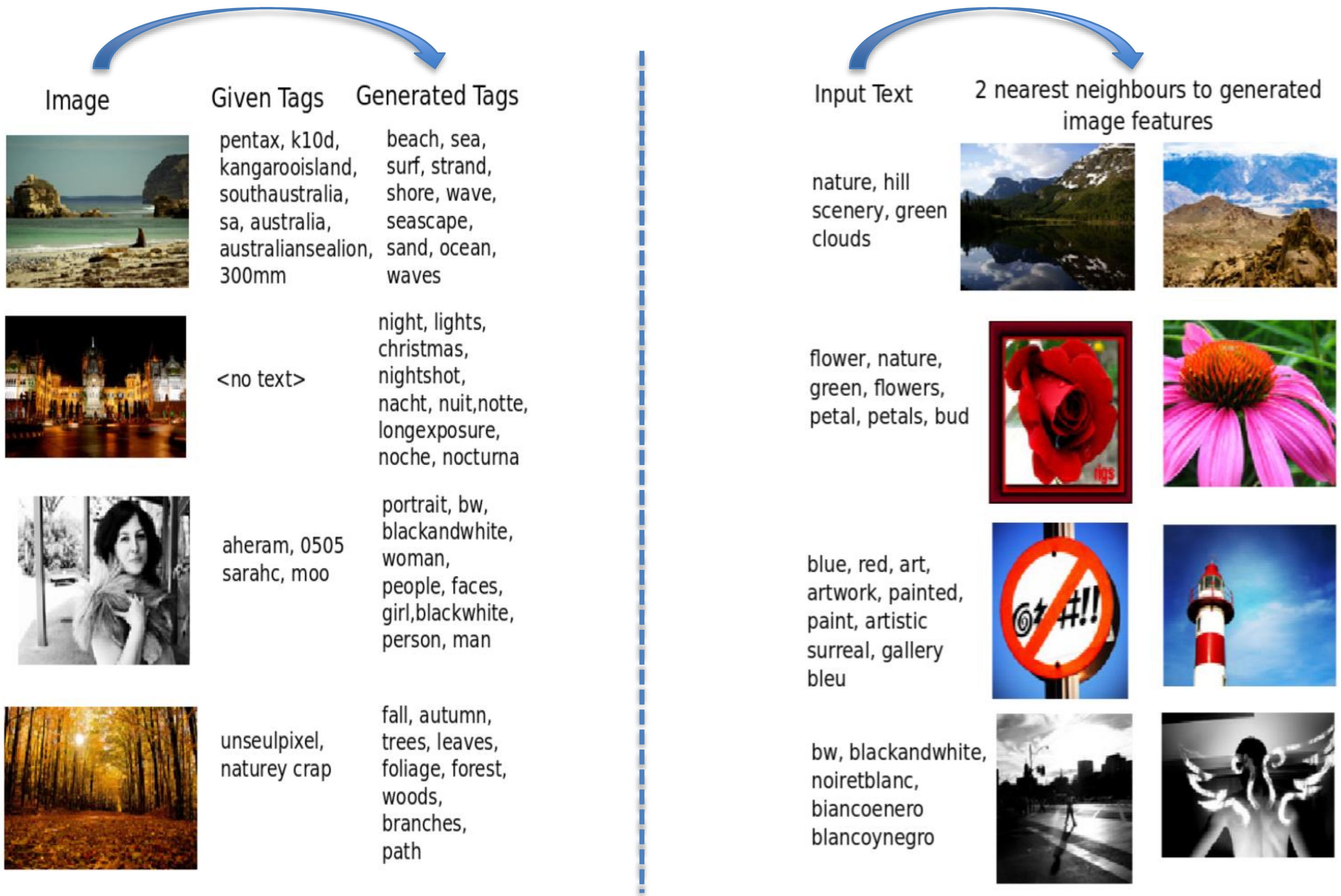
One of the first multimodal representation learning paper was using Boltzmann machines!

- Generative model: models the joint probability between modalities
- It can sample both text and image modalities



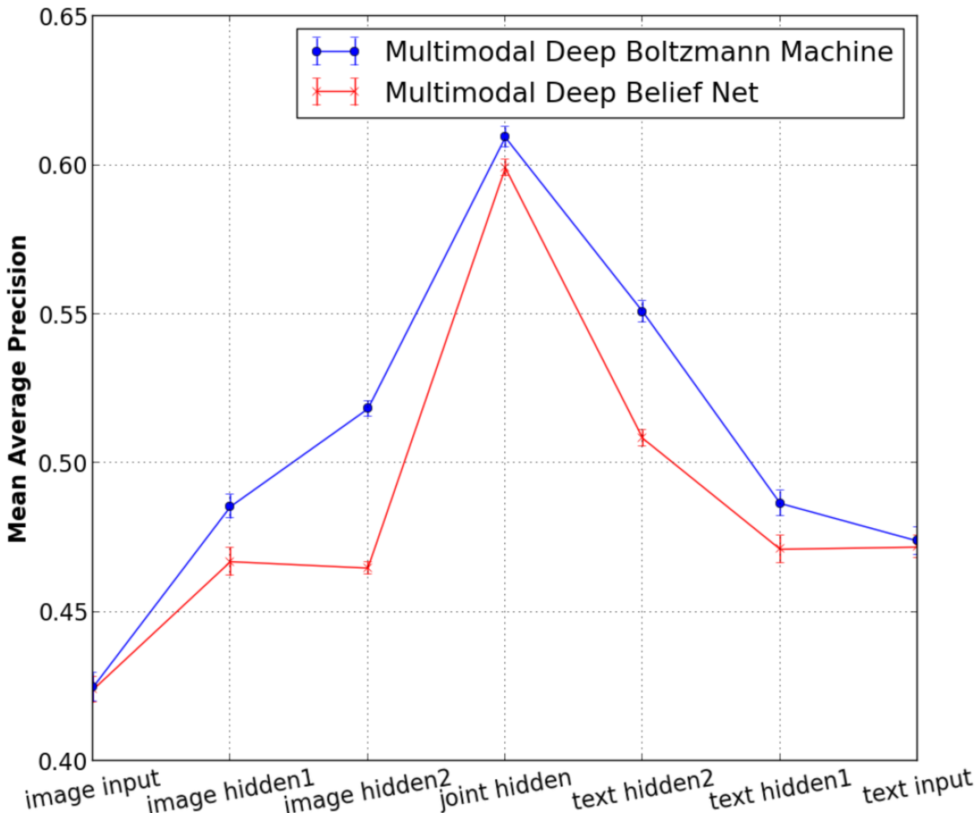
[Srivastava and Salakhutdinov, Multimodal Learning with Deep Boltzmann Machines, 2012, 2014]

Deep Multimodal Boltzmann Machines



Deep Multimodal Boltzmann Machines

Performance on image retrieval task



Paired text is only used during training!

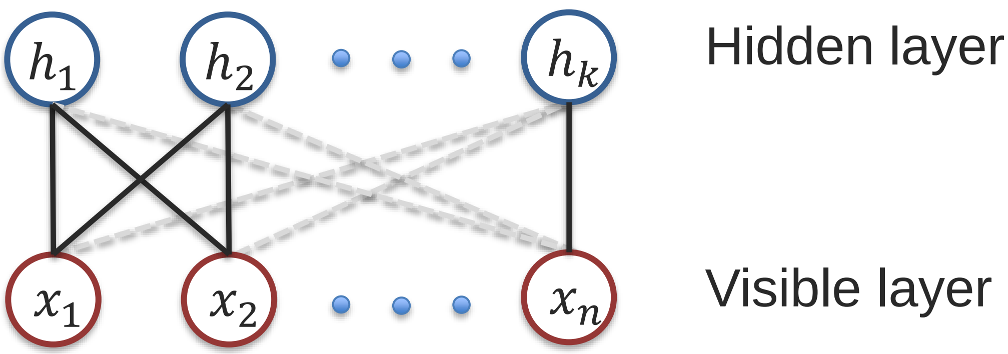
But what are Boltzmann machines?

Let's start with the "restricted" version…

Restricted Boltzmann Machine (RBM)

Undirected Graphical Model

- A generative rather than discriminative model
- Connections from every hidden unit to every visible one
- No connections across units (hence “Restricted”), makes it easier to train and run inference



[Smolensky, Information Processing in Dynamical Systems: Foundations of Harmony Theory, 1986]

Restricted Boltzmann Machine (RBM)

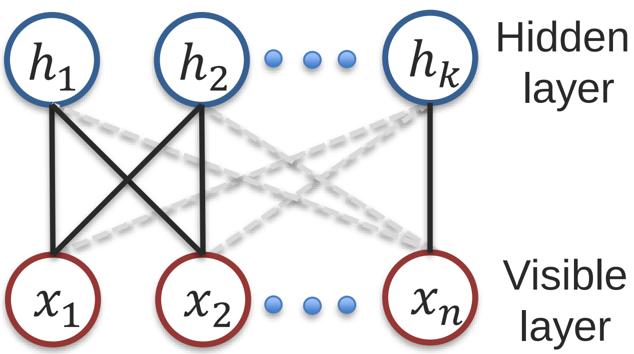
$$p(\boldsymbol{x},\boldsymbol{h};\theta) = \frac{\exp(-\mathrm{E}(\boldsymbol{x},\boldsymbol{h};\theta))}{\underbrace{\sum_{\boldsymbol{x}'}\sum_{\boldsymbol{h}'}\exp(-\mathrm{E}(\boldsymbol{x}',\boldsymbol{h}';\theta))}_{\text{Partition function } Z}}$$

Hidden and visible layers are binary (e.g. $\boldsymbol{x} = \{0, \dots, 1,0,1\}$)

Model parameters $\theta = \{W, \boldsymbol{b}, \boldsymbol{a}\}$

$$\mathrm{E} = -\boldsymbol{x}W\boldsymbol{h} - \boldsymbol{b}\boldsymbol{x} - \boldsymbol{a}\boldsymbol{h}$$

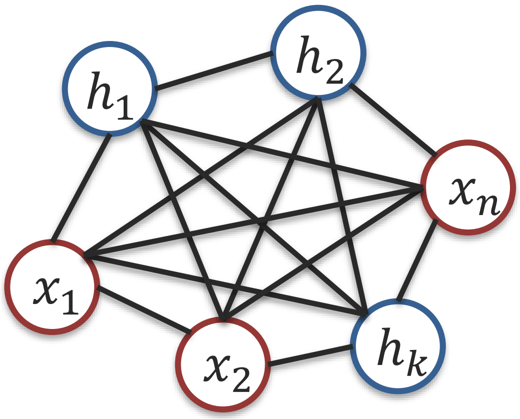
$$\mathrm{E} = -\underbrace{\sum_i\sum_j w_{i,j}x_i h_j}_{\text{Interaction term}} \underbrace{- \sum_i b_i x_i - \sum_j a_j h_j}_{\text{Bias terms}}$$

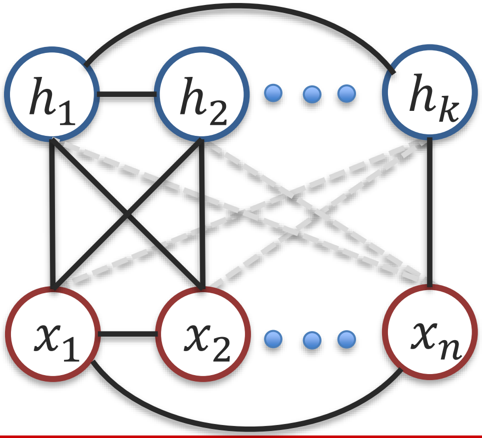


Boltzmann Machine

$$p(\boldsymbol{x}, \boldsymbol{h}; \theta) = \frac{\exp(-\mathrm{E}(\boldsymbol{x}, \boldsymbol{h}; \theta))}{\sum_{\boldsymbol{x}'} \sum_{\boldsymbol{h}'} \exp(-\mathrm{E}(\boldsymbol{x}', \boldsymbol{h}'; \theta))}$$

Hidden and visible layers are binary (e.g. $\boldsymbol{x} = \{0, \dots, 1,0,1\}$)



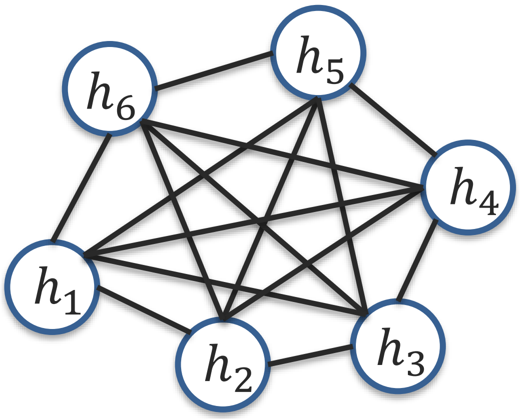


Statistical Mechanics: Boltzmann Distribution

[also called Gibbs measure]

$$p(\boldsymbol{h};\theta) = \frac{\exp(-\mathrm{E}(\boldsymbol{h};\theta)/kT)}{\sum_{\boldsymbol{h}'} \exp(-\mathrm{E}(\boldsymbol{h}';\theta)/kT)}$$

- probability distribution that gives the probability that a system will be in a certain state $\boldsymbol{h}$



$\mathrm{E}(\boldsymbol{h};\theta)$: Energy of state $\boldsymbol{h}$

k: Boltzmann constant

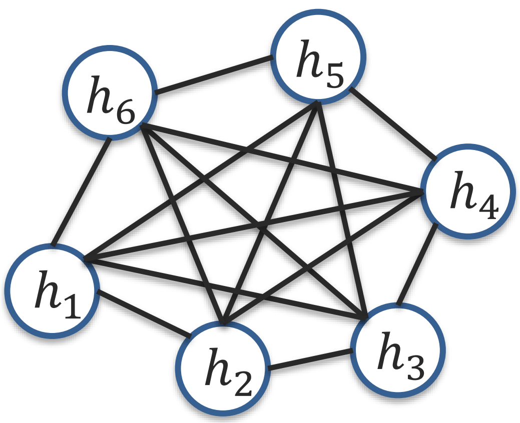
T: Thermodynamic temperature

Markov Random Fields

Markov Random Fields

$$p(H = \boldsymbol{h}; \theta) = \frac{\exp(-\mathrm{E}(\boldsymbol{h}; \theta))}{\sum_{\boldsymbol{h}'} \exp(-\mathrm{E}(\boldsymbol{h}'; \theta))} = \frac{\Phi(\boldsymbol{h}; \theta)}{\sum_{\boldsymbol{h}'} \Phi(\boldsymbol{h}'; \theta)}$$

- Set of random variables H having a Markov property described by undirected graph



$$\Phi(\boldsymbol{h}; \theta) = \prod_k \phi_k(\boldsymbol{h}; \theta_k)$$

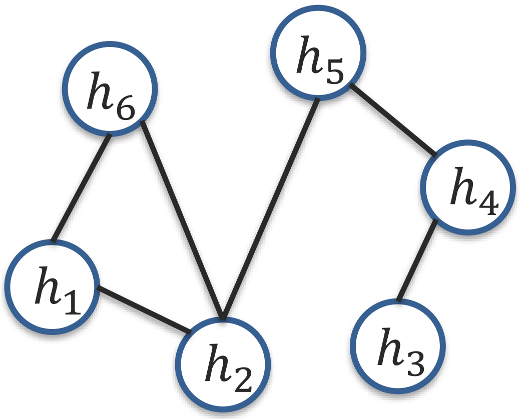
$$= \exp\left(-\sum_k \mathrm{E_k}(\boldsymbol{h}; \theta_k)\right)$$

Potential functions
$\phi_k(\boldsymbol{h}; \theta) > 0$

Markov Random Fields

$$p(H = \boldsymbol{h}; \theta) = \frac{\Phi(\boldsymbol{h}; \theta)}{\sum_{\boldsymbol{h}'} \Phi(\boldsymbol{h}'; \theta)} = \frac{\sum_k \phi_k(\boldsymbol{y}, \boldsymbol{x}; \theta)}{\sum_{\boldsymbol{y}'} \sum_k \phi_k(\boldsymbol{y}', \boldsymbol{x}; \theta)}$$

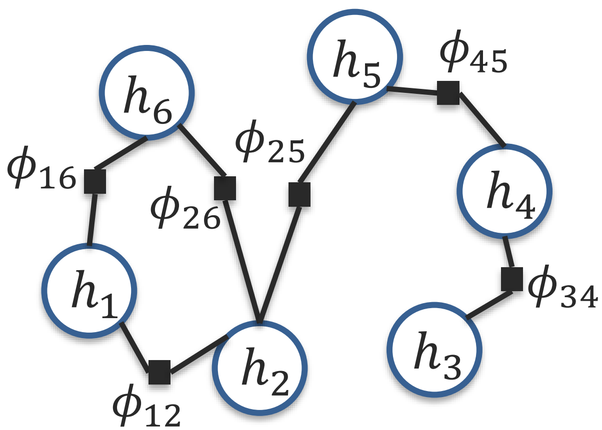
$$\begin{aligned}\Phi(\boldsymbol{h}; \theta) = \; & \phi_{12}(h_1, h_2; \theta_{12}) \times \\ & \phi_{16}(h_1, h_6; \theta_{16}) \times \\ & \phi_{26}(h_2, h_6; \theta_{26}) \times \\ & \phi_{25}(h_2, h_5; \theta_{25}) \times \\ & \phi_{45}(h_4, h_5; \theta_{45}) \times \\ & \phi_{34}(h_3, h_4; \theta_{34})\end{aligned}$$



Markov Random Fields: Factor Graphs

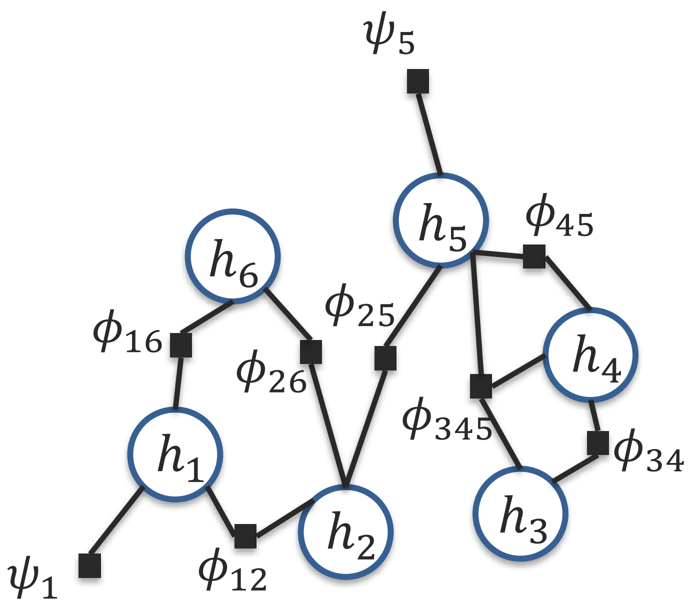
$$p(H = \boldsymbol{h}; \theta) = \frac{\Phi(\boldsymbol{h}; \theta)}{\sum_{\boldsymbol{h}'} \Phi(\boldsymbol{h}'; \theta)} = \frac{\sum_k \phi_k(\boldsymbol{y}, \boldsymbol{x}; \theta)}{\sum_{\boldsymbol{y}'} \sum_k \phi_k(\boldsymbol{y}', \boldsymbol{x}; \theta)}$$

$$\begin{aligned}\Phi(\boldsymbol{h}; \theta) = {} & \phi_{12}(h_1, h_2; \theta_{12}) \times \\ & \phi_{16}(h_1, h_6; \theta_{16}) \times \\ & \phi_{26}(h_2, h_6; \theta_{26}) \times \\ & \phi_{25}(h_2, h_5; \theta_{25}) \times \\ & \phi_{45}(h_4, h_5; \theta_{45}) \times \\ & \phi_{34}(h_3, h_4; \theta_{34})\end{aligned}$$



Markov Random Fields (Factor Graphs)

$$p(H = \boldsymbol{h}; \theta) = \frac{\Phi(\boldsymbol{h}; \theta)}{\sum_{\boldsymbol{h}'} \Phi(\boldsymbol{h}'; \theta)} = \frac{\sum_k \phi_k(\boldsymbol{y}, \boldsymbol{x}; \theta)}{\sum_{\boldsymbol{y}'} \sum_k \phi_k(\boldsymbol{y}', \boldsymbol{x}; \theta)}$$



$$\begin{aligned}
\Phi(\boldsymbol{h}; \theta) = \; & \phi_{12}(h_1, h_2; \theta_{12}) \times \\
& \phi_{16}(h_1, h_6; \theta_{16}) \times \\
& \phi_{26}(h_2, h_6; \theta_{26}) \times \\
& \phi_{25}(h_2, h_5; \theta_{25}) \times \\
& \phi_{45}(h_4, h_5; \theta_{45}) \times \\
& \phi_{34}(h_3, h_4; \theta_{34}) \times
\end{aligned}$$

pairwise potentials

$$\psi_1(h_1; \theta_1) \times \psi_5(h_5; \theta_5)$$

Unary potentials

$$\times \; \phi_{345}\,(h_3, h_4, h_5; \theta_{345})$$

Markov Random Fields – Clique Factorization

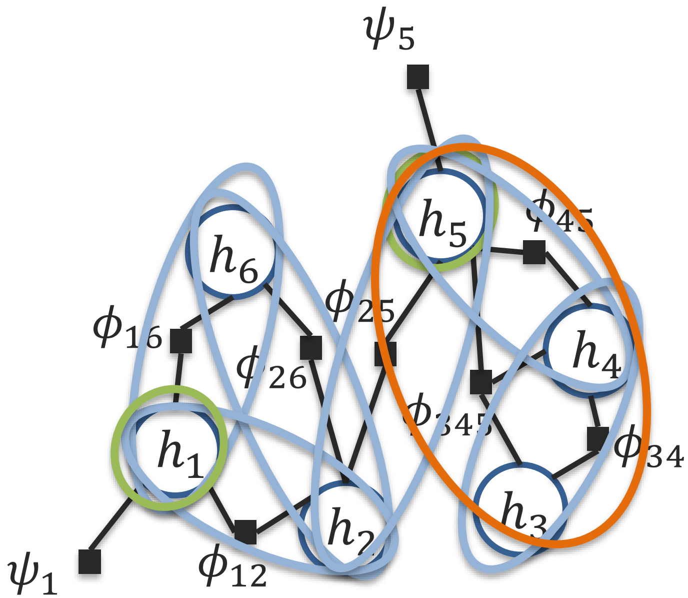
$$p(H = \boldsymbol{h}; \theta) = \frac{\Phi(\boldsymbol{h}; \theta)}{\sum_{\boldsymbol{h}'} \Phi(\boldsymbol{h}'; \theta)} = \frac{\sum_k \phi_k(\boldsymbol{y}, \boldsymbol{x}; \theta)}{\sum_{\boldsymbol{y}'} \sum_k \phi_k(\boldsymbol{y}', \boldsymbol{x}; \theta)}$$

Clique factorization

$$\Phi(\boldsymbol{h}; \theta) = \left.\begin{array}{l} \phi_{12}(h_1, h_2; \theta_{12}) \times \\ \phi_{16}(h_1, h_6; \theta_{16}) \times \\ \phi_{26}(h_2, h_6; \theta_{26}) \times \\ \phi_{25}(h_2, h_5; \theta_{25}) \times \\ \phi_{45}(h_4, h_5; \theta_{45}) \times \\ \phi_{34}(h_3, h_4; \theta_{34}) \times \end{array}\right\} \text{pairwise potentials}$$

$$\psi_1(h_1; \theta_1) \times \psi_5(h_5; \theta_5) \quad \text{Unary potentials}$$

$$\times\, \phi_{345}\,(h_3, h_4, h_5; \theta_{345})$$

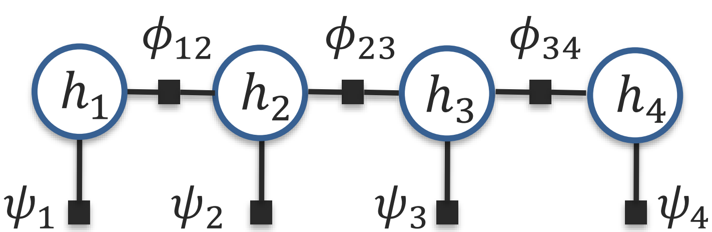


Chain Markov Random Fields (Factor Graphs)

$$p(H = \boldsymbol{h}; \theta) = \frac{\Phi(\boldsymbol{h}; \theta)}{\sum_{\boldsymbol{h}'} \Phi(\boldsymbol{h}'; \theta)} = \frac{\sum_k \phi_k(\boldsymbol{y}, \boldsymbol{x}; \theta)}{\sum_{\boldsymbol{y}'} \sum_k \phi_k(\boldsymbol{y}', \boldsymbol{x}; \theta)}$$

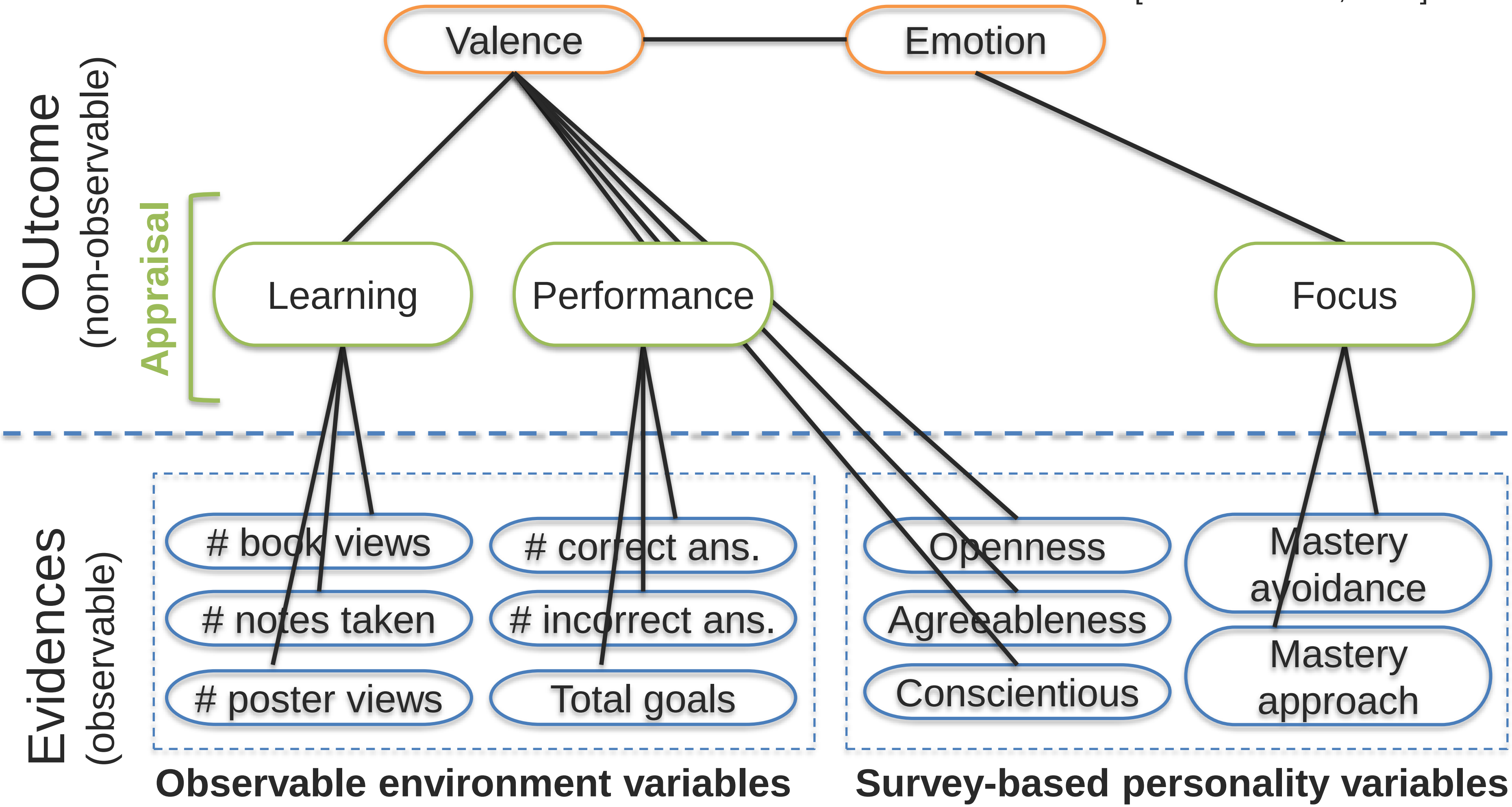
$$\Phi(\boldsymbol{h}; \theta) = \left.\begin{array}{r} \phi_{12}(h_1, h_2; \theta_{12}) \times \\ \phi_{23}(h_2, h_3; \theta_{23}) \times \\ \phi_{34}(h_3, h_4; \theta_{34}) \times \end{array}\right\} \text{pairwise potentials}$$

$$\left.\begin{array}{l} \psi_1(h_1; \theta_1) \times \\ \psi_2(h_2; \theta_2) \times \\ \psi_3(h_3; \theta_3) \times \\ \psi_4(h_4; \theta_4) \end{array}\right\} \text{Unary potentials}$$



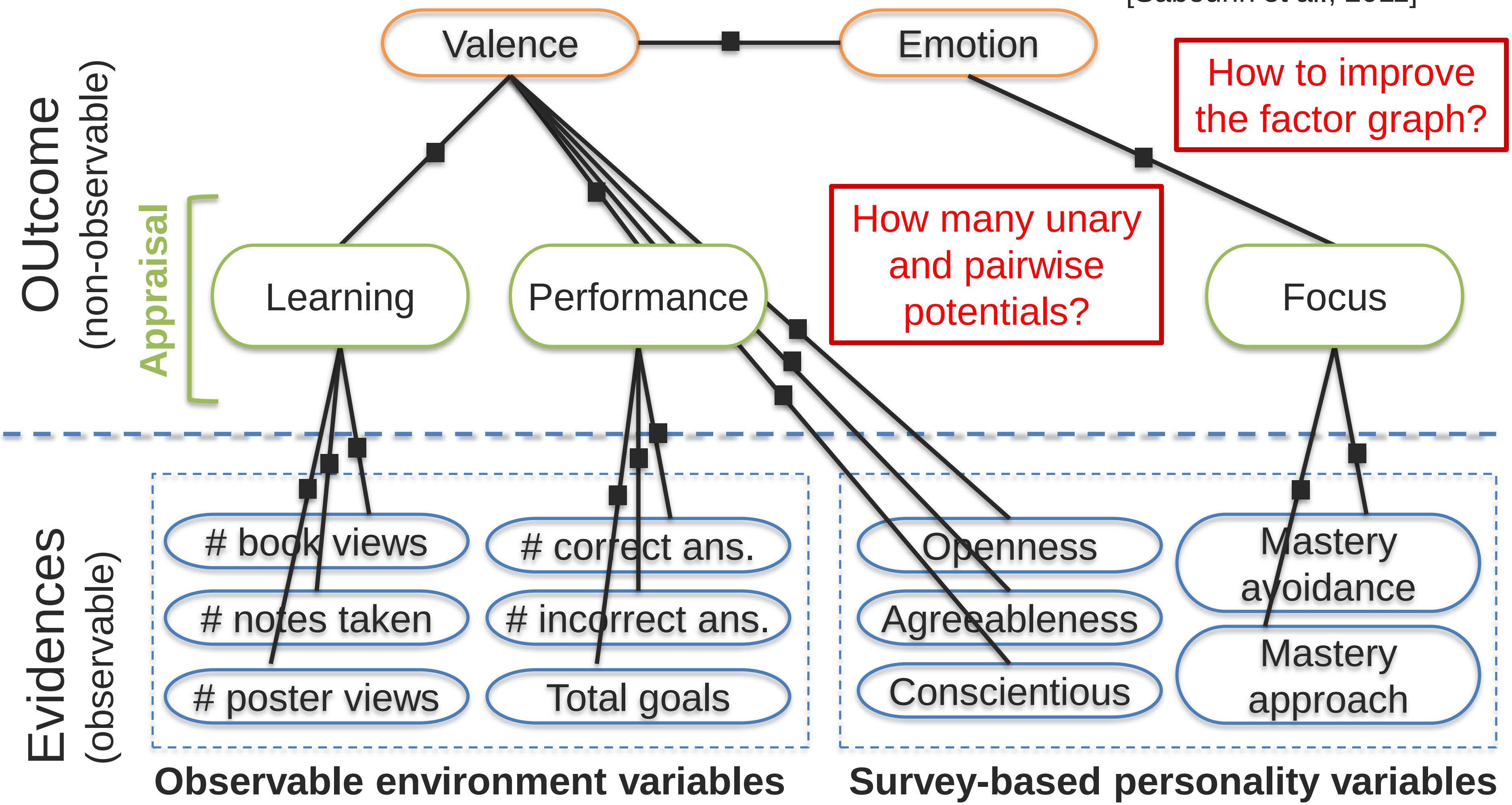
Example: Markov Random Field – Graphical Model

[Sabourin et al., 2011]



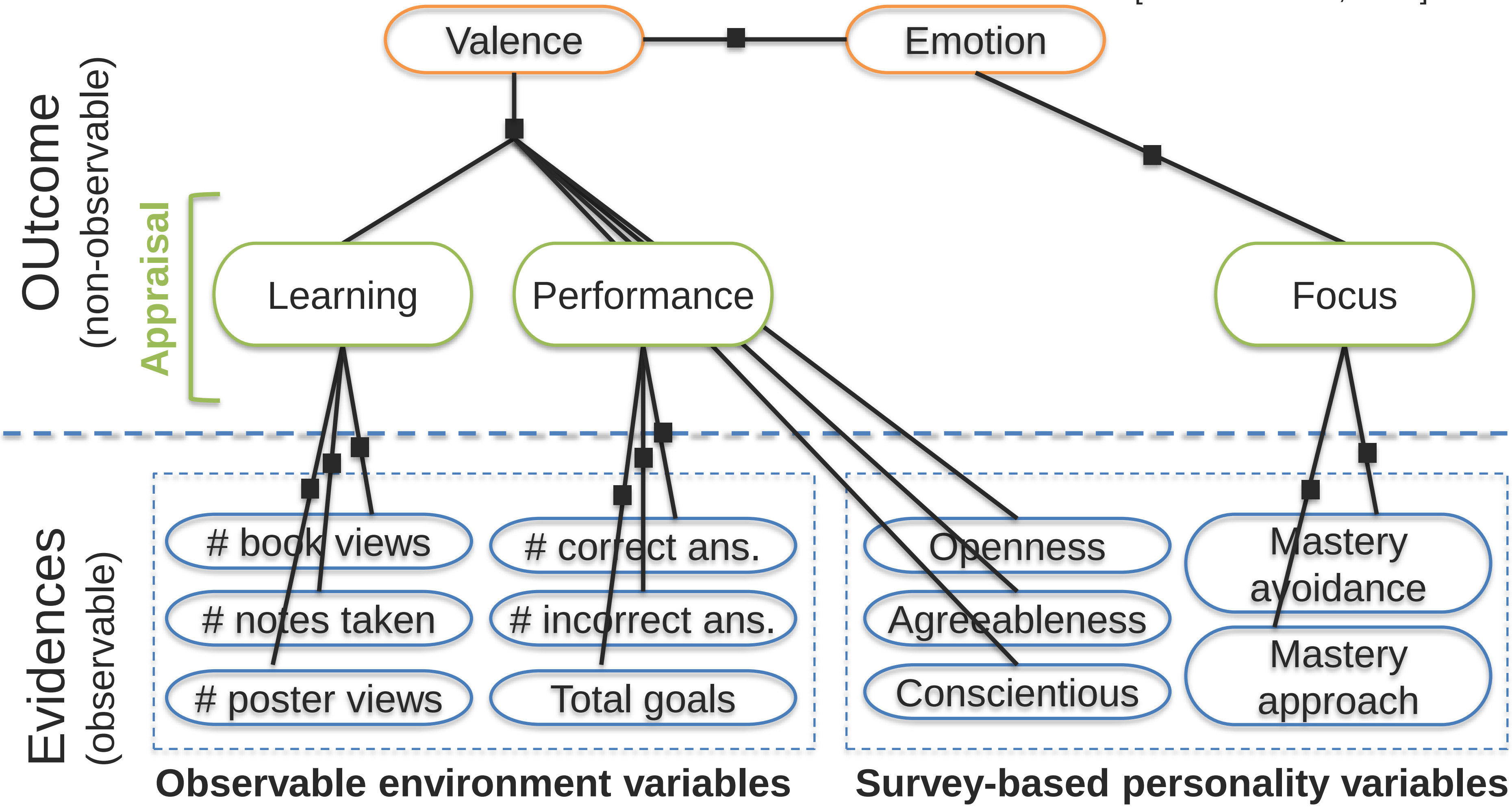
Example: Markov Random Field – Factor Graph

[Sabourin et al., 2011]



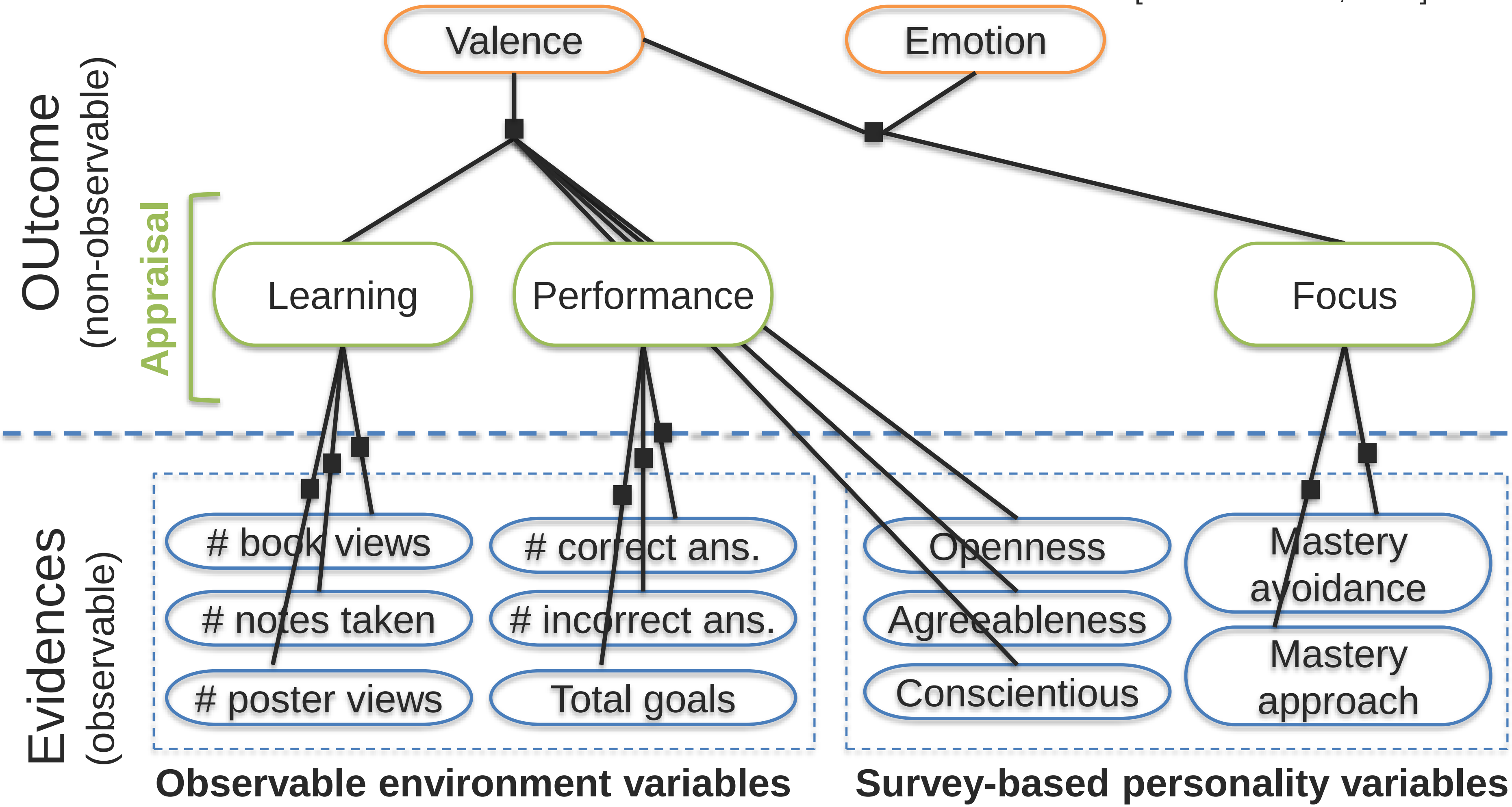
Example: Markov Random Field – Factor Graph

[Sabourin et al., 2011]



Example: Markov Random Field – Factor Graph

[Sabourin et al., 2011]



Conditional Random Fields

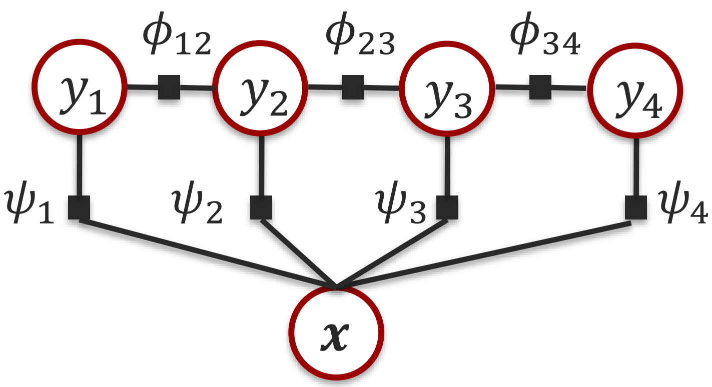
Conditional Random Fields (Factor Graphs)

$$p(\boldsymbol{y}|\boldsymbol{x};\theta) = \frac{\Phi(\boldsymbol{y},\boldsymbol{x};\theta)}{\sum_{\boldsymbol{y}'}\Phi(\boldsymbol{y}',\boldsymbol{x};\theta)} = \frac{\sum_k \phi_k(\boldsymbol{y},\boldsymbol{x};\theta)}{\sum_{\boldsymbol{y}'}\sum_k \phi_k(\boldsymbol{y}',\boldsymbol{x};\theta)}$$

$$\begin{aligned}\Phi(\boldsymbol{y},\boldsymbol{x};\theta) = {} & \phi_{12}(y_1,y_2,\boldsymbol{x};\theta_{12}) \times \\ & \phi_{23}(y_2,y_3,\boldsymbol{x};\theta_{23}) \times \\ & \phi_{34}(y_3,y_4,\boldsymbol{x};\theta_{34}) \times \\ & \psi_1(y_1,\boldsymbol{x};\theta_1) \times \\ & \psi_2(y_2,\boldsymbol{x};\theta_2) \times \\ & \psi_3(y_3,\boldsymbol{x};\theta_3) \times \\ & \psi_4(y_4,\boldsymbol{x};\theta_4)\end{aligned}$$

pairwise potentials

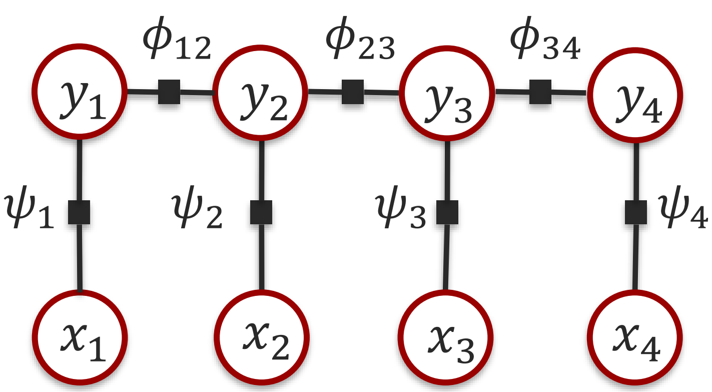
Unary potentials



Conditional Random Fields (Factor Graphs)

$$p(\boldsymbol{y}|\boldsymbol{x};\theta) = \frac{\Phi(\boldsymbol{y},\boldsymbol{x};\theta)}{\sum_{\boldsymbol{y}'}\Phi(\boldsymbol{y}',\boldsymbol{x};\theta)} = \frac{\sum_k \phi_k(\boldsymbol{y},\boldsymbol{x};\theta)}{\sum_{\boldsymbol{y}'}\sum_k \phi_k(\boldsymbol{y}',\boldsymbol{x};\theta)}$$

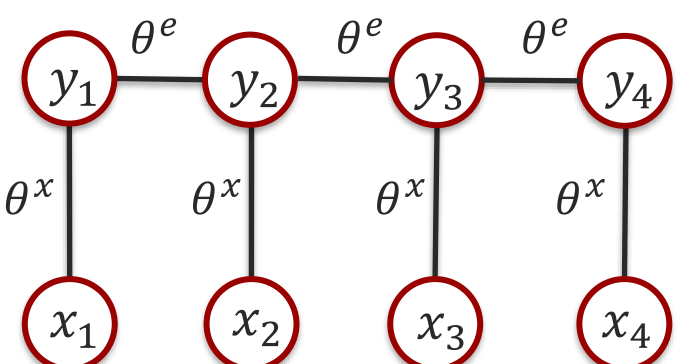
$$\begin{aligned}
\Phi(\boldsymbol{y},\boldsymbol{x};\theta) = &\left.\begin{aligned}
&\phi_{12}(y_1,y_2,\boldsymbol{x};\theta_{12}) \times \\
&\phi_{23}(y_2,y_3,\boldsymbol{x};\theta_{23}) \times \\
&\phi_{34}(y_3,y_4,\boldsymbol{x};\theta_{34}) \times
\end{aligned}\right\}\text{pairwise potentials} \\
&\left.\begin{aligned}
&\psi_1(y_1,x_1;\theta_1) \times \\
&\psi_2(y_2,x_2;\theta_2) \times \\
&\psi_3(y_3,x_3;\theta_3) \times \\
&\psi_4(y_4,x_4;\theta_4)
\end{aligned}\right\}\text{Unary potentials}
\end{aligned}$$



Conditional Random Fields (Log-linear Model)

$$p(\boldsymbol{y}|\boldsymbol{x};\theta) = \frac{\Phi(\boldsymbol{y},\boldsymbol{x};\theta)}{\sum_{\boldsymbol{y}'}\Phi(\boldsymbol{y}',\boldsymbol{x};\theta)} = \frac{\sum_k \phi_k(\boldsymbol{y},\boldsymbol{x};\theta)}{\sum_{\boldsymbol{y}'}\sum_k \phi_k(\boldsymbol{y}',\boldsymbol{x};\theta)}$$

$$= \frac{\exp(\sum_k \theta_k f_k(\boldsymbol{y},\boldsymbol{x}))}{\sum_{\boldsymbol{y}'}\exp(\sum_k \theta_k f_k(\boldsymbol{y}',\boldsymbol{x}))}$$



$f_k(\boldsymbol{y},\boldsymbol{x})$: feature function

- Pairwise feature function

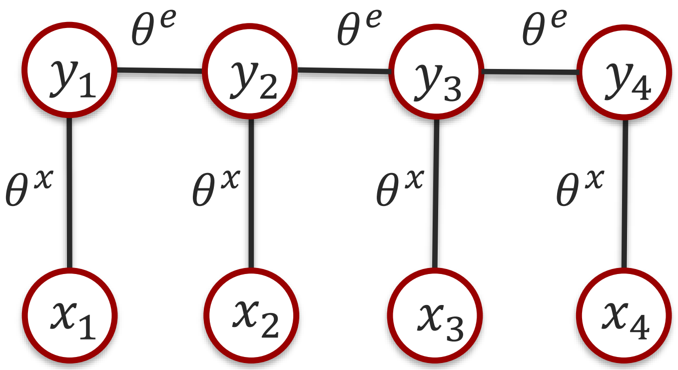
  $f_k(y_i, y_j, \boldsymbol{x}; \theta^e)$
- Unary feature function

  $f_k(y_i, \boldsymbol{x}; \theta^x)$

Learning Parameters of a CRF Model

$$\underset{\hat{y}}{\operatorname{argmax}} \log\big(p(\boldsymbol{y}|\boldsymbol{x};\theta)\big)$$

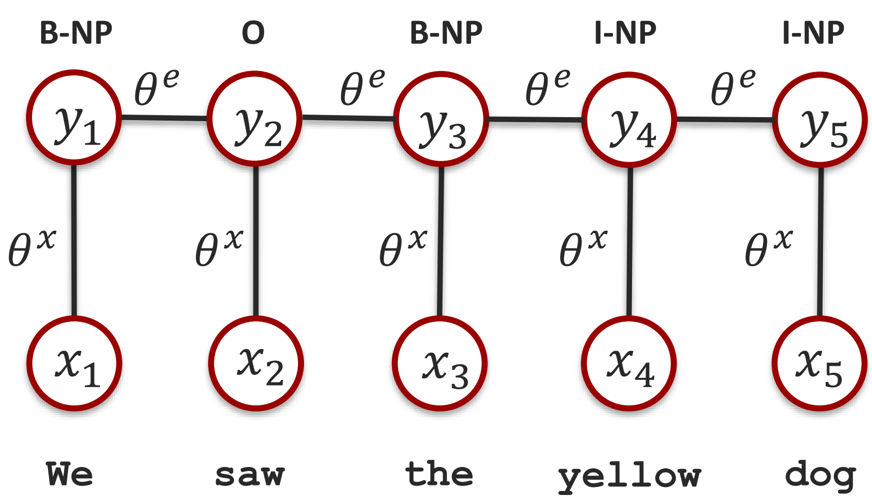
- Gradient can be computed analytically
  - Inference of marginal probabilities using belief propagation (or loopy belief propagation for cyclic graphs)
- Optimized with stochastic or batch approaches



CRFs for Shallow Parsing

$$p(\boldsymbol{y}|\boldsymbol{x};\theta) = \frac{\Phi(\boldsymbol{y},\boldsymbol{x};\theta)}{\sum_{\boldsymbol{y}'}\Phi(\boldsymbol{y}',\boldsymbol{x};\theta)} = \frac{\exp(\sum_k \theta_k f_k(\boldsymbol{y},\boldsymbol{x}))}{\sum_{\boldsymbol{y}'}\exp(\sum_k \theta_k f_k(\boldsymbol{y}',\boldsymbol{x}))}$$

- How many θ^x parameters?
- What did θ^x learn?
- What did θ^e learn?



	B-NP	I-NP	O
B-NP	θ_{11}	θ_{21}	θ_{31}
I-NP	θ_{12}	θ_{22}	θ_{32}
O	θ_{13}	θ_{23}	θ_{33}

Labels:

B-NP: Beginning of a noun phrase
I-NP: Continuation of a noun phrase
O: Outside a noun phrase

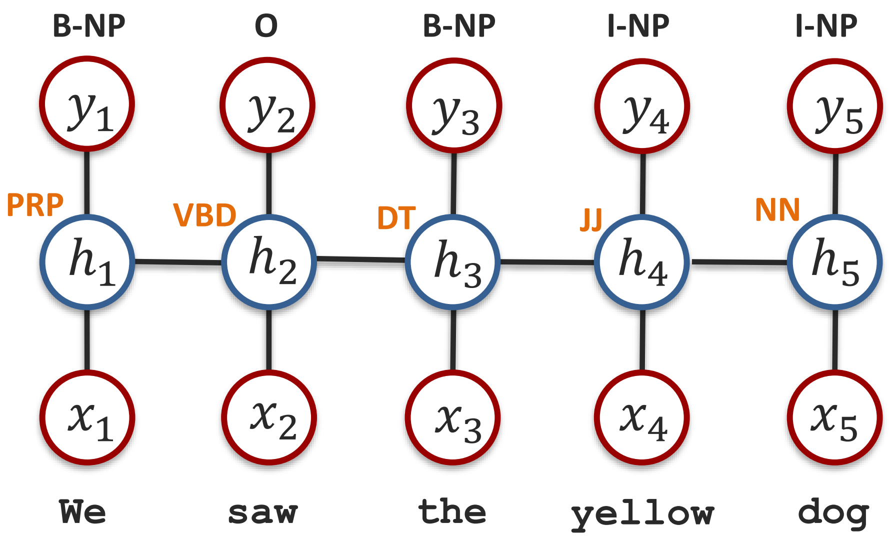
Dictionary size: 10,000 words

Latent-Dynamic CRF

$$p(\boldsymbol{y}|\boldsymbol{x};\theta) = \sum_{\boldsymbol{h}} p(\boldsymbol{y}|\boldsymbol{h};\boldsymbol{\theta})p(\boldsymbol{h}|\boldsymbol{x};\boldsymbol{\theta})$$

where $p(\boldsymbol{y}|\boldsymbol{h};\boldsymbol{\theta}) = \begin{cases} 1 & \text{if } \forall h_t \in \mathcal{H}_{y_t} \\ 0 & \text{otherwise} \end{cases}$

$$= \sum_{\boldsymbol{h}:\forall h_t \in \mathcal{H}_{y_t}} p(\boldsymbol{h}|\boldsymbol{x};\boldsymbol{\theta}) = \sum_{\boldsymbol{h}:\forall h_t \in \mathcal{H}_{y_t}} \frac{\Phi(\boldsymbol{h},\boldsymbol{x};\theta)}{\sum_{\boldsymbol{h}'} \Phi(\boldsymbol{h}',\boldsymbol{x};\theta)}$$



Latent variables (e.g., POS tags)

$\boldsymbol{h} = \{h_1, h_2, h_3, \ldots, h_t\}$ where $h_t \in \{\mathcal{H}_{y_t}\}$

For example:

$\mathcal{H} = \{\mathcal{H}_{\mathrm{B-NP}}\ \mathcal{H}_{\mathrm{I-NP}}\ \mathcal{H}_{\mathrm{O}}\}$

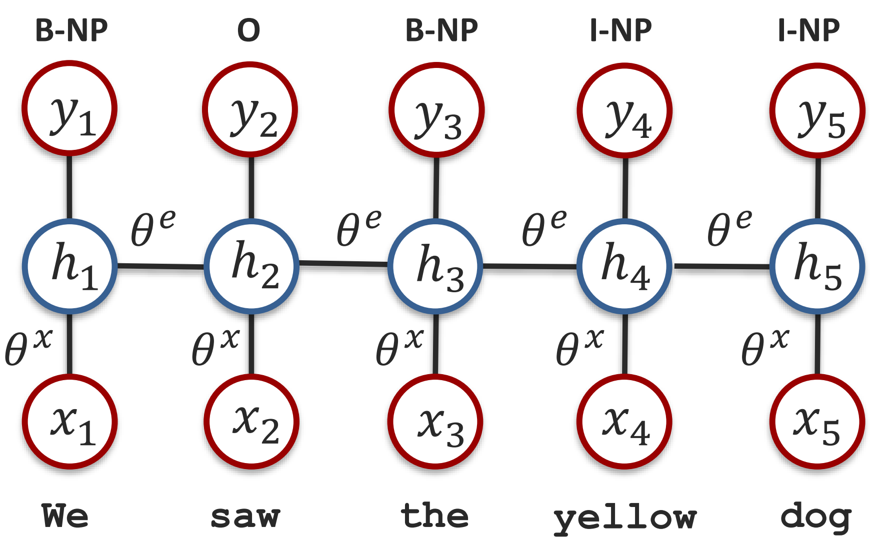
$\mathcal{H} = \{B_1, B_2, B_3, B_4\ I_1, I_2, I_3, I_4\ O_1, O_2, O_3, O_4\}$

Dictionary size: 10,000 words

Latent-Dynamic CRF

$$p(\boldsymbol{y}|\boldsymbol{x};\theta) = \sum_{\boldsymbol{h}:\forall h_t \in \mathcal{H}_{y_t}} \frac{\exp(\sum_k \theta_k f_k(\boldsymbol{h},\boldsymbol{x}))}{\sum_{\boldsymbol{h}'} \exp(\sum_k \theta_k f_k(\boldsymbol{h}',\boldsymbol{x}))}$$

- How many θ^x parameters?
- What did θ^x learn?
- How many θ^e parameters?
- What did θ^e learn?
  - Intrinsic dynamics
  - Extrinsic dynamics



Latent variables (e.g., POS tags)

$\boldsymbol{h} = \{h_1, h_2, h_3, \ldots, h_t\}$ where $h_t \in \{\mathcal{H}_{y_t}\}$

For example:

$\mathcal{H} = \{\mathcal{H}_{\mathrm{B-NP}}\ \mathcal{H}_{\mathrm{I-NP}}\ \mathcal{H}_{\mathrm{O}}\}$

$\mathcal{H} = \{B_1, B_2, B_3, B_4\ I_1, I_2, I_3, I_4\ O_1, O_2, O_3, O_4\}$

Dictionary size: 10,000 words

Latent-Dynamic CRF for Shallow Parsing

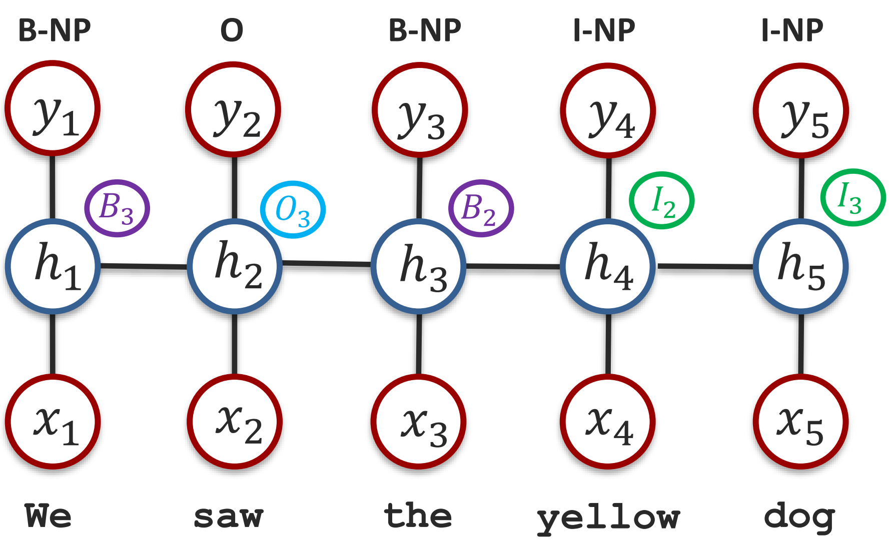
Experiment – Analyzing latent variables

- **Task:** Shallow parsing with CoNLL 2000 dataset
- **Input features:** word feature only
- **Output labels:** Noun phrase labels

1) Select hidden state a^* with highest marginal:

$$a^* = \arg\max_a \, p(\boldsymbol{h_t} = a | \boldsymbol{x}; \boldsymbol{\theta})$$

2) Compute relative frequency for each word



Label	State	Words	POS	Freq.
B	B_1	That	WDT	0.85
		who	WP	0.49
		Who	WP	0.33
	B_2	any	DT	1.00
		an	DT	1.00
		a	DT	0.98
	B_3	They	PRP	1.00
		we	PRP	1.00
		he	PRP	1.00
	B_4	Nasdaq	NNP	1.00
		Florida	NNP	0.99
		cities	NNS	0.99

Label	State	Words	POS	Freq.
O	O_1	but	CC	0.88
		by	IN	0.73
		or	IN	0.67
	O_2	4.6	CD	1.00
		1	CD	1.00
		1 1	CD	0.62
	O_3	were	VBD	0.94
		rose	VBD	0.93
		have	VBP	0.92
	O_4	been	VBN	0.97
		be	VB	0.94
		to	TO	0.92

Latent variables (e.g., POS tags)

$$\boldsymbol{h} = \{h_1, h_2, h_3, \ldots, h_t\} \qquad \text{where } h_t \in \{\mathcal{H}_{y_t}\}$$

For example:

$$\mathcal{H} = \{\mathcal{H}_{\mathrm{B-NP}} \; \mathcal{H}_{\mathrm{I-NP}} \; \mathcal{H}_{\mathrm{O}}\}$$

$$\mathcal{H} = \{B_1, B_2, B_3, B_4 \; I_1, I_2, I_3, I_4 \; O_1, O_2, O_3, O_4\}$$

Dictionary size: 10,000 words

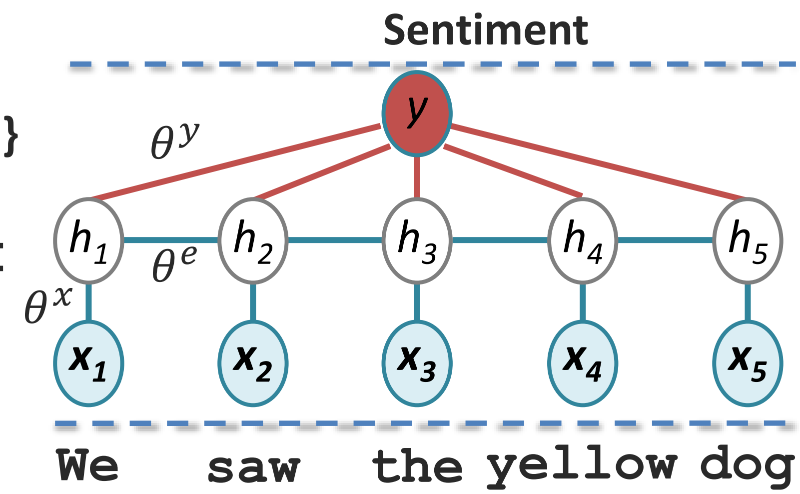
Hidden Conditional Random Field

Sequence label:

$y \in \mathcal{Y}$ for example, $\mathcal{Y}$: **{positive, negative}**

Latent variables with shared hidden states:

$\boldsymbol{h} = \{h_1, h_2, h_3, \dots, h_t\}$ where $h_t \in \mathcal{H}$



$$p(y, \boldsymbol{h} \mid \mathbf{x}; \boldsymbol{\theta}) = \frac{1}{\mathcal{Z}(\boldsymbol{x}; \boldsymbol{\theta})} \exp\left\{ \sum_t \boldsymbol{\theta}^x \cdot f^x(h_t, \mathbf{x}_t) + \underbrace{\sum_t \boldsymbol{\theta}^e \cdot f^e(h_t, h_{t-1}, y)}_{} + \underbrace{\sum_t \boldsymbol{\theta}^y \cdot f^y(y, \boldsymbol{h}_t)}_{\text{Shared hidden states}} \right\}$$

$$p(y \mid \boldsymbol{x}; \boldsymbol{\theta}) = \sum_{\boldsymbol{h}} p($$

- **Inference is tractable: $O(YH^2T)$**
  - **Linear in sequence length T !**
- **Parameter learning ($\theta^x, \theta^e, \theta^y$):**
  - **Gradient descent or L-BFGS**

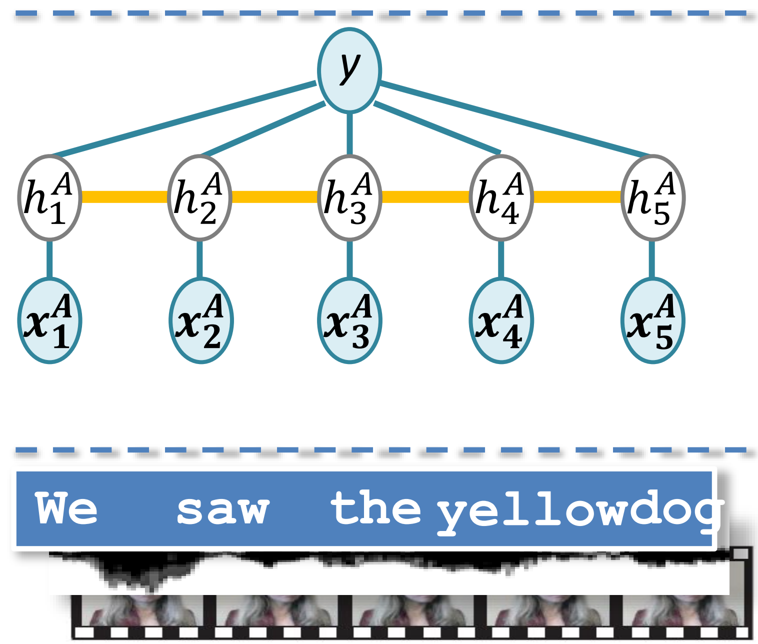
Learning Multimodal Structure

Modality-***private*** structure

- Internal grouping of observations

Modality-***shared*** structure

- Interaction and synchrony



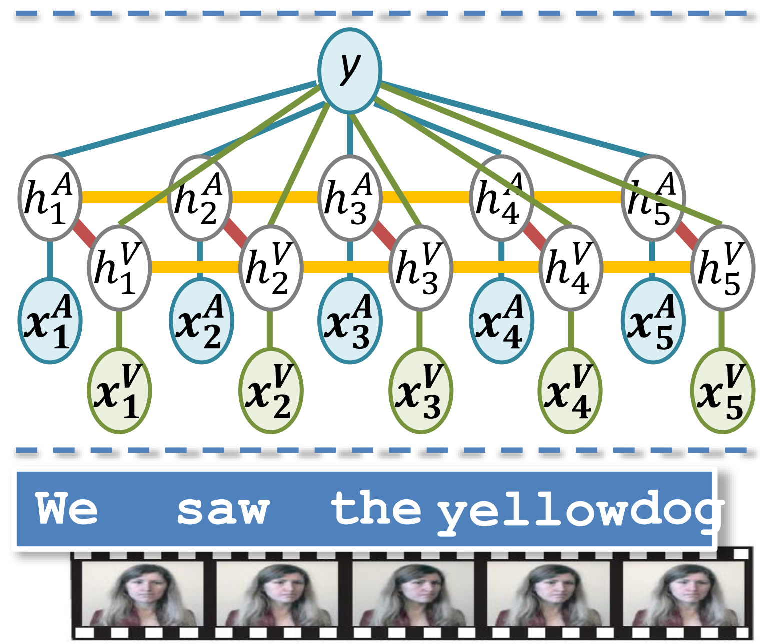
Multi-view Latent Variable Discriminative Models

Modality-*private* structure

- Internal grouping of observations

Modality-***shared*** structure

- Interaction and synchrony



$$p(y|\,\boldsymbol{x}^A,\boldsymbol{x}^V;\boldsymbol{\theta}) = \sum_{\boldsymbol{h}^A,\boldsymbol{h}^V} p\left(y,\boldsymbol{h}^A,\boldsymbol{h}^V \middle| \boldsymbol{x}^A,\boldsymbol{x}^V;\boldsymbol{\theta}\right)$$

➢ Approximate inference using loopy-belief

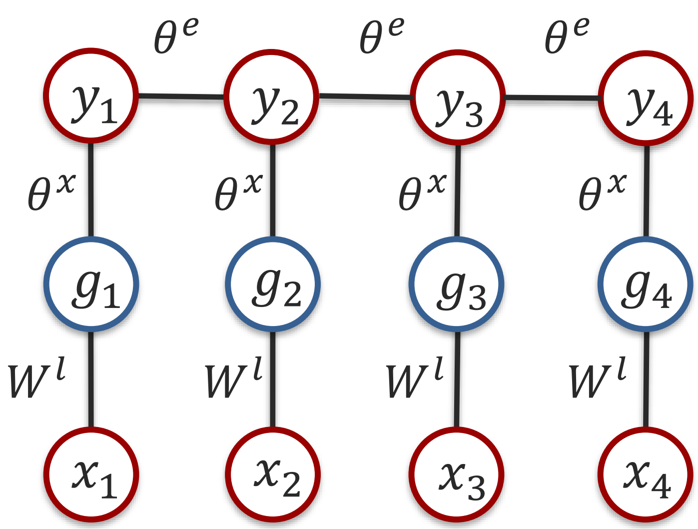
CRFs and Deep Learning

Conditional Neural Fields

$$\mathcal{G}^{l}(\boldsymbol{x}_i, \boldsymbol{W}^{l}) = [g_1^l(\boldsymbol{x}_i \cdot \boldsymbol{W}_1^l), g_2^l(\boldsymbol{x}_t \cdot \boldsymbol{W}_i^l), \ldots, g_n^l(\boldsymbol{x}_i \cdot \boldsymbol{W}_n^l)]$$

$$p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta}) \propto \exp\left\{\sum_i \boldsymbol{\theta}^x \cdot \boldsymbol{f}^x(y_i, \mathbf{x}_i) + \sum_i \boldsymbol{\theta}^e \cdot \boldsymbol{f}^e(y_i, y_{i-1})\right\}$$

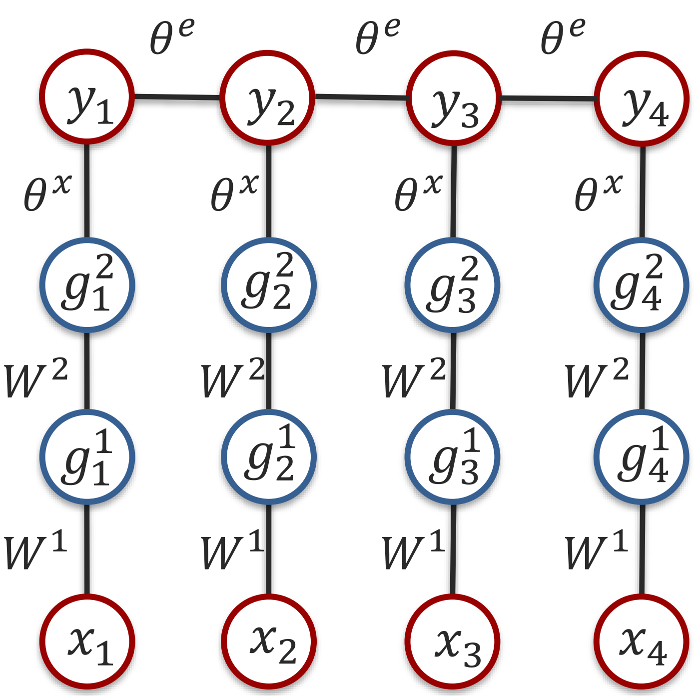
$$f^x(y_i, \mathbf{x}_i) = \mathbb{I}[y_i = y'] \cdot \mathcal{G}(\mathbf{x}_i, \boldsymbol{W}^l)$$



Deep Conditional Neural Fields

$$\mathcal{G}^{l}(\boldsymbol{x}_i, \boldsymbol{W}^{l}) = [g_1^{l}(\boldsymbol{x}_i \cdot \boldsymbol{W}_1^{l}), g_2^{l}(\boldsymbol{x}_t \cdot \boldsymbol{W}_i^{l}), \dots, g_n^{l}(\boldsymbol{x}_i \cdot \boldsymbol{W}_n^{l})]$$

$$p(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta}) \propto \exp\left\{\sum_i \boldsymbol{\theta}^x \cdot \boldsymbol{f}^x(y_i, \mathbf{x}_i) + \sum_i \boldsymbol{\theta}^e \cdot \boldsymbol{f}^e(y_i, y_{i-1})\right\}$$



$$f^x(y_i, \mathbf{x}_i) = \mathbb{I}[y_i = y'] \cdot \mathcal{G}(\boldsymbol{a}_i^{m-1}, \boldsymbol{W}^{l})$$

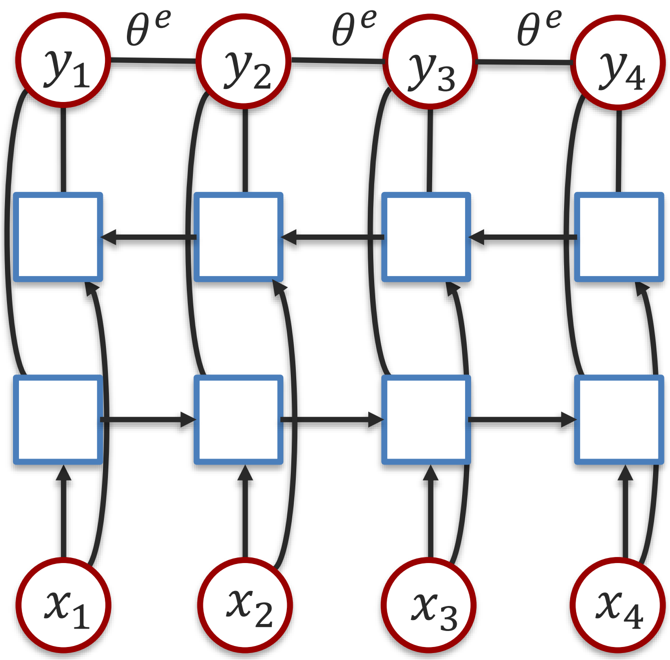
$$a^l = \mathcal{G}(\boldsymbol{a}_i^{l-1}, \boldsymbol{\theta}^g) \quad \text{for } l = 2 \dots m-1$$

Iterate

CRF and Bilinear LSTM [Dyer, 2016]

Learning:

1. Feedforward
2. Gradient
   a) Belief propagation
3. Backpropagation



Output labels:
- Name entities

Input features:
- Word embedding

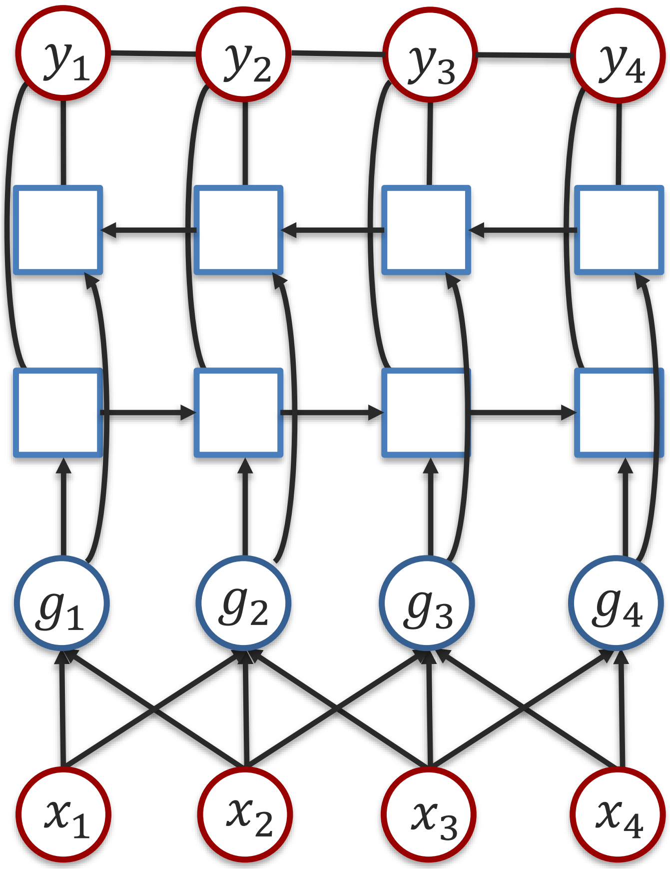
➢ What did θ^e paramters learn?

➢ What does LSTM parameters learns?

CNN and CRF and Bilinear LSTM [Hovy, 2016]

Learning:

1. Feedforward
2. Gradient
   a) Belief propagation
3. Backpropagation



Output labels:

- Name entities

Input features:

- Character embedding

Continuous and Fully-Connected CRFs

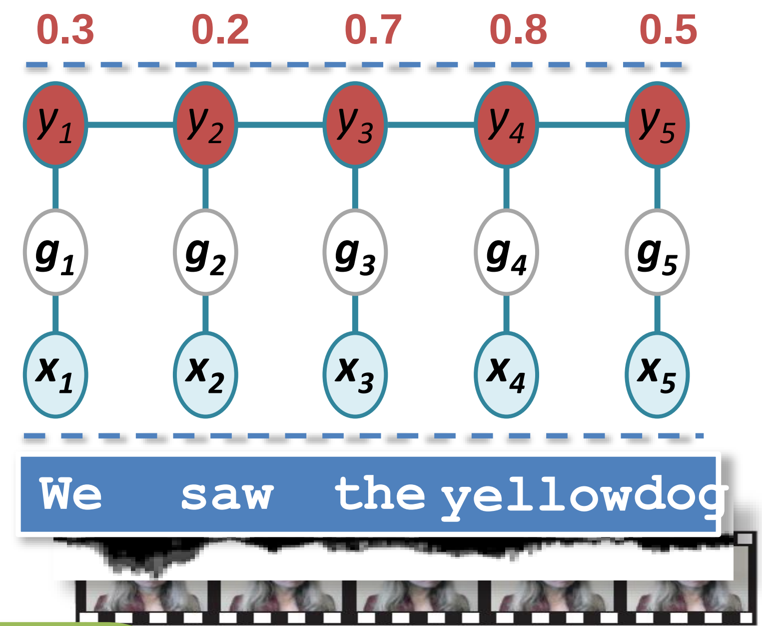
Continuous Conditional Neural Field [Baltrusaitis 2014]

Continuous output variables: **(e.g., continuous emotional label)**

$$\boldsymbol{y} = \{y_1, y_2, y_3, \ldots, y_t\} \text{ where } y_t \in \mathbb{R}$$

$$p(\boldsymbol{y}|\,\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{\mathcal{Z}(\mathbf{x};\boldsymbol{\theta})}\exp\left\{\sum_t \boldsymbol{\theta}\cdot F(y_t, y_{t-1}, \mathbf{x}_t, \boldsymbol{\theta}^g)\right\}$$

$$\mathcal{Z}(\mathbf{x};\boldsymbol{\theta}) = \int_{-\infty}^{\infty} \exp\left\{\sum_t \boldsymbol{\theta}\cdot F(y_t, y_{t-1}, \mathbf{x}_t, \boldsymbol{\theta}^g)\right\} d\boldsymbol{y}$$



➢ How to solve

Multivariate Gaussian integral:

$$\int_{-\infty}^{\infty} \exp\{\tfrac{1}{2}\,\boldsymbol{y}^T\Sigma^{-1}\boldsymbol{y} + \boldsymbol{y}\Sigma^{-1}\boldsymbol{\mu}\}d\boldsymbol{y}$$

$$= \frac{(2\pi)^{n/2}}{|\Sigma^{-1}|^{1/2}}\exp\left(\tfrac{1}{2}\,\boldsymbol{\mu}\,\Sigma^{-1}\boldsymbol{\mu}\right)$$

[Radosavljevic et al., 2010]

Continuous Conditional Neural Field

Continuous output variables: **(e.g., continuous emotional label)**

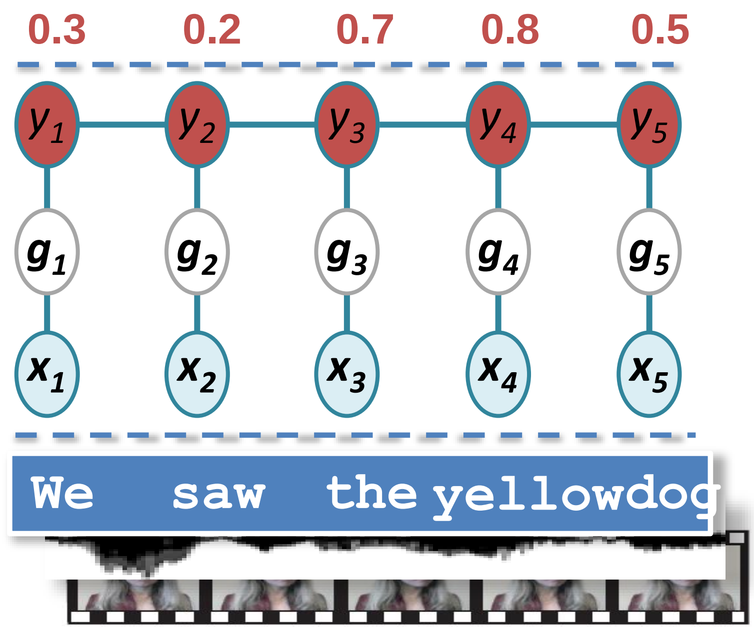
$$\boldsymbol{y} = \{y_1, y_2, y_3, \dots, y_t\} \text{ where } y_t \in \mathbb{R}$$

$$p(\boldsymbol{y}|\,\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{\mathcal{Z}(\mathbf{x};\boldsymbol{\theta})} \exp\left\{\sum_t \boldsymbol{\theta}\cdot F(y_t, y_{t-1}, \mathbf{x}_t, \boldsymbol{\theta}^g)\right\}$$

$$\mathcal{Z}(\mathbf{x};\boldsymbol{\theta}) = \int_{-\infty}^{\infty} \exp\left\{\sum_t \boldsymbol{\theta}\cdot F(y_t, y_{t-1}, \mathbf{x}_t, \boldsymbol{\theta}^g)\right\} dy$$

$$f^x(y_t, x_t, \theta^g) = -\left(y_t - g_k(x_t, \theta_k^g)\right)^2$$

$$f^e(y_t, y_{t-1}) = -\tfrac{1}{2}(y_t - y_{t-1})^2$$



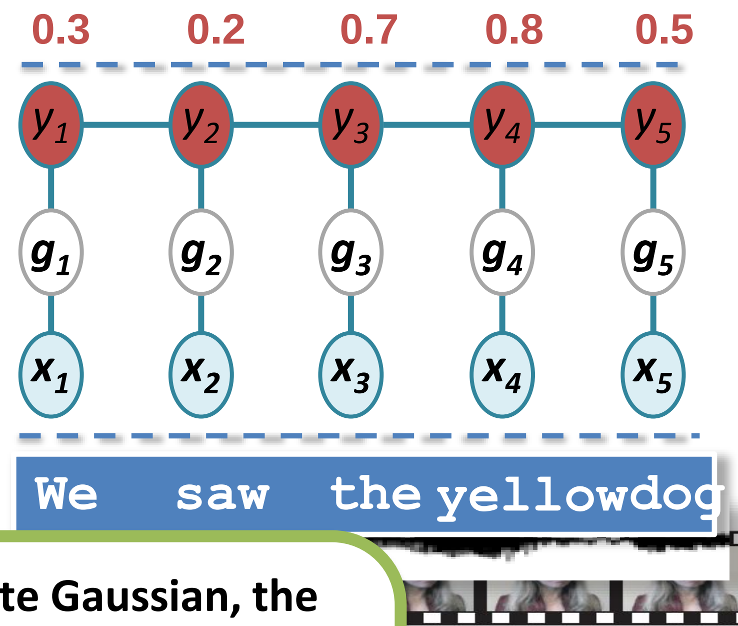
Continuous Conditional Neural Field

Continuous output variables: **(e.g., continuous emotional label)**

$$\boldsymbol{y} = \{y_1, y_2, y_3, \dots, y_t\} \text{ where } y_t \in \mathbb{R}$$

Multivariate Gaussian distribution:

$$p(\boldsymbol{y}|\,\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{y}-\boldsymbol{\mu})\right)$$



where Σ

matrix ve

and $\boldsymbol{\mu} = \Sigma$

Since CCNF can be viewed as a multivariate Gaussian, the prediction of $\boldsymbol{y}'$ is simply the mean value of distribution:

$$\boldsymbol{y}' = \arg\max_{\boldsymbol{y}}\left(P(\boldsymbol{y}|\boldsymbol{x})\right) = \boldsymbol{\mu}$$

- Optimized using gradient ascent or BFGS.

High-Order Continuous Conditional Neural Field

Continuous output variables: **(e.g., continuous emotional label)**

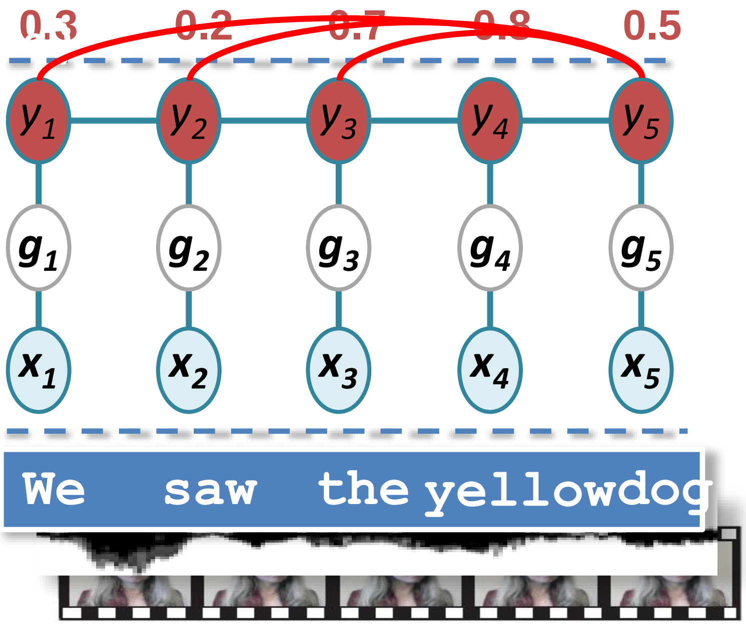
$$\boldsymbol{y} = \{y_1, y_2, y_3, \dots, y_t\} \text{ where } y_t \in \mathbb{R}$$

Multivariate Gaussian distribution:

$$p(\boldsymbol{y}|\,\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{y}-\boldsymbol{\mu})\right)$$

k-order potential functions:

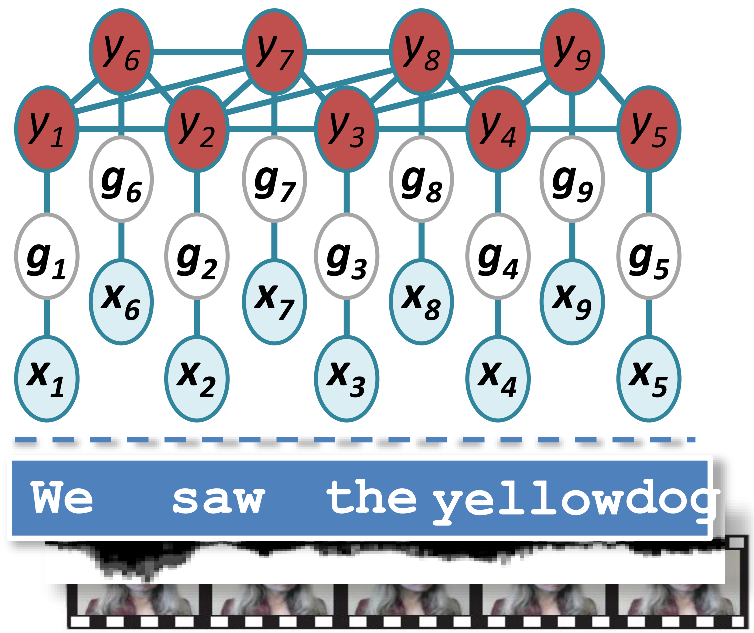
$$f^{e_k}(y_t, y_{t-k}) = -\frac{1}{2}(y_t - y_{t-k})^2$$



Fully-Connected Continuous Conditional Neural Field

Continuous output variables: **(e.g., continuous emotional label)**

$$\boldsymbol{y} = \{y_1, y_2, y_3, \dots, y_t\} \text{ where } y_t \in \mathbb{R}$$



Multivariate Gaussian distribution:

$$p(\boldsymbol{y}|\,\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{y}-\boldsymbol{\mu})\right)$$

k-order potential functions:

$$f^{e_k}(y_t, y_{t-k}) = -\frac{1}{2}(y_t - y_{t-k})^2$$

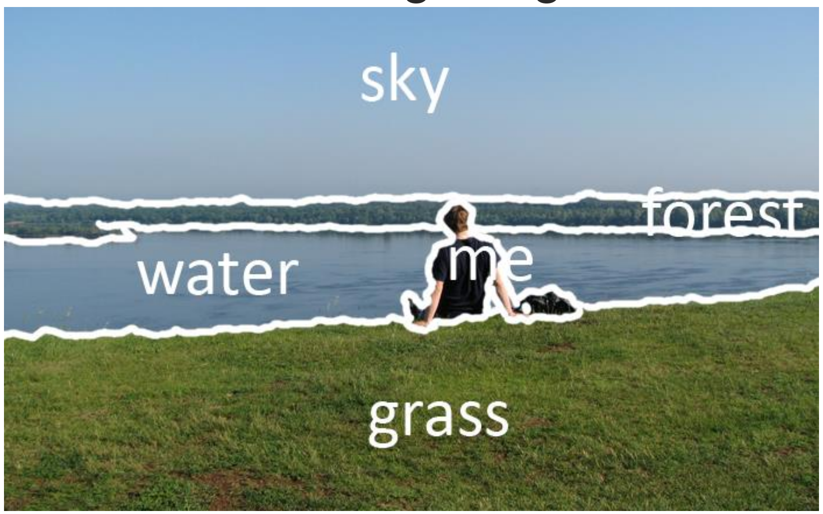
Grid potential functions:

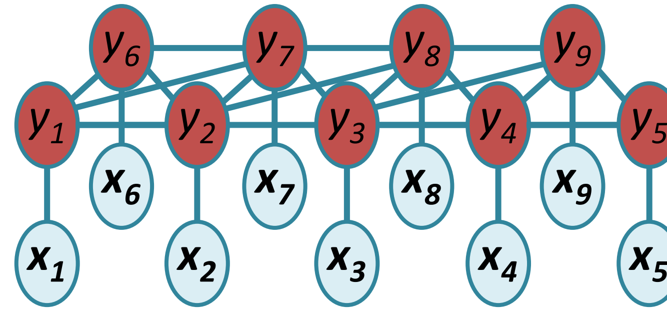
$$f^{2D}(y_i, y_j) = -\frac{1}{2} S_{ij}(y_i - y_j)^2$$

where $S_{i,j}$ specifies which nodes are connected.

Fully-Connected CRF [Krahenbuhl and Koltun, 2013]

“Semantic” image segmentation





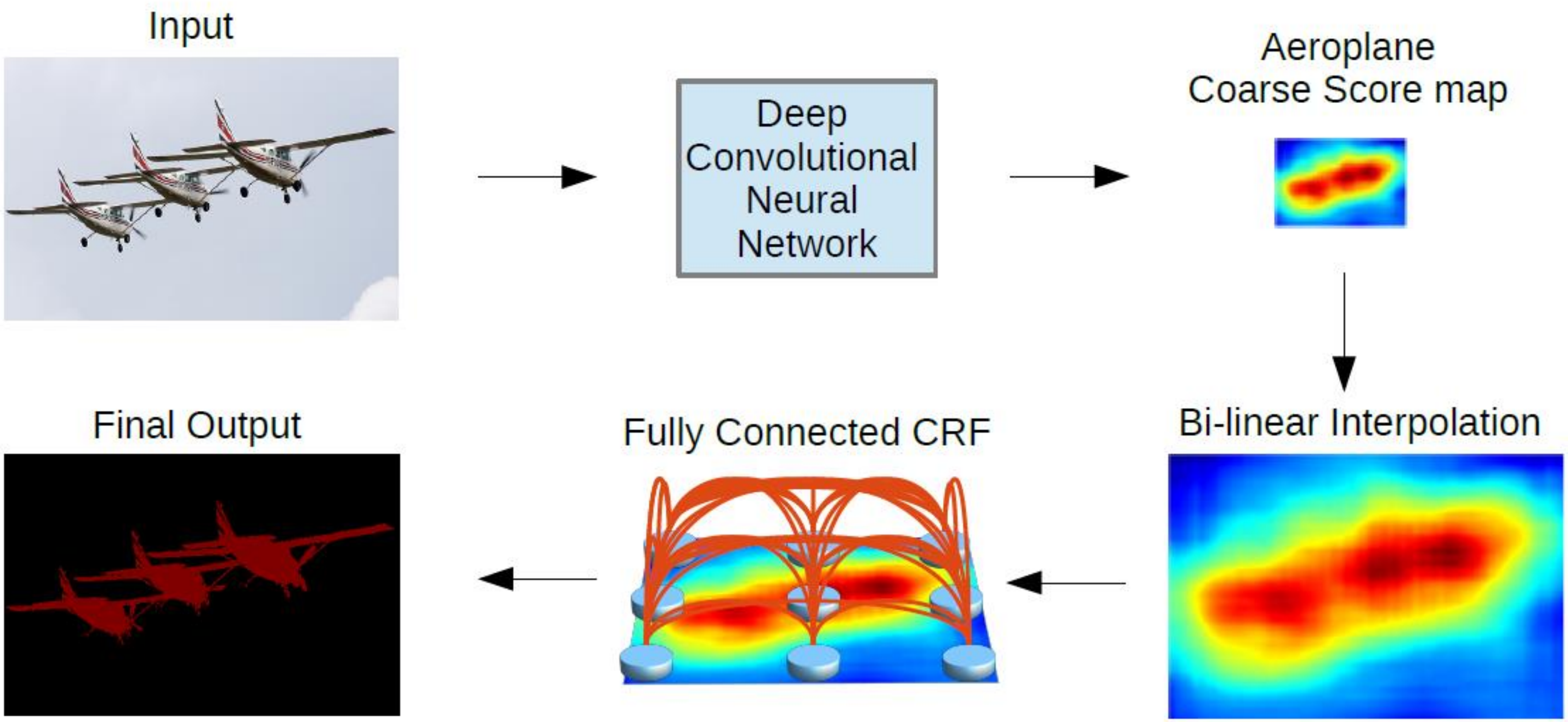
y_i: object class label

$\boldsymbol{x}_i$: local pixel features

$$p(\boldsymbol{y}|\boldsymbol{x};\theta) = \frac{\Phi(\boldsymbol{y},;\theta)}{\sum_{\boldsymbol{y}'}\Phi(\boldsymbol{y}',\boldsymbol{x};\theta)}$$

where $$\Phi_{ij}(y_i, y_j; \boldsymbol{\theta}) = \sum_{m=1}^{C} \overbrace{u^{(m)}(y_i, y_j|\boldsymbol{\theta})k^{(m)}(\boldsymbol{x}_i, \boldsymbol{x}_j)}^{\text{Mixture of kernels}}$$

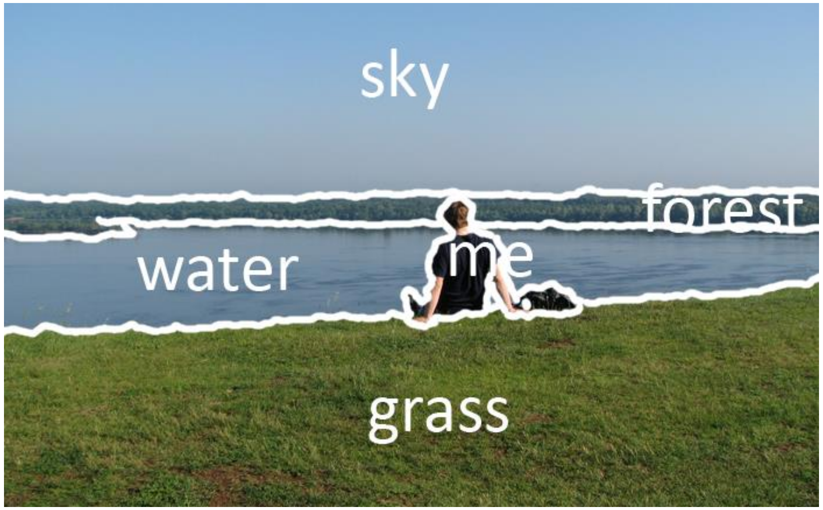
CNN and Fully-Connected CRF [Chen et al., 2014]

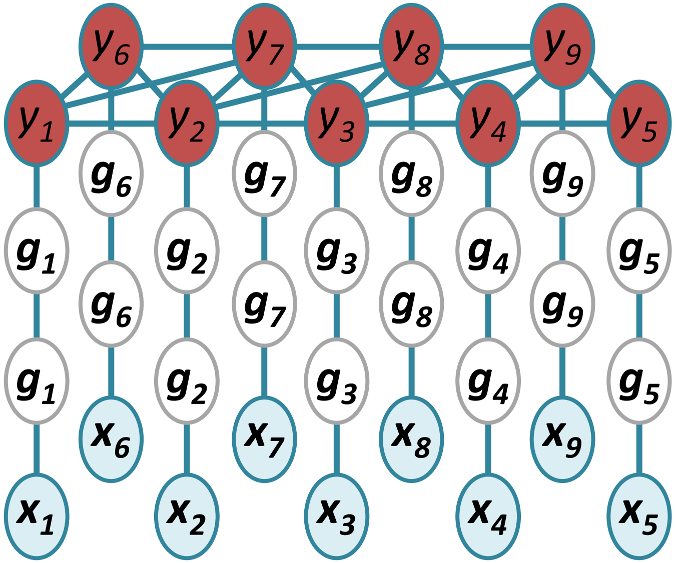


Fully Connected Deep Structured Networks

[Zheng et al., 2015; Schwing and Urtasun, 2015]

"Semantic" image segmentation





Algorithm: Learning Fully Connected Deep Structured Models

Repeat until stopping criteria

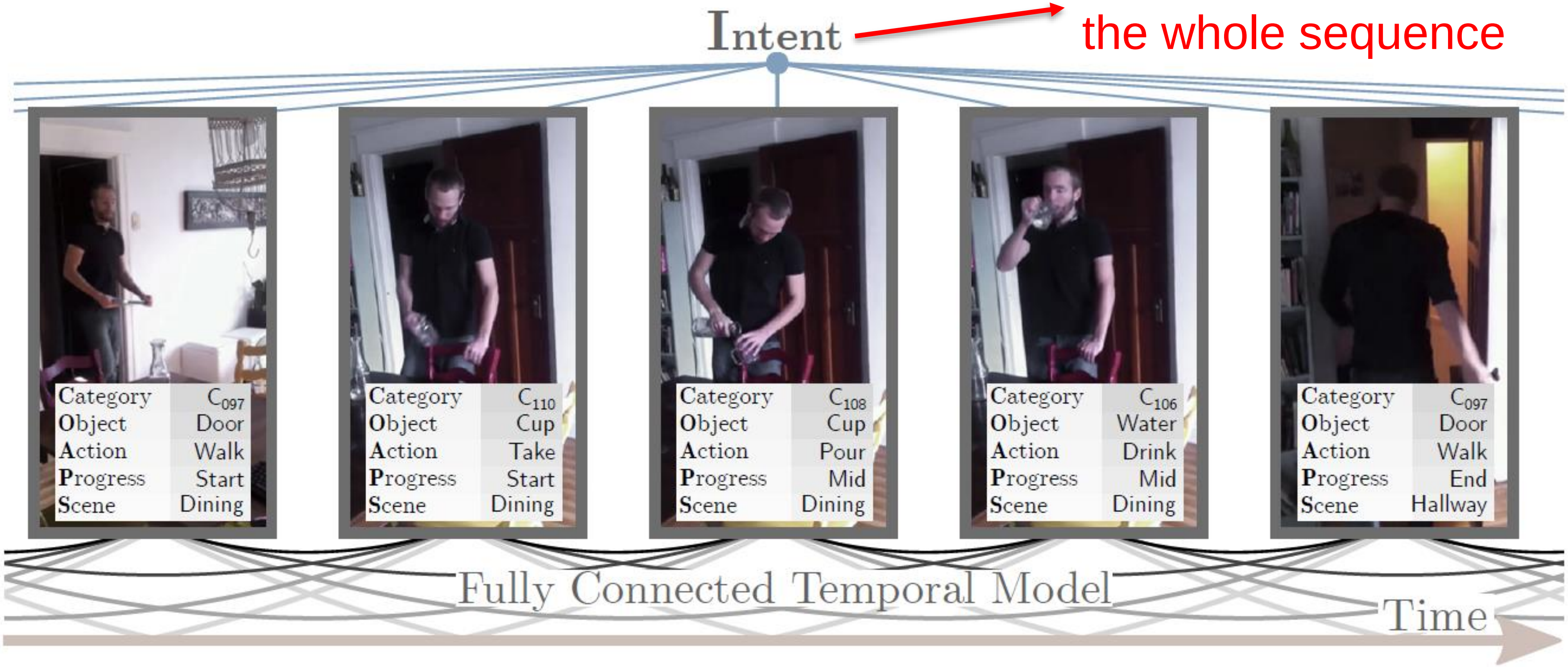
1. Forward pass to compute $f_r(x, \hat{y}_r; w)\ \forall r \in \mathcal{R}, y_r \in \mathcal{Y}_r$
2. Computation of marginals $q^t_{(x,y),i}(\hat{y}_i)$ via filtering for $t \in \{1, \dots, T\}$
3. Backtracking through the marginals $q^t_{(x,y),i}(\hat{y}_i)$ from $t = T - 1$ down to $t = 1$
4. Backward pass through definition of function via chain rule
5. Parameter update

Using mean field approximation

Fully-Connected Temporal CRF

Fully-connected CRF applied to video sequence:

Add latent variable over the whole sequence



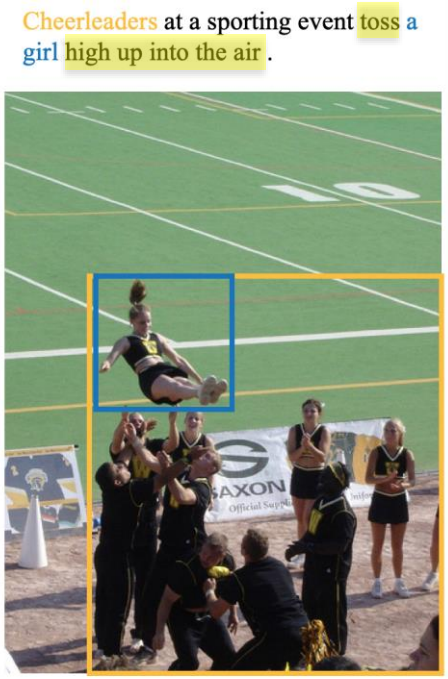
Sigurdsson et al., Asynchronous Temporal Fields for Action Recognition, CVPR 2017

Soft-Label Chain CRF

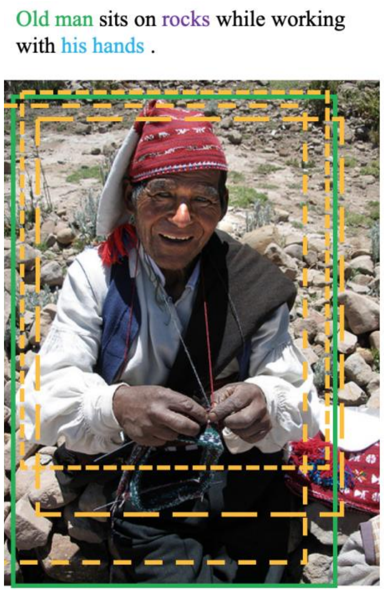
Phrase Grounding by Soft-Label Chain CRF

Two main problems:

(1) Dependencies between entities



(2) Multiple region proposals

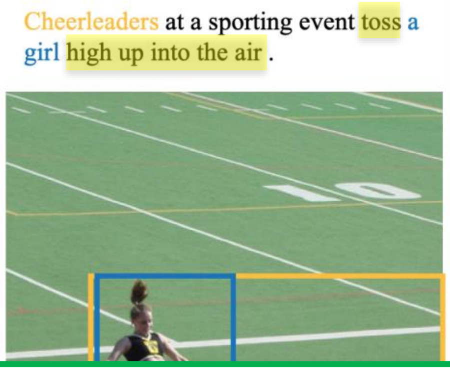


Liu J, Hockenmaier J. “Phrase Grounding by Soft-Label Chain Conditional Random Field” EMNLP 2019

Phrase Grounding by Soft-Label Chain CRF

Two main problems:

(1) Dependencies between entities



(2) Multiple region proposals

Solution: Formulate the phrase grounding as a **sequence labeling task**

- ❑ Treat the candidate regions as **potential labels**
- ❑ Propose the Soft-Label Chain CRFs to model **dependencies** among regions
- ❑ Address the **multiplicity** of gold labels

Phrase Grounding by Soft-Label Chain CRF

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp s(\boldsymbol{y}, \boldsymbol{x})}{\sum_{\boldsymbol{y}'} \exp s(\boldsymbol{y}', \boldsymbol{x})}$$

- Input sequence: $\boldsymbol{x} = x^{1:T}$
- Label sequence: $\boldsymbol{y} = y^{1:T}$
- Score function: $s(\boldsymbol{x}, \boldsymbol{y})$

Standard CRF

- Cross-entropy Loss: $L = -\log p(\boldsymbol{y}|\boldsymbol{x}) = -s(\boldsymbol{y}, \boldsymbol{x}) + \log Z(\boldsymbol{x})$

  ➢ Each input x^i is associated to only one label y^i

Soft-Label CRF:

- KL-divergence between the model and target distribution:

$$L = \sum_{\boldsymbol{y}} \left\{ q(\boldsymbol{y}|\boldsymbol{x}) \log \frac{q(\boldsymbol{y}|\boldsymbol{x})}{p(\boldsymbol{y}|\boldsymbol{x})} \right\}$$

- Sequence of target distribution: $\boldsymbol{q} = q^{1:T}$
- **Label distribution** over all K possible labels for input x^t: $q^t \in \mathbb{R}^K$

  ➢ Each input x^i is associated to a distribution of labels y^i

Liu J, Hockenmaier J. "Phrase Grounding by Soft-Label Chain Conditional Random Field" EMNLP 2019

Phrase Grounding by Soft-Label Chain CRF

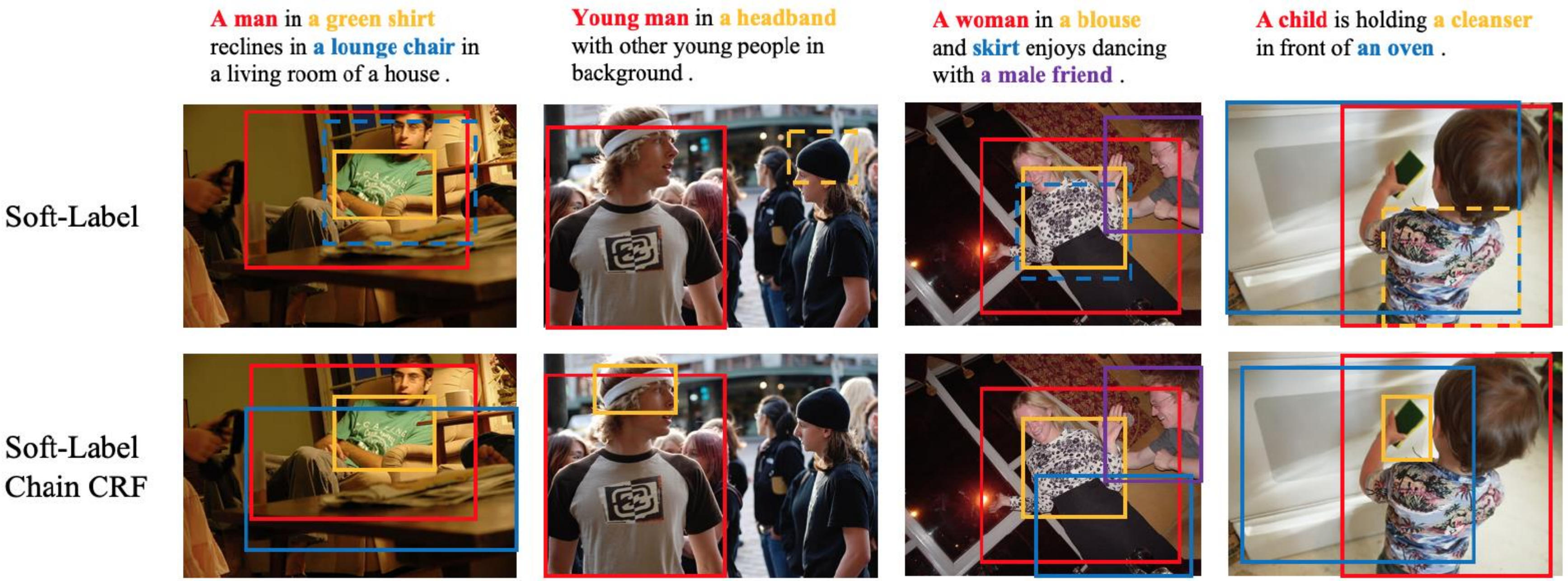
For efficiency: Reduce the model to a first-order linear chain CRF, whose scoring function factorizes as:

$$\begin{aligned} s(\boldsymbol{y}, \boldsymbol{x}) &= \sum_t s(y^t, y^{t-1}, \boldsymbol{x}) \\ &= \sum_t \left\{ \tau(y^t, y^{t-1}, \boldsymbol{x}) + \varepsilon(y^t, \boldsymbol{x}) \right\} \end{aligned}$$

where $\tau(\cdot,\cdot,\cdot)$ are the pairwise potentials between labels at $t-1$ and t
$\varepsilon(\cdot,\cdot)$ are the unary potentials between label and input at t

Liu J, Hockenmaier J. "Phrase Grounding by Soft-Label Chain Conditional Random Field" EMNLP 2019

Phrase Grounding by Soft-Label Chain CRF



Liu J, Hockenmaier J. “Phrase Grounding by Soft-Label Chain Conditional Random Field” EMNLP 2019