Modeling Structure with Undirected Neural Networks

Tsvetomila Mihaylova[1], Vlad Niculae[2], André F. T. Martins[1,3,4]

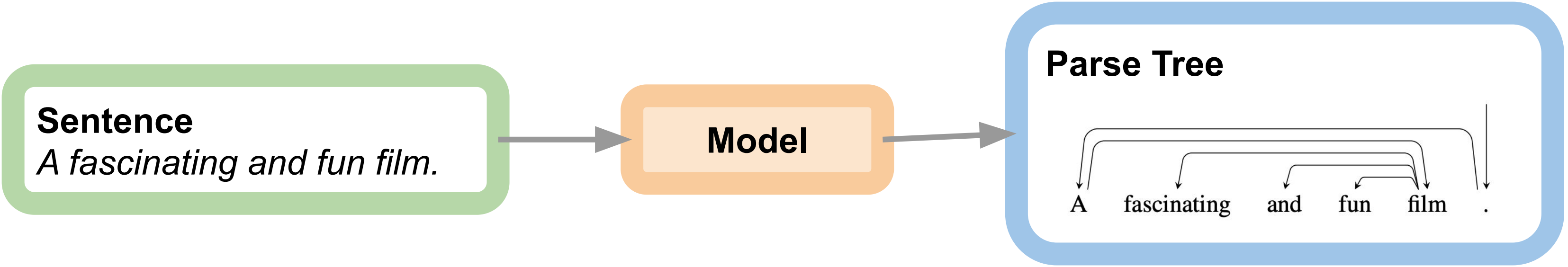
ICML, July 2022

1: Instituto de Telecomunicações, Instituto Superior Técnico
2: Informatics Institute, University of Amsterdam
3: LUMLIS (Lisbon ELLIS Unit)
4: Unbabel

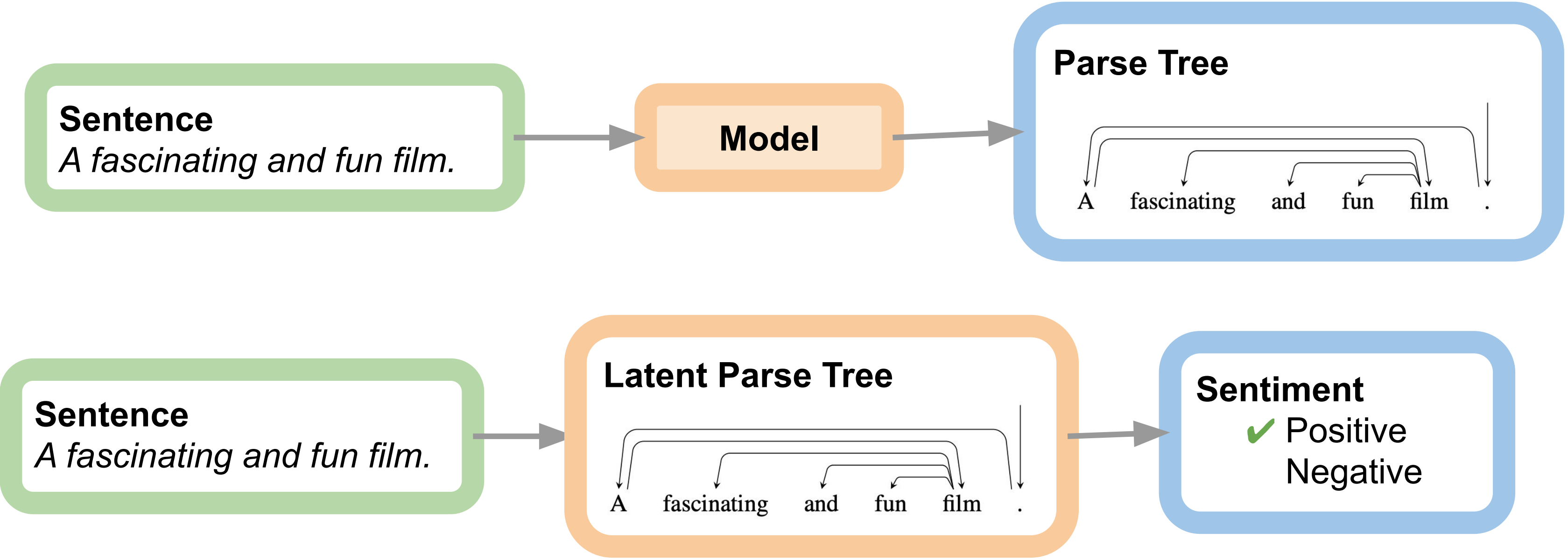
Neural Networks are

a preferred choice for modeling structured data

Neural Networks are
a preferred choice for modeling structured data



Neural Networks are
a preferred choice for modeling structured data



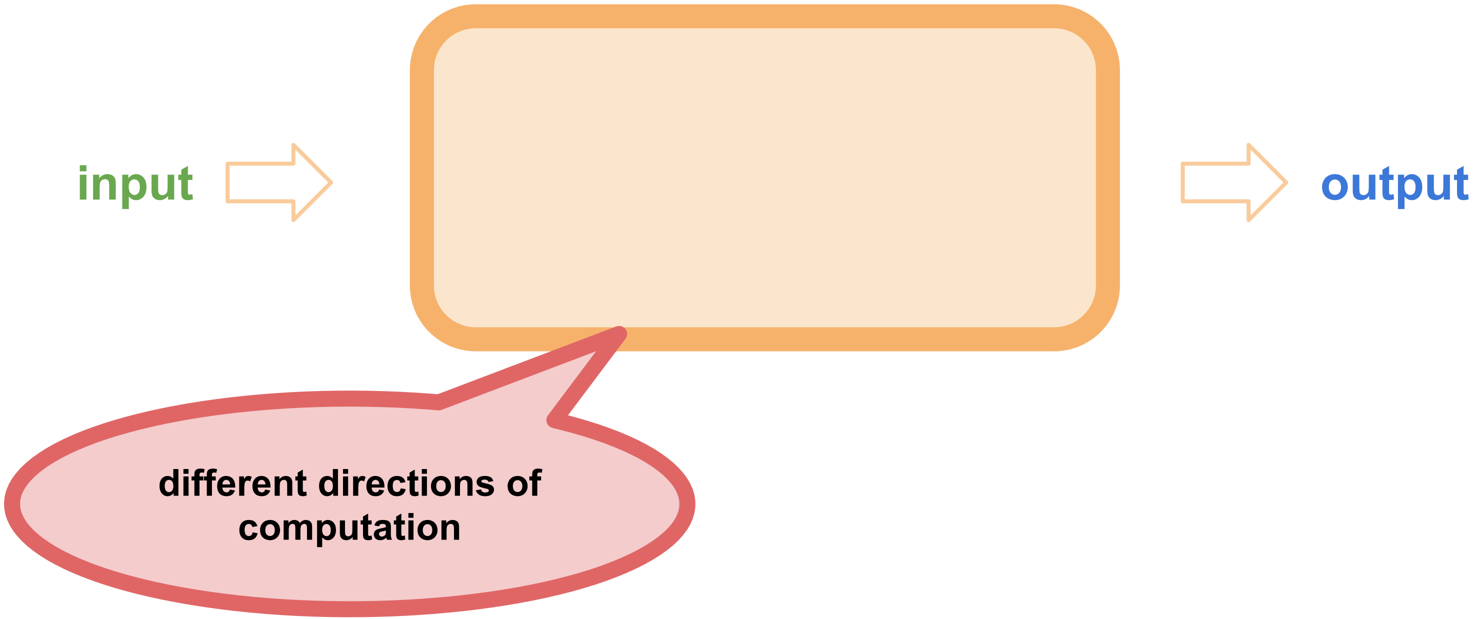
Neural Networks are
(usually) monolithic mappings from inputs to outputs

Neural Networks are

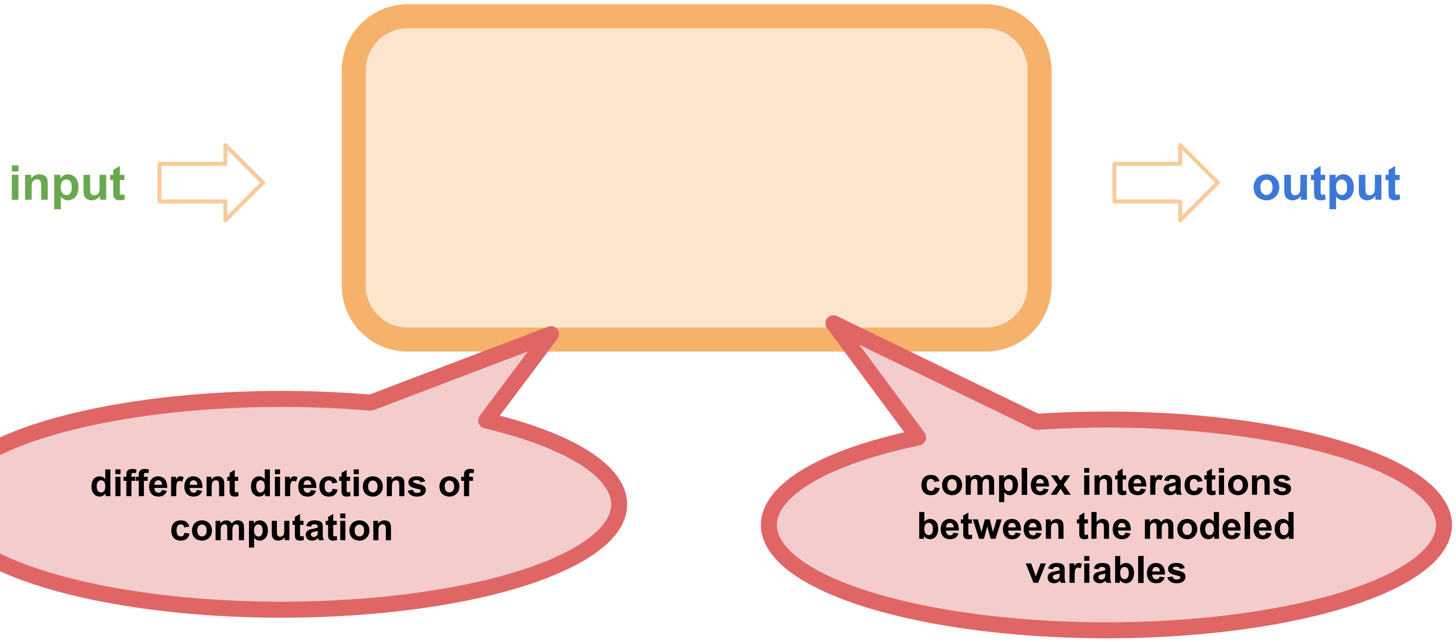
(usually) monolithic mappings from inputs to outputs with fixed computation order

Which prevents them from capturing…

Which prevents them from capturing…



Which prevents them from capturing…



In this work:

Combine **factor graphs** and **neural networks**

In this work:

Combine **factor graphs** and **neural networks**

proposing

Undirected Neural Networks (UNNs)

In this work:

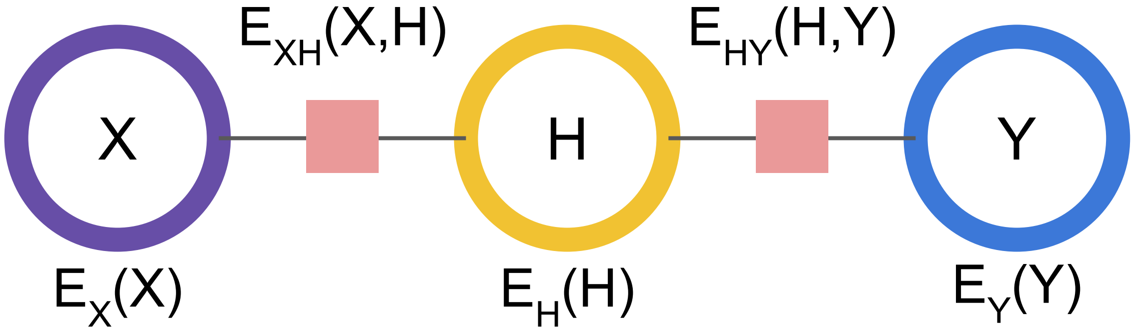
Combine **factor graphs** and **neural networks**

proposing

Undirected Neural Networks (UNNs)

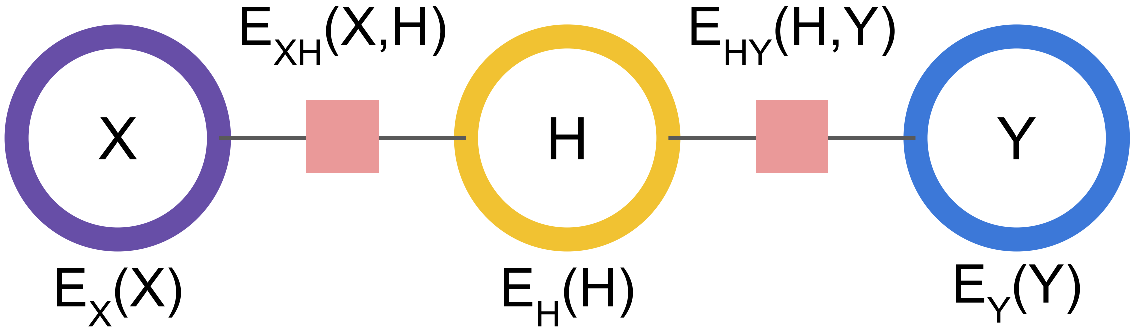
flexible framework, computations that can be performed in any order

Neural Networks + Factor Graphs = **Undirected Neural Networks**



Outputs are:

Neural Networks + Factor Graphs =
Undirected Neural Networks

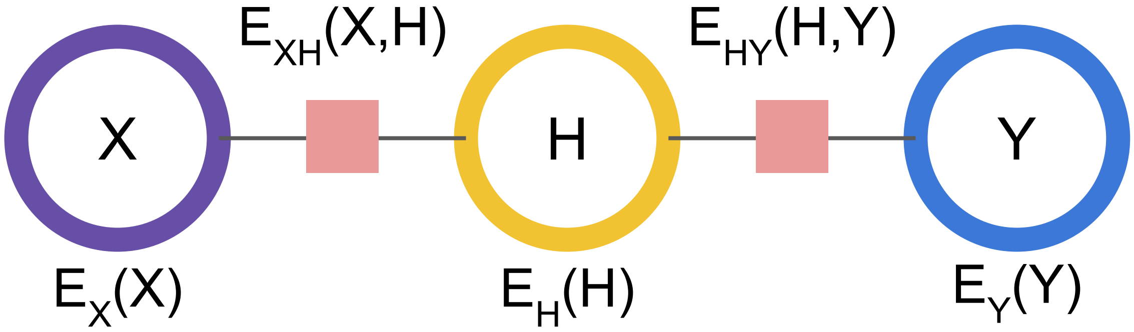


Outputs are:

not computed by evaluating a composition of functions in a given order, but

Neural Networks + Factor Graphs =

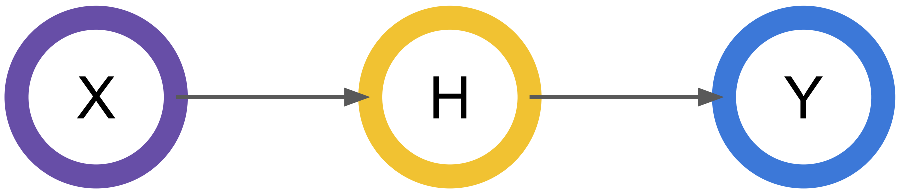
Undirected Neural Networks



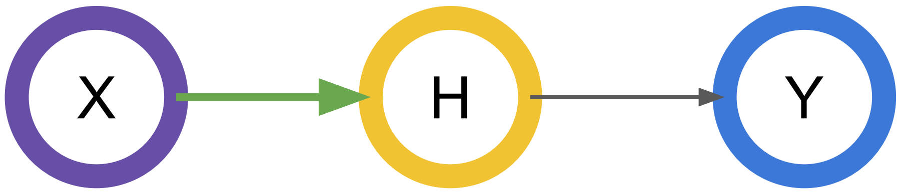
Outputs are:

not computed by evaluating a composition of functions in a given order, but **obtained implicitly by minimizing an energy function** which factors over a graph.

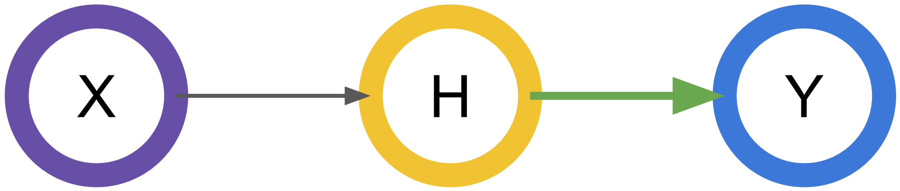
Example: MLP



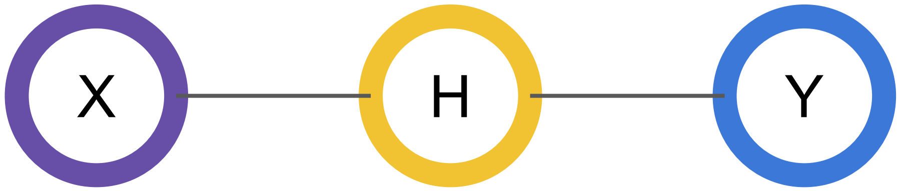
Example: MLP



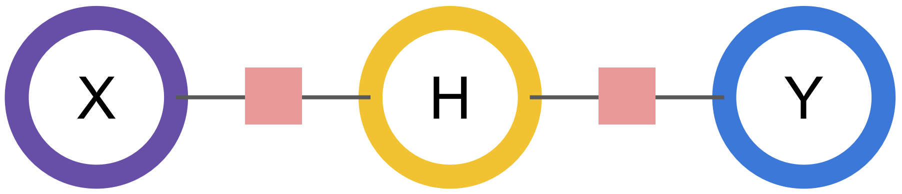
Example: MLP



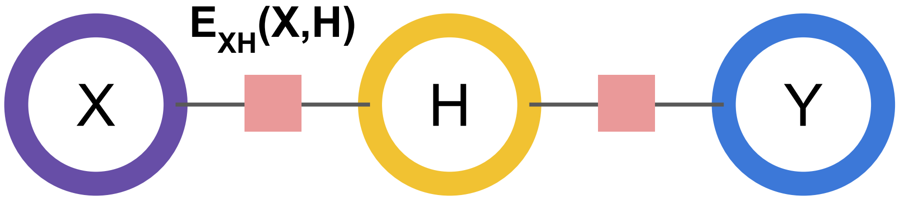
Example: MLP



Example: MLP

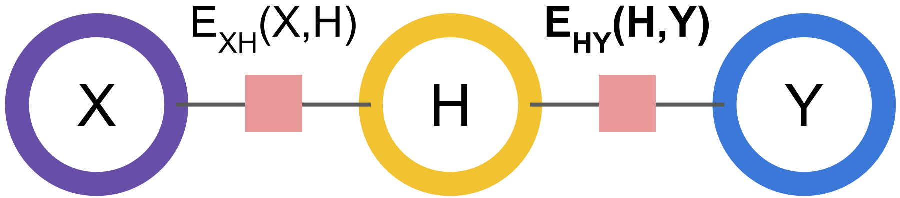


Example: MLP



$$E_{\mathsf{XH}}(x,h) = -\langle h, Wx \rangle$$

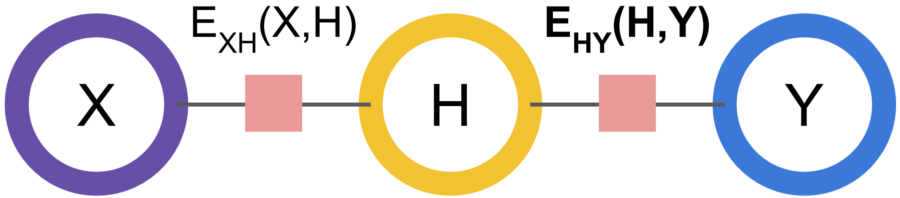
Example: MLP



$$E_{\mathsf{XH}}(x,h) = -\langle h, Wx \rangle$$

$$E_{\mathsf{HY}}(h,y) = -\langle y, Vh \rangle$$

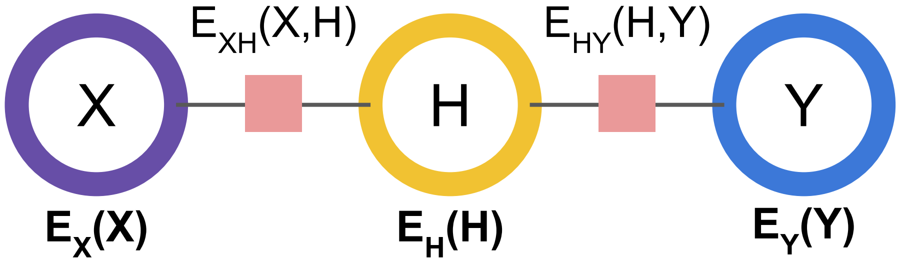
Example: MLP



$$\left.\begin{aligned} E_{\mathsf{XH}}(x,h) &= -\langle h, Wx\rangle \\ E_{\mathsf{HY}}(h,y) &= -\langle y, Vh\rangle \end{aligned}\right\}$$

multilinear energy

Example: MLP



$$E_{\mathsf{XH}}(x,h) = -\langle h, Wx \rangle$$

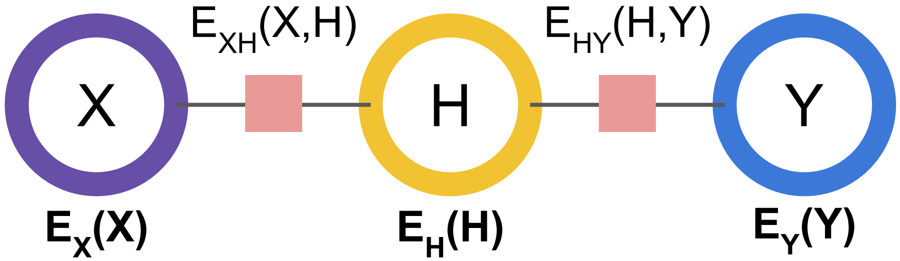
$$E_{\mathsf{HY}}(h,y) = -\langle y, Vh \rangle$$

multilinear energy

$$E_{\mathsf{Z}}(x) = -\langle x, b_{\mathsf{Z}} \rangle + \Psi_{\mathsf{Z}}(x)$$

$$\mathsf{Z} \in \{\mathsf{X}, \mathsf{H}, \mathsf{Y}\}$$

Example: MLP



$$E_{\mathsf{XH}}(x,h) = -\langle h, Wx \rangle$$

$$E_{\mathsf{HY}}(h,y) = -\langle y, Vh \rangle$$

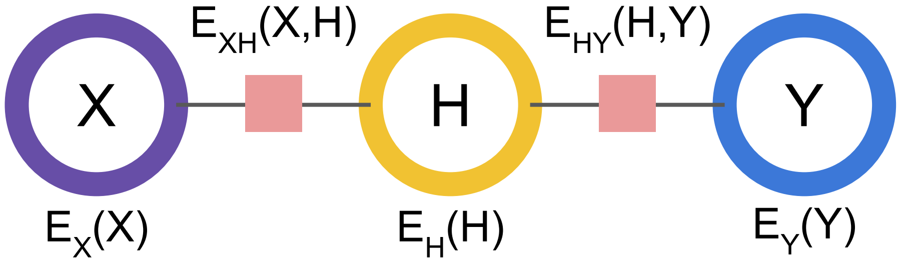
multilinear energy

$$E_{\mathsf{Z}}(x) = -\langle x, b_{\mathsf{Z}} \rangle + \Psi_{\mathsf{Z}}(x)$$

$$\mathsf{Z} \in \{\mathsf{X}, \mathsf{H}, \mathsf{Y}\}$$

linear plus a **convex regularizer Ψ**

Example: MLP



$$E_{\mathsf{XH}}(x,h) = -\langle h, Wx \rangle$$

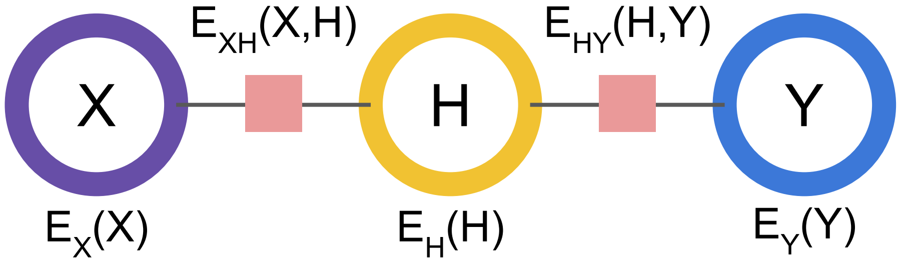
$$E_{\mathsf{HY}}(h,y) = -\langle y, Vh \rangle$$

$$E_{\mathsf{Z}}(x) = -\langle x, b_{\mathsf{Z}} \rangle + \Psi_{\mathsf{Z}}(x)$$

$$\mathsf{Z} \in \{\mathsf{X}, \mathsf{H}, \mathsf{Y}\}$$

?

Example: MLP



$$\left.\begin{aligned} E_{\mathsf{XH}}(x,h) &= -\langle h, Wx\rangle \\ E_{\mathsf{HY}}(h,y) &= -\langle y, Vh\rangle \\ E_{\mathsf{Z}}(x) &= -\langle x, b_{\mathsf{Z}}\rangle + \Psi_{\mathsf{Z}}(x) \\ \mathsf{Z} &\in \{\mathsf{X},\mathsf{H},\mathsf{Y}\} \end{aligned}\right\}$$

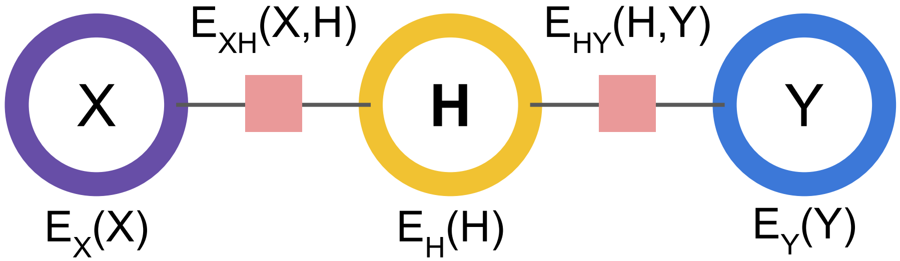
For specific choices of **Ψ**

minimization wrt **every variable**

can be done in closed form

given the others.

Example: MLP



$$E_{\mathsf{XH}}(x,h) = -\langle h, Wx \rangle$$

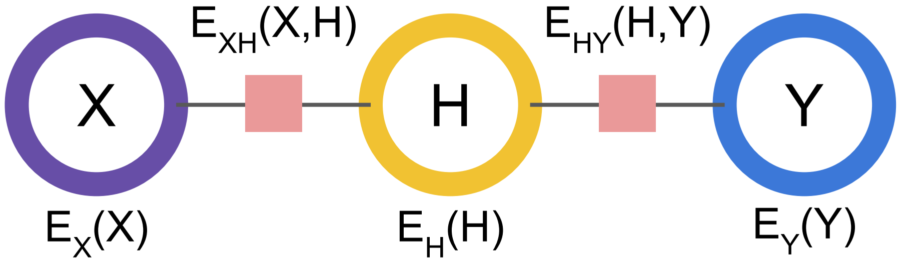
$$E_{\mathsf{HY}}(h,y) = -\langle y, Vh \rangle$$

$$E_{\mathsf{Z}}(x) = -\langle x, b_{\mathsf{Z}} \rangle + \Psi_{\mathsf{Z}}(x)$$

$$\mathsf{Z} \in \{\mathsf{X}, \mathsf{H}, \mathsf{Y}\}$$

$$h_\star = (\nabla \Psi^*_{\mathsf{H}})(Wx + V^\top y + b_{\mathsf{H}})$$

Example: MLP



$$E_{\mathsf{XH}}(x,h) = -\langle h, Wx \rangle$$

$$E_{\mathsf{HY}}(h,y) = -\langle y, Vh \rangle$$

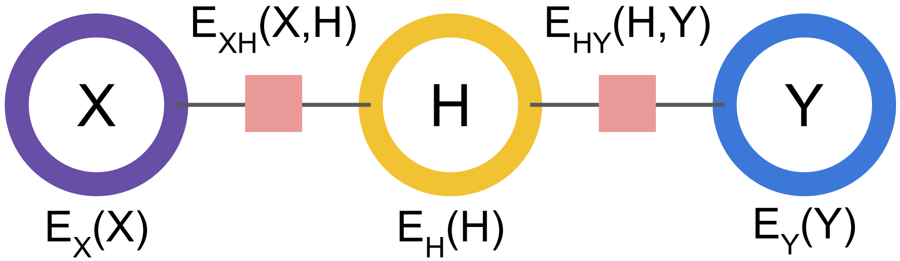
$$E_{\mathsf{Z}}(x) = -\langle x, b_{\mathsf{Z}} \rangle + \Psi_{\mathsf{Z}}(x)$$

$\mathsf{Z} \in \{\mathsf{X}, \mathsf{H}, \mathsf{Y}\}$

$$h_\star = (\nabla \Psi_{\mathsf{H}}^*)(Wx + V^\top y + b_{\mathsf{H}})$$

$$\boxed{\Psi(h) = \frac{1}{2}||h||^2 + \iota_{\mathbb{R}_+}(h)}$$

Example: MLP



$$E_{\mathsf{XH}}(x,h) = -\langle h, Wx \rangle$$

$$E_{\mathsf{HY}}(h,y) = -\langle y, Vh \rangle$$

$$E_{\mathsf{Z}}(x) = -\langle x, b_{\mathsf{Z}} \rangle + \Psi_{\mathsf{Z}}(x)$$

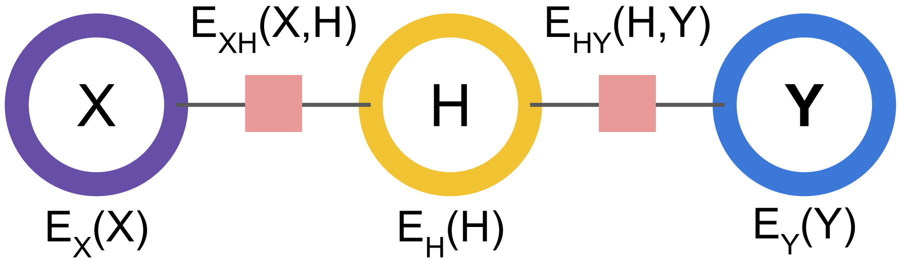
$$\mathsf{Z} \in \{\mathsf{X}, \mathsf{H}, \mathsf{Y}\}$$

$$h_\star = (\nabla \Psi^*_{\mathsf{H}})(Wx + V^\top y + b_{\mathsf{H}})$$

$$\Psi(h) = \frac{1}{2}||h||^2 + \iota_{\mathbb{R}_+}(h)$$

$$h_\star = \mathrm{ReLU}\,(Wx + V^\top y + b_{\mathsf{H}})$$

Example: MLP



$$E_{\mathsf{XH}}(x,h) = -\langle h, Wx \rangle$$

$$E_{\mathsf{HY}}(h,y) = -\langle y, Vh \rangle$$

$$E_{\mathsf{Z}}(x) = -\langle x, b_{\mathsf{Z}} \rangle + \Psi_{\mathsf{Z}}(x)$$

$$\mathsf{Z} \in \{\mathsf{X}, \mathsf{H}, \mathsf{Y}\}$$

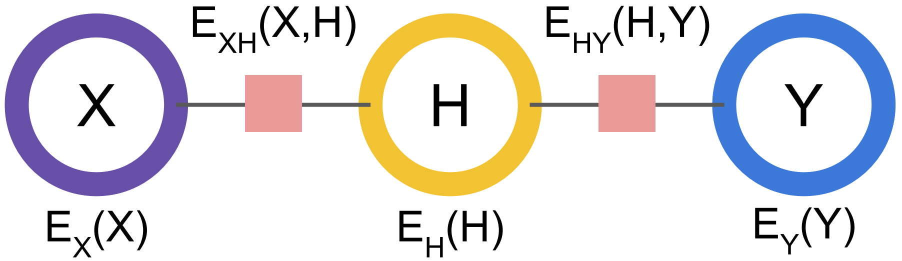
$$h_\star = (\nabla \Psi^*_{\mathsf{H}})(Wx + V^\top y + b_{\mathsf{H}})$$

$$y_\star = (\nabla \Psi^*_{\mathsf{Y}})(Vh + b_{\mathsf{Y}})$$

$\Psi(h)$	$(\nabla\Psi^*)(t)$
$\frac{1}{2}\|h\|^2$	t
$\frac{1}{2}\|h\|^2+\iota_{\mathbb{R}_+}(h)$	$\mathrm{relu}(t)$
$\sum_j \big(\phi(h_j)+\phi(1-h_j)\big)+\iota_{[0,1]^d}(h)$	$\mathrm{sigmoid}(t)$
$\sum_j \left(\phi\left(\frac{1+h_j}{2}\right)+\phi\left(\frac{1-h_j}{2}\right)\right)+\iota_{[-1,1]^d}(h)$	$\tanh(t)$
$-\mathcal{H}(h)+\iota_\Delta(h)$	$\mathrm{softmax}(t)$

Table 1: Examples of regularizers $\Psi(h)$ corresponding to some common activation functions, where $\phi(t)=t\log t$.

Example: MLP



$$E_{\mathsf{XH}}(x,h) = -\langle h, Wx \rangle$$

$$E_{\mathsf{HY}}(h,y) = -\langle y, Vh \rangle$$

$$E_{\mathsf{Z}}(x) = -\langle x, b_{\mathsf{Z}} \rangle + \Psi_{\mathsf{Z}}(x)$$

$$\mathsf{Z} \in \{\mathsf{X}, \mathsf{H}, \mathsf{Y}\}$$

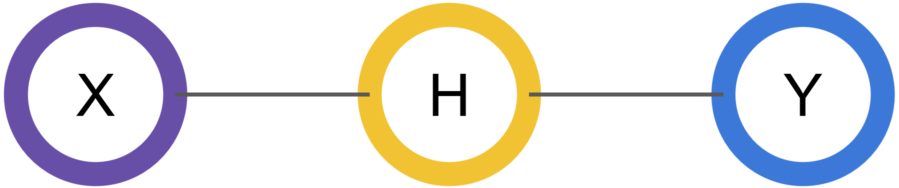
$$h_\star = (\nabla \Psi_{\mathsf{H}}^*)(Wx + V^\top y + b_{\mathsf{H}}),$$

$$y_\star = (\nabla \Psi_{\mathsf{Y}}^*)(Vh + b_{\mathsf{Y}}).$$

*(For **k** iterations.)*

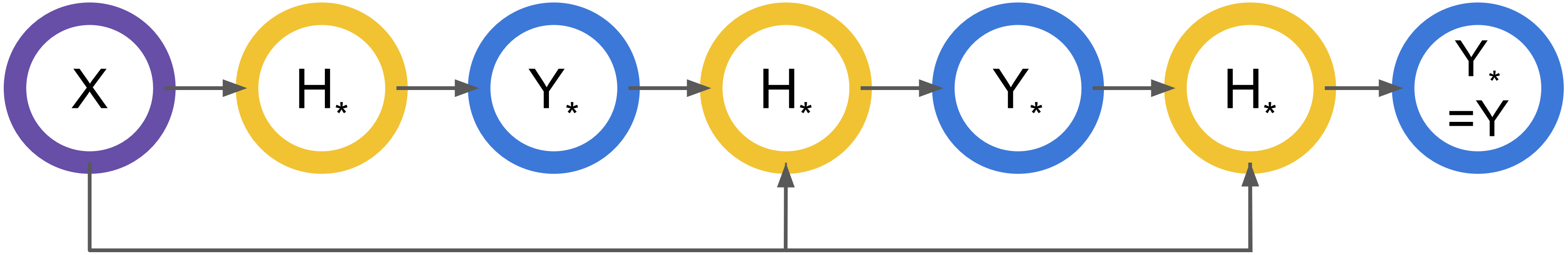
Unrolling the UNN

k=3



Unrolling the UNN

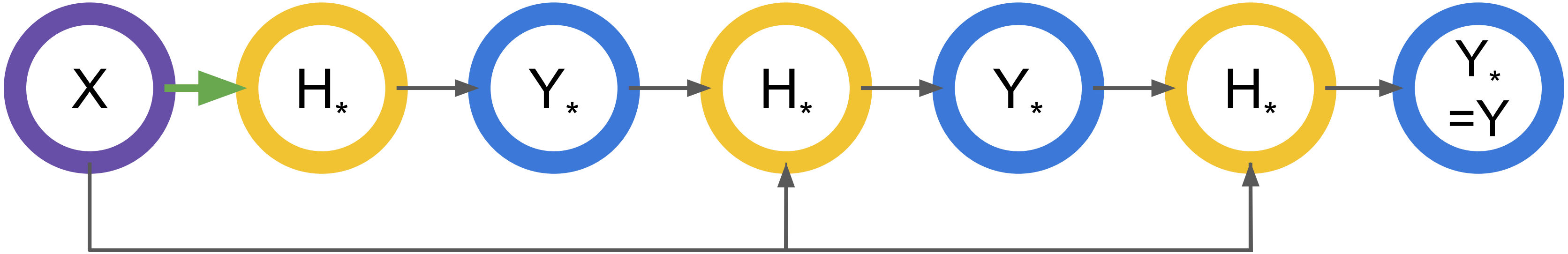
k=3



Unrolling the UNN

k=3

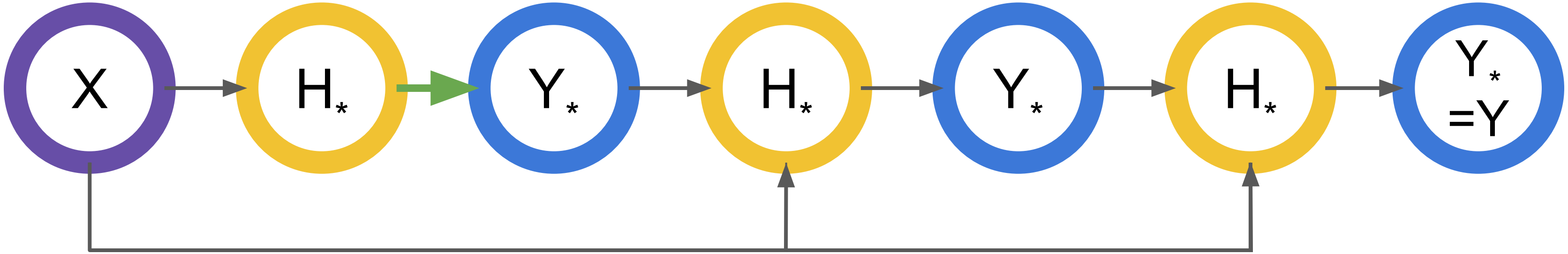
i=1



Unrolling the UNN

k=3

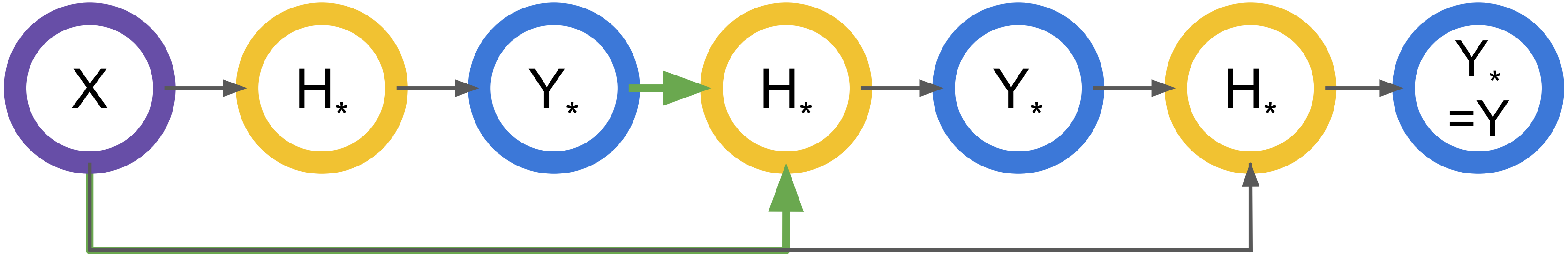
i=1



Unrolling the UNN

k=3

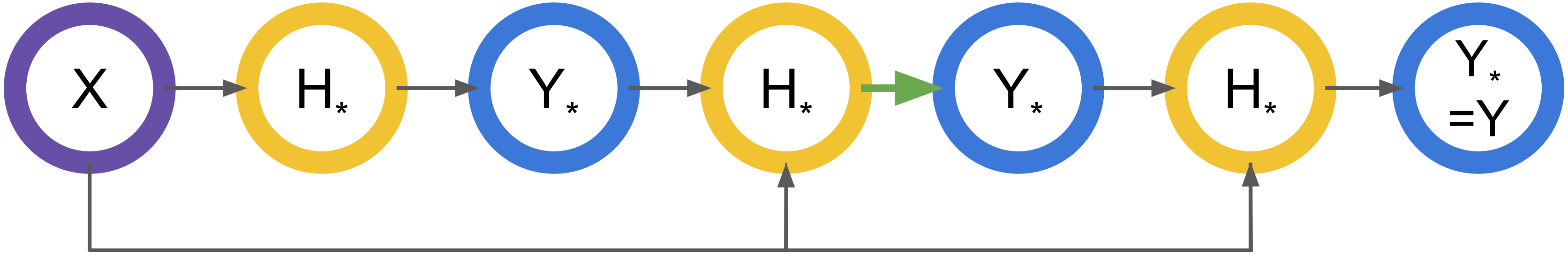
i=2



Unrolling the UNN

k=3

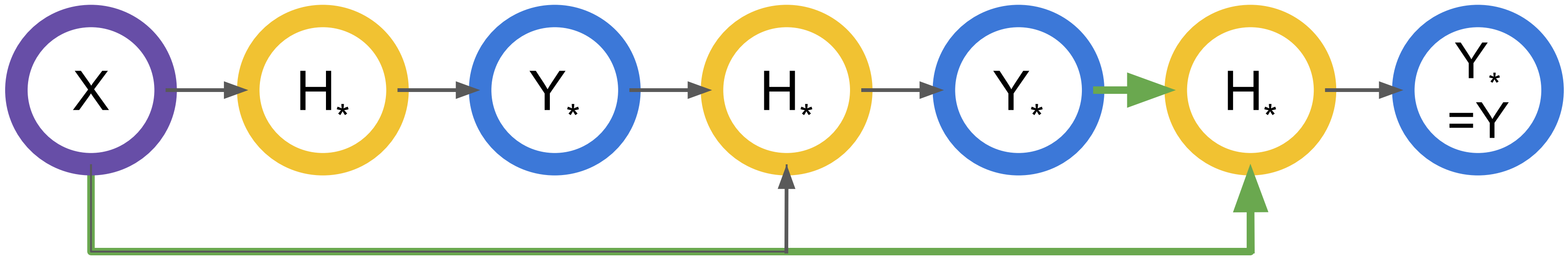
i=2



Unrolling the UNN

k=3

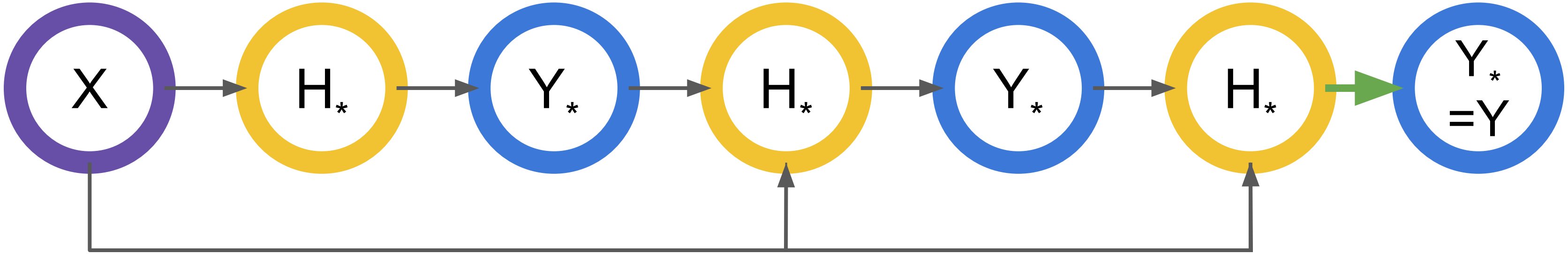
i=3



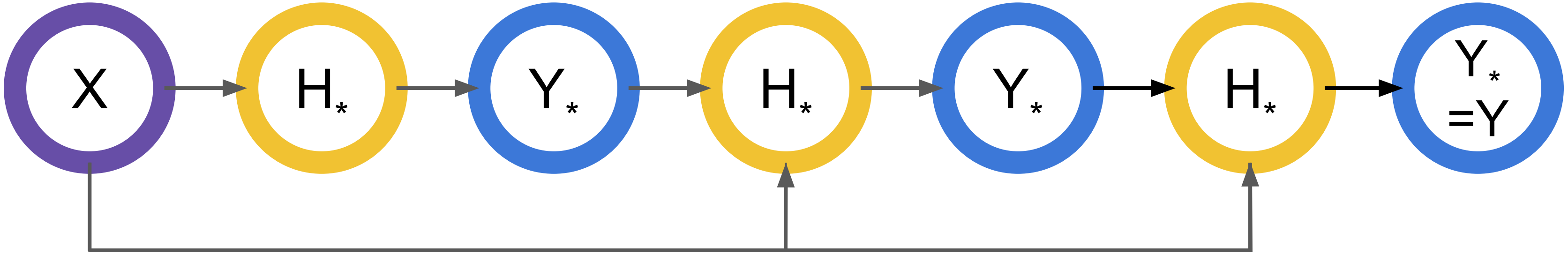
Unrolling the UNN

k=3

i=3

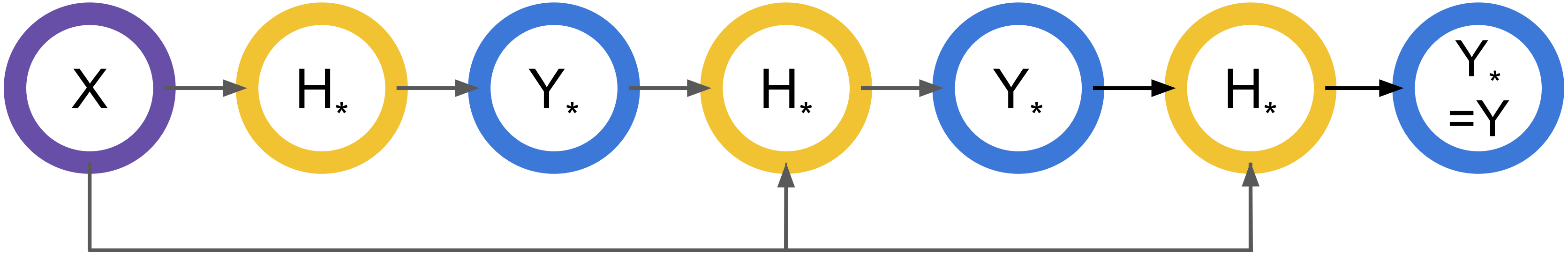


Unrolling the UNN



The unrolled computation ~ FFNN with skip connections and shared weights.

Unrolling the UNN



We can train the parameters effectively using standard gradient methods.

Experiments

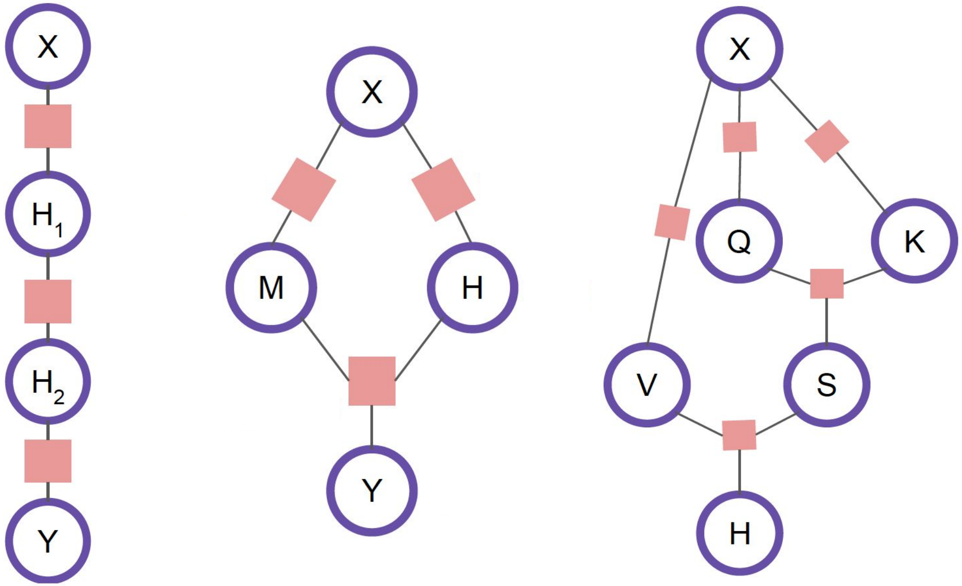


Image Classification and Visualization

- ➔ Convolutional hidden layers
- ➔ Forward direction: $y_*(x)$
- ➔ Backward direction: $x_*(y)$

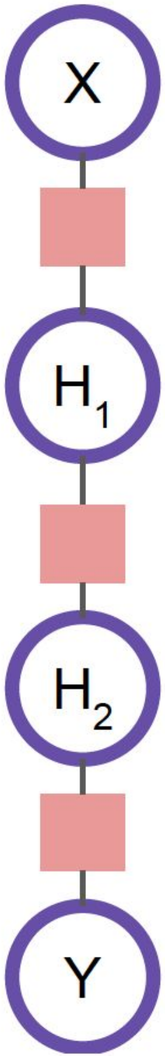
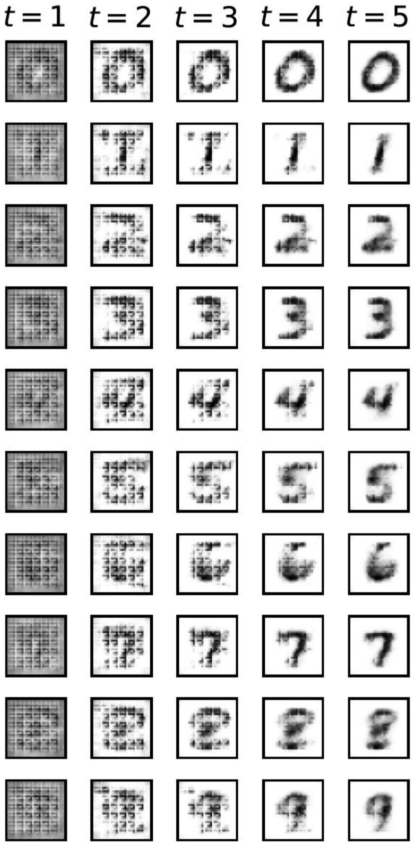


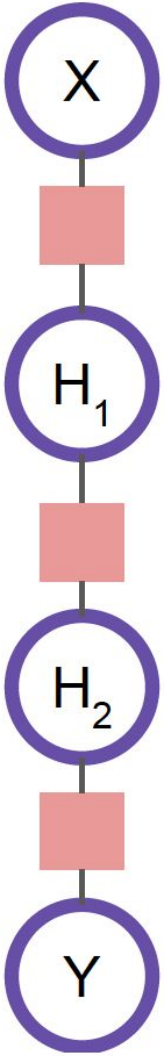
Image Classification and Visualization

➔ Convolutional hidden layers

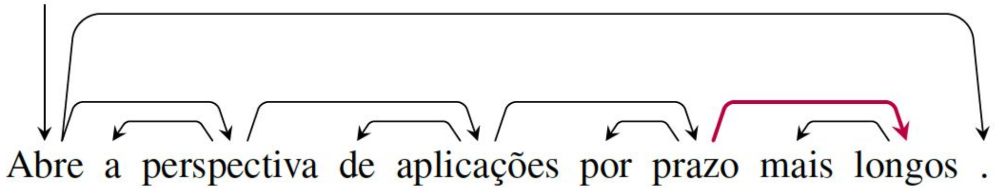
➔ Forward direction: $y_*(x)$

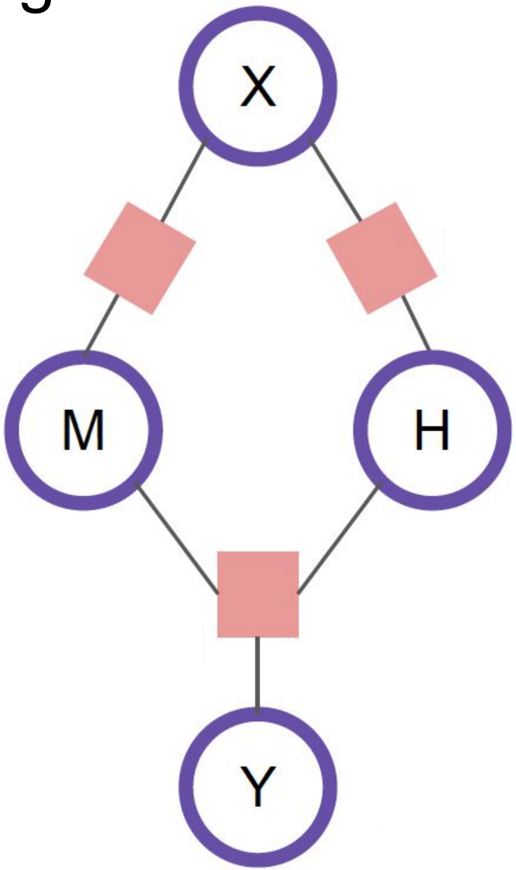
➔ Backward direction: $x_*(y)$





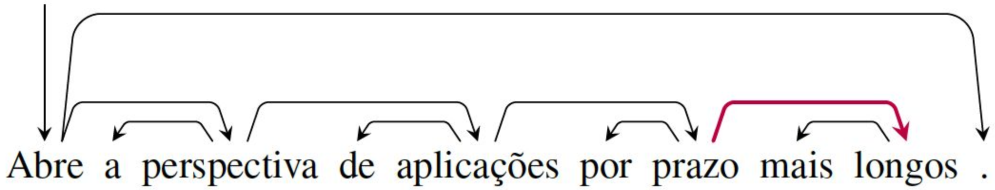
Structured UNNs for Dependency Parsing

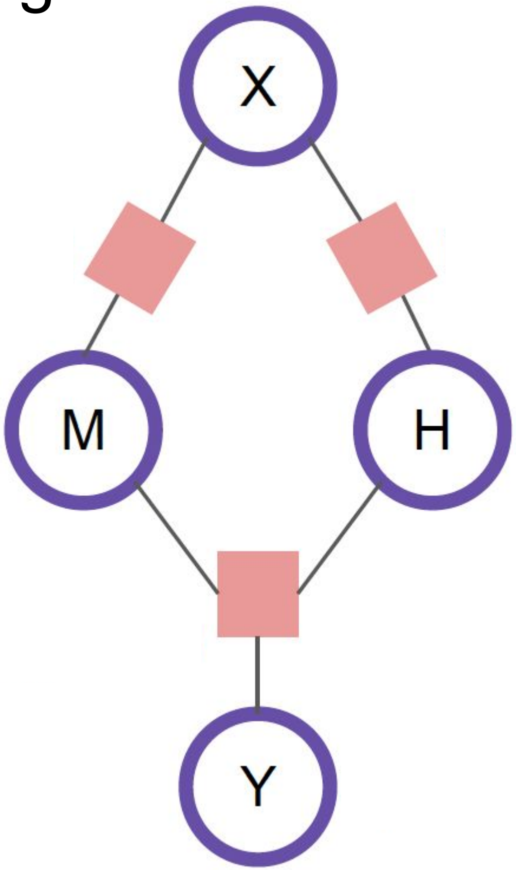




Structured UNNs for Dependency Parsing

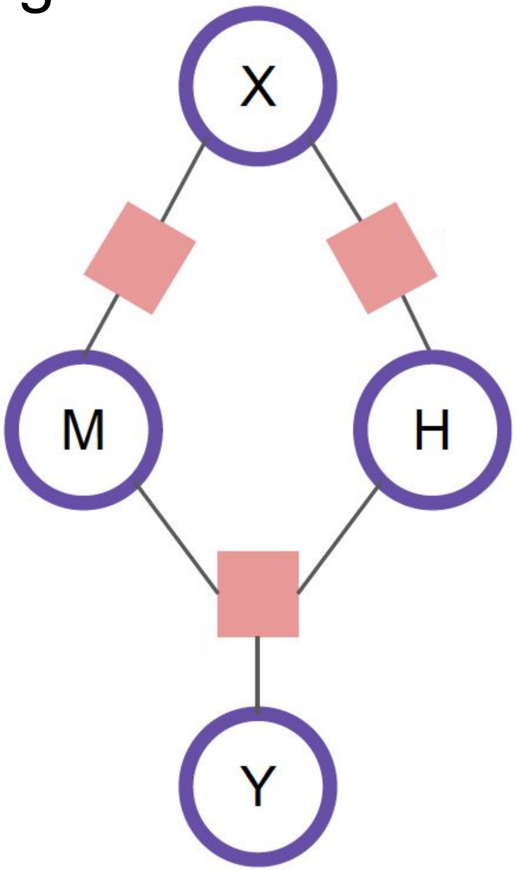
- ➔ Structured factors
- ➔ Higher-order factors





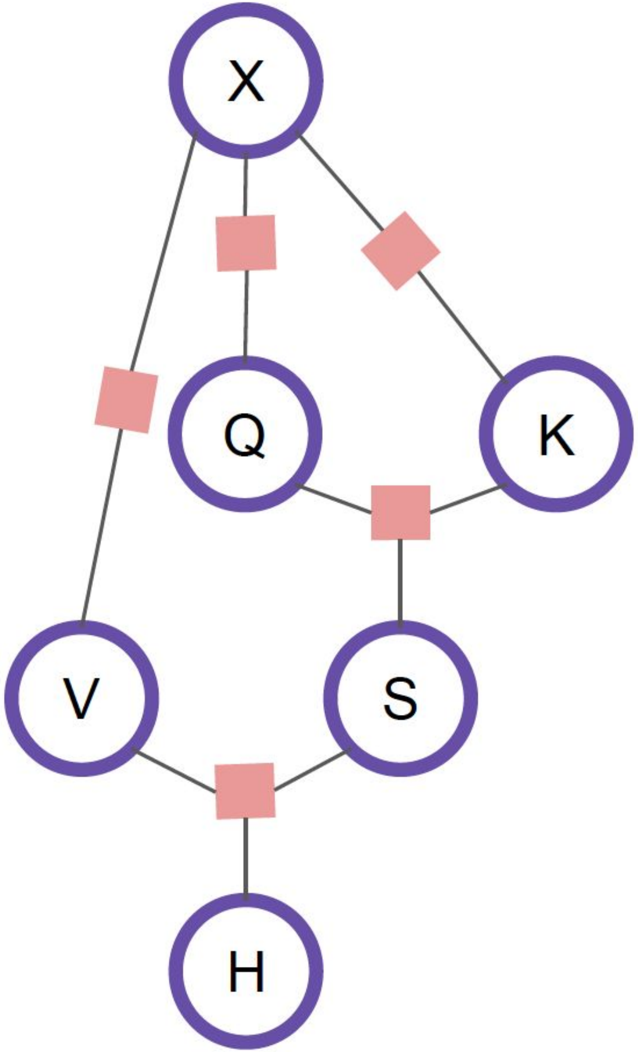
Structured UNNs for Dependency Parsing

Language	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
			Unlabeled attachment score		
CS	93.79	**93.83**	93.82	93.60	93.77
HU	85.11	**85.77**	84.47	85.13	84.09
TE	89.72	89.72	**90.00**	88.45	87.75
			Modifier list accuracy		
CS	84.46	84.82	**84.93**	84.12	84.49
HU	64.13	**66.07**	64.37	62.91	64.13
TE	72.87	72.87	**73.68**	66.80	65.99
			Exact match		
CS	59.17	60.76	**60.92**	59.42	59.84
HU	21.13	23.40	**24.15**	23.40	21.51
TE	75.69	77.08	**79.17**	71.53	70.14



Undirected Attention Mechanism

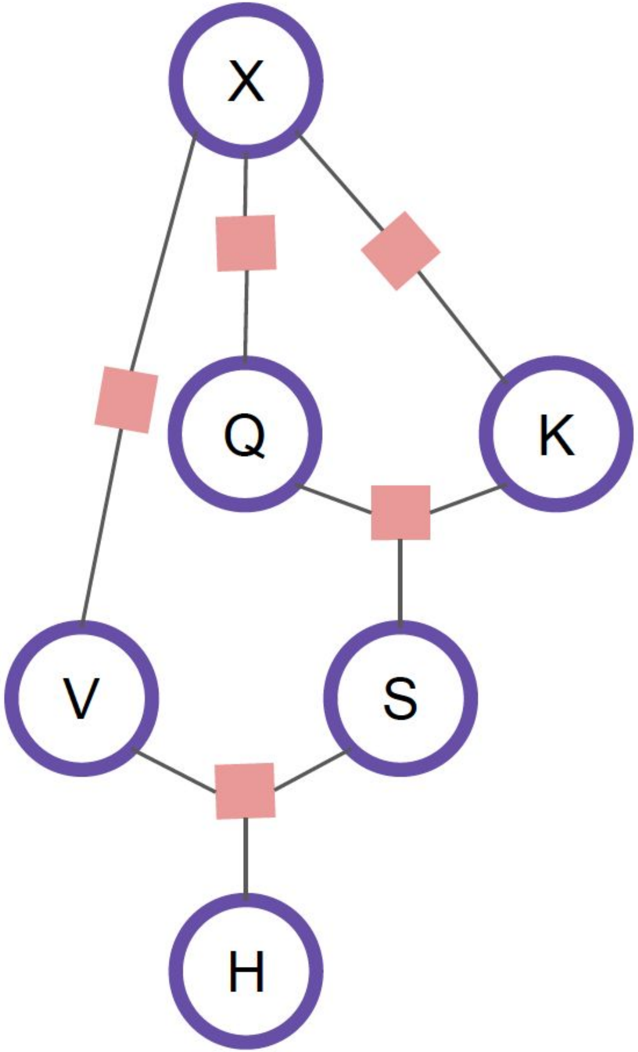
30 29 28 ? ? 25 24 23 22 21 20



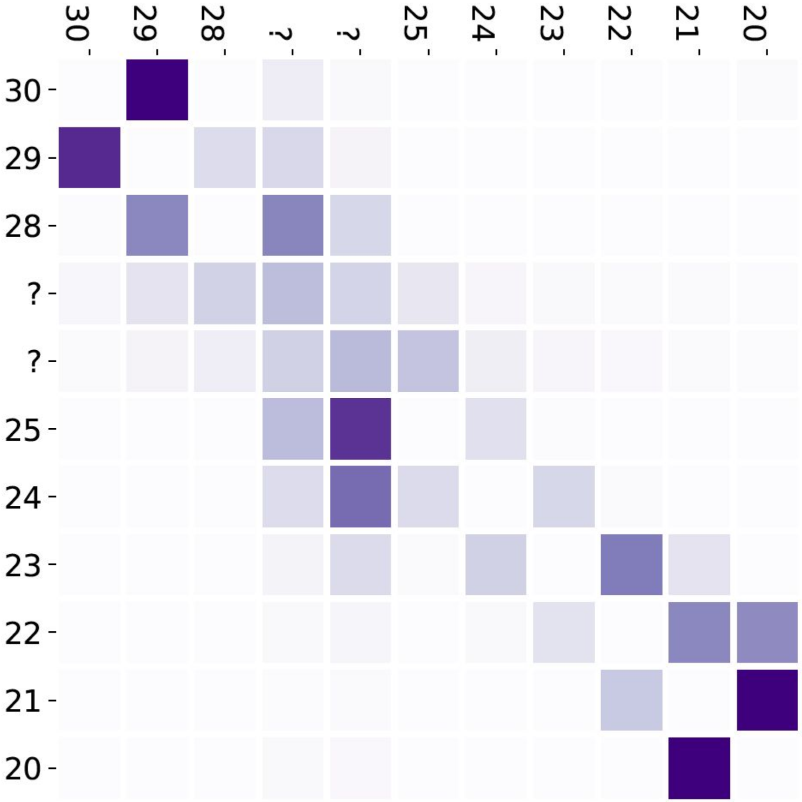
Undirected Attention Mechanism

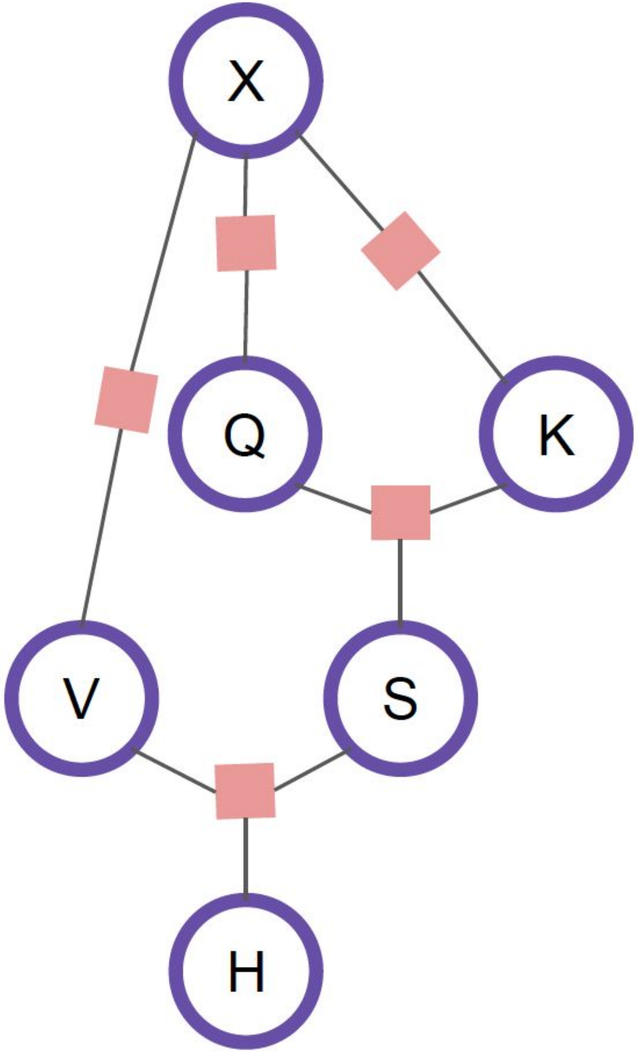
➔ Undirected, "auto-encoding" kind of attention mechanism

30 29 28 ? ? 25 24 23 22 21 20



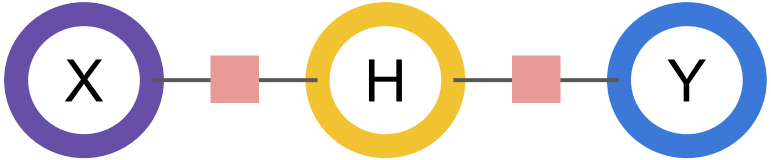
Undirected Attention Mechanism





In this work:

Undirected Neural Networks

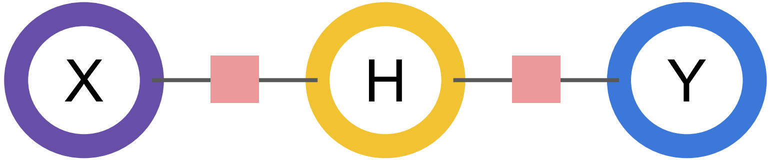


https://github.com/deep-spin/unn https://tsvm.github.io

In this work:

➔ Combine the representational strengths of factor graphs and of neural networks.

Undirected Neural Networks

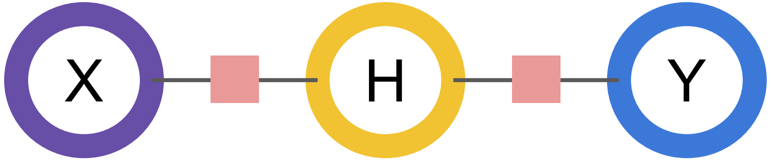


https://github.com/deep-spin/unn

https://tsvm.github.io

In this work:

Undirected Neural Networks

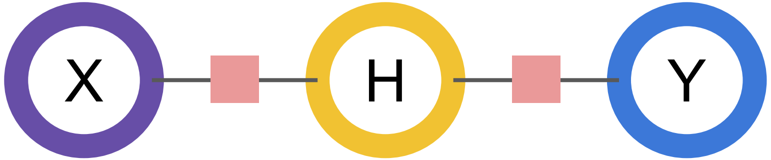


➔ Combine the representational strengths of factor graphs and of neural networks.
➔ Flexible framework for specifying computations that can be performed in any order.

https://github.com/deep-spin/unn   https://tsvm.github.io

In this work:

Undirected Neural Networks



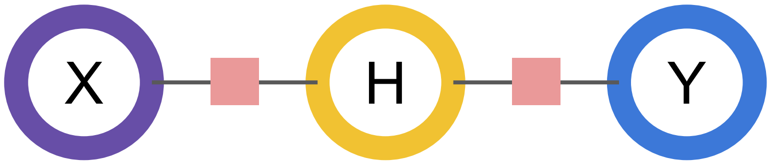
- ➔ Combine the representational strengths of factor graphs and of neural networks.
- ➔ Flexible framework for specifying computations that can be performed in any order.
- ➔ Unstructured and structured examples for three tasks.

https://github.com/deep-spin/unn

https://tsvm.github.io

In this work:

Undirected Neural Networks



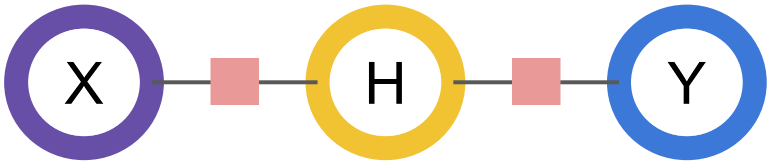
- ➔ Combine the representational strengths of factor graphs and of neural networks.
- ➔ Flexible framework for specifying computations that can be performed in any order.
- ➔ Unstructured and structured examples for three tasks.
- ➔ Subsume and extend many existing architectures: feed-forward, recurrent, self-attention networks, auto-encoders, and networks with implicit layers *(in the paper)*.

https://github.com/deep-spin/unn

https://tsvm.github.io

In this work:

Undirected Neural Networks



- ➔ Combine the representational strengths of factor graphs and of neural networks.
- ➔ Flexible framework for specifying computations that can be performed in any order.
- ➔ Unstructured and structured examples for three tasks.
- ➔ Subsume and extend many existing architectures: feed-forward, recurrent, self-attention networks, auto-encoders, and networks with implicit layers *(in the paper)*.

https://github.com/deep-spin/unn

https://tsvm.github.io