Structured Attention & Differentiable Dynamic Programming

Mathieu Blondel
Staff Research Scientist
NTT, Kyoto, Japan

January 4th, 2019

Outline

1. Structured attention

2. Differentiable dynamic programming

Outline

1. Stru

2. Differe ble dynamic programming

differentiable
max and argmax
operators!

Outline

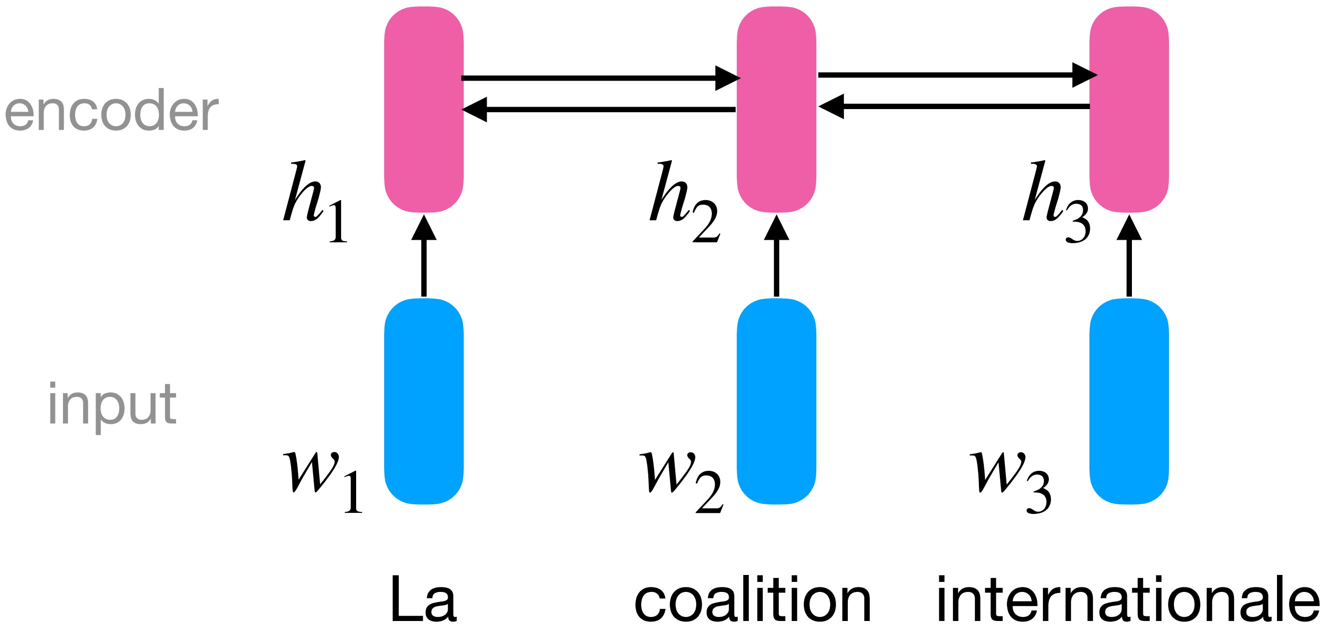
1. Structured attention

2. Differentiable dynamic programming

Sequence to sequence with attention

Bahdanau et al., ICLR, 2015

Sequence to sequence with attention

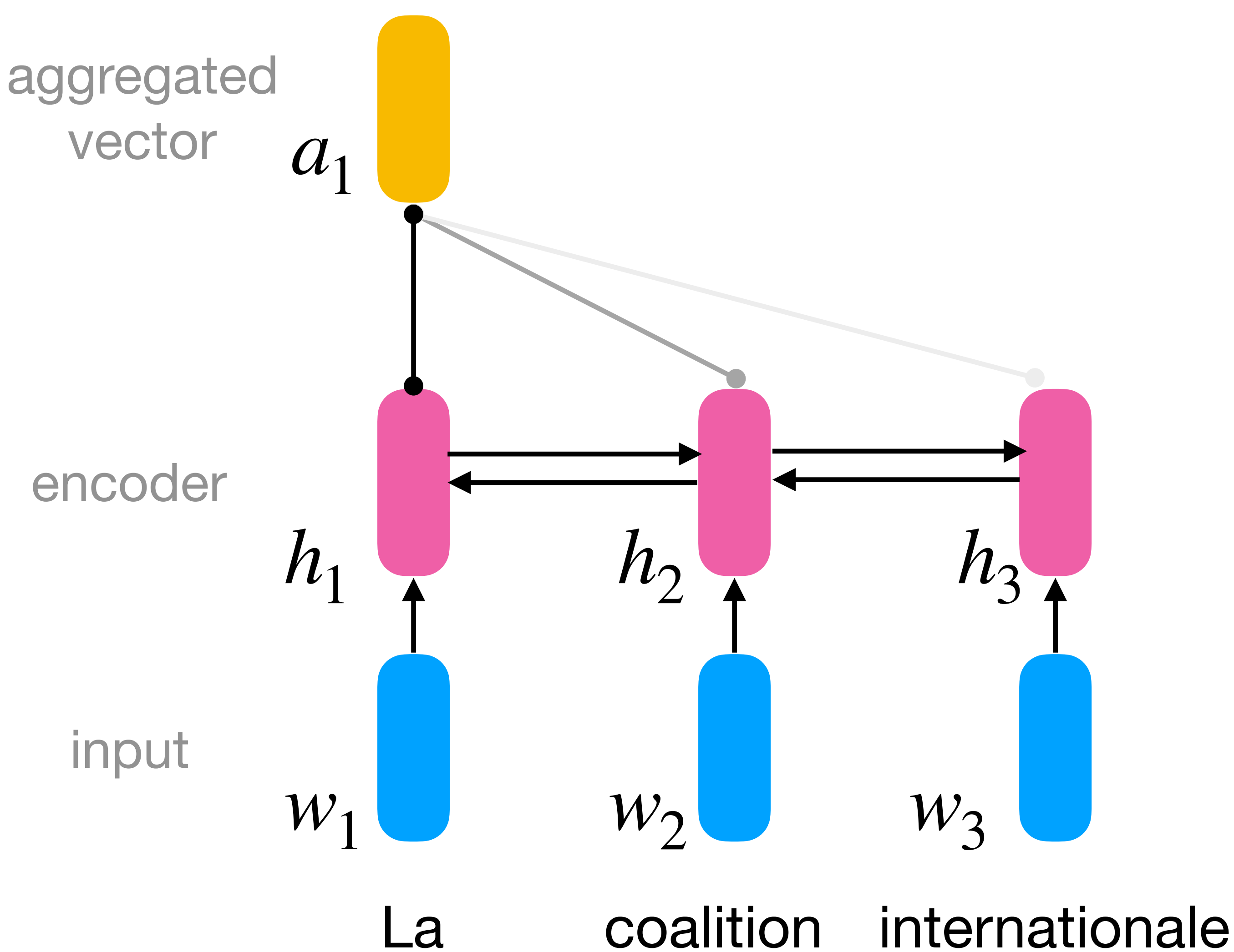


$$H = \text{encode}(W)$$

$$W = \text{lookup}(\text{words})$$

Bahdanau et al., ICLR, 2015

Sequence to sequence with attention



$\theta_t = Hq_{t-1}$ # attn scores

$p_t =$ **softmax**(θ_t) # attn proba

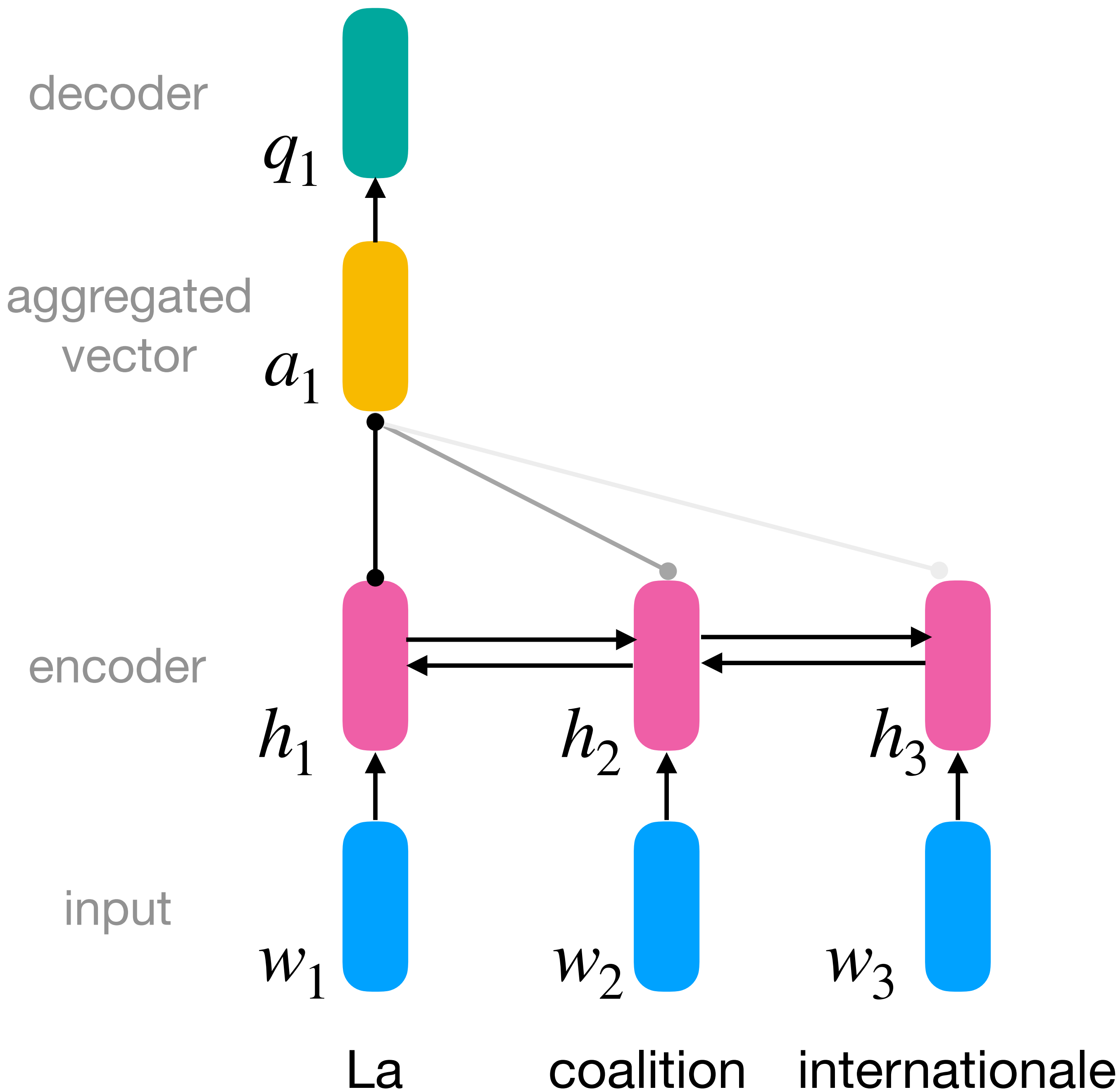
$a_t = p_t^\top H$ # aggregated vector

$H = \text{encode}(W)$

$W = \text{lookup}(\text{words})$

Bahdanau et al., ICLR, 2015

Sequence to sequence with attention



$$q_t = \text{decode}(a_t, y_{t-1}, q_{t-1})$$

$\theta_t = Hq_{t-1}$ # attn scores

$p_t = \textbf{softmax}(\theta_t)$ # attn proba

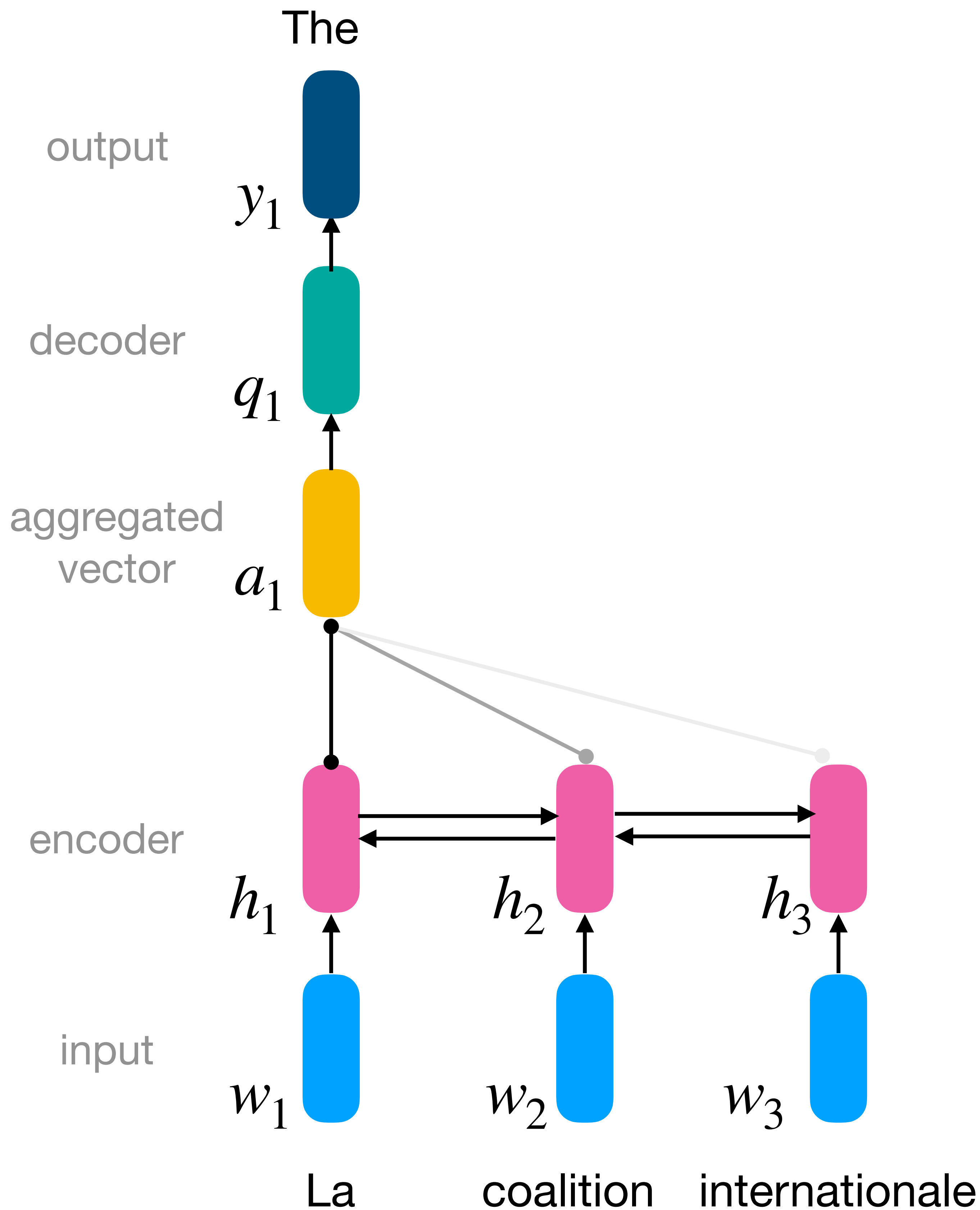
$a_t = p_t^\top H$ # aggregated vector

$$H = \text{encode}(W)$$

$$W = \text{lookup}(\text{words})$$

Bahdanau et al., ICLR, 2015

Sequence to sequence with attention



$$y_t = Mq_t + b$$

$$q_t = \text{decode}(a_t, y_{t-1}, q_{t-1})$$

$\theta_t = Hq_{t-1}$ # attn scores

$p_t = \textbf{softmax}(\theta_t)$ # attn proba

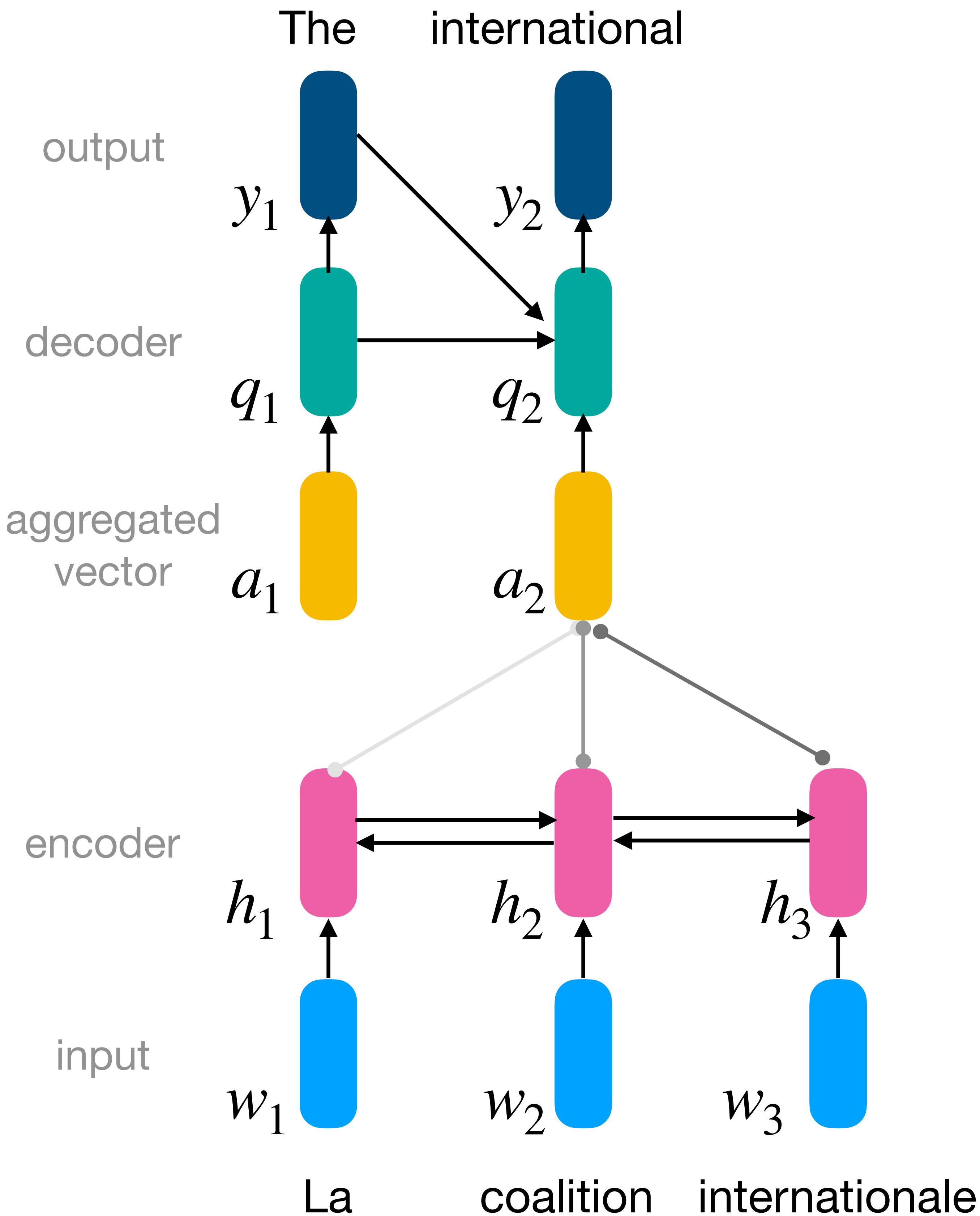
$a_t = p_t^\top H$ # aggregated vector

$$H = \text{encode}(W)$$

$$W = \text{lookup}(\text{words})$$

Bahdanau et al., ICLR, 2015

Sequence to sequence with attention



$$y_t = Mq_t + b$$

$$q_t = \text{decode}(a_t, y_{t-1}, q_{t-1})$$

$$\theta_t = Hq_{t-1}$$ # attn scores

$$p_t = \textbf{softmax}(\theta_t)$$ # attn proba

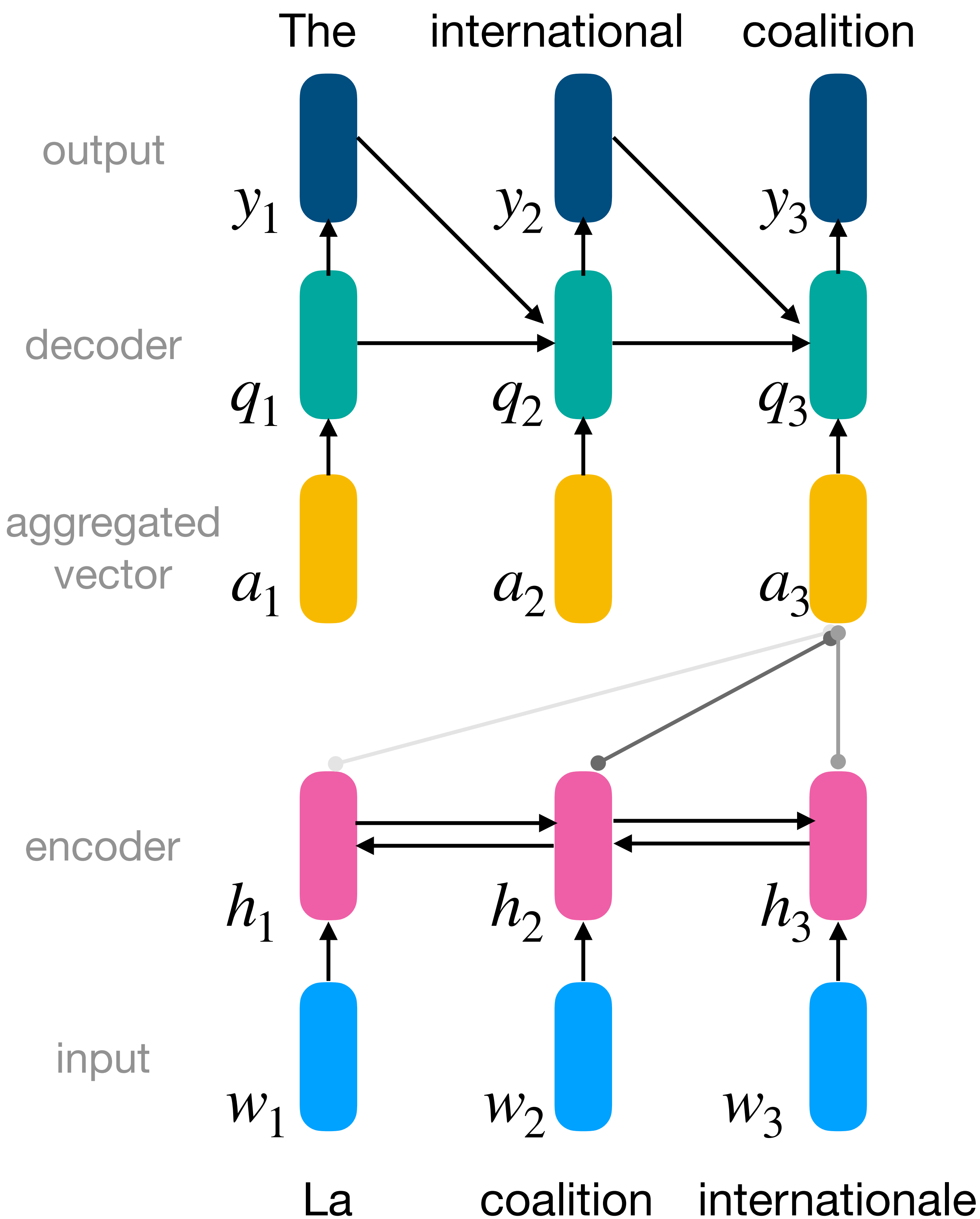
$$a_t = p_t^\top H$$ # aggregated vector

$$H = \text{encode}(W)$$

$$W = \text{lookup}(\text{words})$$

Bahdanau et al., ICLR, 2015

Sequence to sequence with attention



$$y_t = Mq_t + b$$

$$q_t = \text{decode}(a_t, y_{t-1}, q_{t-1})$$

$\theta_t = Hq_{t-1}$ # attn scores

$p_t = \textbf{softmax}(\theta_t)$ # attn proba

$a_t = p_t^\top H$ # aggregated vector

$$H = \text{encode}(W)$$

$$W = \text{lookup}(\text{words})$$

Bahdanau et al., ICLR, 2015

Softmax attention

$$\textbf{softmax}(\theta) \triangleq \frac{\exp(\theta)}{\sum_{i=1}^{m} \exp(\theta_i)}$$

Bahdanau et al., ICLR, 2015

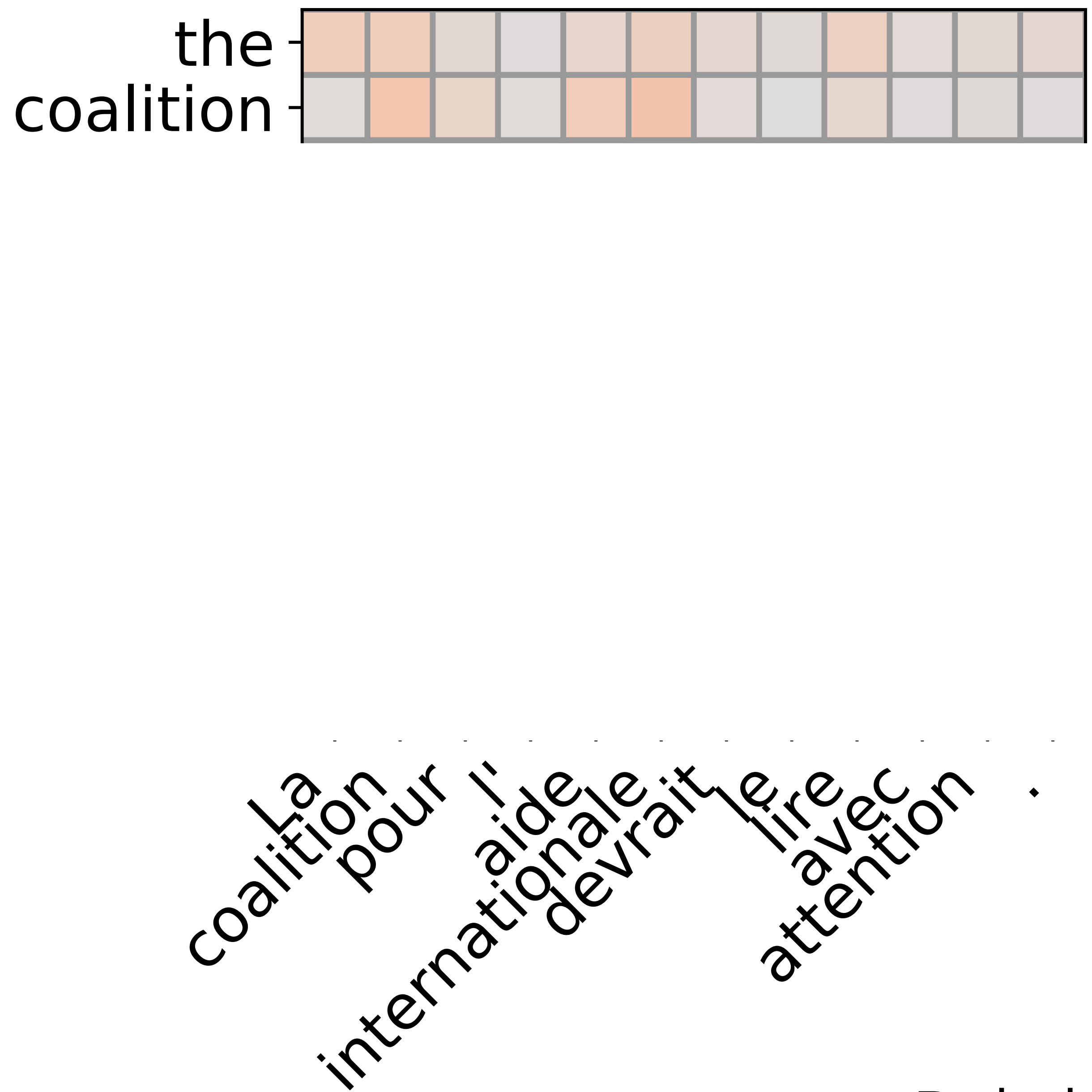
Softmax attention

$$\textbf{softmax}(\theta) \triangleq \frac{\exp(\theta)}{\sum_{i=1}^{m} \exp(\theta_i)}$$

Bahdanau et al., ICLR, 2015

Softmax attention

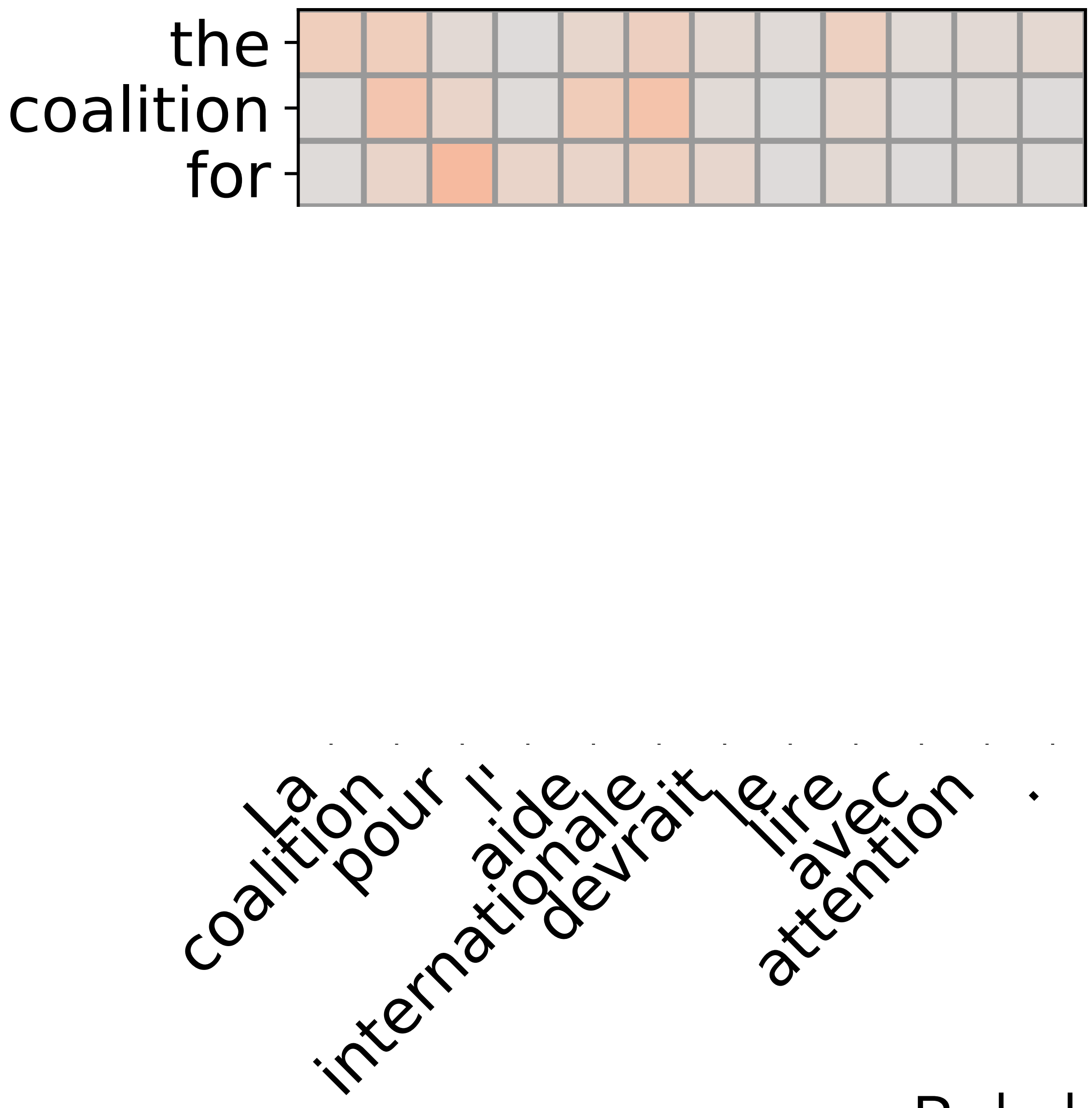
$$\textbf{softmax}(\theta) \triangleq \frac{\exp(\theta)}{\sum_{i=1}^{m} \exp(\theta_i)}$$



Bahdanau et al., ICLR, 2015

Softmax attention

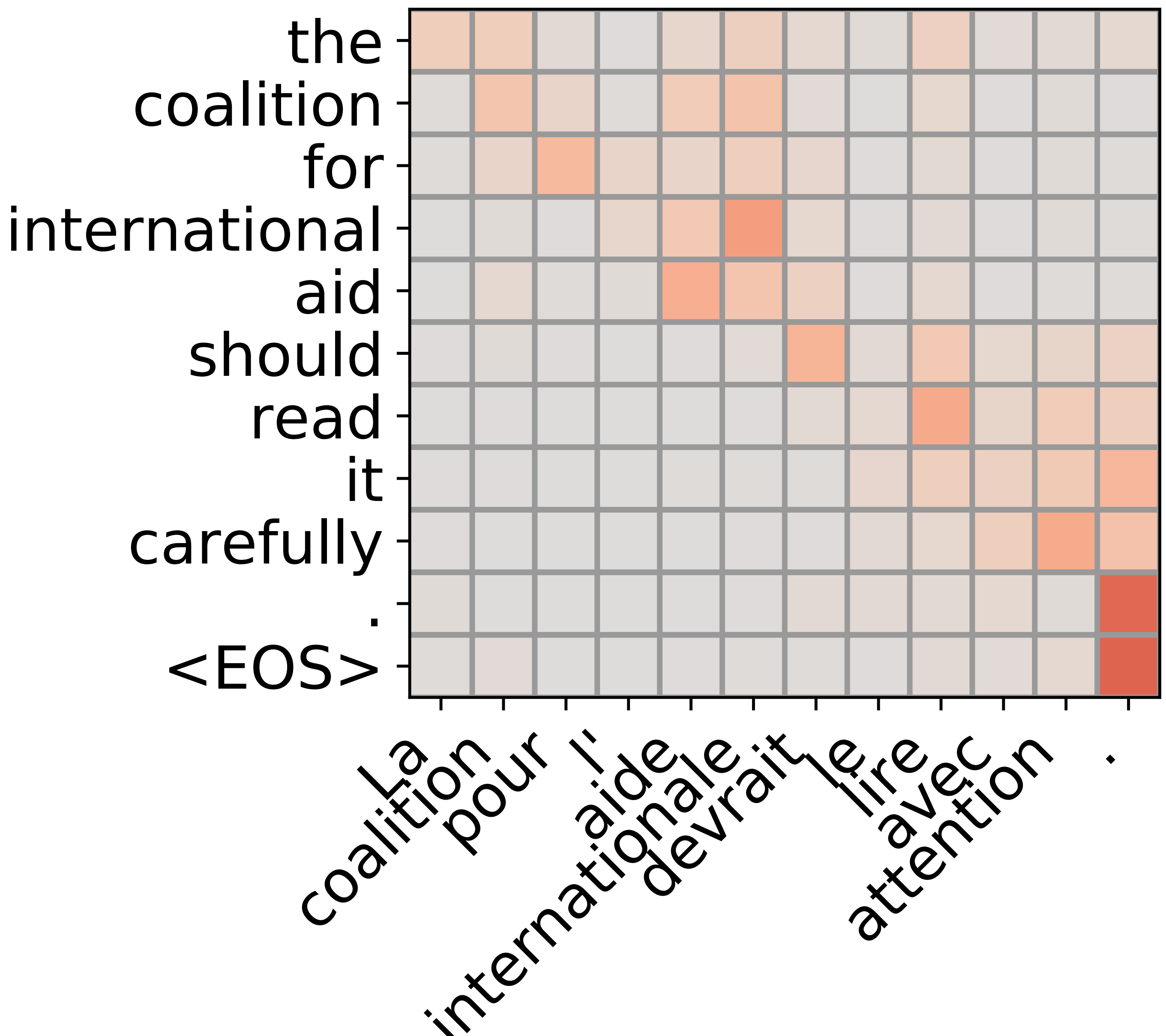
$$\textbf{softmax}(\theta) \triangleq \frac{\exp(\theta)}{\sum_{i=1}^{m} \exp(\theta_i)}$$



Bahdanau et al., ICLR, 2015

Softmax attention

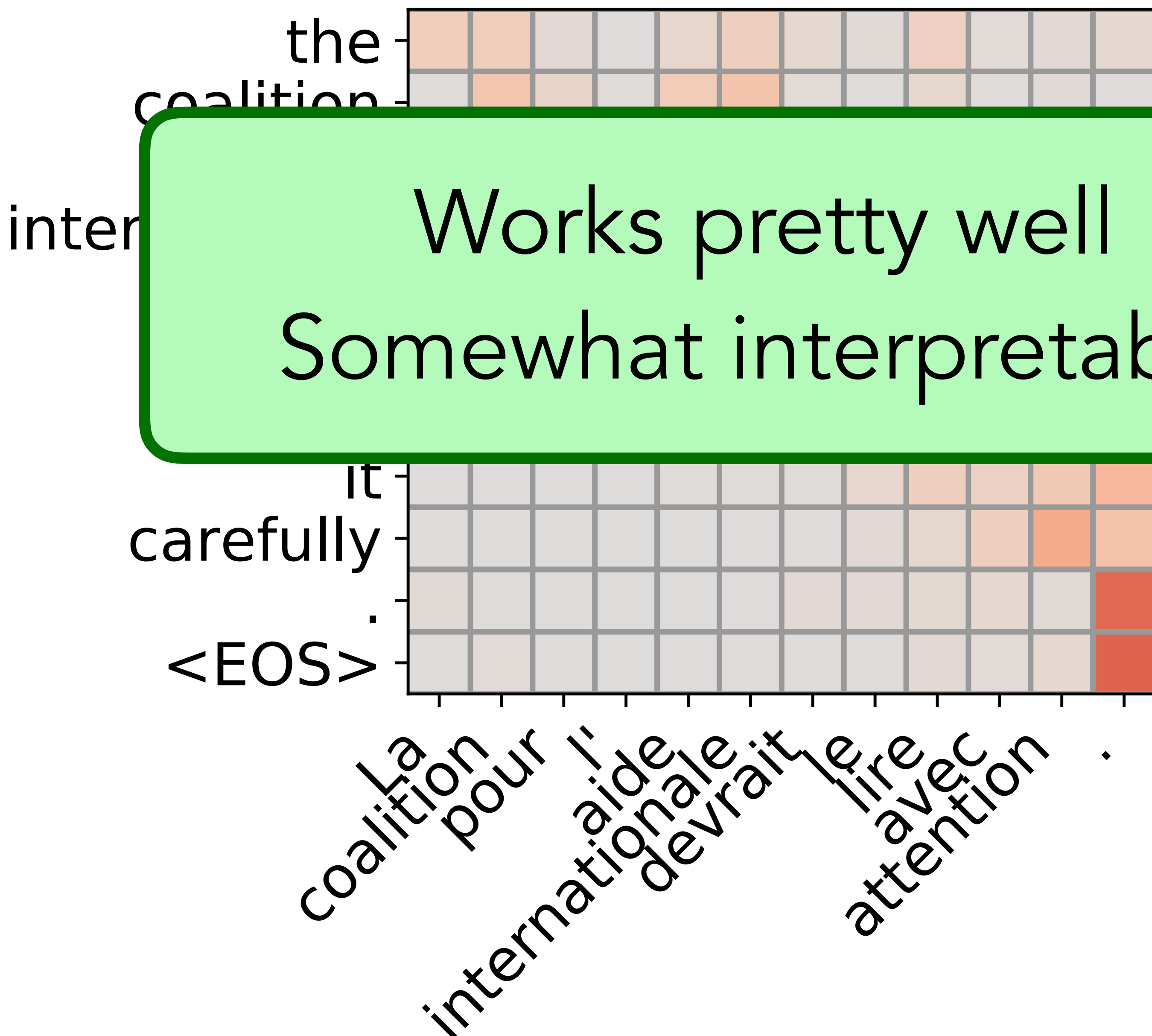
$$\textbf{softmax}(\theta) \triangleq \frac{\exp(\theta)}{\sum_{i=1}^{m} \exp(\theta_i)}$$



Bahdanau et al., ICLR, 2015

Softmax attention

$$\textbf{softmax}(\theta) \triangleq \frac{\exp(\theta)}{\sum_{i=1}^{m} \exp(\theta_i)}$$

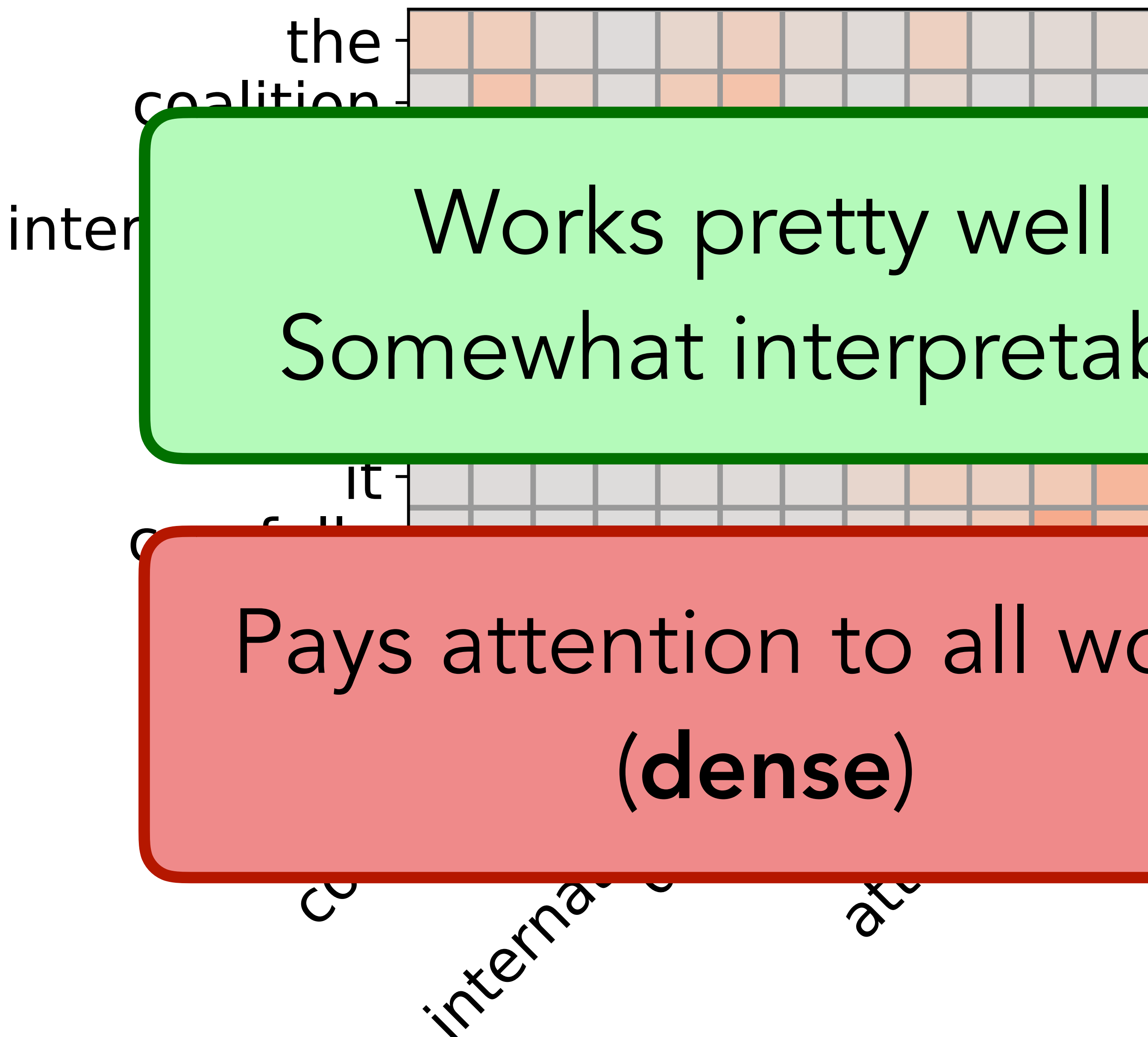


Works pretty well
Somewhat interpretable

Bahdanau et al., ICLR, 2015

Softmax attention

$$\textbf{softmax}(\theta) \triangleq \frac{\exp(\theta)}{\sum_{i=1}^{m} \exp(\theta_i)}$$



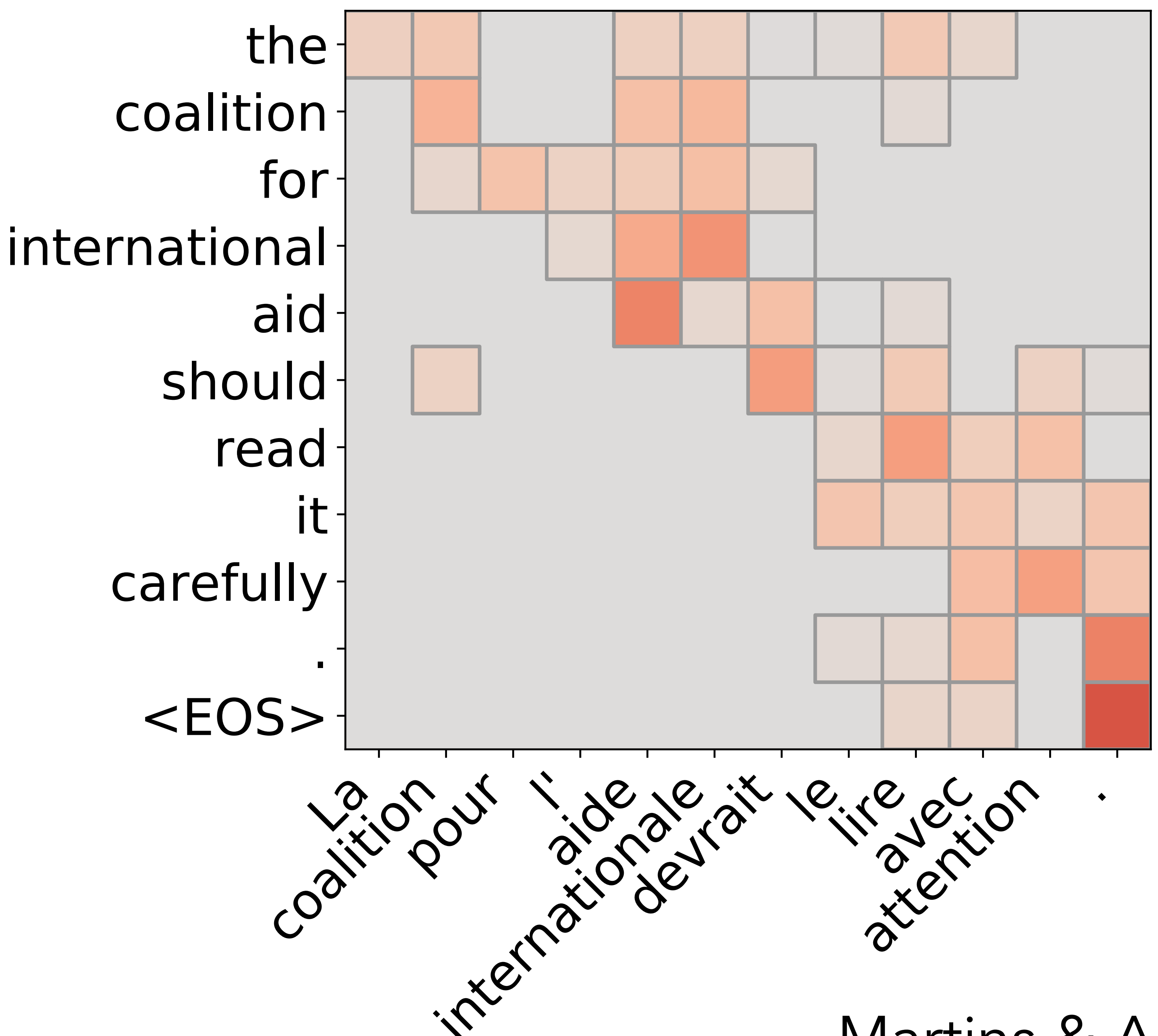
Works pretty well
Somewhat interpretable

Pays attention to all words (**dense**)

Bahdanau et al., ICLR, 2015

Sparsemax attention

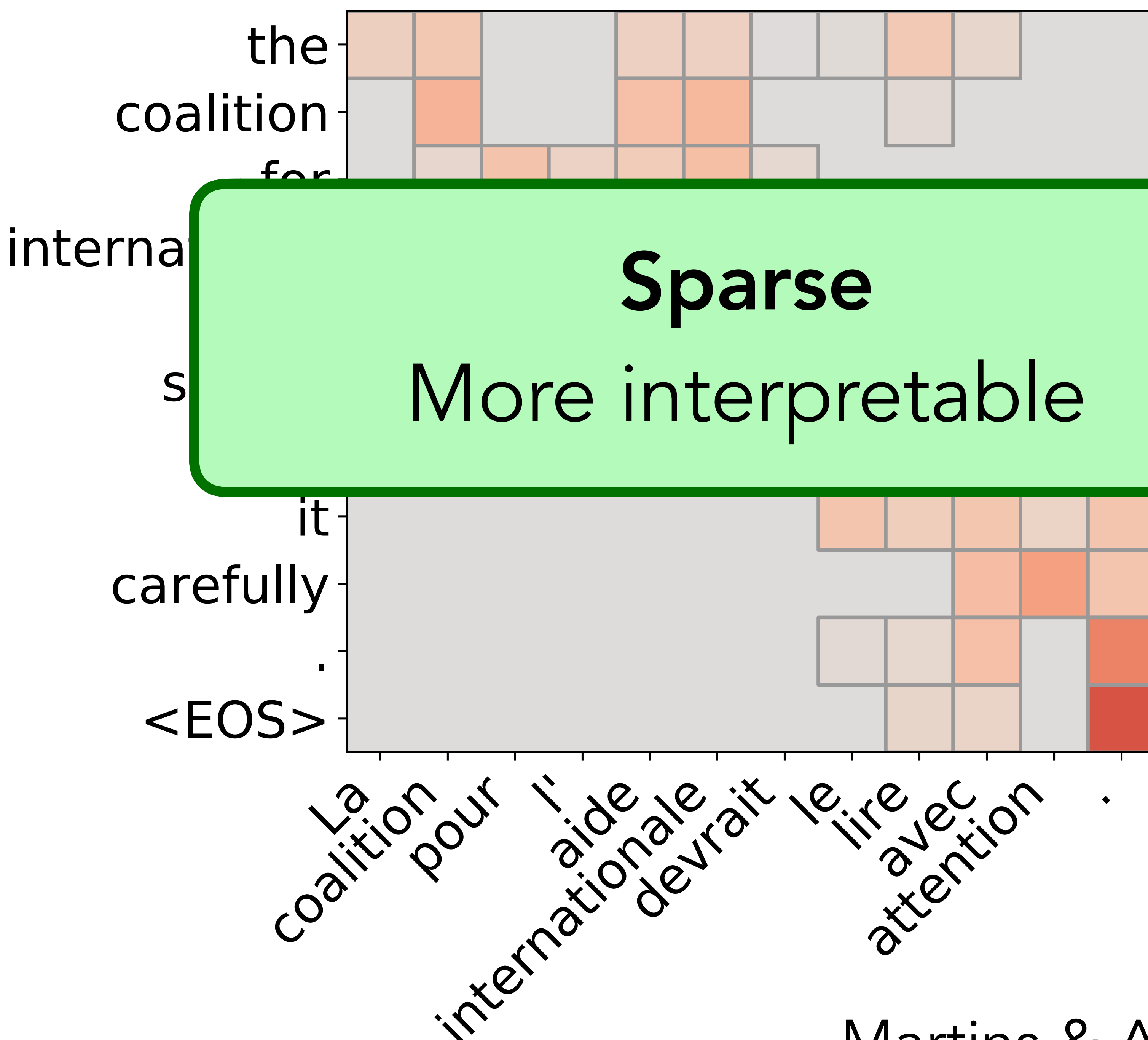
$$\textbf{sparsemax}(\theta) \triangleq \arg\min_{p \in \triangle^m} \|p - \theta\|^2$$



Martins & Astudillo, ICML, 2016

Sparsemax attention

$$\textbf{sparsemax}(\theta) \triangleq \arg\min_{p \in \triangle^m} \|p - \theta\|^2$$



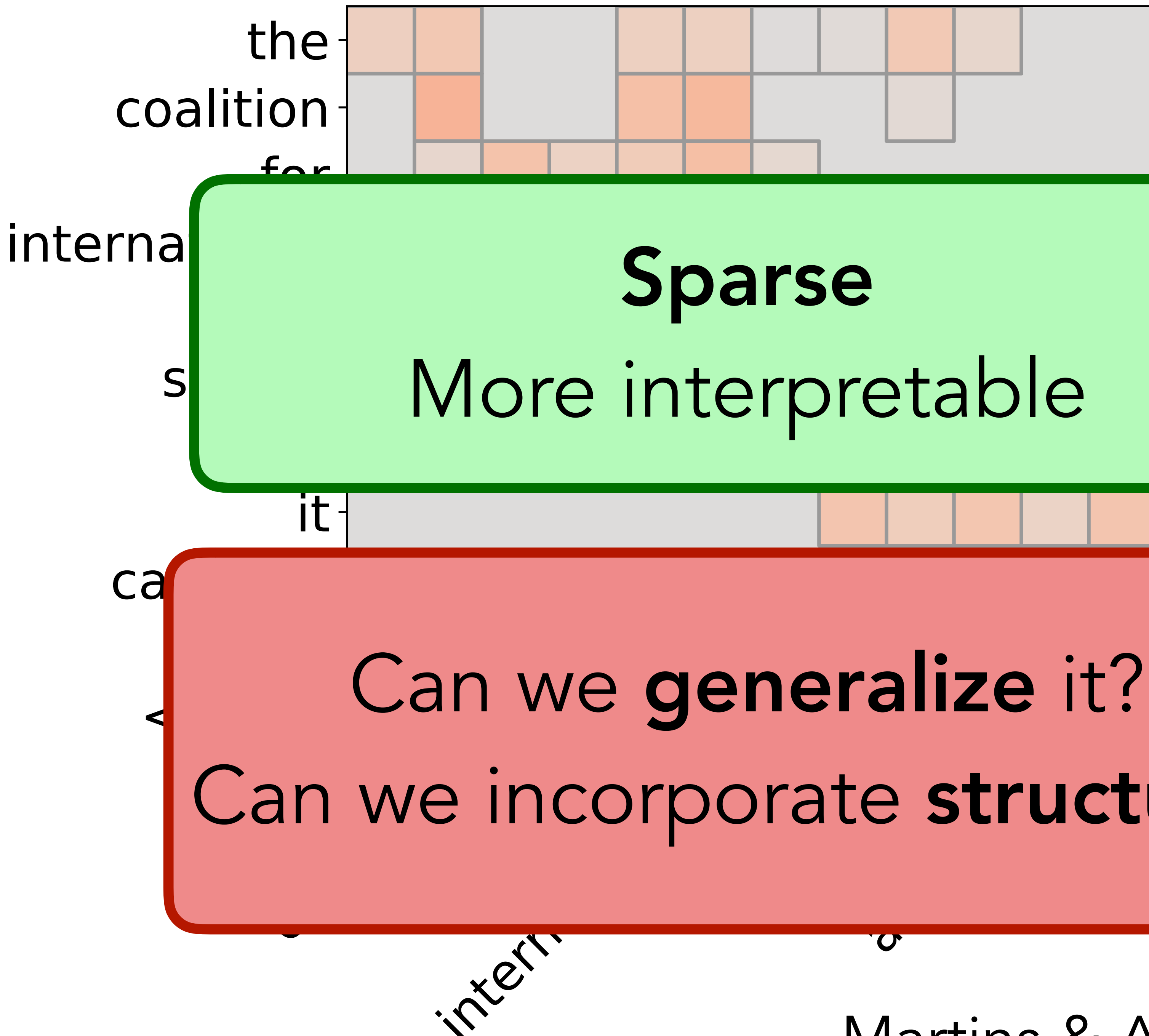
Sparse

More interpretable

Martins & Astudillo, ICML, 2016

Sparsemax attention

$$\textbf{sparsemax}(\theta) \triangleq \arg\min_{p \in \triangle^m} \|p - \theta\|^2$$



Sparse

More interpretable

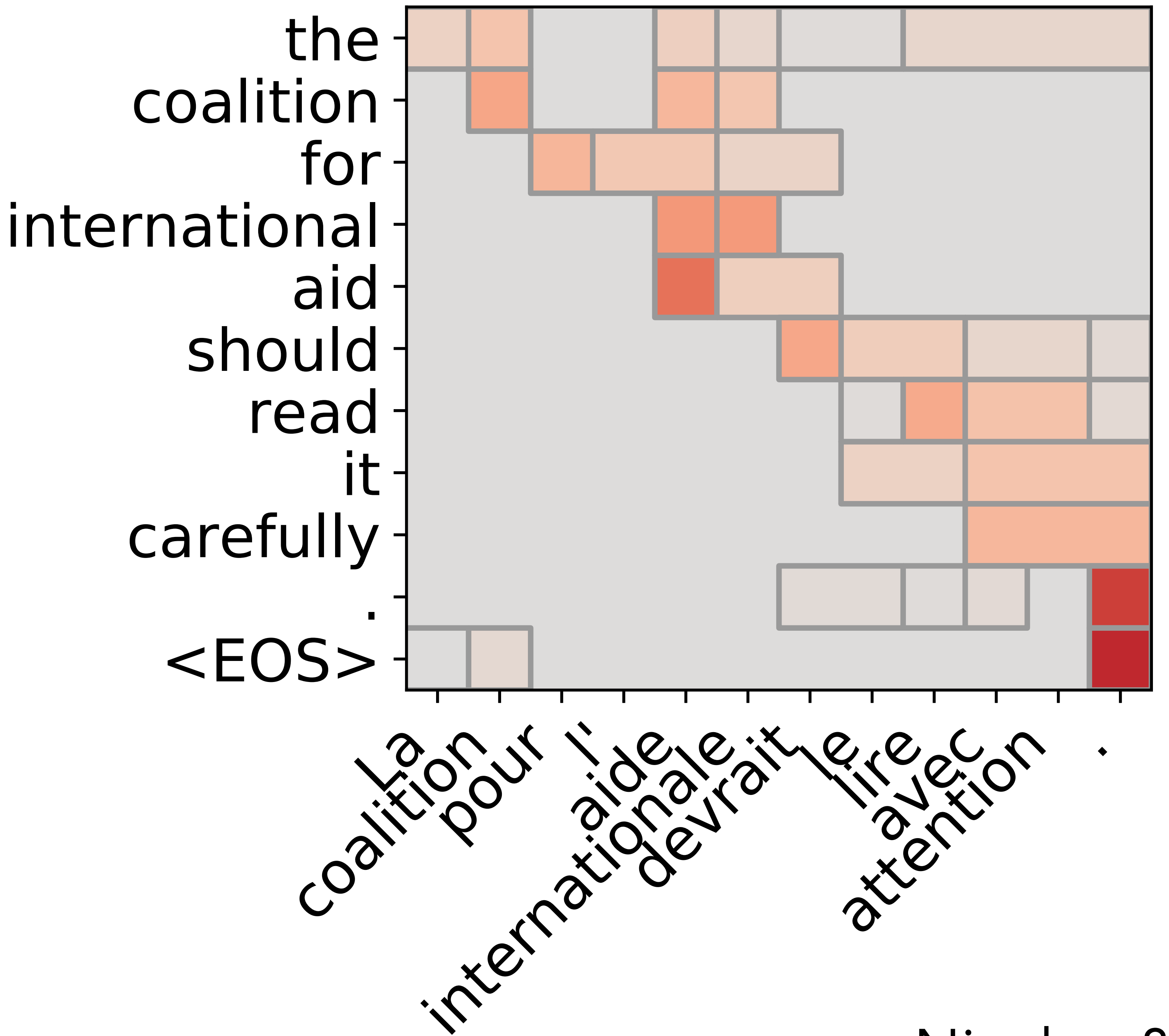
Can we **generalize** it?

Can we incorporate **structure**?

Martins & Astudillo, ICML, 2016

Fusedmax attention (proposed)

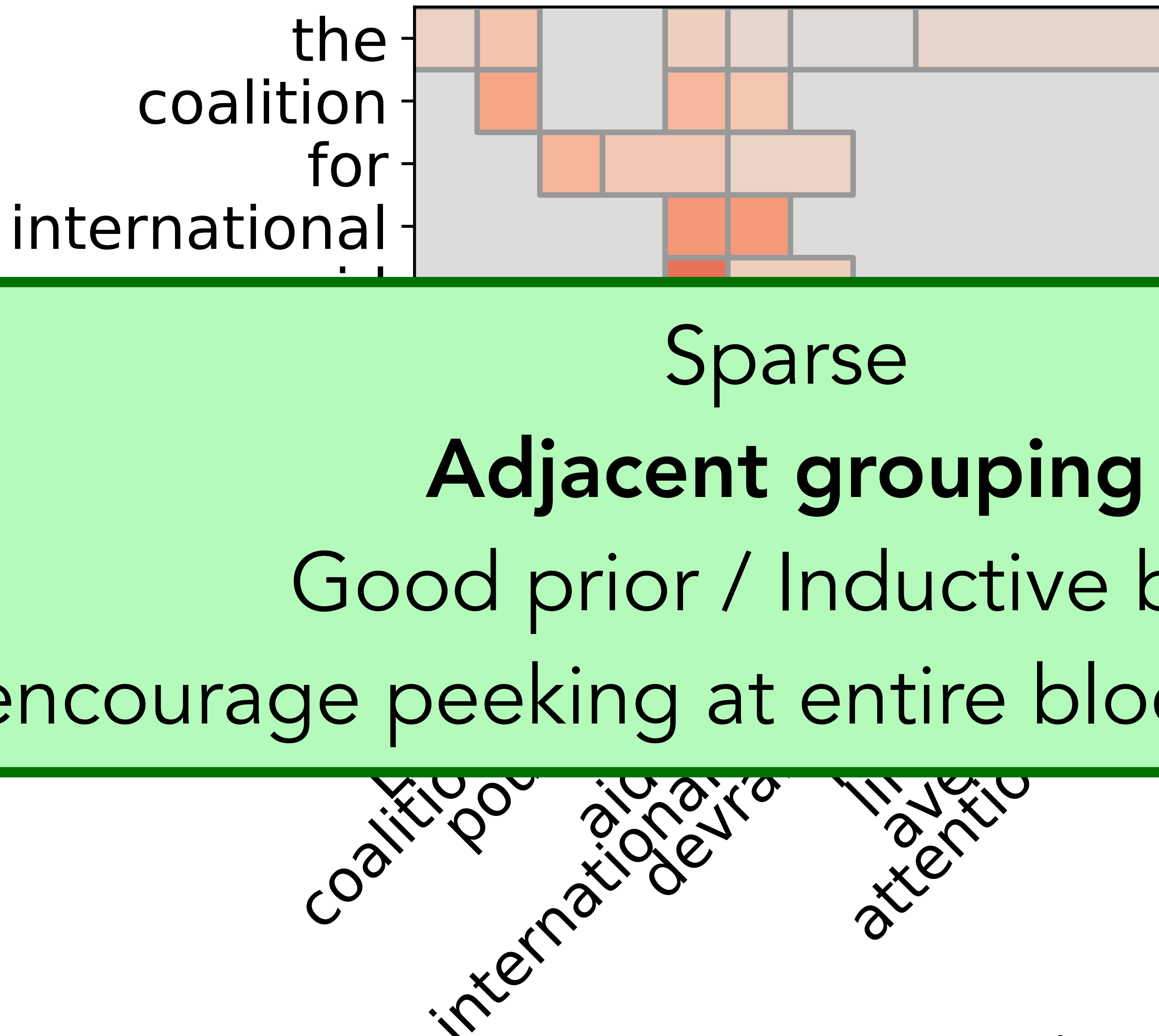
$$\textbf{fusedmax}(\theta) \triangleq \textbf{???}$$



Niculae & Blondel, NIPS 2017

Fusedmax attention (proposed)

$$\textbf{fusedmax}(\theta) \triangleq \textbf{???}$$



Sparse

Adjacent grouping

Good prior / Inductive bias

(encourage peeking at entire blocks of words)

Niculae & Blondel, NIPS 2017

Our contributions

Our contributions

- A principled framework for **differentiable argmax** operators

Niculae & Blondel, NIPS 2017

Our contributions

- A principled framework for **differentiable argmax** operators
  - Recovers softmax and sparsemax as special cases

Niculae & Blondel, NIPS 2017

Our contributions

- A principled framework for **differentiable argmax** operators
    - Recovers softmax and sparsemax as special cases
    - Enables to construct new operators easily

Niculae & Blondel, NIPS 2017

Our contributions

- A principled framework for **differentiable argmax** operators
  - Recovers softmax and sparsemax as special cases
  - Enables to construct new operators easily
- Efficient **forward** and **backward** computations for **fusedmax**

Niculae & Blondel, NIPS 2017

Our contributions

- A principled framework for **differentiable argmax** operators
  - Recovers softmax and sparsemax as special cases
  - Enables to construct new operators easily
- Efficient **forward** and **backward** computations for **fusedmax**
- Extensive experiments on NMT and sentence summarization

Niculae & Blondel, NIPS 2017

From argmax to softmax

From argmax to softmax

$$i^\star \in \arg\max_{i \in [m]} \theta_i$$

From argmax to softmax

$$\textbf{argmax}(\theta) \triangleq e_{i^\star} \qquad i^\star \in \arg\max_{i \in [m]} \theta_i$$

One-hot representation of integer argmax

From argmax to softmax

Function from
$\mathbb{R}^m$ to $\{e_1, \ldots, e_m\}$

$$\textbf{argmax}(\theta) \triangleq e_{i^\star} \qquad i^\star \in \arg\max_{i \in [m]} \theta_i$$

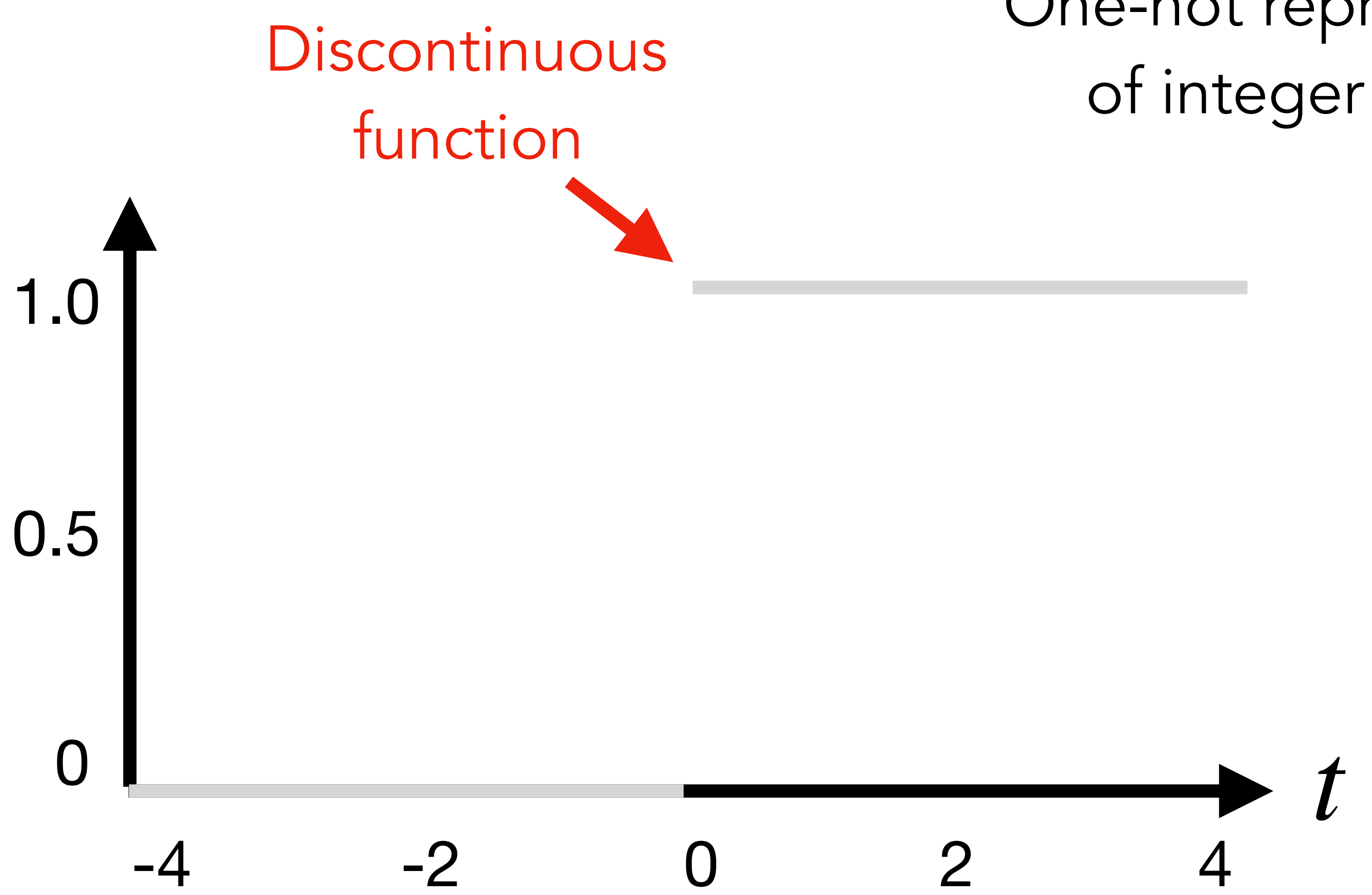
One-hot representation
of integer argmax

From argmax to softmax

Function from $\mathbb{R}^m$ to $\{e_1, \ldots, e_m\}$

$$\textbf{argmax}(\theta) \triangleq e_{i^\star} \qquad i^\star \in \arg\max_{i \in [m]} \theta_i$$

One-hot representation of integer argmax



$\textbf{argmax}([t, 0])_1$

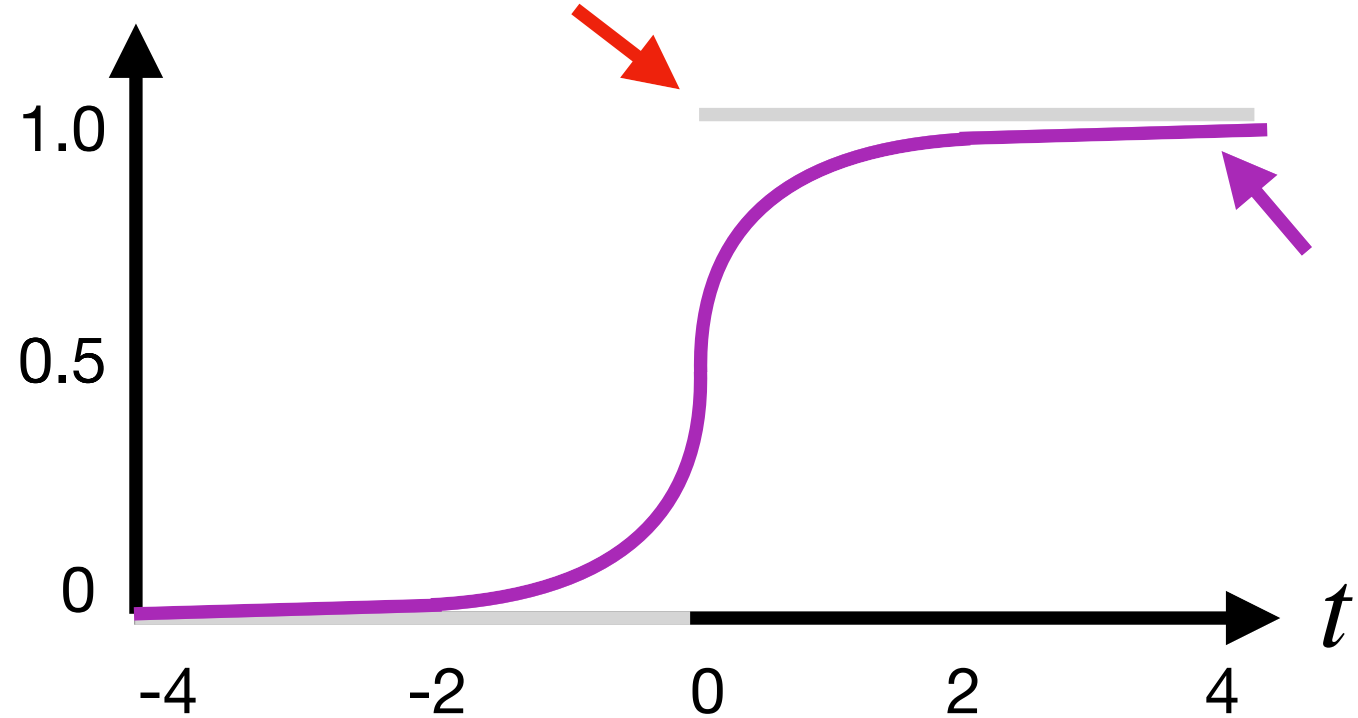
From argmax to softmax

Function from $\mathbb{R}^m$ to $\{e_1, \ldots, e_m\}$

$$\textbf{argmax}(\theta) \triangleq e_{i^\star} \qquad i^\star \in \arg\max_{i \in [m]} \theta_i$$

One-hot representation of integer argmax

Discontinuous function



usual sigmoid function when m = 2

$\textbf{argmax}([t,0])_1$ $\textbf{softmax}([t,0])_1$

Should really be called soft *argmax*

From argmax to softmax

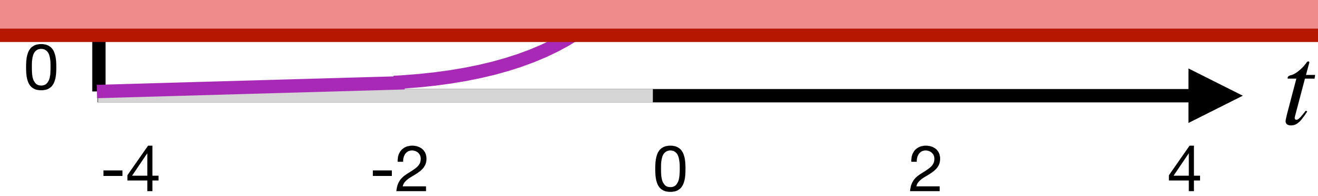
Function from $\mathbb{R}^m$ to $\{e_1, \ldots, e_m\}$

$$\textbf{argmax}(\theta) \triangleq e_{i^\star} \qquad i^\star \in \arg\max_{i \in [m]} \theta_i$$

One-hot representation

Where does the softmax come from?

Can we generalize it?



$\textbf{argmax}([t,0])_1$ $\textbf{softmax}([t,0])_1$

Should really be called soft *argmax*

Differentiable argmax: a variational perspective

Differentiable argmax: a variational perspective

$$\mathbf{argmax}(\theta) = \underset{p \in \{e_1, \ldots, e_m\}}{\arg\max} \; \langle p, \theta \rangle$$

Differentiable argmax: a variational perspective

$$\textbf{argmax}(\theta) = \underset{p \in \{e_1, \ldots, e_m\}}{\arg\max} \langle p, \theta \rangle$$

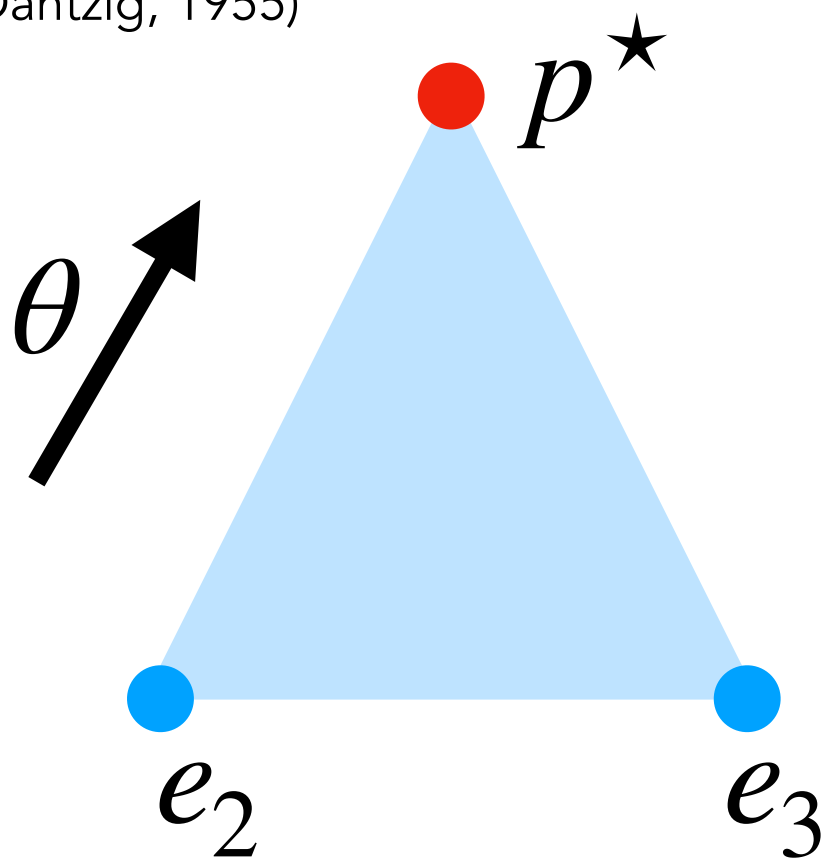
$$= \underset{p \in \triangle^m}{\arg\max} \langle p, \theta \rangle$$

Differentiable argmax: a variational perspective

$$\textbf{argmax}(\theta) = \underset{p \in \{e_1, \ldots, e_m\}}{\arg\max} \langle p, \theta \rangle$$

$$= \underset{p \in \triangle^m}{\arg\max} \langle p, \theta \rangle$$

Fundamental theorem
of linear programming
(Dantzig, 1955)



unregularized (Ω=0)

Differentiable argmax: a variational perspective

Introduce regularization

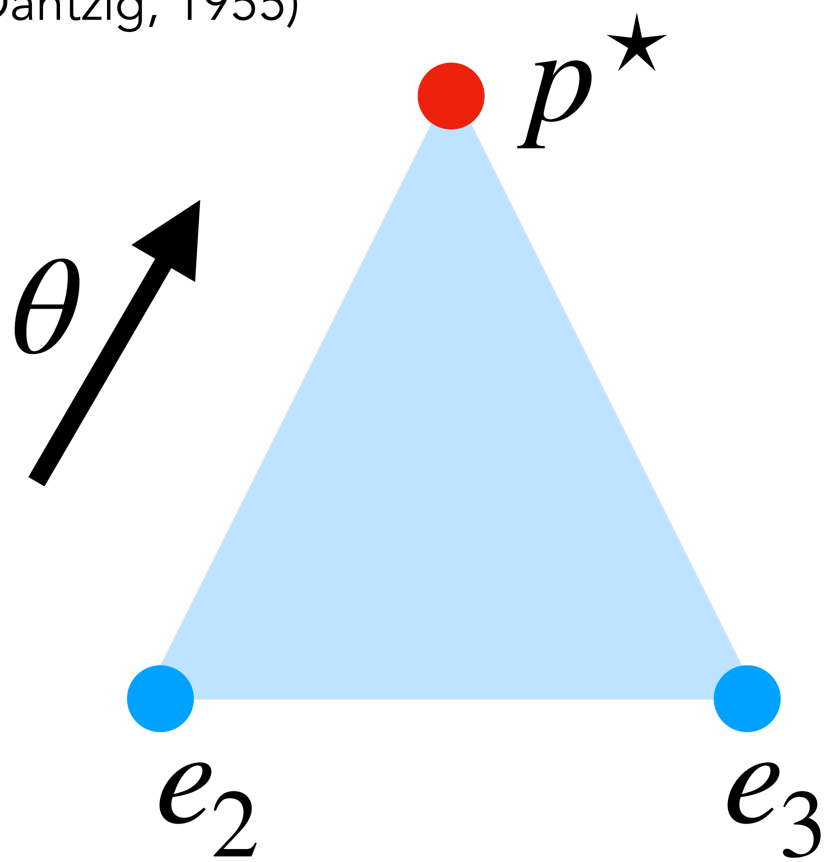
$$\mathbf{argmax}(\theta) = \underset{p \in \{e_1, \ldots, e_m\}}{\arg\max} \langle p, \theta \rangle$$

$$= \underset{p \in \triangle^m}{\arg\max} \langle p, \theta \rangle$$

Fundamental theorem of linear programming (Dantzig, 1955)

$$\mathbf{argmax}_{\Omega}(\theta) \triangleq \underset{p \in \triangle^m}{\arg\max} \langle p, \theta \rangle - \Omega(p)$$

Strongly-convex regularizer



unregularized (Ω=0)

Differentiable argmax: a variational perspective

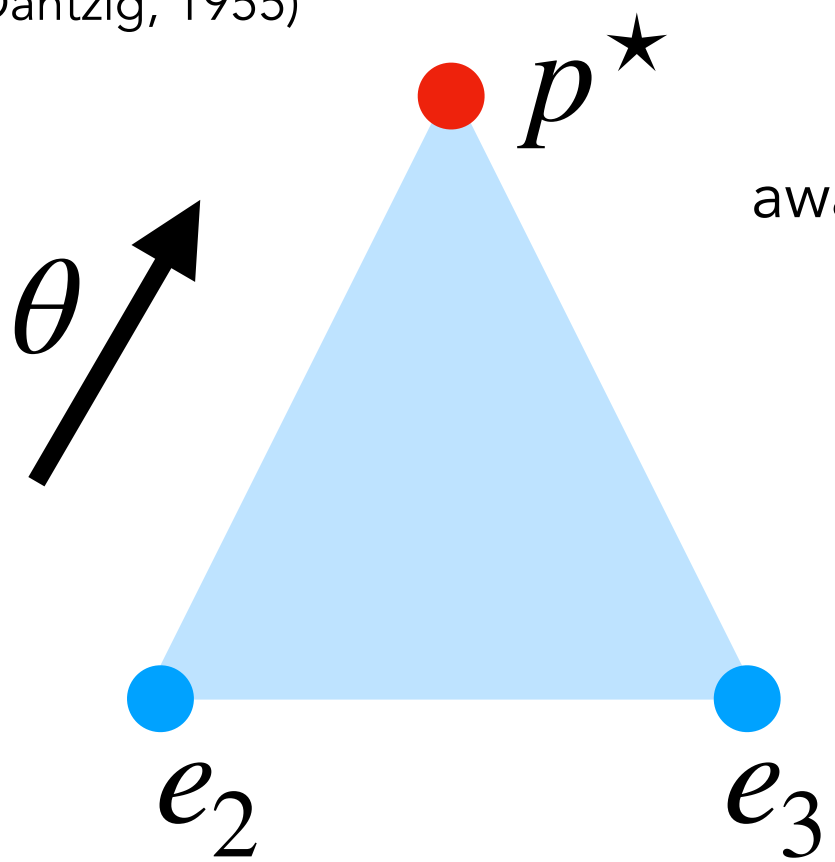
Introduce regularization

$$\textbf{argmax}(\theta) = \underset{p \in \{e_1, \ldots, e_m\}}{\arg\max} \; \langle p, \theta \rangle \qquad \textbf{argmax}_{\Omega}(\theta) \triangleq \underset{p \in \triangle^m}{\arg\max} \langle p, \theta \rangle - \Omega(p)$$

$$= \underset{p \in \triangle^m}{\arg\max} \langle p, \theta \rangle$$

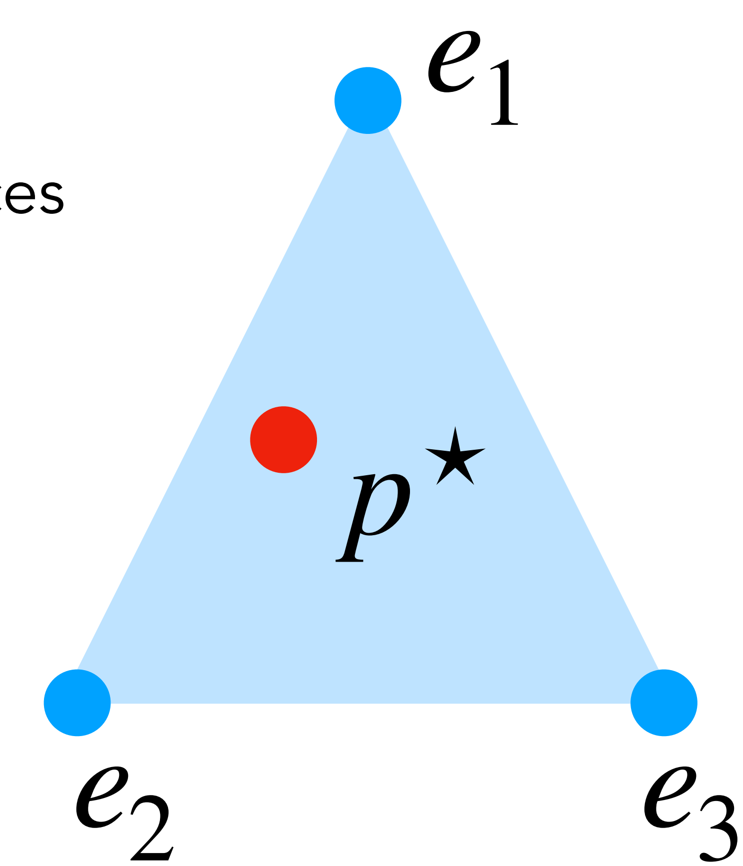
Strongly-convex regularizer

Fundamental theorem of linear programming (Dantzig, 1955)



unregularized (Ω=0)

Move solution
away from the simplex vertices
(spread probability mass)



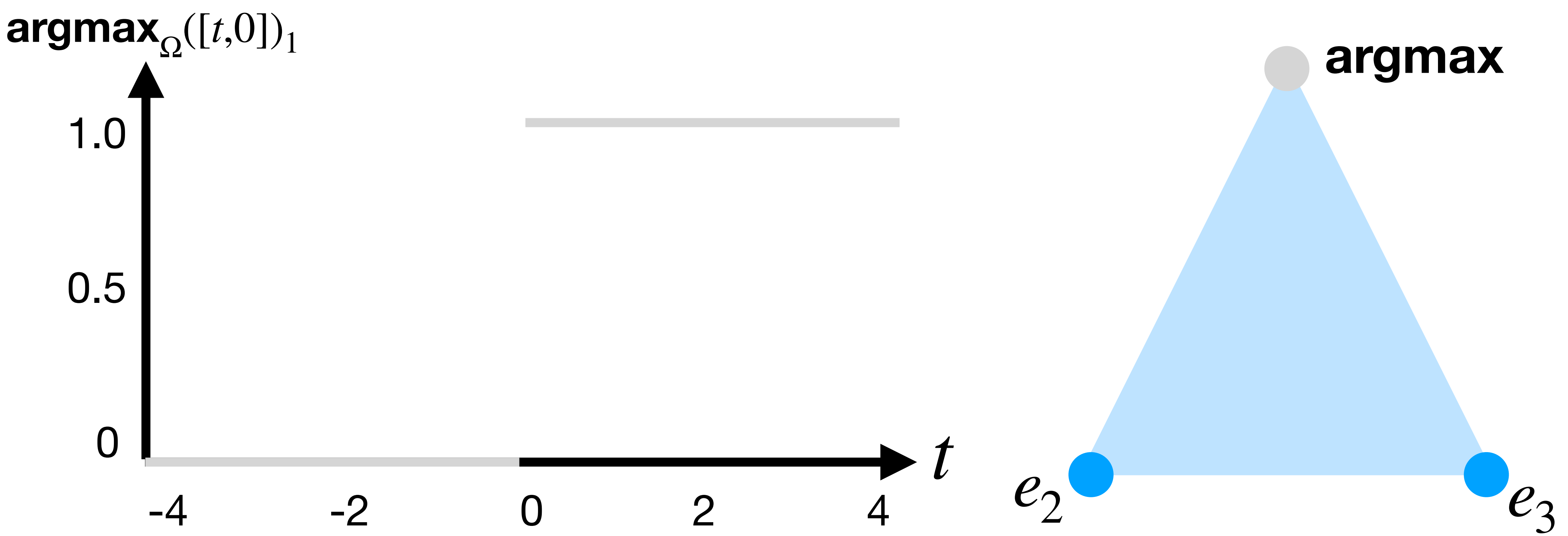
regularized

Examples

Examples

Unregularized

$$\Omega(p) = 0$$



$\textbf{argmax}([t,0])_1$

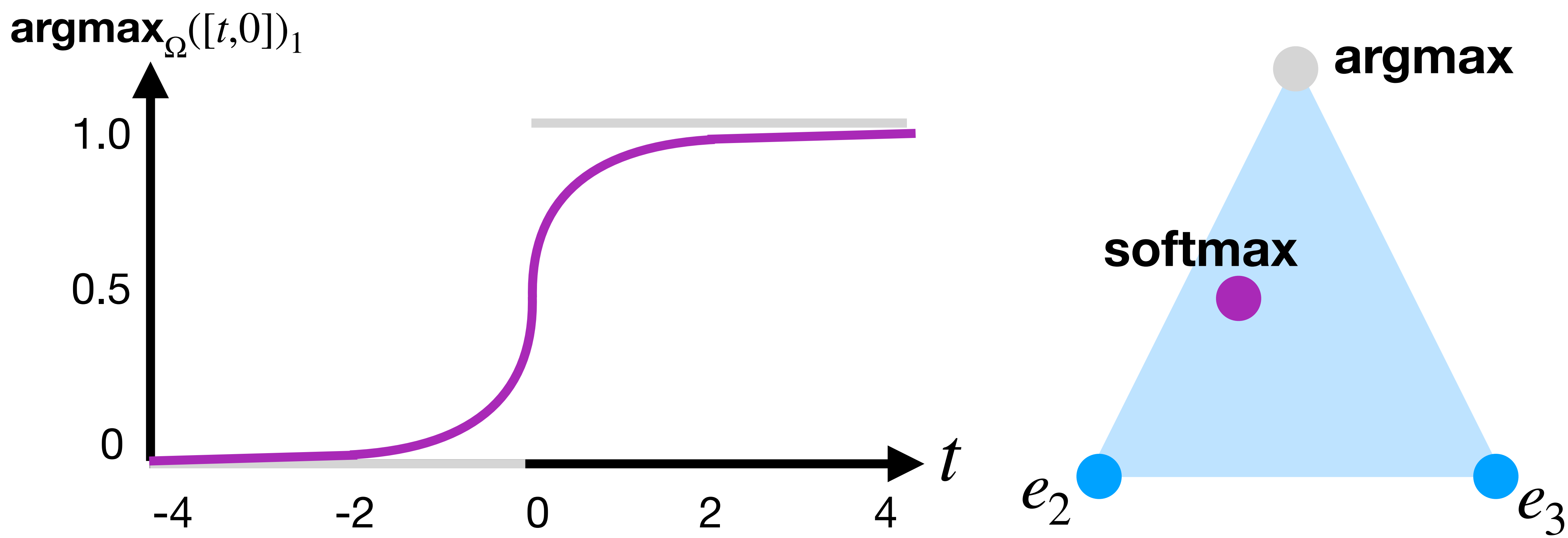
Examples

Unregularized

$$\Omega(p) = 0$$

Shannon (negative) entropy

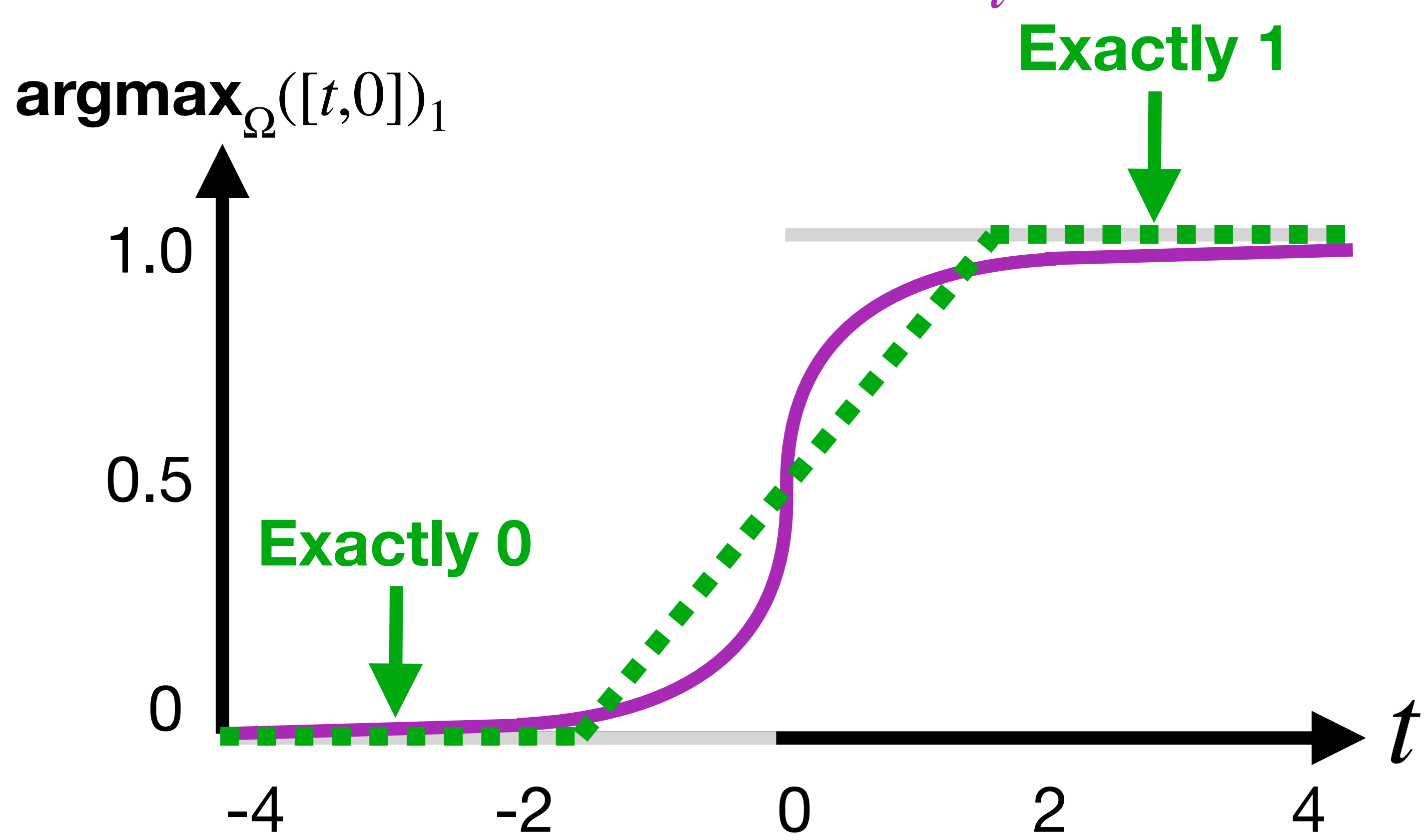
$$\Omega(p) = \sum_i p_i \log p_i$$

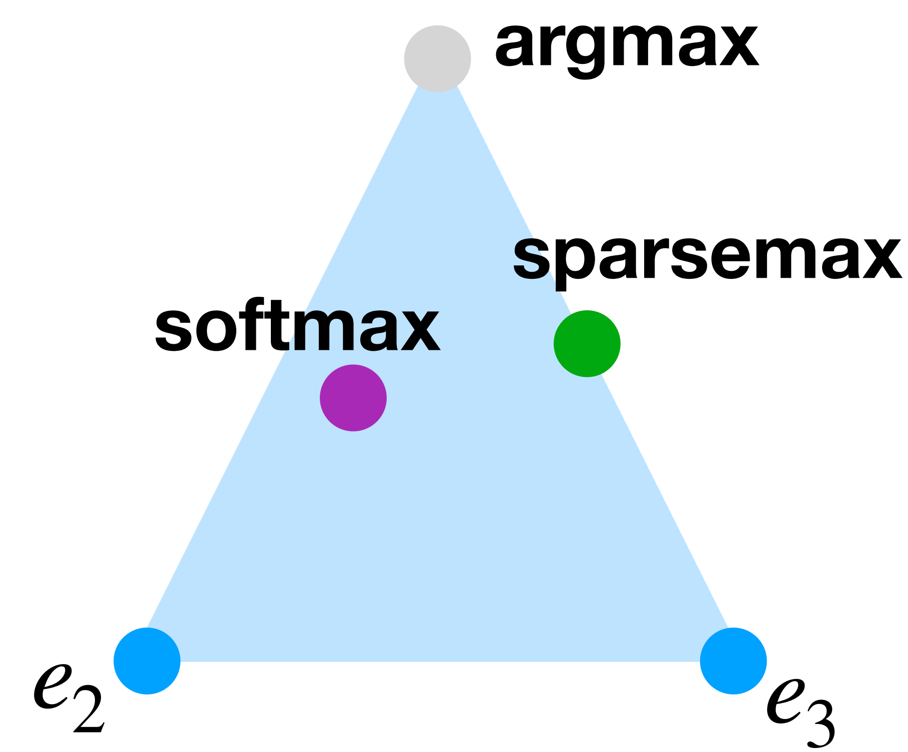


argmax$([t,0])_1$ **softmax**$([t,0])_1$

Examples

Unregularized	Shannon (negative) entropy	Squared norm
$\Omega(p) = 0$	$\Omega(p) = \sum_i p_i \log p_i$	$\Omega(p) = \frac{1}{2}\|p\|^2$

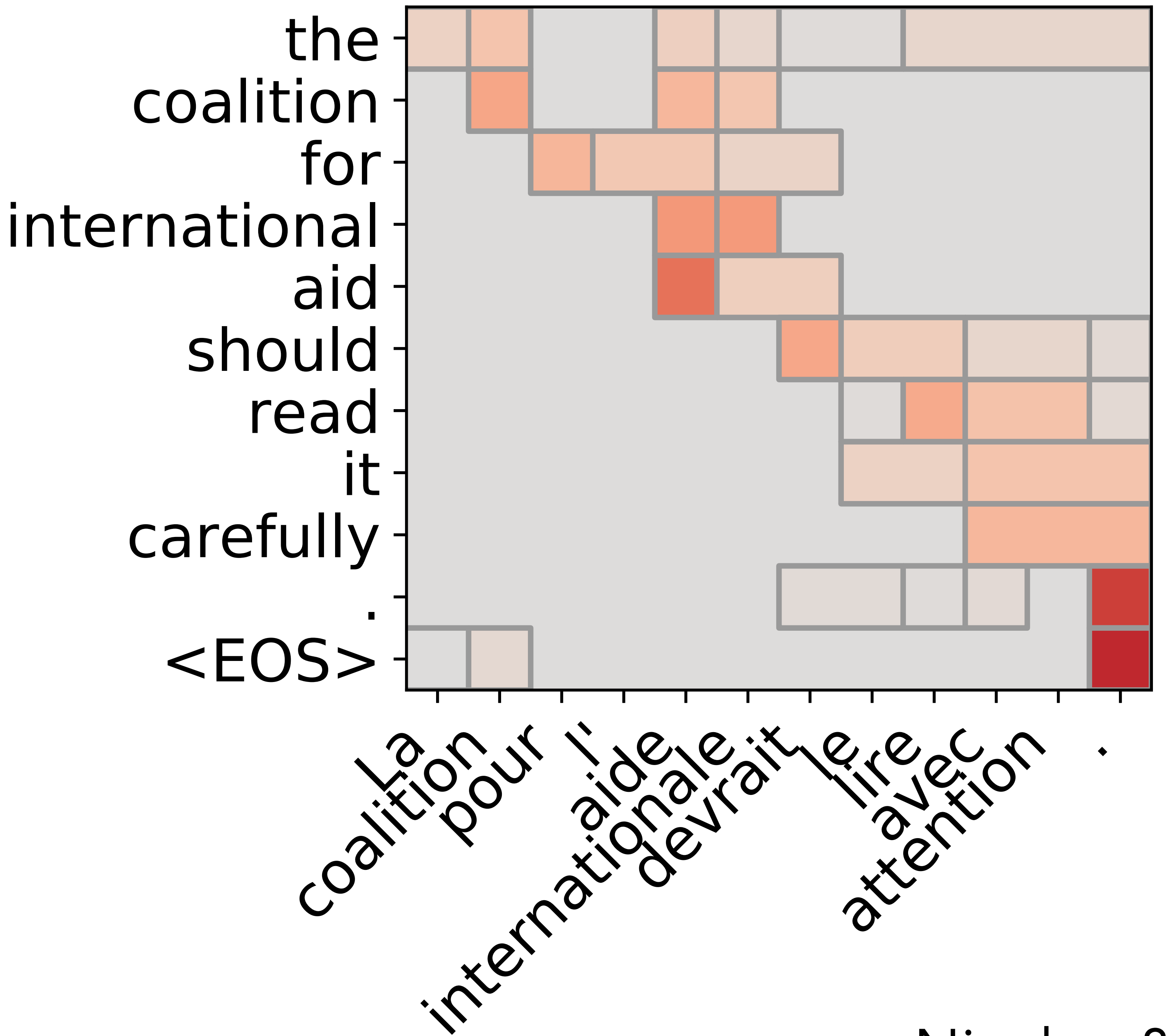




argmax([t,0])$_1$ softmax([t,0])$_1$ sparsemax([t,0])$_1$

Fusedmax attention

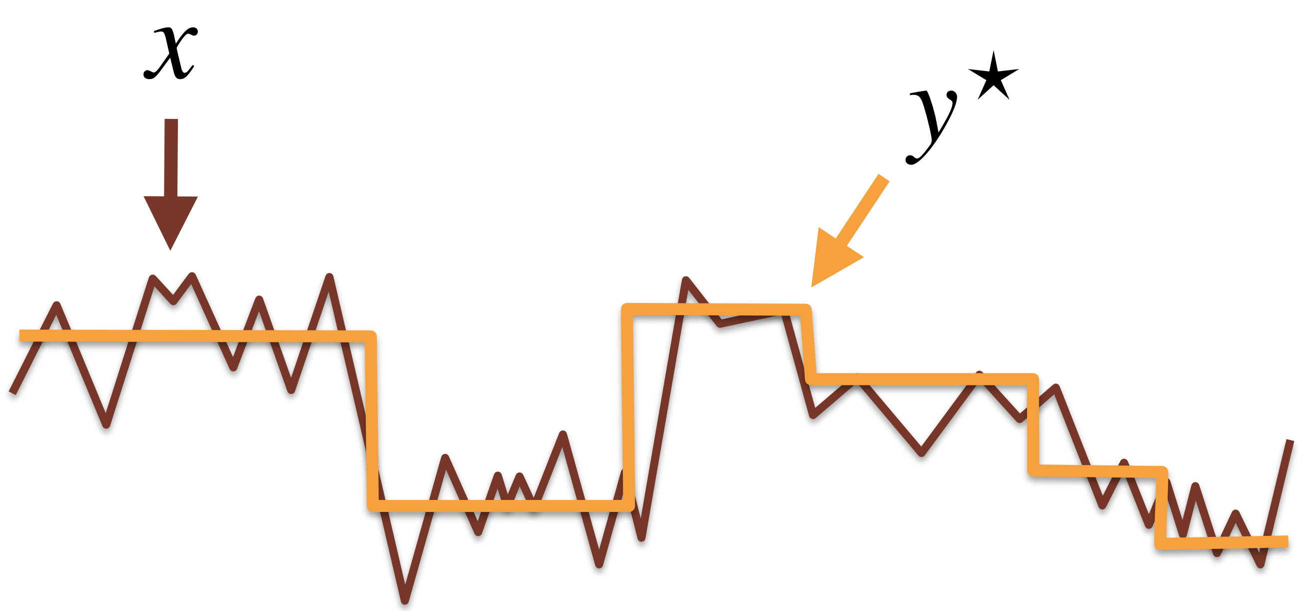
$$\textbf{fusedmax}(\theta) = \textbf{argmax}_{\Omega}(\theta)$$



Niculae & Blondel, NIPS 2017

Fused Lasso (a.k.a. 1d total variation)

$$\mathbf{prox}_{TV}(x) \triangleq \underset{y \in \mathbb{R}^m}{\arg\min} \|x - y\|^2 + \lambda \sum_{i=1}^{m-1} |y_{i+1} - y_i|$$



Total variation signal denoising

Fusedmax attention

sparsemax **fused lasso**

We choose $$\Omega(p) \triangleq \frac{1}{2}\|p\|^2 + \lambda \sum_{i=1}^{m-1} |p_{i+1} - p_i|$$

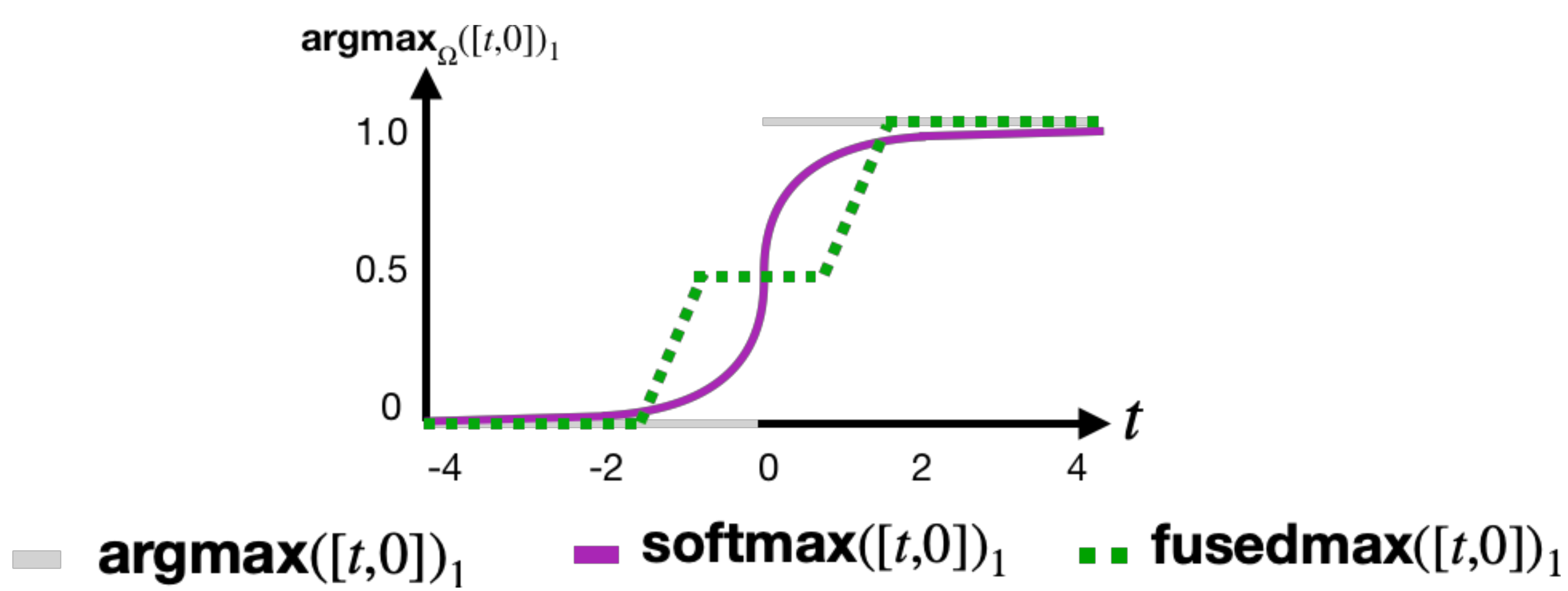
Fusedmax attention

sparsemax fused lasso

We choose $$\Omega(p) \triangleq \frac{1}{2}\|p\|^2 + \lambda \sum_{i=1}^{m-1} |p_{i+1} - p_i|$$

leading to

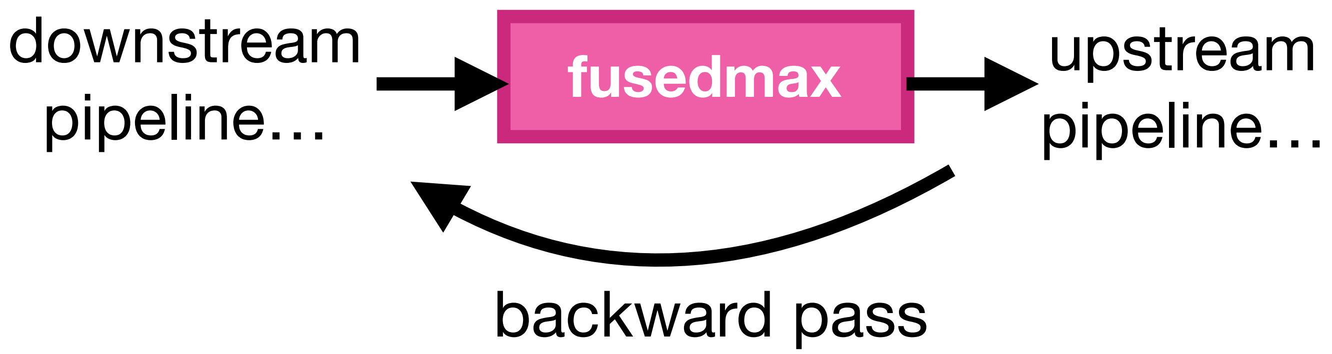
$$\textbf{fusedmax}(\theta) \triangleq \operatorname*{arg\,min}_{p \in \triangle^m} \frac{1}{2}\|p - \theta\|^2 + \lambda \sum_{i=1}^{m-1} |p_{i+1} - p_i|$$



Fusedmax: computation

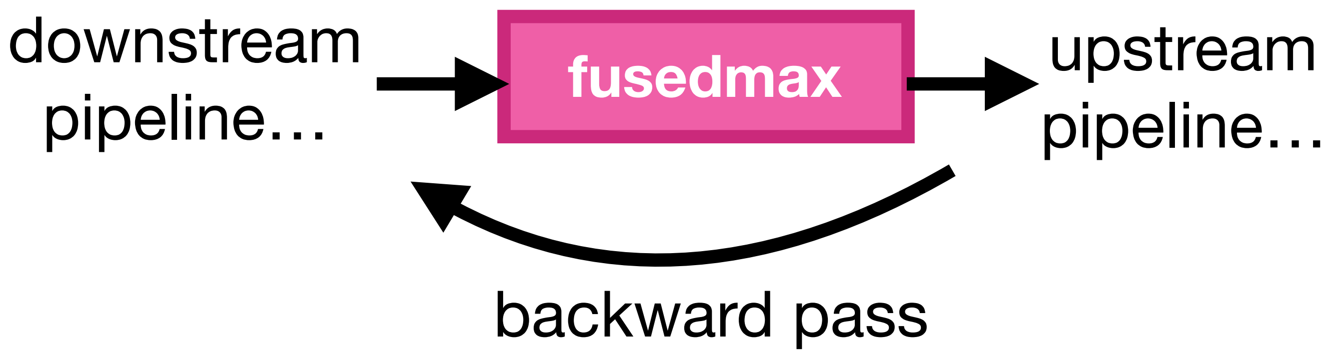
Fusedmax: computation

How to compute forward and backward passes?



Fusedmax: computation

How to compute forward and backward passes?

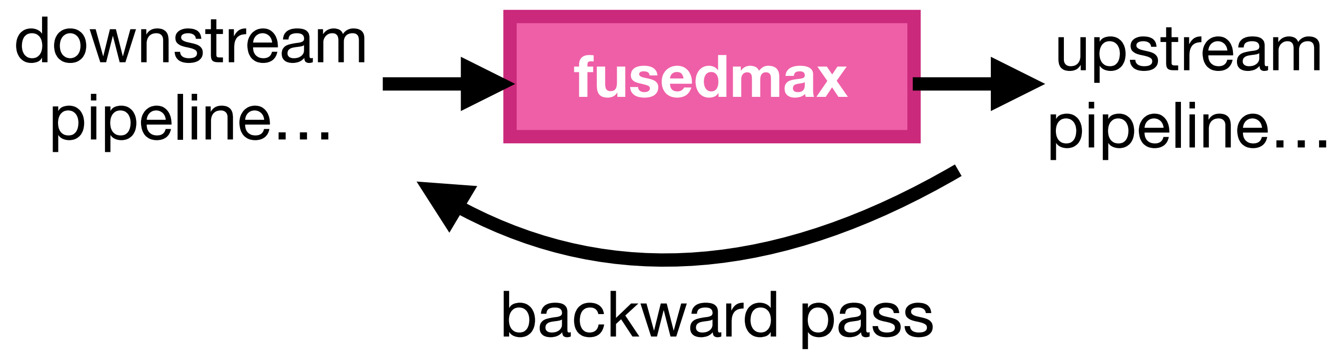


Proposition (Niculae & Blondel, 2017)

$$\textbf{fusedmax} = \textbf{sparsemax} \circ \textbf{prox}_{TV}$$

Fusedmax: computation

How to compute forward and backward passes?



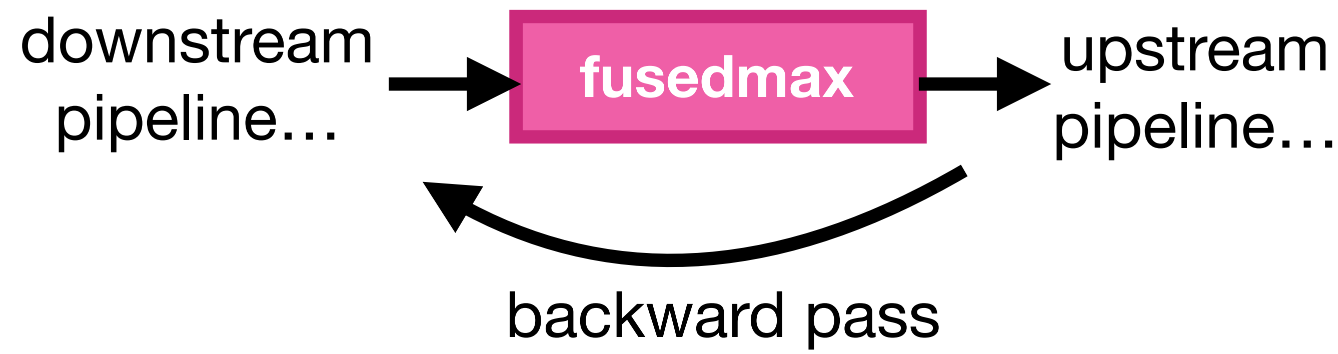
Proposition (Niculae & Blondel, 2017)

Not true for all regularizers!

$$\textbf{fusedmax} = \textbf{sparsemax} \circ \textbf{prox}_{TV}$$

Fusedmax: computation

How to compute forward and backward passes?



<u>Proposition (Niculae & Blondel, 2017)</u>

Not true for all regularizers!

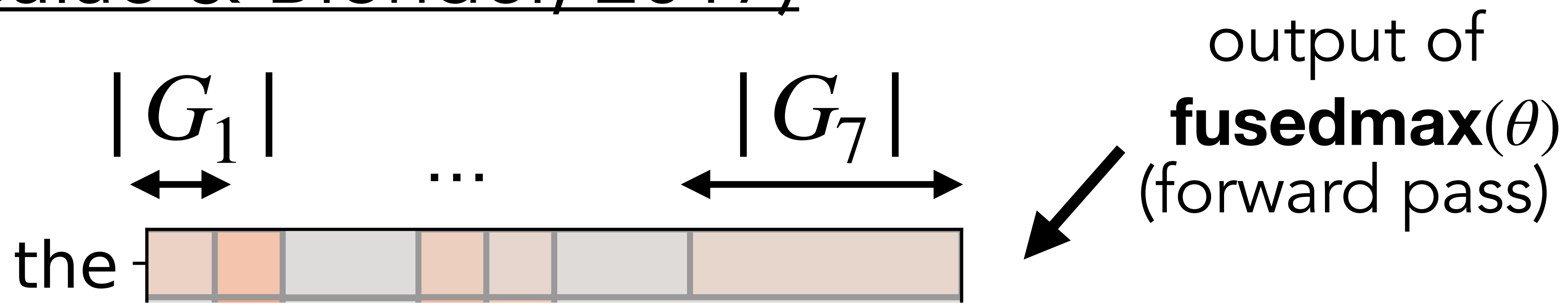
$$\textbf{fusedmax} = \textbf{sparsemax} \circ \textbf{prox}_{TV}$$

	sparsemax	prox_{TV}
forward	Michelot, 1986	Condat, 2013
backward (Jacobian)	Martins & Atstudillo, 2016	?

Jacobian of $\mathbf{prox}_{TV}$

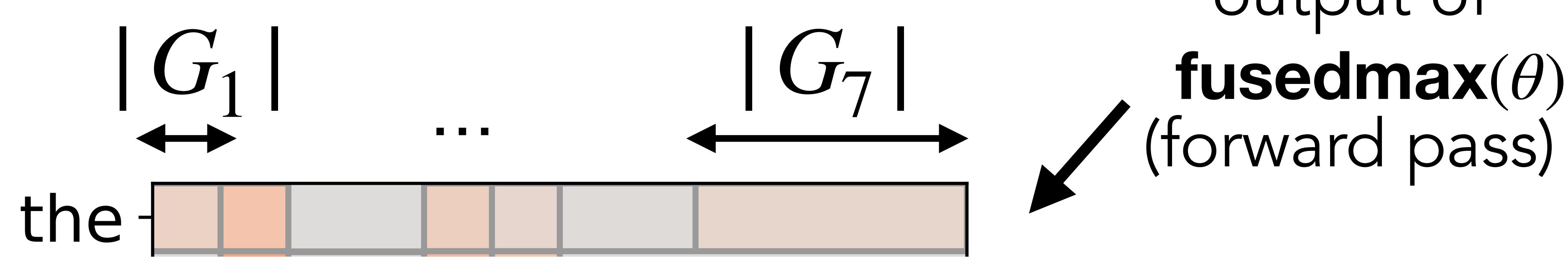
Jacobian of $\mathbf{prox}_{TV}$

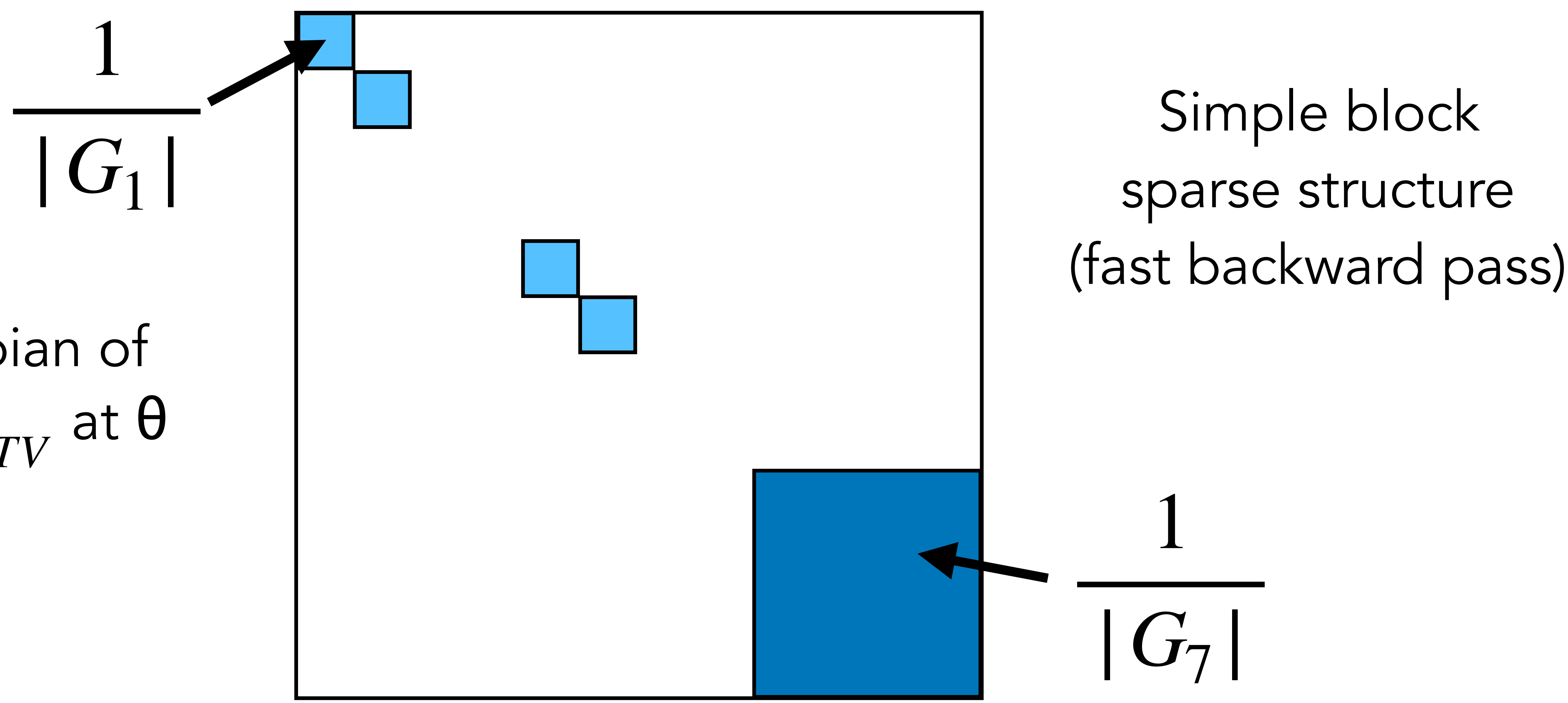
Proposition (Niculae & Blondel, 2017)



Jacobian of **prox**$_{TV}$

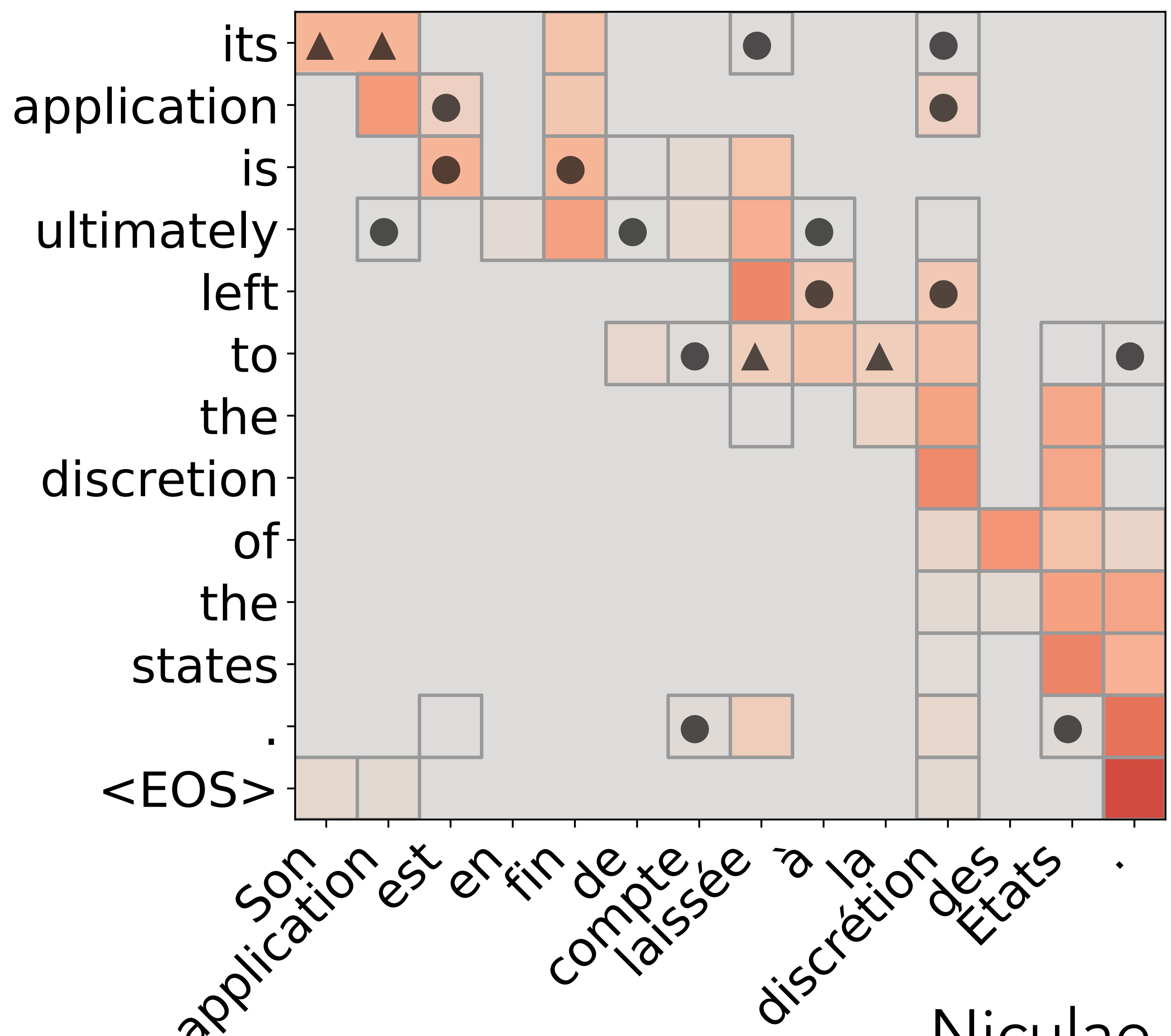
Proposition (Niculae & Blondel, 2017)





Oscarmax attention

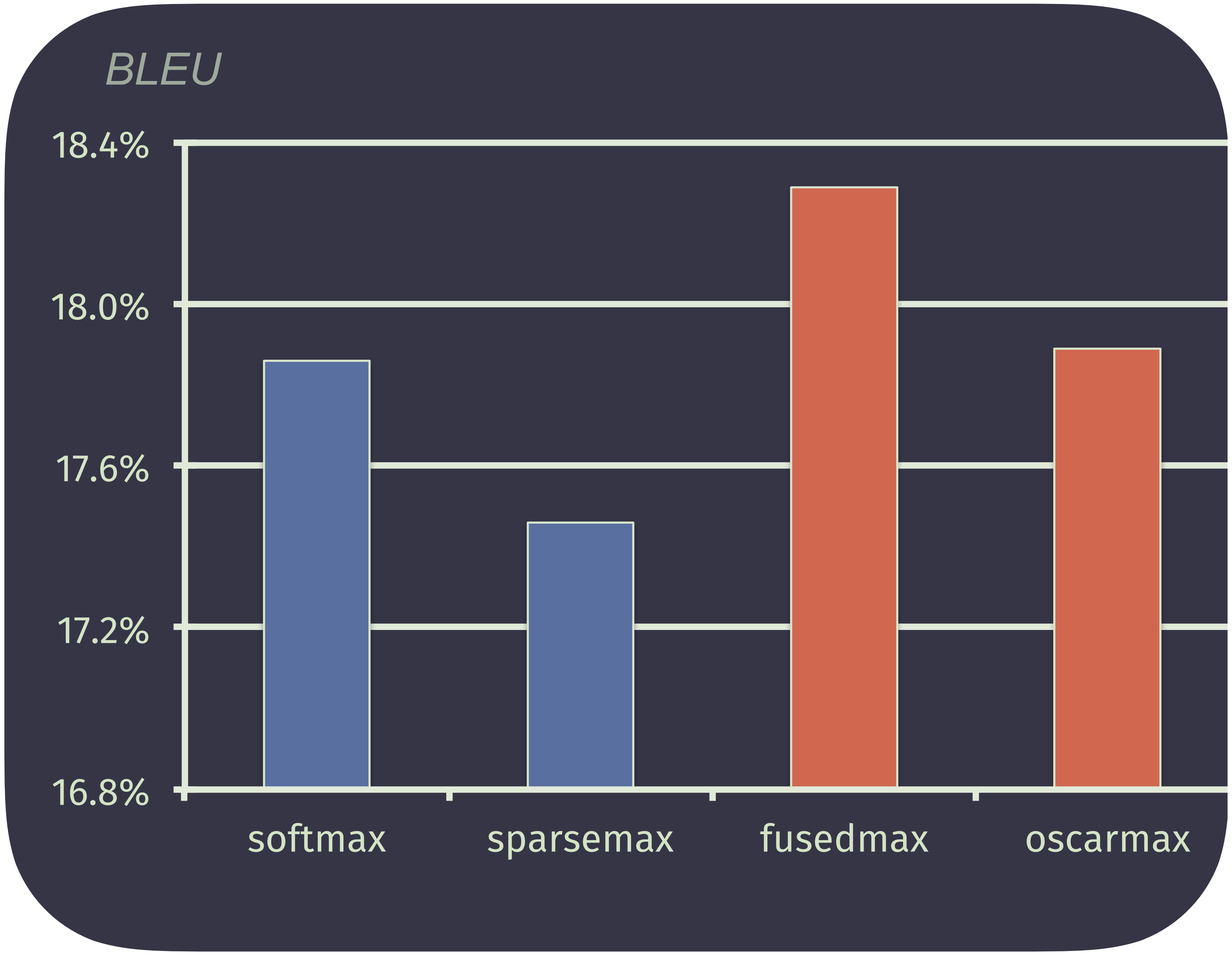
$$\textbf{oscarmax}(\theta) \triangleq \underset{p \in \triangle^m}{\arg\min} \frac{1}{2}\|p - \theta\|^2 + \lambda \sum_{i<j} \max\{\,|p_i|, |p_j|\,\}$$



Niculae & Blondel, NIPS 2017

Neural Machine Translation

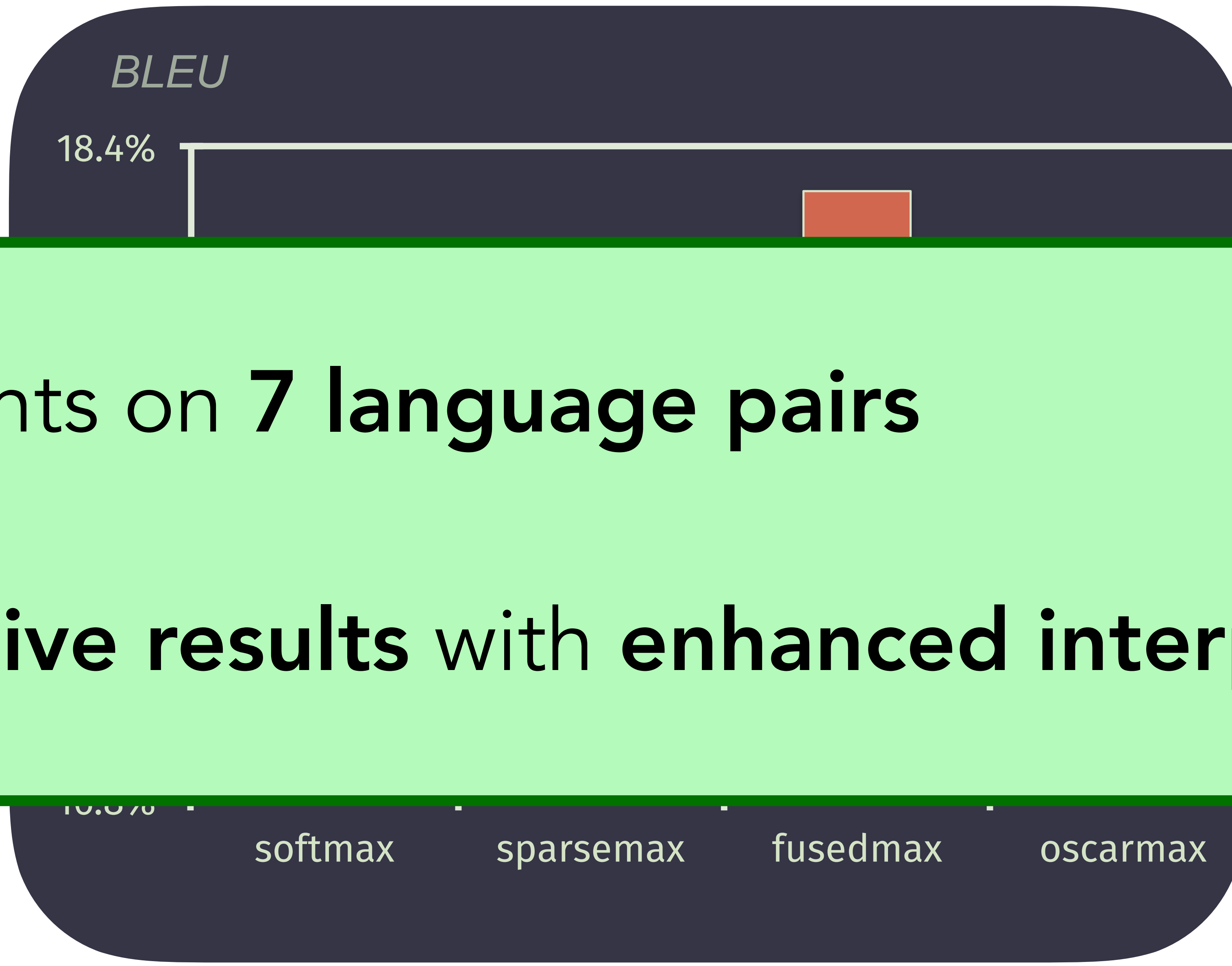
Romanian-English



Experiments based on Open-NMT
using WMT16 dataset

Neural Machine Translation

Romanian-English

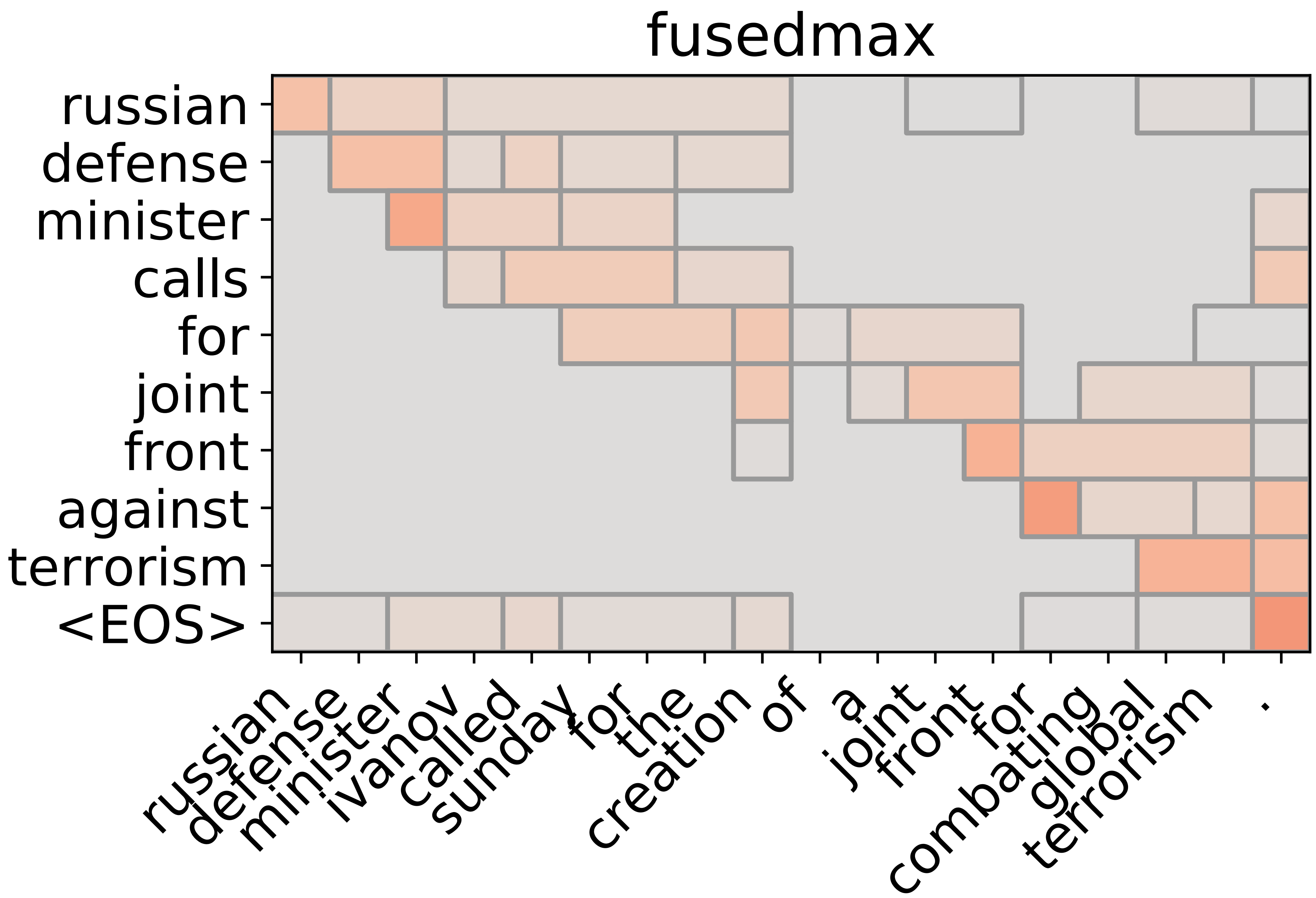


. Experiments on **7 language pairs**

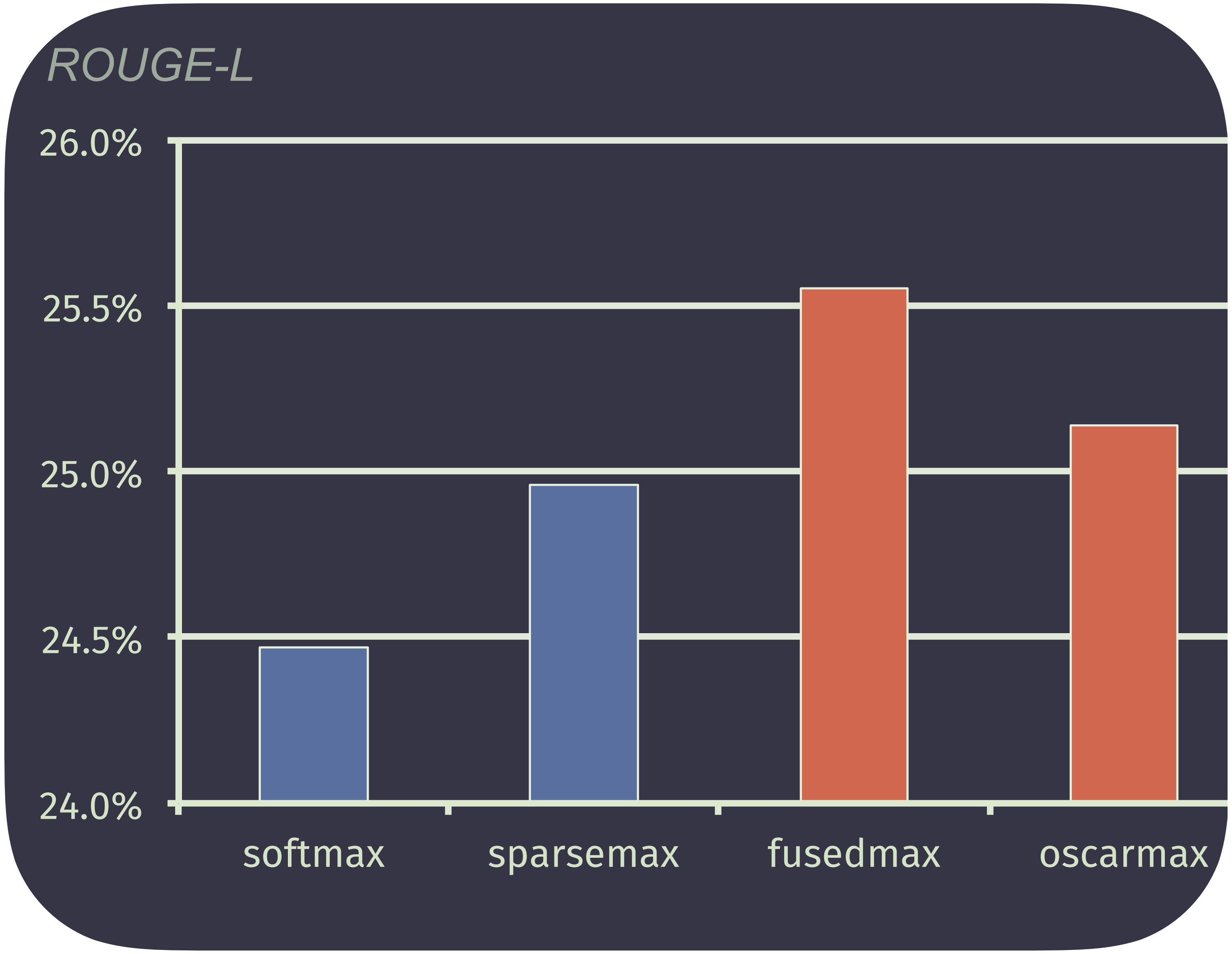
. **Competitive results** with **enhanced interpretability**!

Experiments based on Open-NMT
using WMT16 dataset

Sentence summarization



Sentence summarization

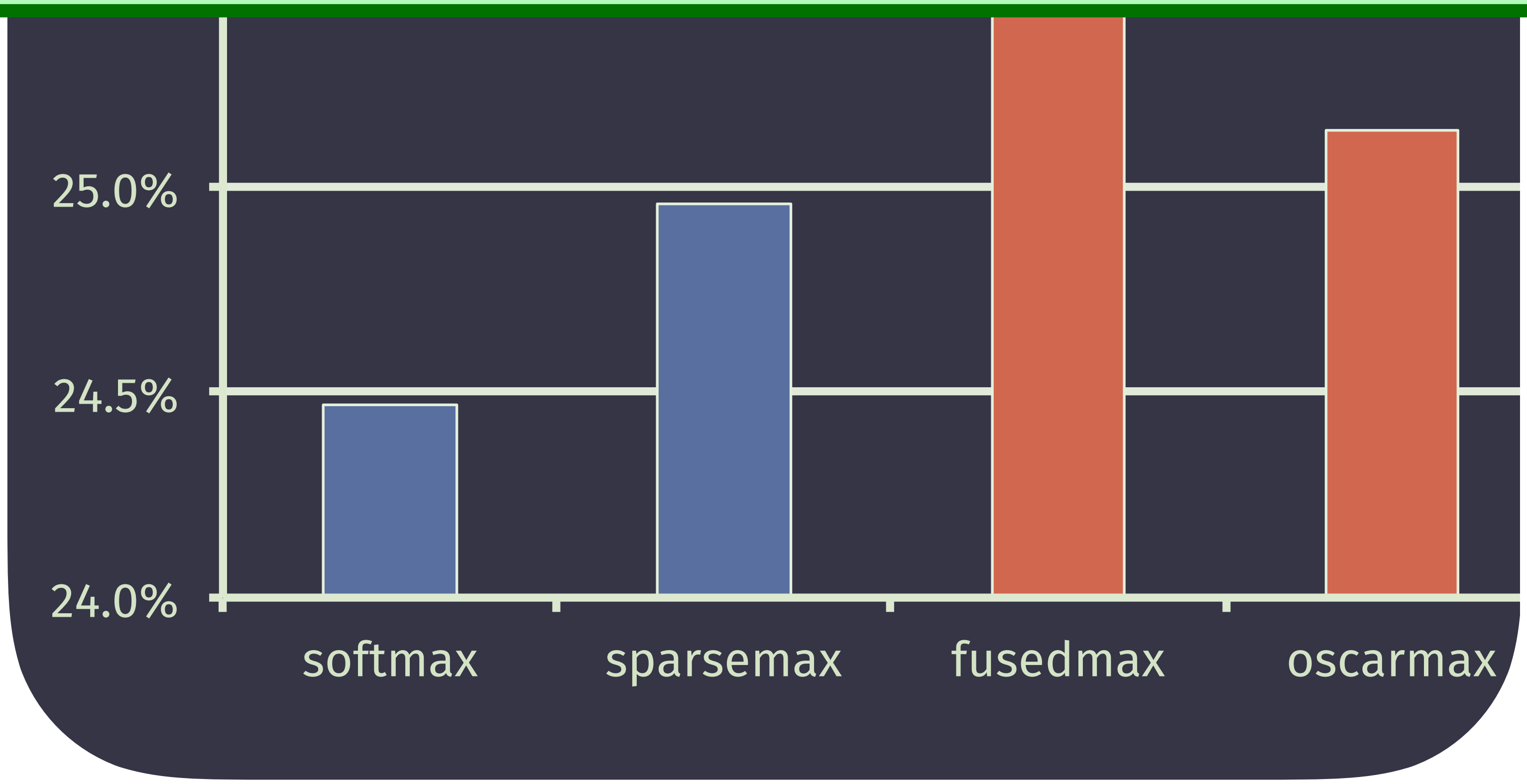


Experiments based on Open-NMT
using the Gigaword sentence summarization dataset

Sentence summarization

. **Significant accuracy improvement**

. Greatly **enhanced interpretability**



Experiments based on Open-NMT
using the Gigaword sentence summarization dataset

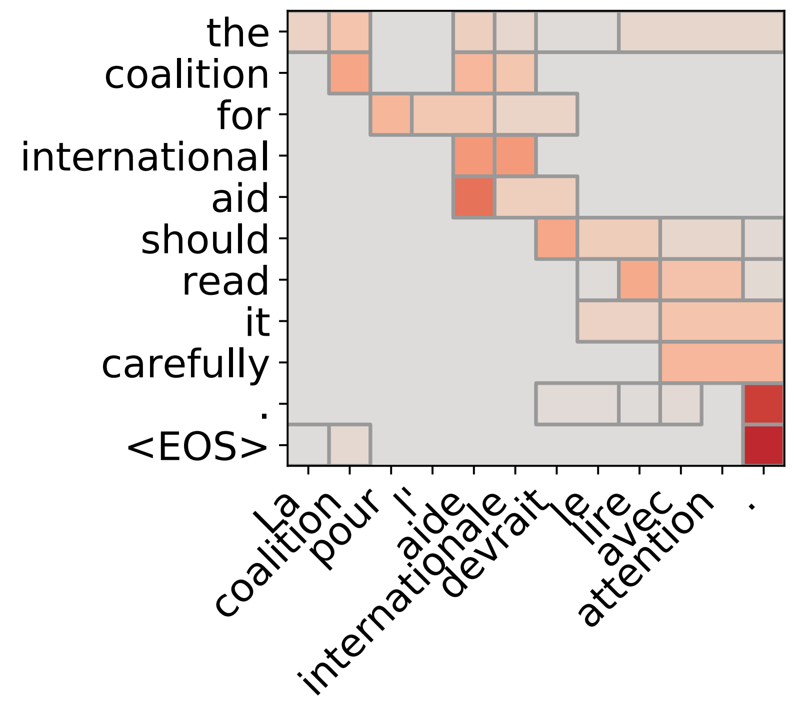
Summary so far

Principled framework for differentiable argmax operators

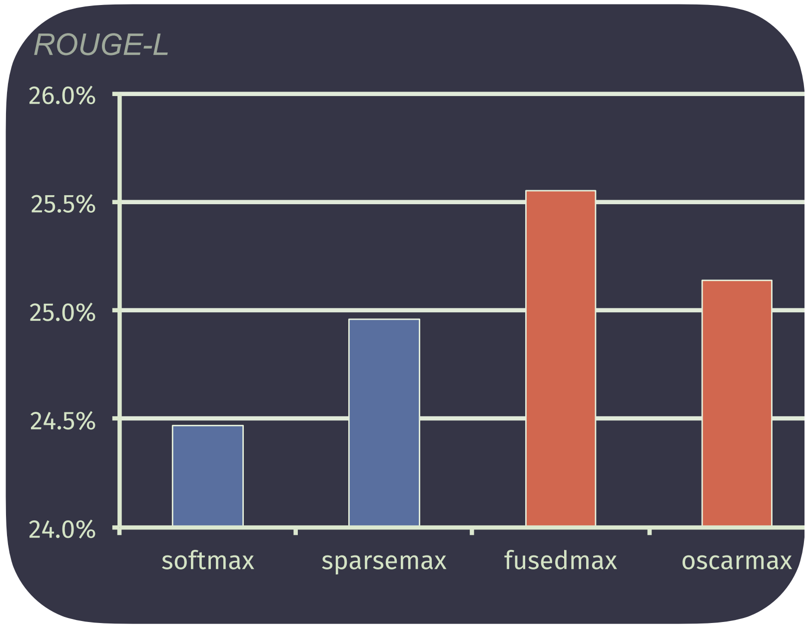
$$\textbf{argmax}_{\Omega}(\theta) \triangleq \arg\max_{p \in \triangle^m} \langle p, \theta \rangle - \Omega(p)$$

mechanism	regularization Ω
softmax	Shannon's neg-entropy
sparsemax	squared norm
fusedmax	squared norm + fused lasso

New interpretable attention mechanisms



Great accuracy on various applications



Faster training by leveraging sparsity

attention	time per epoch
softmax	1h 26m 40s ± 51s
sparsemax	1h 24m 21s ± 54s
fusedmax	1h 23m 58s ± 50s
oscarmax	1h 23m 19s ± 50s

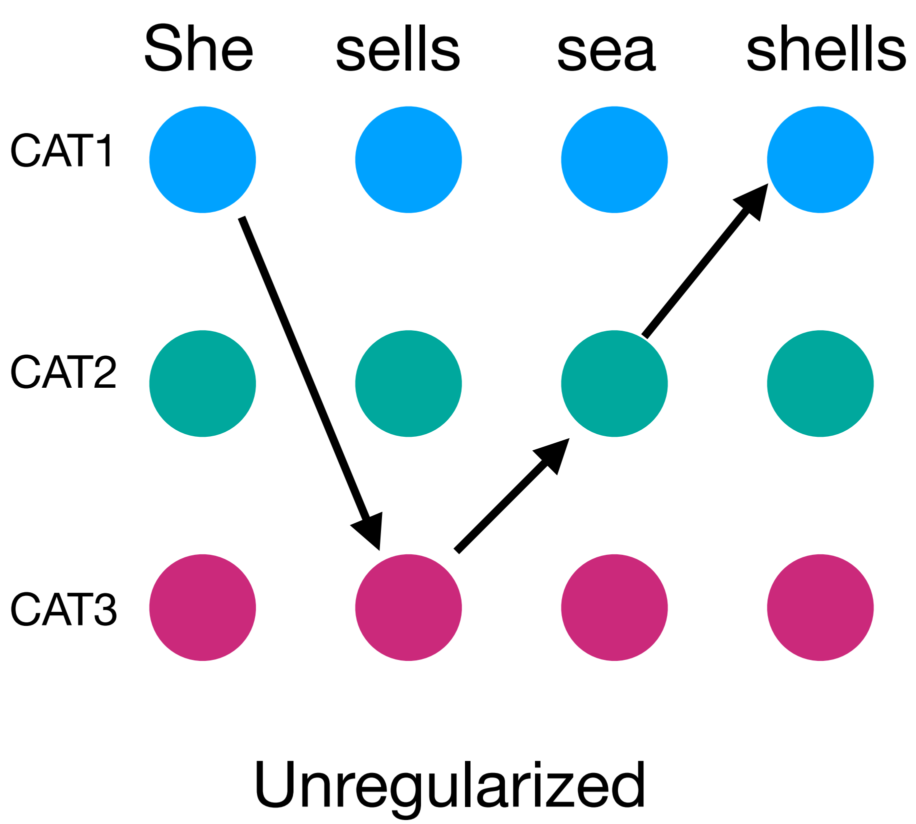
Outline

1. Structured attention

2. Differentiable dynamic programming

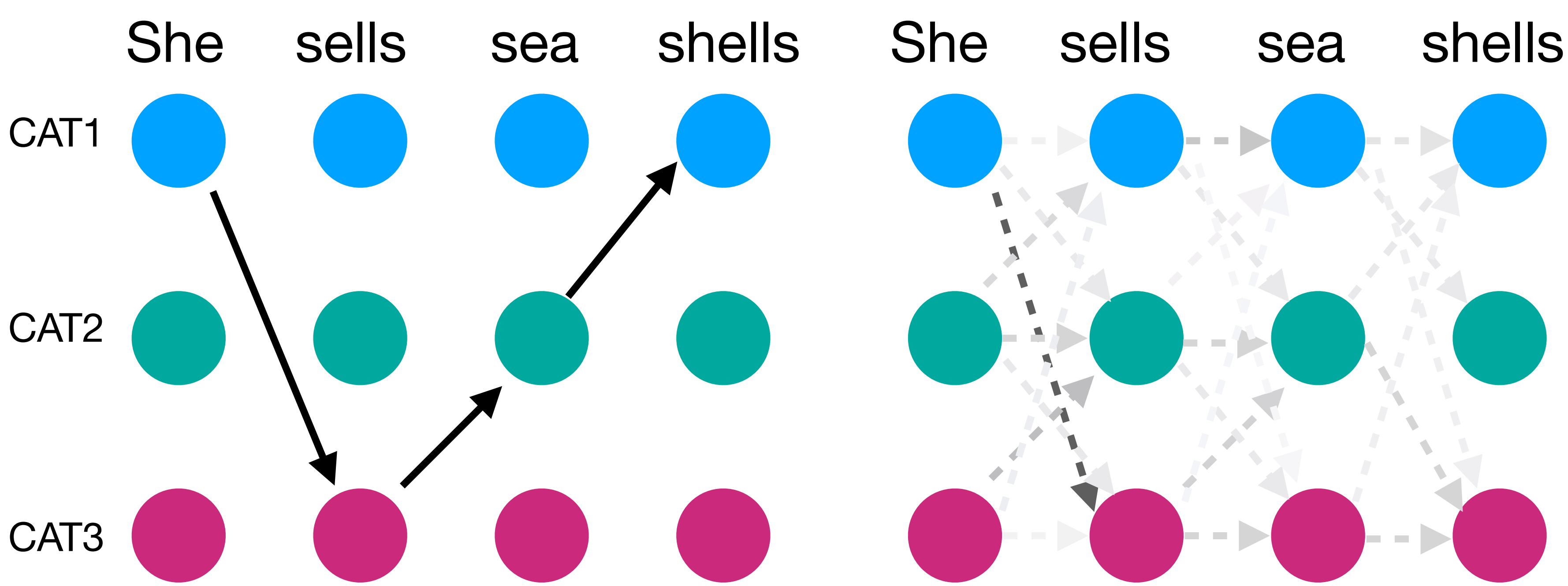
Soft Viterbi algorithm: sequence tagging

Soft Viterbi algorithm: sequence tagging



one path in the DAG = one possible tag sequence

Soft Viterbi algorithm: sequence tagging



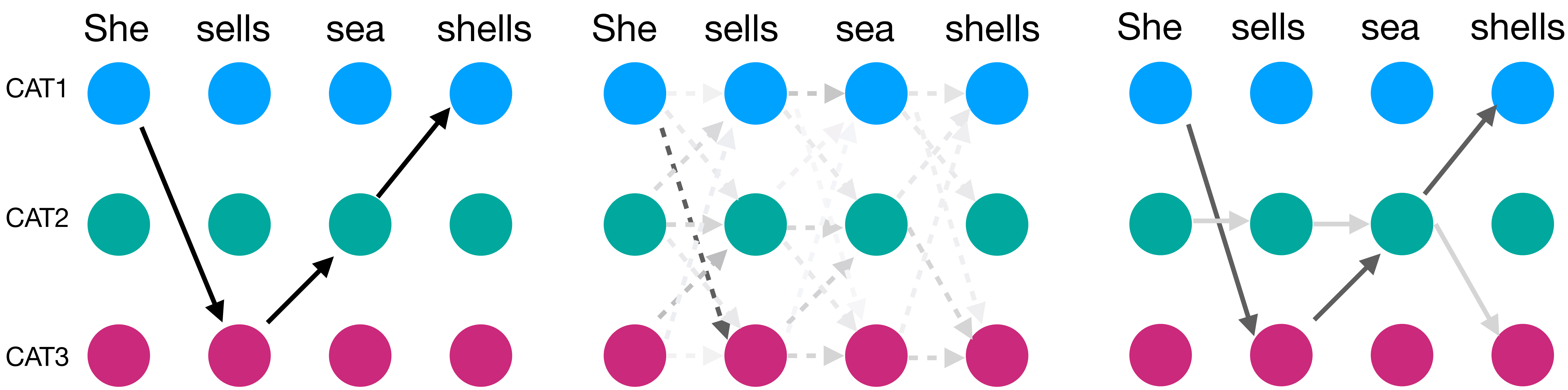
Unregularized

Entropic regularization
⇔ Linear-chain CRF

(Lafferty et al., 2001)

one path in the DAG = one possible tag sequence

Soft Viterbi algorithm: sequence tagging



Unregularized

Entropic regularization
⇔ Linear-chain CRF
(Lafferty et al., 2001)

Quadratic regularization
(Mensch & Blondel, 2018)
Sparse sequence distribution

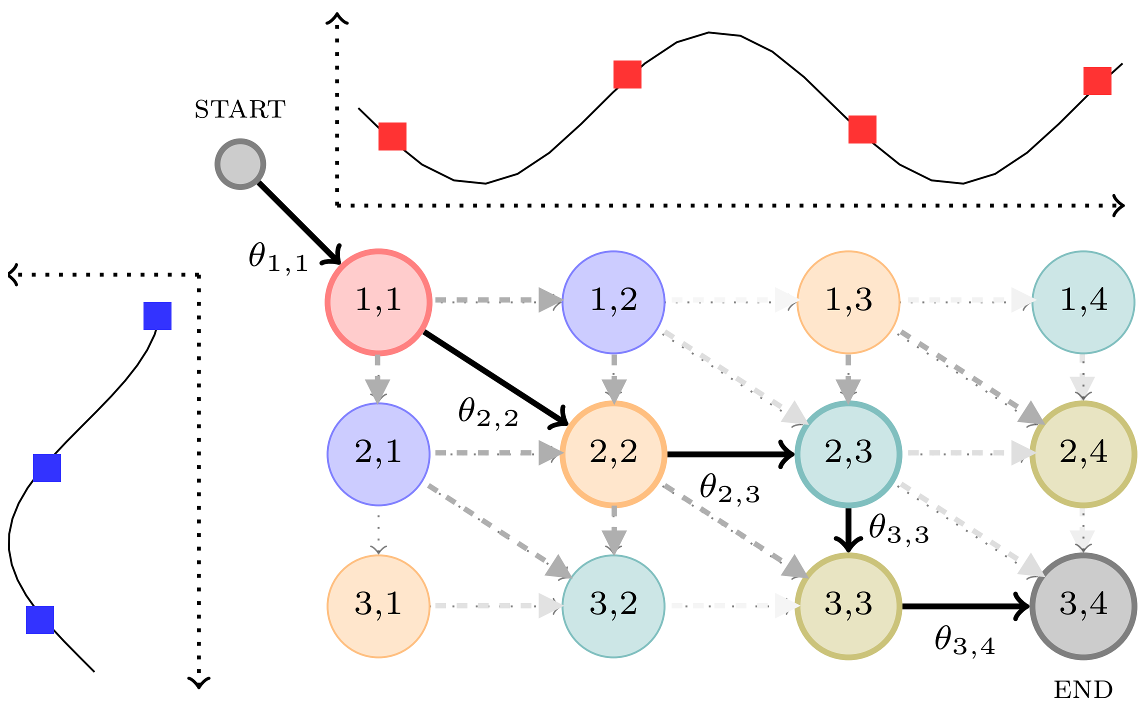
one path in the DAG = one possible tag sequence

Soft DTW: time series alignment

DTW = Dynamic Time Warping [Sakoe & Chiba, 1978]

Soft DTW: time series alignment

DTW = Dynamic Time Warping [Sakoe & Chiba, 1978]



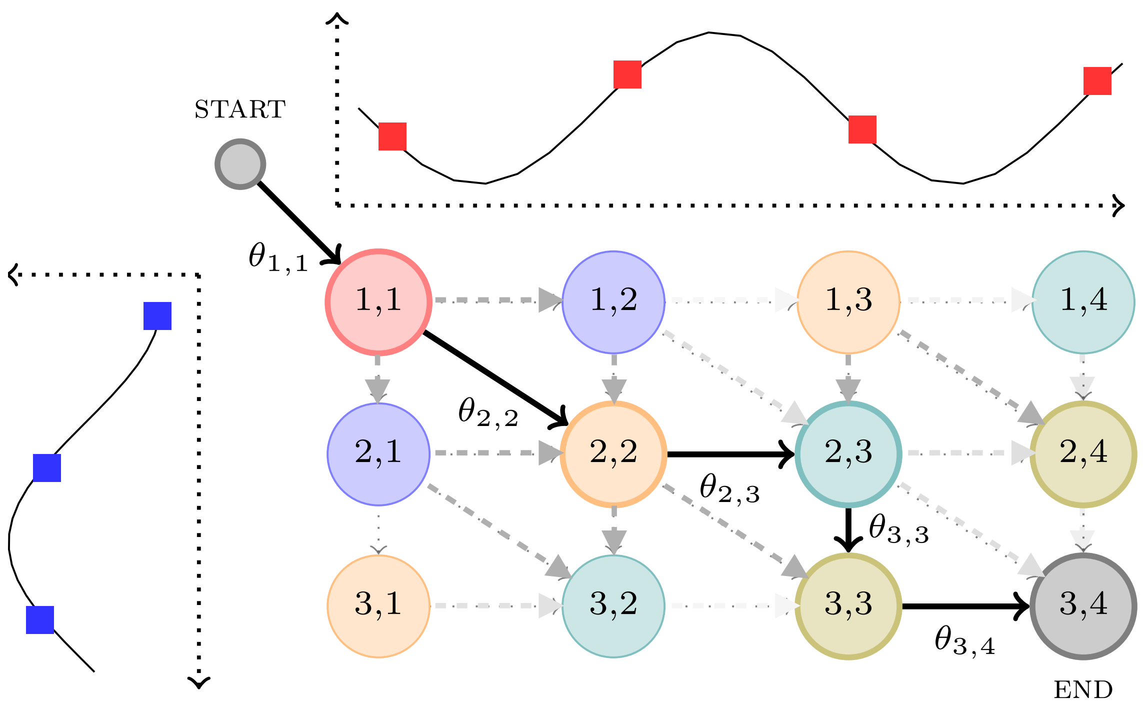
Entropic regularization
⇔ Soft-DTW

(Cuturi & Blondel, 2017)

one path in the DAG
=
one possible **monotonic** time-series alignment

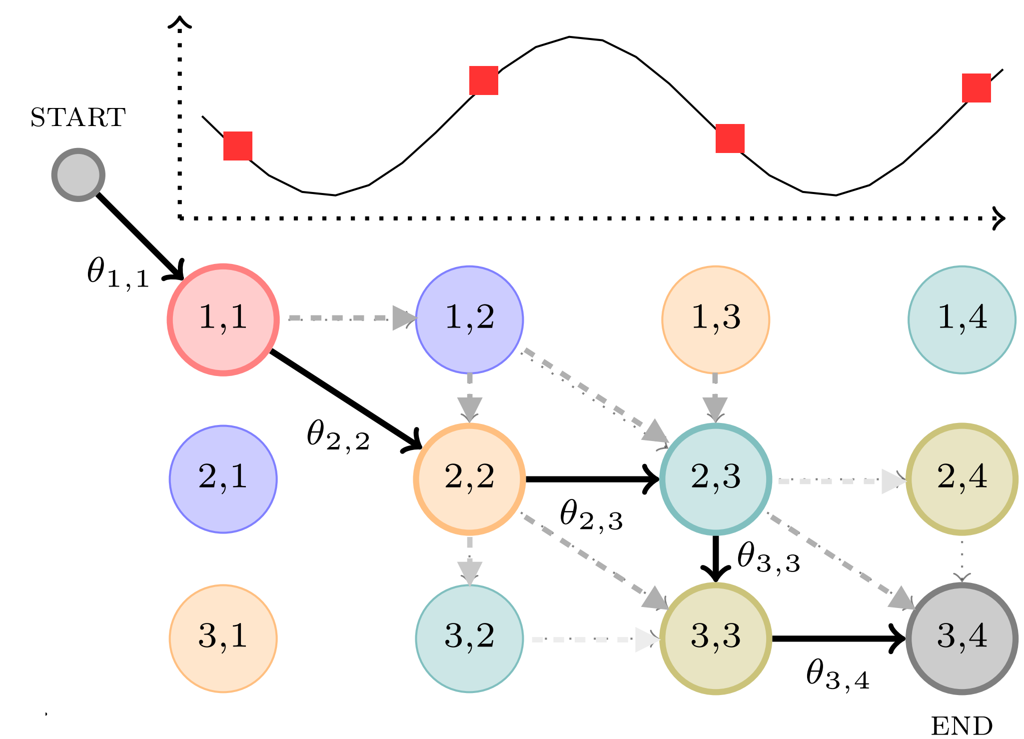
Soft DTW: time series alignment

DTW = Dynamic Time Warping [Sakoe & Chiba, 1978]



Entropic regularization
⇔ Soft-DTW

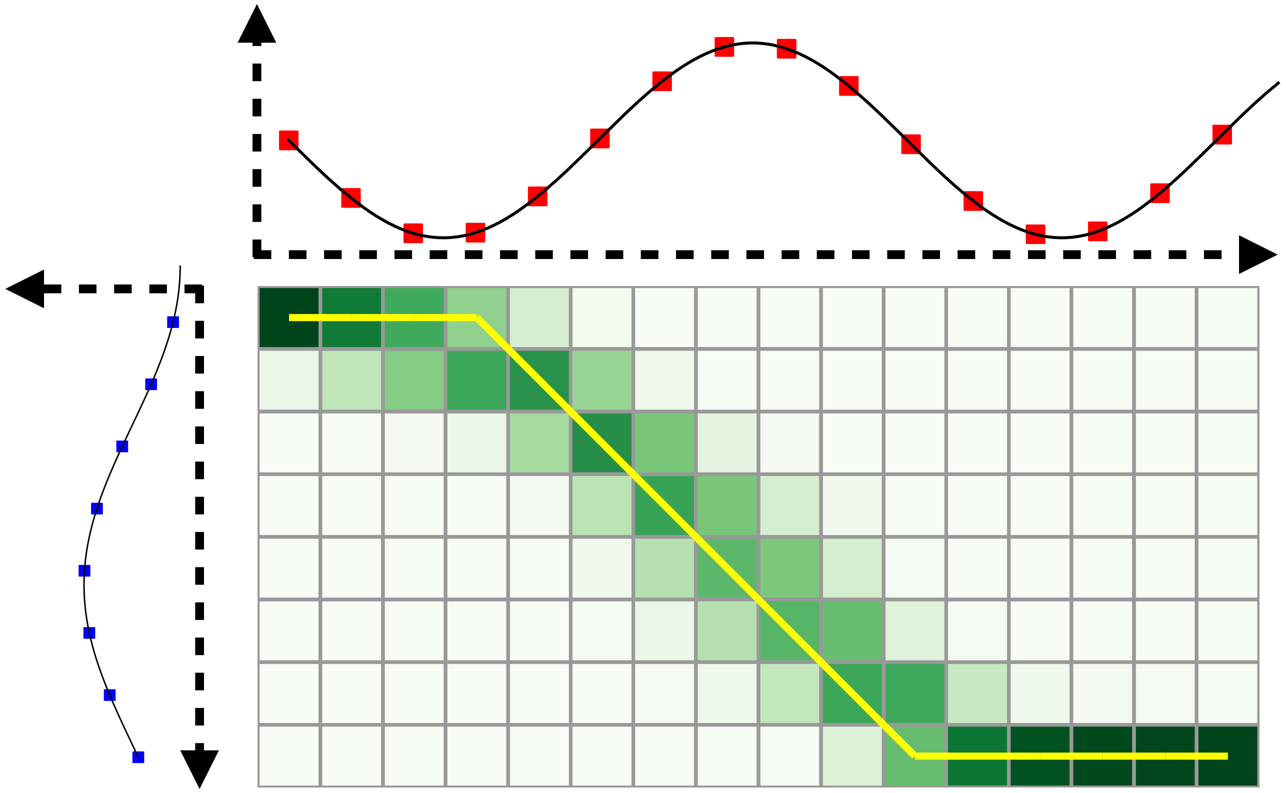
(Cuturi & Blondel, 2017)



Quadratic regularization
(Mensch & Blondel, 2018)
Sparse alignment distribution

one path in the DAG
=
one possible **monotonic** time-series alignment

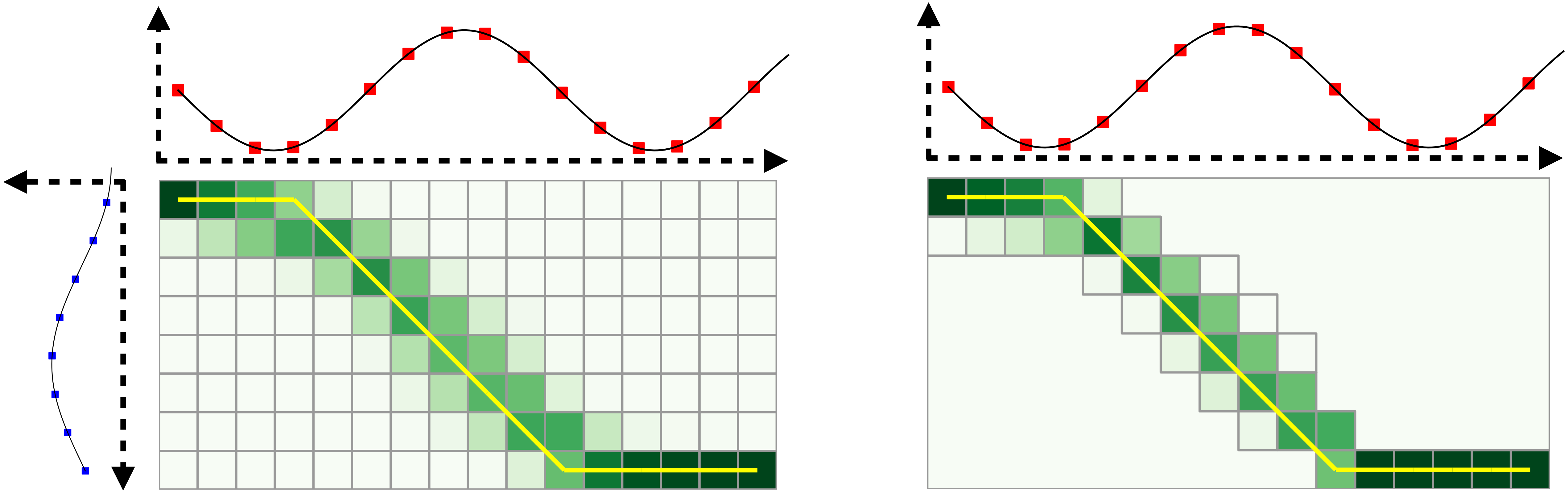
Expected Alignment (Path)



Entropic regularization
(Cuturi & Blondel, 2017)

Hard solution (DTW alignment) Soft solution (**expected alignment** $\mathbb{E}_p[Y]$)

Expected Alignment (Path)



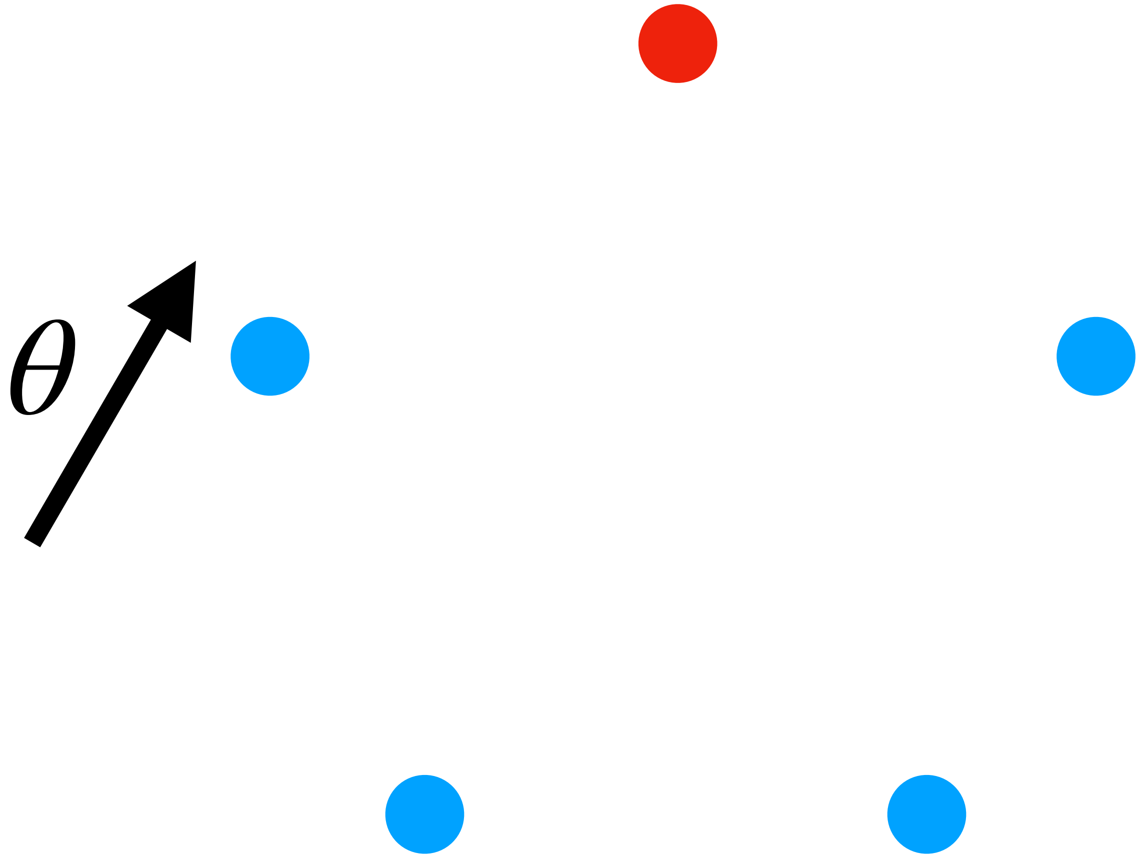
Entropic regularization
(Cuturi & Blondel, 2017)

Quadratic regularization
(Mensch & Blondel, 2018)

Hard solution (DTW alignment) Soft solution (**expected alignment** $\mathbb{E}_p[Y]$)

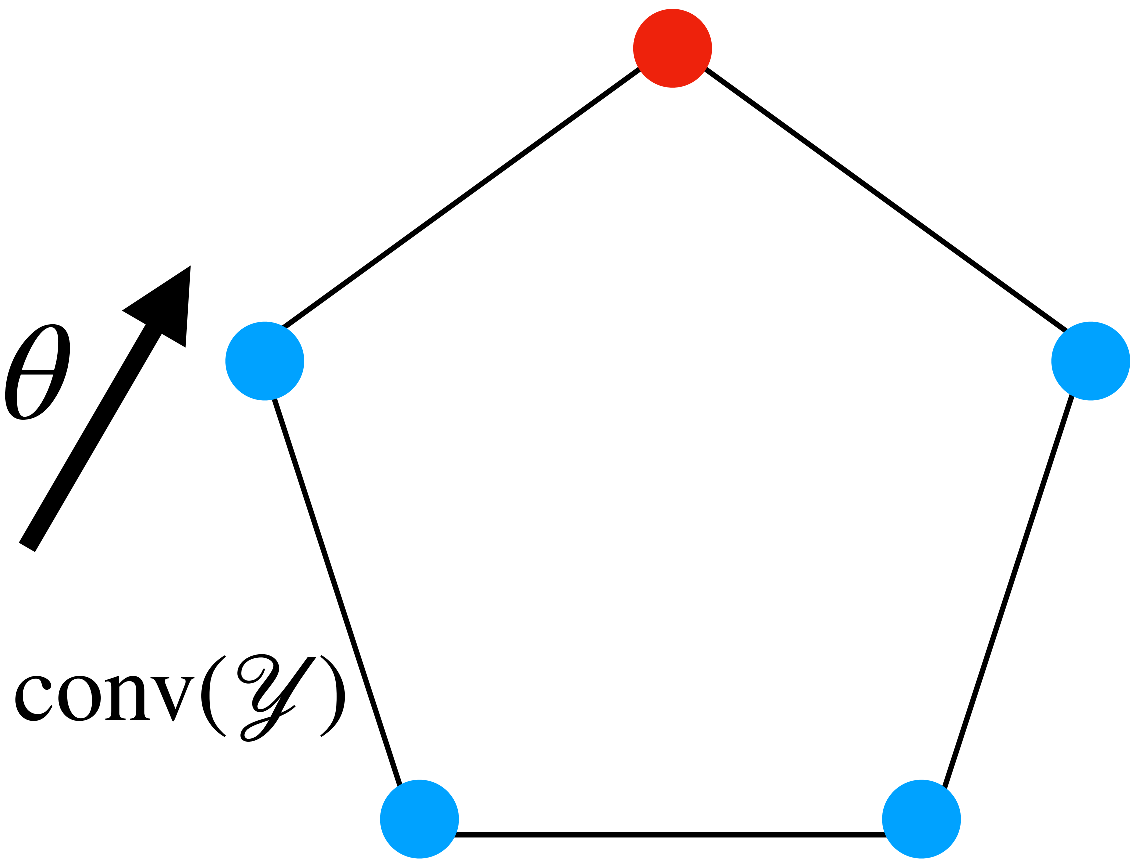
MAP inference: Highest-scoring Structure

$$\textbf{MAP}(\theta) \triangleq \mathop{\arg\max}_{y \in \mathcal{Y} \subseteq \mathbb{R}^m} \langle y, \theta \rangle$$



MAP inference: Highest-scoring Structure

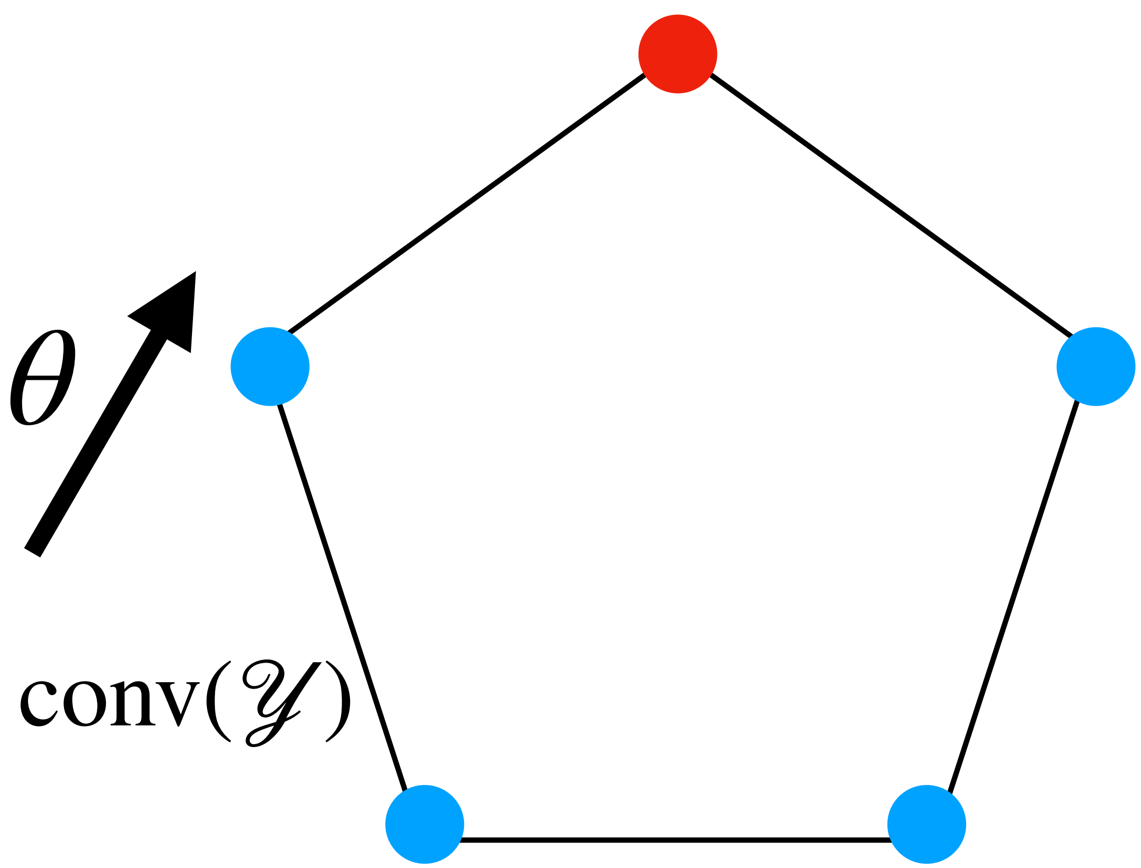
$$\mathbf{MAP}(\theta) \triangleq \underset{y \in \mathcal{Y} \subseteq \mathbb{R}^m}{\arg\max} \langle y, \theta \rangle = \underset{y \in \mathrm{conv}(\mathcal{Y})}{\arg\max} \langle y, \theta \rangle$$



Marginal polytope
(Wainwright & Jordan, 2008)

MAP inference: Highest-scoring Structure

$$\textbf{MAP}(\theta) \triangleq \underset{y \in \mathcal{Y} \subseteq \mathbb{R}^m}{\arg\max} \ \langle y, \theta \rangle = \underset{y \in \text{conv}(\mathcal{Y})}{\arg\max} \ \langle y, \theta \rangle$$

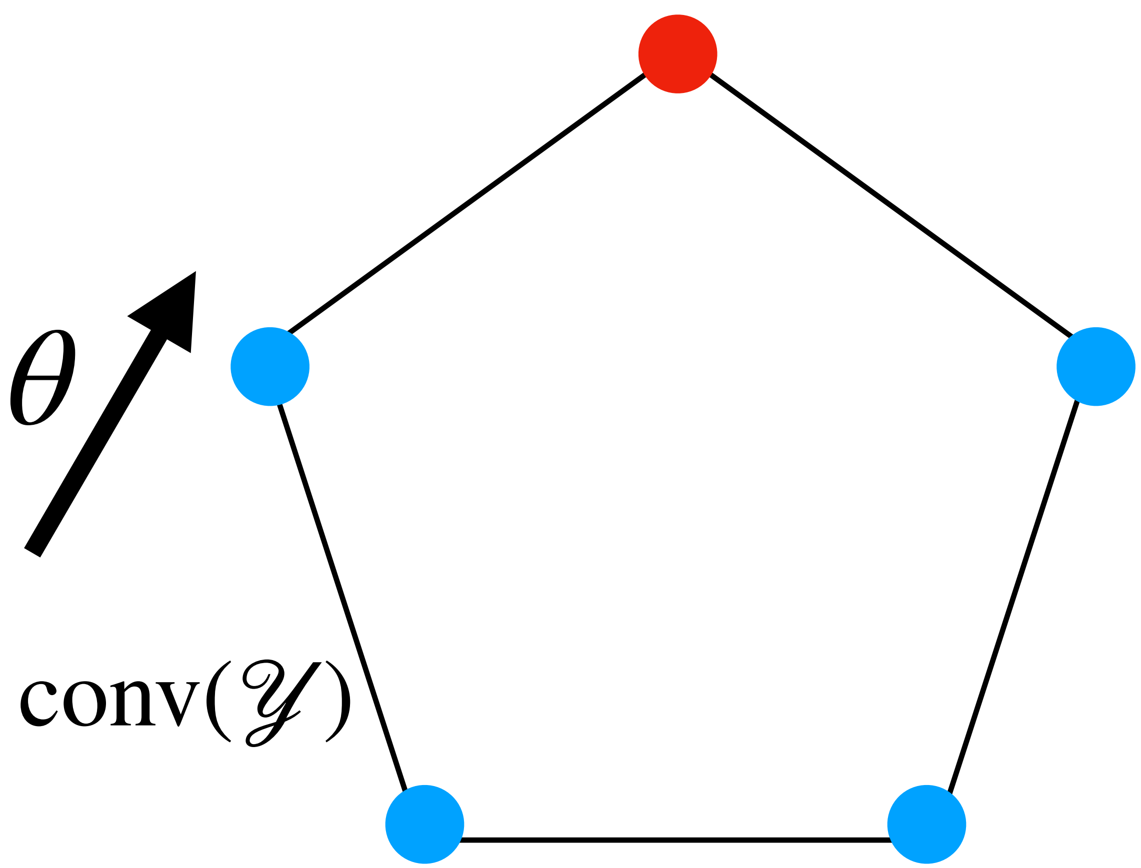


Marginal polytope
(Wainwright & Jordan, 2008)

Can be computed efficiently by dynamic programming in the case of DAGs (no cycle)

MAP inference: Highest-scoring Structure

$$\textbf{MAP}(\theta) \triangleq \underset{y \in \mathcal{Y} \subseteq \mathbb{R}^m}{\arg\max} \langle y, \theta \rangle = \underset{y \in \mathrm{conv}(\mathcal{Y})}{\arg\max} \langle y, \theta \rangle$$



Marginal polytope
(Wainwright & Jordan, 2008)

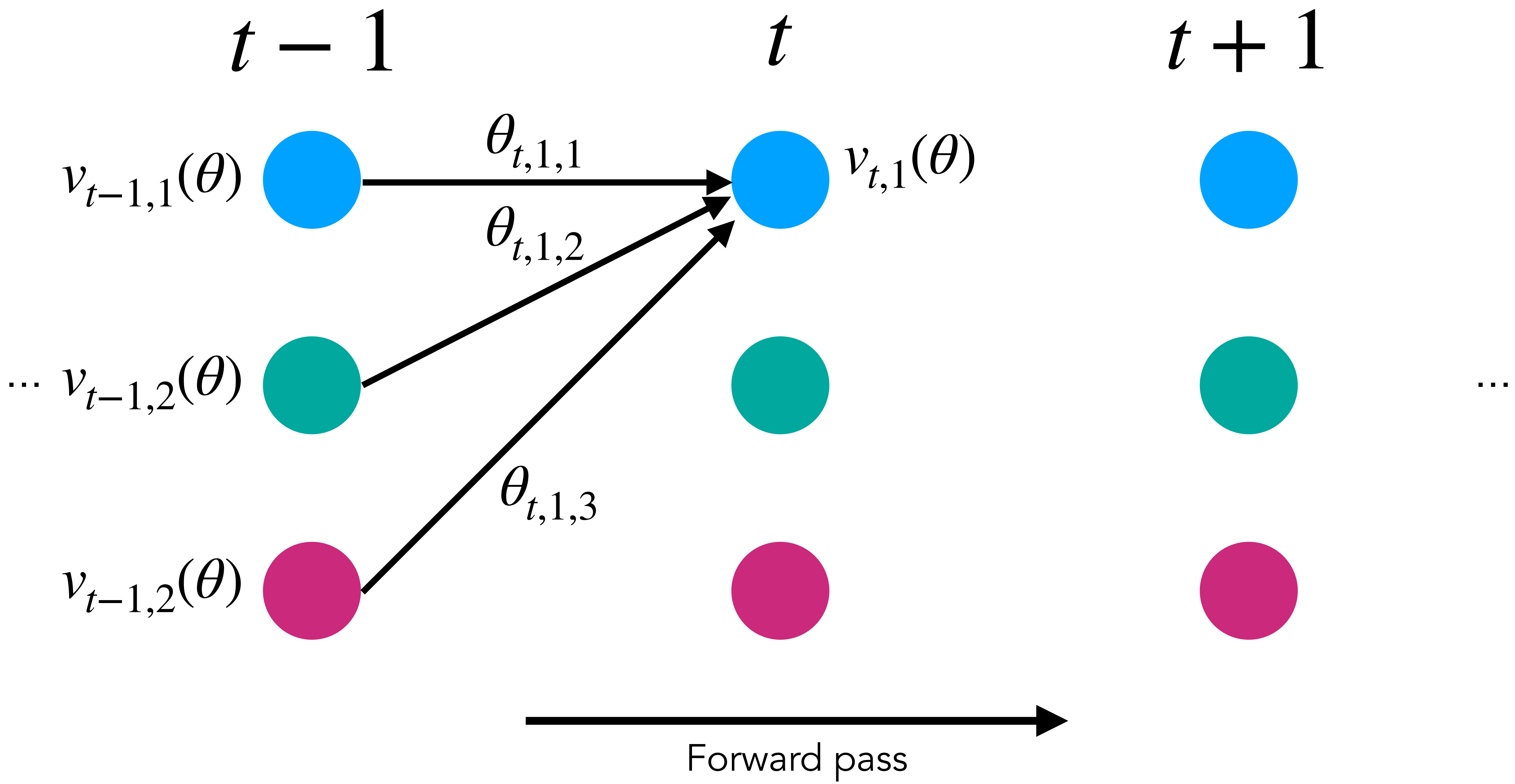
Can be computed efficiently by dynamic programming in the case of DAGs (no cycle)

MAP(θ) is a discontinuous function

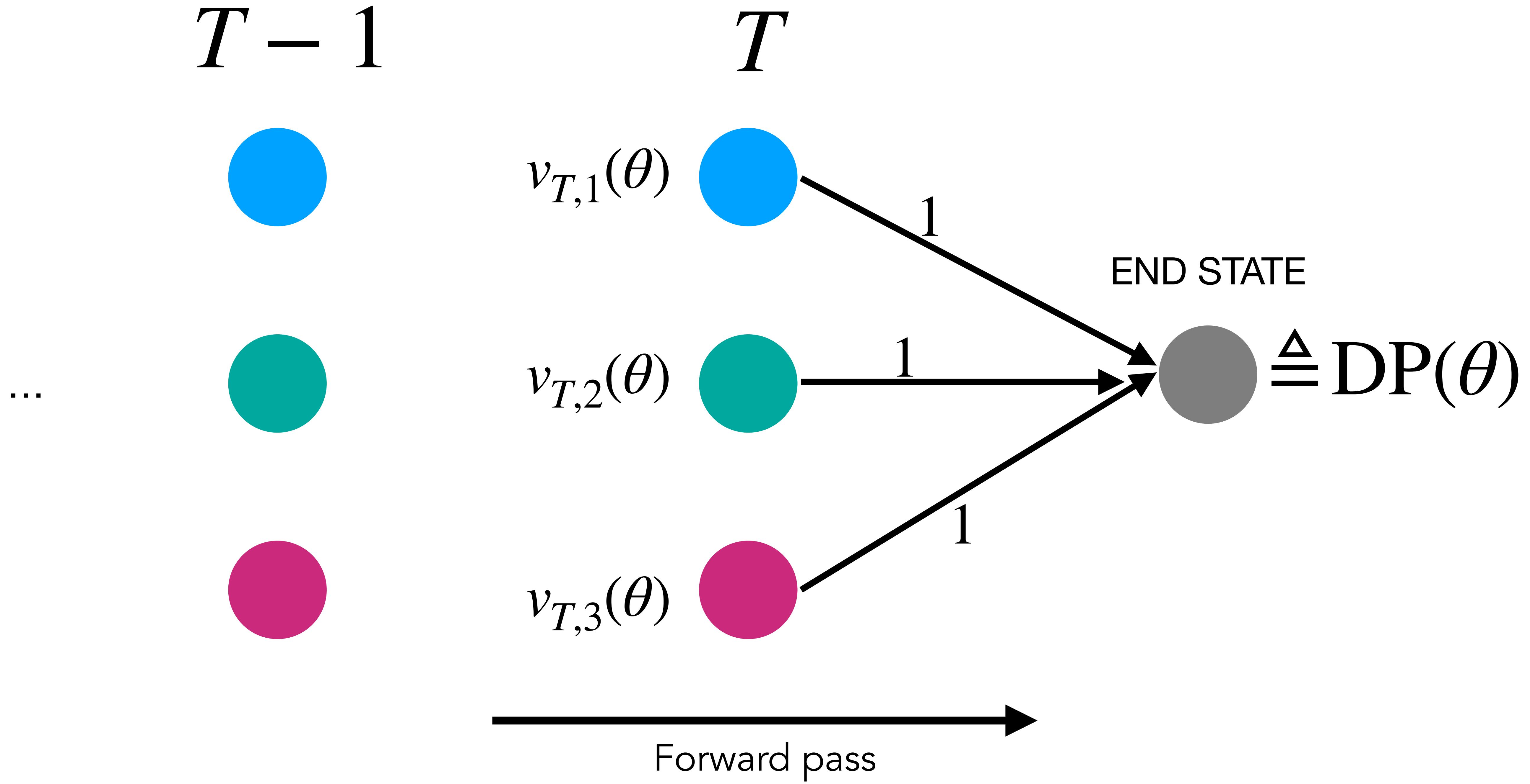
Bellman's recursion

Best value in state i up to time t

$$v_{t,i}(\theta) = \max_{j} v_{t-1,j}(\theta) + \theta_{t,i,j}$$

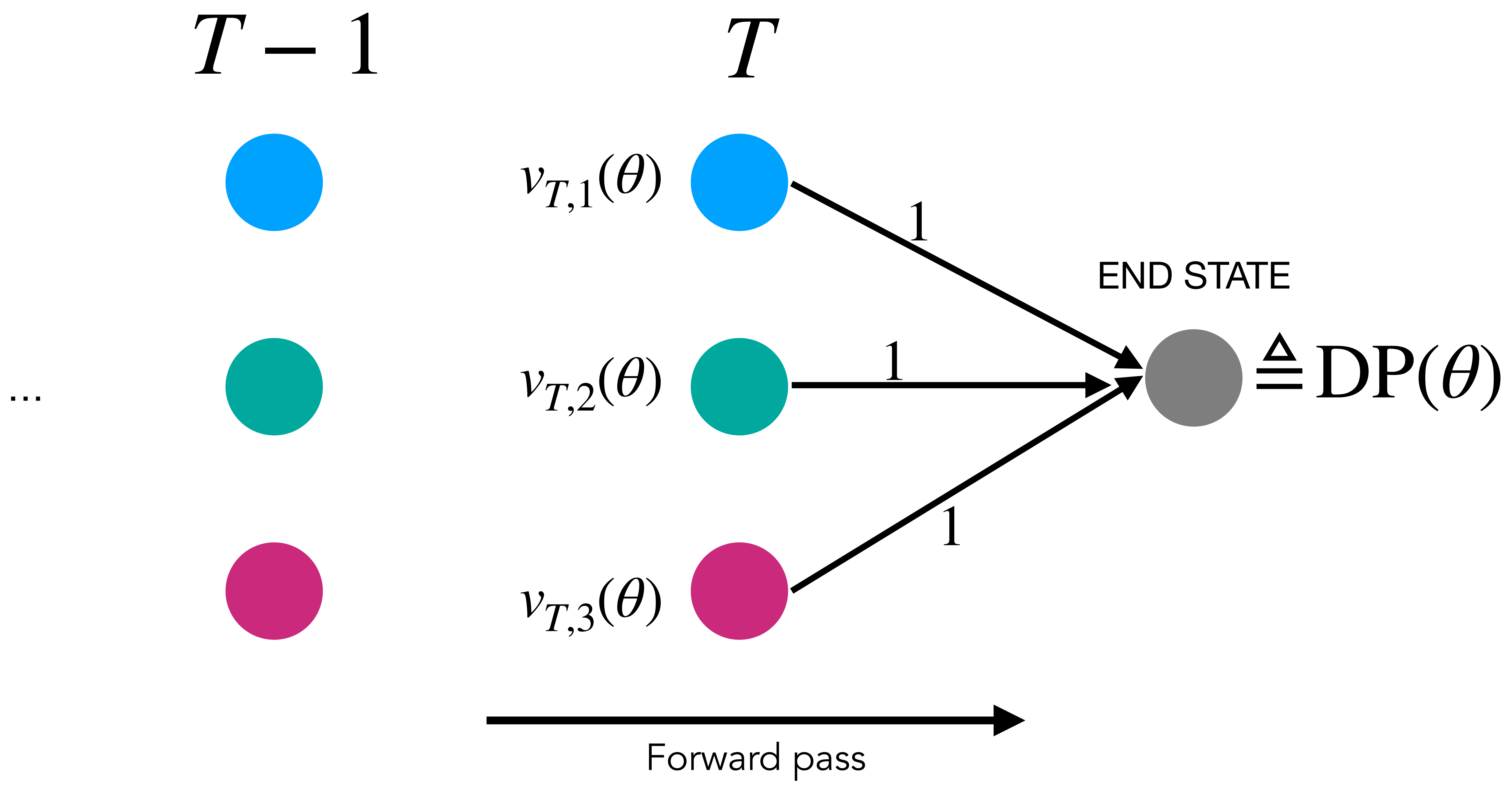


DP value and optimality



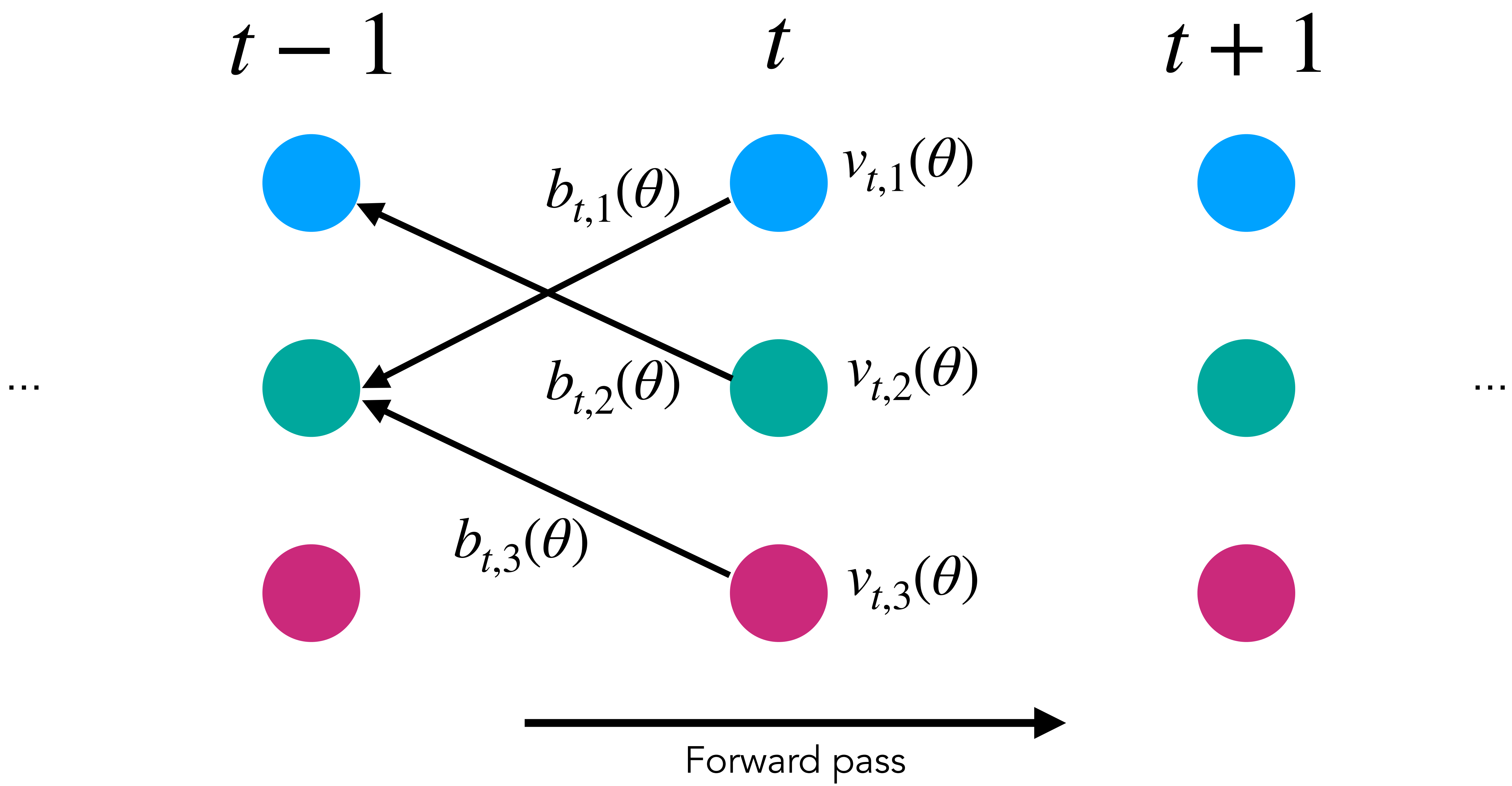
DP value and optimality

Optimality: $\mathrm{DP}(\theta) = \max_{y \in \mathcal{Y}} \langle y, \theta \rangle \in \mathbb{R}$



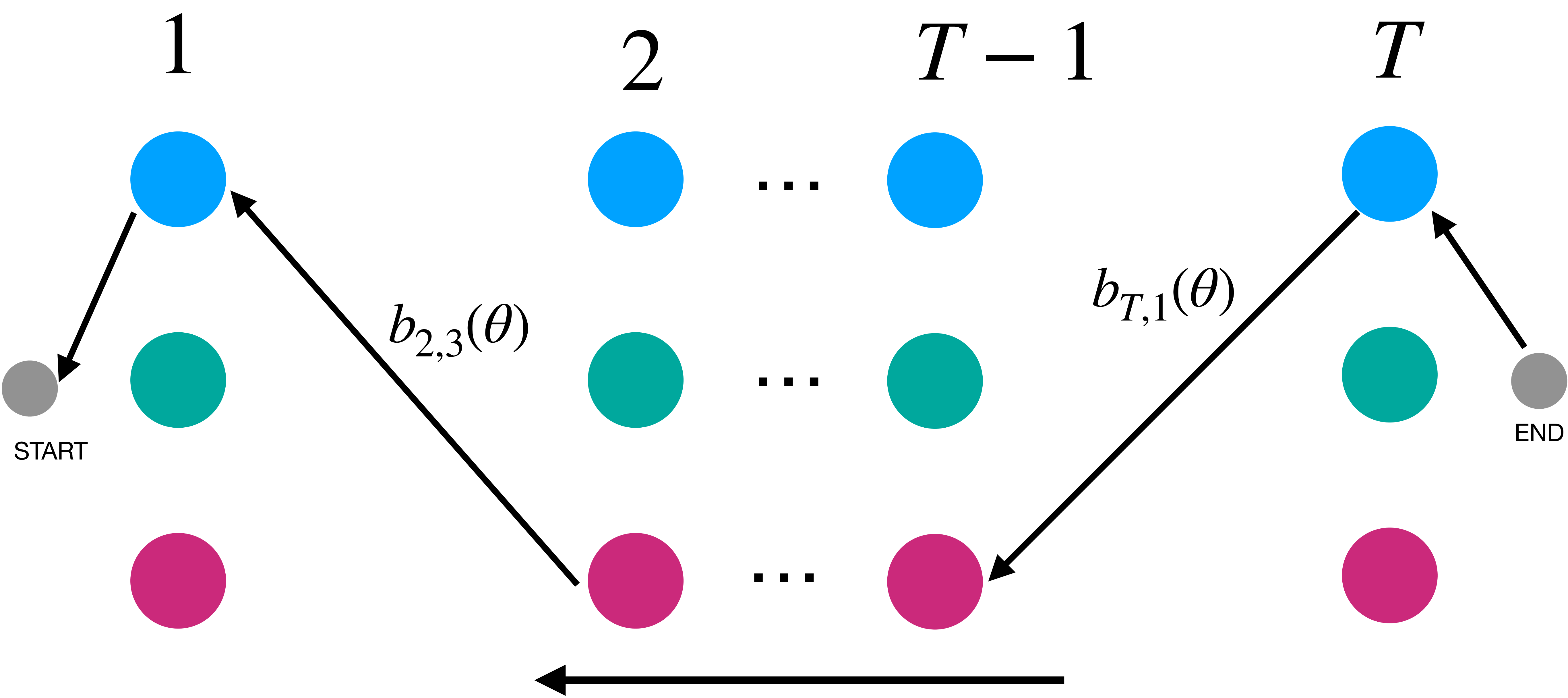
Maintaining back pointers

$$b_{t,i}(\theta) = \arg\max_j \; v_{t-1,j}(\theta) + \theta_{t,i,j} \in [S]$$



Backtracking

Optimal path equals $\textbf{MAP}(\theta) = \arg\max_{y \in \mathcal{Y} \subseteq \mathbb{R}^m} \langle y, \theta \rangle$

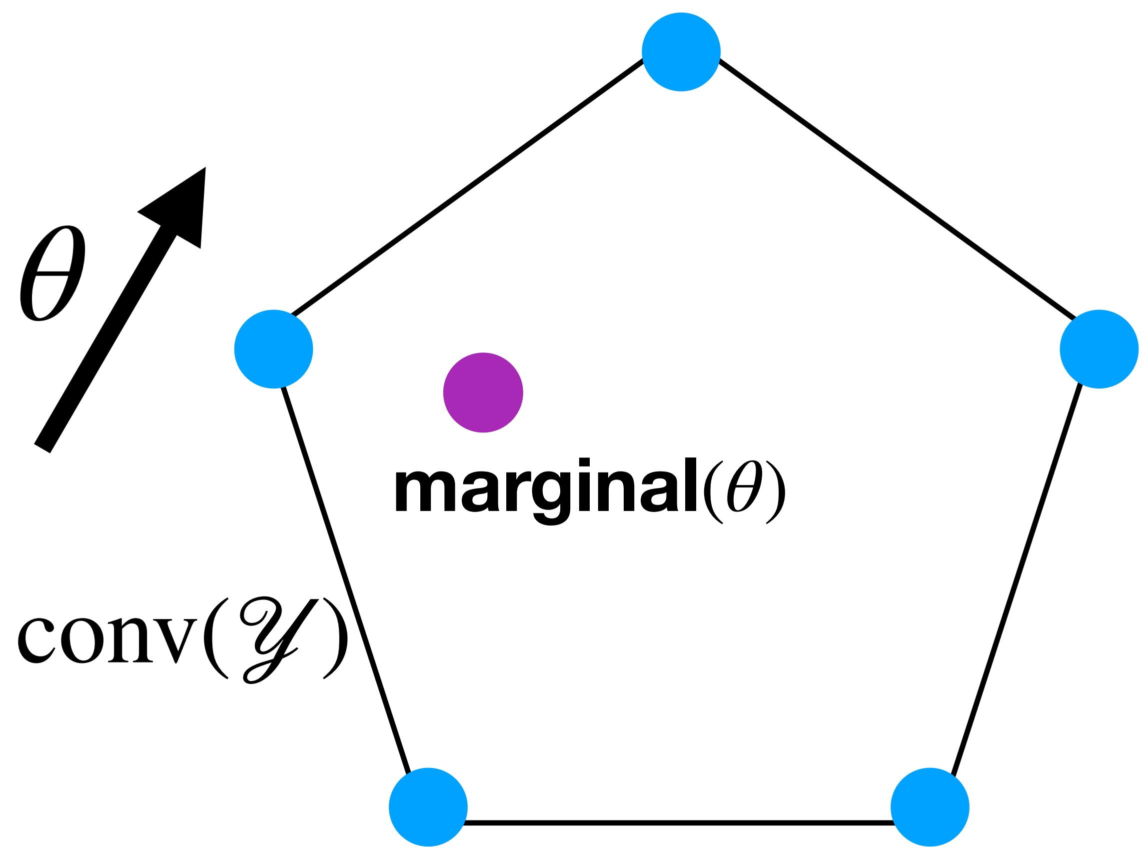


Start from end state and follow back pointers

Marginal inference: Expected Structure

Gibbs distribution

$$\textbf{marginal}(\theta) \triangleq \mathbb{E}_p[Y] \qquad p = \textbf{softmax}\left((\langle y, \theta \rangle)_{y \in \mathcal{Y}}\right) \in \triangle^{|\mathcal{Y}|}$$

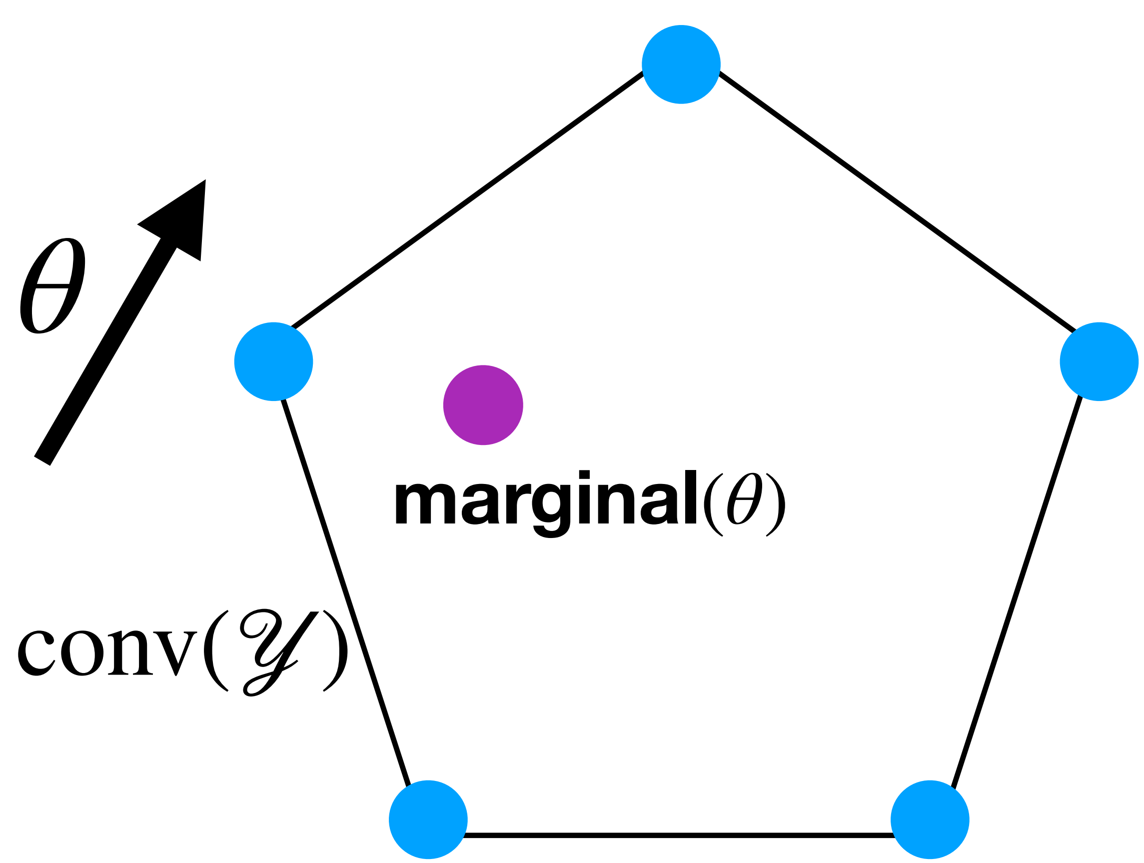


Marginal polytope
(Wainwright & Jordan, 2008)

Marginal inference: Expected Structure

Gibbs distribution

$\textbf{marginal}(\theta) \triangleq \mathbb{E}_p[Y] \qquad p = \textbf{softmax}\left((\langle y, \theta \rangle)_{y \in \mathcal{Y}}\right) \in \triangle^{|\mathcal{Y}|}$



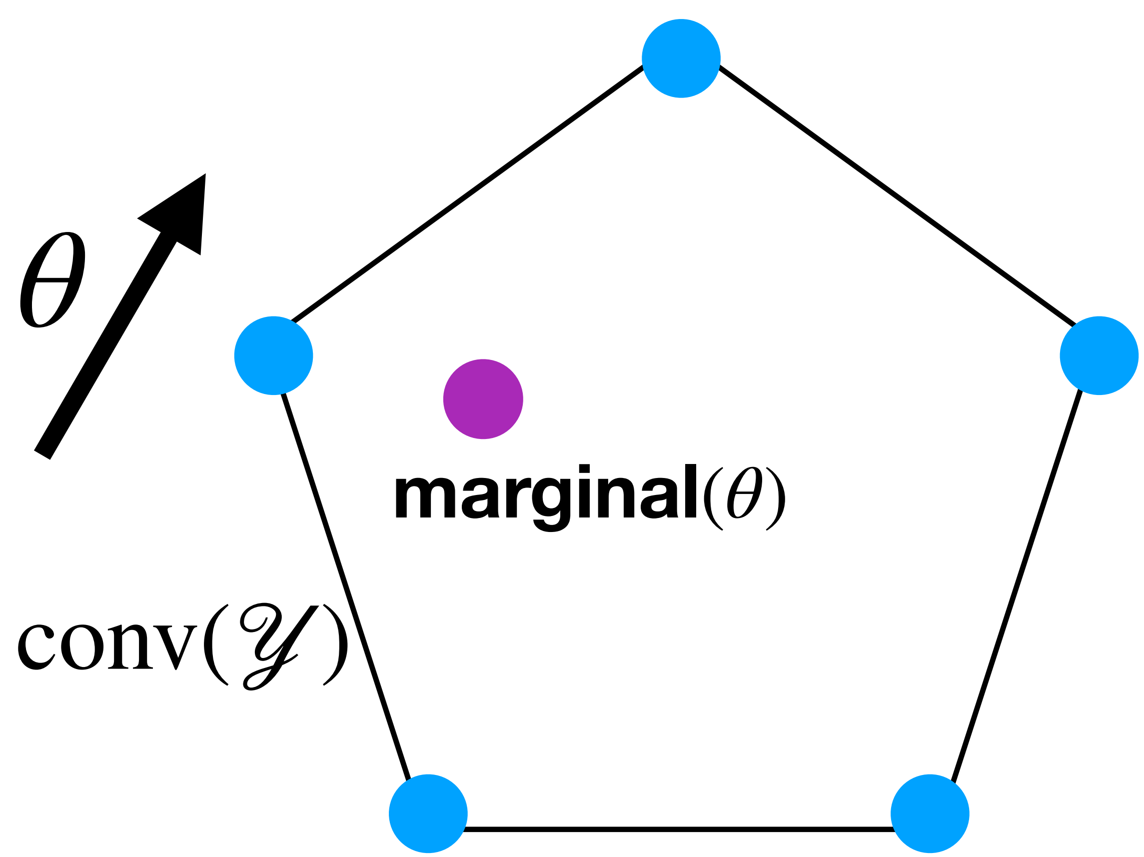
Marginal polytope
(Wainwright & Jordan, 2008)

Differentiable **but** **completely dense**
(always in the interior of the polytope)

Marginal inference: Expected Structure

Gibbs distribution

$$\textbf{marginal}(\theta) \triangleq \mathbb{E}_p[Y] \qquad p = \textbf{softmax}\left((\langle y, \theta \rangle)_{y \in \mathcal{Y}}\right) \in \triangle^{|\mathcal{Y}|}$$



Marginal polytope
(Wainwright & Jordan, 2008)

Differentiable but **completely dense**
(always in the interior of the polytope)

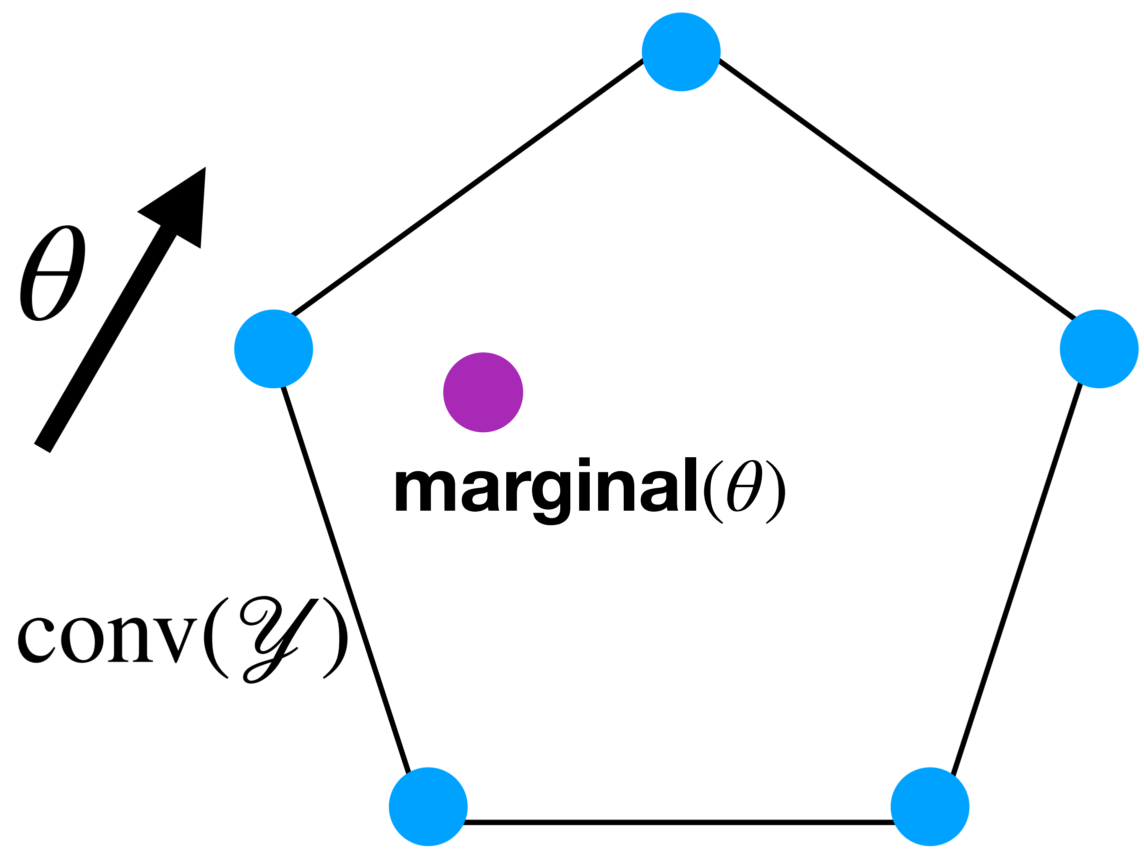
Computation: change semiring

$$x \to e^x \quad (\max, +) \to (+, \times)$$

Marginal inference: Expected Structure

Gibbs distribution

$$\textbf{marginal}(\theta) \triangleq \mathbb{E}_p[Y] \qquad p = \textbf{softmax}\left((\langle y, \theta\rangle)_{y\in\mathcal{Y}}\right) \in \triangle^{|\mathcal{Y}|}$$



Marginal polytope
(Wainwright & Jordan, 2008)

Differentiable but **completely dense**
(always in the interior of the polytope)

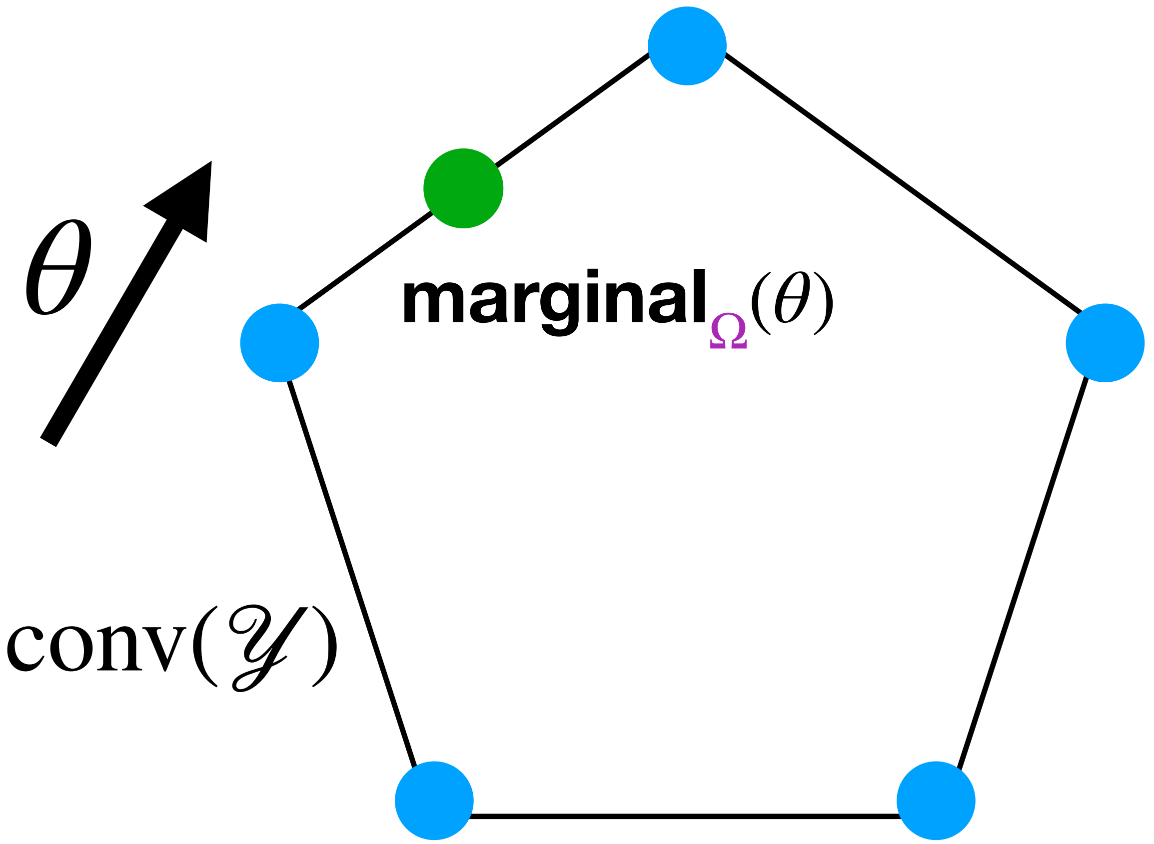
Computation: change semiring

$$x \to e^x \quad (\max, +) \to (+, \times)$$

Viterbi → Forward-Backward
CKY → Inside-Outside
DTW → Soft-DTW
max-sum → sum-product (BP)

Sparse marginal inference?

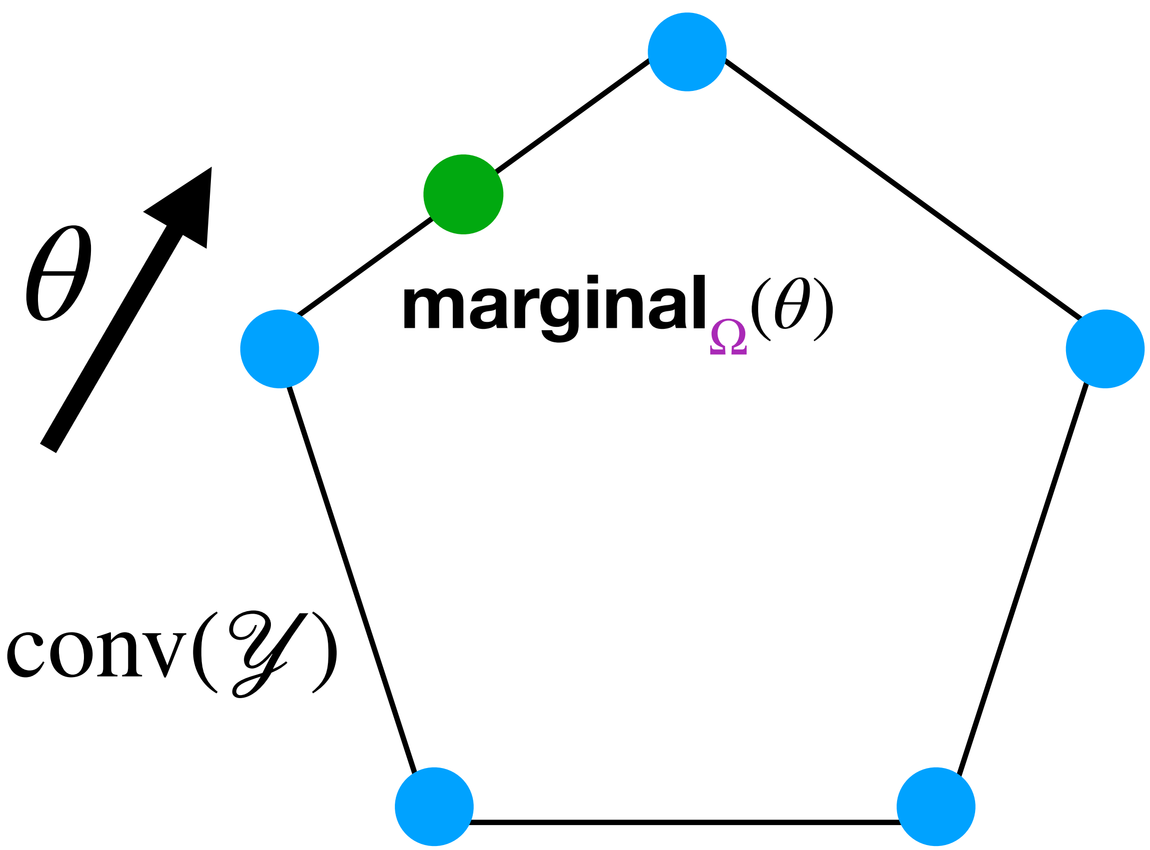
$\textbf{marginal}_{\Omega}(\theta) \triangleq \mathbb{E}_p[Y] \quad p = \textbf{argmax}_{\Omega}\left((\langle y, \theta \rangle)_{y \in \mathcal{Y}}\right) \in \triangle^{|\mathcal{Y}|}$



Sparse marginal inference?

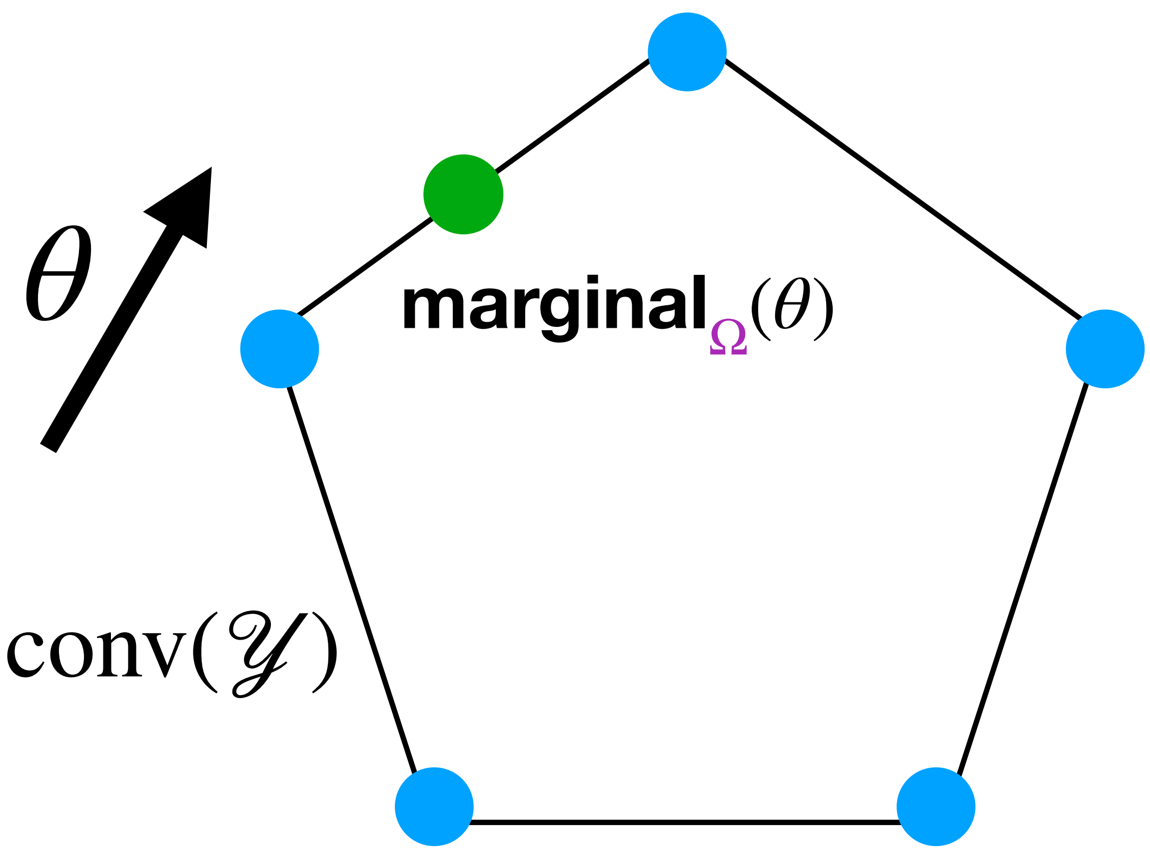
$$\textbf{marginal}_{\Omega}(\theta) \triangleq \mathbb{E}_p[Y] \quad p = \textbf{argmax}_{\Omega}\left((\langle y, \theta \rangle)_{y \in \mathcal{Y}}\right) \in \triangle^{|\mathcal{Y}|}$$

Can we use $\Omega(p) = \frac{1}{2}\|p\|^2$?



Sparse marginal inference?

$$\textbf{marginal}_{\Omega}(\theta) \triangleq \mathbb{E}_p[Y] \quad p = \textbf{argmax}_{\Omega}\left((\langle y, \theta \rangle)_{y \in \mathcal{Y}}\right) \in \triangle^{|\mathcal{Y}|}$$



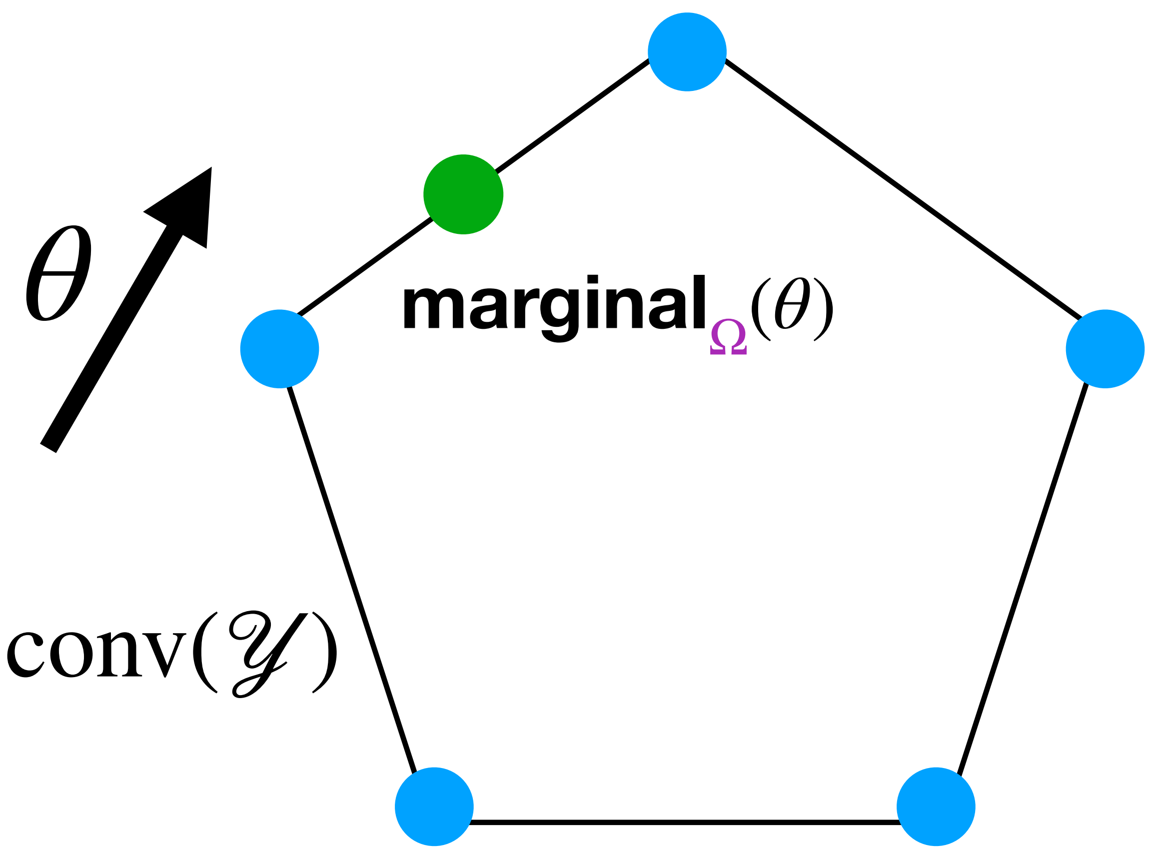
Can we use $\Omega(p) = \frac{1}{2}\|p\|^2$?

No longer a semiring change in general

Sparse marginal inference?

$$\textbf{marginal}_{\Omega}(\theta) \triangleq \mathbb{E}_p[Y] \quad p = \textbf{argmax}_{\Omega}\left((\langle y, \theta\rangle)_{y\in\mathcal{Y}}\right) \in \triangle^{|\mathcal{Y}|}$$

Can we use $\Omega(p) = \frac{1}{2}\|p\|^2$?



No longer a semiring change in general

Difficult to compute exactly

Our proposal for differentiable DP

Mensch & Blondel, ICML 2018

Our proposal for differentiable DP

- Based on the novel viewpoint of **smoothed max operators**

Mensch & Blondel, ICML 2018

Our proposal for differentiable DP

- Based on the novel viewpoint of **smoothed max operators**
- Works for any **shortest path** problem over a **DAG**

Mensch & Blondel, ICML 2018

Our proposal for differentiable DP

- Based on the novel viewpoint of **smoothed max operators**
- Works for any **shortest path** problem over a **DAG**
- Enjoys **same big-O complexity** as regular DP

Mensch & Blondel, ICML 2018

Our proposal for differentiable DP

- Based on the novel viewpoint of **smoothed max operators**
- Works for any **shortest path** problem over a **DAG**
- Enjoys **same big-O complexity** as regular DP
- **Sparse solutions** when using quadratic regularization

Mensch & Blondel, ICML 2018

Our proposal for differentiable DP

- Based on the novel viewpoint of **smoothed max operators**
- Works for any **shortest path** problem over a **DAG**
- Enjoys **same big-O complexity** as regular DP
- **Sparse solutions** when using quadratic regularization
- **Probabilistic interpretation**

Mensch & Blondel, ICML 2018

Our proposal for differentiable DP

- Based on the novel viewpoint of **smoothed max operators**
- Works for any **shortest path** problem over a **DAG**
- Enjoys **same big-O complexity** as regular DP
- **Sparse solutions** when using quadratic regularization
- **Probabilistic interpretation**
- **Unified** and **numerically stable** implementation (computations directly in log-domain!)

Mensch & Blondel, ICML 2018

Smoothed max operators

Smoothed max operators

Recall the definition of differentiable **argmax** operator

$$\mathbf{argmax}_{\Omega}(\theta) \triangleq \arg\max_{p \in \triangle^m} \langle p, \theta \rangle - \Omega(p) \in \triangle^m$$

Smoothed max operators

Recall the definition of differentiable **argmax** operator

$$\mathbf{argmax}_{\Omega}(\theta) \triangleq \arg\max_{p \in \triangle^m} \langle p, \theta \rangle - \Omega(p) \in \triangle^m$$

Similarly we define the smoothed **max** operator (Nesterov, 2005)

$$\max_{\Omega}(\theta) \triangleq \max_{p \in \triangle^m} \langle p, \theta \rangle - \Omega(p) \in \mathbb{R}$$

Smoothed max operators

Recall the definition of differentiable **argmax** operator

$$\textbf{argmax}_{\Omega}(\theta) \triangleq \arg\max_{p \in \triangle^m} \langle p, \theta \rangle - \Omega(p) \in \triangle^m$$

Similarly we define the smoothed **max** operator (Nesterov, 2005)

$$\max_{\Omega}(\theta) \triangleq \max_{p \in \triangle^m} \langle p, \theta \rangle - \Omega(p) \in \mathbb{R}$$

From the duality between smoothness and strong convexity

Strongly convex Ω over Δ $\Leftrightarrow$ Smooth $\max_{\Omega}$

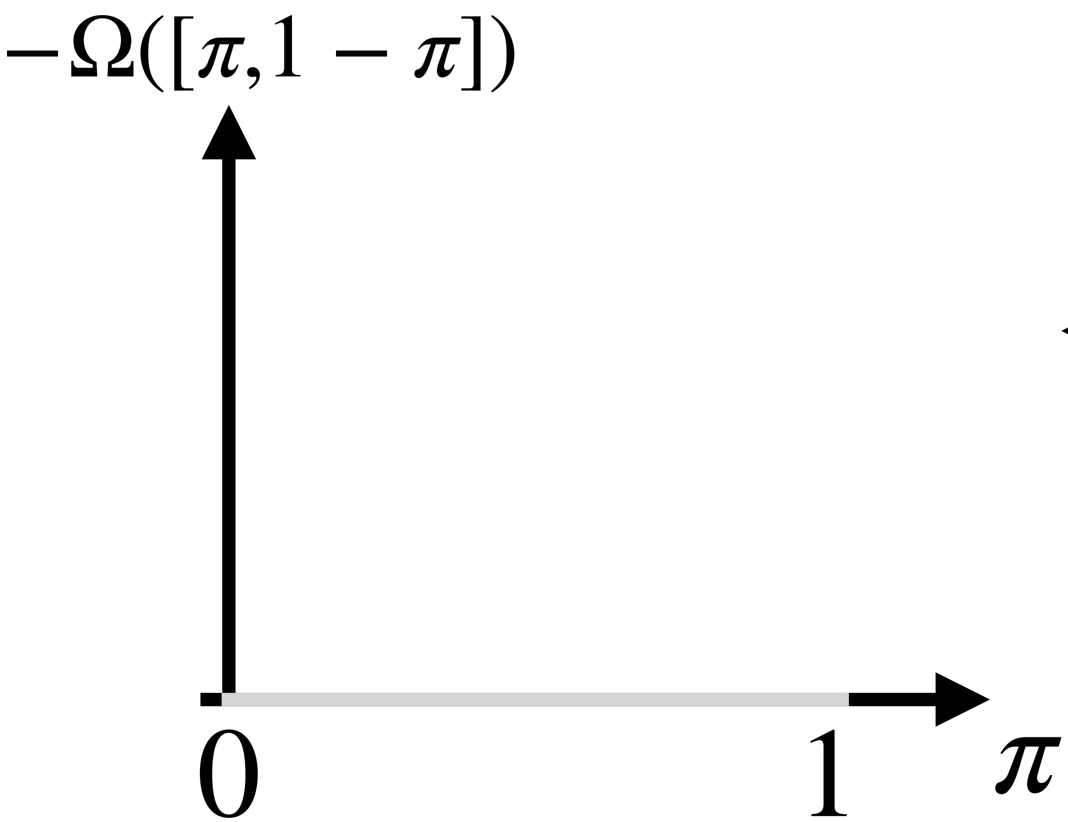
Examples

$$\max_\Omega(\theta) \triangleq \max_{p \in \triangle^m} \langle p, \theta \rangle - \Omega(p)$$

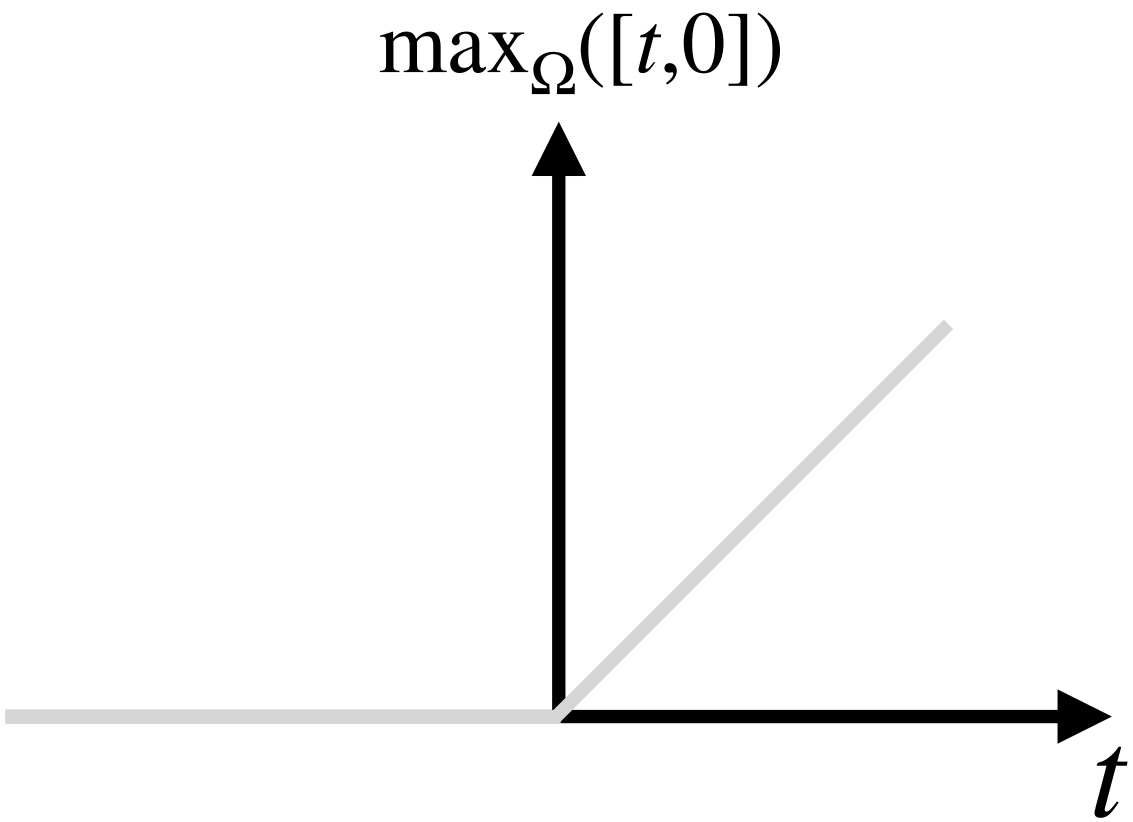
Unregularized

$$\Omega(p) = 0$$

Regularization



Smoothed max



Examples

$$\max_{\Omega}(\theta) \triangleq \max_{p \in \triangle^m} \langle p, \theta \rangle - \Omega(p)$$

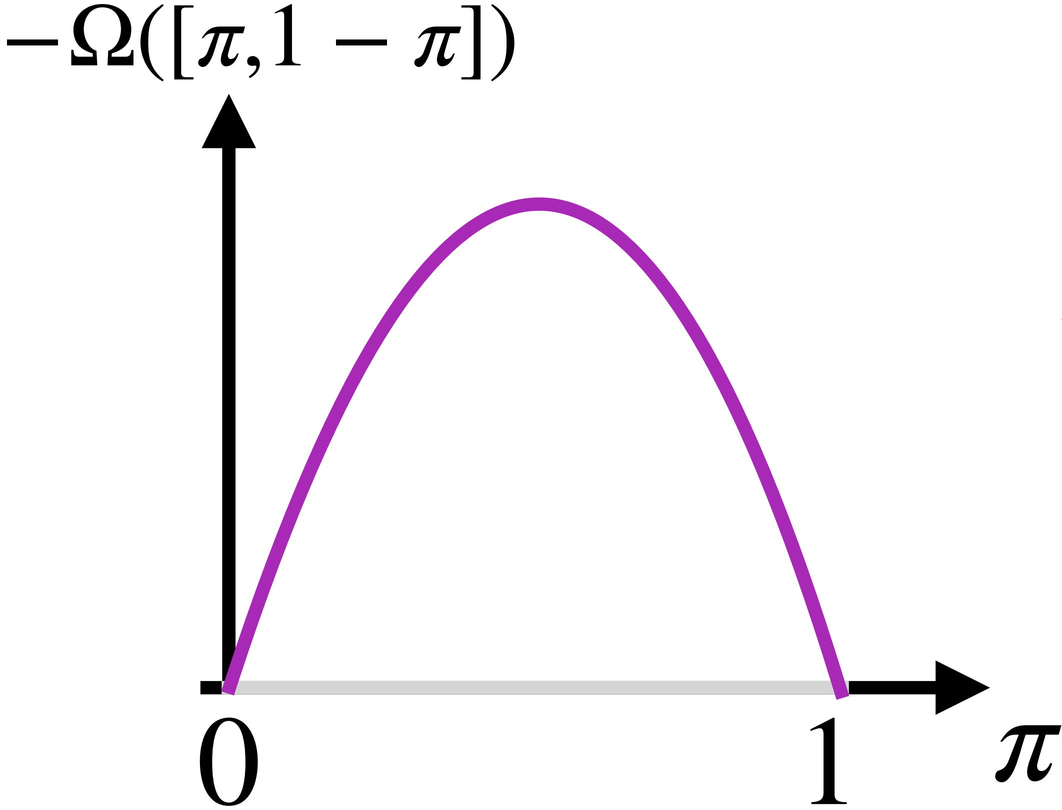
Unregularized

$$\Omega(p) = 0$$

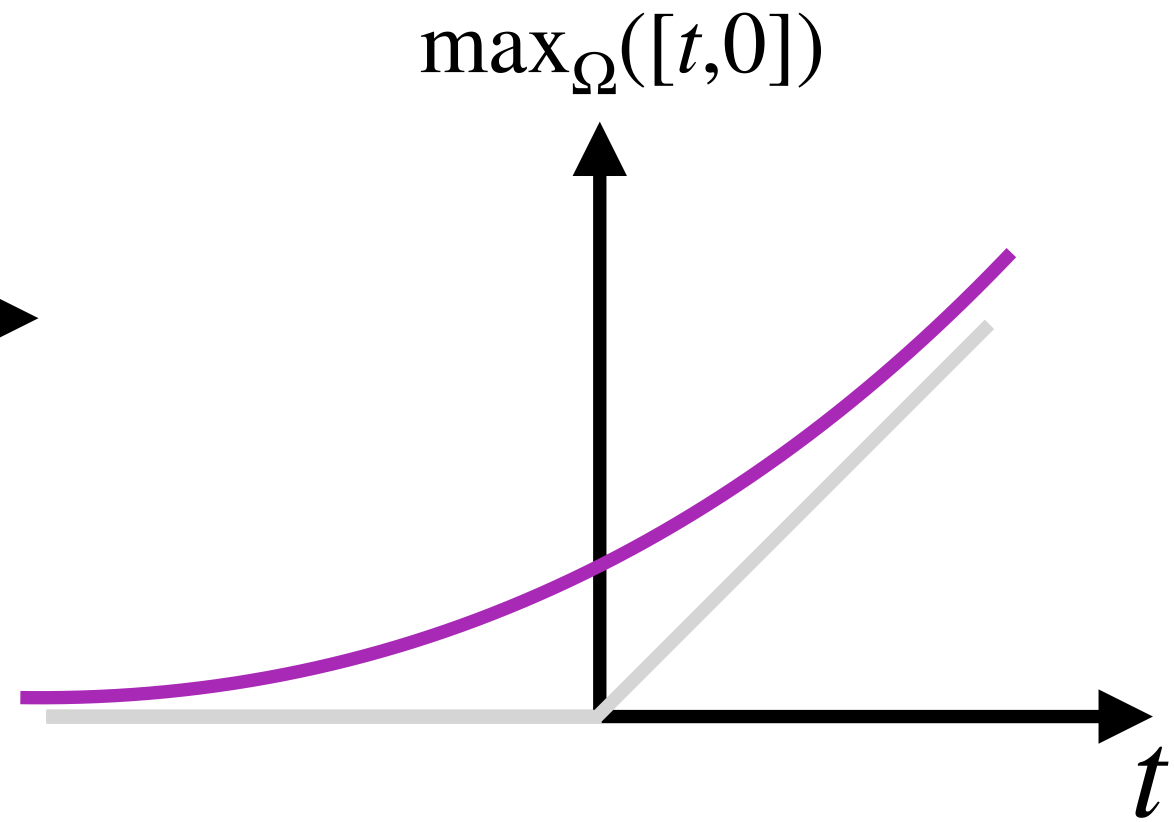
Shannon (negative) entropy

$$\Omega(p) = \sum_i p_i \log p_i$$

Regularization



Smoothed max



Examples

$$\max_\Omega(\theta) \triangleq \max_{p \in \triangle^m} \langle p, \theta \rangle - \Omega(p)$$

Unregularized

$$\Omega(p) = 0$$

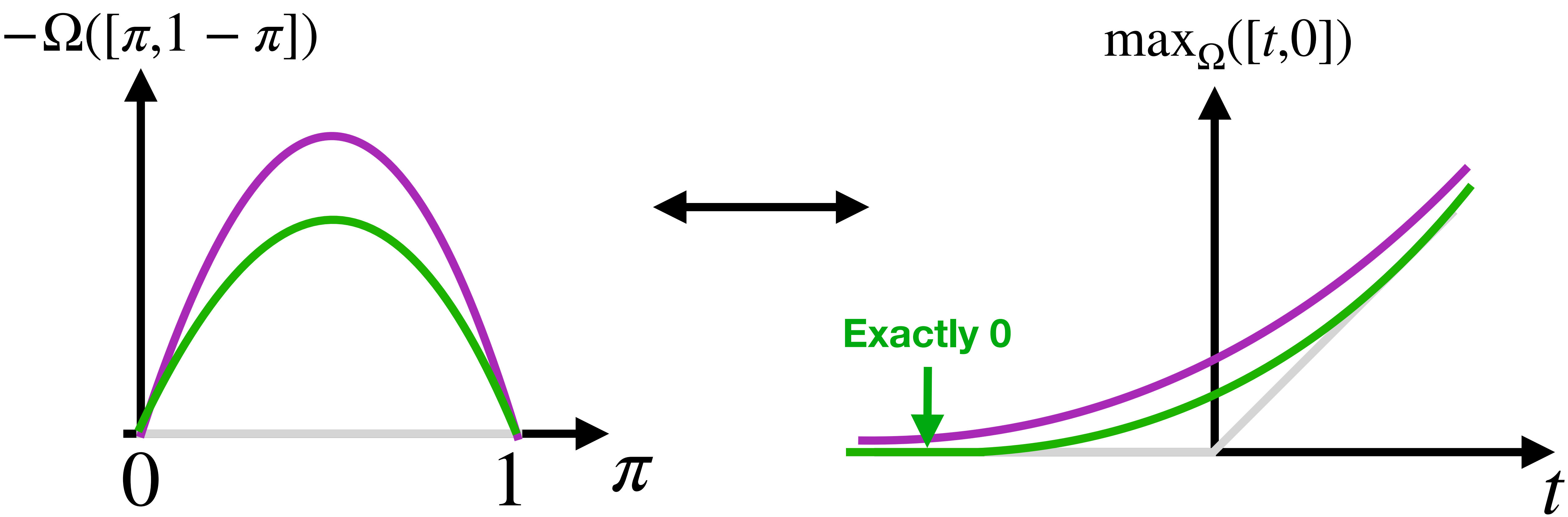
Shannon (negative) entropy

$$\Omega(p) = \sum_i p_i \log p_i$$

Gini (negative) index

$$\Omega(p) = \frac{1}{2}(\|p\|^2 - 1)$$

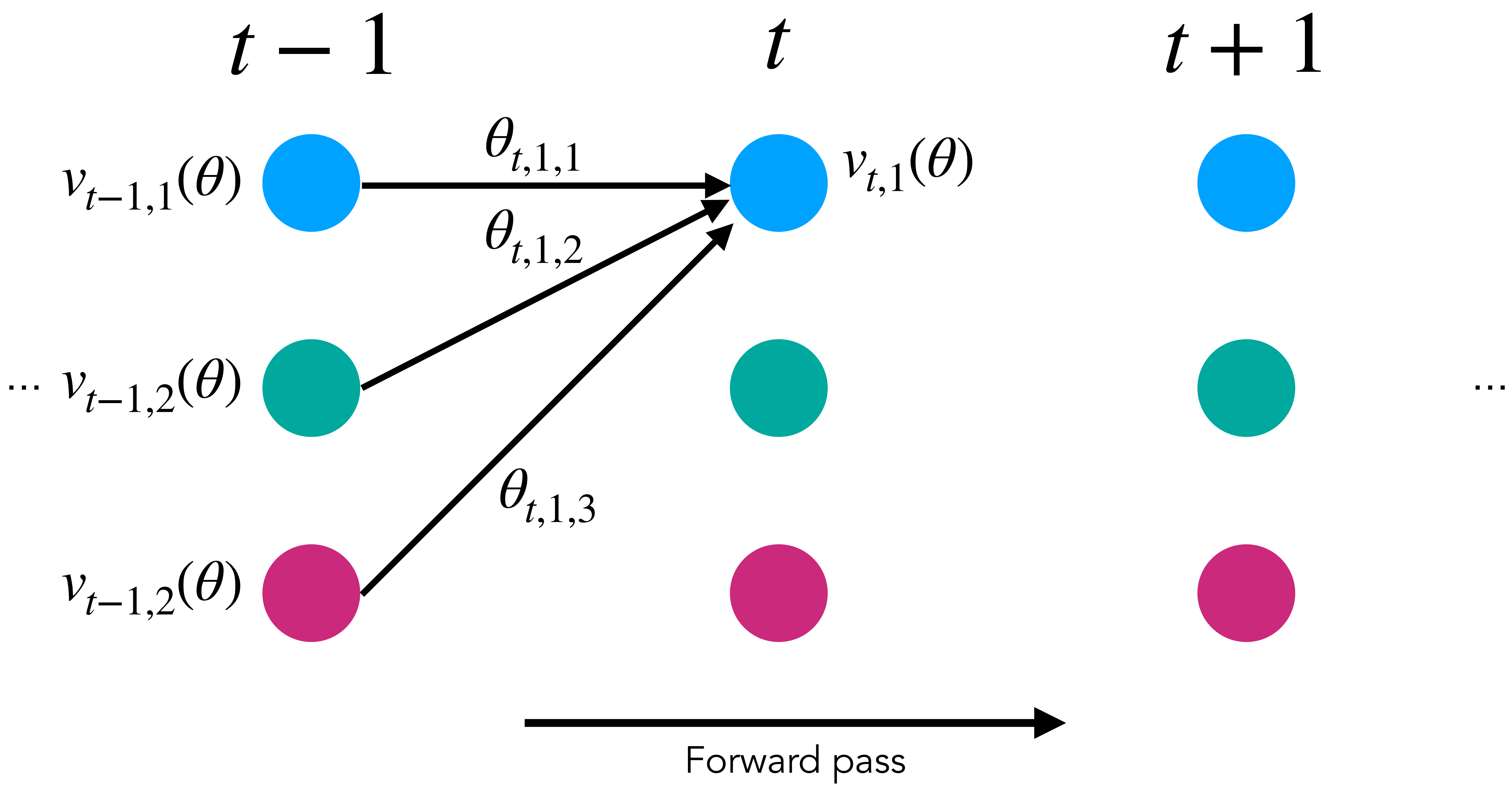
Regularization **Smoothed max**



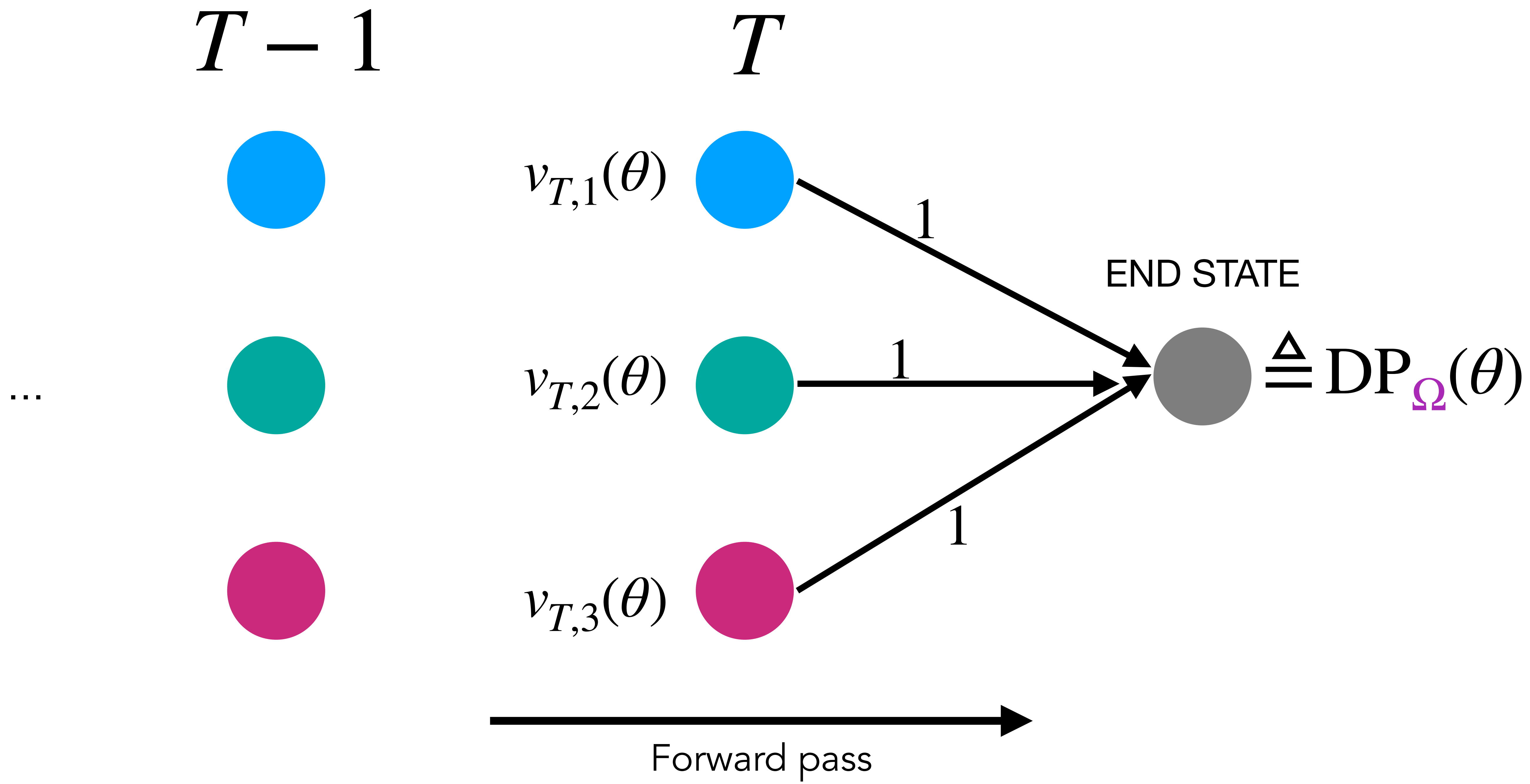
Smoothed Bellman's recursion

(max, +)
↓
(max$_\Omega$, +)

$$v_{t,i}(\theta) = \max_\Omega((v_{t-1,j}(\theta) + \theta_{t,i,j})_{j\in[S]})$$

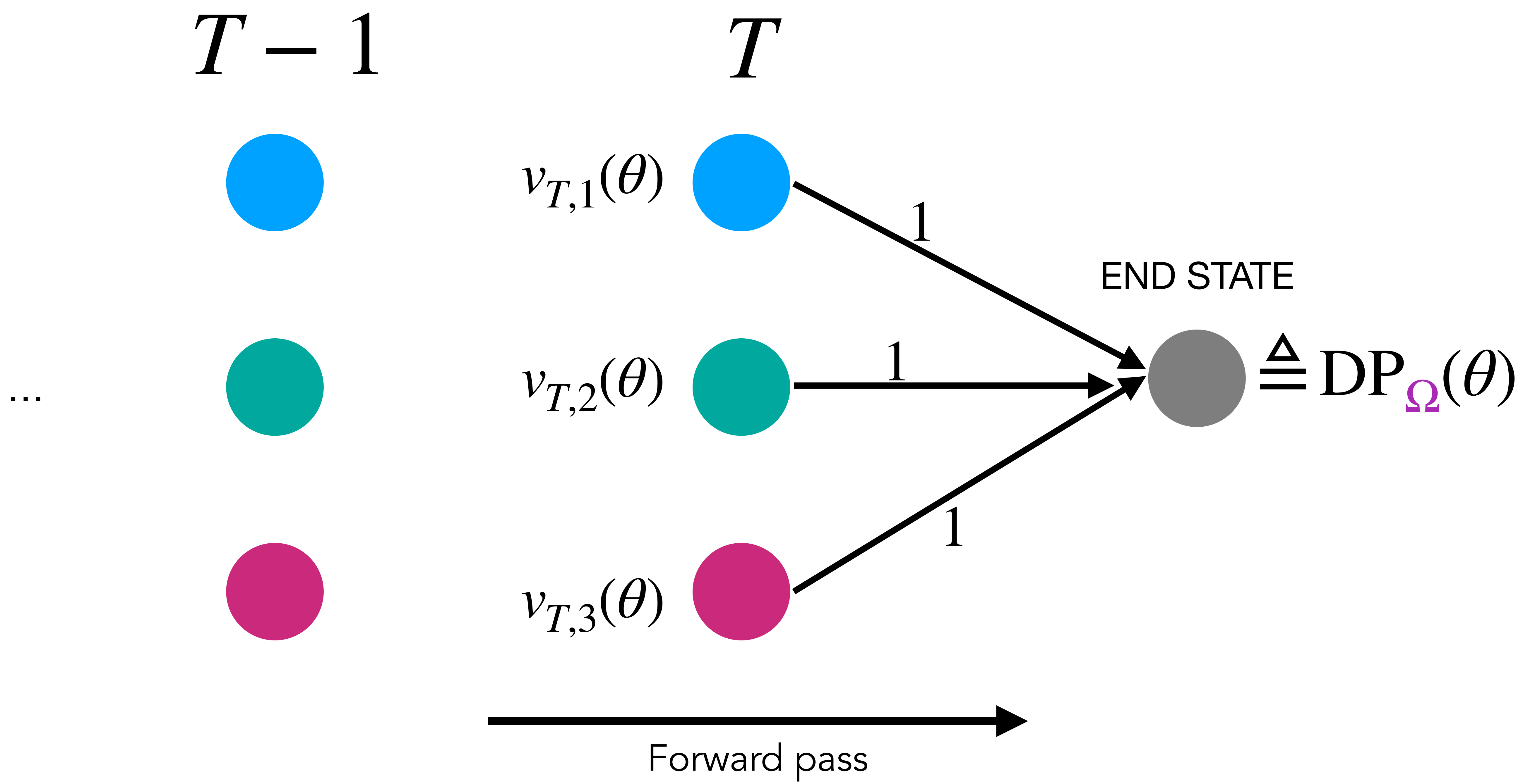


Smoothed DP value



Smoothed DP value

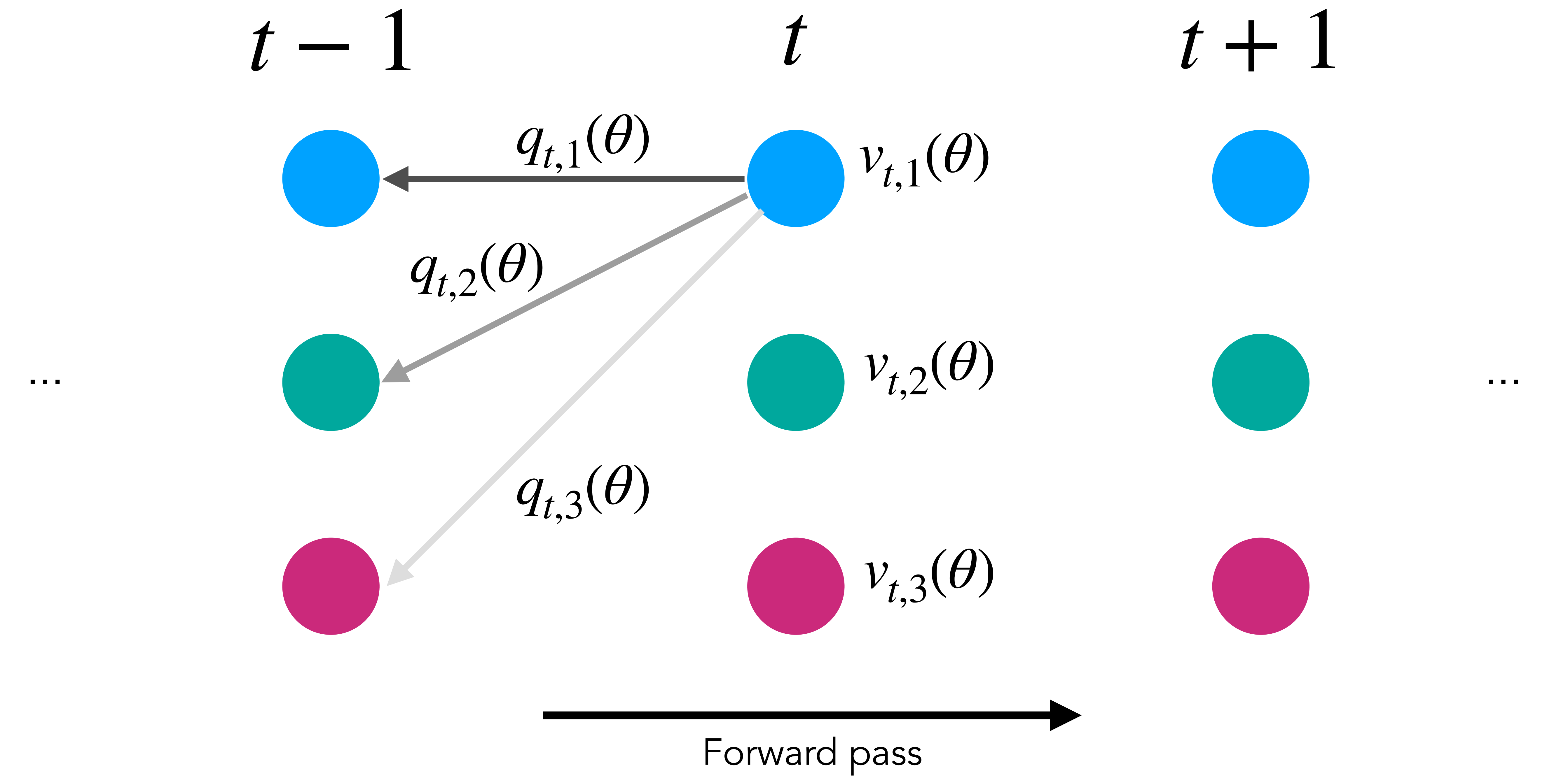
$$\mathrm{DP}_{\Omega}(\theta) \leq \max_{\Omega}((\langle y, \theta \rangle)_{y \in \mathcal{Y}})$$



Probabilistic backpointers

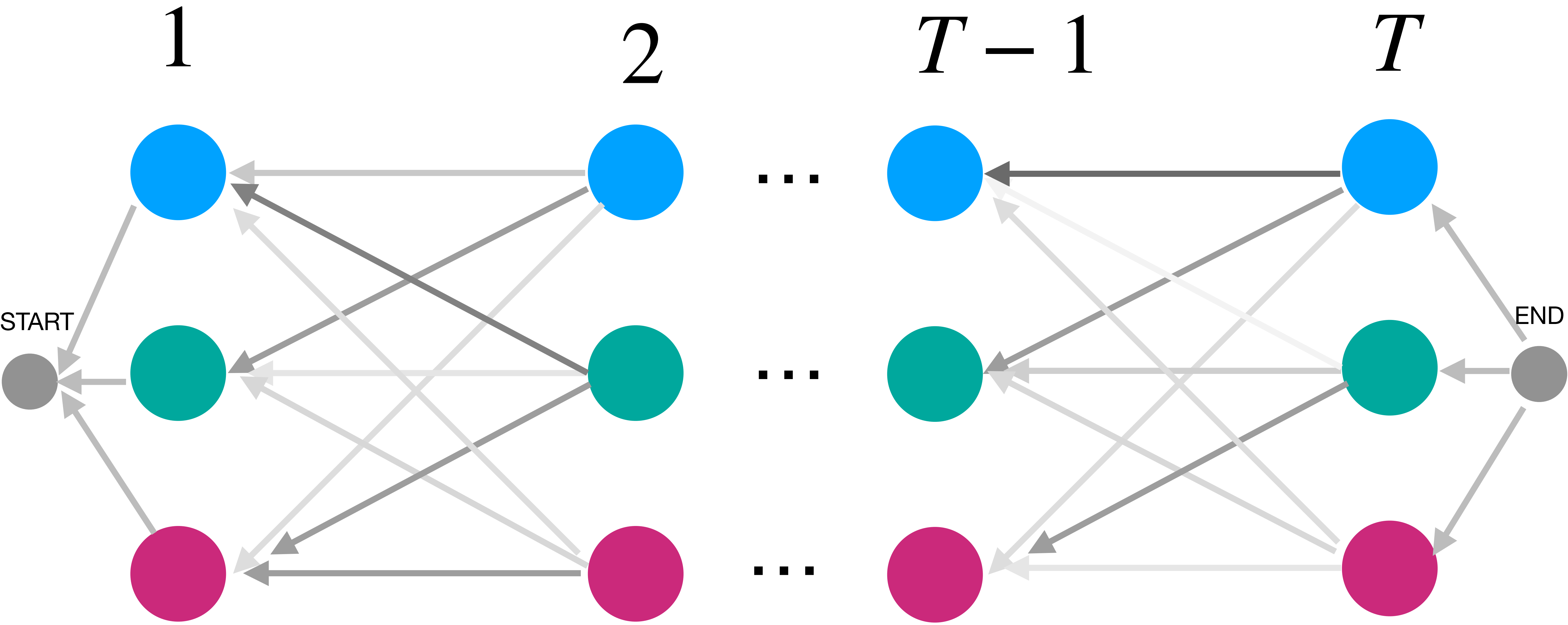
Replace back pointers with **distribution** over states

$$q_{t,i}(\theta) = \textbf{argmax}_{\Omega}((v_{t-1,j}(\theta) + \theta_{t,i,j})_{j \in [S]}) \in \triangle^S$$



Random walk

Random walk (finite Markov chain) defines
a distribution p over paths

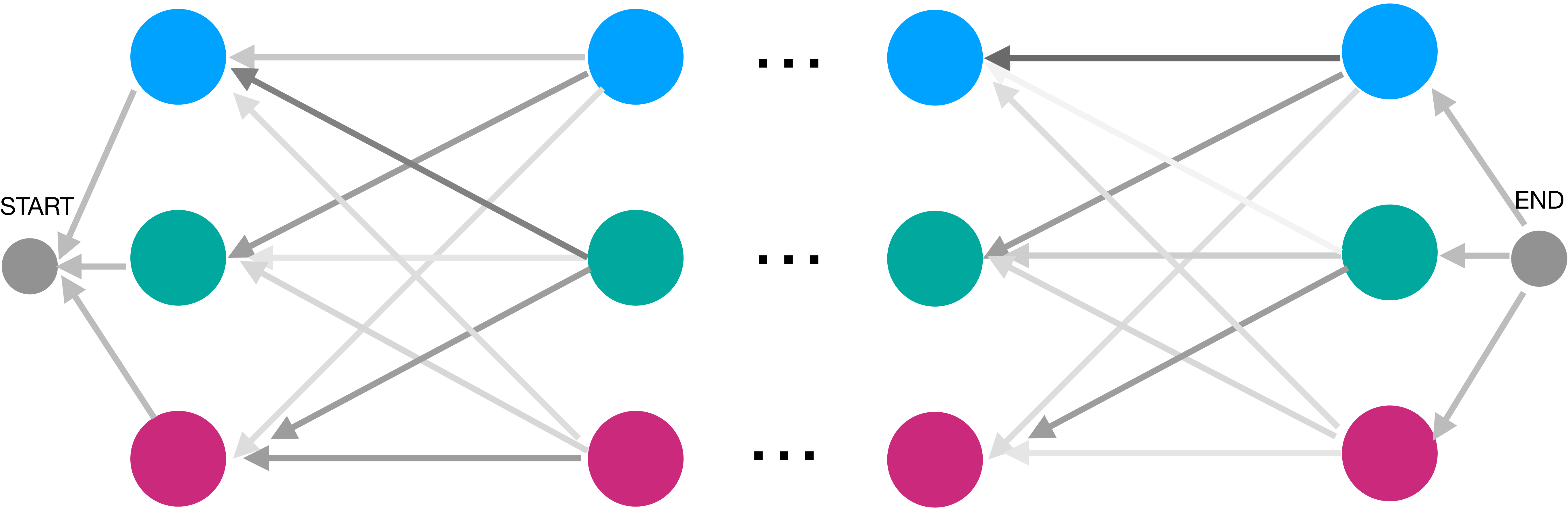


Each time step t has its own transition matrix $Q_t \in \mathbb{R}^{S \times S}$

Random walk

Sampling is easy.

How to compute **expectation** $\mathbb{E}_p[Y]$? T



Each time step t has its own transition matrix $Q_t \in \mathbb{R}^{S \times S}$

Gradient = Expected path

Proposition (Mensch & Blondel, 2018) (See also Eisner, 2016)

$$\nabla \mathrm{DP}_{\Omega}(\theta) = \mathbb{E}_p[Y] \in \mathrm{conv}(\mathscr{Y})$$

Gradient = Expected path

Proposition (Mensch & Blondel, 2018) (See also Eisner, 2016)

$$\nabla \mathrm{DP}_{\Omega}(\theta) = \mathbb{E}_p[Y] \in \mathrm{conv}(\mathcal{Y})$$

Can compute $\mathbb{E}_p[Y]$ at the same cost as computing $\mathrm{DP}_{\Omega}(\theta)$ by **backpropagation**

Gradient = Expected path

Proposition (Mensch & Blondel, 2018)   (See also Eisner, 2016)

$$\nabla \mathrm{DP}_{\Omega}(\theta) = \mathbb{E}_p[Y] \in \mathrm{conv}(\mathcal{Y})$$

Can compute $\mathbb{E}_p[Y]$ at the same cost as computing $\mathrm{DP}_{\Omega}(\theta)$ by **backpropagation**

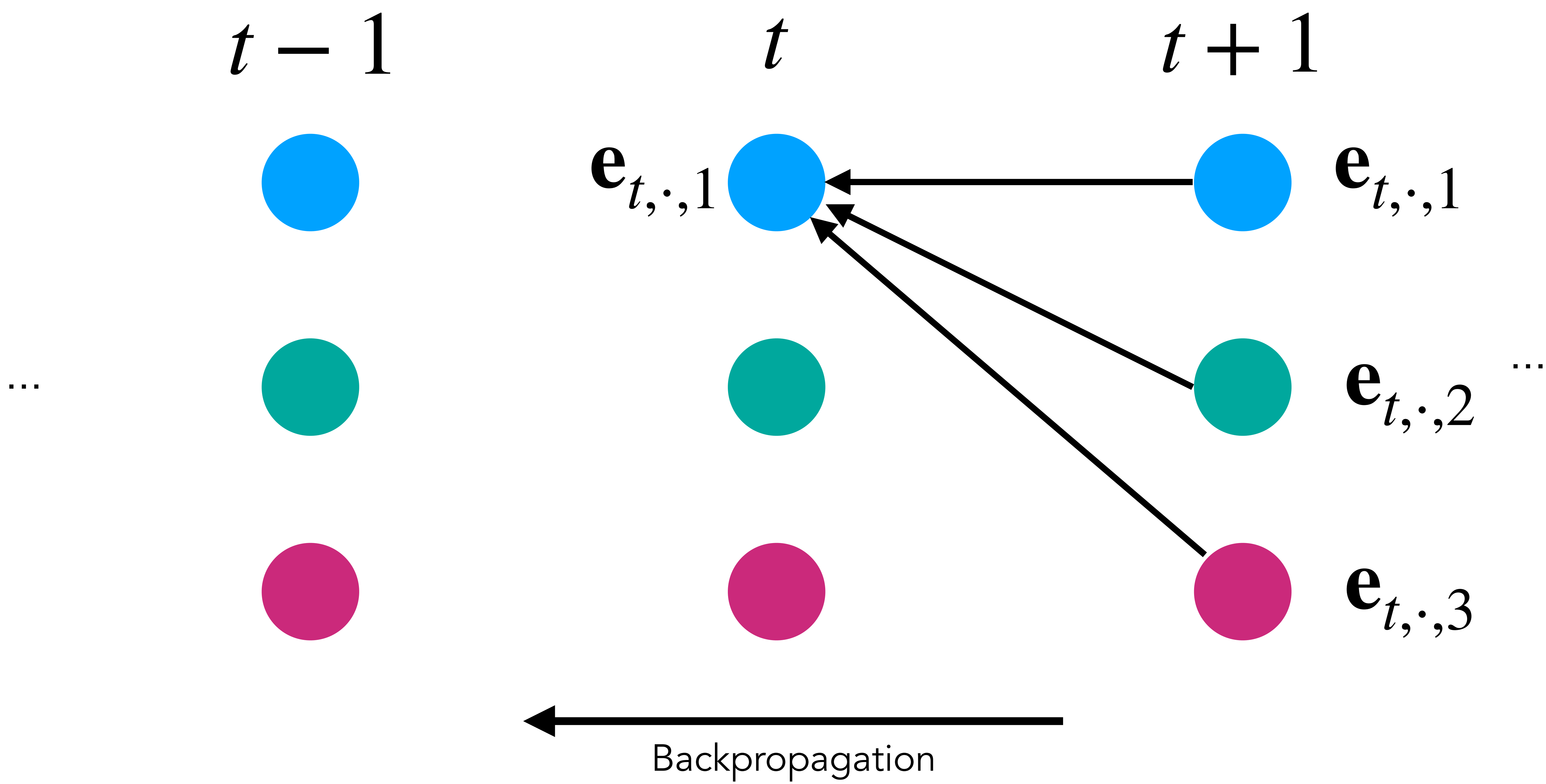
For Ω = negative entropy, we have

Intractable sum if computed naively

$$\nabla \mathrm{DP}_{\Omega}(\theta) = \mathbb{E}_p[Y] = \frac{\sum_{y \in \mathcal{Y}} \exp\langle y, \theta \rangle y}{Z(\theta)}$$

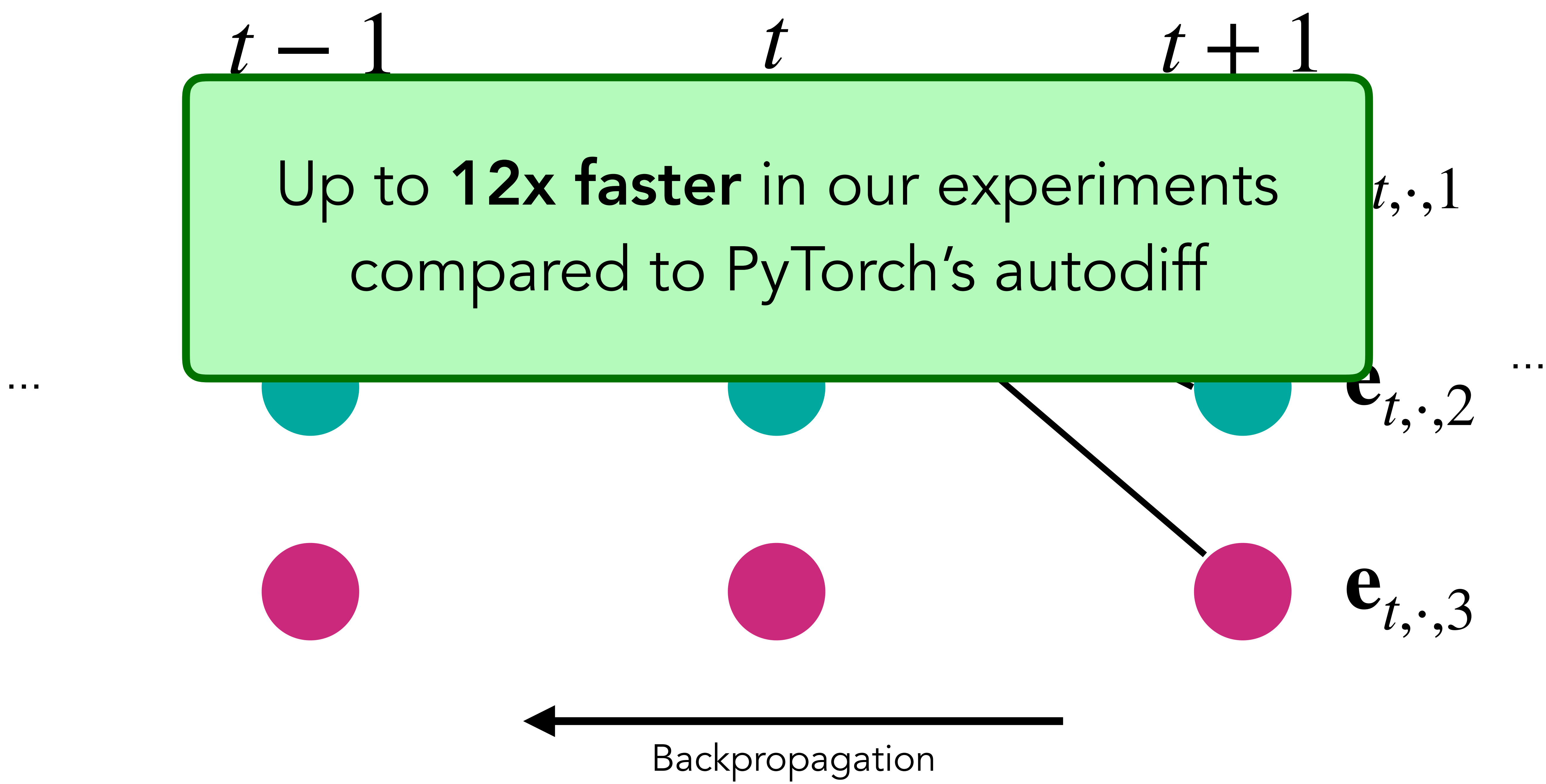
Backpropagation

$$E \triangleq \mathbb{E}_p[Y] \qquad \mathbf{e}_{t,\cdot,j} = \mathbf{q}_{t+1,\cdot,j} \circ (\mathbf{e}_{t+1,\cdot,j}^{\top}\mathbf{1})$$



Backpropagation

$E \triangleq \mathbb{E}_p[Y] \qquad \mathbf{e}_{t,\cdot,j} = \mathbf{q}_{t+1,\cdot,j} \circ (\mathbf{e}_{t+1,\cdot,j}^{\top}\mathbf{1})$



Theoretical results

Theoretical results

1. $DP_{\Omega}(\theta)$ is convex

Proof uses that $x \leq y \Rightarrow max_{\Omega}(x) \leq max_{\Omega}(y)$

Theoretical results

1. $DP_{\Omega}(\theta)$ is convex

   Proof uses that $x \leq y \Rightarrow \max_{\Omega}(x) \leq \max_{\Omega}(y)$

2. Approximation error

N: #nodes in DAG
L, U: constants that depend on Ω

$$(N-1)\, L \leq DP_{\Omega}(\theta) - DP(\theta) \leq (N-1)\, U$$

Theoretical results

1. $DP_\Omega(\theta)$ is convex

Proof uses that $x \leq y \Rightarrow \max_\Omega(x) \leq \max_\Omega(y)$

2. Approximation error

N: #nodes in DAG
L, U: constants that depend on Ω

$$(N-1)\,L \leq DP_\Omega(\theta) - DP(\theta) \leq (N-1)\,U$$

3. $\mathrm{DP}_\Omega(\theta) = \max_\Omega((\langle y, \theta \rangle)_{y \in \mathcal{Y}}) \Leftrightarrow \Omega = -H$ (Shannon's negentropy)

Proof reduces to showing that $\max_{-H}$ is the only $\max_\Omega$ supporting **associativity**, i.e., $\max_{-H}(x, \max_{-H}(y, z)) = \max_{-H}(\max_{-H}(x, y), z)$

Structured prediction losses

Structured prediction losses

Training time

Structured perceptron loss (Collins, 2002)

$$\max_{y \in \mathcal{Y}} \langle \theta, y \rangle - \langle \theta, y_{\text{true}} \rangle$$

Structured prediction losses

Training time

Structured perceptron loss (Collins, 2002)

$$\max_{y \in \mathcal{Y}} \langle \theta, y \rangle - \langle \theta, y_{\text{true}} \rangle = \text{DP}(\theta) - \langle \theta, y_{\text{true}} \rangle$$

Structured prediction losses

Training time

Structured perceptron loss (Collins, 2002)

$$\max_{y \in \mathcal{Y}} \langle \theta, y \rangle - \langle \theta, y_{\text{true}} \rangle = \text{DP}(\theta) - \langle \theta, y_{\text{true}} \rangle$$

Smoothed loss (proposed)

$$\text{DP}_{\Omega}(\theta) - \langle \theta, y_{\text{true}} \rangle$$

Structured prediction losses

Training time

Structured perceptron loss (Collins, 2002)

$$\max_{y \in \mathcal{Y}} \langle \theta, y \rangle - \langle \theta, y_{\text{true}} \rangle = \text{DP}(\theta) - \langle \theta, y_{\text{true}} \rangle$$

Smoothed loss (proposed)

$$\text{DP}_{\Omega}(\theta) - \langle \theta, y_{\text{true}} \rangle$$

Entropic regularization → CRF loss

Quadratic regularization → new loss

Structured prediction losses

Training time

Structured perceptron loss (Collins, 2002)

$$\max_{y \in \mathcal{Y}} \langle \theta, y \rangle - \langle \theta, y_{\text{true}} \rangle = \text{DP}(\theta) - \langle \theta, y_{\text{true}} \rangle$$

Smoothed loss (proposed)

$$\text{DP}_{\Omega}(\theta) - \langle \theta, y_{\text{true}} \rangle$$

Entropic regularization → CRF loss
Quadratic regularization → new loss

Test time

MAP solution

$$\arg\max_{y \in \mathcal{Y} \subseteq \mathbb{R}^m} \langle y, \theta \rangle$$

Structured prediction losses

Training time

Structured perceptron loss (Collins, 2002)

$$\max_{y \in \mathcal{Y}} \langle \theta, y \rangle - \langle \theta, y_{\text{true}} \rangle = \text{DP}(\theta) - \langle \theta, y_{\text{true}} \rangle$$

Smoothed loss (proposed)

$$\text{DP}_{\Omega}(\theta) - \langle \theta, y_{\text{true}} \rangle$$

Entropic regularization → CRF loss
Quadratic regularization → new loss

Test time

MAP solution

$$\underset{y \in \mathcal{Y} \subseteq \mathbb{R}^m}{\arg\max} \ \langle y, \theta \rangle$$

Expected solution

$$\nabla \text{DP}_{\Omega}(\theta) = \mathbb{E}_p[Y]$$

Structured prediction losses

Training time

Structured perceptron loss (Collins, 2002)

$$\max_{y \in \mathcal{Y}} \langle \theta, y \rangle - \langle \theta, y_{\text{true}} \rangle = \text{DP}(\theta) - \langle \theta, y_{\text{true}} \rangle$$

Smoothed loss (proposed)

$$\text{DP}_{\Omega}(\theta) - \langle \theta, y_{\text{true}} \rangle$$

Entropic regularization → CRF loss
Quadratic regularization → new loss

Test time

MAP solution

$$\arg\max_{y \in \mathcal{Y} \subseteq \mathbb{R}^m} \langle y, \theta \rangle$$

Expected solution

$$\nabla \text{DP}_{\Omega}(\theta) = \mathbb{E}_p[Y]$$

Ranking

Sort by probability (sparse case)

NER experiments

S-ORG	O	B-PER	E-PER	O	O	O	O	S-LOC
Apple	CEO	**Tim**	**Cook**	introduces	new	iphone	in	**Cupertino**.

Tags: {Location, Organization, Person, Misc} x {Singleton, Begin, Inside, End}

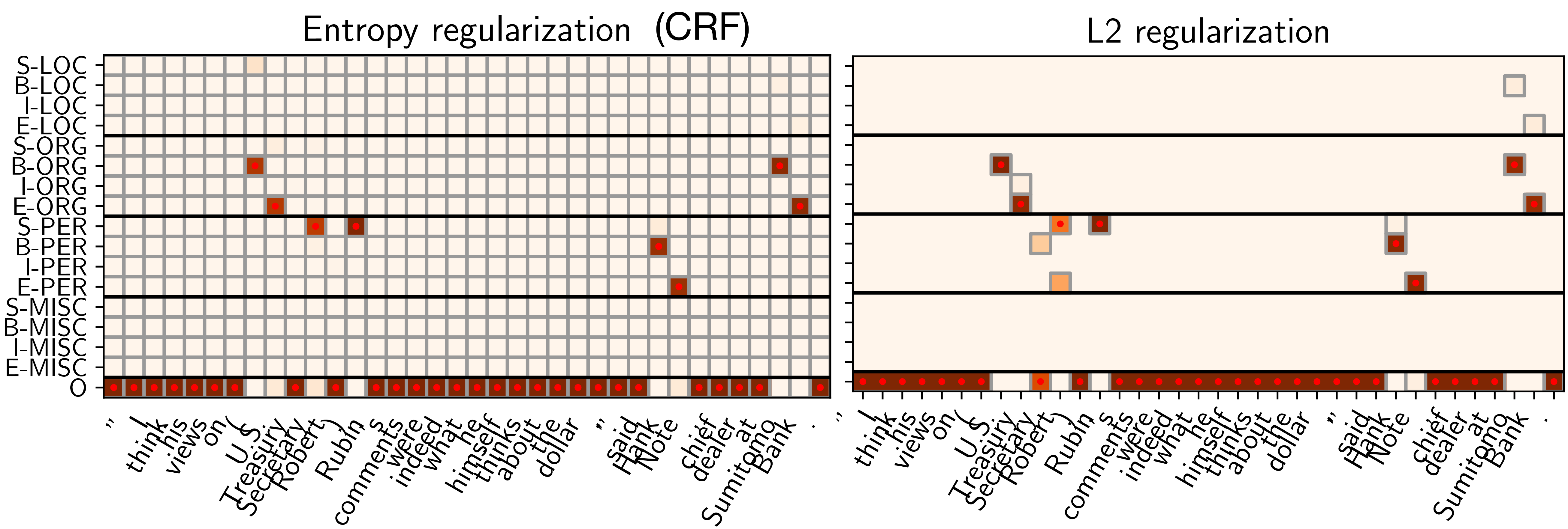
NER experiments

S-ORG O B-PER E-PER O O O O S-LOC

Apple CEO **Tim Cook** introduces new iphone in **Cupertino**.

Tags: {Location, Organization, Person, Misc} x {Singleton, Begin, Inside, End}

Examples of predicted soft assignments at test time



NER experiments

F_1 score comparison on CoNLL03 NER datasets

	English	Spanish	German	Dutch
CRF loss (Entropy)	90.80	86.68	77.35	87.56
Squared norm	90.86	85.51	76.01	86.58
Lample et al 2016 (CRF loss)	90.96	85.75	78.76	81.74

NER experiments

F_1 score comparison on CoNLL03 NER datasets

. Competitive results with other losses

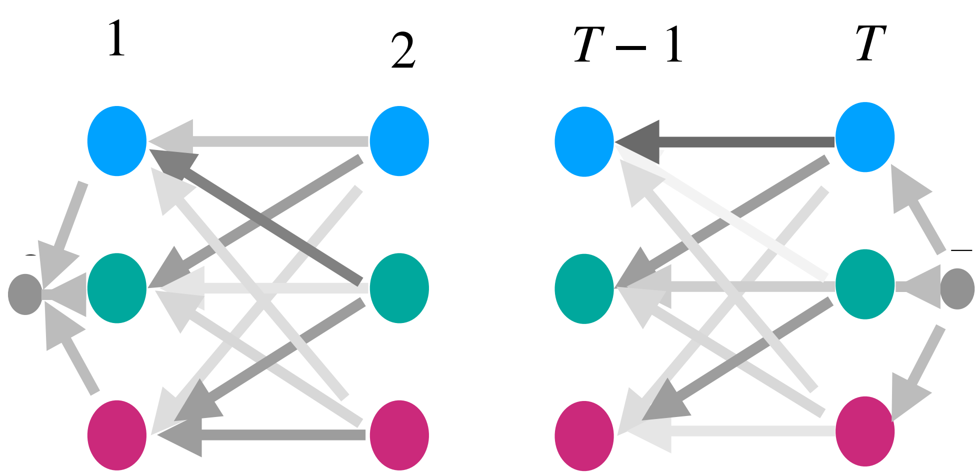
. **Fast convergence** at train time thanks to **smoothness**

. **Sparse probabilistic model** available at test time!

Squared norm	90.86	85.51	76.01	86.58
Lample et al 2016 (CRF loss)	90.96	85.75	78.76	81.74

Summary of second part

Smoothing induces a random walk



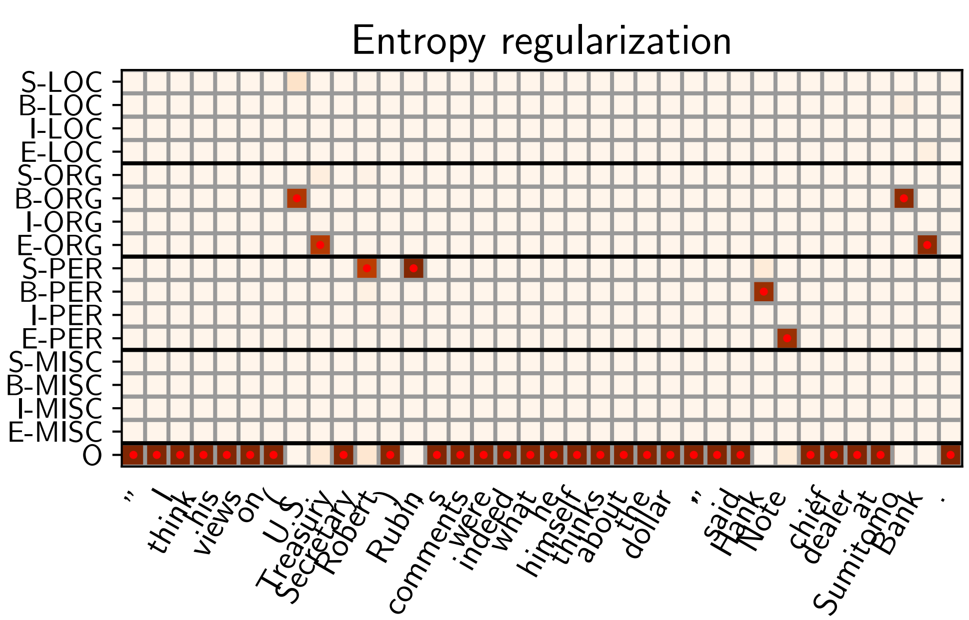
a distribution over paths in the DAG

Gradient = Expected path

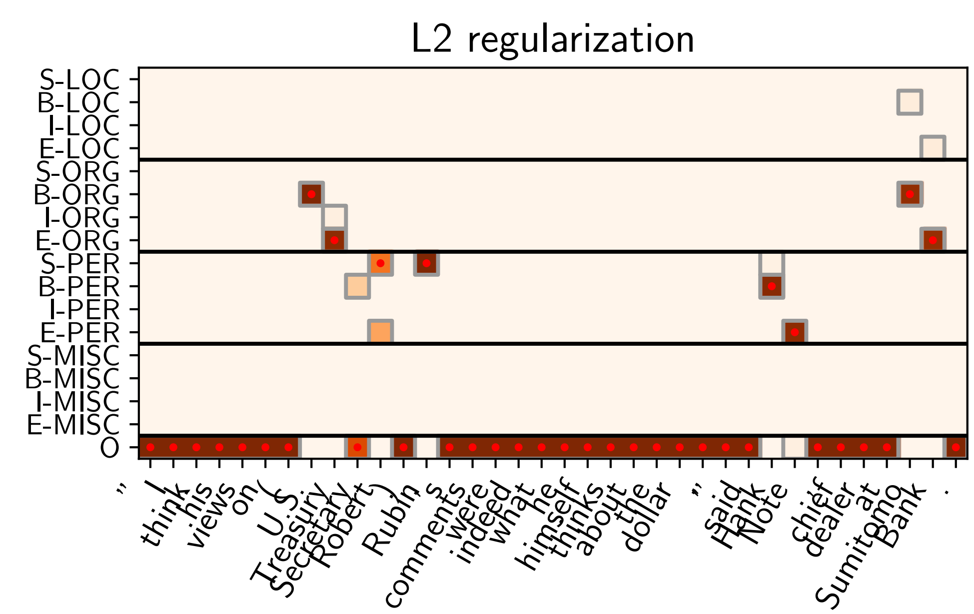
$$\nabla \mathrm{DP}_{\Omega}(\theta) = \mathbb{E}_p[Y]$$

computed efficiently by backprop

Entropic regularization = CRF



L2 regularization = new sparse model



Conclusion

Conclusion

- The log-sum-exp and softmax are ubiquitous in deep learning

Conclusion

- The log-sum-exp and softmax are ubiquitous in deep learning
- $\max_\Omega$ and argmax_Ω operators provide **drop-in replacement** for them with **sparse** and/or **structured** outputs

Conclusion

- The log-sum-exp and softmax are ubiquitous in deep learning
- $\max_\Omega$ and argmax_Ω operators provide **drop-in replacement** for them with **sparse** and/or **structured** outputs
- Induce a **probabilistic perspective**

Conclusion

- The log-sum-exp and softmax are ubiquitous in deep learning
- $\max_\Omega$ and argmax_Ω operators provide **drop-in replacement** for them with **sparse** and/or **structured** outputs
- Induce a **probabilistic perspective**
- Many more potential applications to explore