Neural Representation Learning in Linguistic Structured Prediction

Lingpeng Kong
Carnegie Mellon University

Linguistic Structures

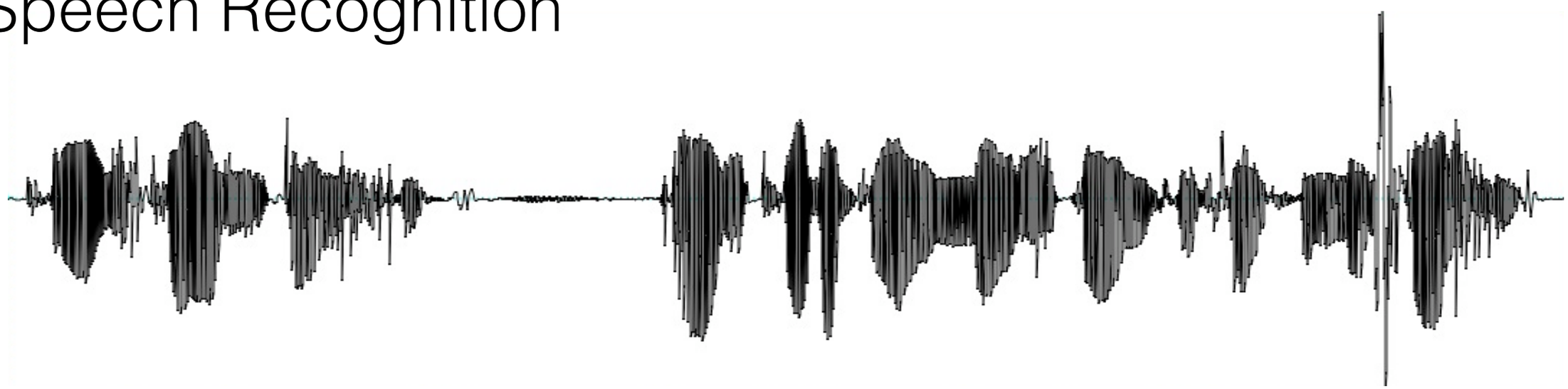
Word Segmentation

1997 年 继 续 主 办 。

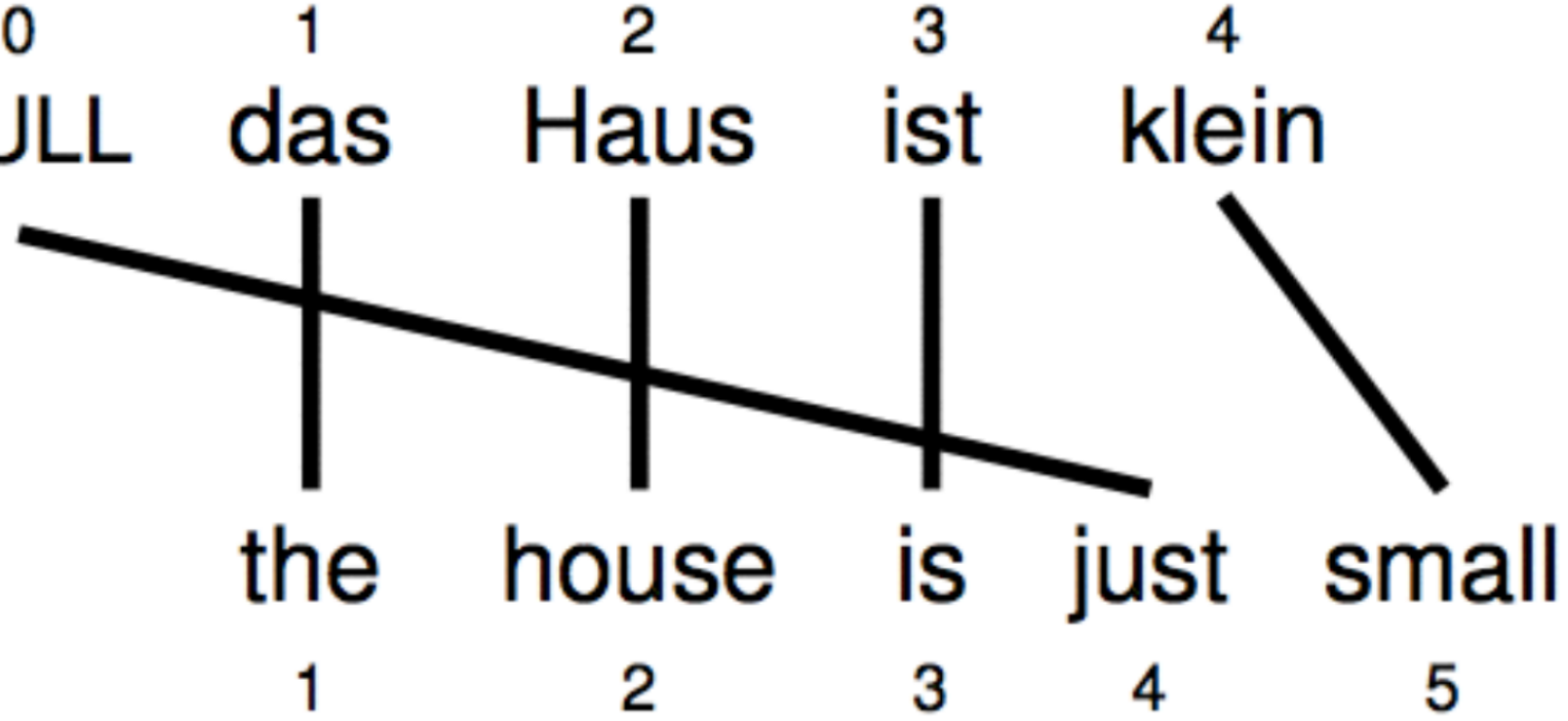
Chunking

NP	PP		NP	PP
The angle	of cats' ears	is	an important clue	to their mood

Speech Recognition



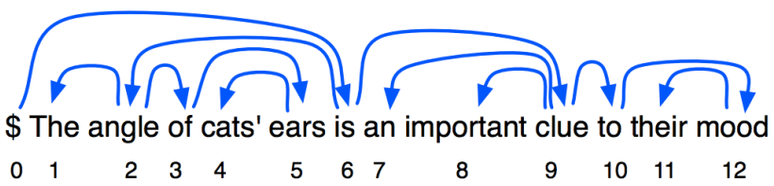
Linguistic Structures



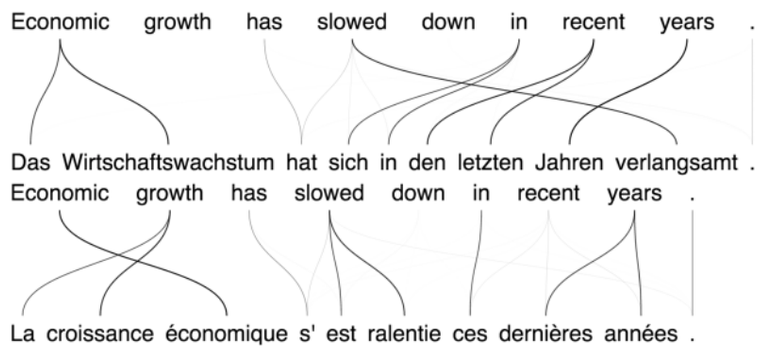
Neural Representation Learning

Speech Recognition (Graves et al, 2013)

Dependency Parsing (Andor et al, 2016)

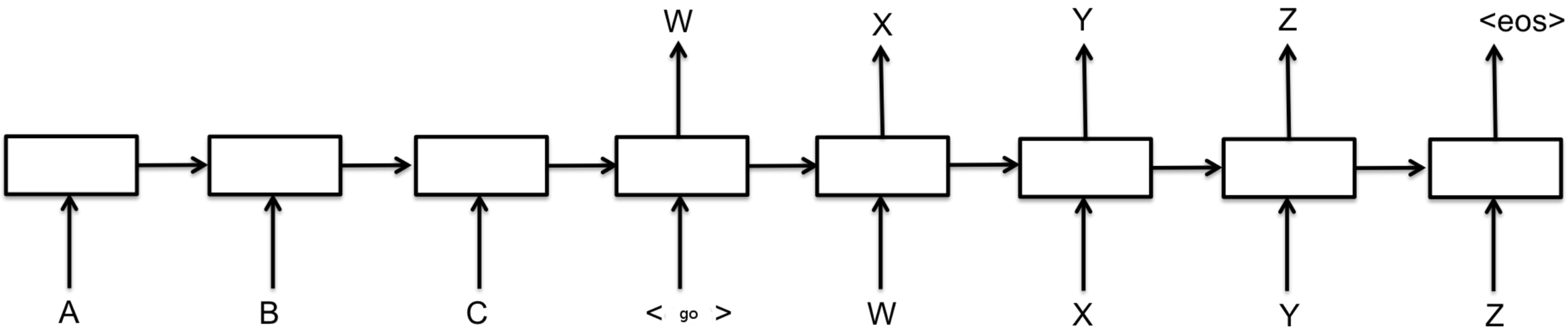


Machine Translation (Bahdanau et al, 2015)



Neural Representation Learning

Sequential Recurrent Neural Networks (RNNs)



Dense Representation

Goal

- The combination of explicit structure representations and learned distributed representations

The importance of linguistic structures:

Automated language understanding (Manning, 2016)

Inductive bias (Mitchell, 1980)

Outline

- Introduction

- Part I — Segmental Recurrent Neural Networks [ICLR 2016, INTERSPEECH 2016, 2017]

- Part II — Inference and Regularization in Sequence to Sequence Models with Stochastic Attention

- Conclusion

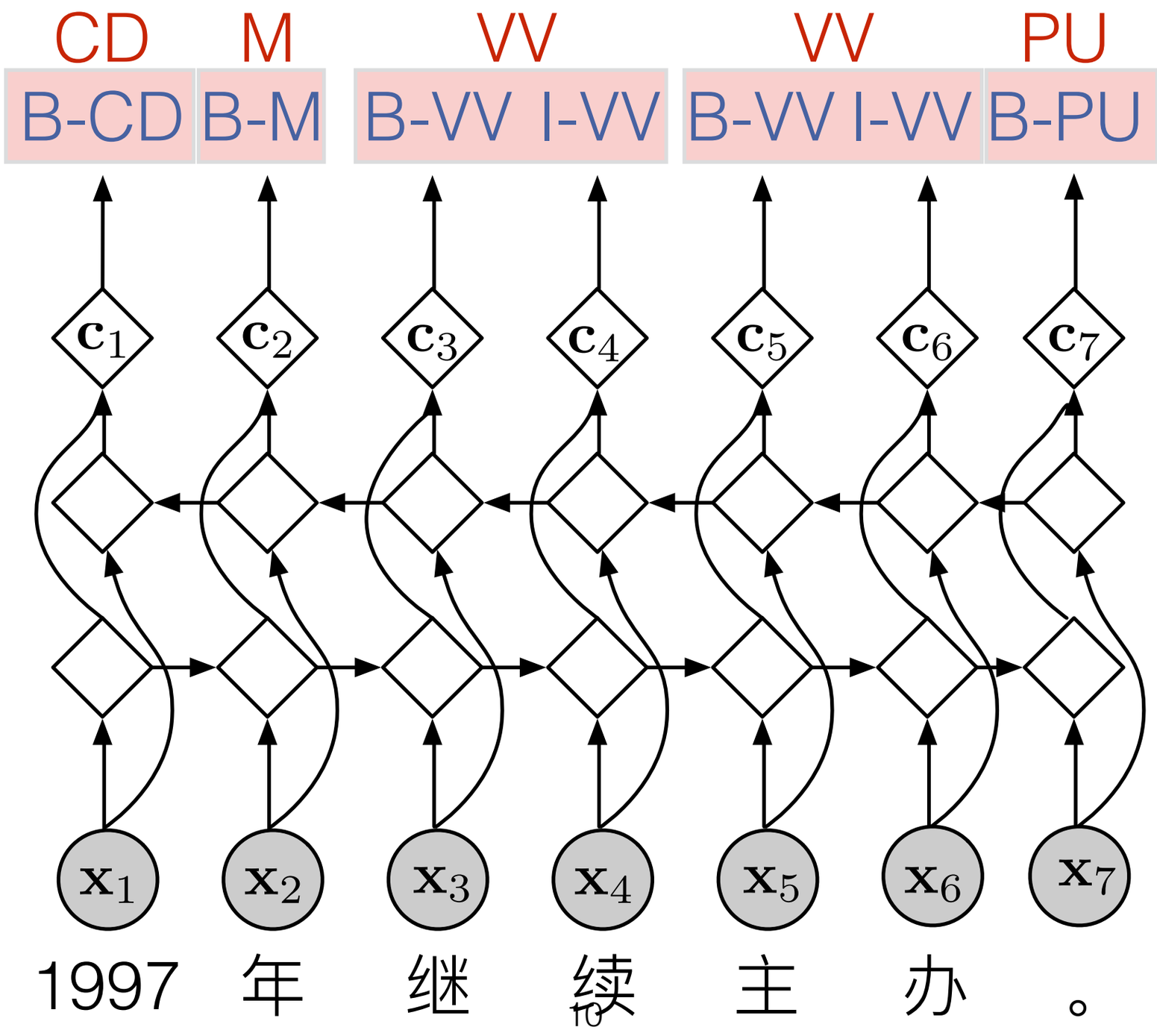
Outline

- Introduction
- Part I — Segmental Recurrent Neural Networks [ICLR 2016, INTERSPEECH 2016, 2017]
- Part II — Inference and Regularization in Sequence to Sequence Models with Stochastic Attention
- Conclusion

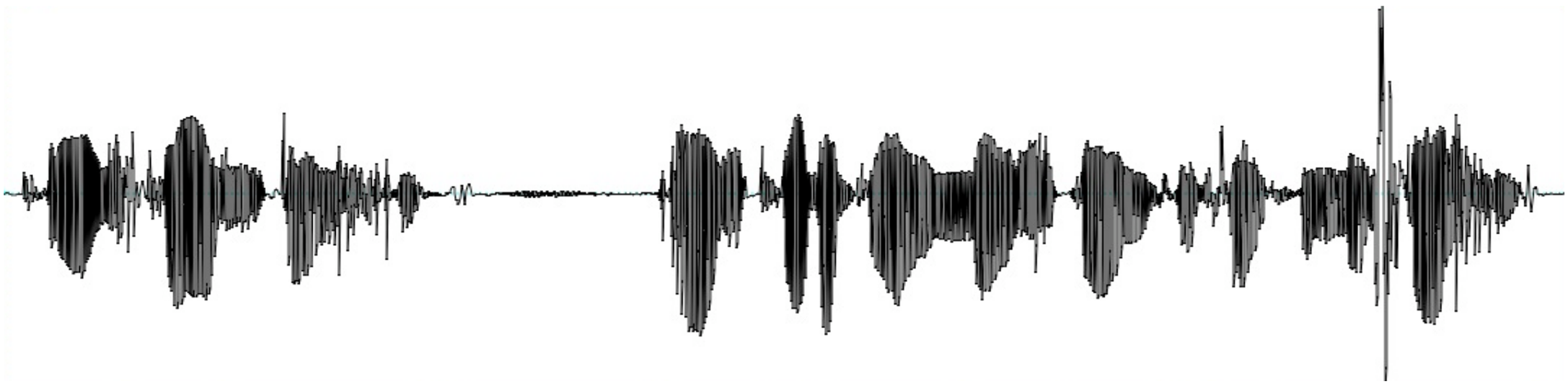
Joint Word Segmentation and POS Tagging

x	1997	年	继　续	主　办	。
y	CD	M	VV	VV	PU

Bi-LSTMs Tagger



Speech Recognition

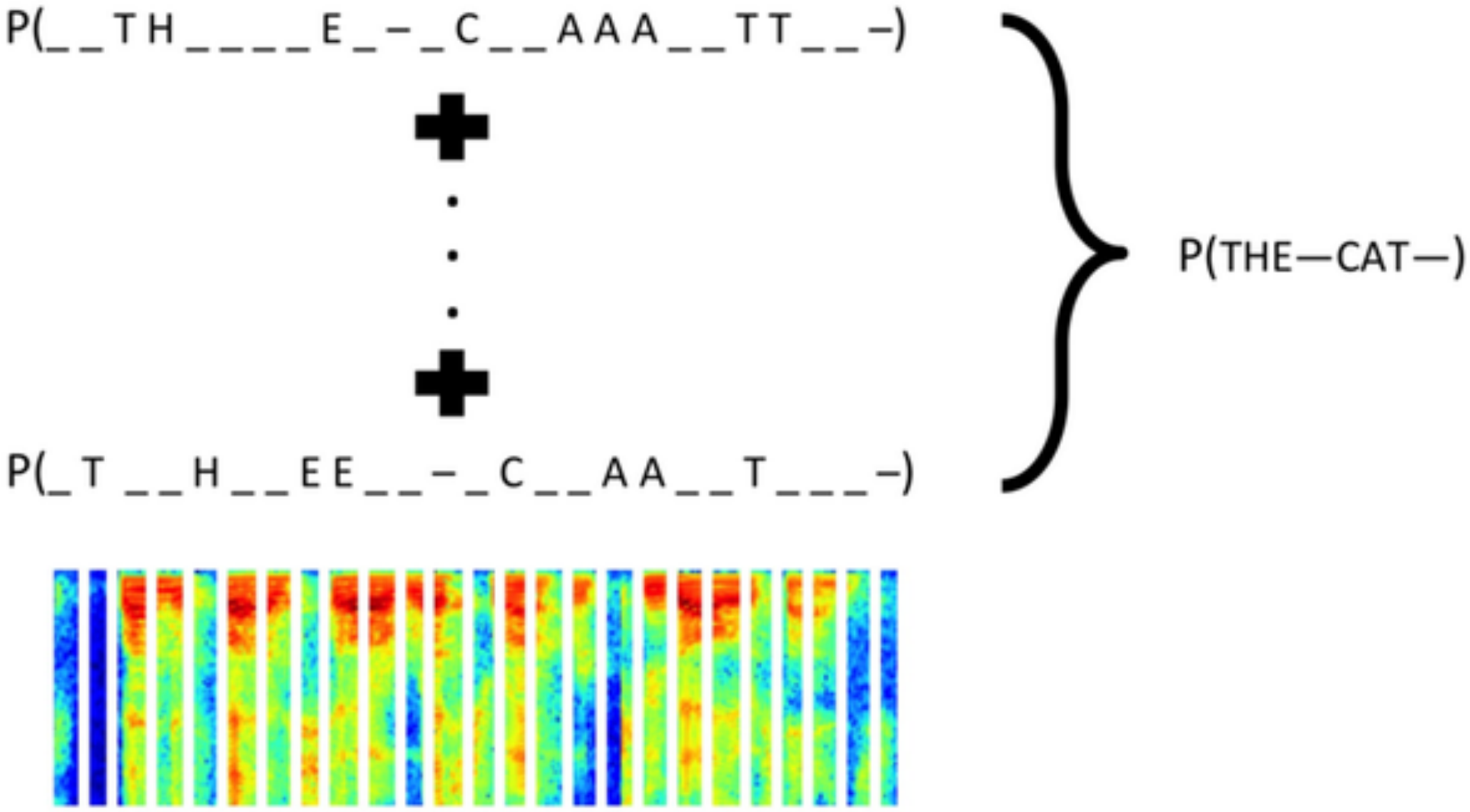
x 

y P IY K L S T L AX AE ER R NG IY Z IH AX F T AO R

Text: *Please call Stella. Ask her to bring these things with her from the store.*

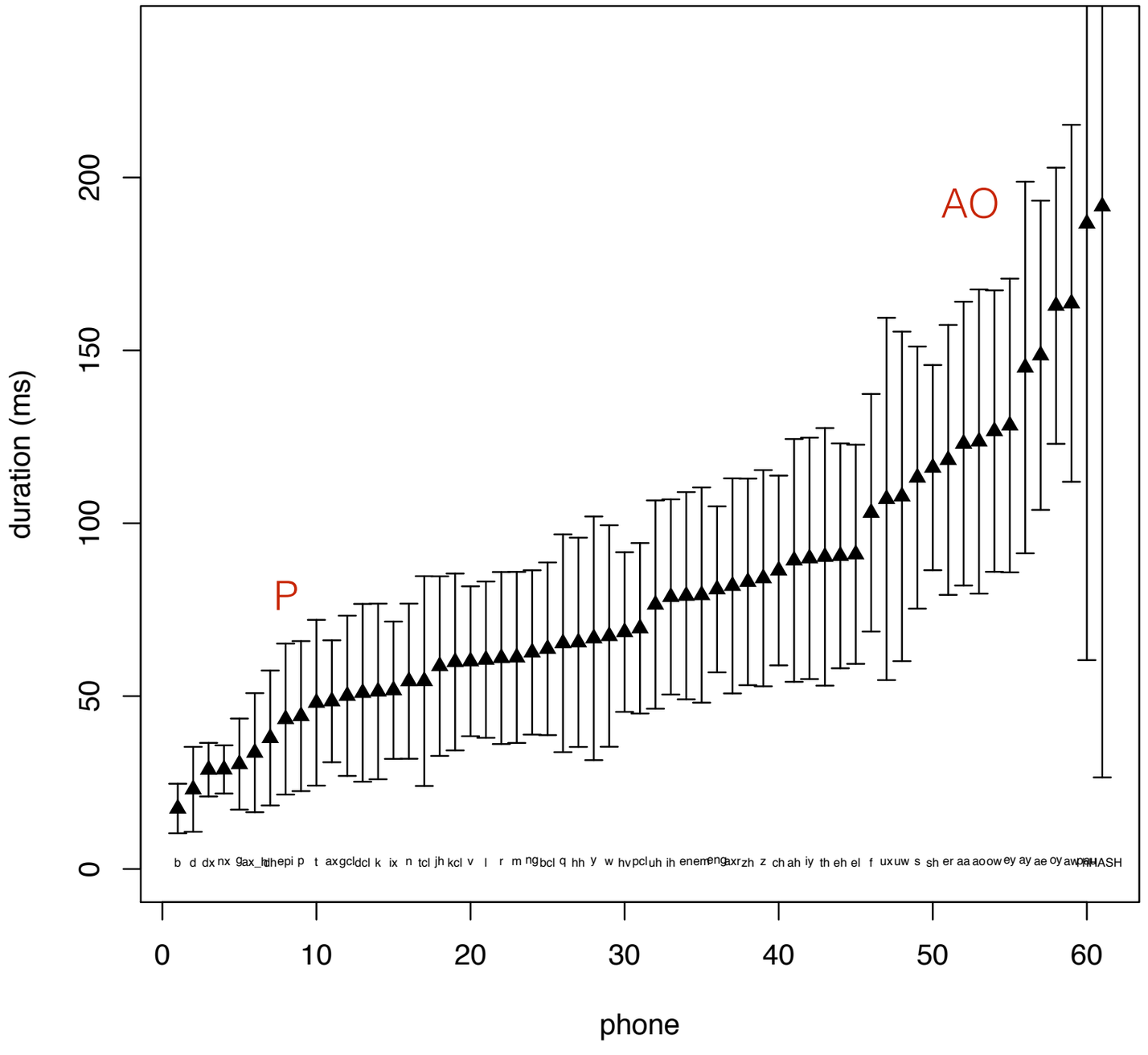
http://groups.linguistics.northwestern.edu/documentation/images/praat_aligned.jpg

Connectionist Temporal Classification (CTC)



(Graves et al, 2013)

Duration Features



Segmental Recurrent Neural Networks (SRNNs)

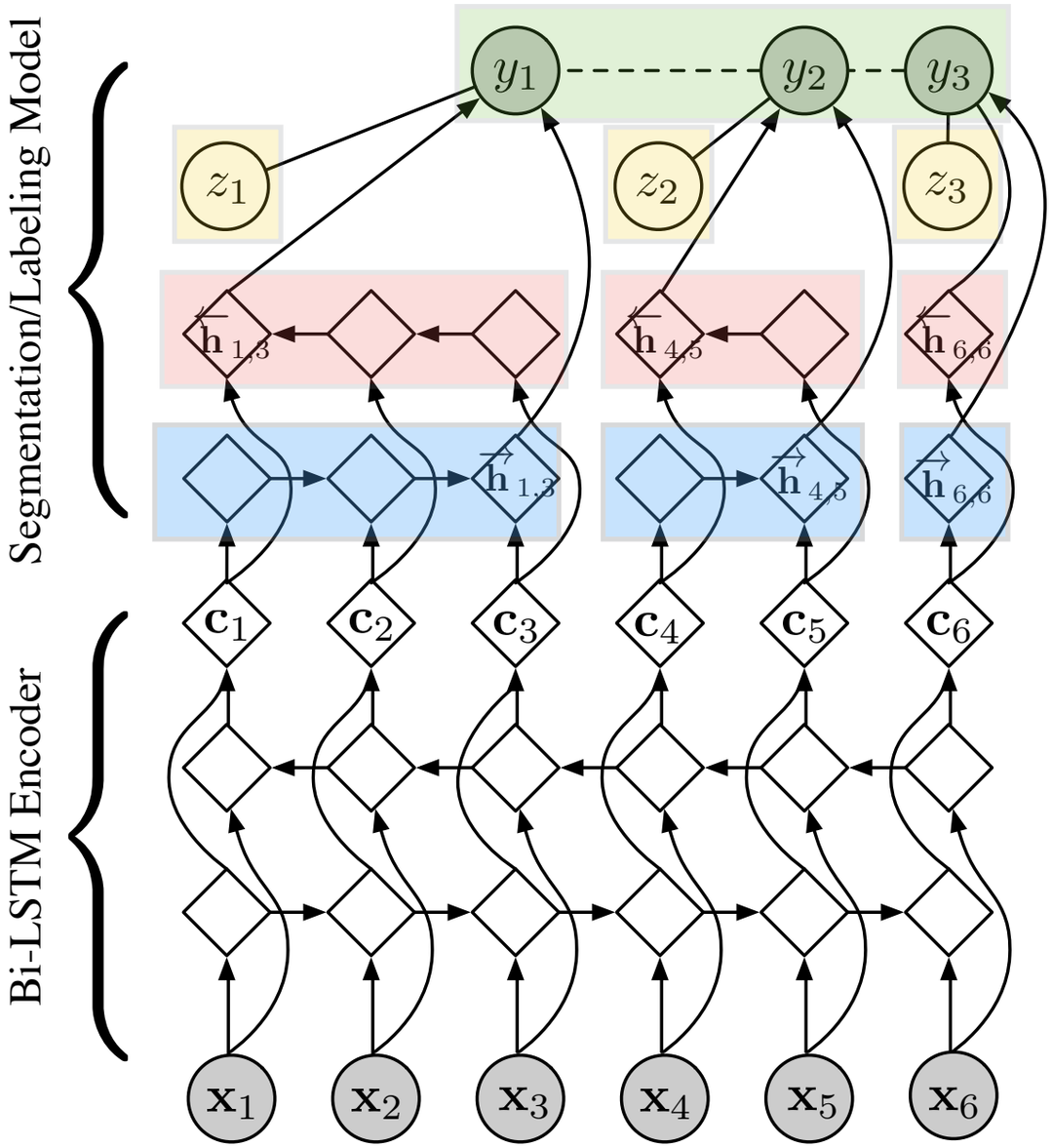
x	1997	年	继	续	主	办	。
z	1	1	2	2	1		
y	CD	M	VV	VV	PU		

SRNNs — $p(y, z|x)$

$$y^* = \arg\max_y \sum_z p(y, z \mid x)$$

$$\approx \arg\max_y \max_z p(y, z \mid x)$$

Segmental Recurrent Neural Networks (SRNNs)



Forward Segment Embedding

Backward Segment Embedding

Duration Embedding

Label Embedding

$$p(\boldsymbol{y}, \boldsymbol{z} \mid \boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{i=1}^{|\boldsymbol{y}|} \exp f(y_{i-k:i}, z_i, \mathbf{x})$$

$$f(y_{i-k:i}, z_i, \mathbf{x}_{s_i:s_i+z_i-1}) =$$
$$\mathbf{w}^\top \phi(\mathbf{V}[\mathbf{g}_y(y_{i-k}); \ldots; \mathbf{g}_y(y_i); \mathbf{g}_z(z_i);$$
$$\overrightarrow{\mathrm{RNN}}(\mathbf{c}_{s_i:s_i+z_i-1}); \overleftarrow{\mathrm{RNN}}(\mathbf{c}_{s_i:s_i+z_i-1})] + \mathbf{a}) + b$$

(Sarawagi and Cohen, 2005)

Parameter Learning

Fully Supervised

$$\mathcal{L} = \sum_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})\in\mathcal{D}} -\log p(\boldsymbol{y},\boldsymbol{z} \mid \boldsymbol{x})$$

$$= \sum_{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})\in\mathcal{D}} \log Z(\boldsymbol{x}) - \log Z(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$$

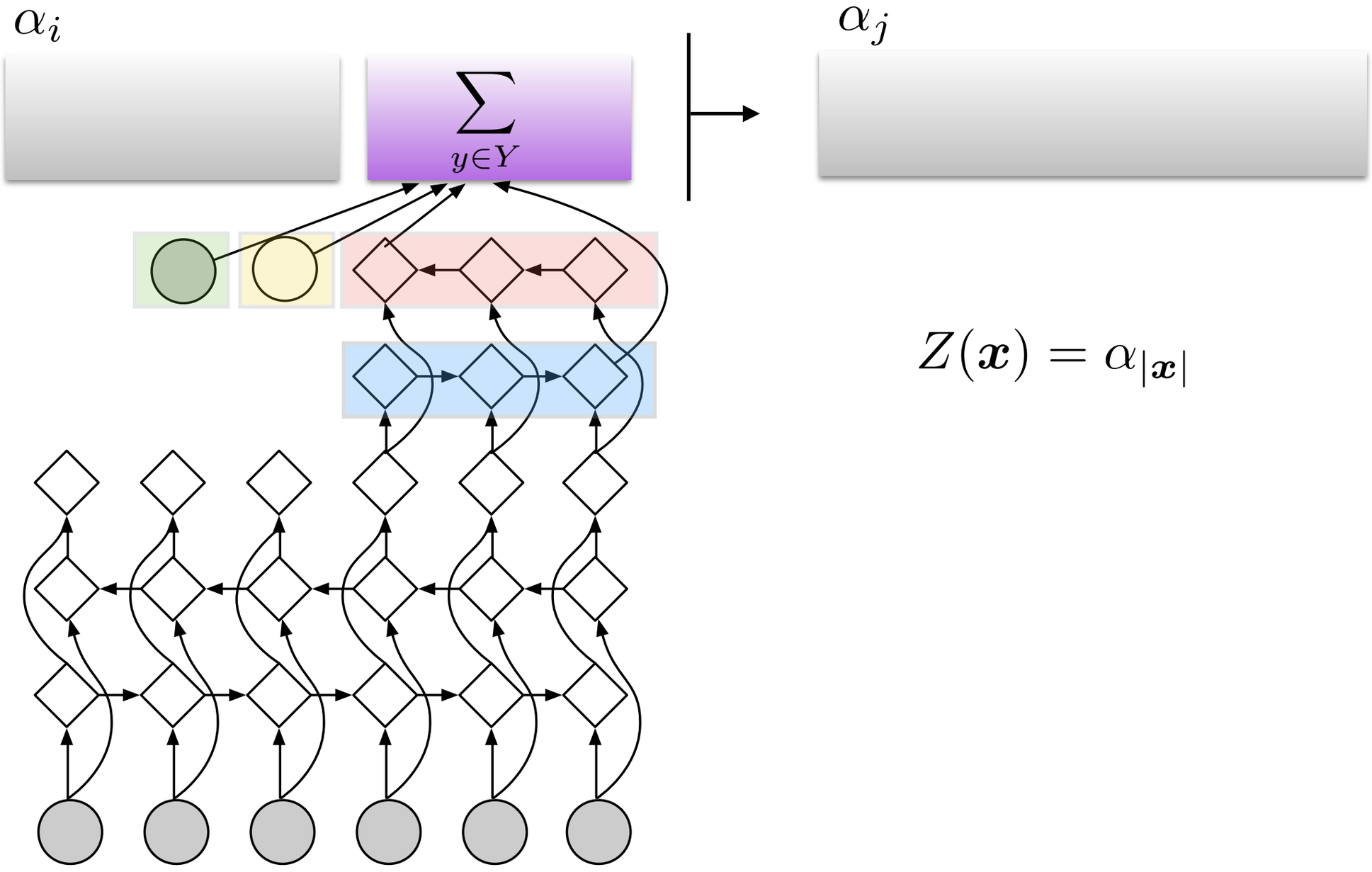
Partially Supervised

$$\mathcal{L} = \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} -\log p(\boldsymbol{y} \mid \boldsymbol{x})$$

$$= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \sum_{\boldsymbol{z}\in\mathcal{Z}(\boldsymbol{x},\boldsymbol{y})} -\log p(\boldsymbol{y},\boldsymbol{z} \mid \boldsymbol{x})$$

$$= \sum_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{D}} \log Z(\boldsymbol{x}) - \log Z(\boldsymbol{x},\boldsymbol{y})$$

Dynamic Programming

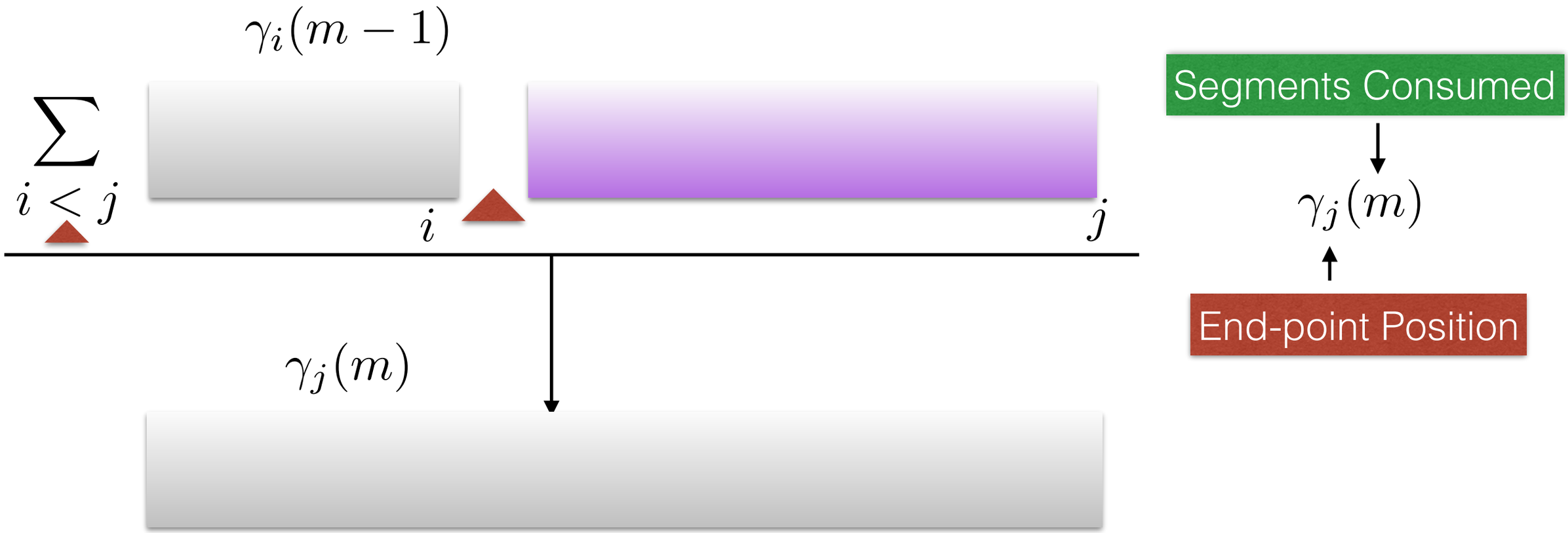


$$Z(\boldsymbol{x}) = \alpha_{|\boldsymbol{x}|}$$

$$\alpha_0 = 1$$

$$\alpha_j = \sum_{i<j} \alpha_i \times \sum_{y \in Y} \exp \mathbf{w}^\top \phi(\mathbf{V}[\mathbf{g}_y(y); \mathbf{g}_z(z_i); \overrightarrow{\mathbf{RNN}}(\mathbf{c}_{s_i:s_i+z_i-1}); \overleftarrow{\mathbf{RNN}}(\mathbf{c}_{s_i:s_i+z_i-1})] + \mathbf{a}) + b$$

Dynamic Programming

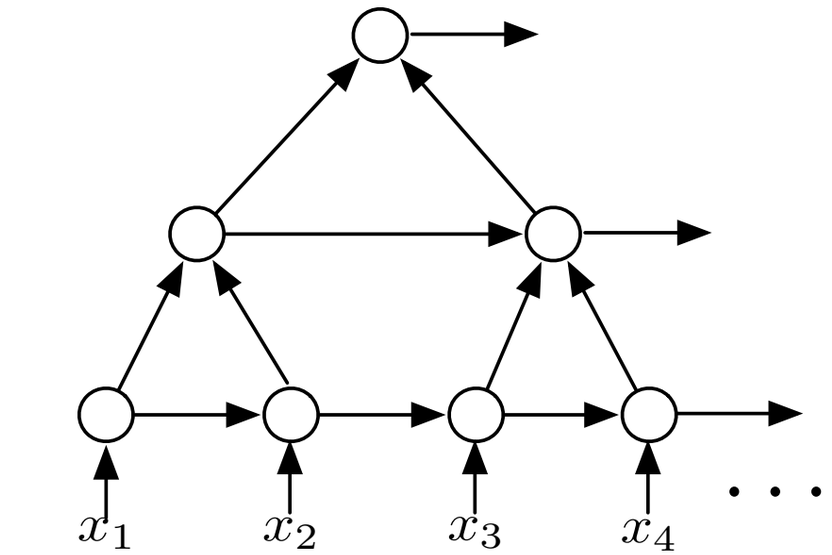


$$Z(\boldsymbol{x}, \boldsymbol{y}) = \gamma_{|\boldsymbol{x}|}(|\boldsymbol{y}|)$$

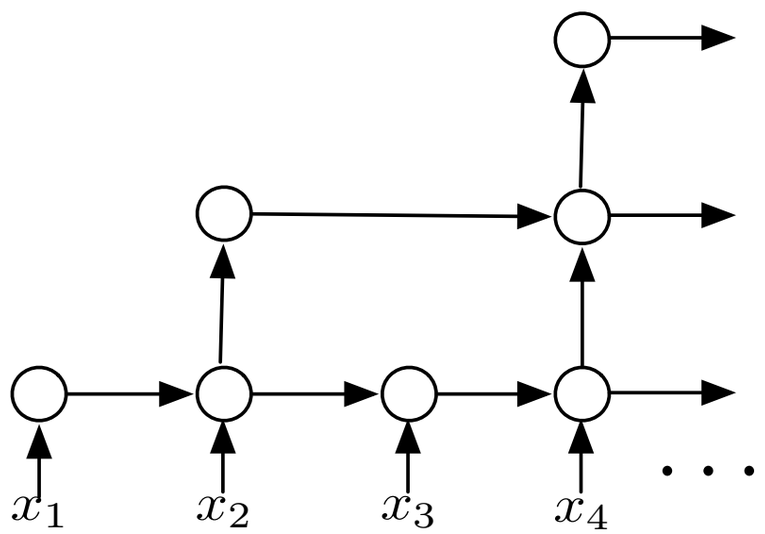
$$\gamma_j(m) = \sum_{i<j} \gamma_i(m-1) \times \left(\exp \mathbf{w}^\top \phi(\mathbf{V}[g_y(y_i); g_z(z_i); \overrightarrow{\text{RNN}}(\mathbf{c}_{s_i:s_i+z_i-1}); \overleftarrow{\text{RNN}}(\mathbf{c}_{s_i:s_i+z_i-1})] + \mathbf{a}) + b \right)$$

Further Speedup

Subsampling



a) concatenate / add



b) skip

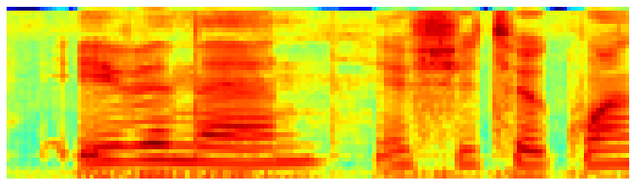
Experiments

Online Hand Writing Recognition

	P (seg)	R (seg)	F (seg)	Error
CTC	-	-	-	13.8%
SRNNs(Full)	98.9%	98.6%	98.6%	5.4%
SRNNs (Partial)	99.2%	99.1%	99.2%	2.7%

(Kassel, 1995)
(Taskar et al. 2004)

Experiments

Speech Recognition 

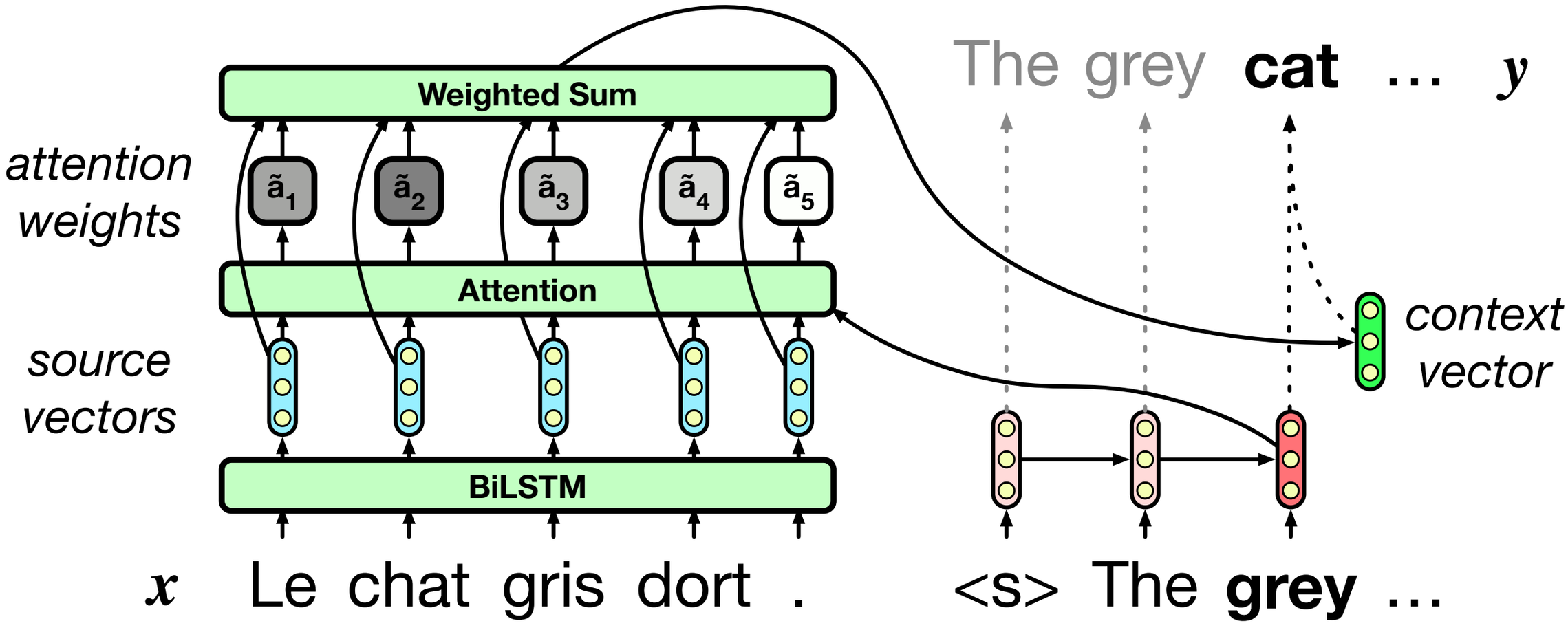
Multi-task Learning with CTC and Segmental CRF for Speech Recognition [INTERSPEECH 2016]

Segmental Recurrent Neural Networks for End-to-end Speech Recognition [INTERSPEECH 2017]

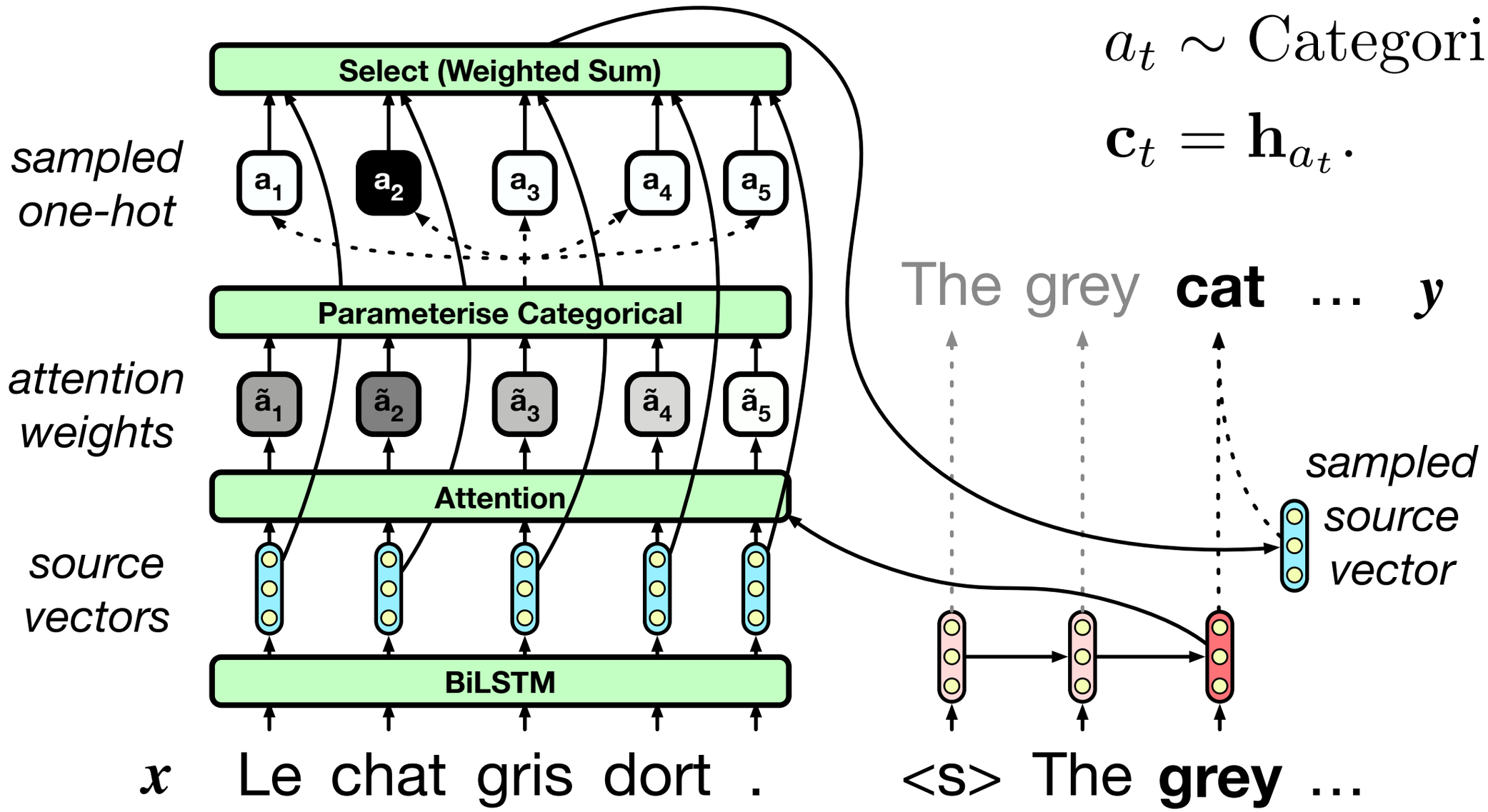
Outline

- Introduction
- Part I — Segmental Recurrent Neural Networks [ICLR 2016, INTERSPEECH 2016, 2017]
- Part II — Inference and Regularization in Sequence to Sequence Models with Stochastic Attention
- Conclusion

Deterministic v.s. Stochastic Attention



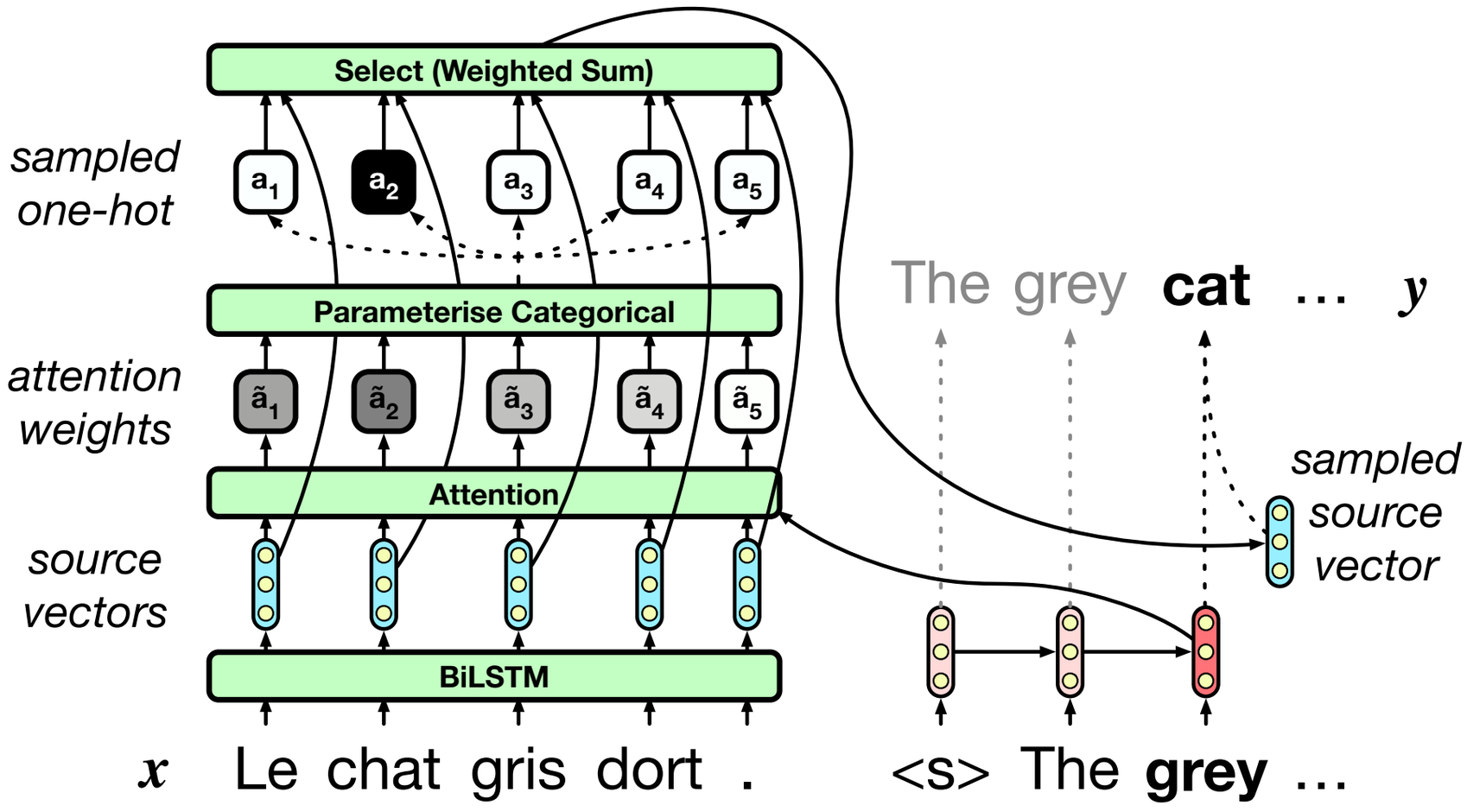
Deterministic v.s. Stochastic Attention



$$a_t \sim \text{Categorical}(a_t; \tilde{\boldsymbol{\alpha}}_t)$$

$$\mathbf{c}_t = \mathbf{h}_{a_t}.$$

Marginal Likelihood and Training Objective



$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{t=1}^{M} \sum_{a_t=1}^{N} p(a_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t}) p(y_t \mid a_t, \boldsymbol{x}, \boldsymbol{y}_{<t})$$

Approximating the Marginal Likelihood

Variational lower bound:

$$\log p(\boldsymbol{y}|\boldsymbol{x}) = \sum_t \log \sum_a p(a|\boldsymbol{x}, \boldsymbol{y}_{<t}) p(y_t|a, \boldsymbol{x}, \boldsymbol{y}_{<t})$$

$$= \sum_t \log \sum_a p(y_t, a|\boldsymbol{x}, \boldsymbol{y}_{<t})$$

$$\leq \sum_t \mathop{\mathbb{E}}_{q(a)} \log \frac{p(y_t, a|\boldsymbol{x}, \boldsymbol{y}_{<t})}{q(a)}$$

$$= \sum_t \mathop{\mathbb{E}}_{q(a)} \log p(y_t, a|\boldsymbol{x}, \boldsymbol{y}_{<t}) - H(q)$$

Approximating the Marginal Likelihood

Variational lower bound:

$$\log p(\boldsymbol{y}|\boldsymbol{x}) = \sum_t \log \sum_a p(a|\boldsymbol{x}, \boldsymbol{y}_{<t}) p(y_t|a, \boldsymbol{x}, \boldsymbol{y}_{<t})$$

$$= \sum_t \log \sum_a p(y_t, a|\boldsymbol{x}, \boldsymbol{y}_{<t})$$

$$\leq \sum_t \mathop{\mathbb{E}}_{q(a)} \log \frac{p(y_t, a|\boldsymbol{x}, \boldsymbol{y}_{<t})}{q(a)}$$

$$= \sum_t \mathop{\mathbb{E}}_{q(a)} \log p(y_t, a|\boldsymbol{x}, \boldsymbol{y}_{<t}) - H(q)$$

REINFORCE:

$$\log p(\boldsymbol{y}|\boldsymbol{x}) = \sum_t \log \sum_a p(a|\boldsymbol{x}, \boldsymbol{y}_{<t}) p(y_t|a, \boldsymbol{x}, \boldsymbol{y}_{<t})$$

One-sample approximation

Experiments: Deterministic vs. Stochastic Attention

Hypothesis: stochasticity in memory access (rather than mean-field-like approximations) results in easier learning problems.

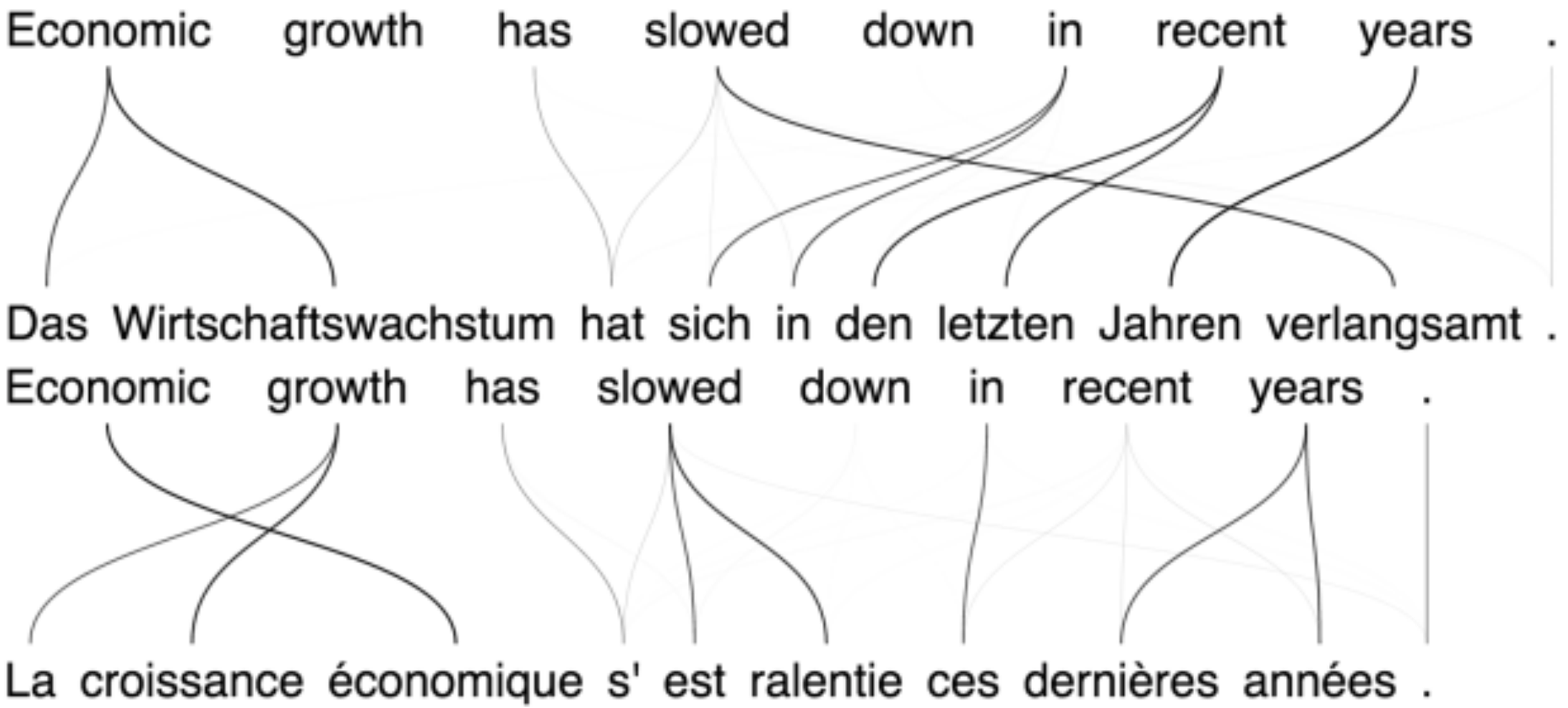
Model	Inference	BLEU	PPL
Deterministic	-	31.87	5.25
Stochastic	exact	**31.91**	**4.65**
Stochastic	variational	30.10	5.40
Stochastic	REINFORCE	29.85	5.31

Punchline: stochasticity doesn't really help us.

Stochastic Attention

- **Let's not give up yet!**
  - Neural nets can fit anything (Zhang, Bengio, Hardt, Recht, Vinyals, ICLR 2017).
  - (Stochastic) attention should be sensible, not just a random fit
    - Let's **regularize** the posterior distributions so they look more like what we expect posteriors to be (Ganchev, Graça, Gillenwater, Taskar, JMLR 2010)
- **Strategy:**
  - Apply KL penalty (true PR penalty)
  - Use variants of IS (biased estimator) using the expected posterior as the instrumental distribution

IBM Models



Great q distribution!

Posterior Regularization

Exact:

$$\sum_t \log \sum_a p(a|\boldsymbol{x}, \boldsymbol{y}_{<t}) p(y_t|a, \boldsymbol{x}, \boldsymbol{y}_{<t})$$

$$+\gamma \times \mathbf{KL}[p(a|y_t, \boldsymbol{x}, \boldsymbol{y}_{<t})|||q(a)]$$

Posterior Regularization

Importance Sampling:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_t \sum_a p(a|\boldsymbol{x}, \boldsymbol{y}_{<t}) p(y_t|a, \boldsymbol{x}, \boldsymbol{y}_{<t})$$

$$= \prod_t \sum_a p(y_t, a|\boldsymbol{x}, \boldsymbol{y}_{<t})$$

$$= \prod_t \sum_a q(a) w(a, \boldsymbol{x}, \boldsymbol{y})$$

$$= \prod_t \mathop{\mathbb{E}}_{q(a)} w(a, \boldsymbol{x}, \boldsymbol{y})$$

Monte Carlo approximation

$$w(a, \boldsymbol{x}, \boldsymbol{y}) = \frac{p(y_t, a|\boldsymbol{x}, \boldsymbol{y}_{<t})}{q(a)}$$

Posterior Regularization

Jensen IS:

$$\log p(\boldsymbol{y}|\boldsymbol{x}) = \sum_t \log \sum_a p(a|\boldsymbol{x}, \boldsymbol{y}_{<t}) p(y_t|a, \boldsymbol{x}, \boldsymbol{y}_{<t})$$

$$= \sum_t \log \sum_a p(y_t, a|\boldsymbol{x}, \boldsymbol{y}_{<t})$$

$$\leq \sum_t \mathop{\mathbb{E}}_{q(a)} \log \frac{p(y_t, a|\boldsymbol{x}, \boldsymbol{y}_{<t})}{q(a)}$$

$$= \sum_t \mathop{\mathbb{E}}_{q(a)} \log p(y_t, a|\boldsymbol{x}, \boldsymbol{y}_{<t}) - H(q)$$

Importance Sampling

$$w(a, \boldsymbol{x}, \boldsymbol{y}) = \frac{p(y_t, a|\boldsymbol{x}, \boldsymbol{y}_{<t})}{q(a)}$$

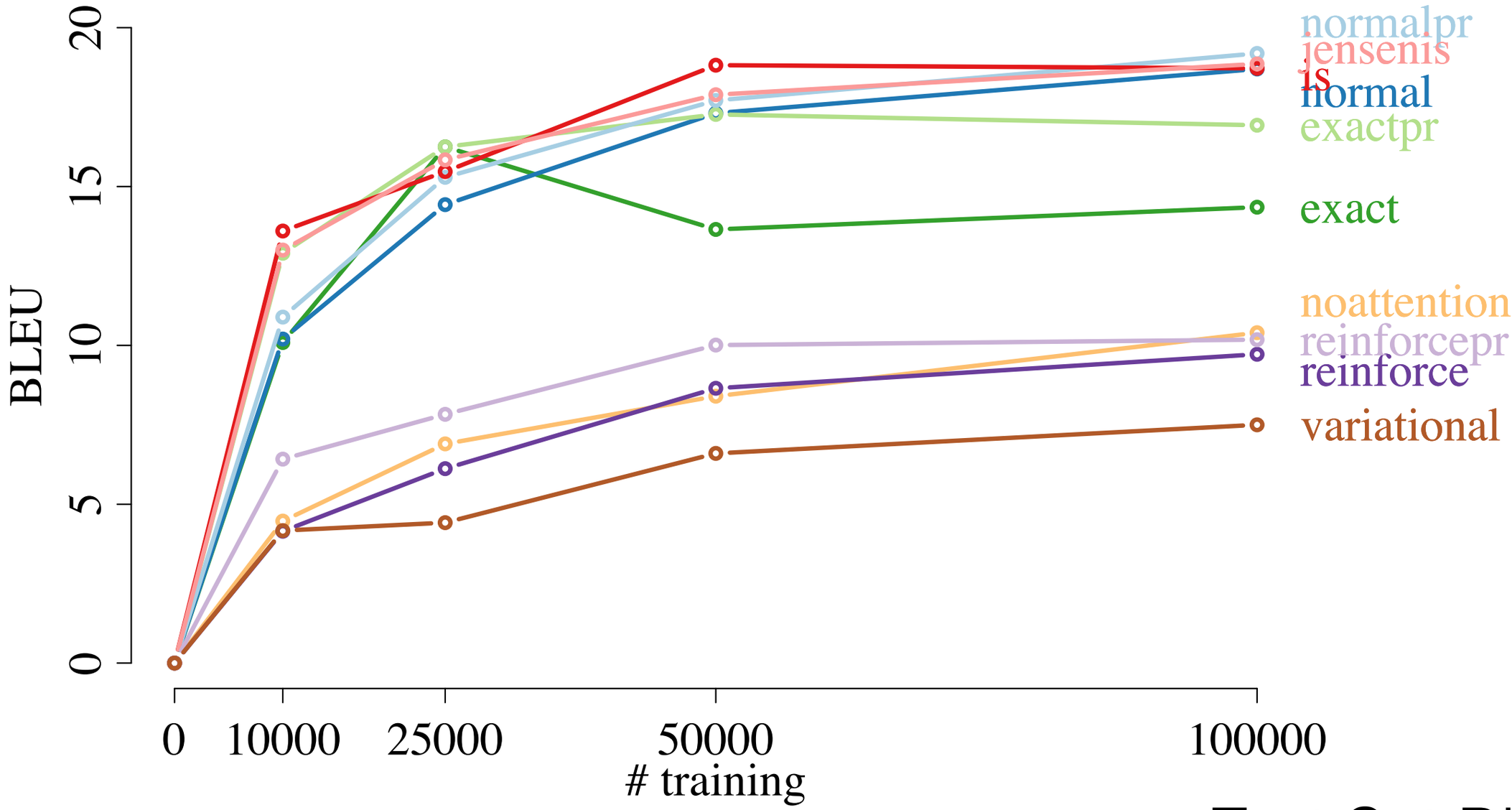
sample from a fixed q(a)

Experiments: Posterior regularization

Hypothesis: neural nets can fit anything (Zhang, Bengio, Hardt, Recht, Vinyals).
Let's regularize posterior distributions (Ganchev, Graca, Gillenwater, Taskar).

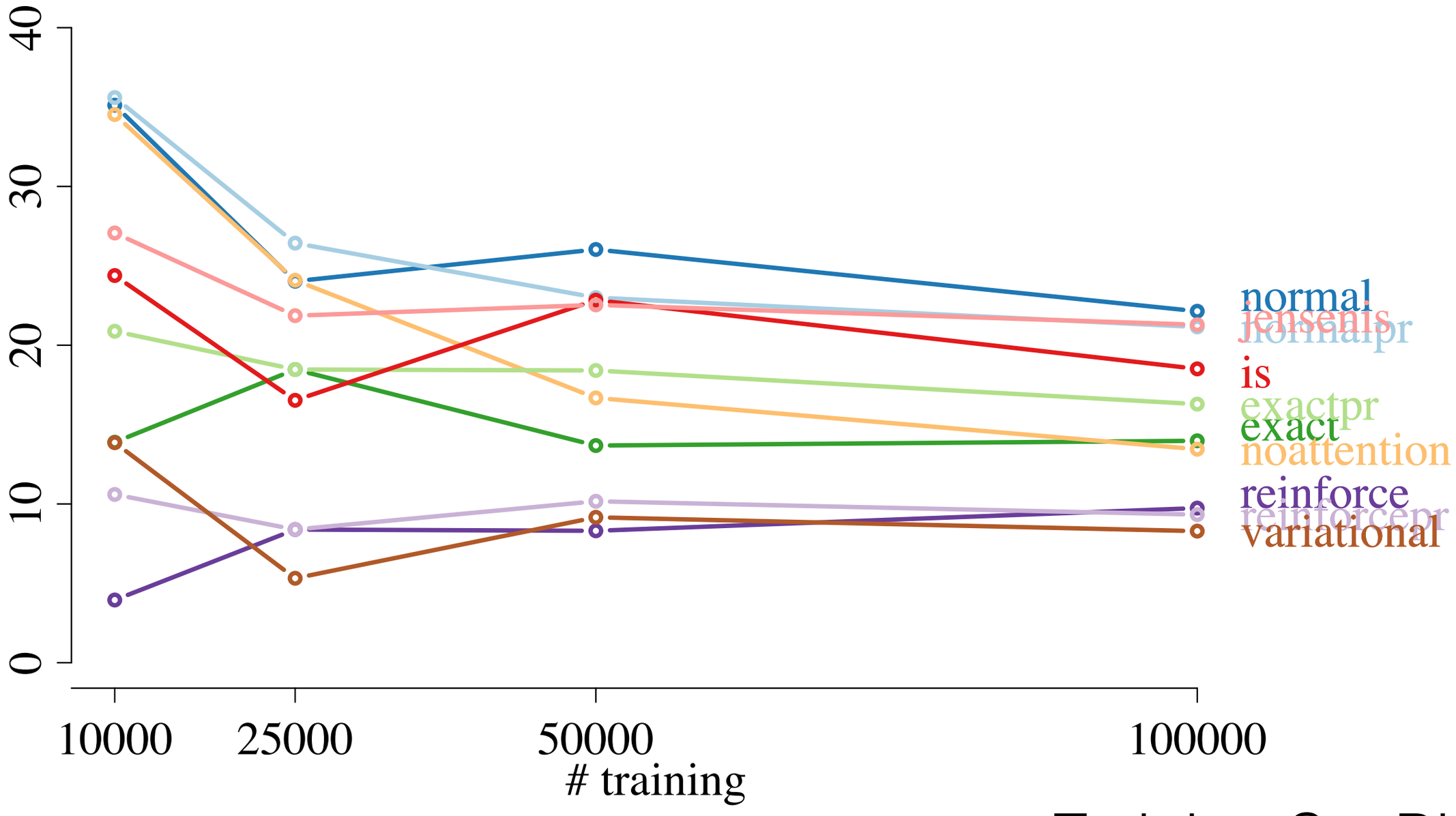
Model	Inference	PR	BLEU	PPL
Deterministic	exact	none	31.87	5.25
Stochastic	exact	none	31.91	4.65
Deterministic	exact	full	32.48	5.20
Stochastic	exact	full	35.17	4.03
Stochastic	IS with q	approximate	34.68	4.04
Stochastic	Jensen bound IS with q	approximate	**35.40**	**3.94**

Experiments: Different training set size



Test Set BLEU

Experiments: Different training set size



Training Set BLEU

Conclusion

- The combination of explicit structure representations and learned distributed representations

- Part I — Segmental Recurrent Neural Networks [ICLR 2016, INTERSPEECH 2016, 2017]

- Part II — Inference and Regularization in Sequence to Sequence Models with Stochastic Attention

Thank you!