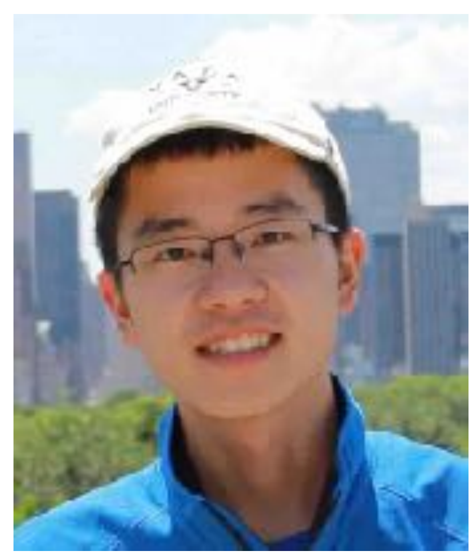
Learning about Language with Normalizing Flows

Graham Neubig

Language Technologies Institute, Carnegie Mellon University

Chunting Zhou

Junxian He



Xuezhe Ma

Di Wang, Daniel Spokoyny, Xian Li, Taylor Berg-Kirkpatrick, Eduard Hovy

Learning about Language?

- **Syntactic structure**

	The	cat	sat	on	a	green	wall
Parts-of-speech:	DT	NN	VBD	IN	DT	JJ	NN

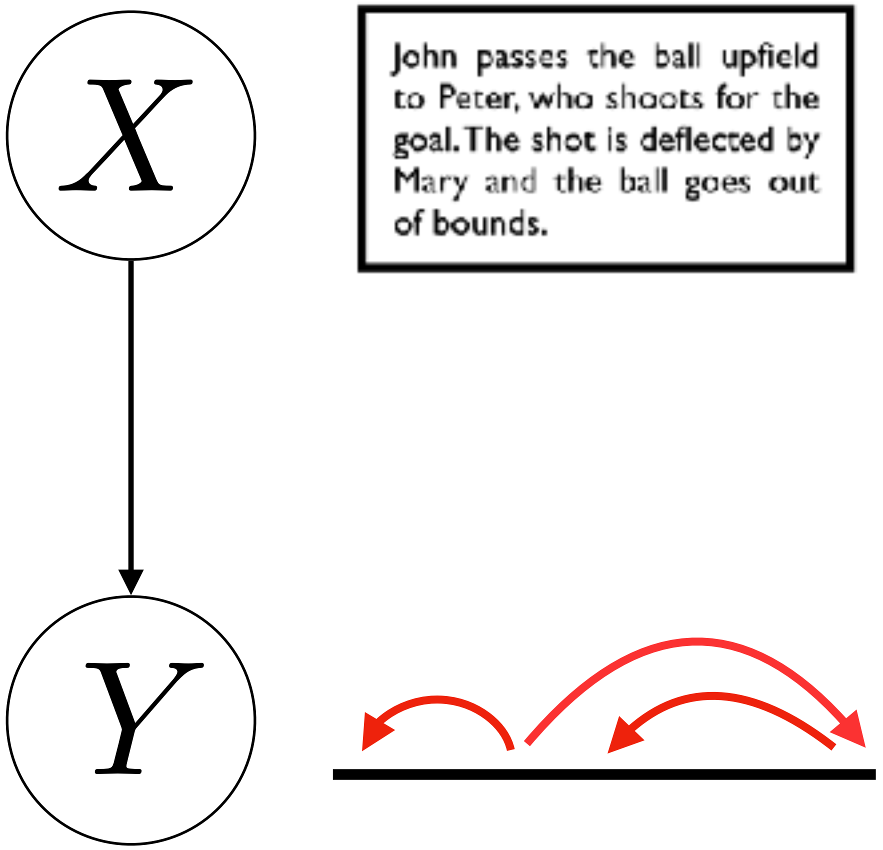
Dependency:

- **Cross-lingual correspondences**

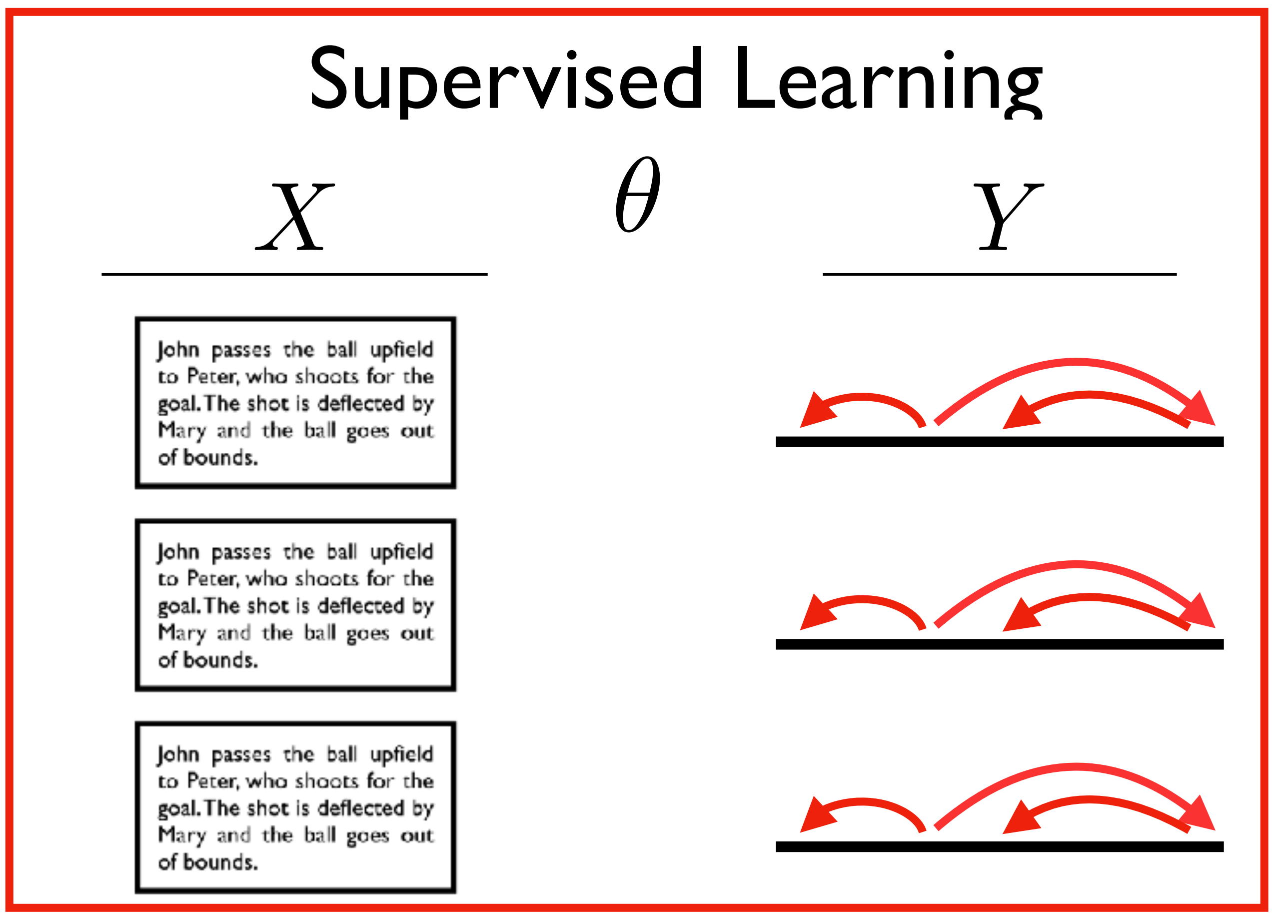
a cat green on sat the wall

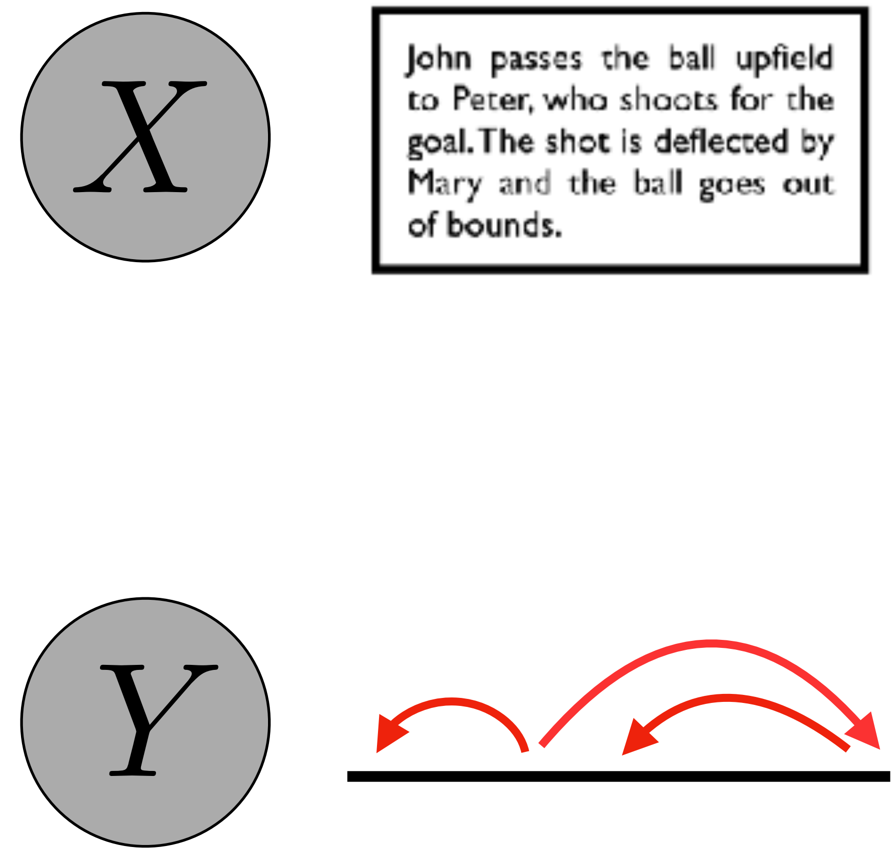
の は 上 壁 猫 緑 座った

Supervised Approaches



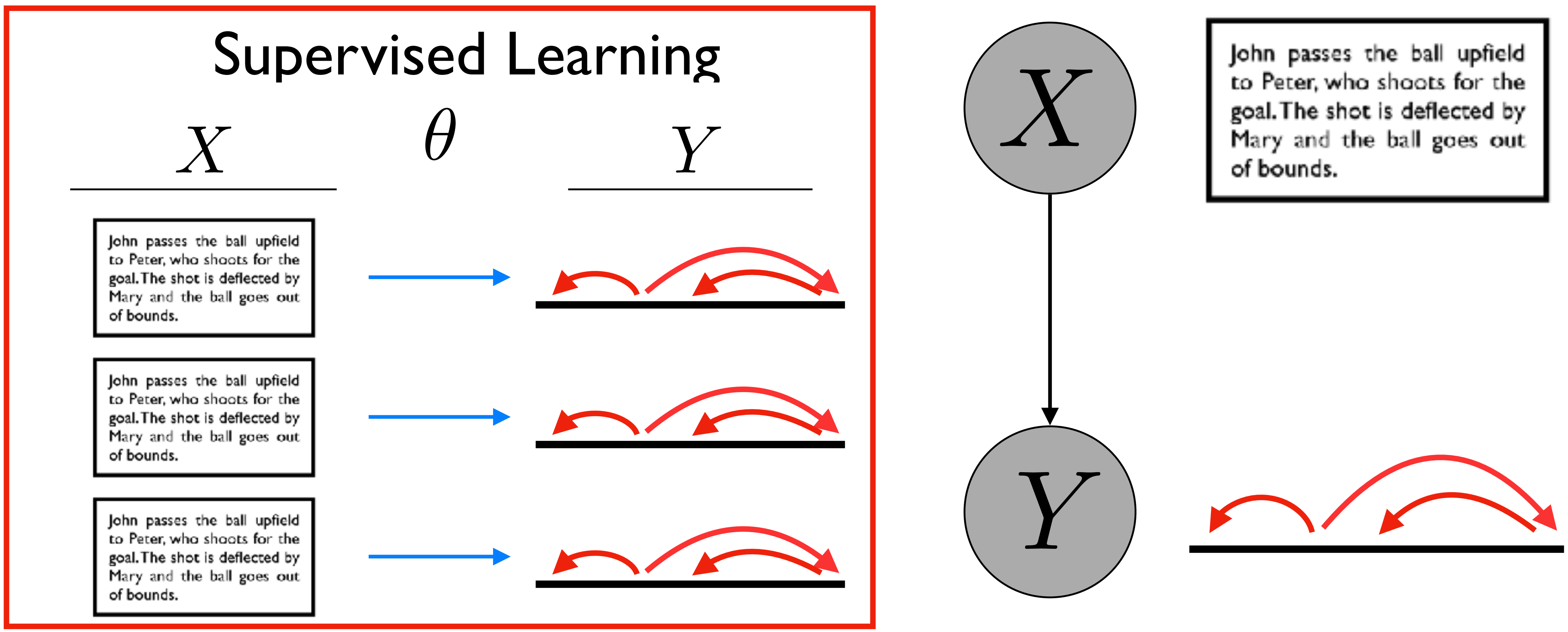
Supervised Approaches



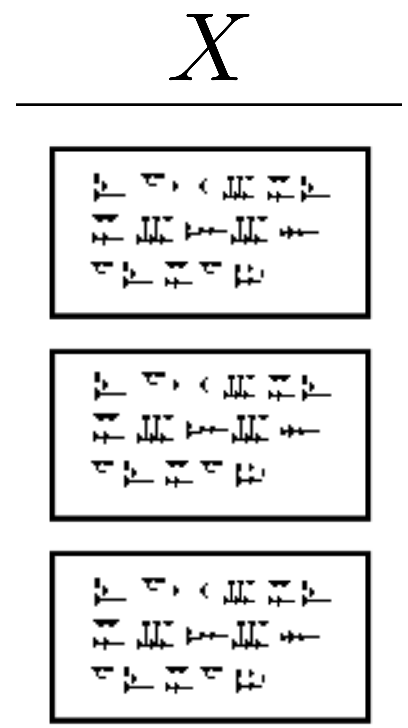


Language Technologies Institute

Supervised Approaches

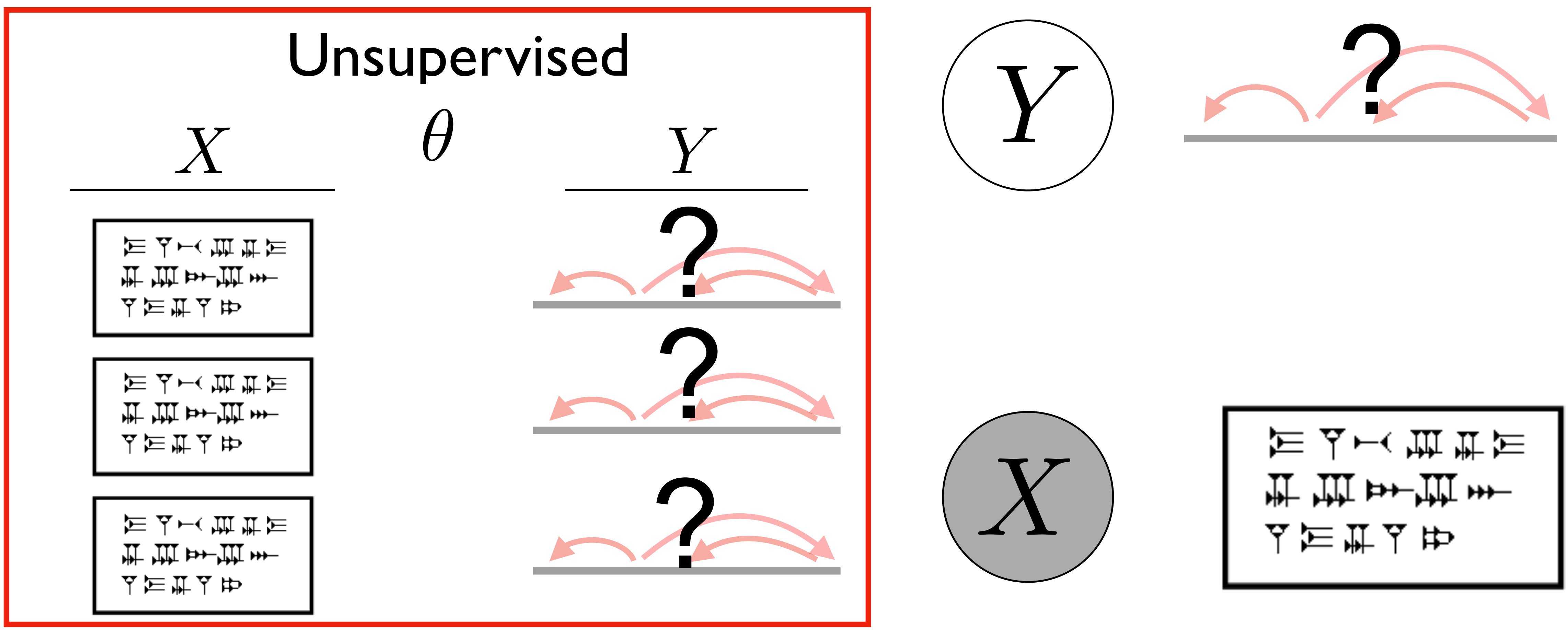


Unsupervised Approaches

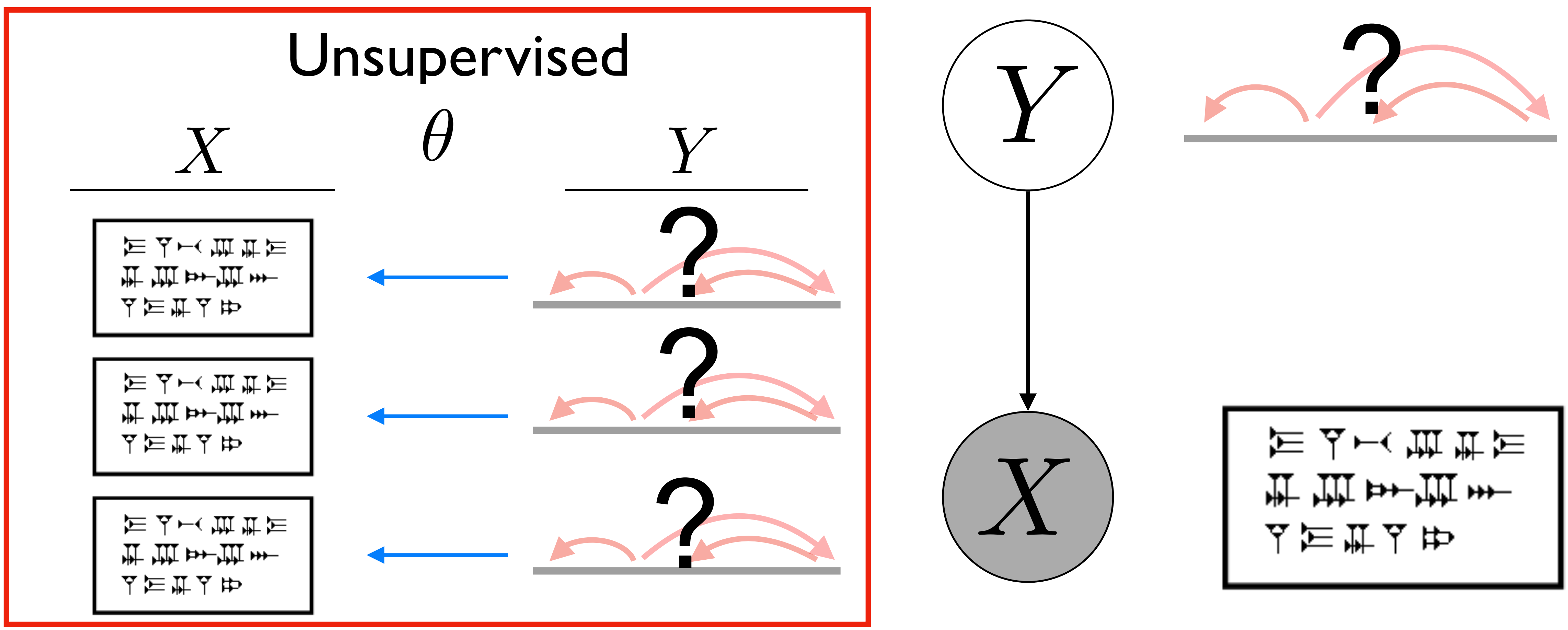


- Learning language models P(X)
- Learning continuous features from language models (e.g. word2vec, skipthought, BERT)
- But how do we turn this into **interpretable structure**?
- How do we do it while **taking advantage of continuous features**?

Latent Variable Approaches



Latent Variable Approaches

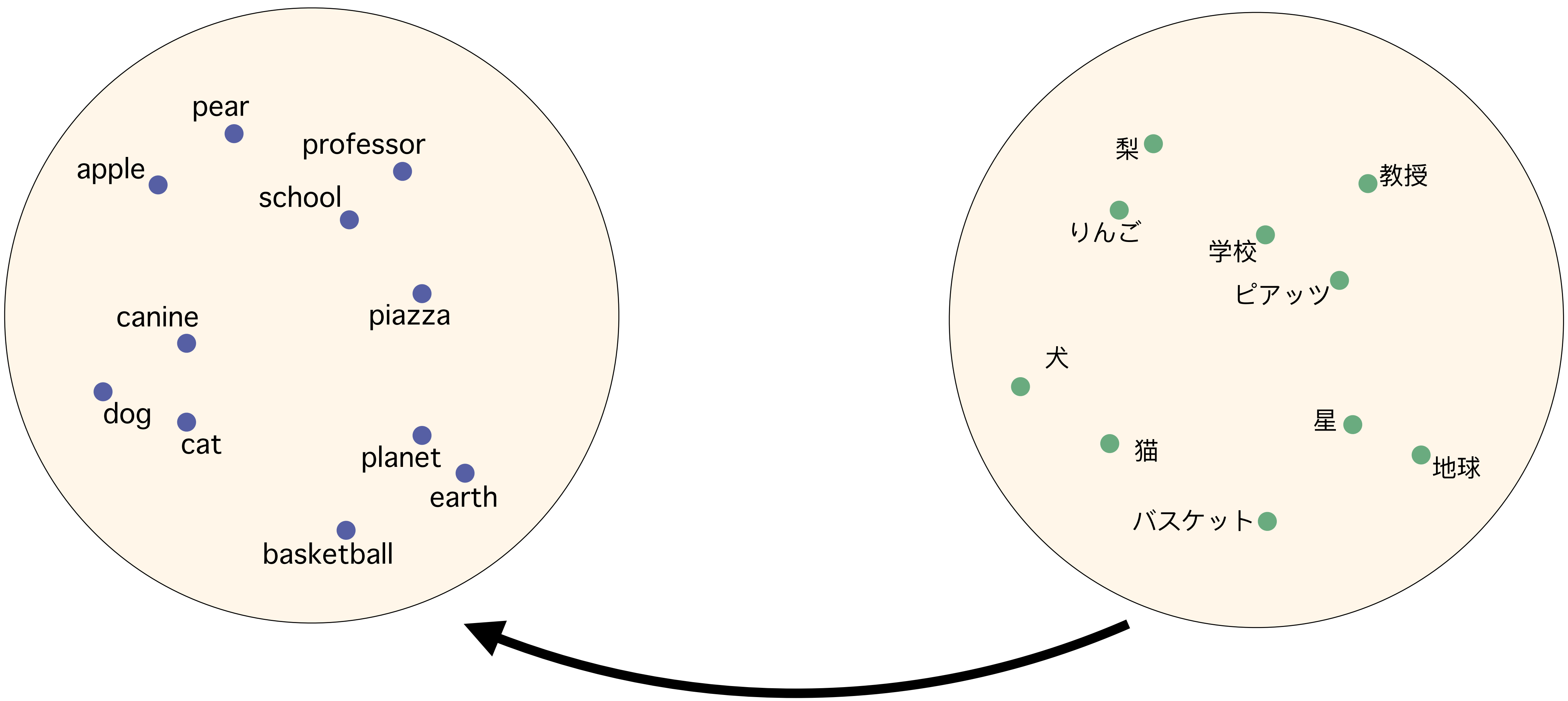


Density Matching for Bilingual Word Embedding

Chunting Zhou, Xuezhe Ma, Di Wang, Graham Neubig

(NAACL 2019)

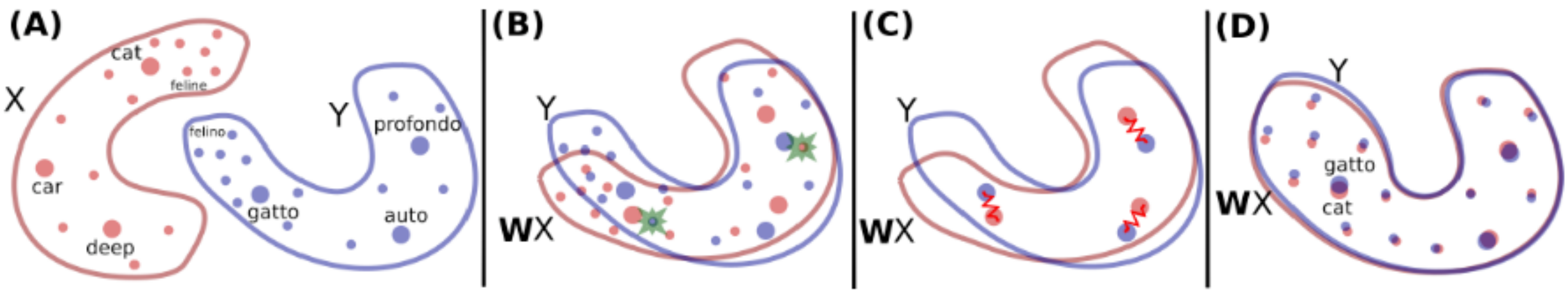
Bilingual Word Embedding



- Map word embeddings from different languages into a single vector space
  - -Cross-lingual transfer
  - -Cross-lingual NLP tasks

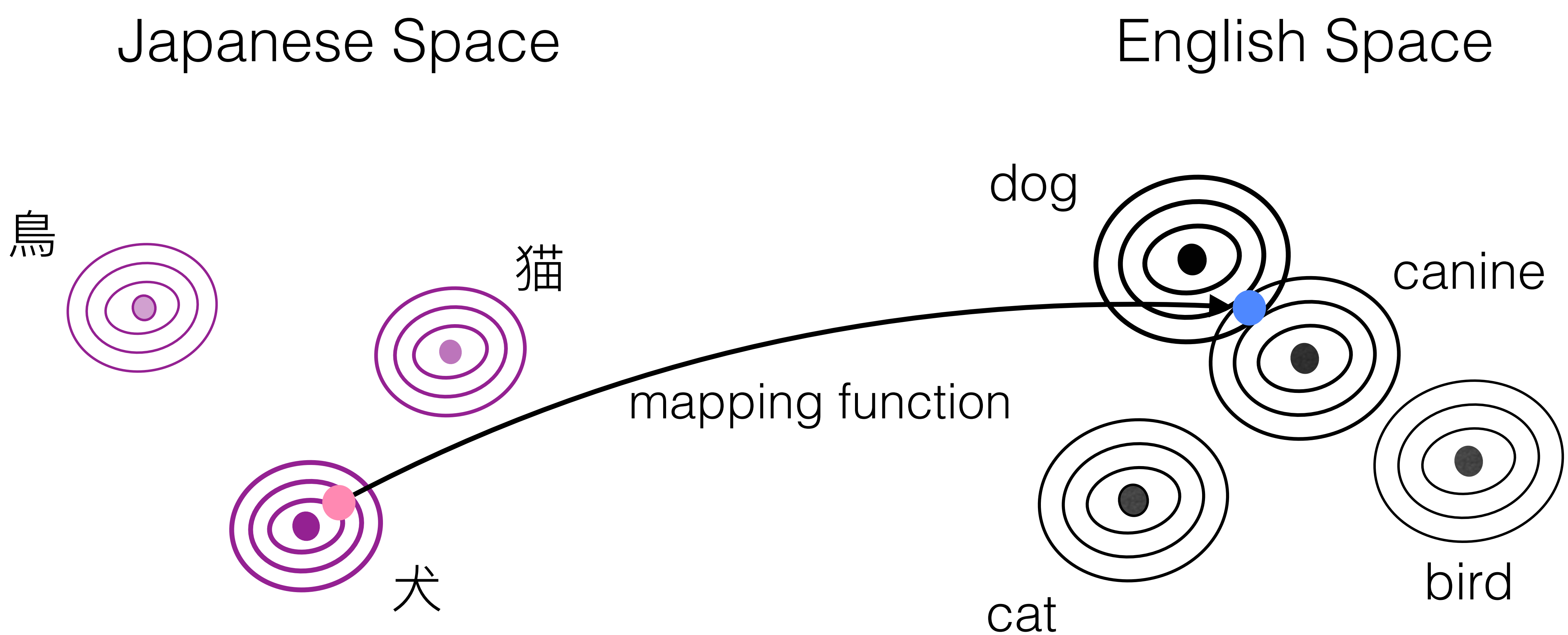
Previous Work on Unsupervised BWE

- Unsupervised methods of minimization some form of distance between distributions of discrete vector sets:



- No direct probabilistic interpretation, not a "typical" unsupervised generative model

Density Mapping for Bilingual Word Embedding (DeMa-BWE)

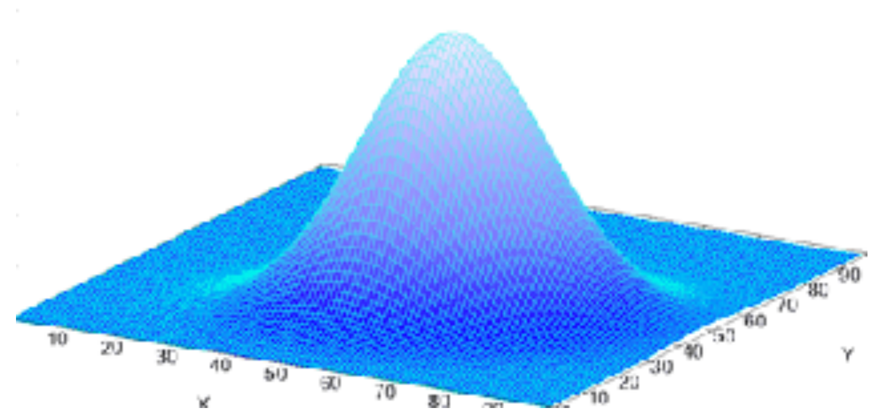


- Mapping function is learned with **normalizing flow**

Normalizing Flows

$$X = f_\theta^{-1}(Z)$$

$$Z = f_\theta(X)$$



$X \sim P(X)$

$Z \sim N(0, I)$

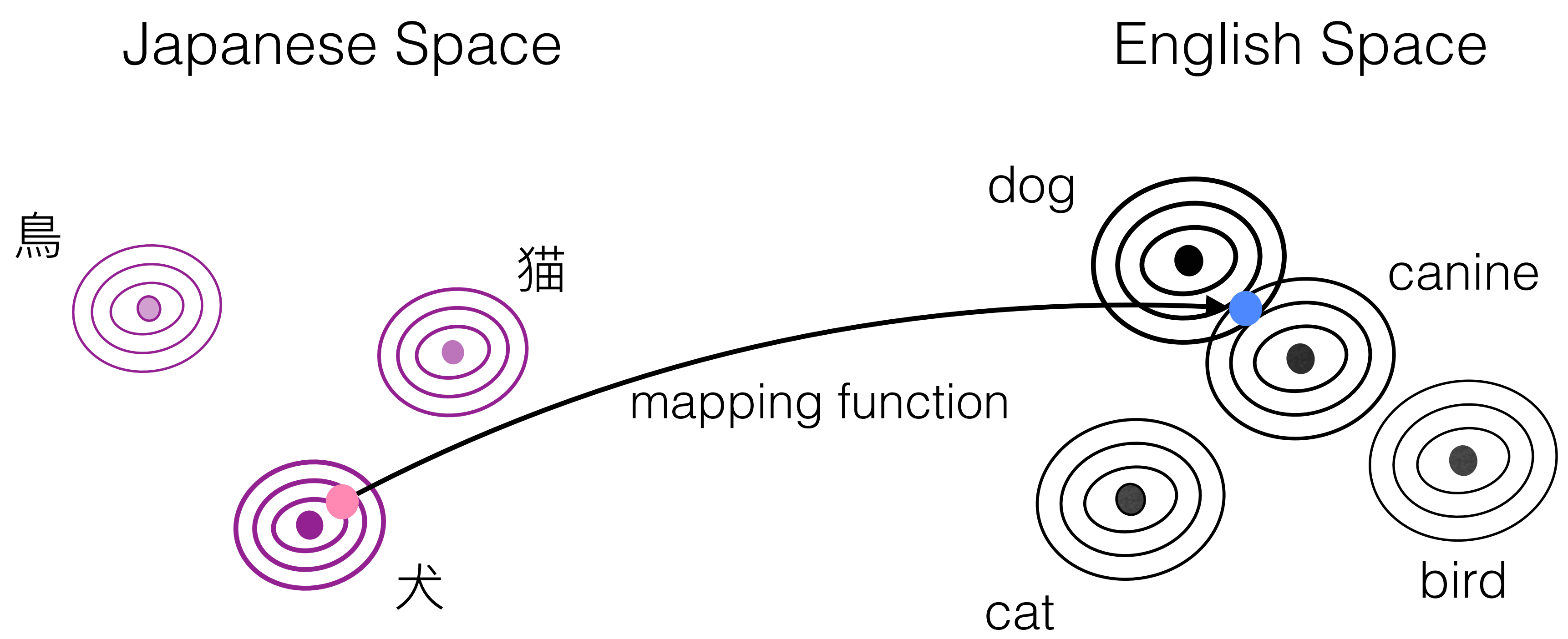
Change of variable formula:

$$p_\theta(x) = p_Z(f_\theta(x)) \boxed{\left|\det\left(\frac{\partial f_\theta(x)}{\partial x}\right)\right|}$$

Intuitively, prevents degenerative mapping of everything to zero vector

Normalizing Flow: A series of such invertible transformations f

DeMa-BWE: Preliminaries



Notations:

$\mathbf{x} \in \mathbb{R}^d, \quad \mathbf{y} \in \mathbb{R}^d$: denote vectors in the src and tgt embedding space

$x_i, \quad y_j$: denote an actual word in src and tgt vocabularies

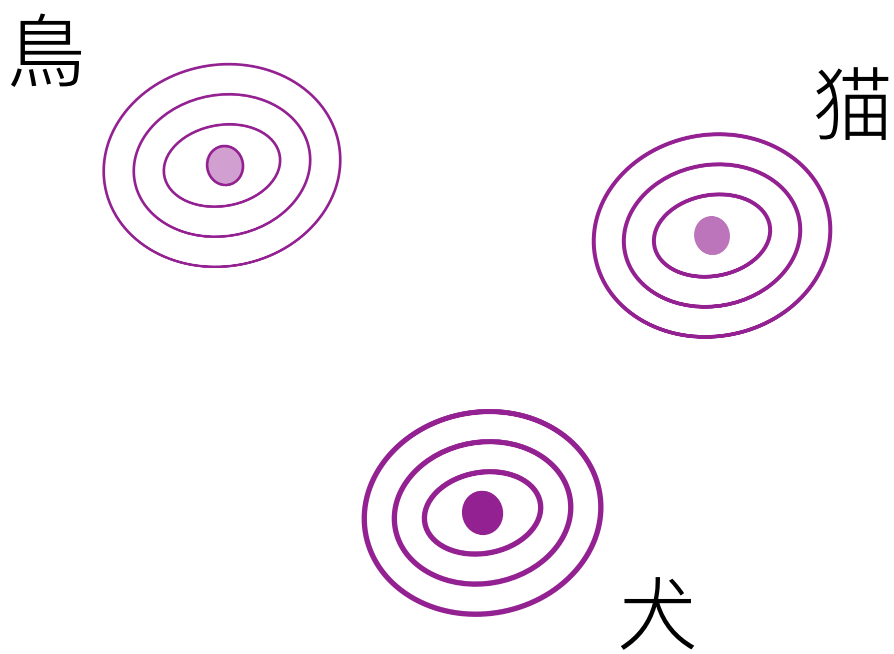
$f_{xy}, \quad f_{yx}$: denote src->tgt, and tgt-src mapping functions

Prior Distribution

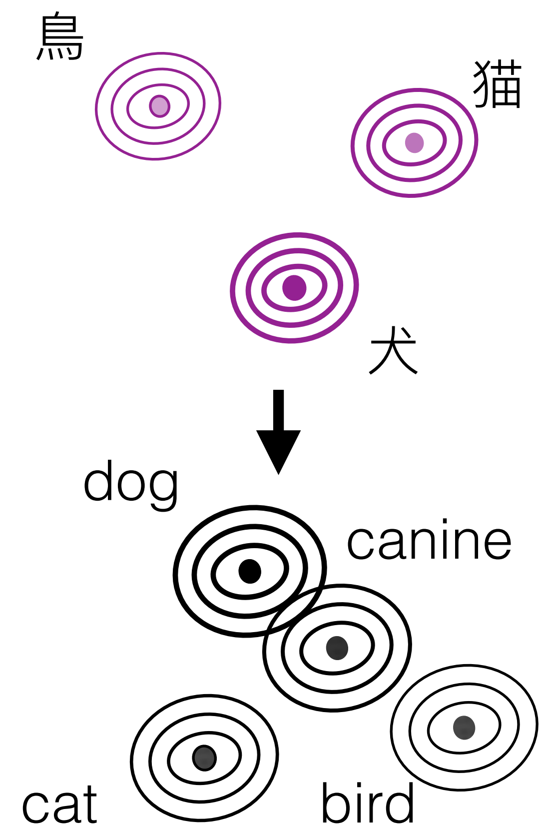
- Assumption on the monolingual word embedding space: Gaussian mixture model

$$p(\mathbf{x}) = \sum_{i \in \{1,\ldots,N_x\}} \pi(x_i)\tilde{p}(\mathbf{x}|x_i)$$

$$\tilde{p}(\mathbf{x}|x_i) = \mathcal{N}(\mathbf{x}|\mathbf{x}_i, \sigma_x^2\mathbf{I})$$



DeMa-BWE: Density Matching



- Sampling a continuous vector from the GMM

$$x_i \sim \pi(x_i) \qquad \mathbf{x} \sim \tilde{p}(\mathbf{x}|x_i)$$

- Apply the mapping function f_{xy} to obtain the transformed vector in the target space.

$$f_{xy}(\cdot) = \mathbf{W}_{xy} \cdot$$

- Computing the density of x in the mapped target space

$$\log p(\mathbf{x}; \mathbf{W}_{xy}) = \log p(\mathbf{y}) + \log \left|\det(\mathbf{W}_{xy})\right|$$

- Objective: $\text{minimize: } \mathrm{KL}(p(\mathbf{x}) || p(\mathbf{x}; \mathbf{W}_{xy}))$

$$\mathcal{L}_{xy} = \mathrm{E}_{\mathbf{x} \sim p(\mathbf{x})}[\log p(\mathbf{y}) + \log \left|\det(\mathbf{W}_{xy})\right|]$$

Language Technologies Institute

Method Details

- **Weak Orthogonality Constraint:** Try to make sure that the transformation is close to orthogonal

$$\mathcal{L}_{bt} = \mathrm{E}_{x_i \sim \pi(x_i), \mathbf{x} \sim \tilde{p}(\mathbf{x}|x_i)} \left[g(\mathbf{W}_{yx} \mathbf{W}_{xy} \mathbf{x}, \mathbf{x}) \right] + \mathrm{E}_{y_j \sim \pi(y_j), \mathbf{y} \sim \tilde{p}(\mathbf{y}|x_j)} \left[g(\mathbf{W}_{xy} \mathbf{W}_{yx} \mathbf{y}, \mathbf{y}) \right]$$

- **Weak Supervision w/ Identical Strings:** Take advantage of the fact that identical strings are usually the same word in both languages

$$\mathcal{L}_{sup} = \sum_{v \in \mathcal{W}_{id}} g(\mathbf{v}_x \mathbf{W}_{xy}^T, \mathbf{v}_y) + g(\mathbf{v}_y \mathbf{W}_{yx}^T, \mathbf{v}_x)$$

- **Alignment Selection Methods:** Use cross-domain similarity local scaling (CSLS)

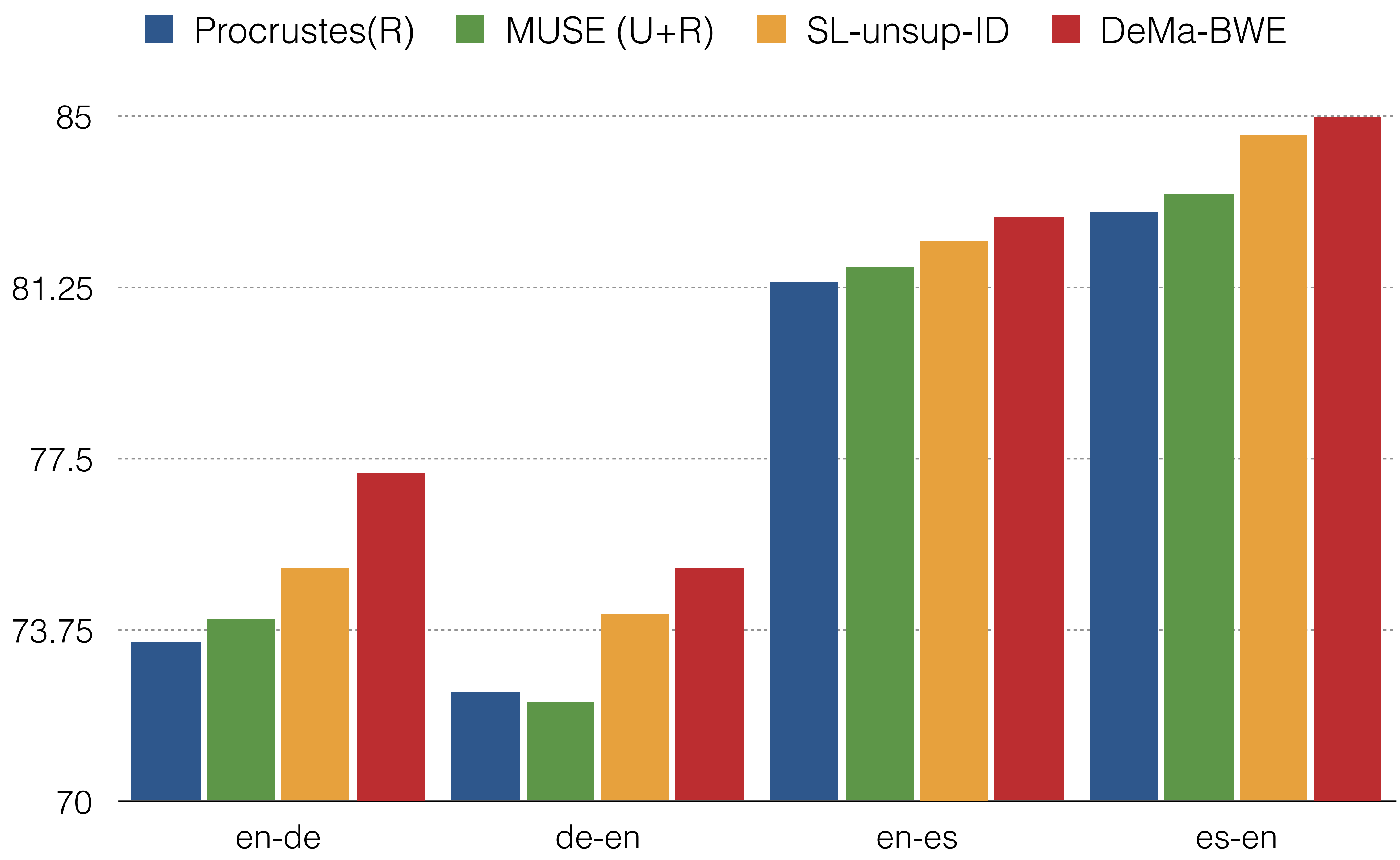
$$\mathrm{CSLS}(\mathbf{x}', \mathbf{y}) = 2\cos(\mathbf{x}', \mathbf{y}) - r_T(\mathbf{x}') - r_S(\mathbf{y})$$

Experiments

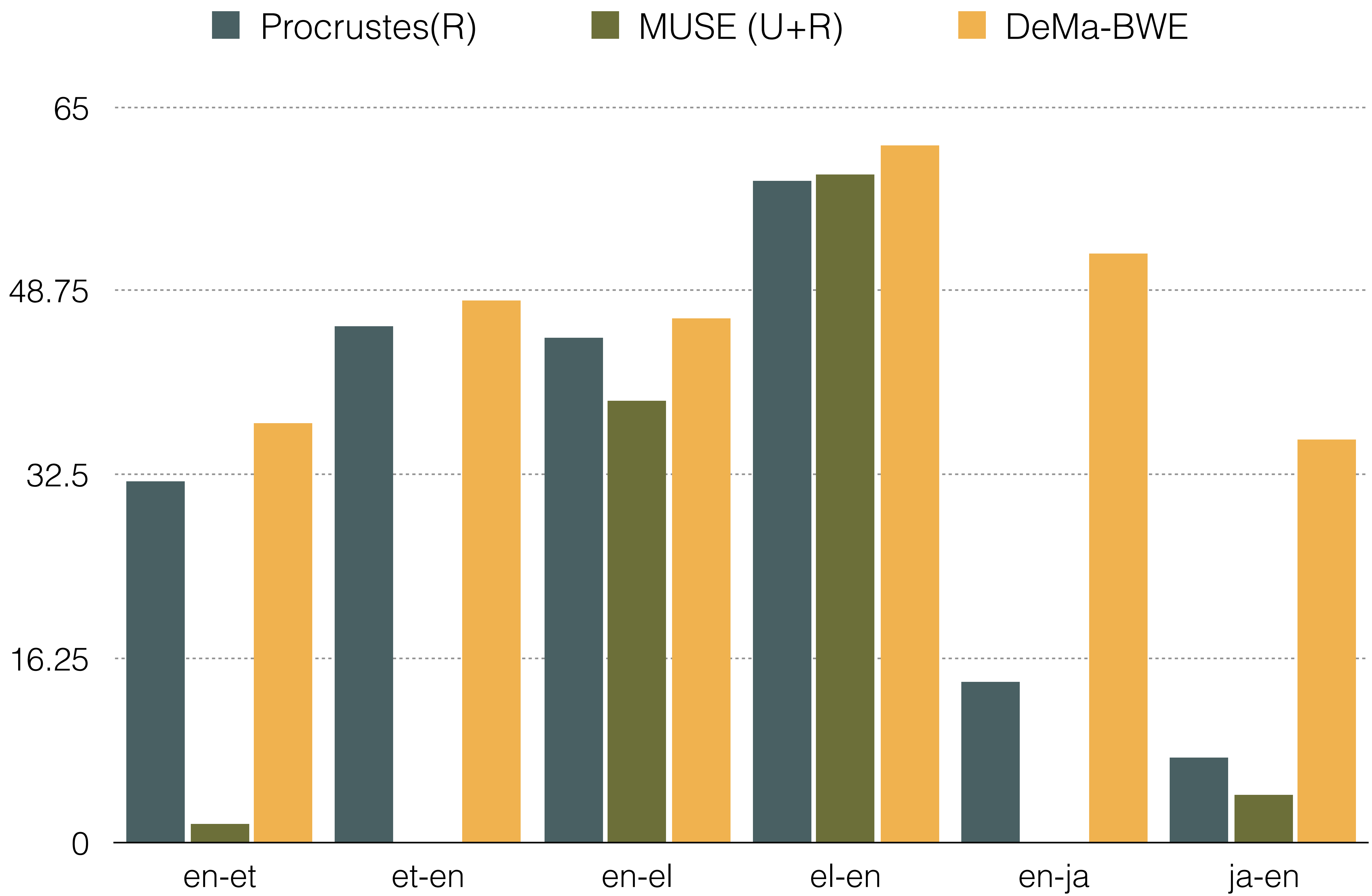
- **Dataset and Tasks**
  - Bilingual Lexicon Induction Task: MUSE dataset (Conneau el al., 2017)
  - Cross-lingual Word Similarity Task: SemEval 2017

- **Languages**
  - Baseline languages: en - es, de, fr, ru, zh, ja
  - Morphologically rich languages: en - et, fi, el, hu, pl, tr

Main Results on BLI (close languages)



Main Results on BLI (distant languages)

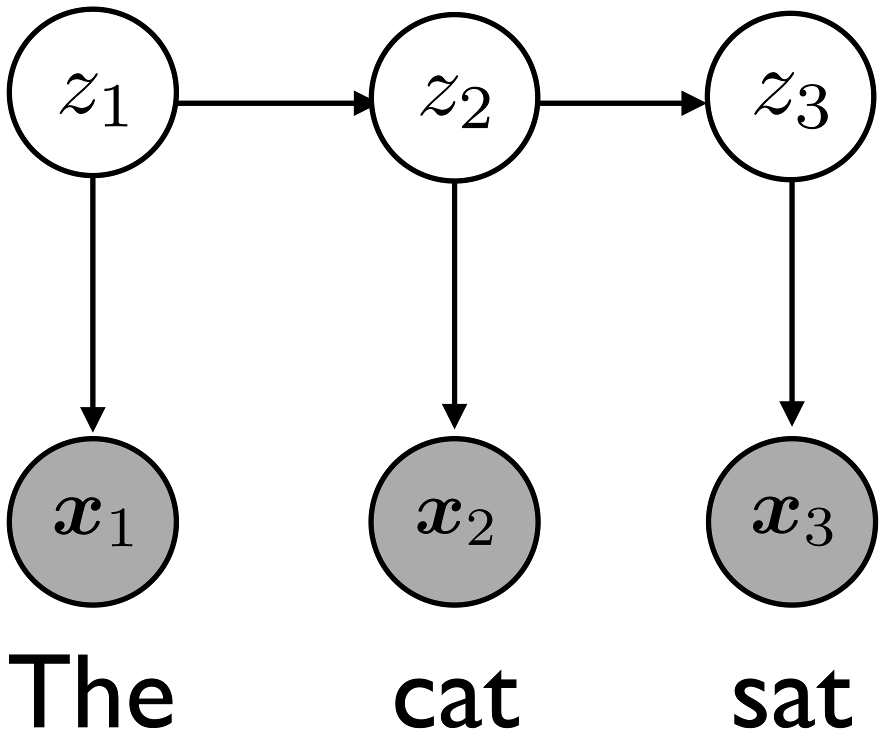


Unsupervised Learning of Syntactic Structure w/ Invertible Neural Projections

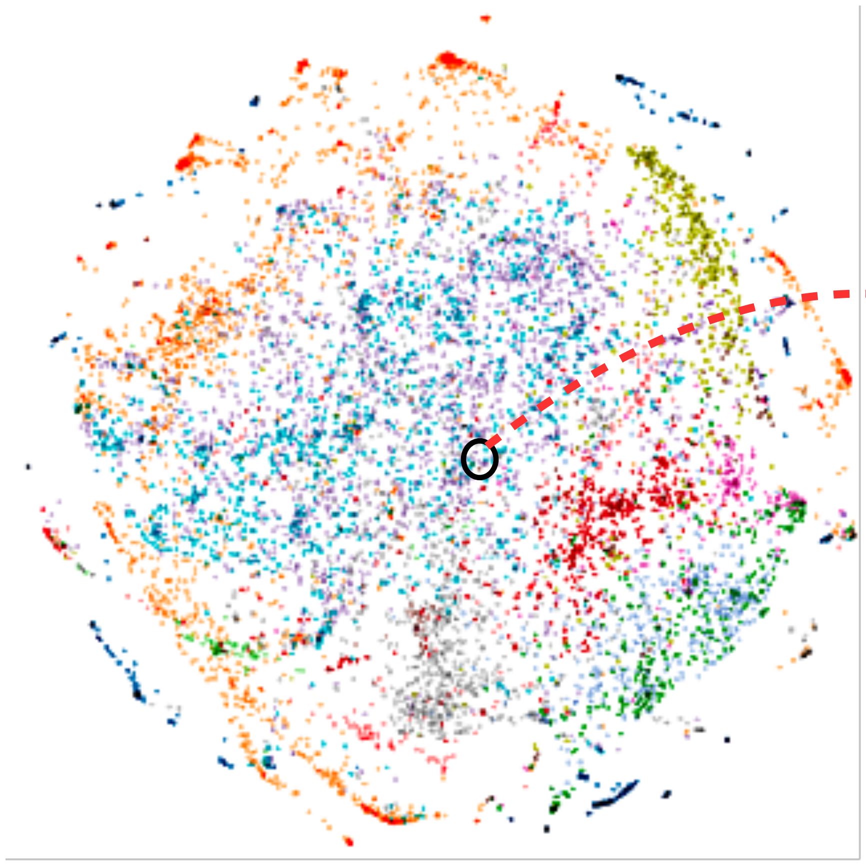
Junxian He, Graham Neubig, Taylor Berg-Kirkpatrick

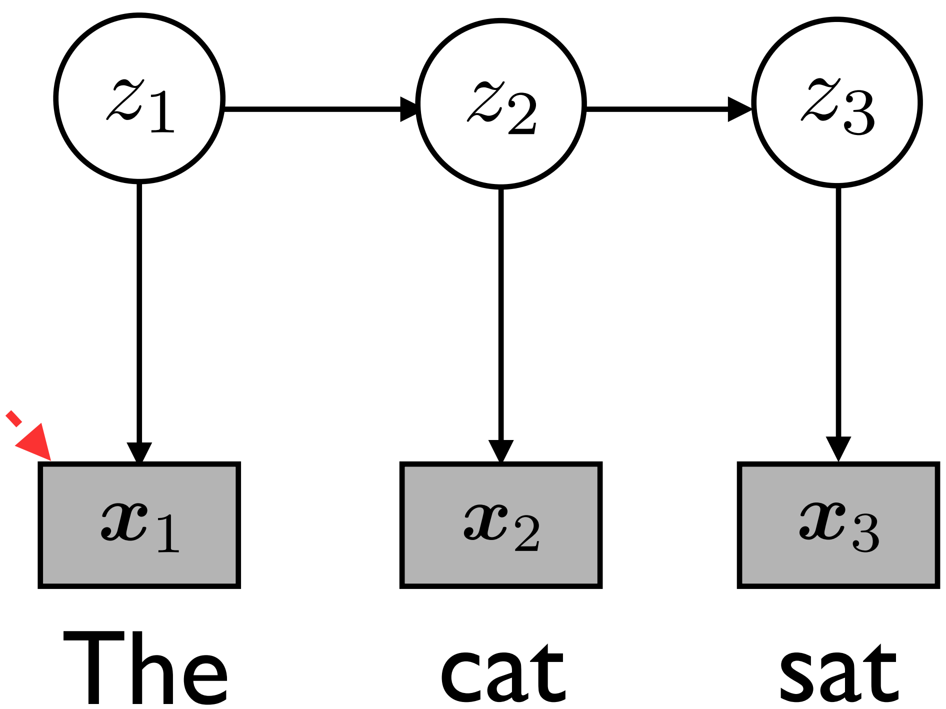
(EMNLP 2018)

HMM for Part-of-Speech Induction



Gaussian HMM for POS Induction

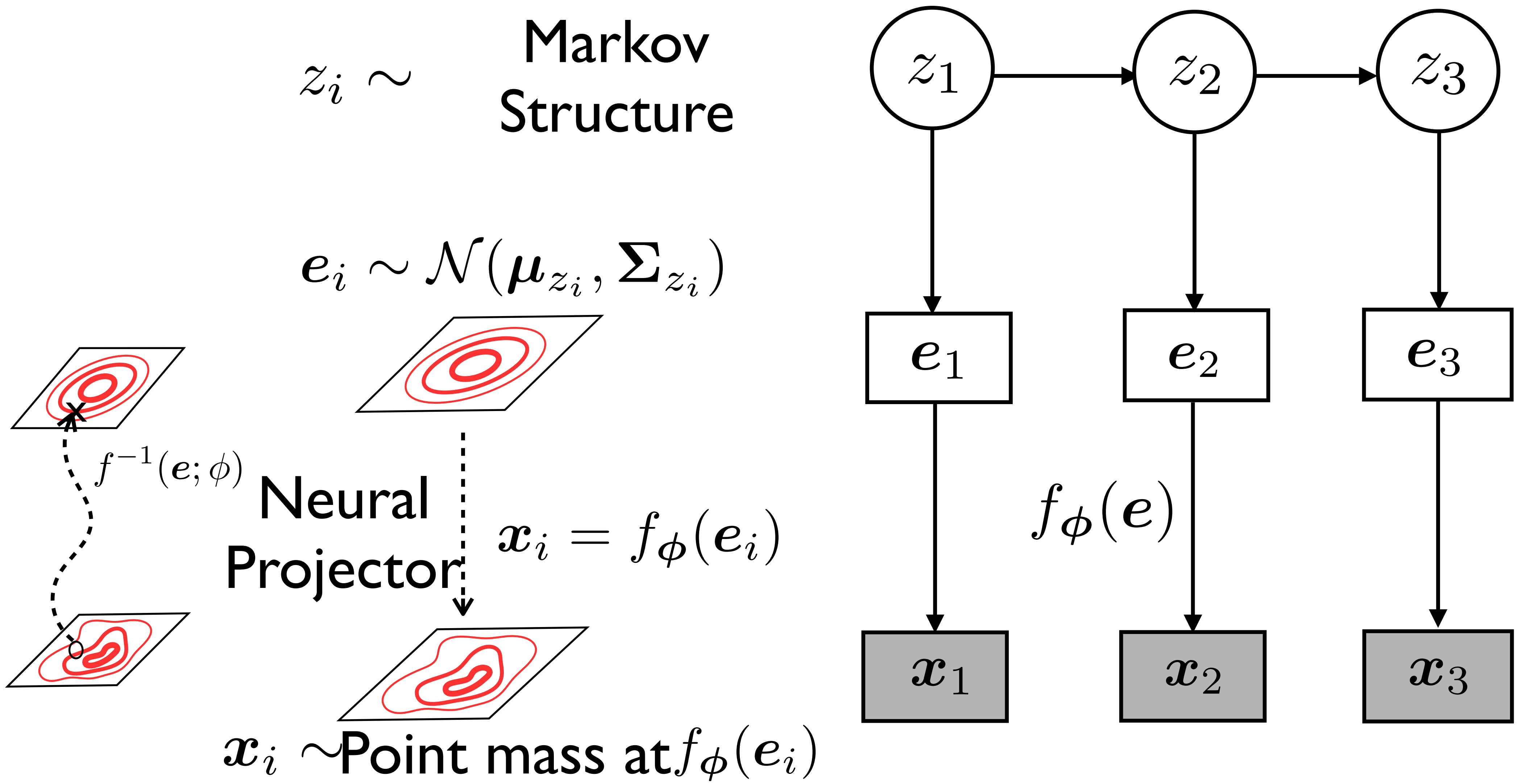




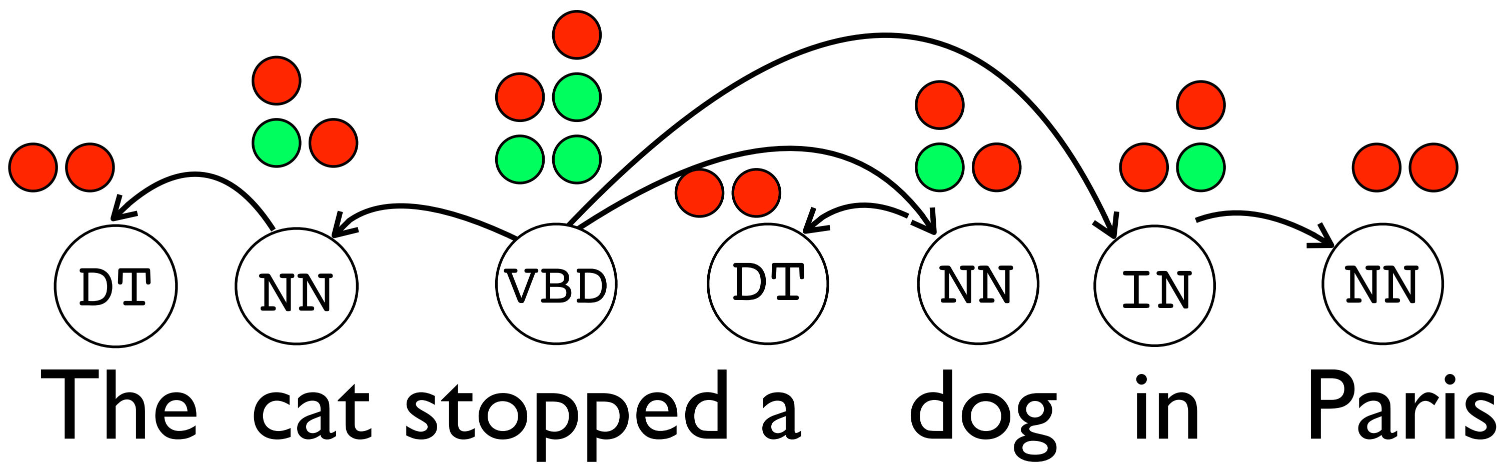
$$\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_{z_i}, \Sigma_{z_i})$$

[Lin et al. 2015]

Latent Embeddings w/ Neural Projection

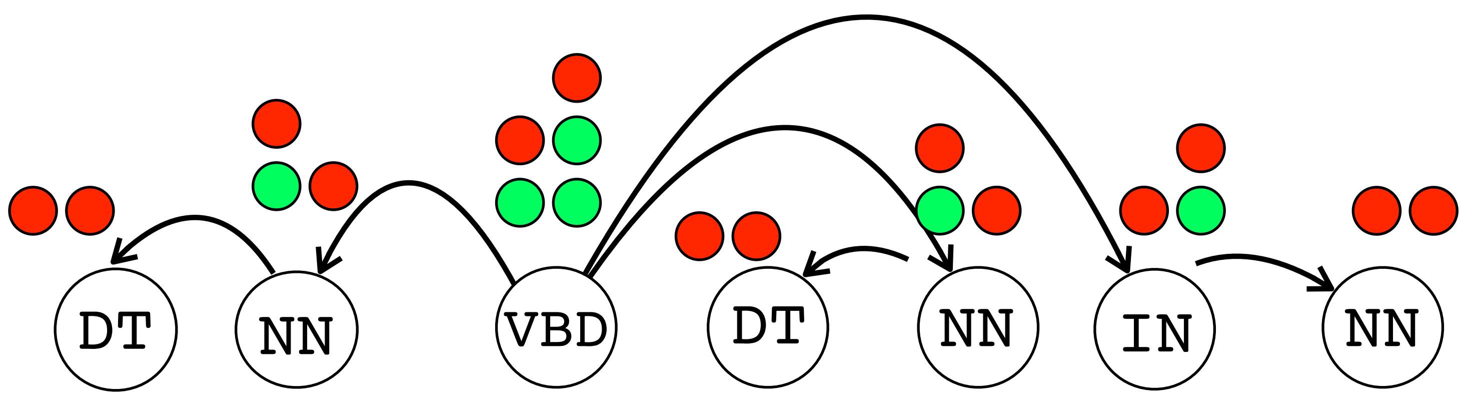


Dependency Model with Valence



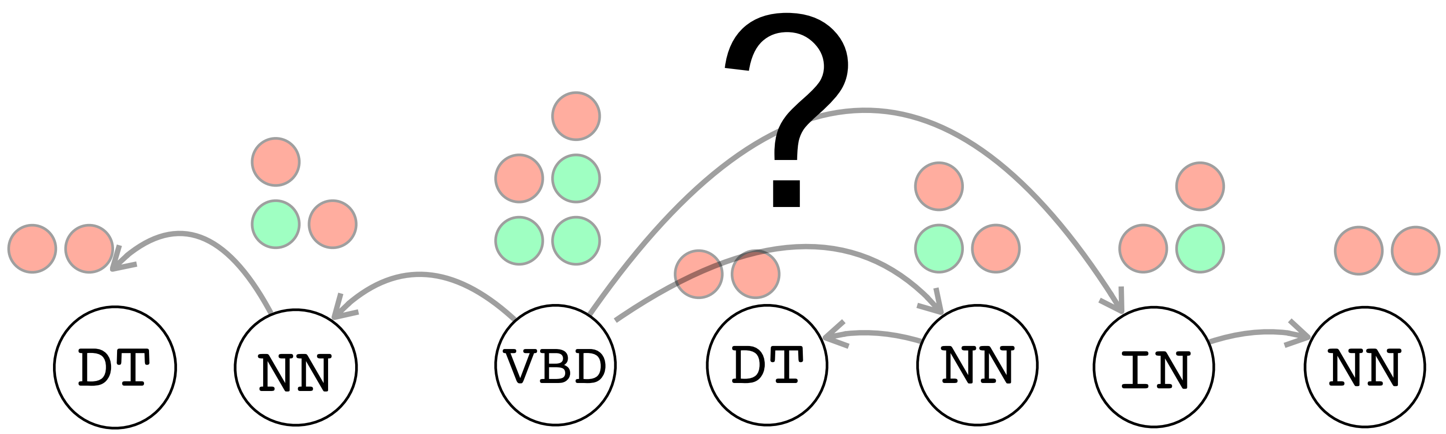
[Klein and Manning 2004]

Dependency Model with Valence

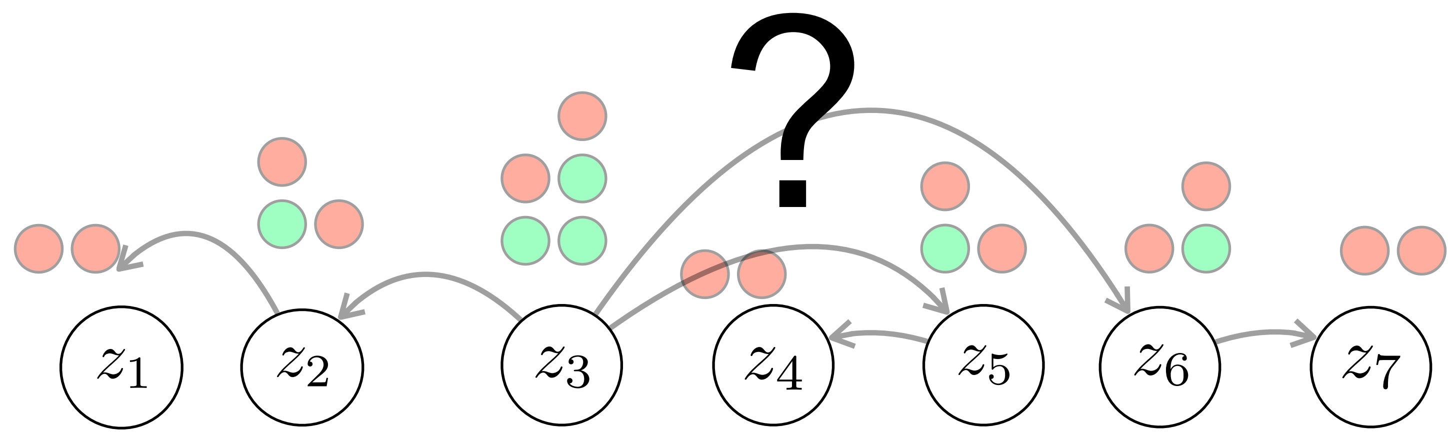


[Klein and Manning 2004]

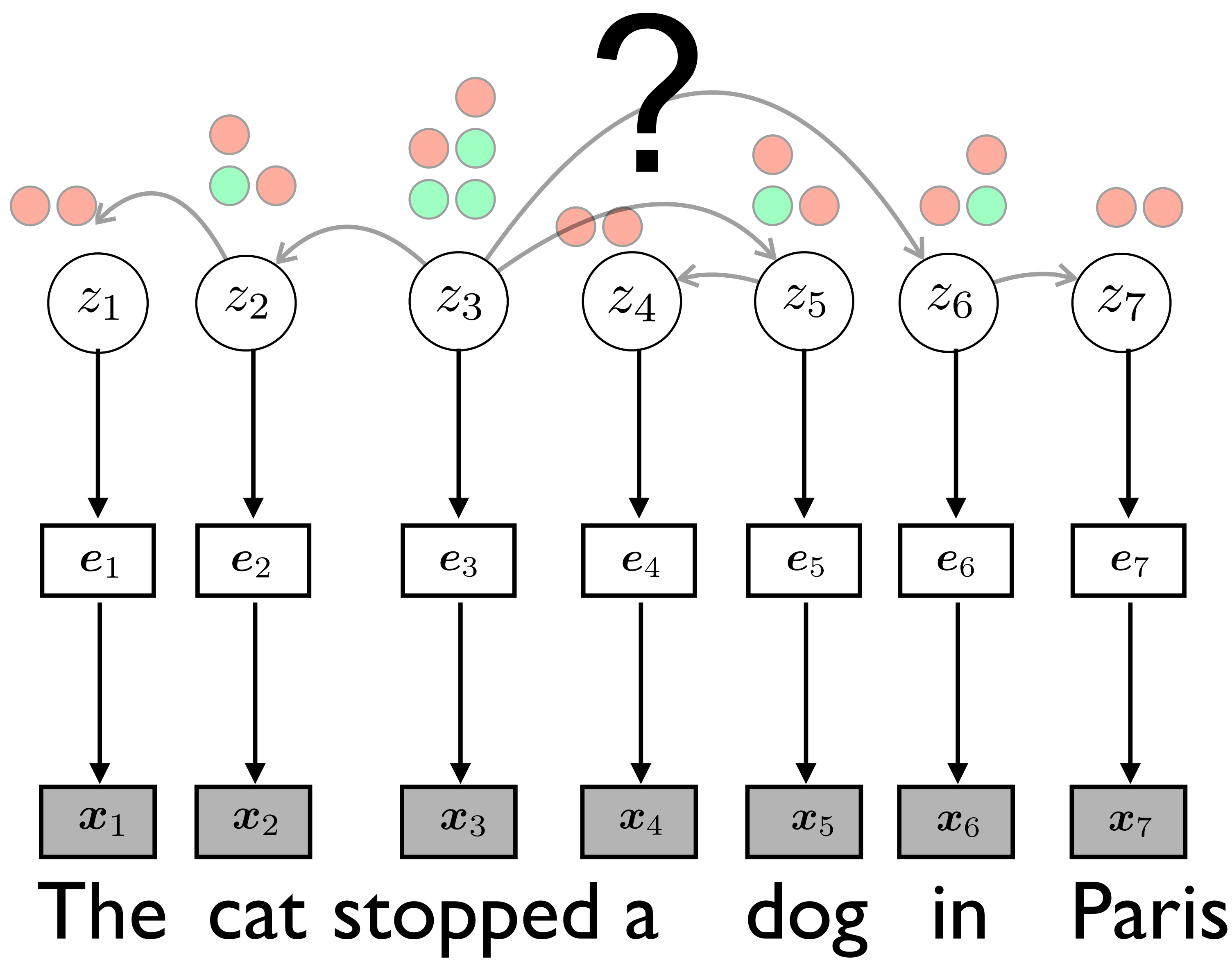
Dependency Parse Induction from POS



Grammar Induction from Raw Text



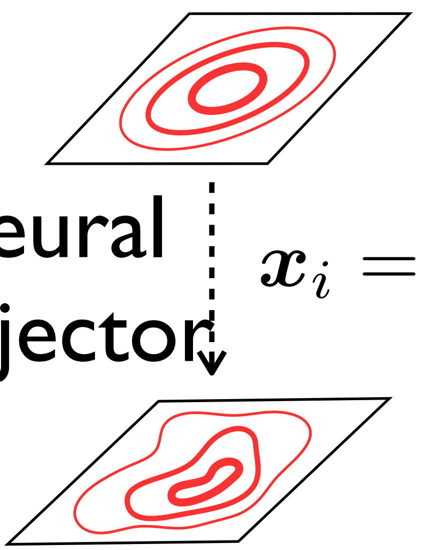
Grammar Induction from Raw Text



Latent Embeddings w/ Neural Projection

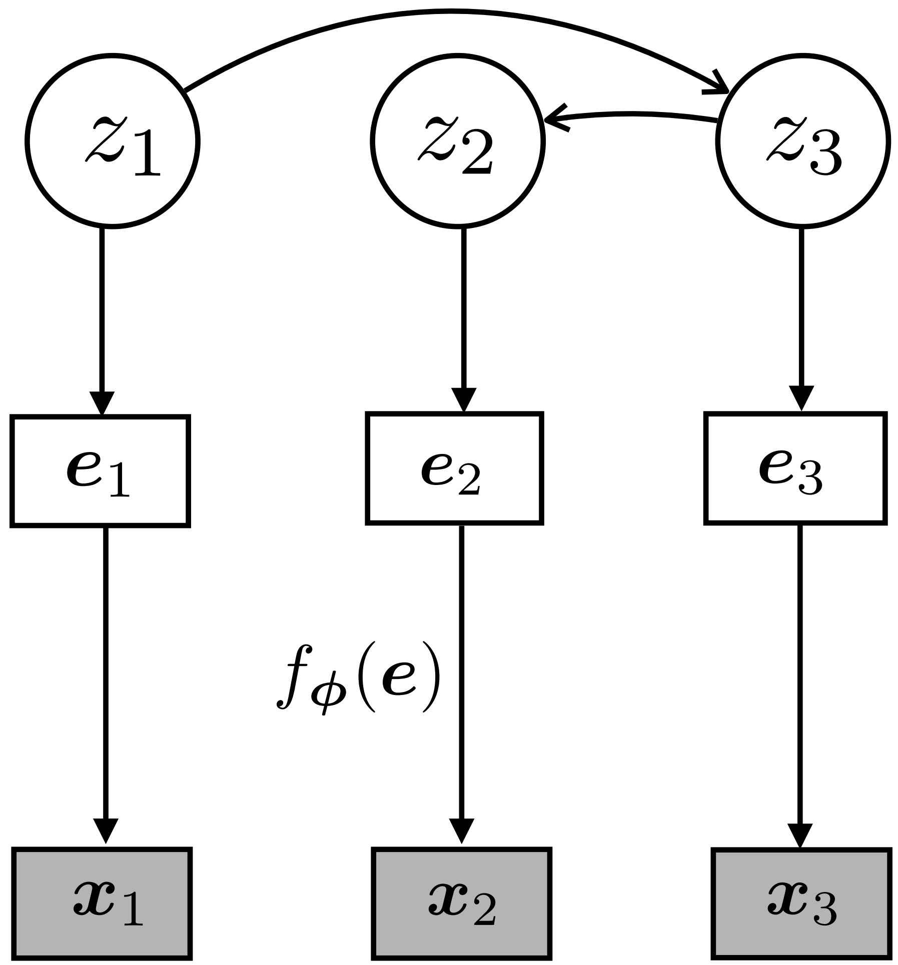
$z_i \sim$ Syntax Model

$\boldsymbol{e}_i \sim \mathcal{N}(\boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i})$

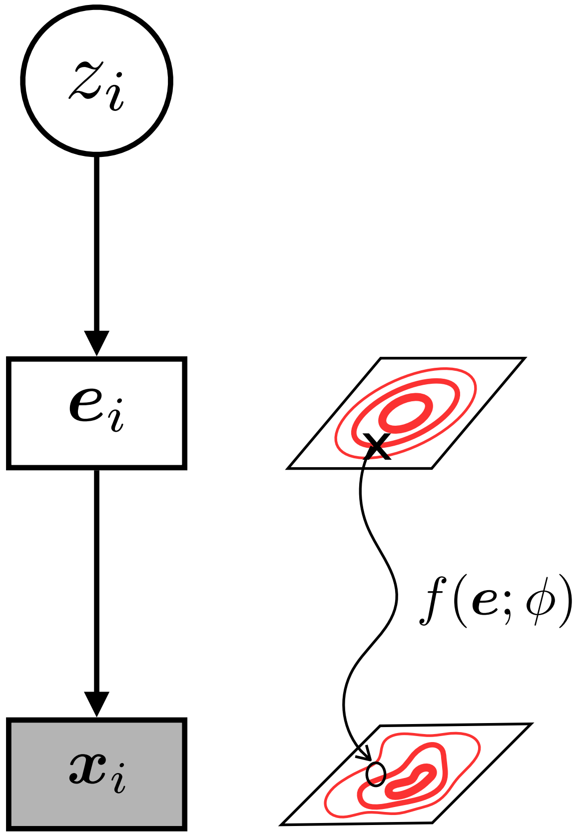


Neural Projector $\boldsymbol{x}_i = f_{\boldsymbol{\phi}}(\boldsymbol{e}_i)$

$\boldsymbol{x}_i \sim$ Point mass at $f_\phi(e_i)$

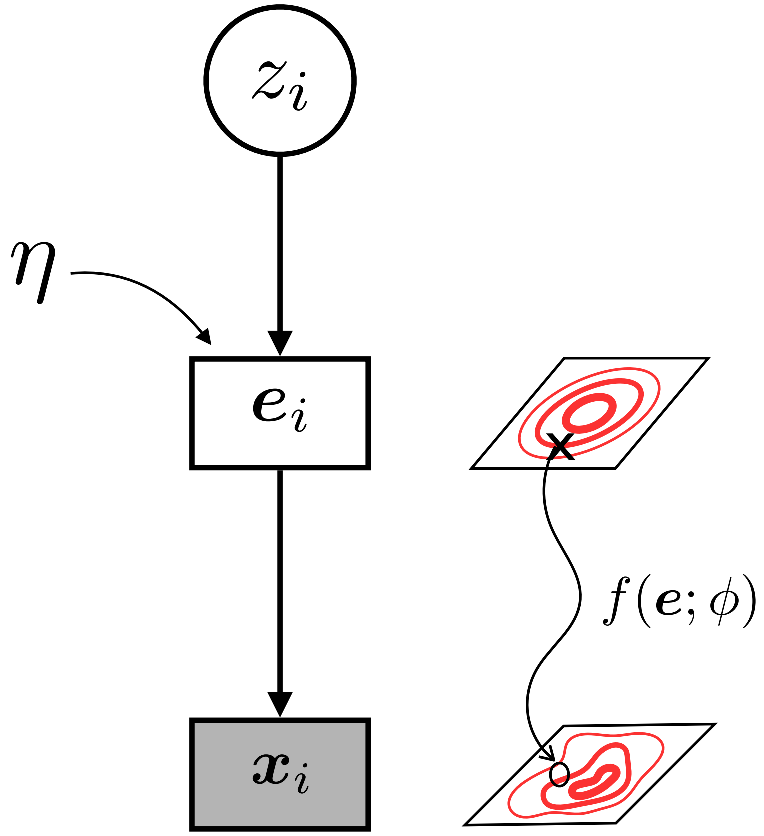


Learning and Inference



$$p(\boldsymbol{x}_i | z_i; \eta, \phi)$$

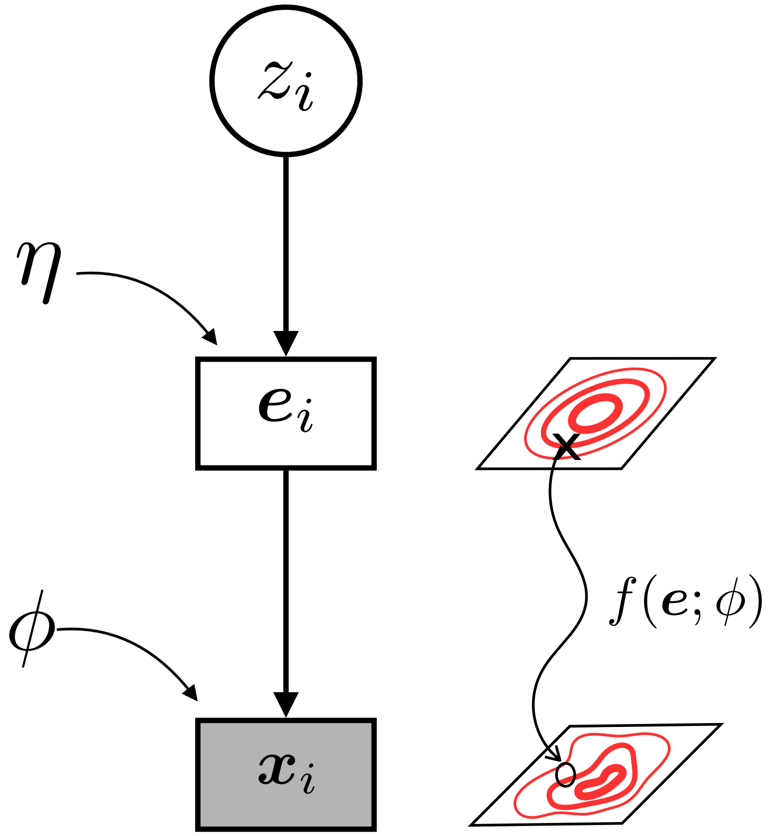
Learning and Inference



$$p(\boldsymbol{x}_i | z_i; \eta, \phi)$$

Gaussian embedding parameters

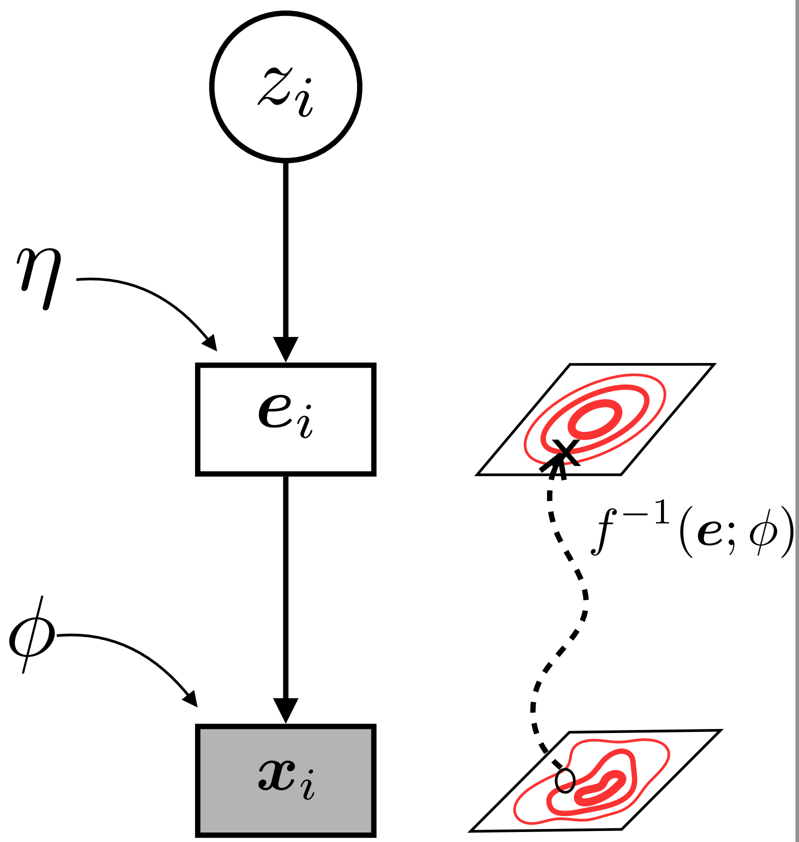
Learning and Inference



$$p(\boldsymbol{x}_i | z_i; \eta, \phi)$$

Projection parameters

Learning and Inference

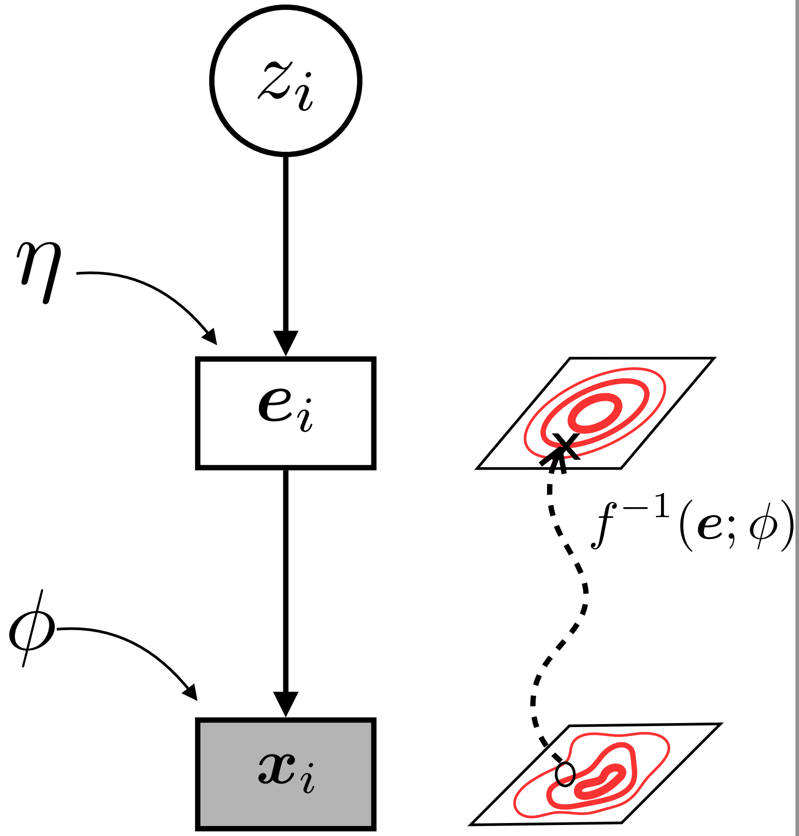


dim(x) = dim(e) and f is invertible

$$p(\boldsymbol{x}_i | z_i; \eta, \phi)$$

$$= p(f_\phi^{-1}(\boldsymbol{x}_i) | z_i; \eta) \left| \det \frac{\partial f^{-1}}{\partial \boldsymbol{x}_i} \right|$$

Learning and Inference

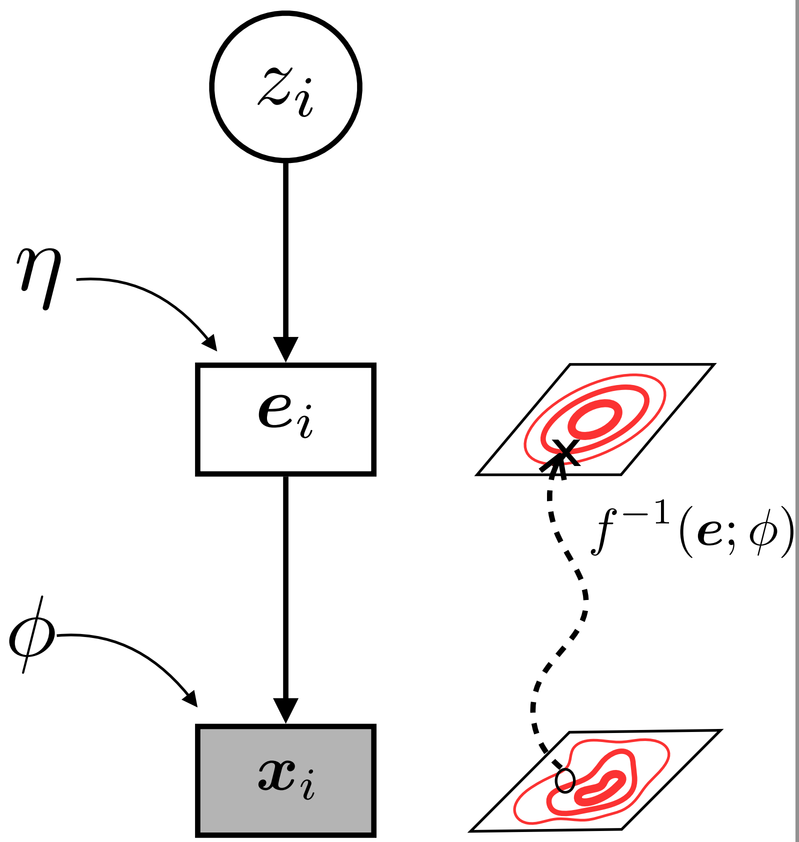


dim(x) = dim(e) and f is invertible

$$p(\boldsymbol{x}_i | z_i; \eta, \phi)$$
$$= p(f_\phi^{-1}(\boldsymbol{x}_i) | z_i; \eta) \left| \det \frac{\partial f^{-1}}{\partial \boldsymbol{x}_i} \right|$$

Determinant of Jacobian matrix

Learning and Inference



dim(x) = dim(e) and f is invertible

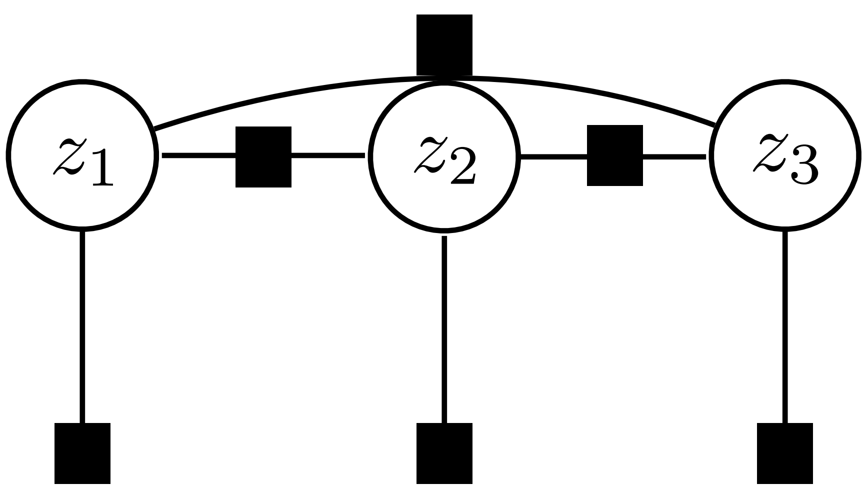
$$p(\boldsymbol{x}_i|z_i;\eta,\phi)$$

$$= p(f_\phi^{-1}(\boldsymbol{x}_i)|z_i;\eta)\left|\det\frac{\partial f^{-1}}{\partial \boldsymbol{x}_i}\right|$$

Gaussian distribution

Determinant of Jacobian matrix

Learning and Inference



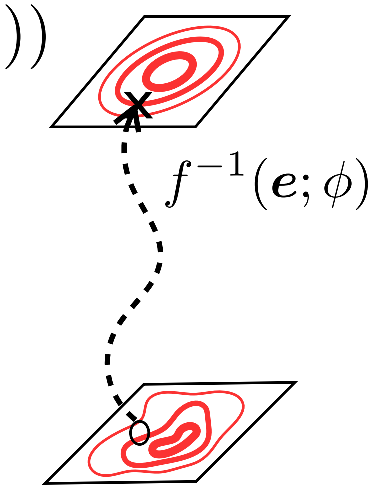
$$\boxed{p(f_\phi^{-1}(\boldsymbol{x}_i)|z_i;\eta)\left|\det\frac{\partial f^{-1}}{\partial \boldsymbol{x}_i}\right|}$$

Example of Markov prior

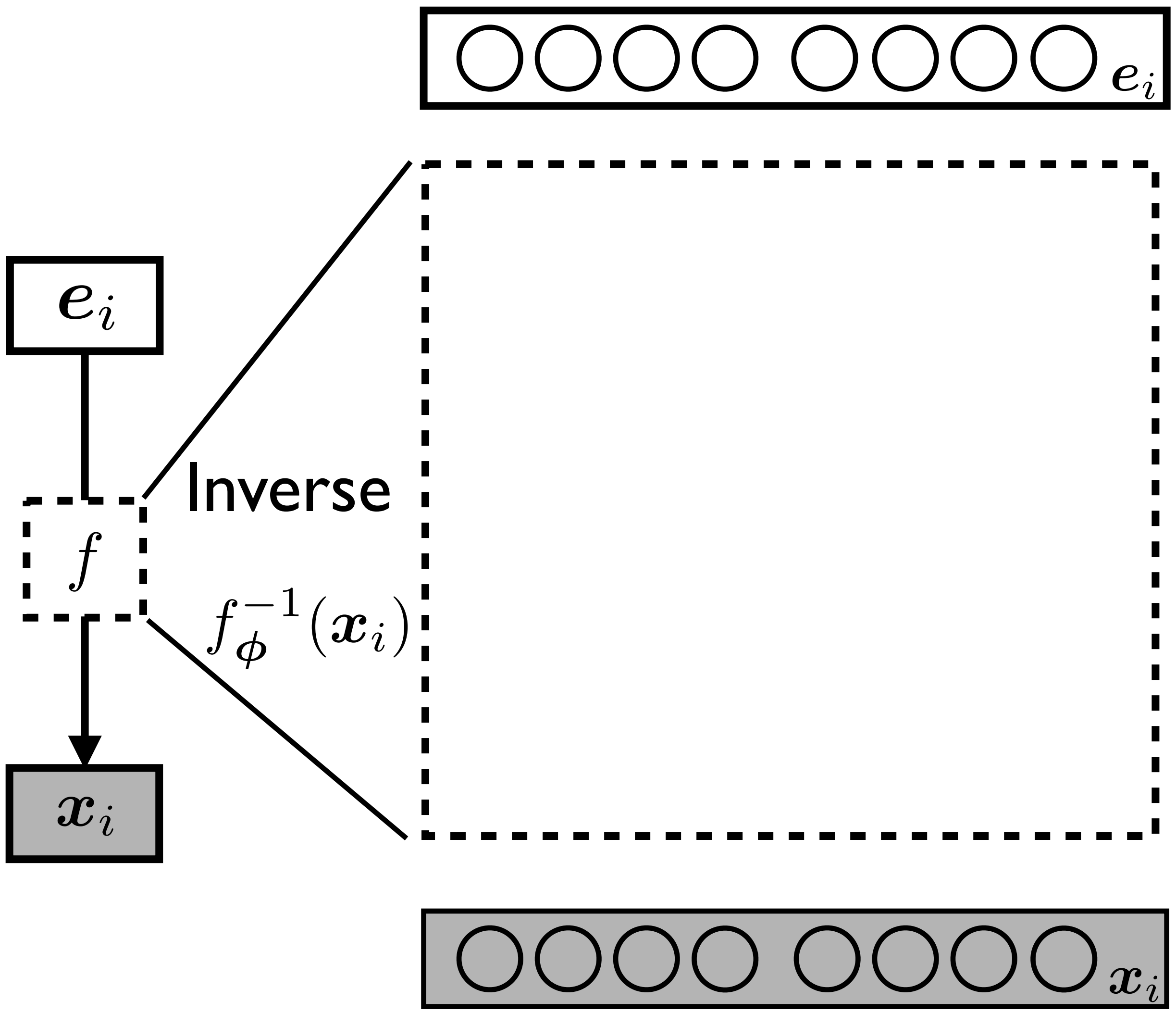
$$\log p(\boldsymbol{x}) = \log p_{\text{GHMM}}(f_\phi^{-1}(\boldsymbol{x}))$$

$$+\sum \log\left|\det\frac{\partial f_\phi^{-1}}{\partial \boldsymbol{x}_i}\right|$$

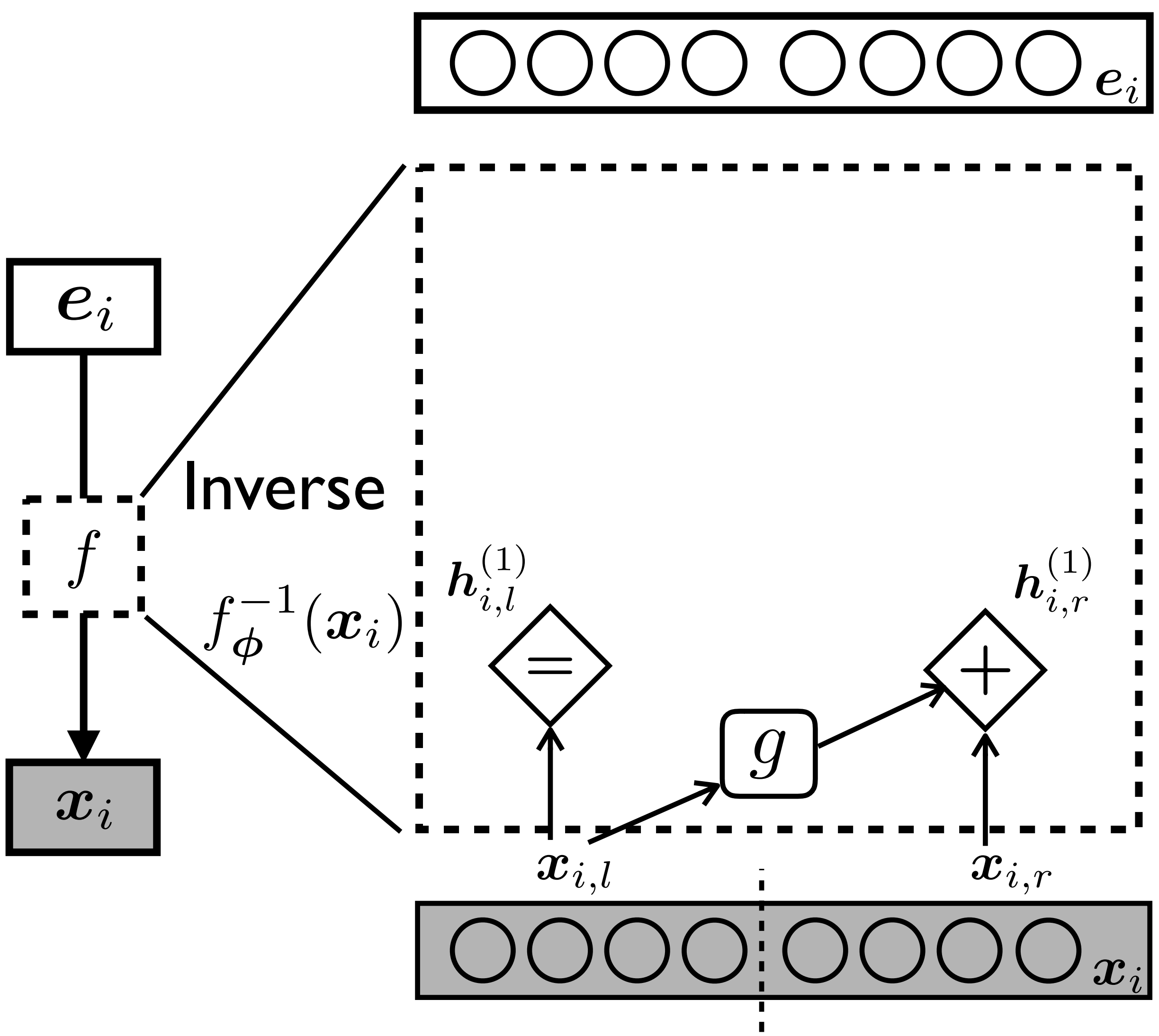
$-\infty$ when f is not invertible



Learning with Inverse Projection



Learning with Inverse Projection



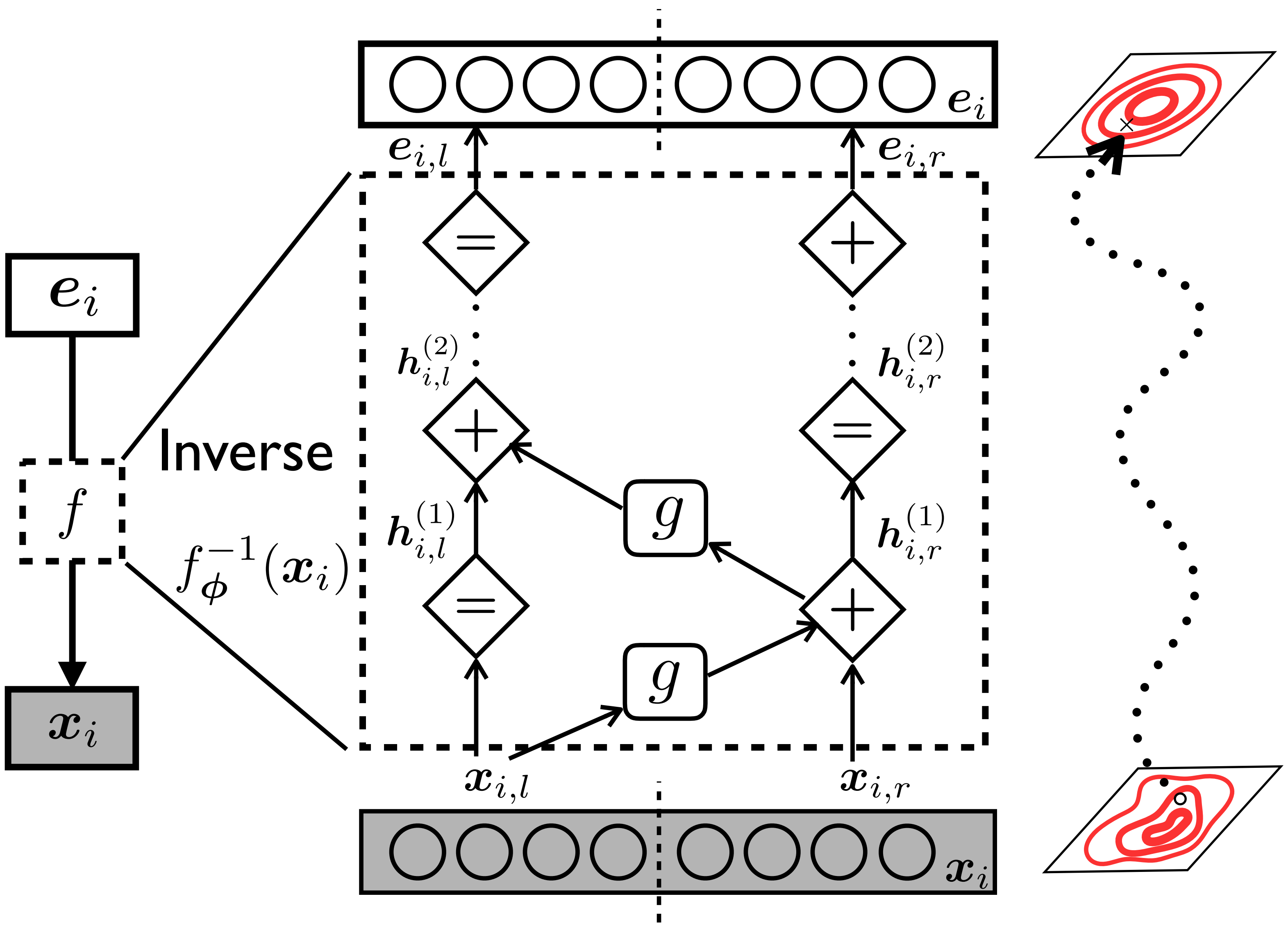
$$h_{i,l}^{(1)} = x_{i,l}$$

$$h_{i,r}^{(1)} = x_{i,r} + g(x_{i,l})$$

$$\boxed{\begin{matrix} 1 & & & & & \\ & 1 & & & 0 & \\ & & 1 & & & \\ & & & 1 & & \\ & \iddots & & & 1 & \\ & & & & & 1 \end{matrix}}$$

[Dinh et al. 2014]

Learning with Inverse Projection



$$\boldsymbol{h}_{i,l}^{(1)} = \boldsymbol{x}_{i,l}$$

$$\boldsymbol{h}_{i,r}^{(1)} = \boldsymbol{x}_{i,r} + g(\boldsymbol{x}_{i,l})$$

$$\boxed{\begin{matrix} 1 & & & & & \\ & 1 & & 0 & & \\ & & 1 & & & \\ & & & 1 & & \\ & \iddots & & & 1 & \\ & & & & & 1 \end{matrix}}$$

[Dinh et al. 2014]

Experiments

- Dataset: English Penn Treebank
- POS tagging

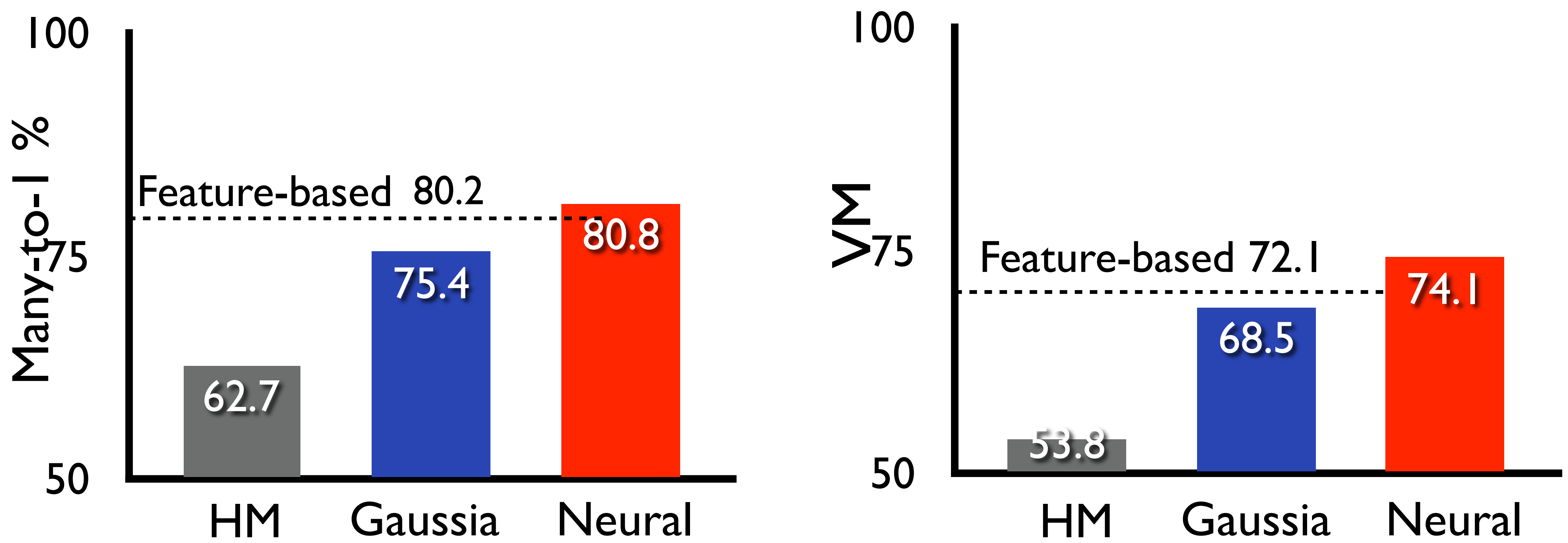
  Trained and tested on whole PTB

- Grammar induction

  Trained on sentences of length <= 10 in section 2-21

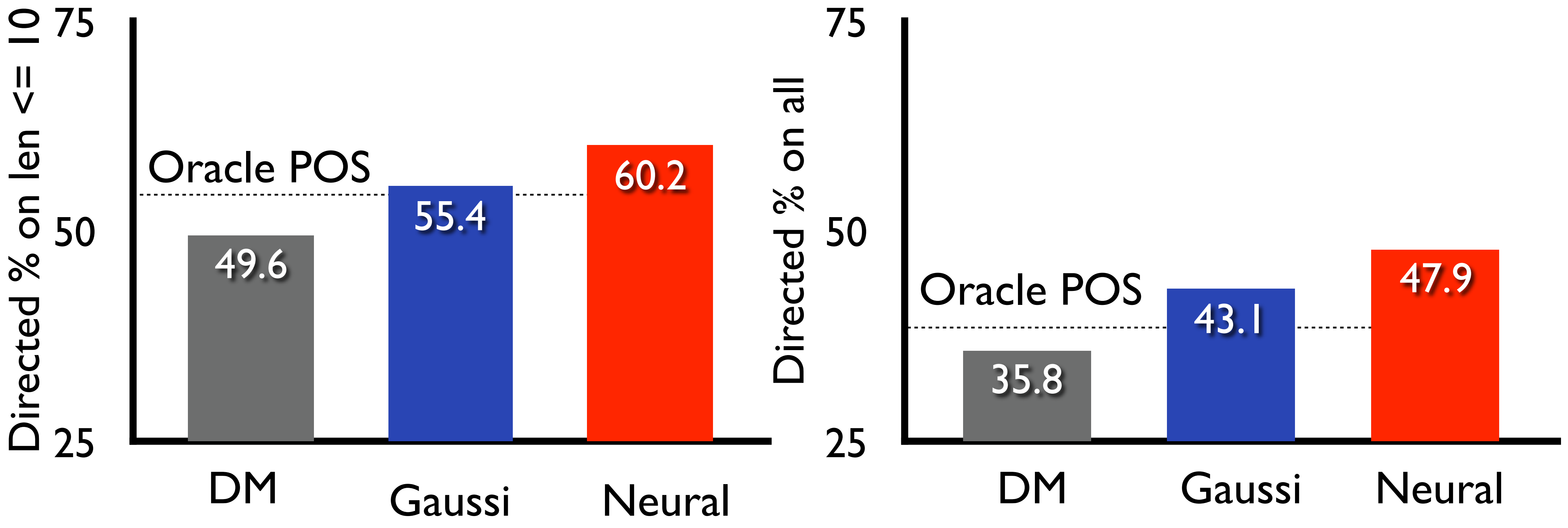
  Tested on sentences in section 23

Part-of-speech Induction

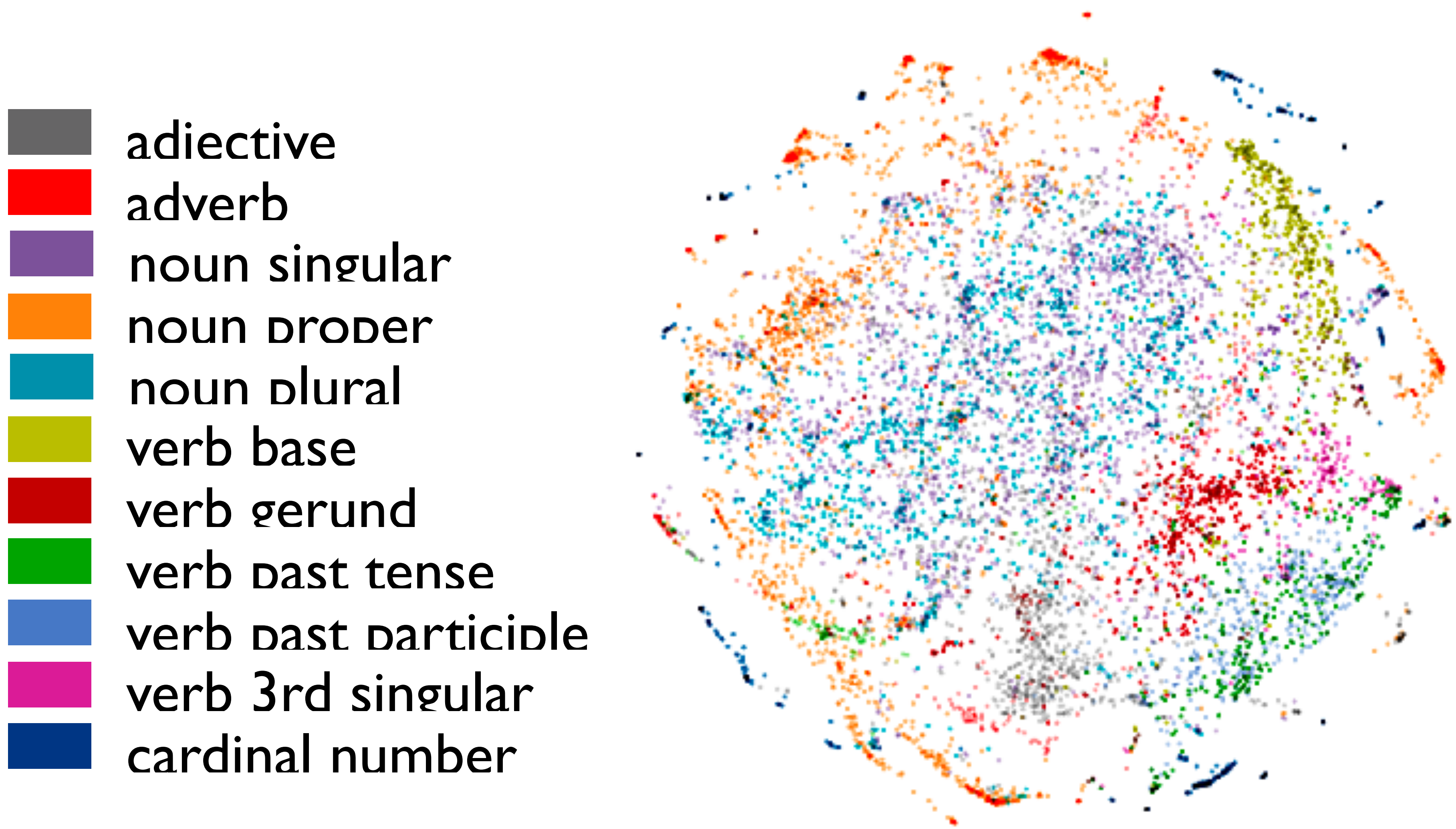


Outperform feature-based SOTA

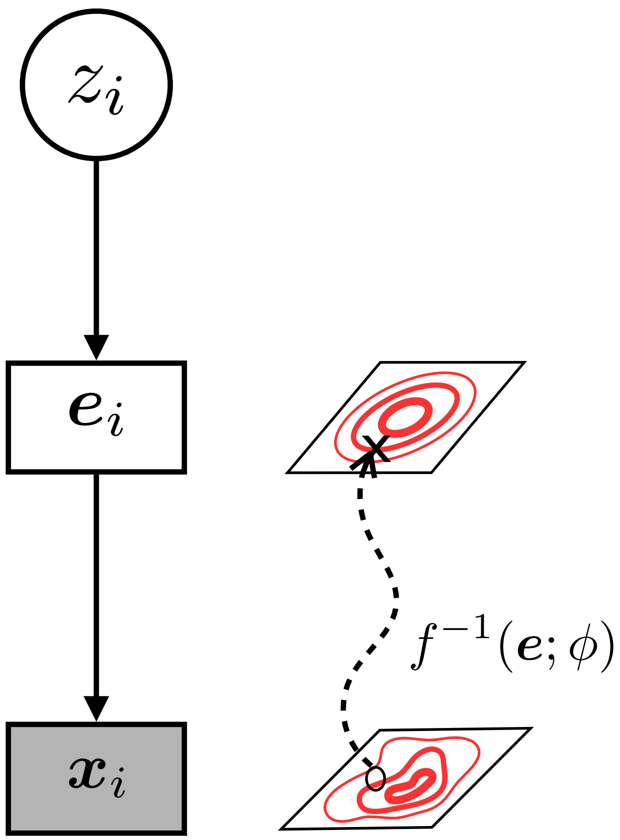
Dependency Parse Induction

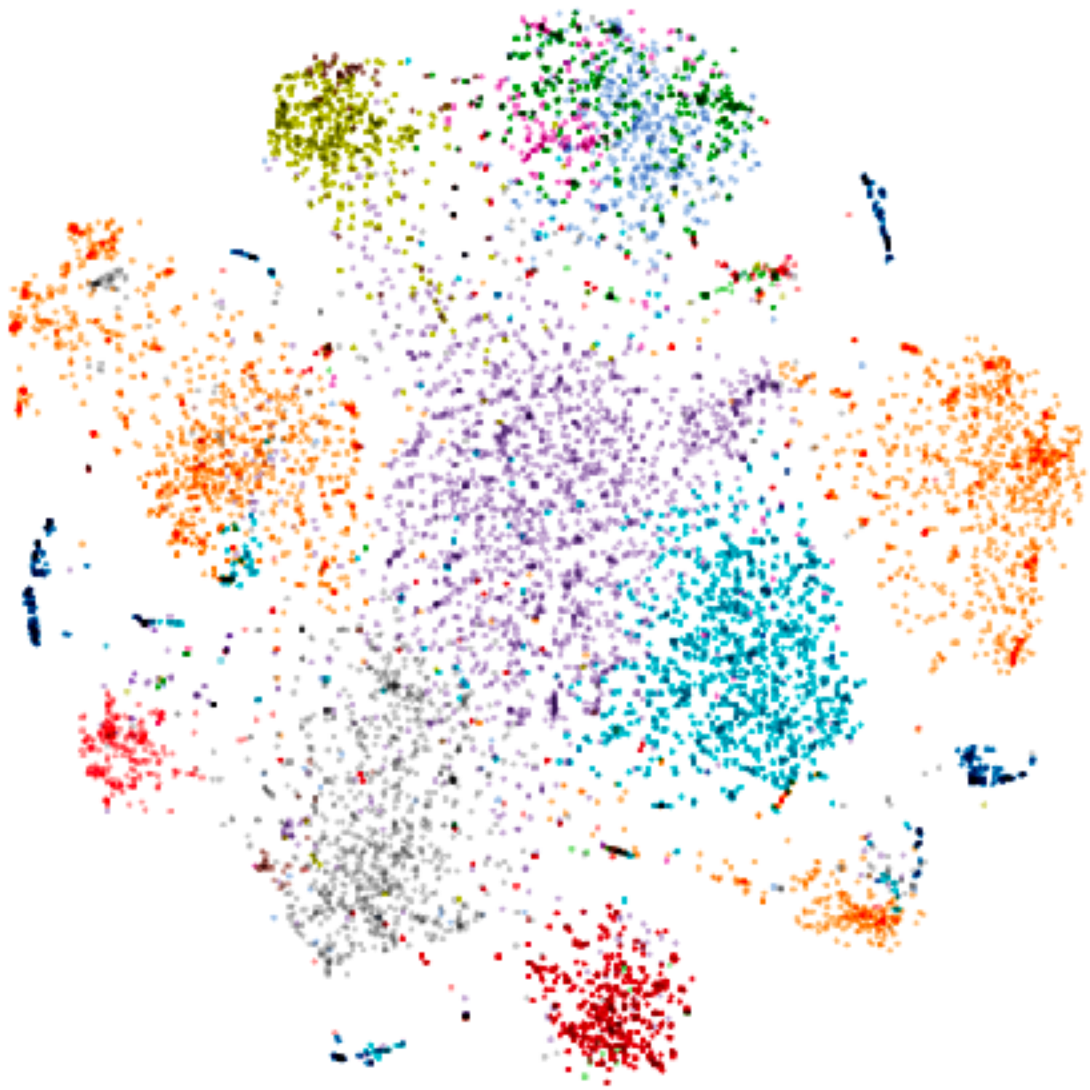


Original Embedding Space

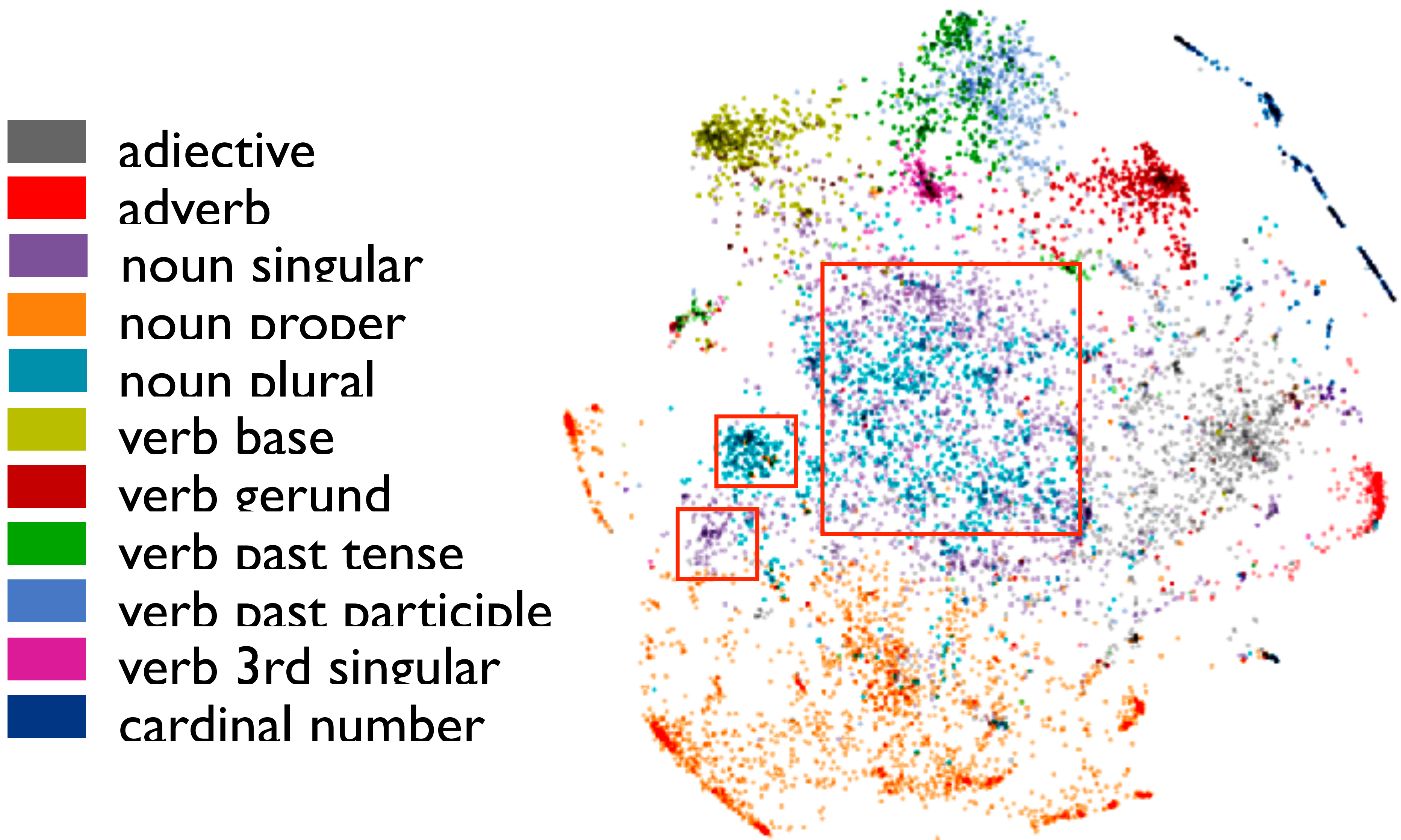


Projected Embedding Space w/ Markov Prior

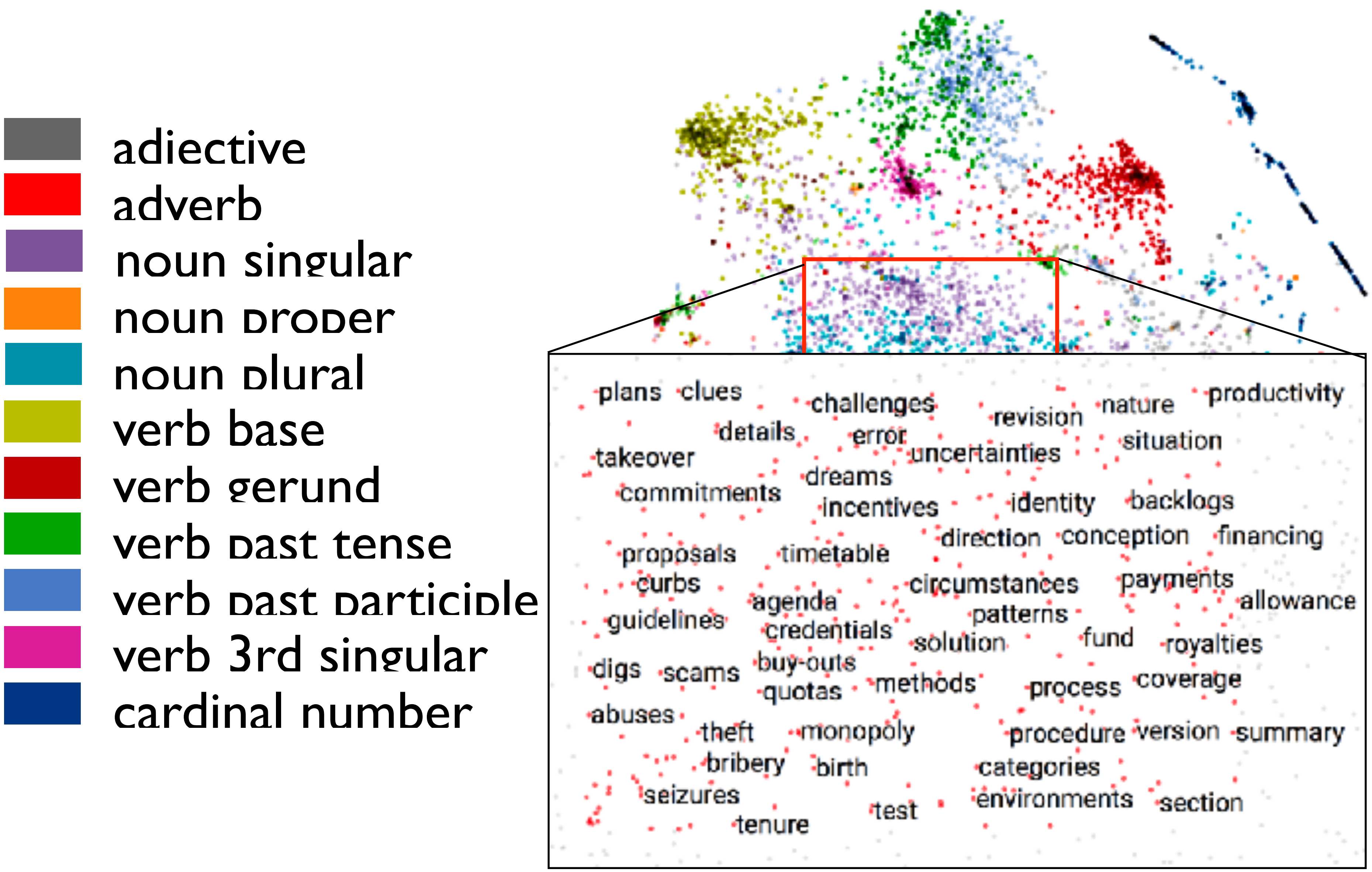




Projected Embedding Space w/ DMV Prior

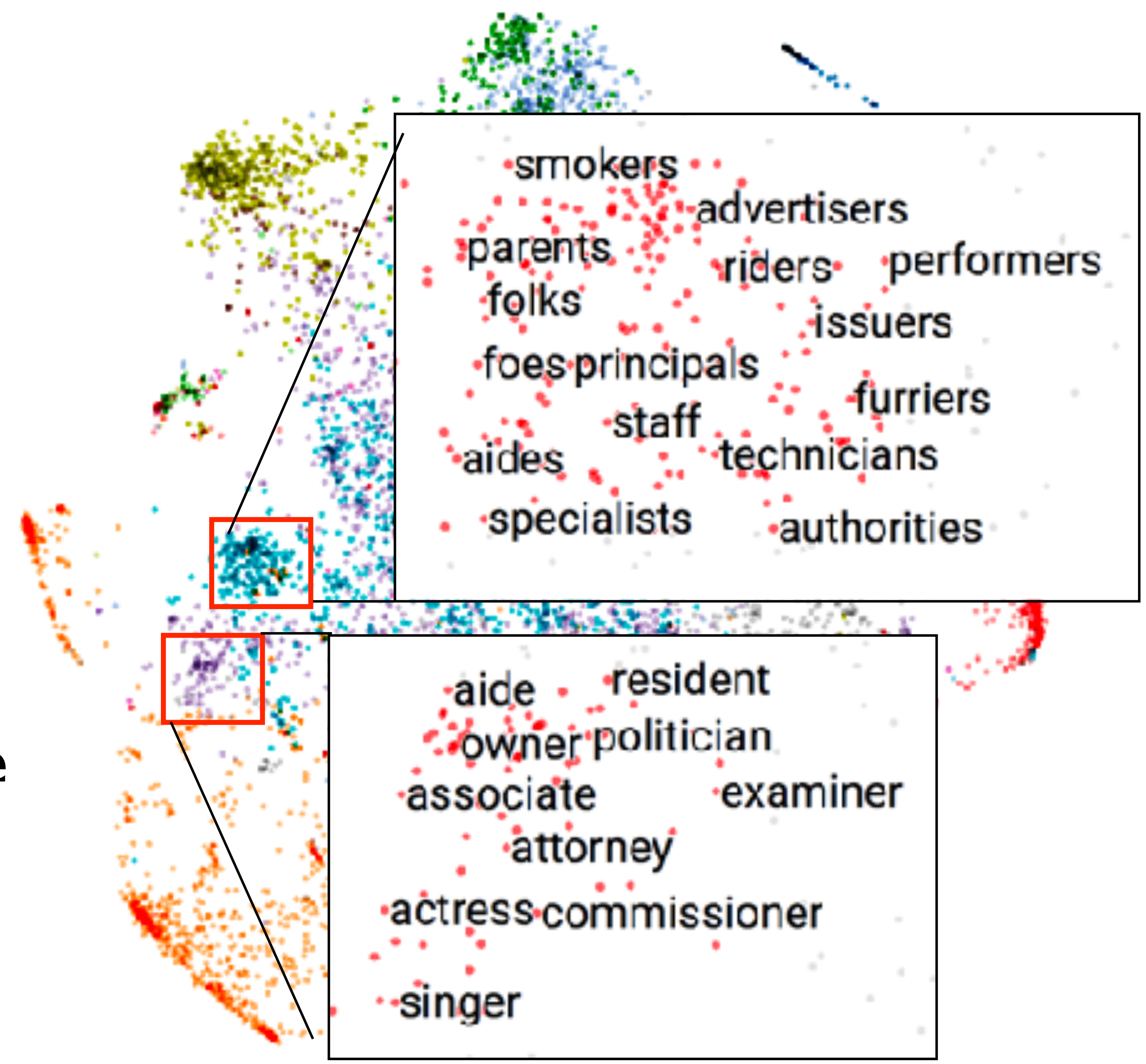


Projected Embedding Space w/ DMV Prior



Projected Embedding Space w/ DMV Prior

- adjective
- adverb
- noun singular
- noun proper
- noun plural
- verb base
- verb gerund
- verb past tense
- verb past participle
- verb 3rd singular
- cardinal number



FlowSeq: Non-Autoregressive Conditional Sequence Generation with Generative

Xuezhe Ma*, Chunting Zhou*, Xian Li, Graham Neubig, Eduard Hovy

Background

- Autoregressive Sequence Generation

$$P_\theta(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} P_\theta(y_t|y_{<t}, \mathbf{x}).$$

  - Left-to-right factorization is not optimal
  - Generation is not easily parallelizable on GPUs
- Non-autoregressive Sequence Generation

$$P_\theta(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} P_\theta(y_t|\mathbf{x}).$$

Motivation

- Latent Variable Model

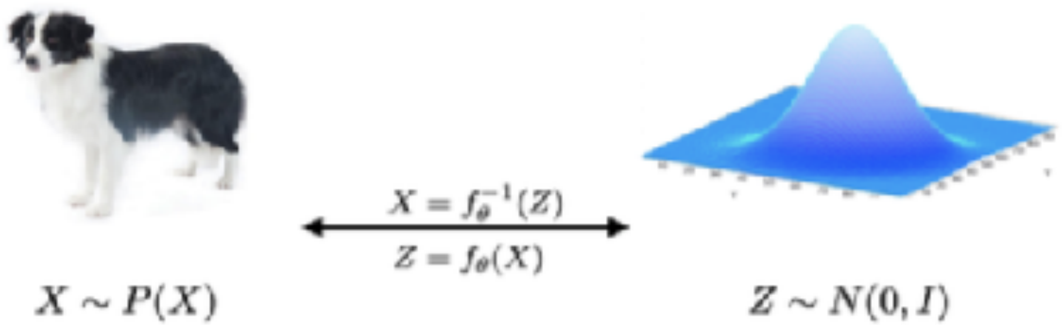
$$P_\theta(\mathbf{y}|\mathbf{x}) = \int_{\mathbf{z}} P_\theta(\mathbf{y}|\mathbf{z},\mathbf{x}) p_\theta(\mathbf{z}|\mathbf{x}) d\mathbf{z},$$

  - $p_\theta(\mathbf{z}|\mathbf{x})$ is the prior distribution over latent z
  - $P_\theta(\mathbf{y}|\mathbf{z},\mathbf{x})$ is the generative distribution (a.k.a decoder)
  - non-autoregressive generation

$$P_\theta(\mathbf{y}|\mathbf{z},\mathbf{x}) = \prod_{t=1}^{T} P_\theta(y_t|\mathbf{z},\mathbf{x}).$$

Reminder: Flow-based Generative Models

- What is Generative Flows:
  - Transform a simple distribution into a complex one through a chain of invertible transformations

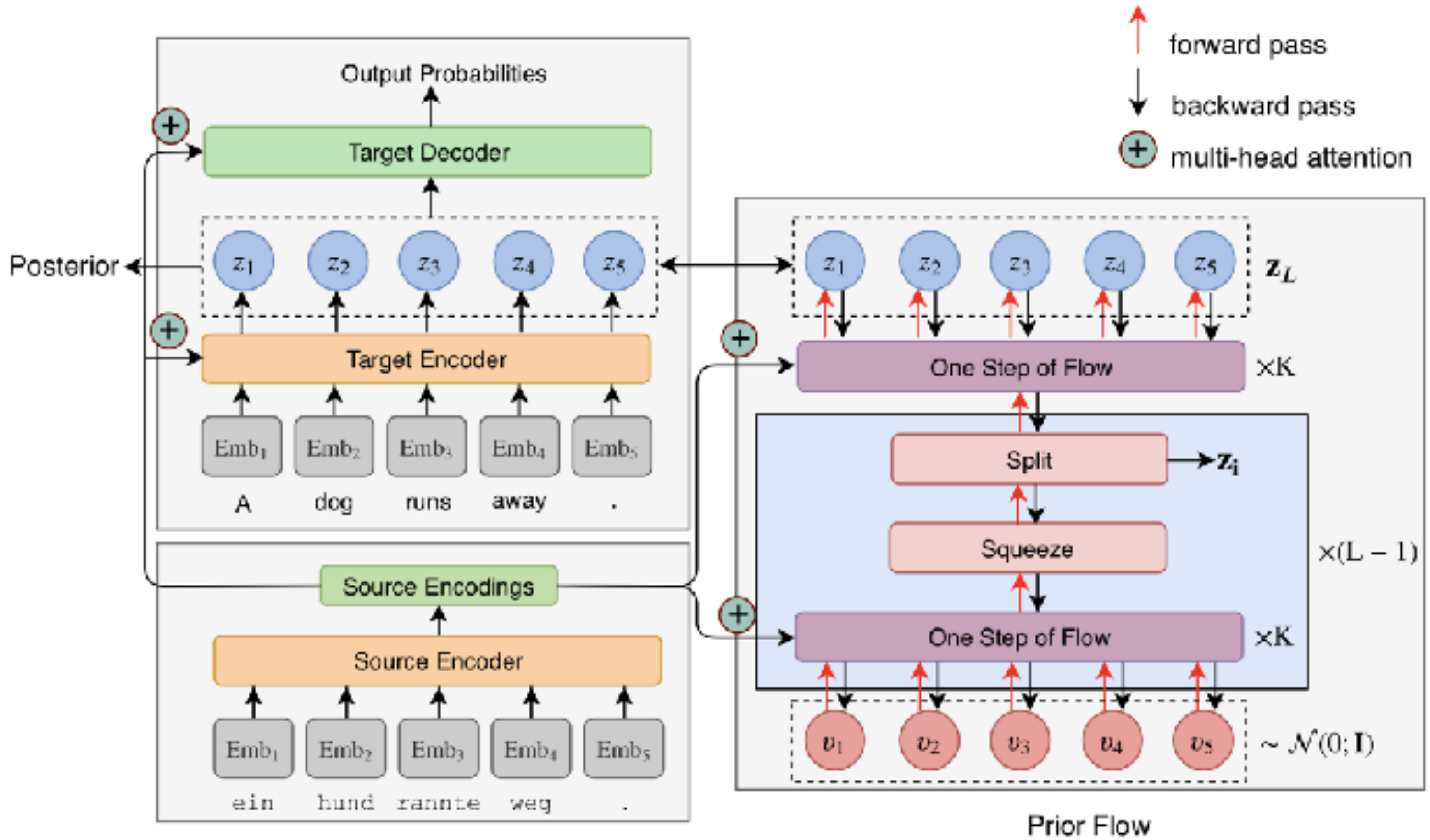


  - Change of variable formula: $$p_\theta(\mathbf{z}) = p_\Upsilon(f_\theta(\mathbf{z})) \left|\det\left(\frac{\partial f_\theta(\mathbf{z})}{\partial \mathbf{z}}\right)\right|.$$
  - Generative Flow: $$\mathbf{z} \underset{g_1}{\overset{f_1}{\longleftrightarrow}} H_1 \underset{g_2}{\overset{f_2}{\longleftrightarrow}} H2 \underset{g_3}{\overset{f_3}{\longleftrightarrow}} \cdots \underset{g_K}{\overset{f_K}{\longleftrightarrow}} \boldsymbol{v},$$

FlowSeq

- Variational Training: FlowSeq optimizes the *evidence lower bound* (ELBO)

$$\log P_\theta(\mathbf{y}|\mathbf{x}) \geq \mathrm{E}_{q_\phi(\mathbf{z}|\mathbf{y},\mathbf{x})}[\log P_\theta(\mathbf{y}|\mathbf{z},\mathbf{x})] - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{y},\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})).$$



FlowSeq Architecture

- Source Encoder
  - Standard Transformer encoder
- Posterior: diagonal Gaussian
  - The latent variables z are represented as a sequence of continuous random variables with the same length as the target sequence y:
  -

$$\mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$$

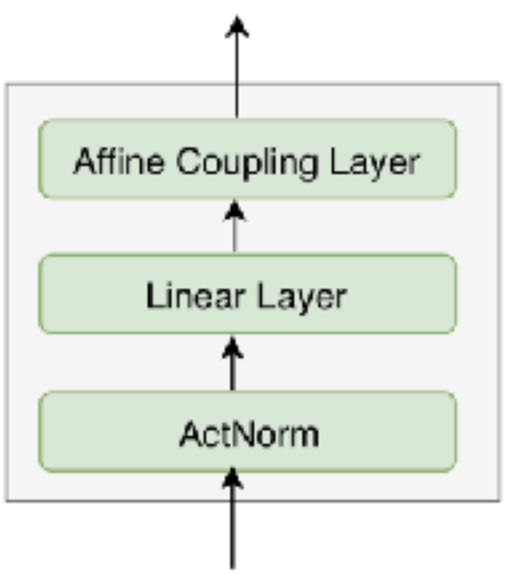
- Decoder: Transformer decoder w/o causal masking

$$q_\phi(\mathbf{z}|\mathbf{y}, \mathbf{x}) = \prod_{t=1}^{T} \mathcal{N}(\mathbf{z}_t | \mu_t(\mathbf{x}, \mathbf{y}), \sigma_t^2(\mathbf{x}, \mathbf{y}))$$

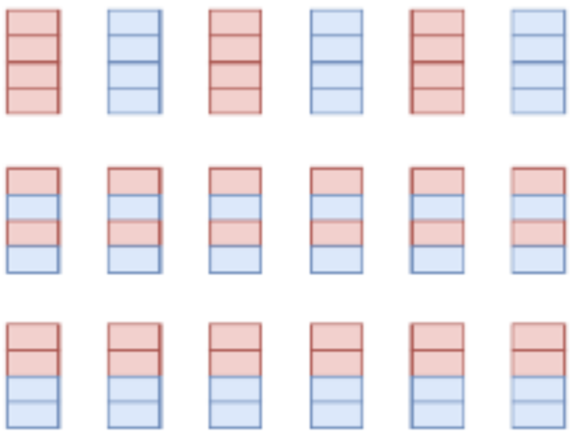
Flow Architecture for Prior

- Actnorm (activation normalization layer): $\mathbf{z}'_t = \mathbf{s} \odot \mathbf{z}_t + \mathbf{b}.$
- Invertible Multi-head Linear Layers: $\mathbf{z}'_t = \mathbf{z}_t \mathbf{W},$ (W: [dz x dz])
- Affine Coupling Layers

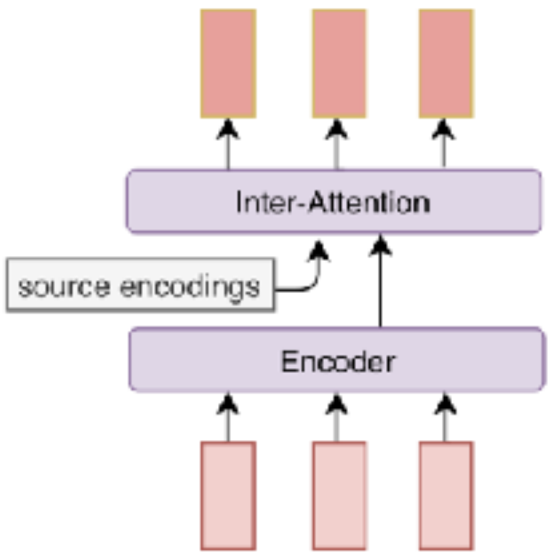
$$
\begin{aligned}
\mathbf{z}_a, \mathbf{z}_b &= \mathrm{split}(\mathbf{z}) \\
\mathbf{z}'_a &= \mathbf{z}_a \\
\mathbf{z}'_b &= \mathrm{s}(\mathbf{z}_a, \mathbf{x}) \odot \mathbf{z}_b + \mathrm{b}(\mathbf{z}_a, \mathbf{x}) \\
\mathbf{z}' &= \mathrm{concat}(\mathbf{z}'_a, \mathbf{z}'_b),
\end{aligned}
$$



(a) One step of flow.



(b) Coupling layer splits.



(c) NN function on the split of the coupling layer.

Decoding Process

- **Argmax** Decoding
$$\begin{aligned} \mathbf{z}^* &= \operatorname*{argmax}_{\mathbf{z}\in\mathcal{Z}} p_\theta(\mathbf{z}|\mathbf{x}) \\ \mathbf{y}^* &= \operatorname*{argmax}_{\mathbf{y}} P_\theta(\mathbf{y}|\mathbf{z}^*, \mathbf{x}) \end{aligned}$$
- **Noisy Parallel Decoding (NPD)**: rescoring multiple samples by a pre-trained auto-regressive model.
- **Importance Weighted Decoding (IWD)**: rescoring multiple candidates by importance samples.

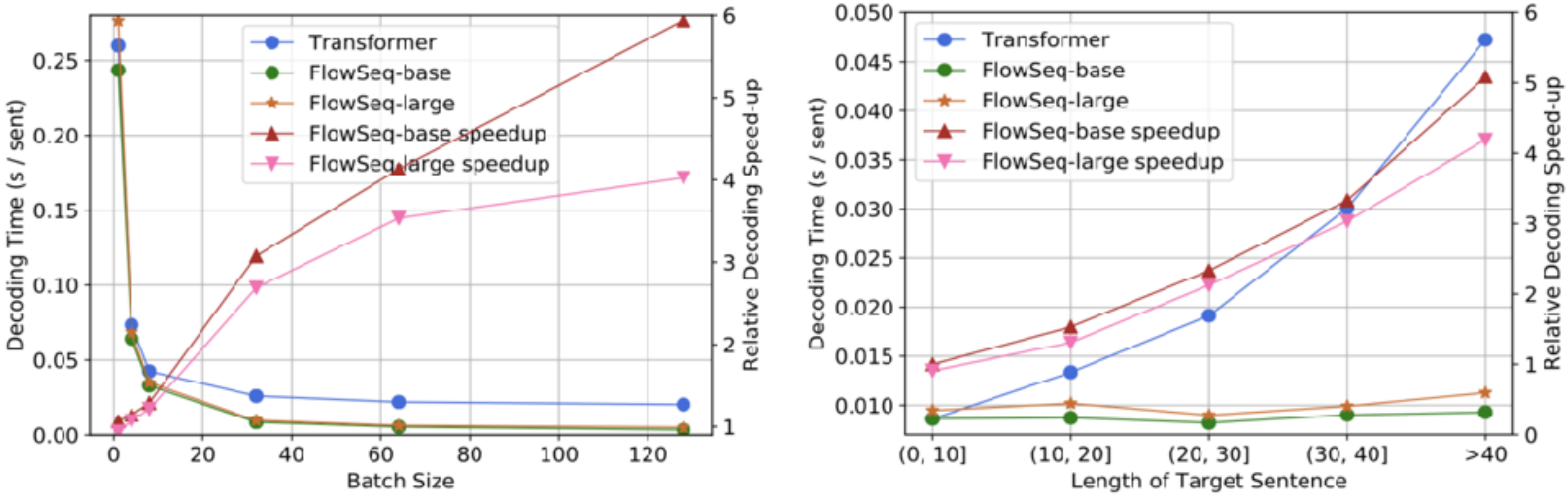
$$\begin{aligned} \mathbf{z}_i &\sim p_\theta(\mathbf{z}|\mathbf{x}), \forall i = 1, \ldots, N \\ \hat{\mathbf{y}}_i &= \operatorname*{argmax}_{\mathbf{y}} P_\theta(\mathbf{y}|\mathbf{z}_i, \mathbf{x}) \\ \mathbf{z}_i^{(k)} &\sim q_\phi(\mathbf{z}|\hat{\mathbf{y}}_i, \mathbf{x}), \forall k = 1, \ldots, K \\ P(\hat{\mathbf{y}}_i|\mathbf{x}) &\approx \frac{1}{K}\sum_{k=1}^{K} \frac{P_\theta(\hat{\mathbf{y}}_i|\mathbf{z}_i^{(k)}, \mathbf{x}) p_\theta(\mathbf{z}_i^{(k)}|\mathbf{x})}{q_\phi(\mathbf{z}_i^{(k)}|\hat{\mathbf{y}}_i, \mathbf{x})} \end{aligned}$$

Experiments

- MT benchmark datasets:
  - IWSLT 2014 EN-DE
  - WMT14 EN-DE, DE-EN
  - WMT16 EN-RO, RO-EN

Models	WMT2014 EN-DE	WMT2014 DE-EN	WMT2016 EN-RO	WMT2016 RO-EN
Autoregressive Methods				
Transformer-base	27.30	–	–	–
Our Implementation	27.16	31.44	32.92	33.09
Raw Data				
CMLM (refinement 1)	10.88	–	20.24	–
CMLM (refinement 4)	22.06	–	30.89	–
CMLM (refinement 10)	**24.65**	–	**32.53**	–
FlowSeq-large (Argmax)	20.85	25.40	29.86	30.69
FlowSeq-large (IWD $n = 15$)	22.94	27.16	31.08	32.03
FlowSeq-large (NPD $n = 15$)	23.14	27.71	31.97	32.46
FlowSeq-large (NPD $n = 30$)	23.64	**28.29**	32.35	**32.91**
Knowledge Distillation				
NAT w/ FT (Argmax)	17.69	21.47	27.29	29.06
NAT w/ FT (NPD $n = 10$)	18.66	22.42	29.02	31.44
NAT-IR (refinement 1)	13.91	16.77	24.45	25.73
NAT-IR (refinement 10)	21.61	25.48	29.32	30.19
NAT-REG (NPD $n = 9$)	24.61	28.90	–	–
CMLM (refinement 1)	18.12	22.26	23.65	22.78
CMLM (refinement 4)	26.08	30.11	31.78	31.76
CMLM (refinement 10)	**26.92**	**30.86**	**32.42**	**33.06**
FlowSeq-large (Argmax)	23.72	28.39	29.73	30.72
FlowSeq-large (IWD $n = 15$)	24.70	29.44	31.02	31.97
FlowSeq-large (NPD $n = 15$)	25.03	30.48	31.89	32.43
FlowSeq-large (NPD $n = 30$)	25.31	30.68	32.20	32.84

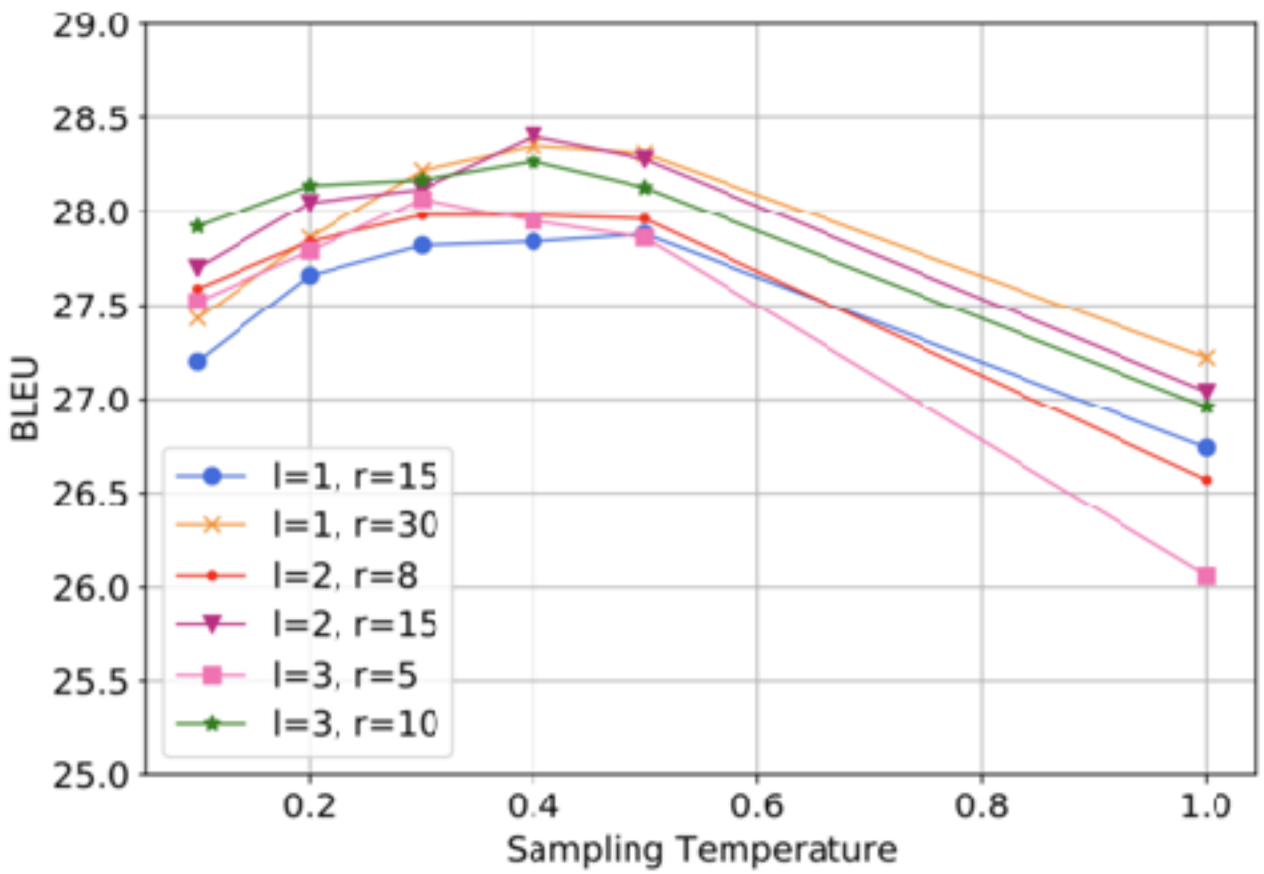
Decoding Speed



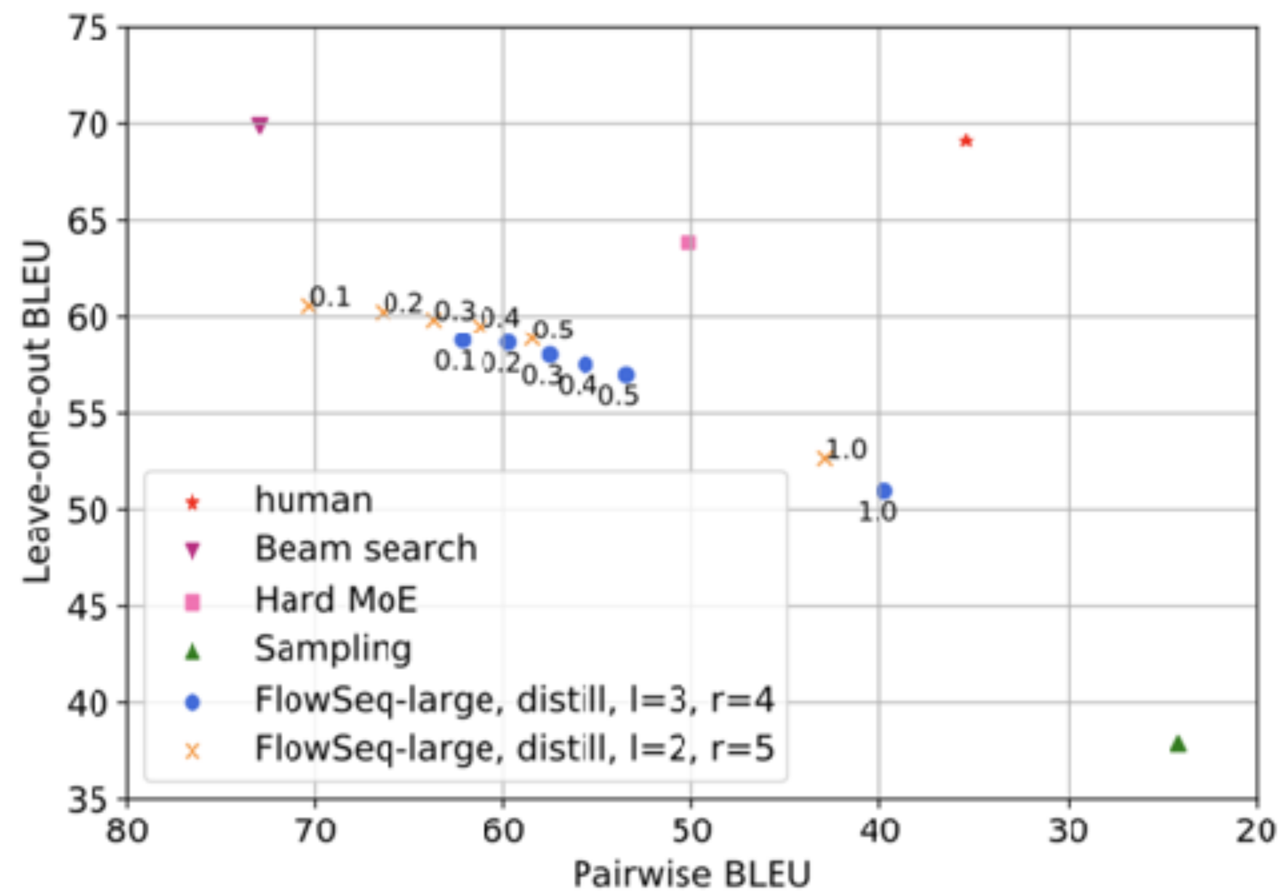
(a) batch size

(b) target length

Analysis of Translations



(c) Rescoring



(d) Diversity

An Example

Source	Es kann nicht erklären, weshalb die National Security Agency Daten ber das Privatleben von Amerikanern sammelt und warum Whistleblower bestraft werden, die staatliches Fehlverhalten offenlegen.
Ground Truth	And, most recently, it cannot excuse the failure to design a simple website more than three years since the Affordable Care Act was signed into law.
Sample 1	And recently, it cannot apologise for the inability to design a simple website in the more than three years since the adoption of Affordable Care Act.
Sample 2	And recently, it cannot excuse the inability to design a simple website in more than three years since the adoption of Affordable Care Act.
Sample 3	Recently, it cannot excuse the inability to design a simple website in more than three years since the Affordable Care Act has passed.

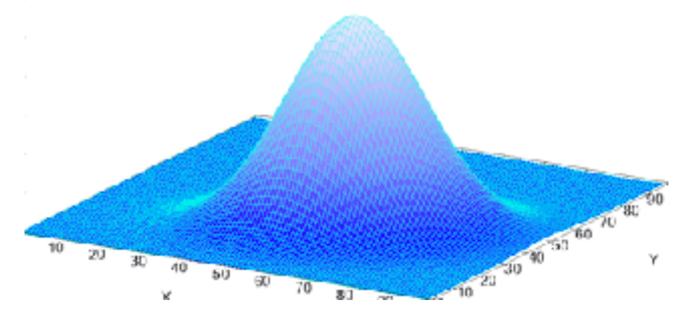
Conclusion

Conclusion

- Normalizing flows for unsupervised learning

$$X = f_\theta^{-1}(Z)$$

$$Z = f_\theta(X)$$

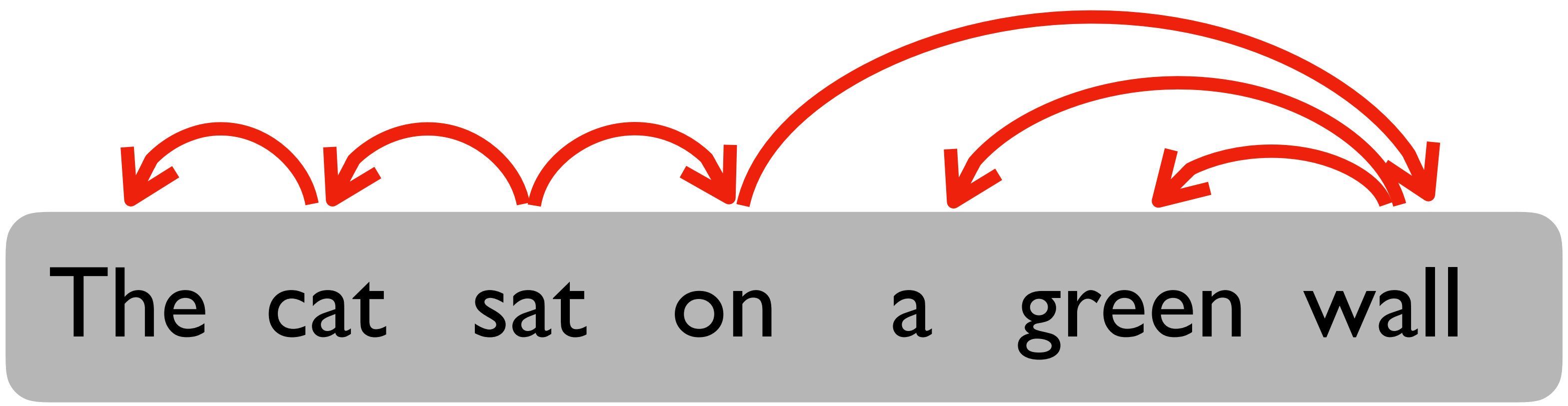


- Learning of bilingual lexicons
- Learning of latent structure
- Learning of sequence-to-sequence models

Thank You! Questions?

DeMa-BWE

https://github.com/violet-zct/DeMa-BWE



https://github.com/jxhe/struct-learning-with-flow

https://github.com/XuezheMax/flowseq

Language Technologies Institute