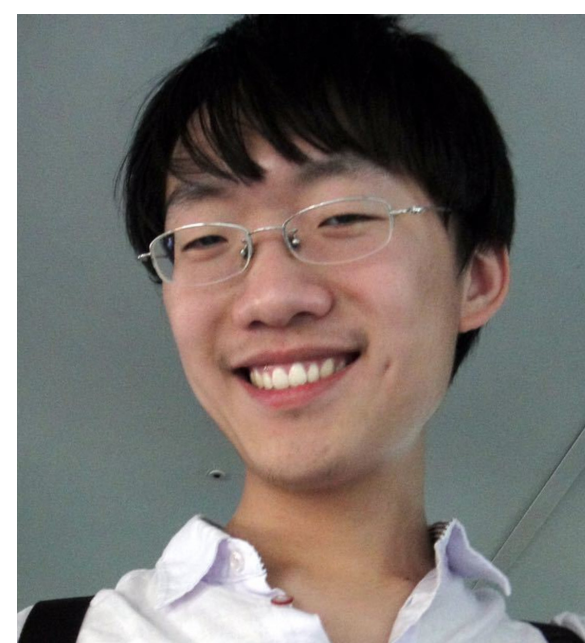
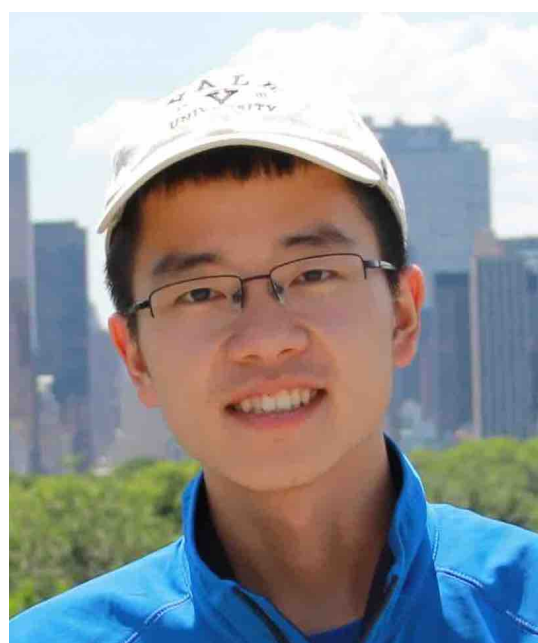




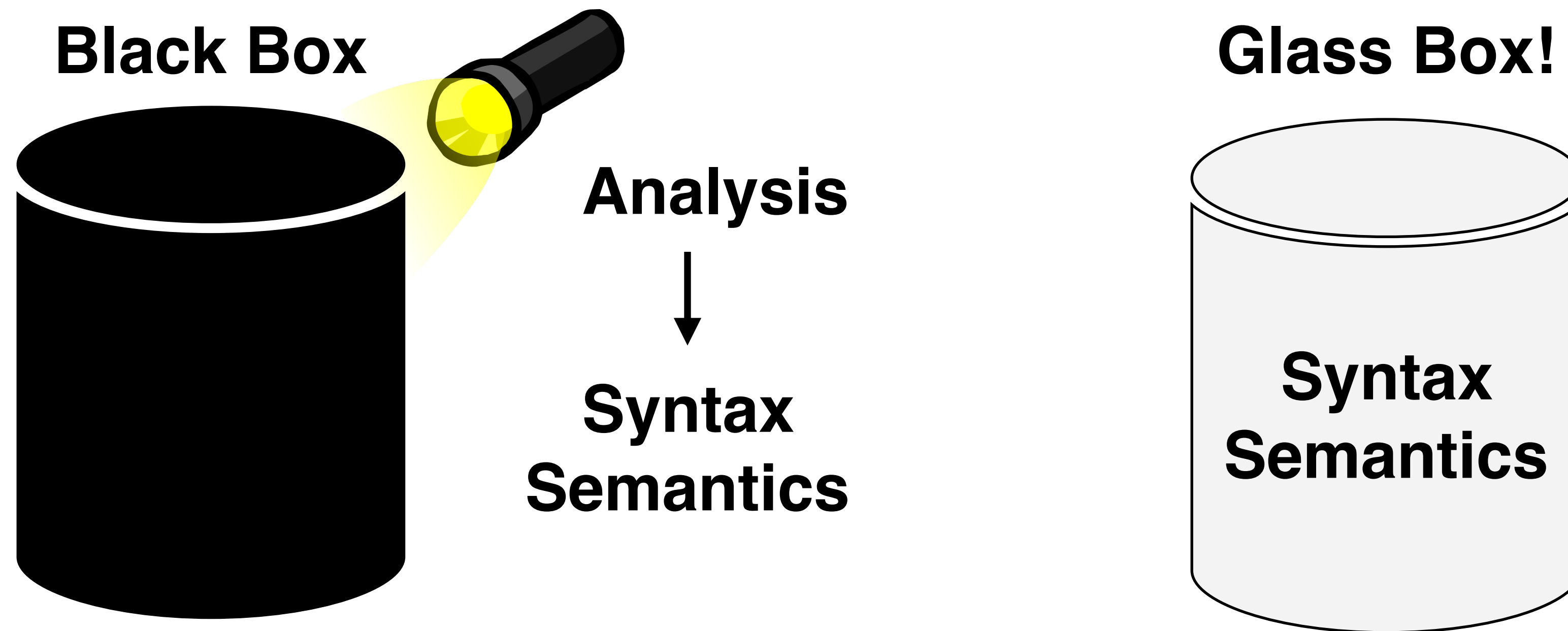
Learning with Latent Linguistic Structure

Graham Neubig
@ BlackBoxNLP 11/1/2018

with: Junxian He Pengcheng Yin Chunting Zhou Taylor Berg-Kirkpatrick



How to Achieve Interpretability in Neural Nets?



Research Problems

- Fundamentally highly interpretable models (e.g. discrete HMMs) are not sufficiently powerful
- How can we harness the power of neural networks, with underlying interpretable representations?
- How can we learn them on unlabeled data?



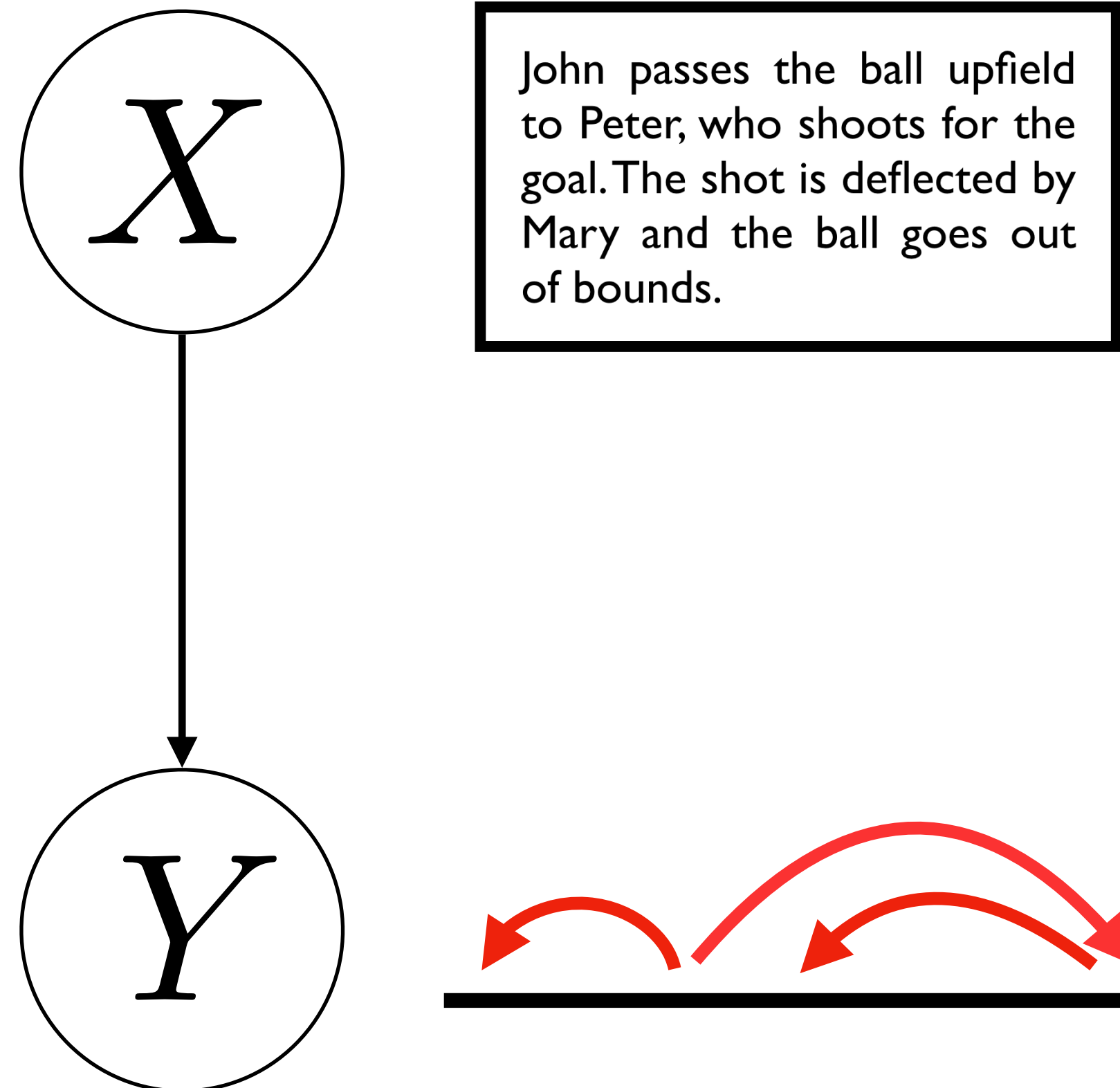
e.g. Syntactic Analysis

Dependency:

Parts-of-speech:



Supervised Approach





Supervised Approach

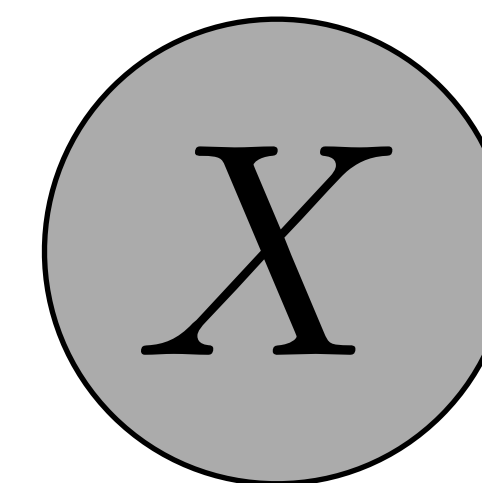
Supervised Learning

 X
 θ
 Y

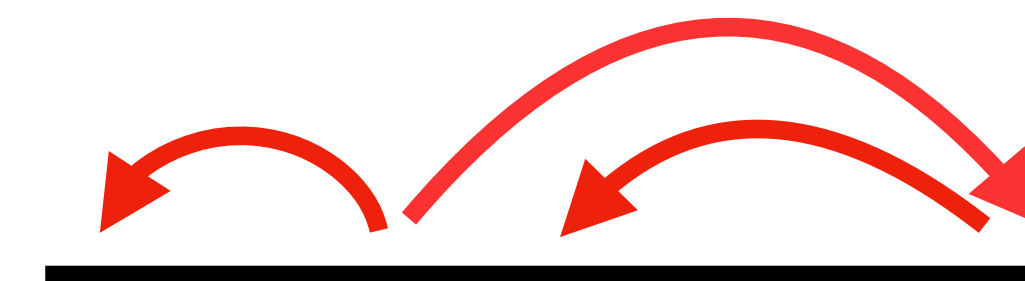
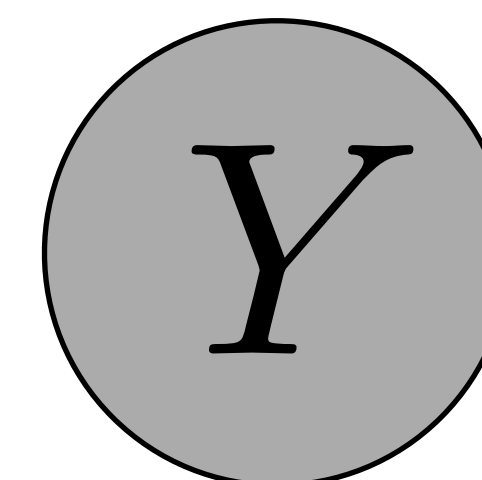
John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.

John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.

John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.

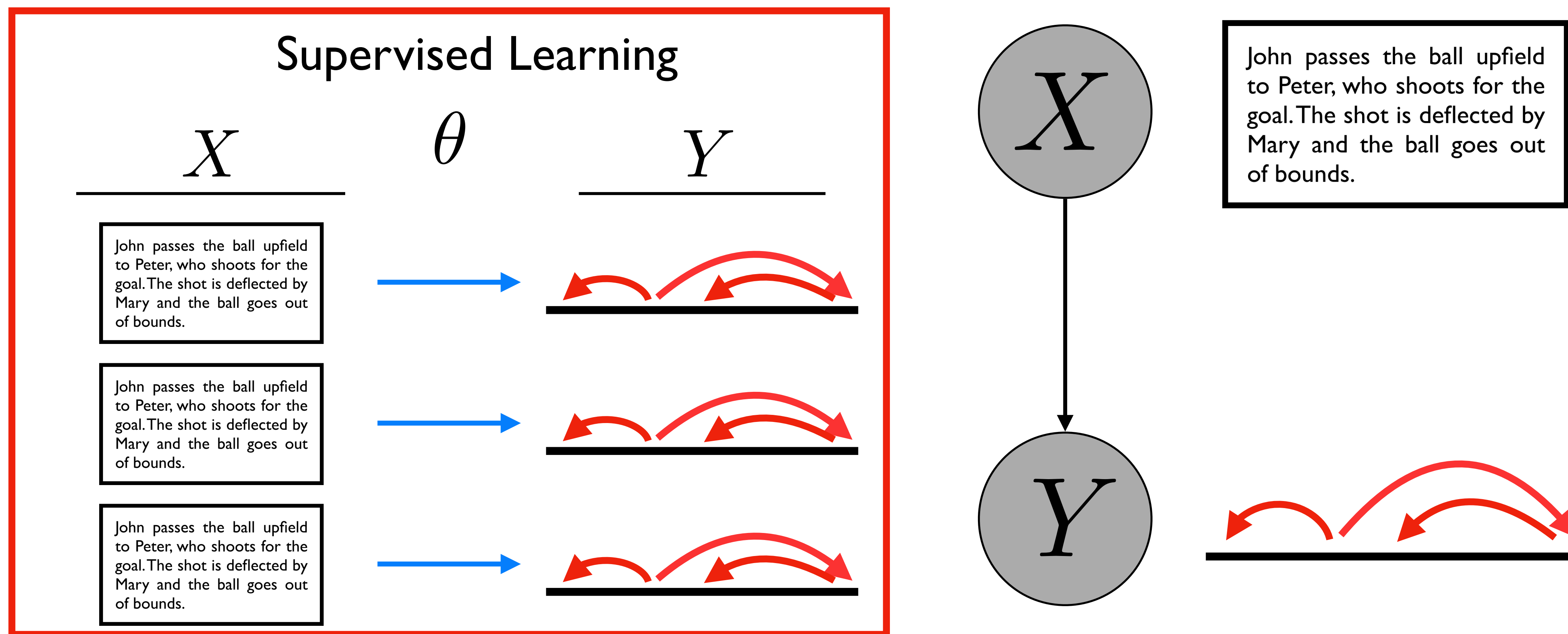


John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.

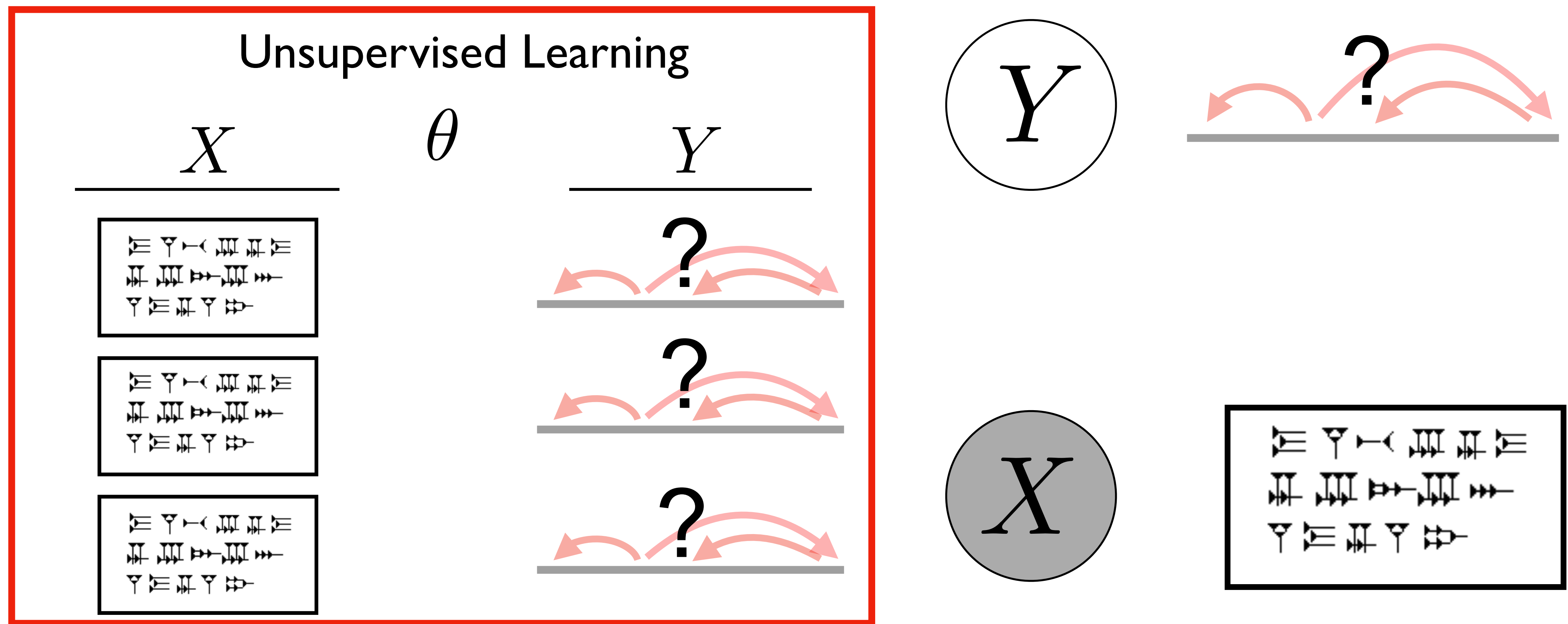




Supervised Approach

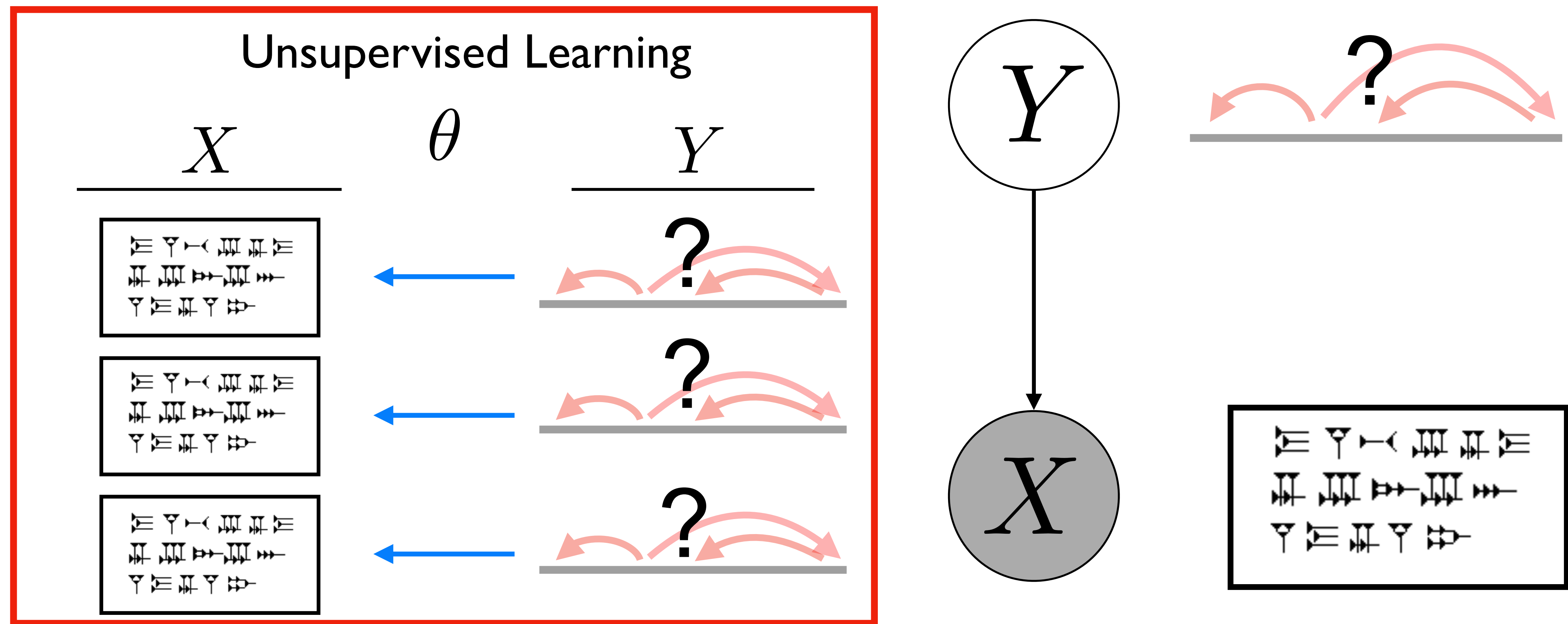


Latent Variable Approach





Latent Variable Approach





Carnegie Mellon University
School of Computer Science



Language
Technologies
Institute

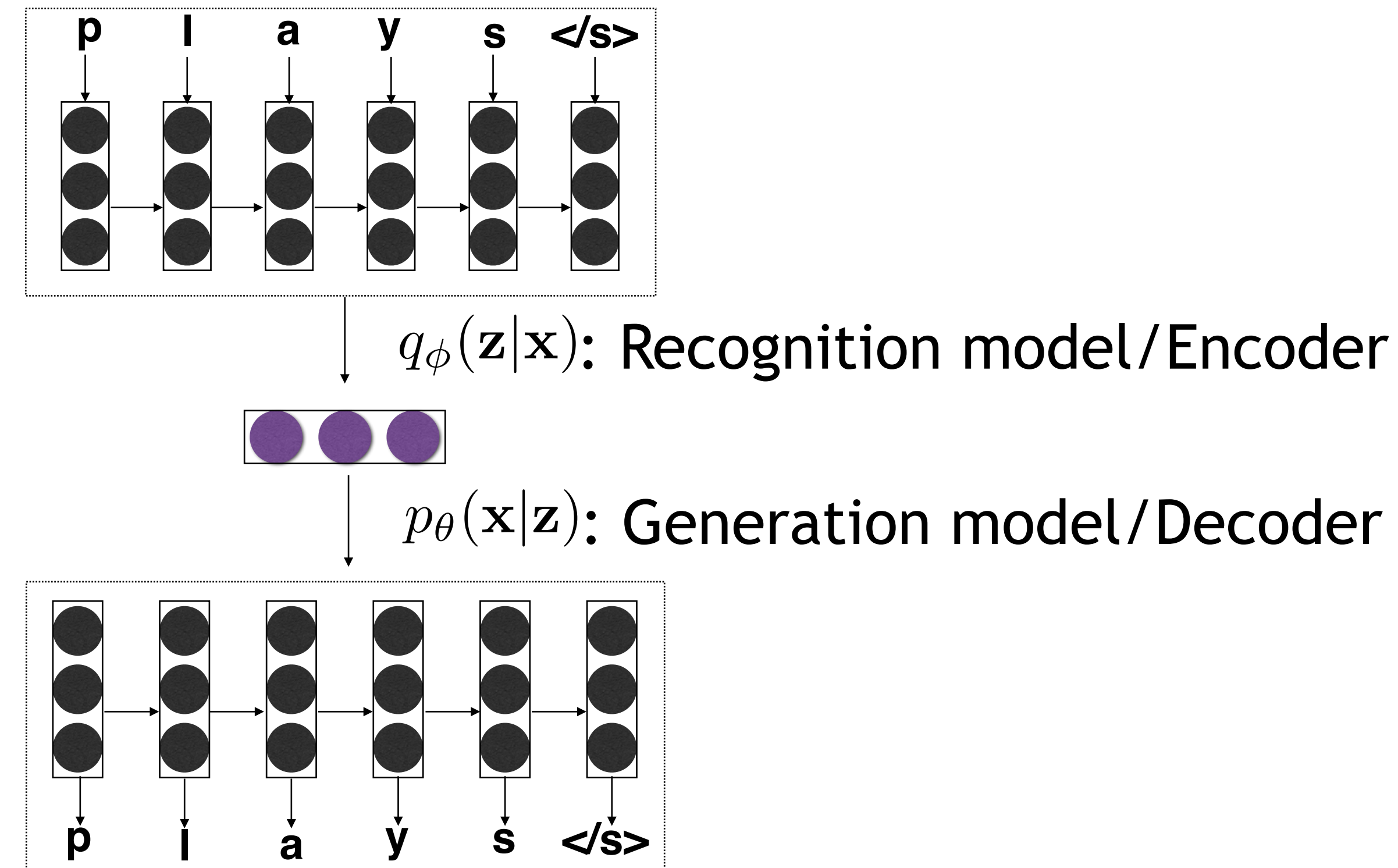
Multi-space Variational Encoder-Decoders

Chunting Zhou and Graham Neubig
(ACL 2017)

Features of Words

- **Syntax:**
 - What syntactic features does the word have?
 - Closed-class, generally enumerable for a specific language.
- **Meaning/Symbol:**
 - What is the meaning of the word, how is it spelled/pronounced?
 - Open-class, complicated regularities and relationships.
 - Can we create a model that elegantly models both?

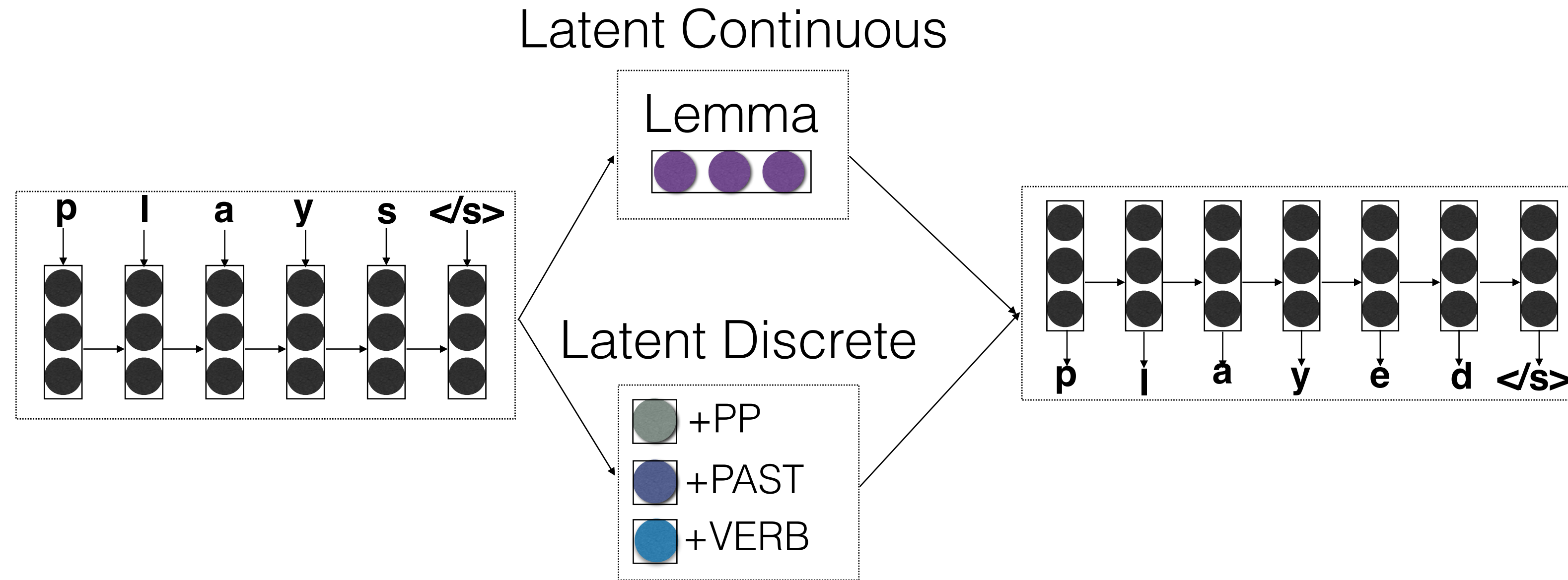
Background: Variational Auto-encoder (Kingma et al., 2014, Bowman et al., 2016)



Maximize the Variational lower bound:

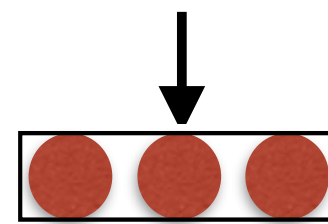
$$\log p_{\theta}(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z})] - \text{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}) || p(\mathbf{z}))$$

Proposed Model: Multi-space Variational Encoder-Decoders



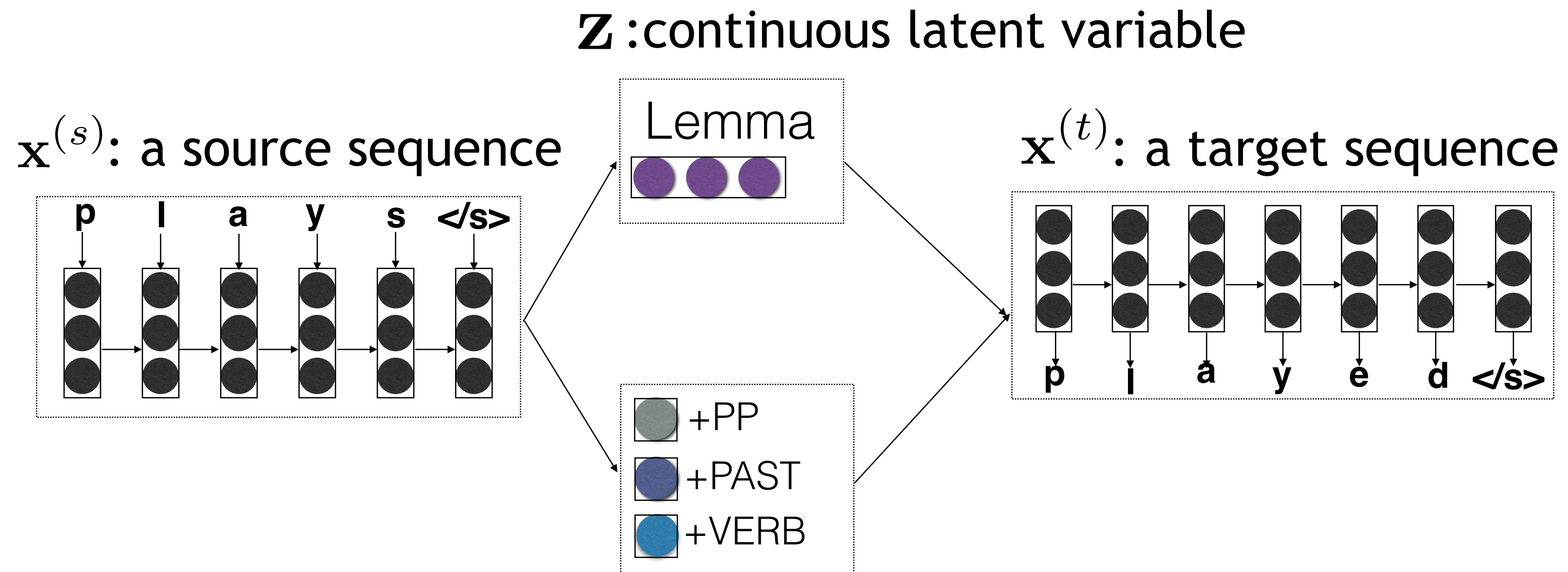
- Modeling complicated higher-level structure (e.g. meaning or symbol of the word): **incorporation of continuous latent variables**
- Modeling closed-class and interpretable features (e.g. syntax): **incorporation of discrete latent variables**

plays, played, playing



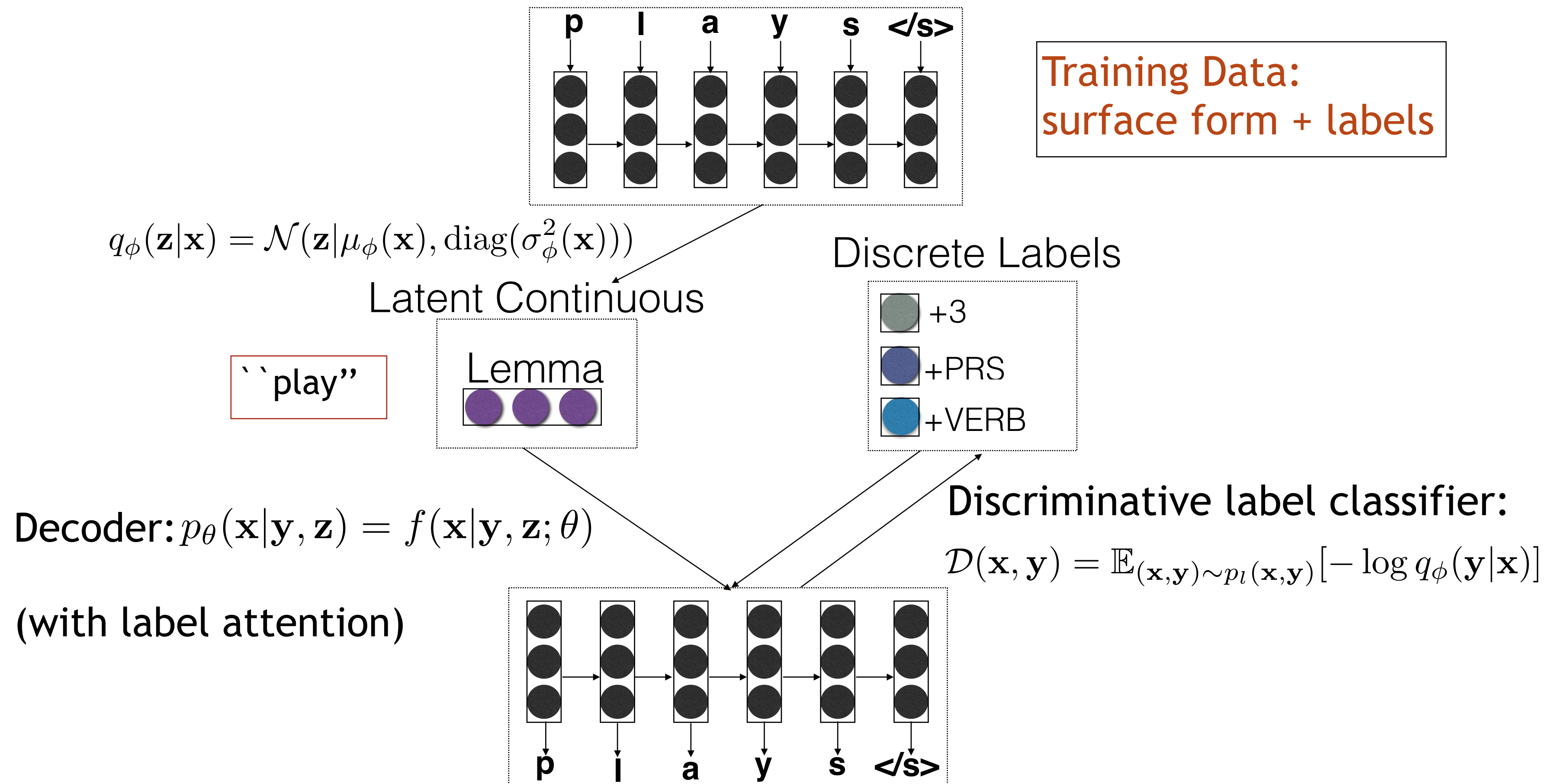
- How can we learn in a un- or semi-supervised way?

Variable Definitions



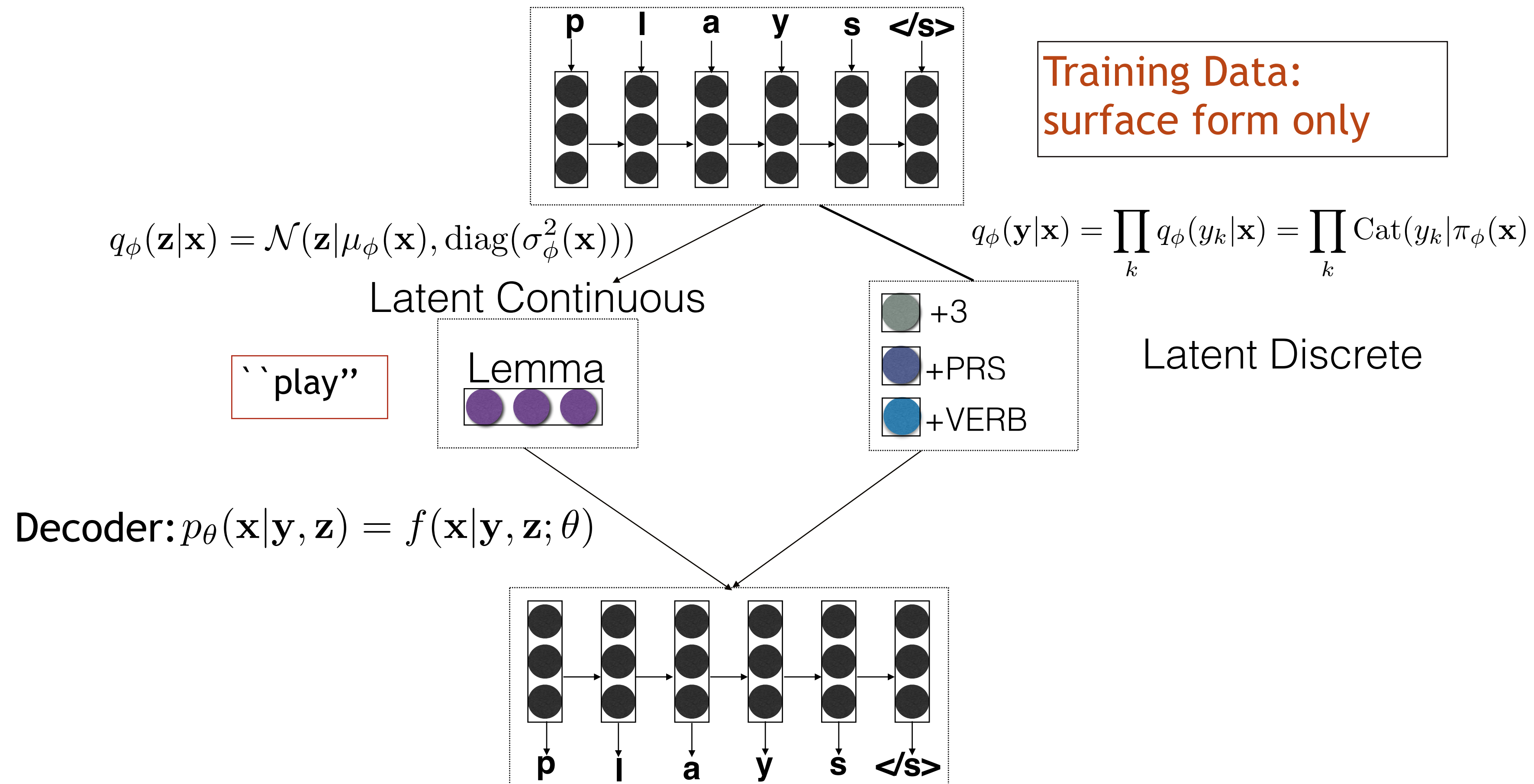
$\mathbf{y}^{(t)} = [y_1^{(t)}, y_2^{(t)}, \dots, y_K^{(t)}]$: discrete labels for each target sequence

Supervised Learning: Labeled Multi-space Variational Autoencoders



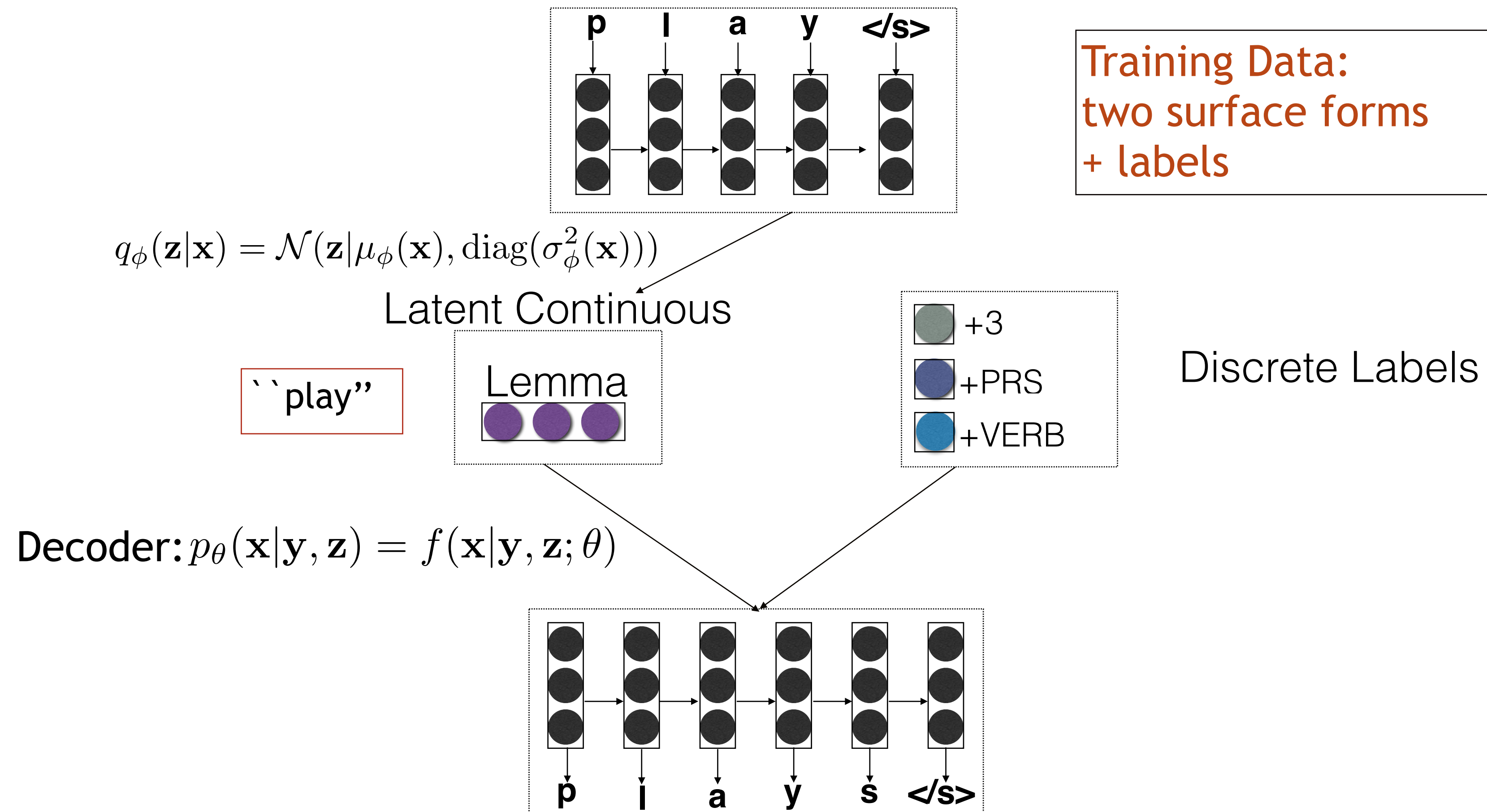
Maximize : $\mathcal{U}(\mathbf{x}) = \text{Variational Lower Bound of } \log p(\mathbf{x}, \mathbf{y})$

Unsupervised Learning: Unlabeled Multi-space Variational Auto-encoders



Maximize : $\mathcal{U}(\mathbf{x}) = \text{Variational Lower Bound of } \log p(\mathbf{x}, \mathbf{y})$

Labeled Sequence-to-sequence Training: Multi-space Variational Encoder-Decoders



Maximize : $\mathcal{L}_l(\mathbf{x}^{(t)}, \mathbf{y}^{(t)} | \mathbf{x}^{(s)}) = \text{Variational Lower Bound of } \log p(\mathbf{x}^{(t)}, \mathbf{y}^{(t)} | \mathbf{x}^{(s)})$

Learning MSVED

- **Learning Continuous Latent Variables:**

Reparameterization trick (Kingma et al., 2014):

$$\epsilon \sim \mathcal{N}(0, 1), \quad \mathbf{z} = \mu_{\phi}(x) + \sigma_{\phi}(x) \circ \epsilon$$

- **Learning Discrete Latent Variables:**

Gumbel-Softmax (Maddison et al., 2017)

$$\hat{y}_{ij} = \frac{\exp((\log(\pi_{ij}) + g_{ij})/\tau)}{\sum_{k=1}^{N_i} \exp((\log(\pi_{ik}) + g_{ik})/\tau)}$$

- **Training tricks (Bowman et al. 2016):**

- KL-divergence Annealing
- Input dropout in the decoder

Experimental Setup

Task: Morphology re-inflection

Dataset: SIGMORPHON 2016 task 3

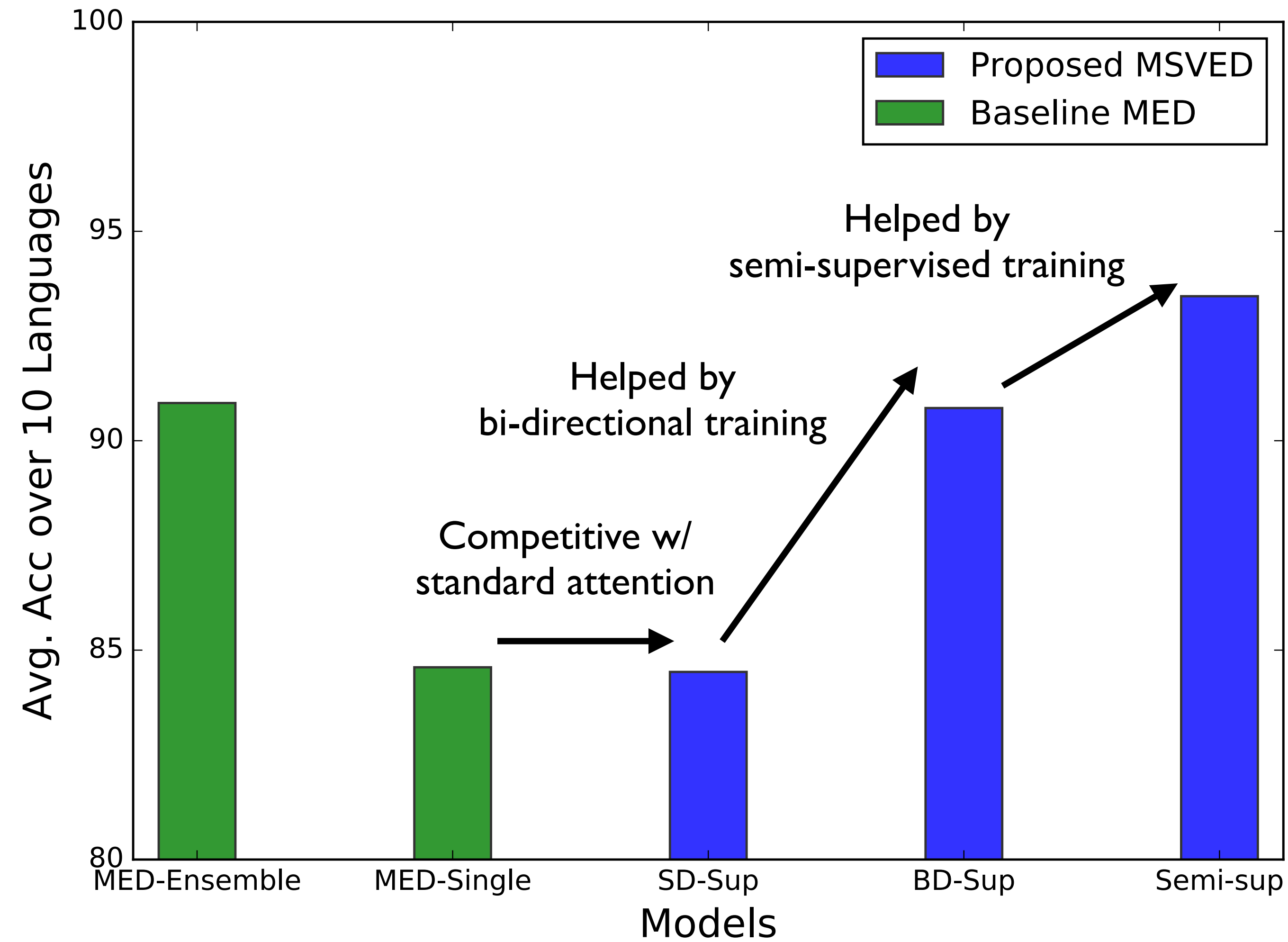
source word: communicated

target word: communicates

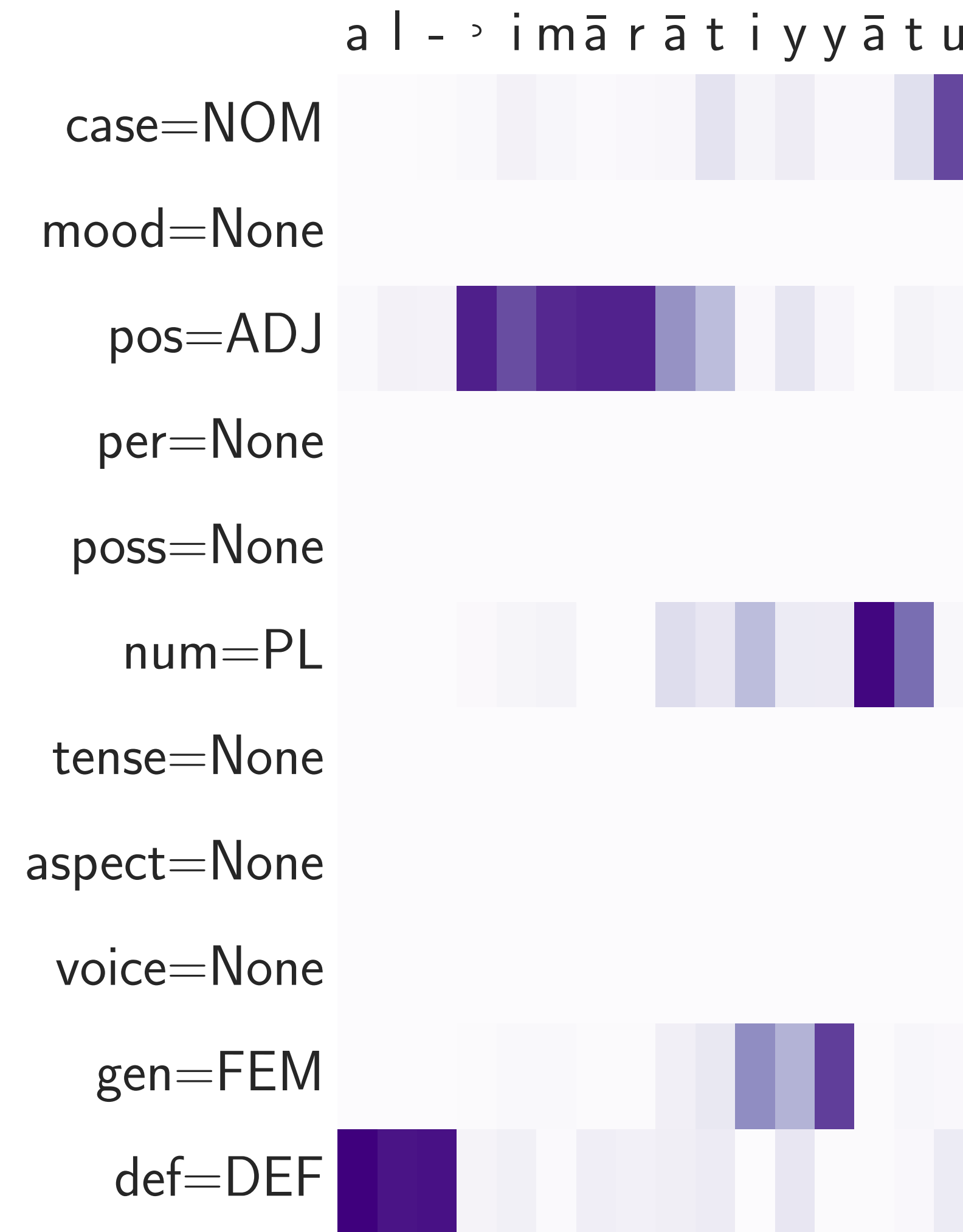
target labels: V;3;SG;PRS

Language: Turkish, Arabic, Maltese, Finnish,
Spanish, German, Hungarian, Navajo,
Georgian, Russian

Results and Analysis

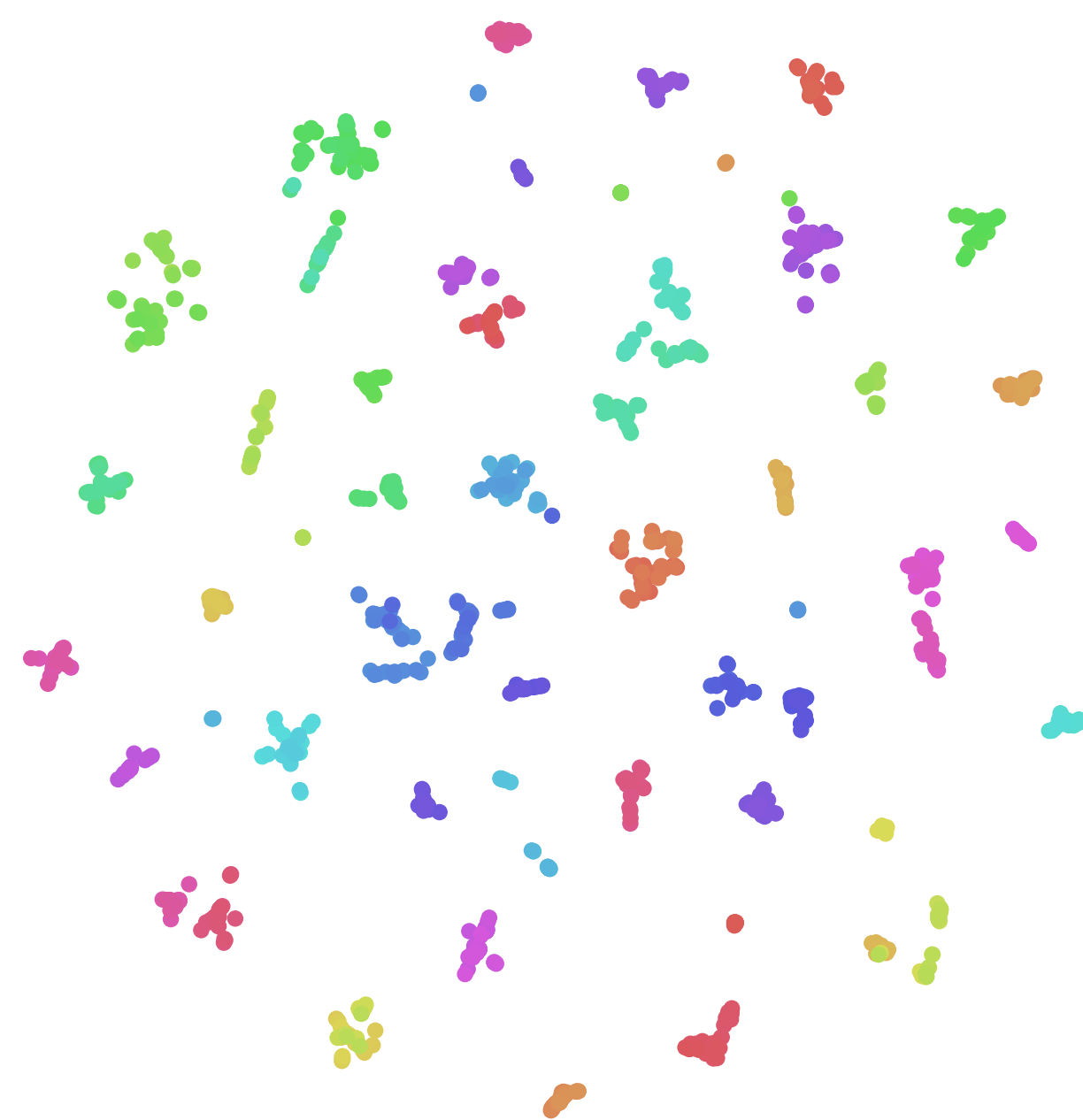


Analysis on Tag Attention

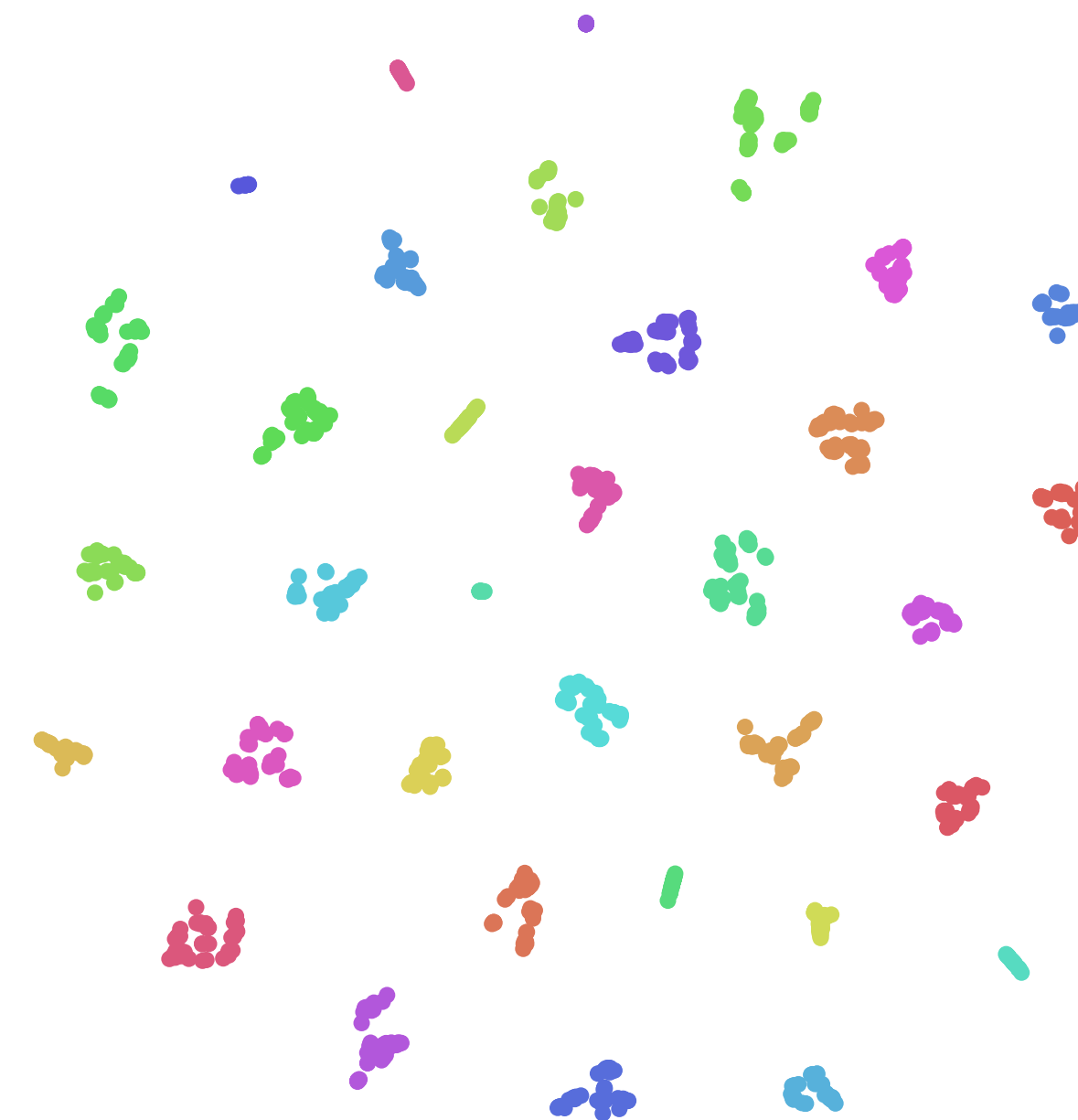


Visualization of Latent Continuous Variables

- Clusters colored by actual lemma:



Turkish



Maltese



Carnegie Mellon University

School of Computer Science

StructVAE: Tree-structured Latent Variable Models for Semi-supervised Semantic Parsing

Pengcheng Yin, Chunting Zhou, Junxian He, Graham Neubig
(ACL 2018)

Language
Technologies
Institute

NEULAB

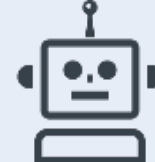
What About More Complicated Structure?

Semantic Parsing: Transducing natural language utterances (e.g., queries) into machine-executable formal meaning representations (e.g., logical form, source code)

Domain-Specific Meaning Representations



 *Show me flights from Pittsburgh to Washington*

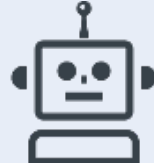
 `lambda $0 e (and (flight $0)
 (from $0
 san_Francisco:ci)
 (to $0 washington:ci))`

lambda-calculus logical form

General-Purpose Programming Languages



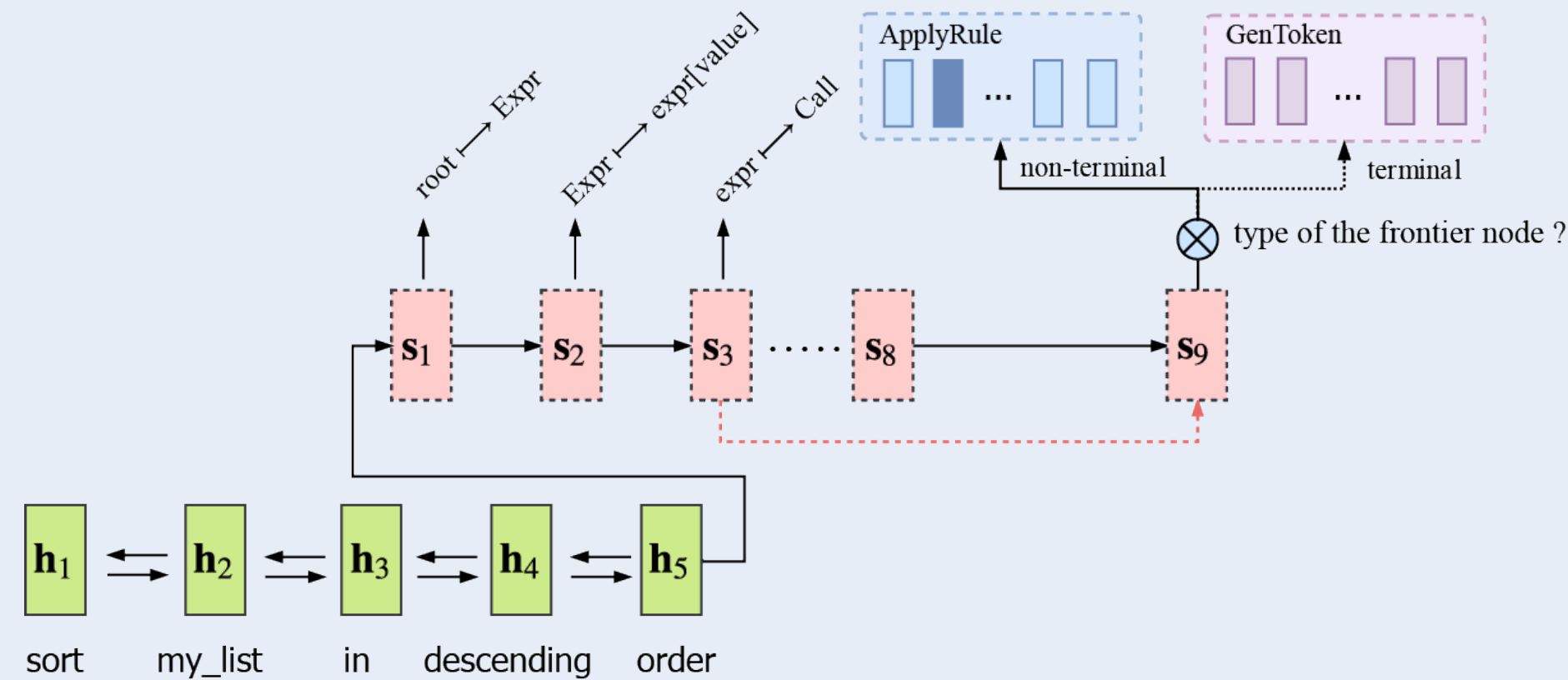
 *Sort my_list in descending order*

 `sorted(my_list,
 reverse=True)`

Python

Research Issue

Neural Models are Data Hungry



Purely supervised neural semantic parsing models require large amounts of training data



Data Collection is Costly

Copy the content of file 'file.txt' to file 'file2.txt'

```
shutil.copy('file.txt', 'file2.txt')
```

Get a list of words 'words' of a file 'myfile'

```
words = open('myfile').read().split()
```

Check if all elements in list 'mylist' are the same

```
len(set(mylist)) == 1
```

Collecting parallel training

data costs

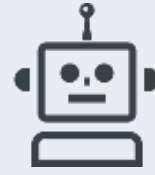


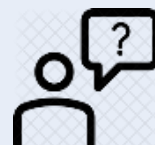
and

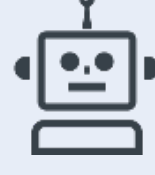


Semi-supervised Semantic Parsing

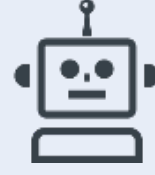
Limited Amount of Labeled Data

 *Sort my_list in descending order*
 `sorted(my_list, reverse=True)`

 *Copy the content of file 'file.txt' to file 'file2.txt'*

 `shutil.copy('file.txt',
 'file2.txt')`

 *Check if all elements in list 'mylist' are the same*

 `len(set(mylist)) == 1`

+

Extra Unlabeled Utterances

 *Get a list of words 'words' of a file 'myfile'*

 *Convert a list of integers into a single integer*

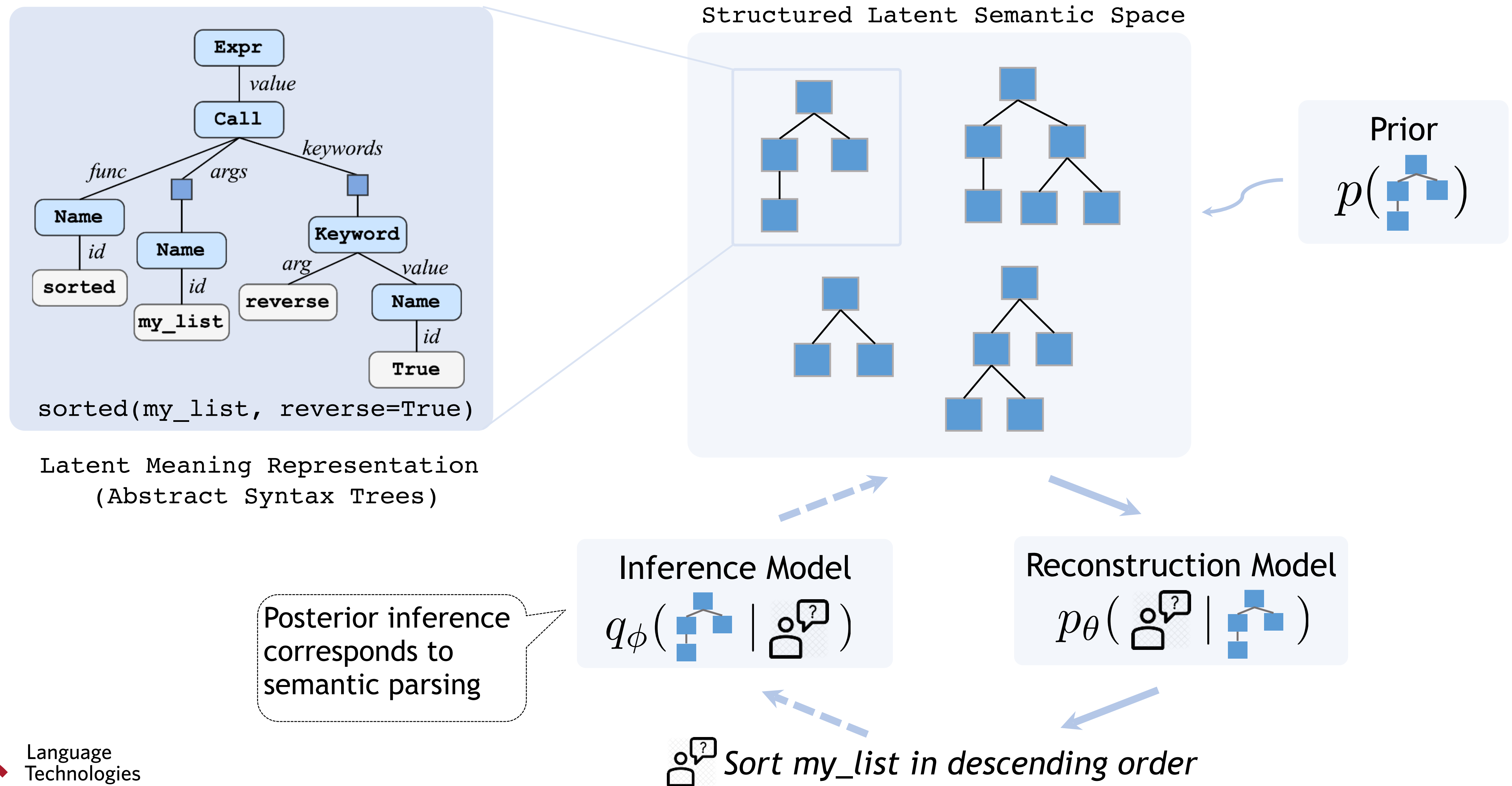
 *Format a datetime object 'when' to extract date only*

 *Swap values in a tuple/list in list 'mylist'*

 *BeautifulSoup search string 'Elsie' inside tag 'a'*

 *Convert string to lowercase*

Meaning Representations as Tree-structured Latent Variables



Semi-supervised Learning with StructVAE



Supervised Objective

$$\sum \log q_{\phi}(\text{tree icon} \mid \text{person icon with ?})$$

($\text{person icon with ?}$, tree icon) Labeled Data

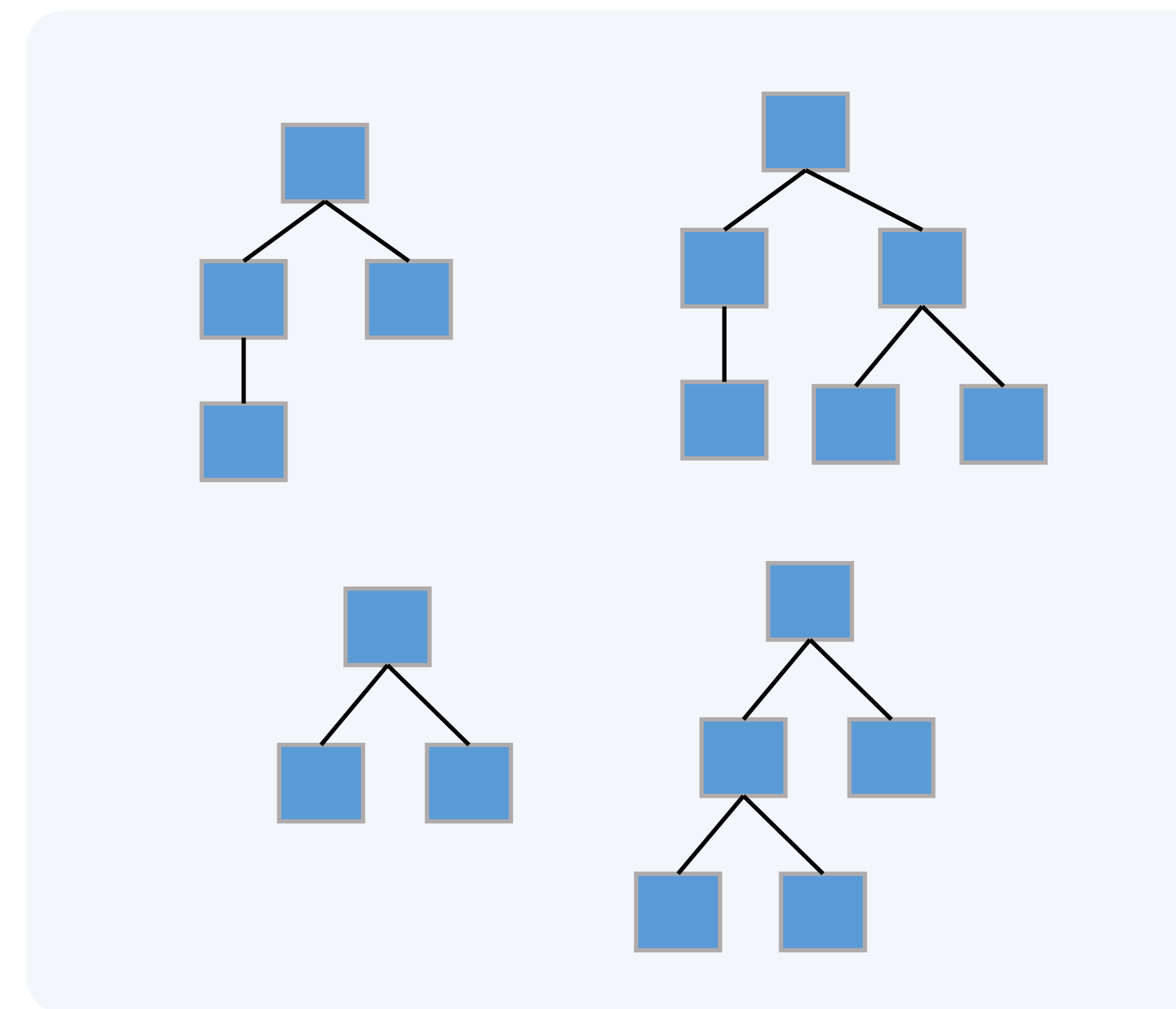
+

Unsupervised Objective

$$\sum \log p(\text{tree icon})$$

$\text{person icon with ?}$ Unlabeled Data

Structured Latent Semantic Space



Prior

$$p(\text{tree icon})$$

Inference Model

$$q_{\phi}(\text{tree icon} \mid \text{person icon with ?})$$

Reconstruction Model

$$p_{\theta}(\text{person icon with ?} \mid \text{tree icon})$$

$\text{person icon with ?}$ *Sort my_list in descending order*

$$p(\text{person icon with ?}) = \int p(\text{person icon with ?} \mid \text{tree icon}) p(\text{tree icon})$$

StructVAE: VAEs with Tree-structured Latent Variables

Inference Model

$$q_{\phi}(\text{tree} \mid \text{input_tree})$$

Neural semantic parser

Reconstruction Model

$$p_{\psi}(\text{output_tree} \mid \text{input_tree})$$

Neural sequence-to-sequence model

Prior

$$p(\text{tree})$$

Neural Language Model

(use linearized trees as inputs)

Unsupervised Objective

$$\sum_{\text{input_tree}} \log p(\text{tree})$$

Unlabeled Data

Variational approximation of the marginal likelihood

$$\log p(\text{input_tree}) \geq \sum_{\text{tree}' \sim q_{\phi}(\text{tree} \mid \text{input_tree})} \log p_{\theta}(\text{input_tree} \mid \text{tree}')$$

$$-\text{KL_Divergence} \left[q_{\phi}(\text{tree} \mid \text{input_tree}) \parallel p(\text{tree}) \right]$$

[Miao and Blunsom, 2016]

How does extra unlabeled data help learning?

Supervised Objective

$$\sum_{(\text{Labeled Data})} \log q_{\phi}(\text{tree} \mid \text{input})$$

$$\nabla = \sum_{\text{Training Examples}} \frac{\partial \log q_{\phi}(\text{tree} \mid \text{input})}{\partial \phi}$$

How does extra unlabeled data help learning?

Unsupervised Objective

$$\sum_{\text{Unlabeled Data}} \log p(\text{tree})$$

$$\nabla \propto \sum_{\text{Sampled tree}} \text{signal} \times \frac{\partial q_{\phi}(\text{tree}' | \text{person with ?})}{\partial \phi}$$

The learning signal



$\approx$

Prior

$$p(\text{tree})$$

+

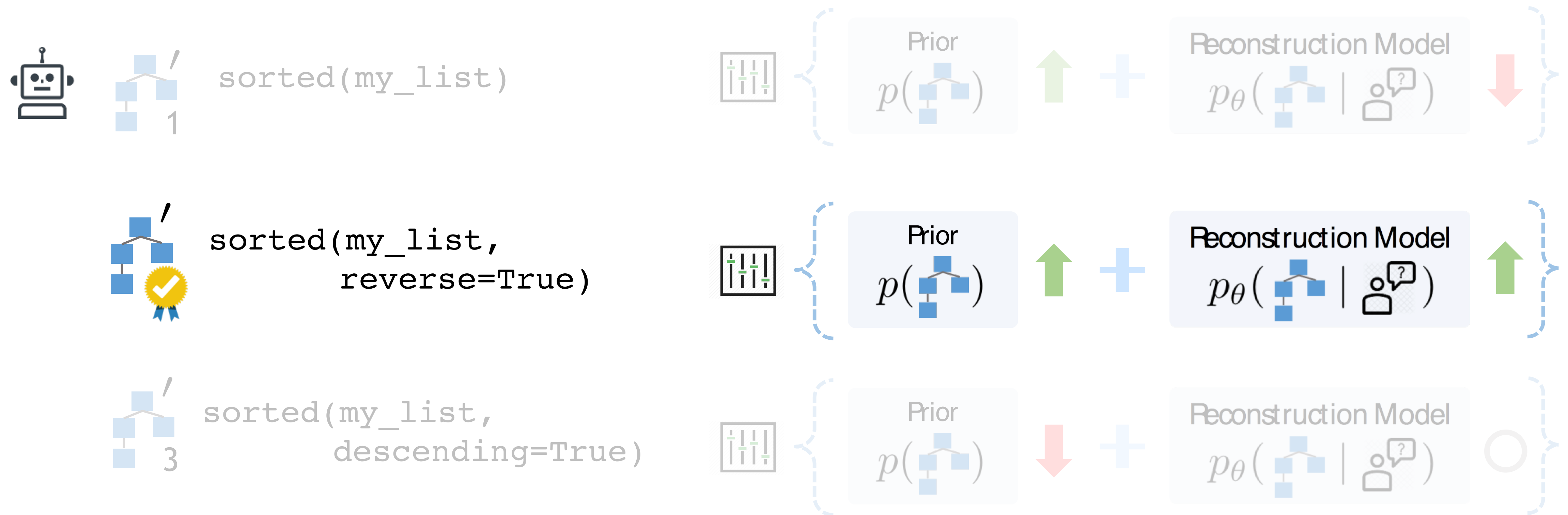
Reconstruction Model

$$p_{\theta}(\text{tree} | \text{person with ?})$$

Learning signal acts as the tuning weights of gradients received by different sampled latent meaning representations from the inference model

How does extra unlabeled data help learning?

 *Sort my_list in descending order*

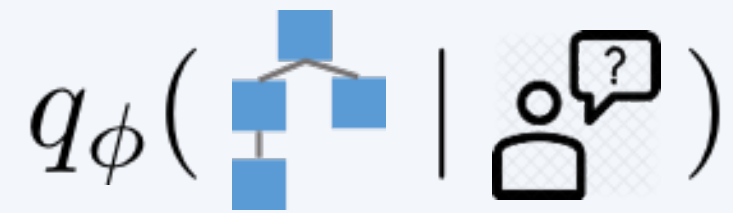


Learning favors sampled latent meaning representations that both:

- Faithfully encode the semantics of the utterance -> high reconstruction score
- Are succinct and natural -> high prior probability

The Inference Model: a Transition-based Parser

Inference Model



A transition-based parser that transduces natural language utterances into Abstract Syntax Trees

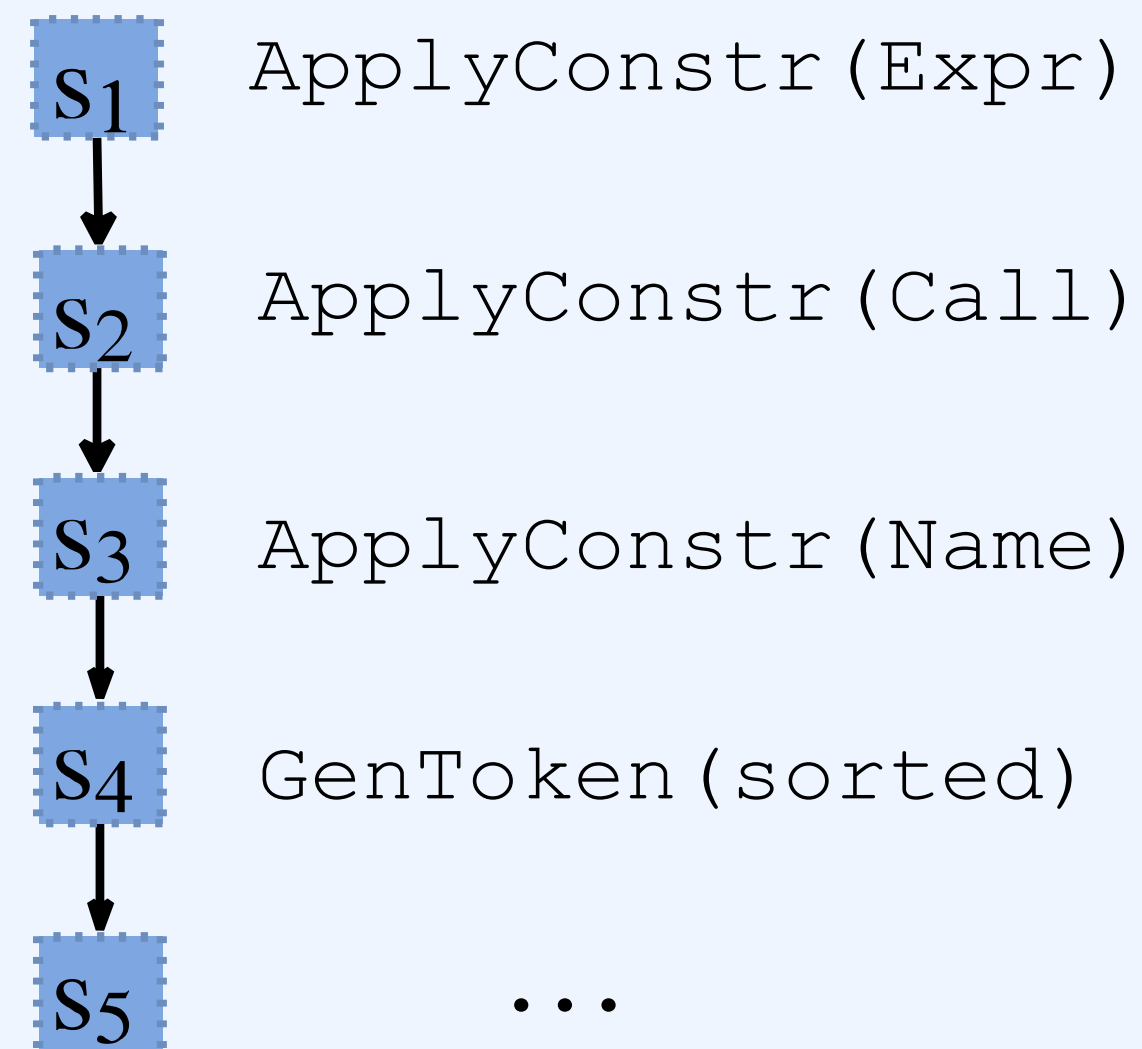
Grammar Specification

```
stmt  $\mapsto$  FunctionDef(identifier name,
                      arguments args, stmt* body)
| Expr(expr value)
expr  $\mapsto$  Call(expr func, expr* args,
               keyword* keywords)
| Name(identifier id)
| Str(string id)
```

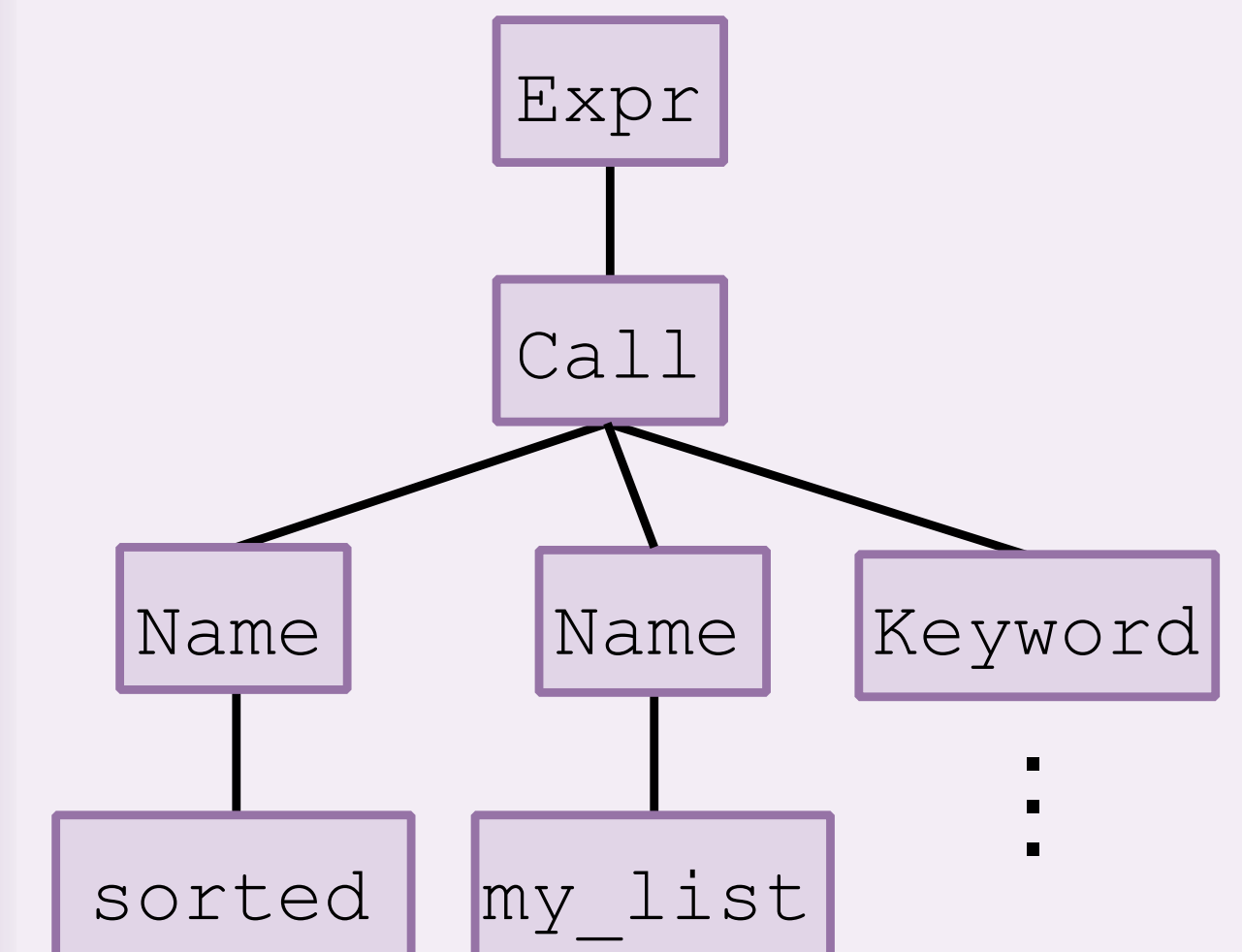
Input Utterance

Sort my_list in descending order

Transition System



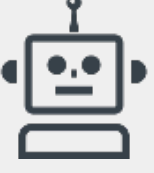
Abstract Syntax Tree



Datasets

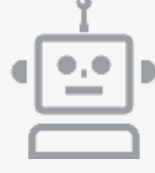
Django Python Code Generation Task

 *Call the function `_generator`, join the result into a string, return the result*

 `return ''.join(_generator())`

ATIS Semantic Parsing Task

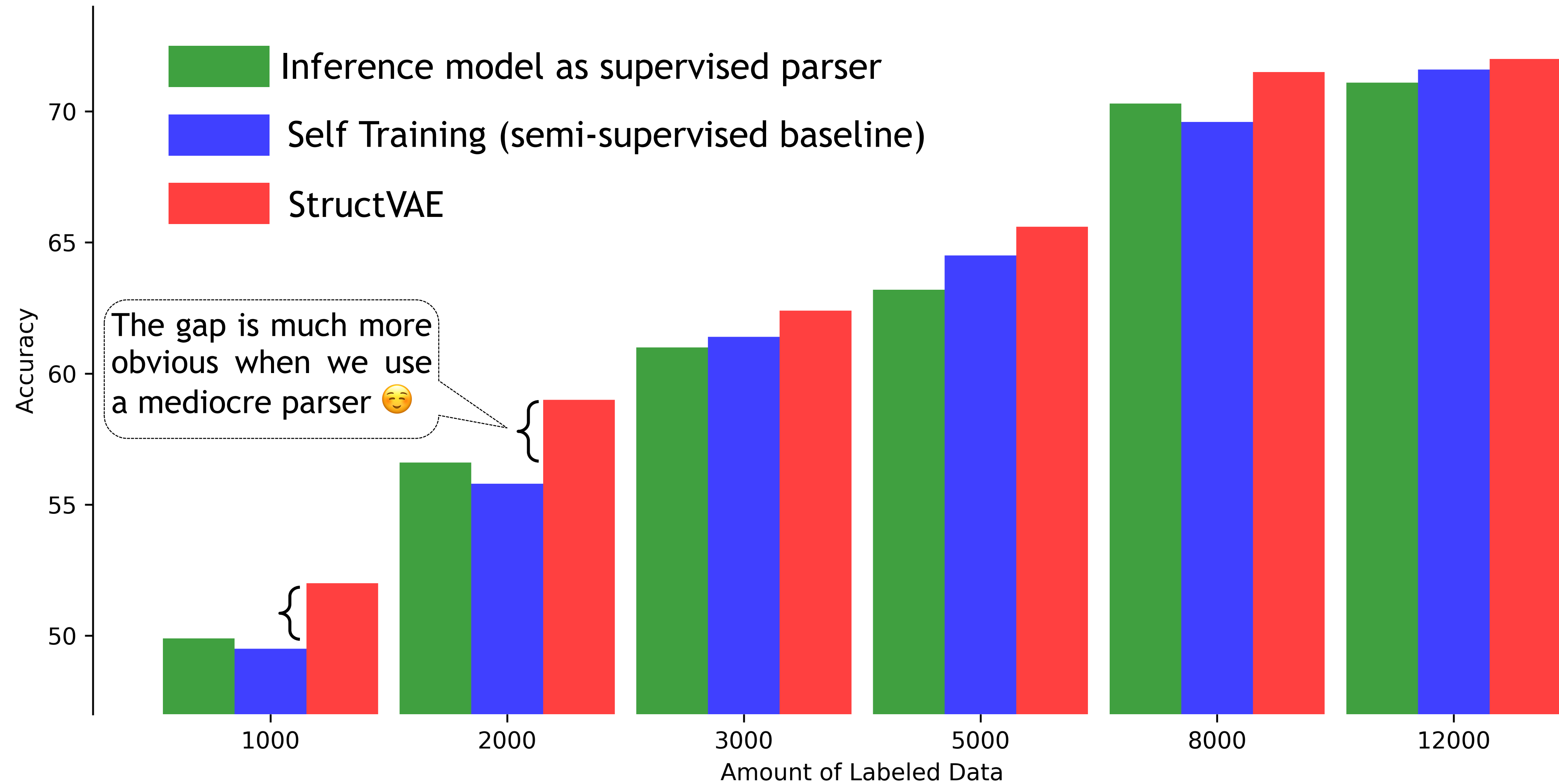
 *Show me flights from San Francisco to Washington*

 `lambda $0 e
 (and (flight $0)
 (from $0 san_Francisco:ci)
 (to $0 washington:ci))`

Research Questions

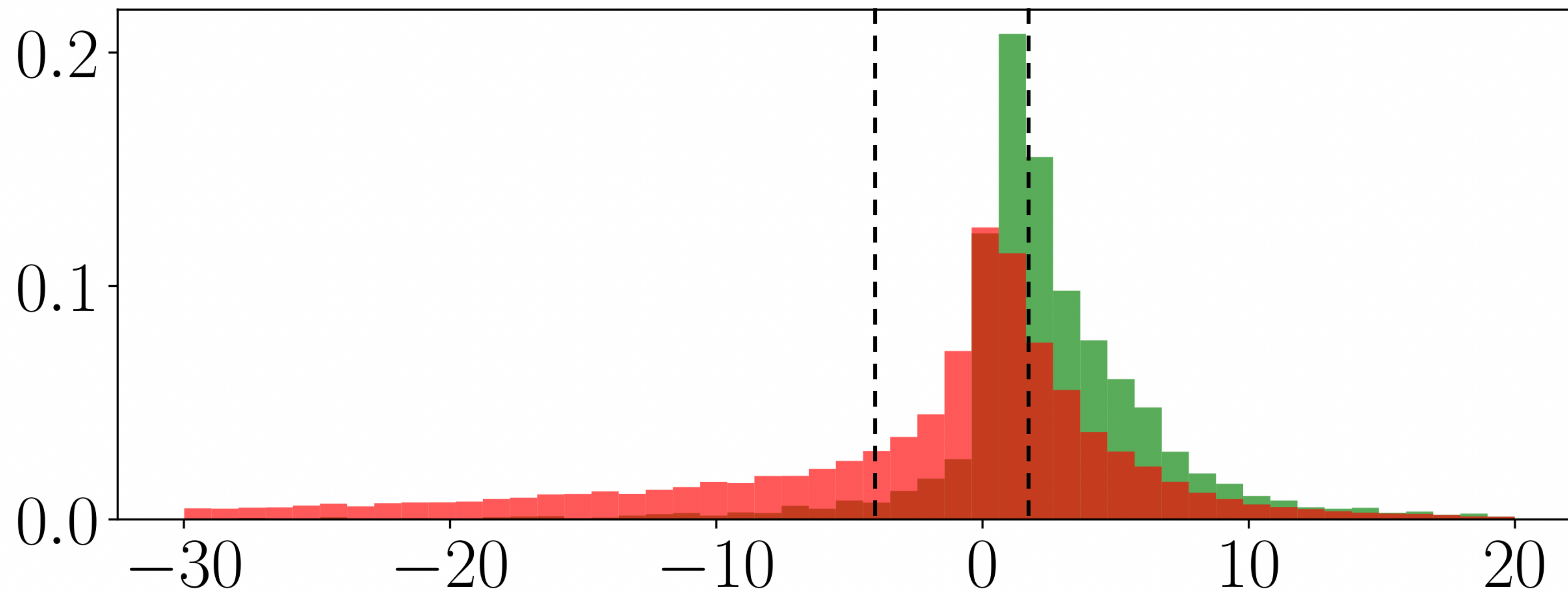
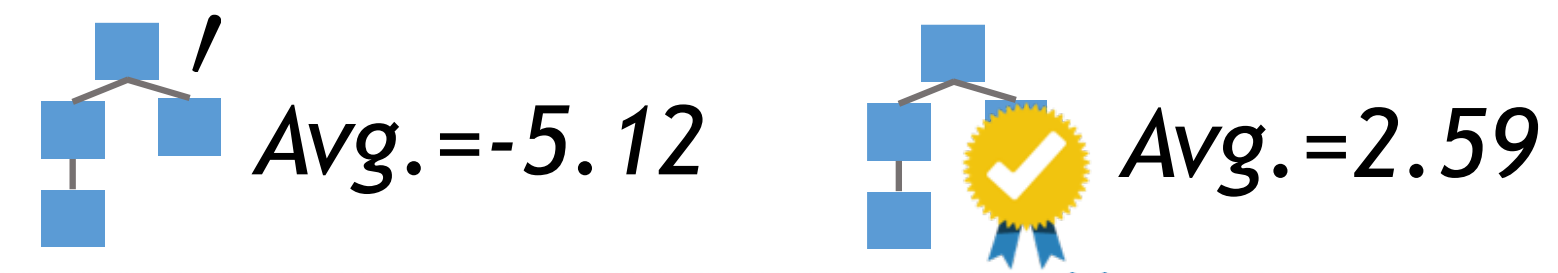
- **RQ1** Does StructVAE outperforms purely supervised semantic parsers with extra unlabeled data?
- **RQ2** Can we get some empirical evidence about why StructVAE works?

StructVAE v.s. Baselines



Why does StructVAE work?

- For each unlabeled utterance , compute the learning signal  for gold samples and other (imperfect) samples



Case Studies

 Join p and cmd into a file path, substitute it for f

	Parser Score $q_{\phi}(\text{tree} / \text{person}?)$	Prior $p(\text{tree})$	Reconstruction Score $p_{\psi}(\text{tree} / \text{person}?)$	Learning Signal 
✓ <code>f = os.path.join(p, cmd)</code>	-1.00	-24.33	-2.00	9.14
✗ <code>p = path.join(p, cmd)</code>	-8.12	-27.89	-20.96	-9.47

 Split string pks by `,`, substitute the result for `primary_keys`

	Parser Score $q_{\phi}(\text{tree} / \text{person}?)$	Prior $p(\text{tree})$	Reconstruction Score $p_{\psi}(\text{tree} / \text{person}?)$	Learning Signal 
✓ <code>primary_keys = pks.split(',')</code>	-2.38	-10.24	-11.39	2.05
✗ <code>primary_keys = pks.split + ','</code>	-1.83	-20.41	-14.87	-2.60



Carnegie Mellon University

School of Computer Science

Unsupervised Learning of Syntactic Structure w/ Invertible Neural Projections

Junxian He, Graham Neubig, Taylor Berg-Kirkpatrick
(EMNLP 2018)



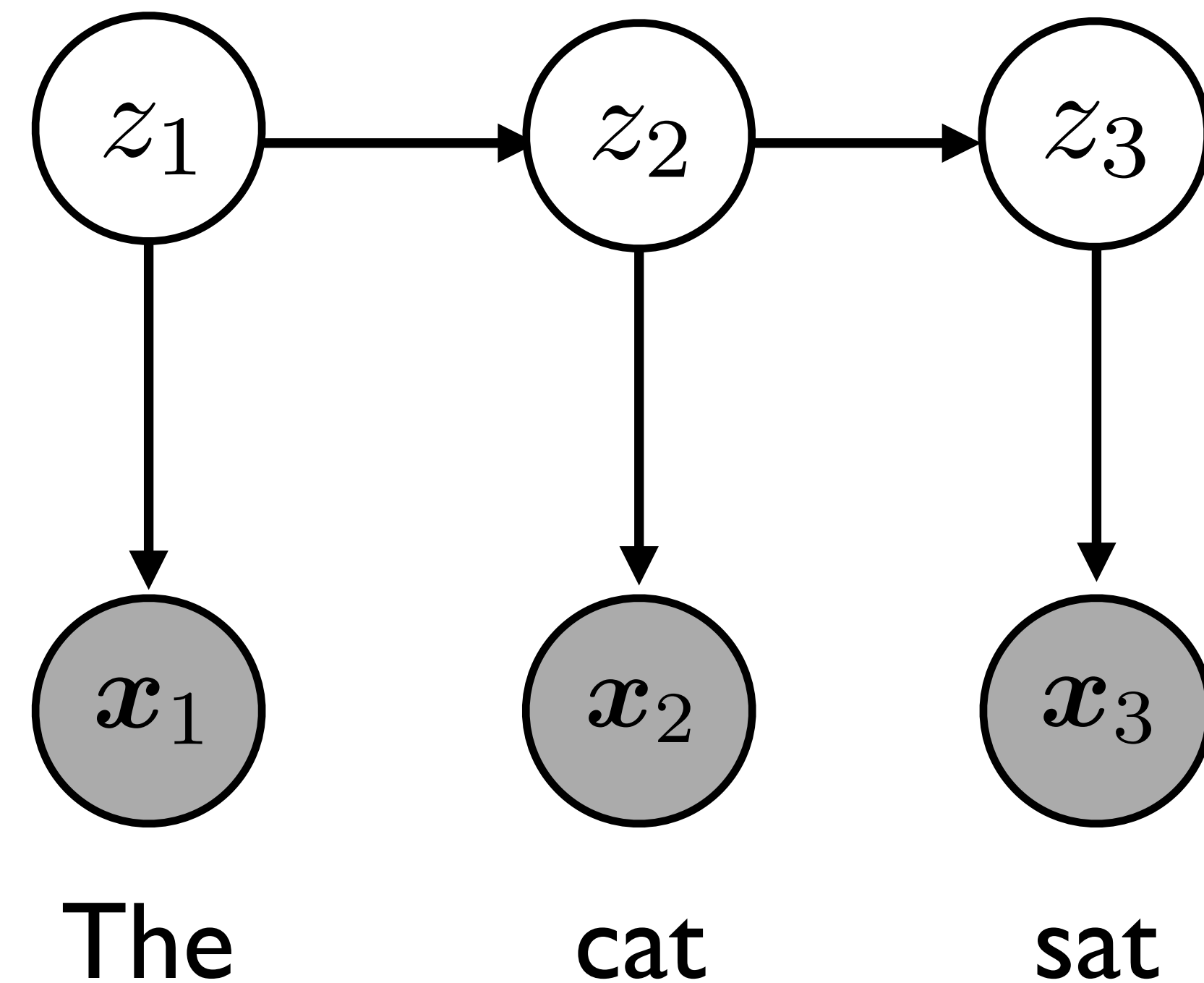
Language
Technologies
Institute



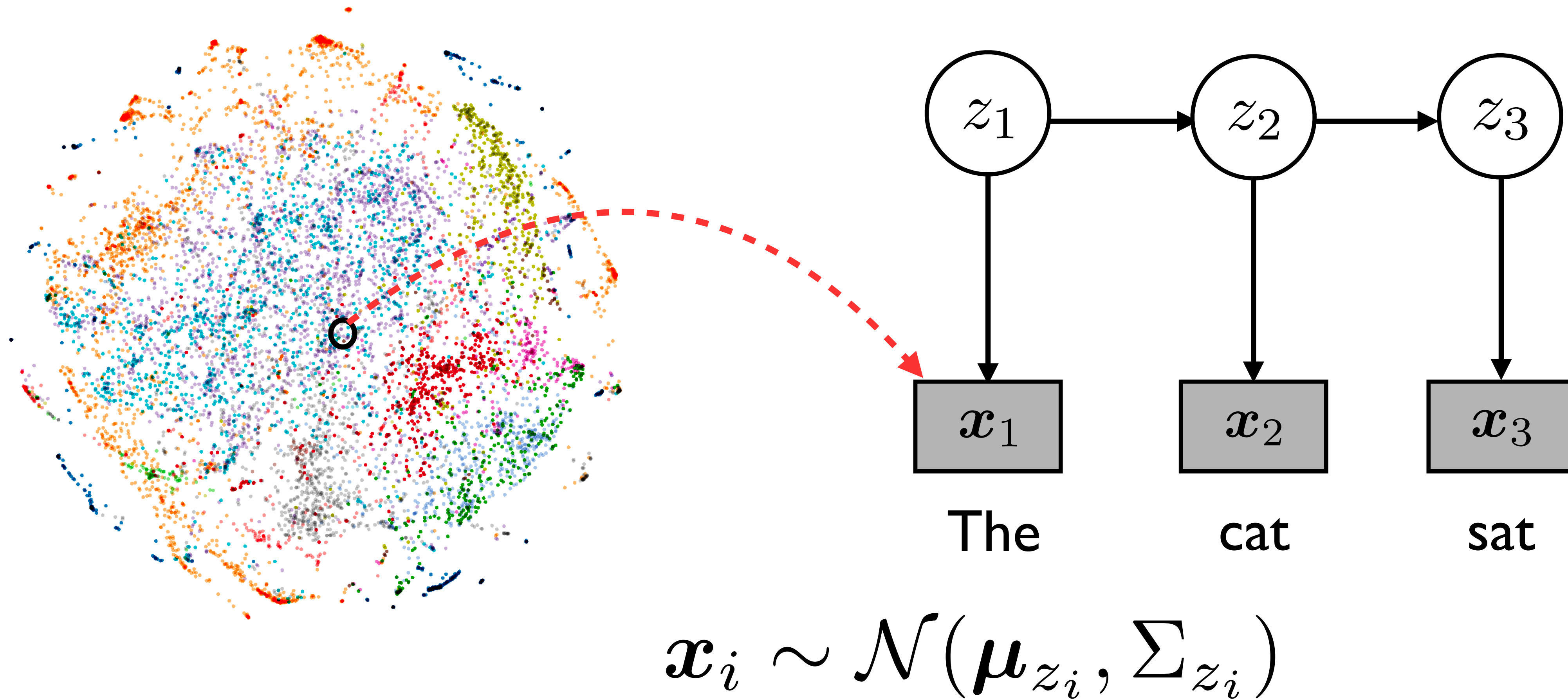
NEULAB



HMM for Part-of-Speech Induction



Gaussian HMM for POS Induction

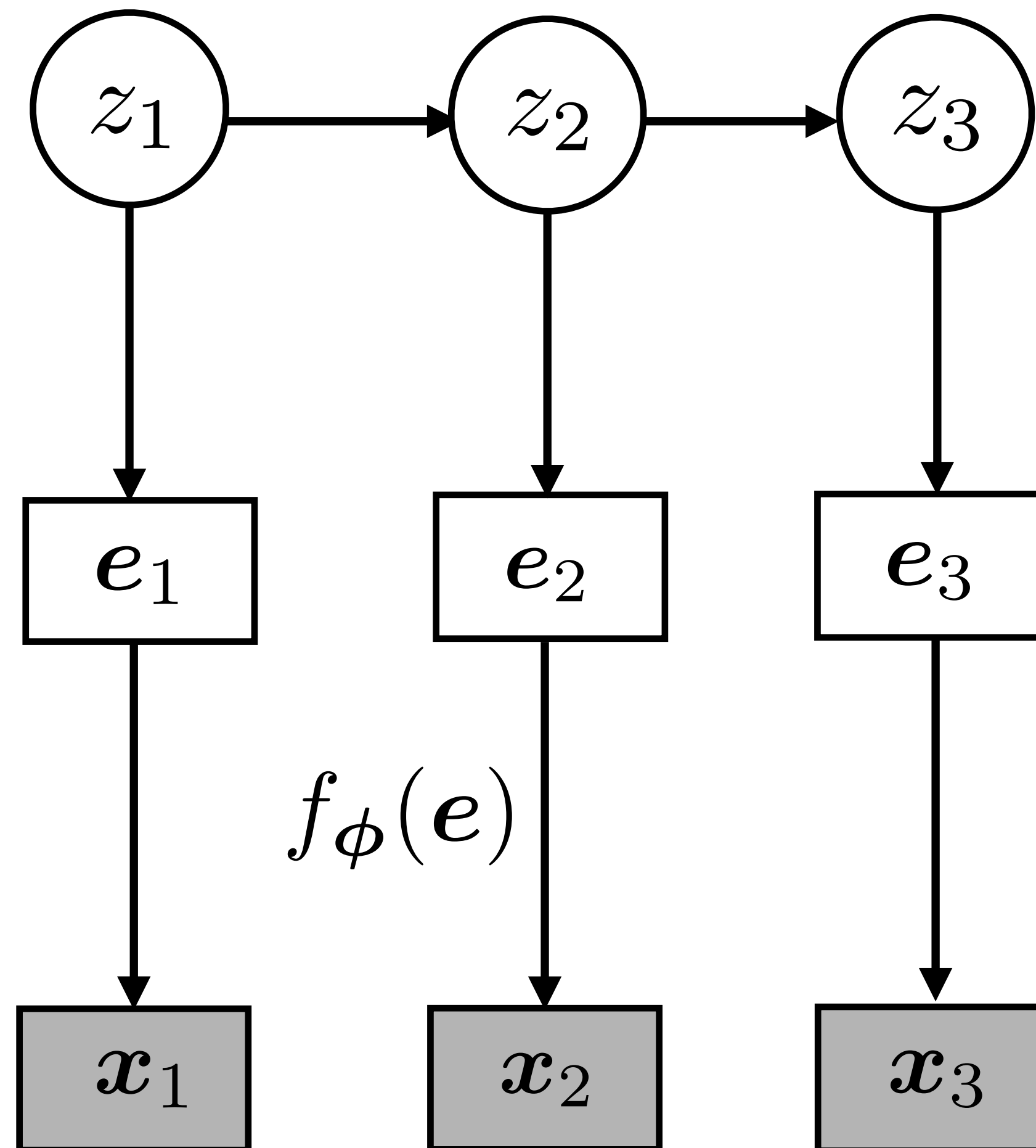
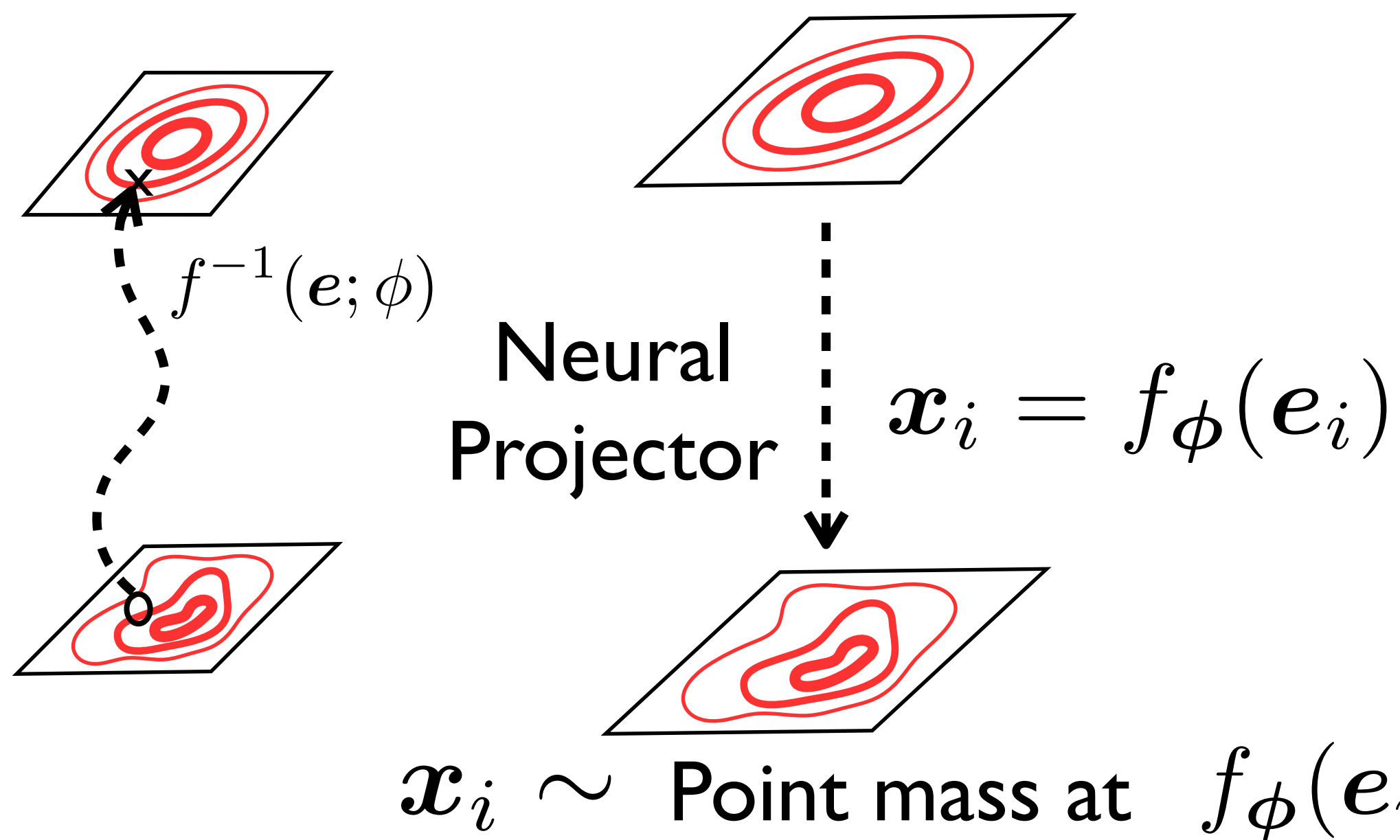


[Lin et al. 2015]

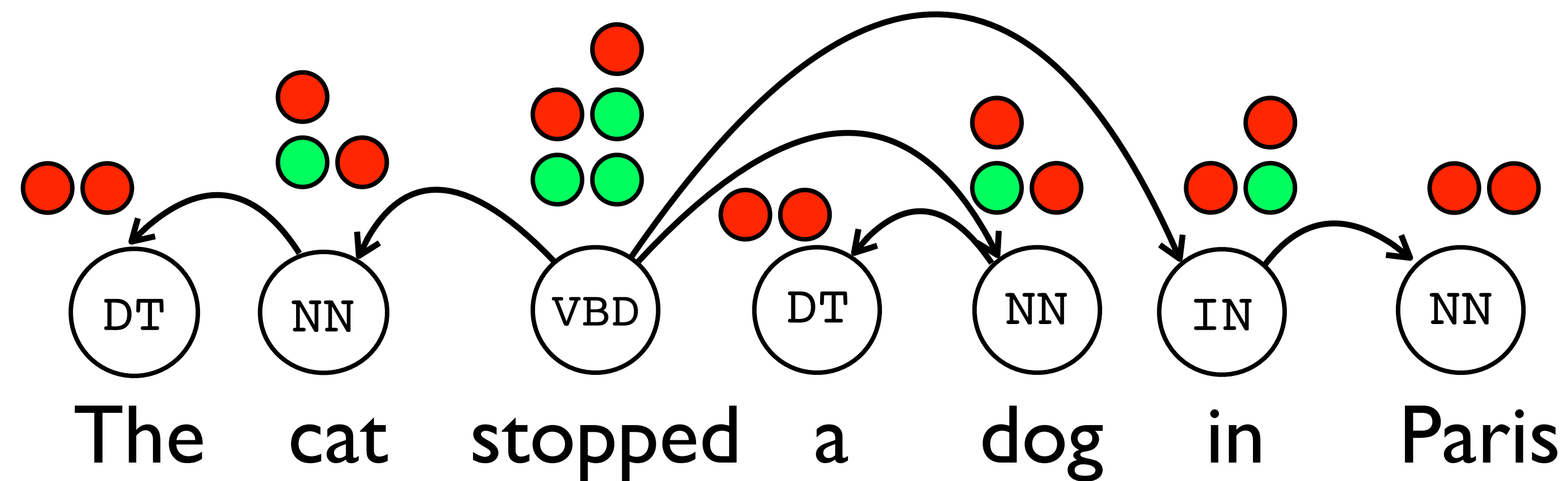
Latent Embeddings w/ Neural Projection

$z_i \sim$ Markov Structure

$$e_i \sim \mathcal{N}(\mu_{z_i}, \Sigma_{z_i})$$

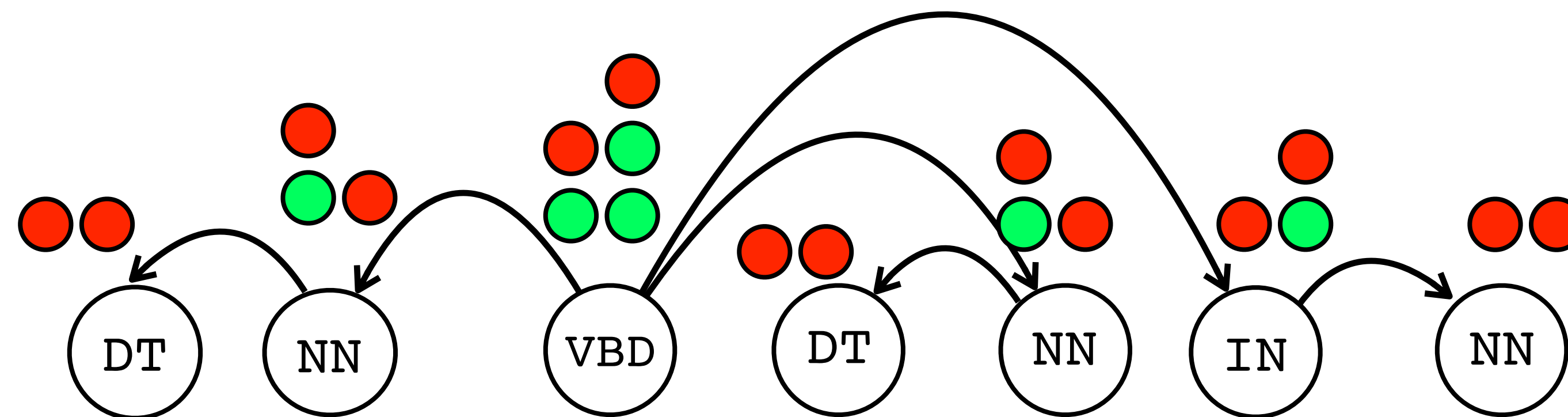


Dependency Model with Valence



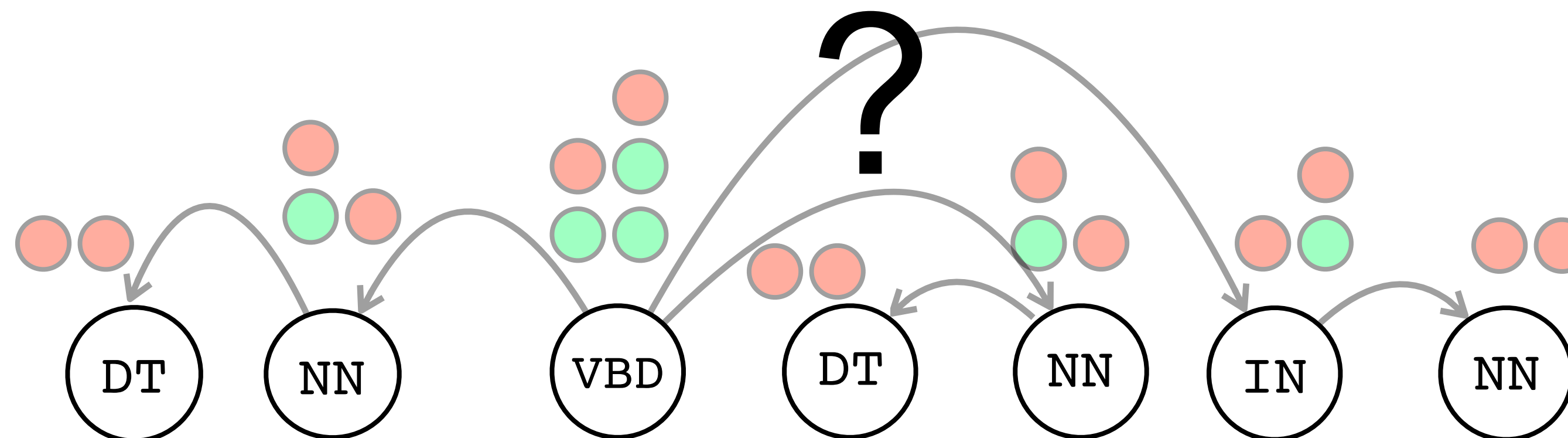
[Klein and Manning 2004]

Dependency Model with Valence

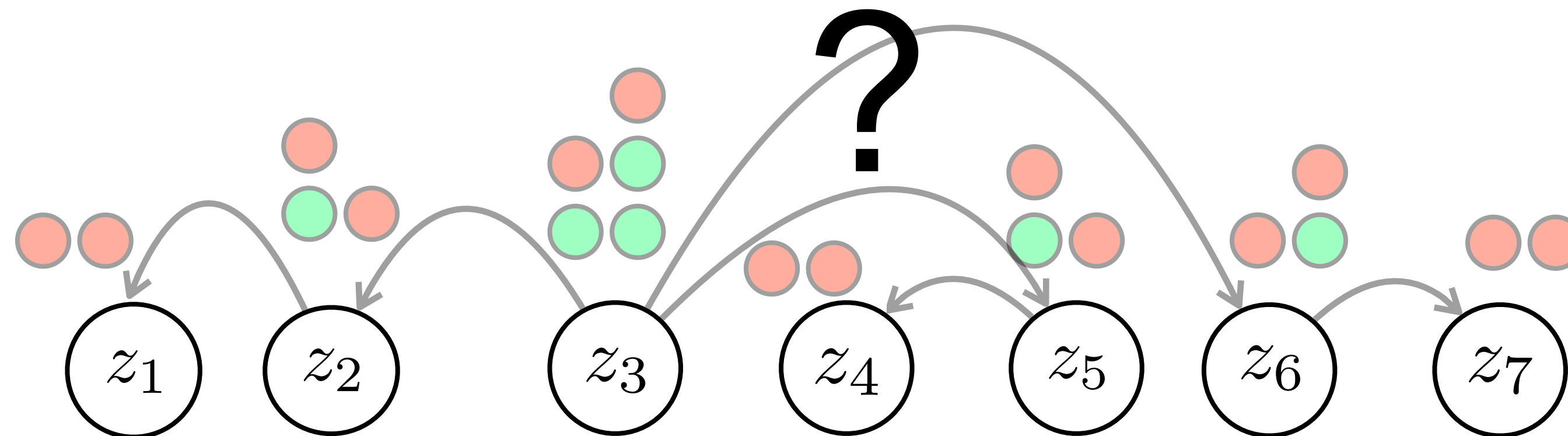


[Klein and Manning 2004]

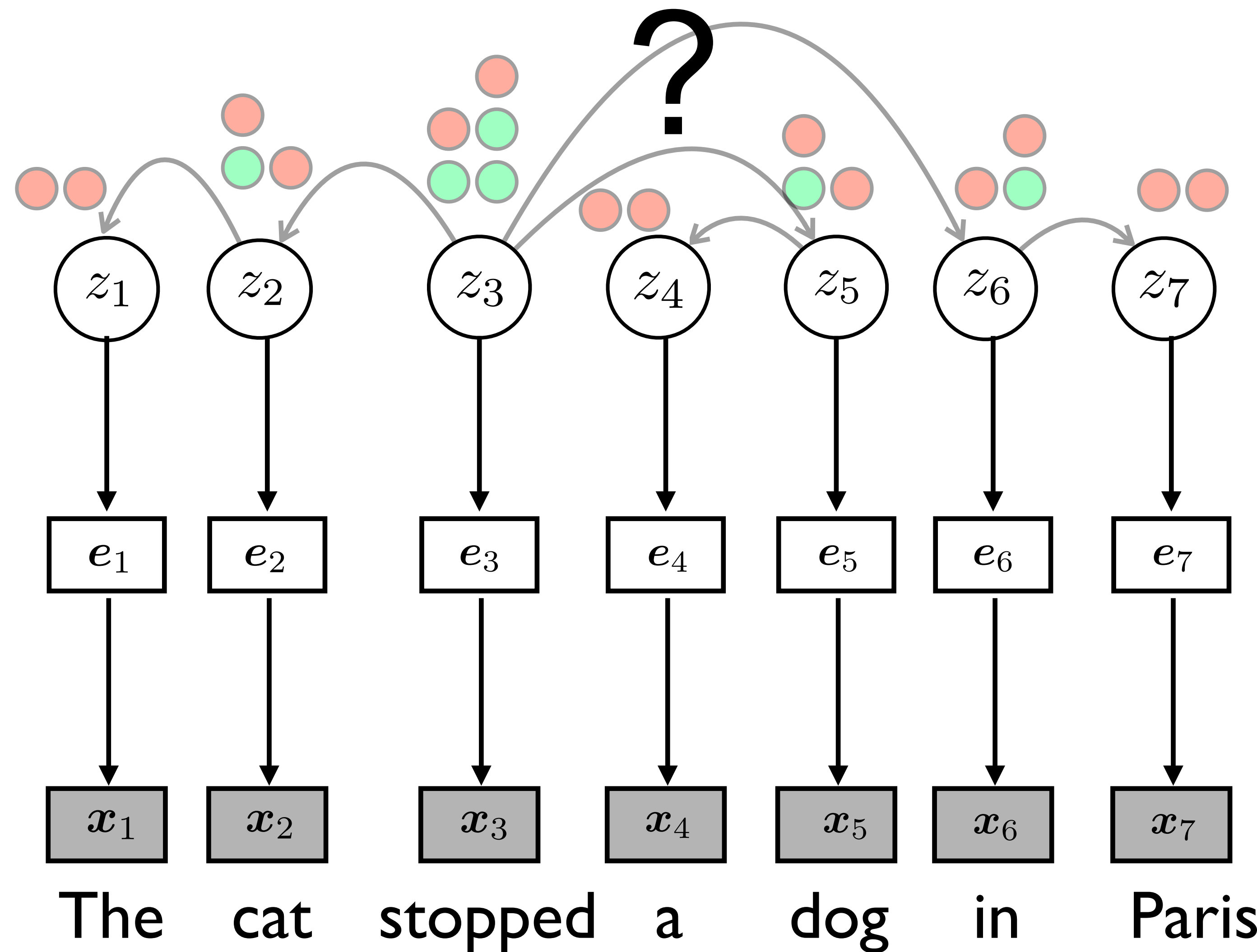
Dependency Parse Induction from POS



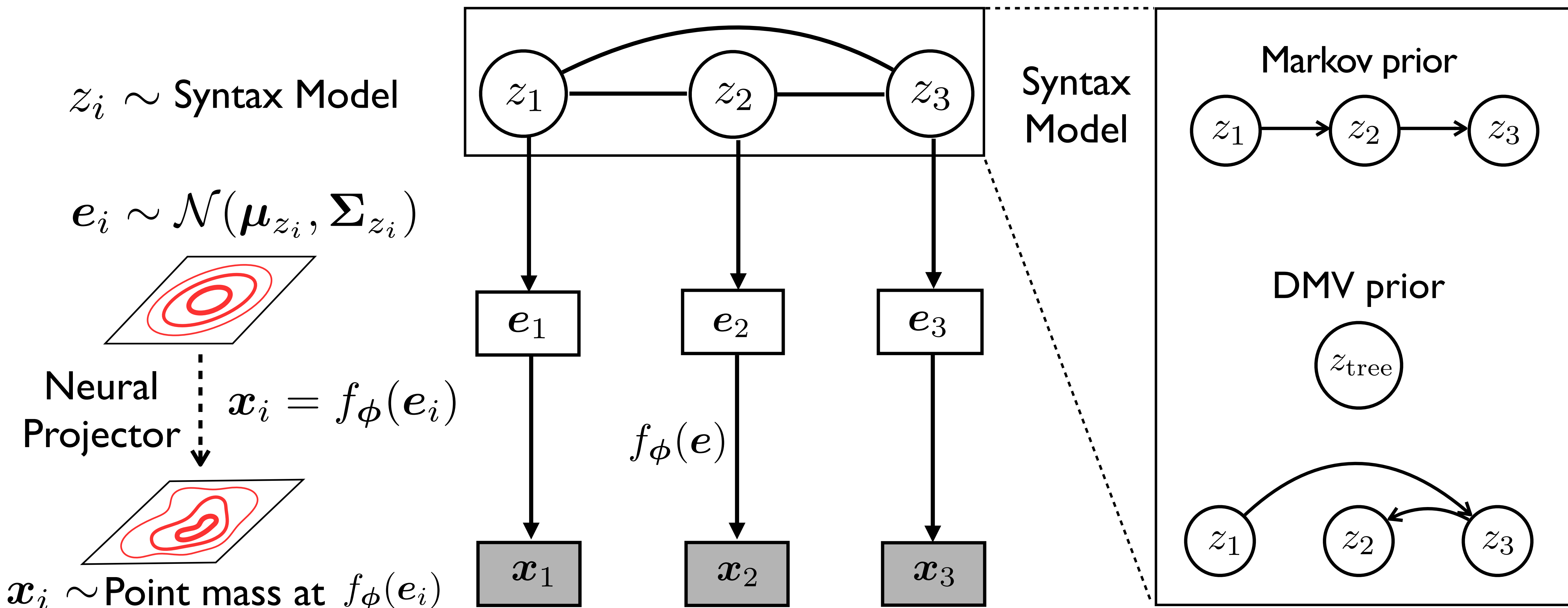
Grammar Induction from Raw Text



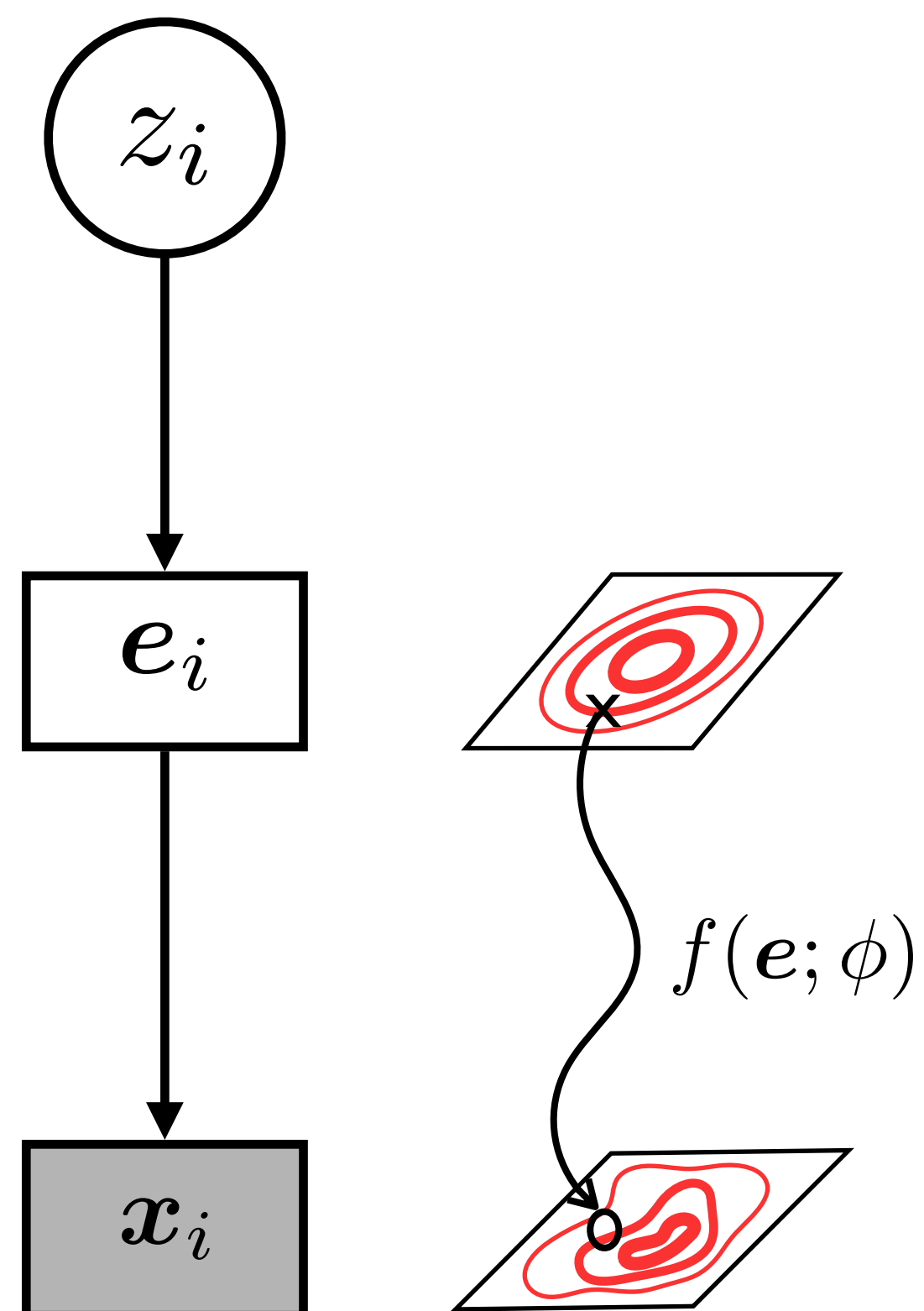
Grammar Induction from Raw Text



Latent Embeddings w/ Neural Projection

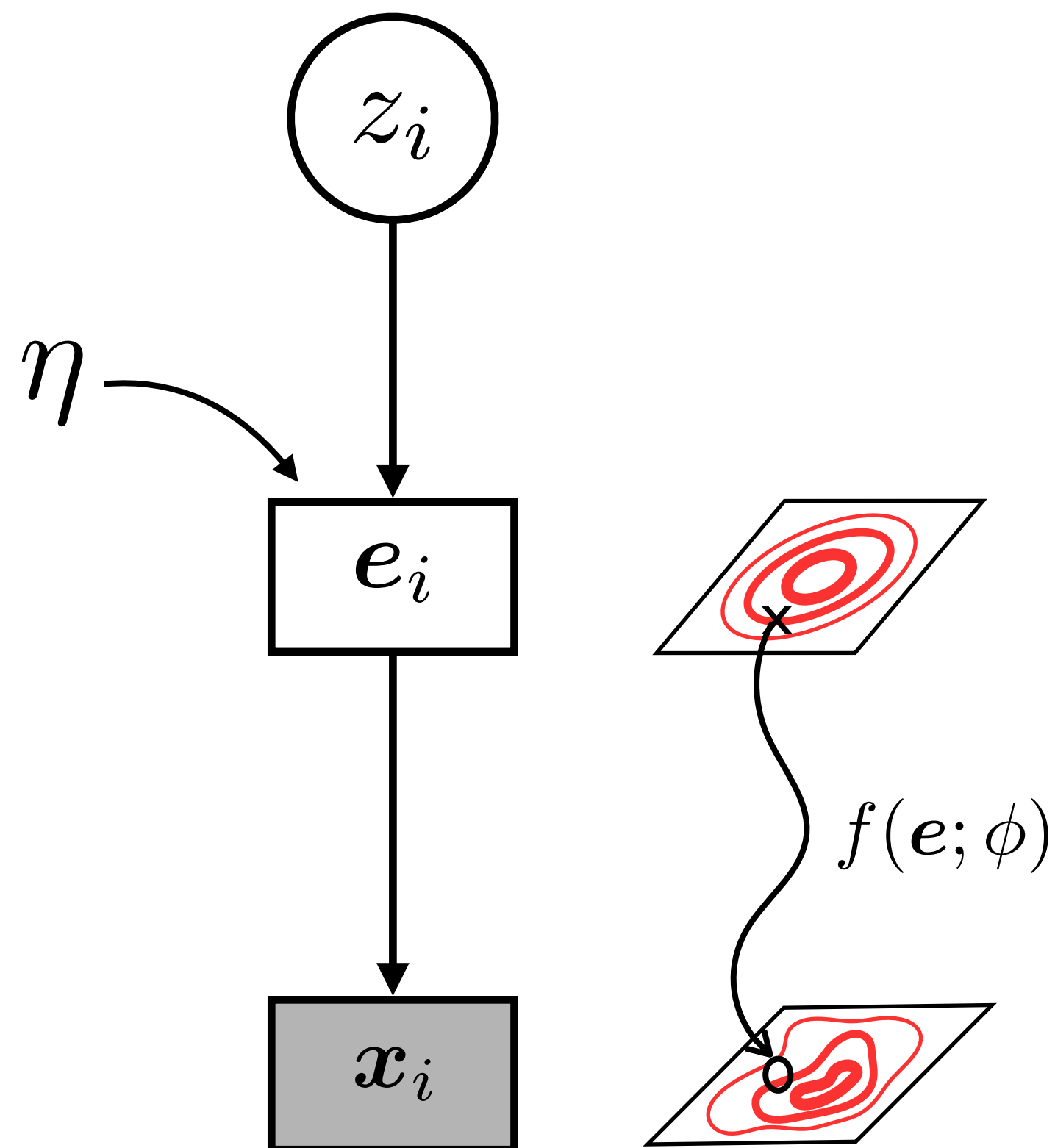


Learning and Inference



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

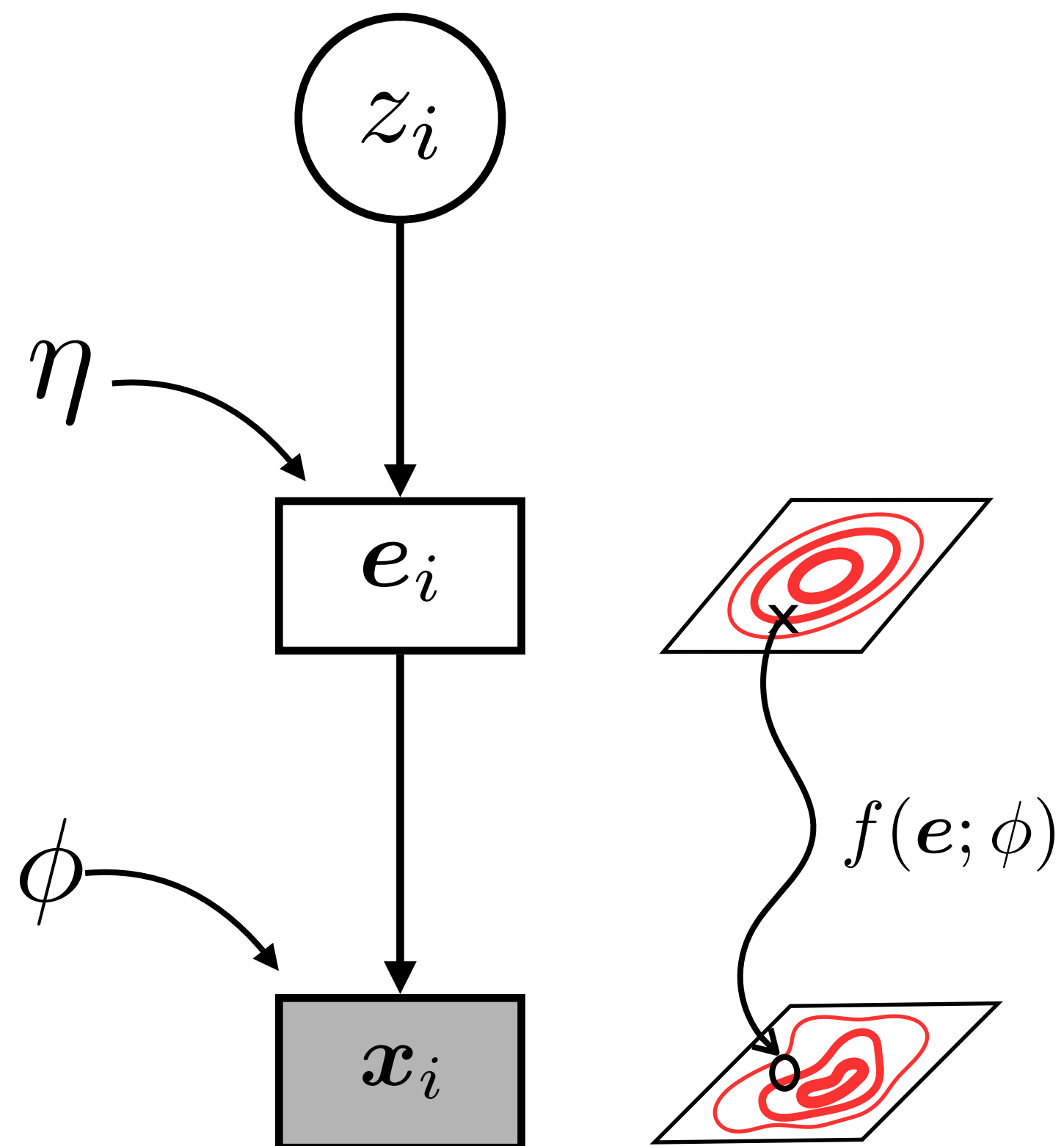
Learning and Inference



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

Gaussian embedding parameters

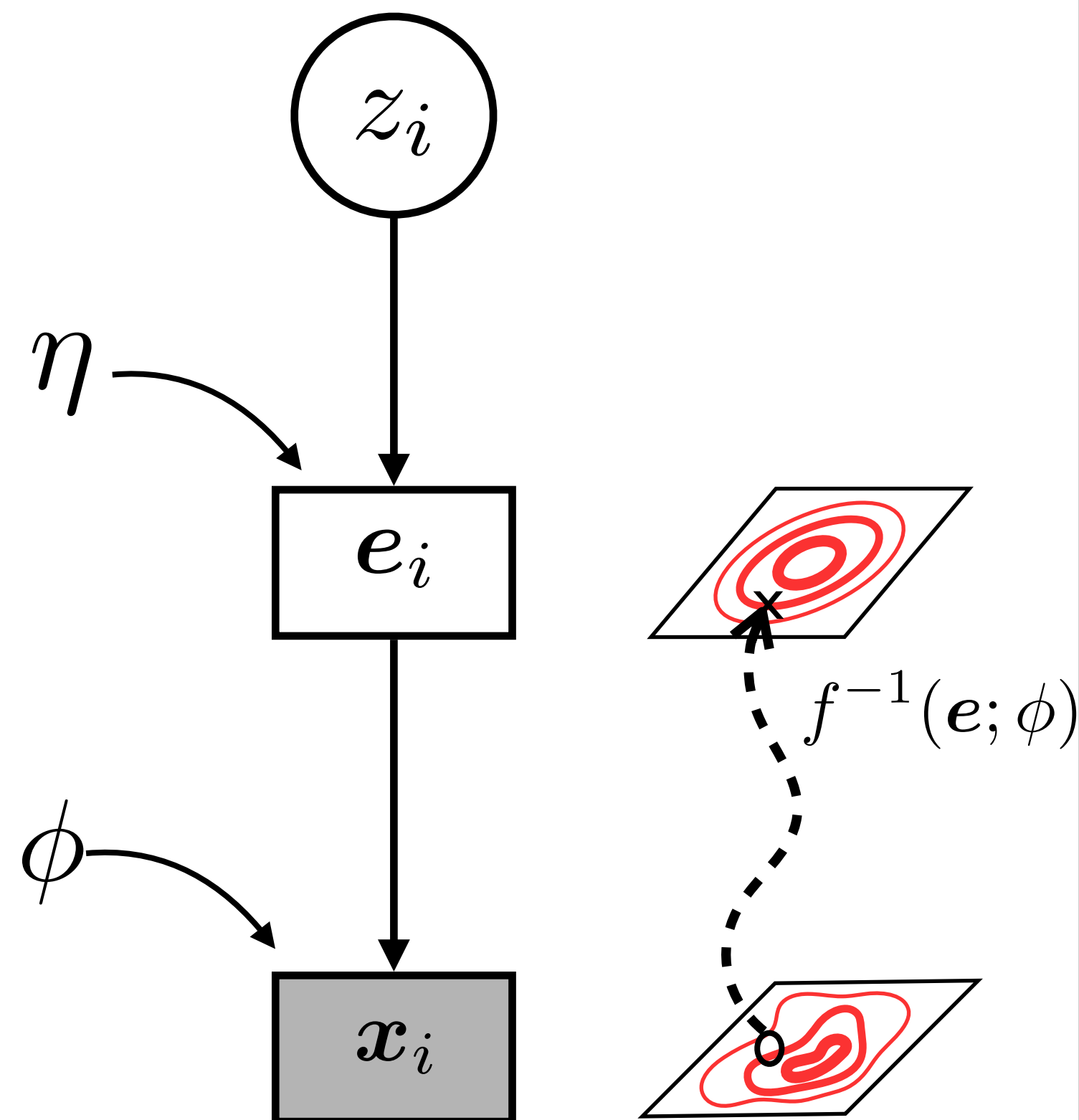
Learning and Inference



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

Projection parameters

Learning and Inference

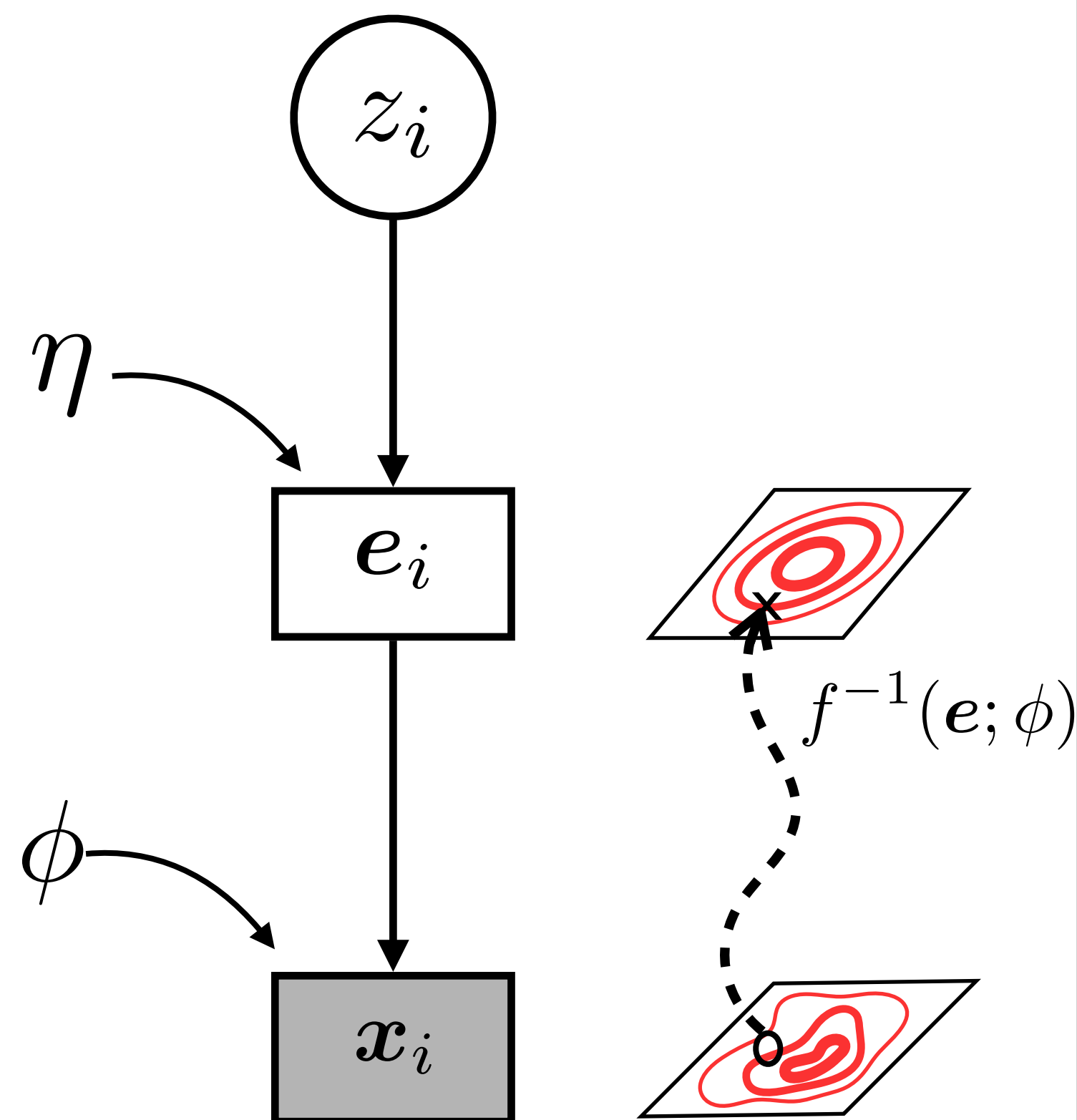


$\dim(\mathbf{x}) = \dim(\mathbf{e})$ and f is invertible

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

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

Learning and Inference



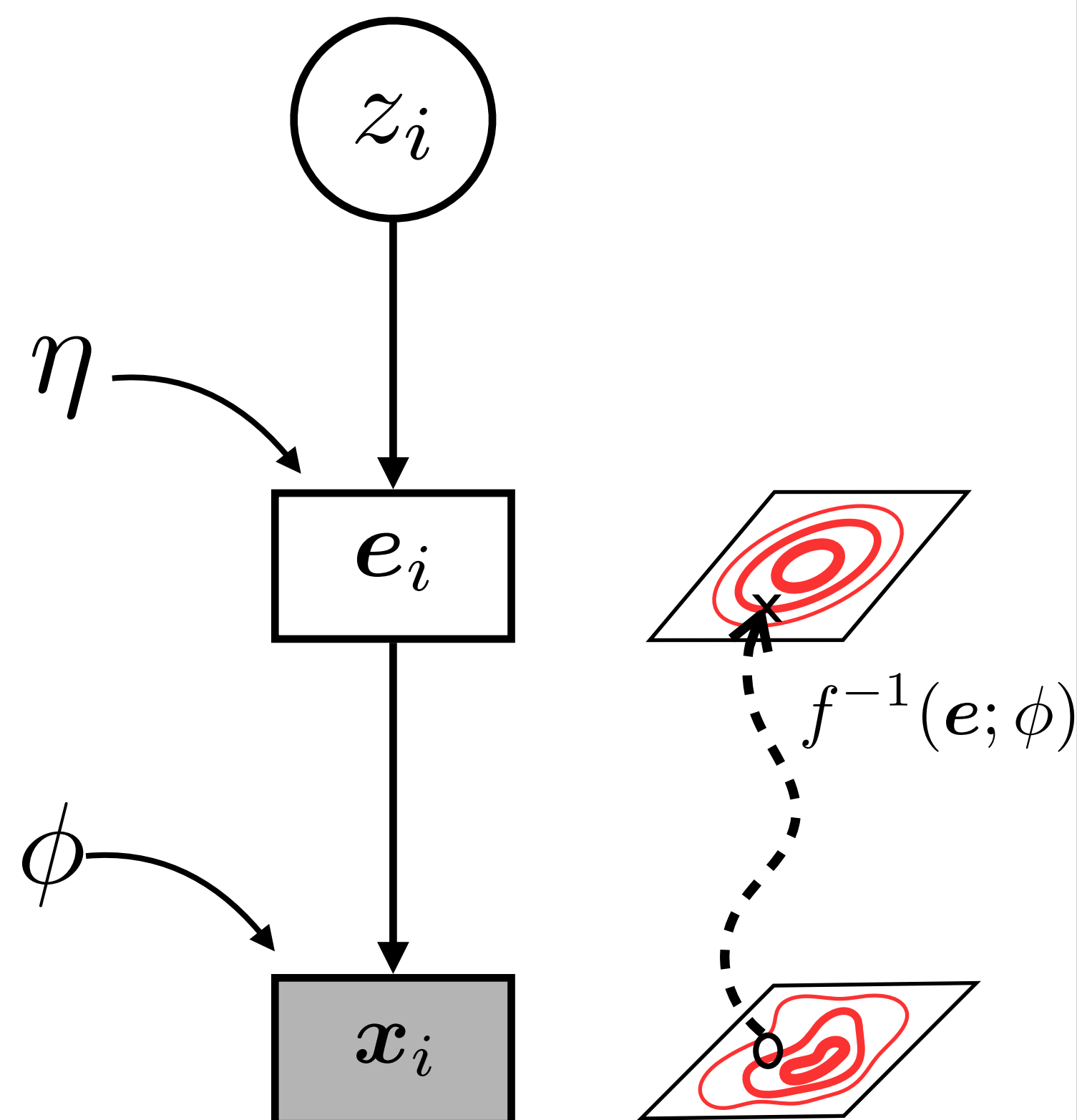
$\dim(\mathbf{x}) = \dim(\mathbf{e})$ and f is invertible

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

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

Determinant of Jacobian matrix

Learning and Inference



$\dim(\mathbf{x}) = \dim(\mathbf{e})$ and f is invertible

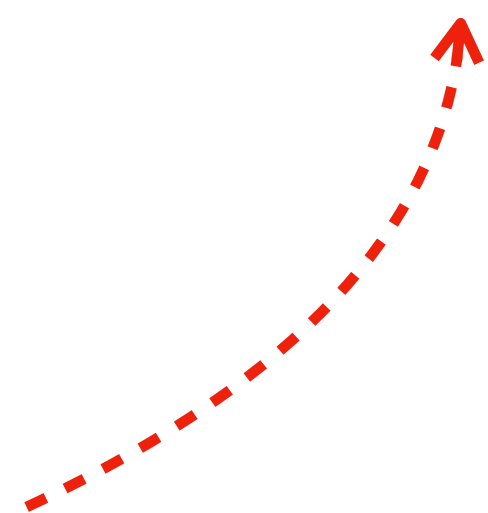
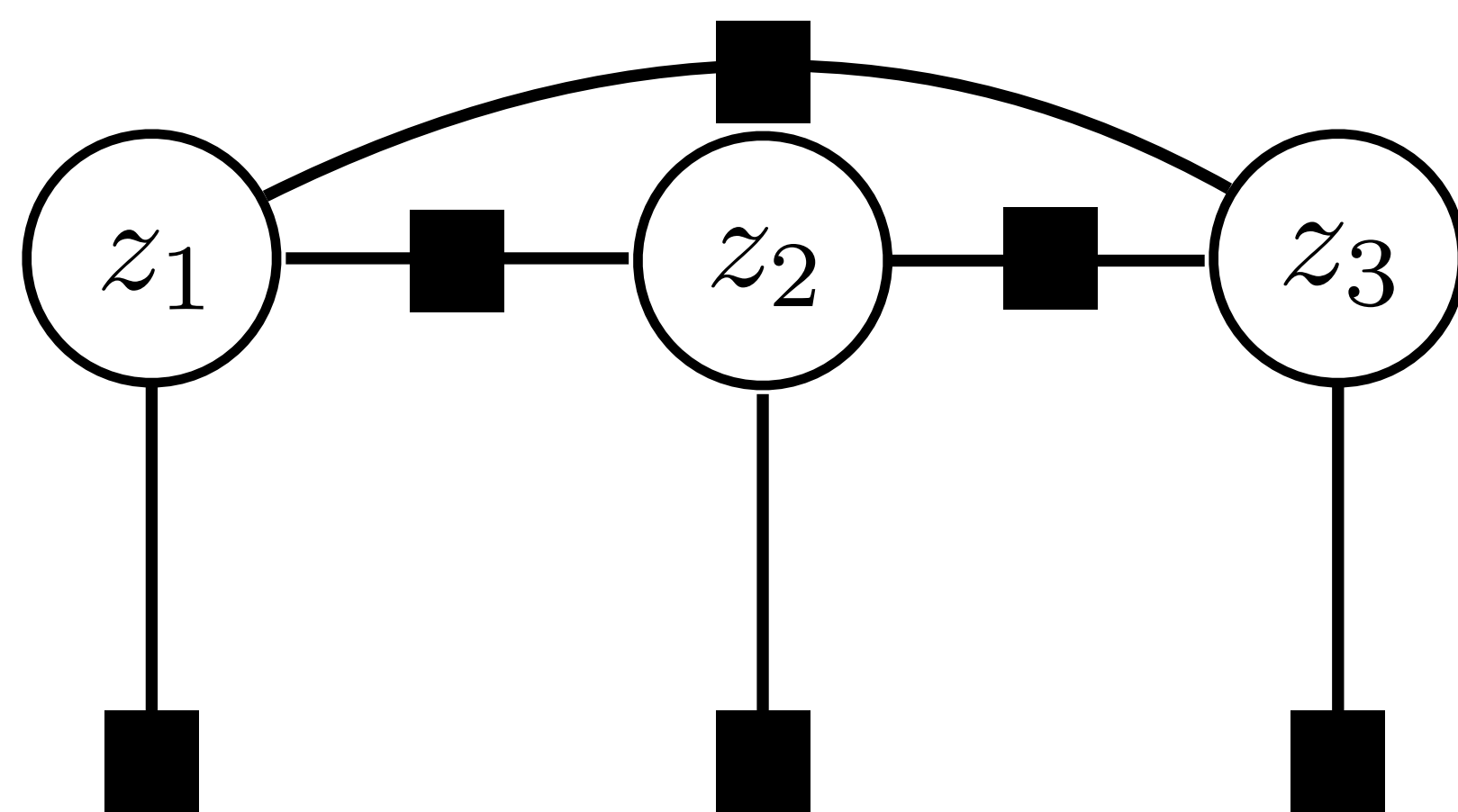
$$p(\mathbf{x}_i | z_i; \eta, \phi)$$

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

Gaussian distribution

Determinant of Jacobian matrix

Learning and Inference



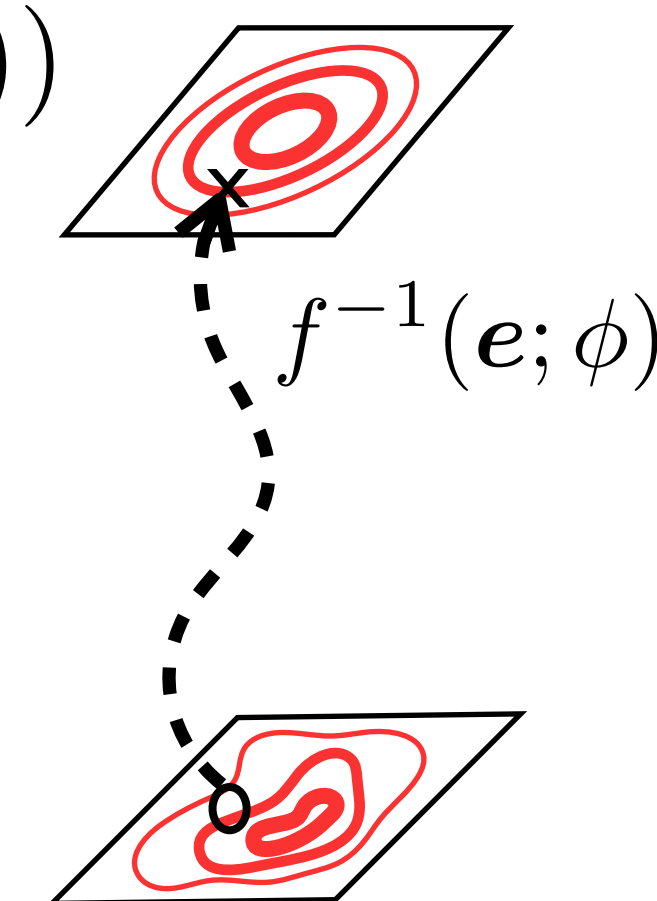
$$p(f_{\phi}^{-1}(\mathbf{x}_i) | z_i; \eta) \left| \det \frac{\partial f^{-1}}{\partial \mathbf{x}_i} \right|$$

Example of Markov prior

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

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

$-\infty$ when f is not invertible

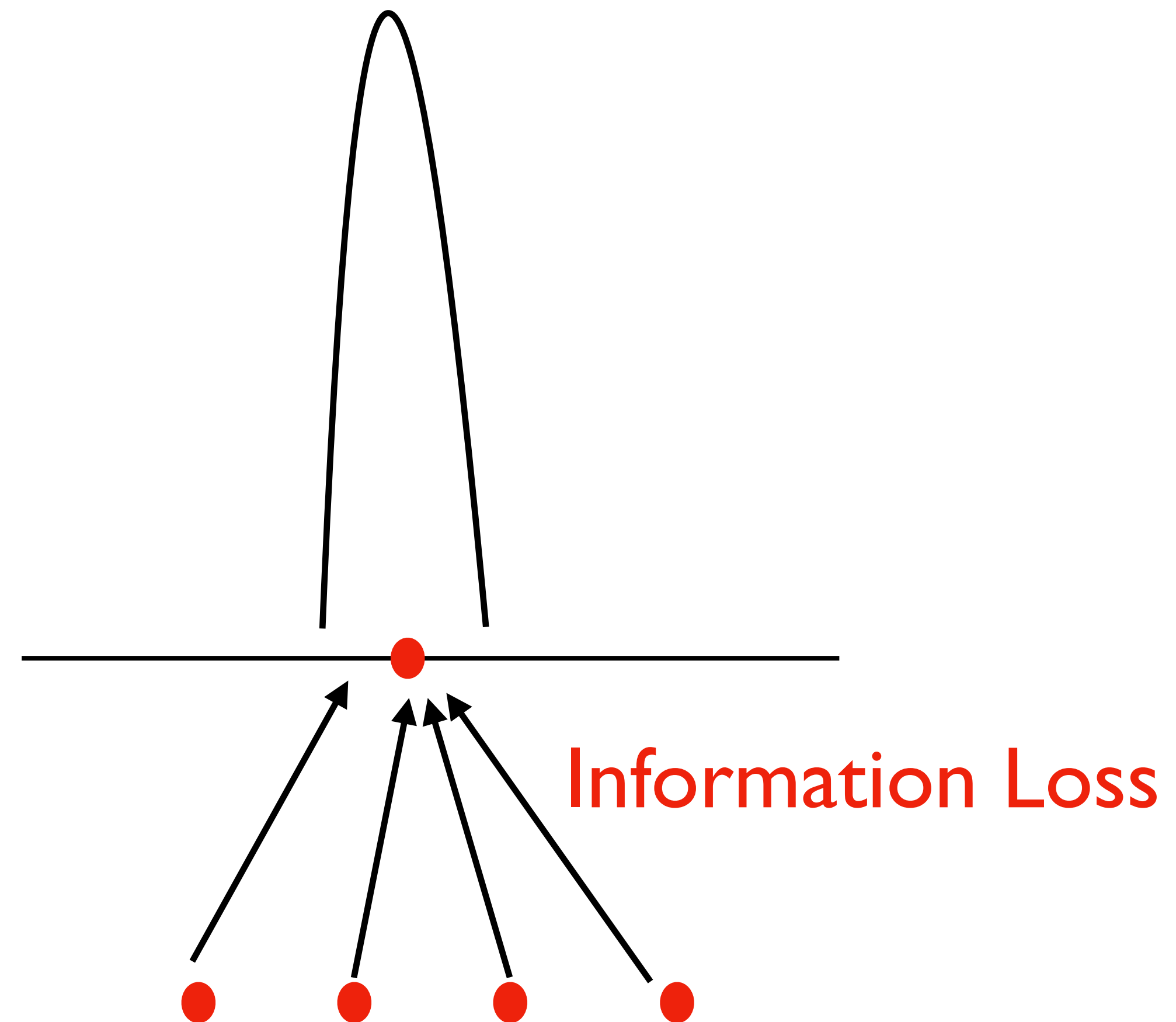
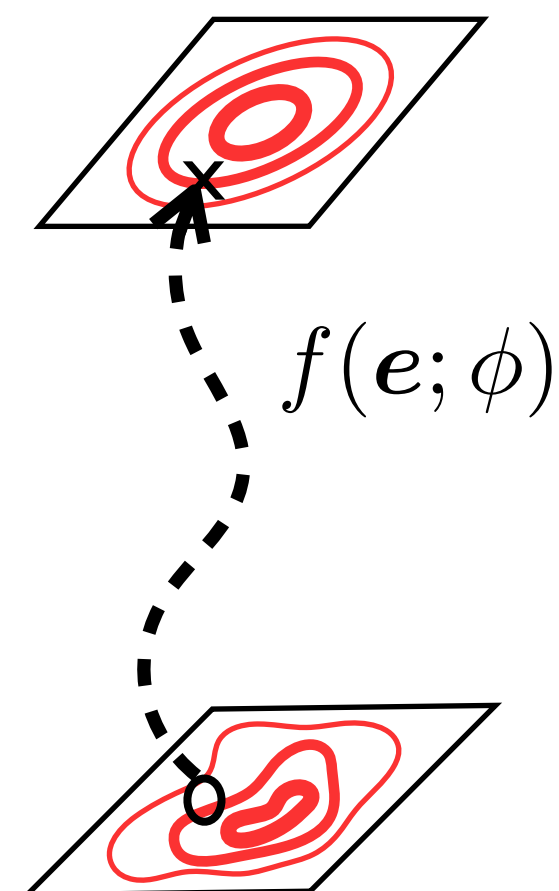


Why Invertible

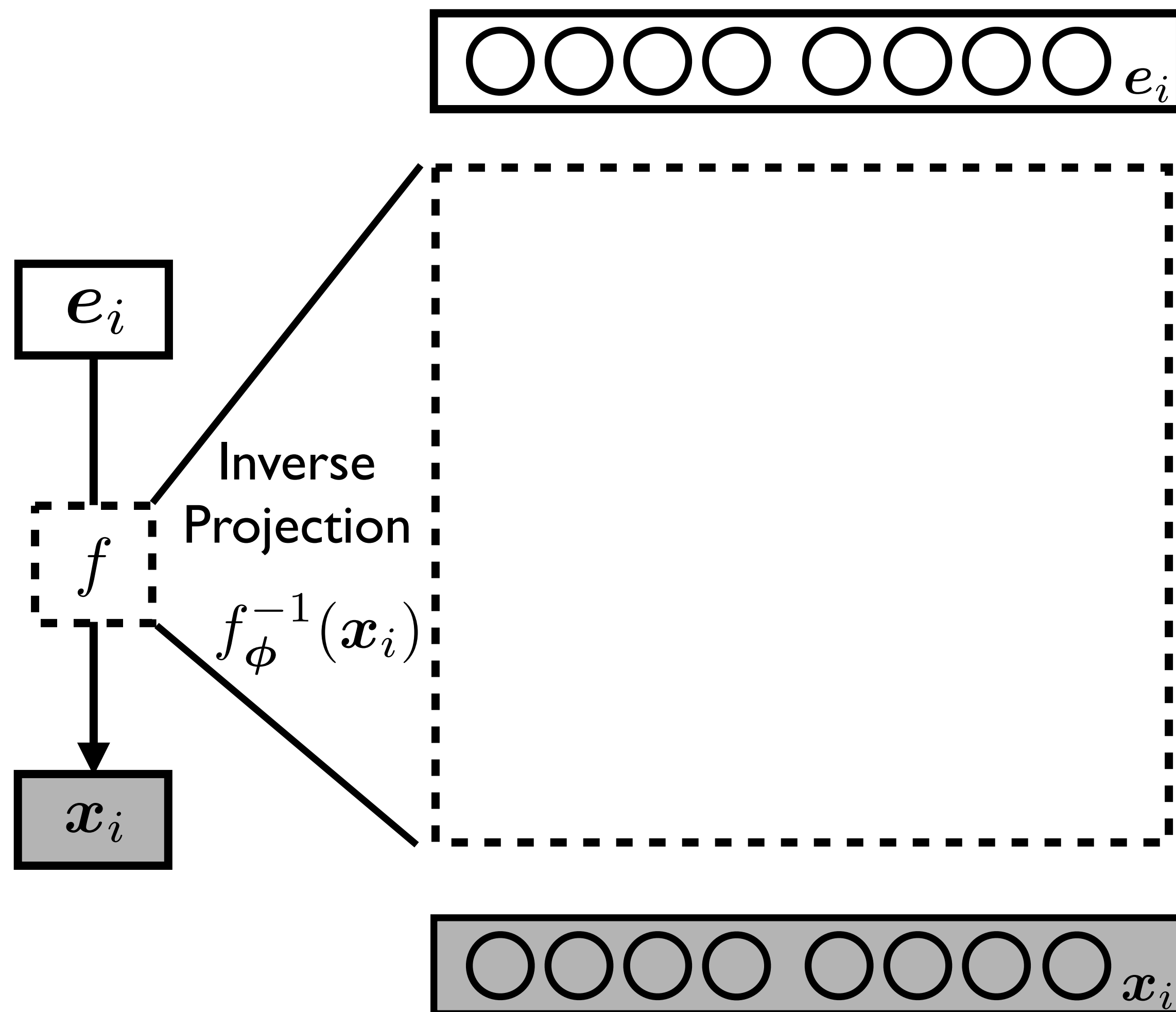
Example of Markov prior

$$\max \log p_{\text{GHMM}}(f_{\phi}(\mathbf{x}))$$

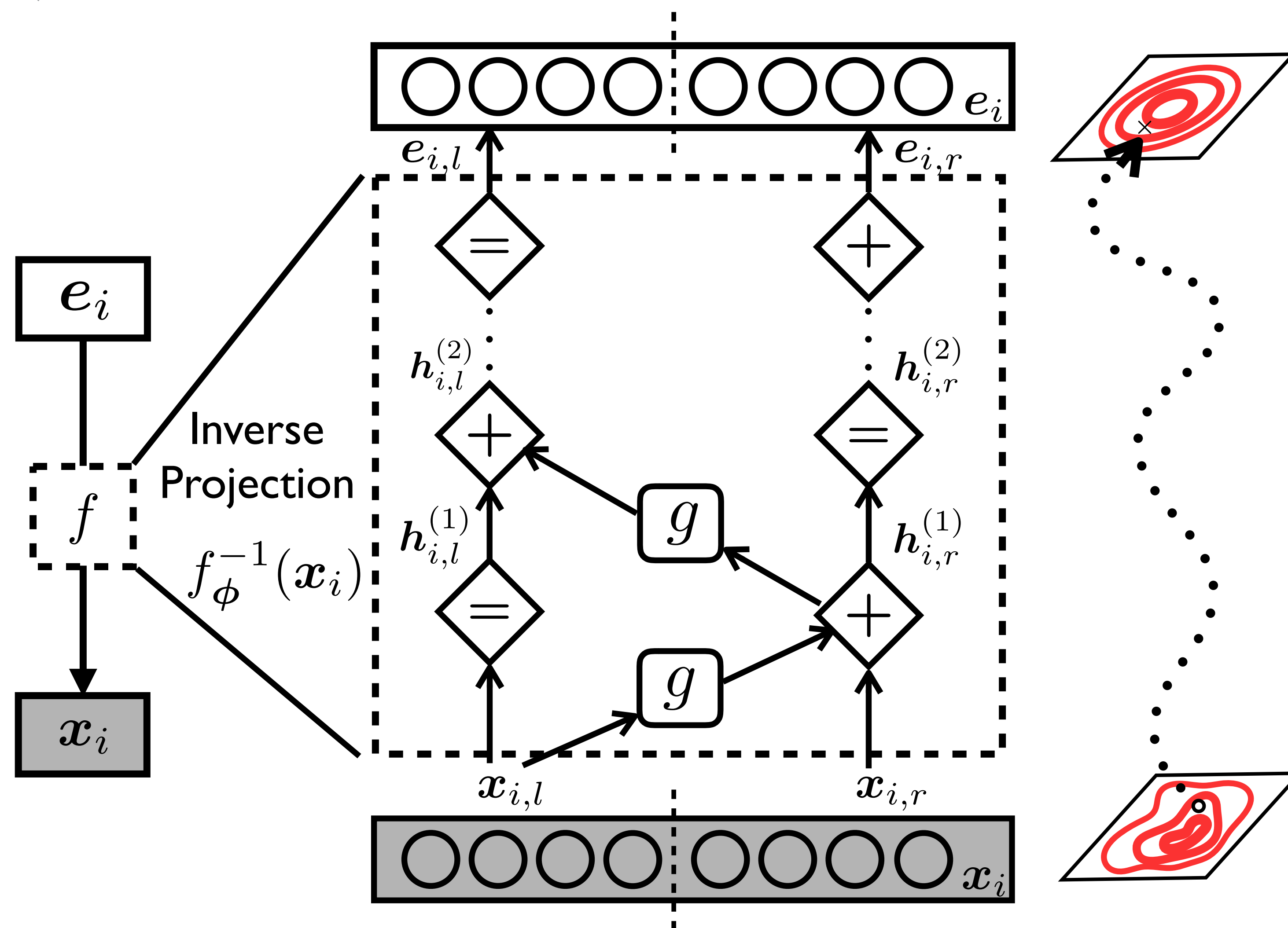
?



Learning with Inverse Projection

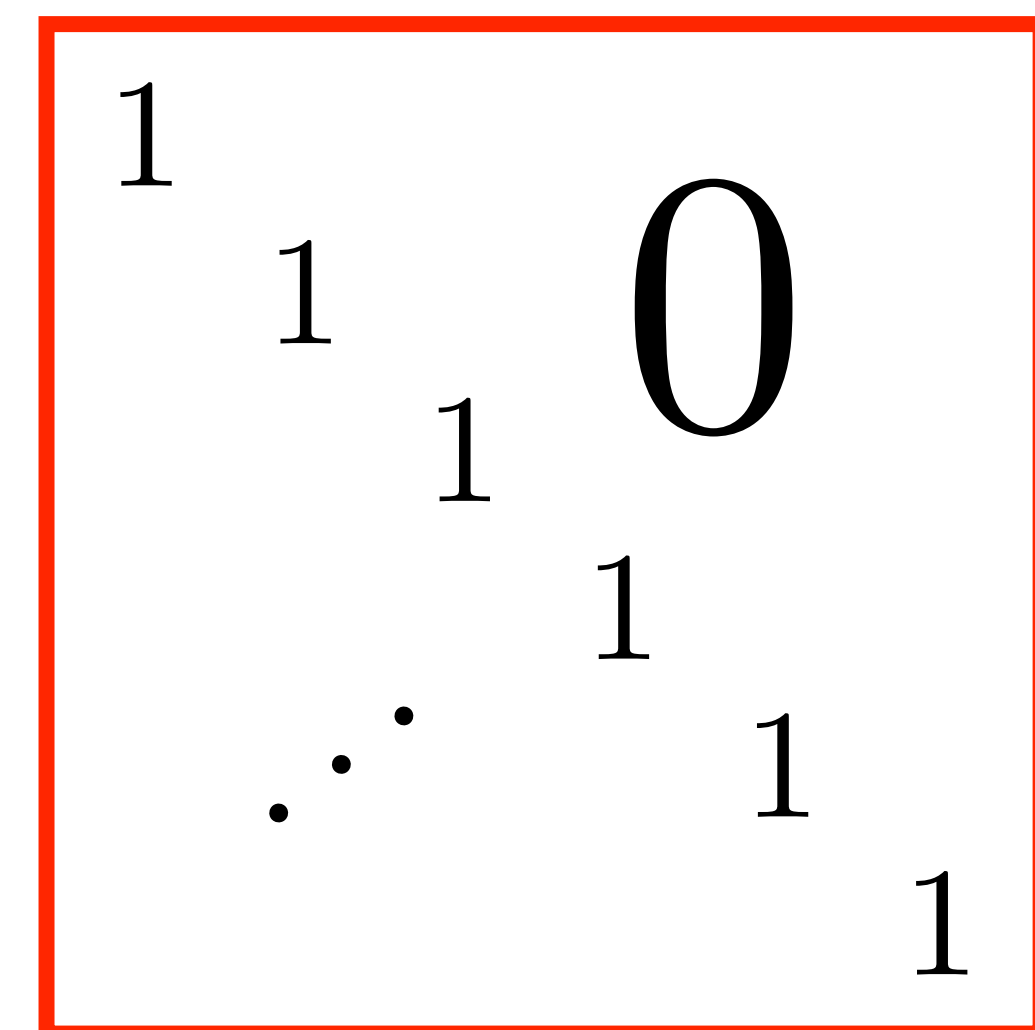


Learning with Inverse Projection



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

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



[Dinh et al. 2014]



Experiments

- Dataset: English Penn Treebank
- POS tagging

Trained and tested on whole PTB

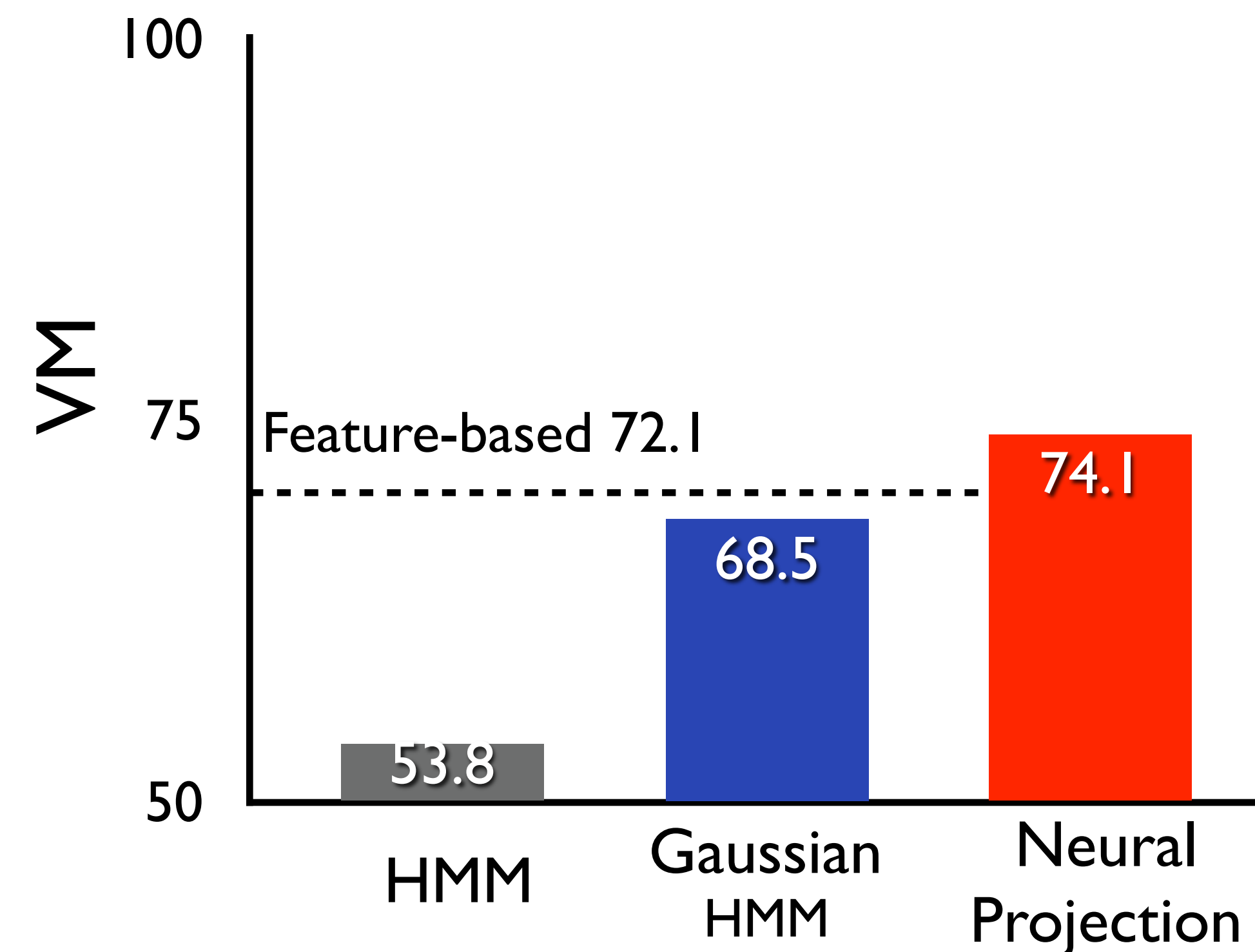
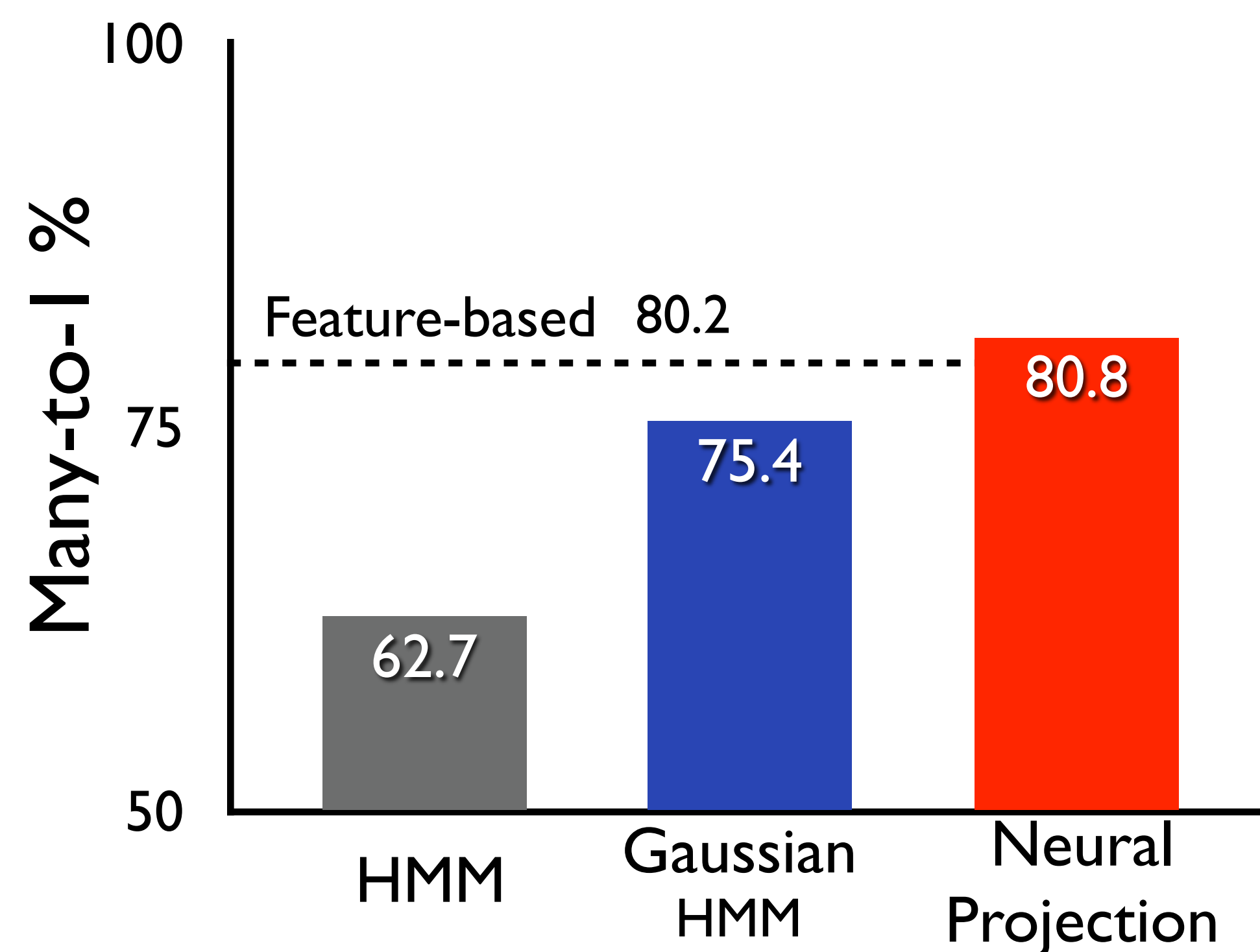
- Grammar induction

Trained on sentences of length ≤ 10 in section 2-21

Tested on sentences in section 23

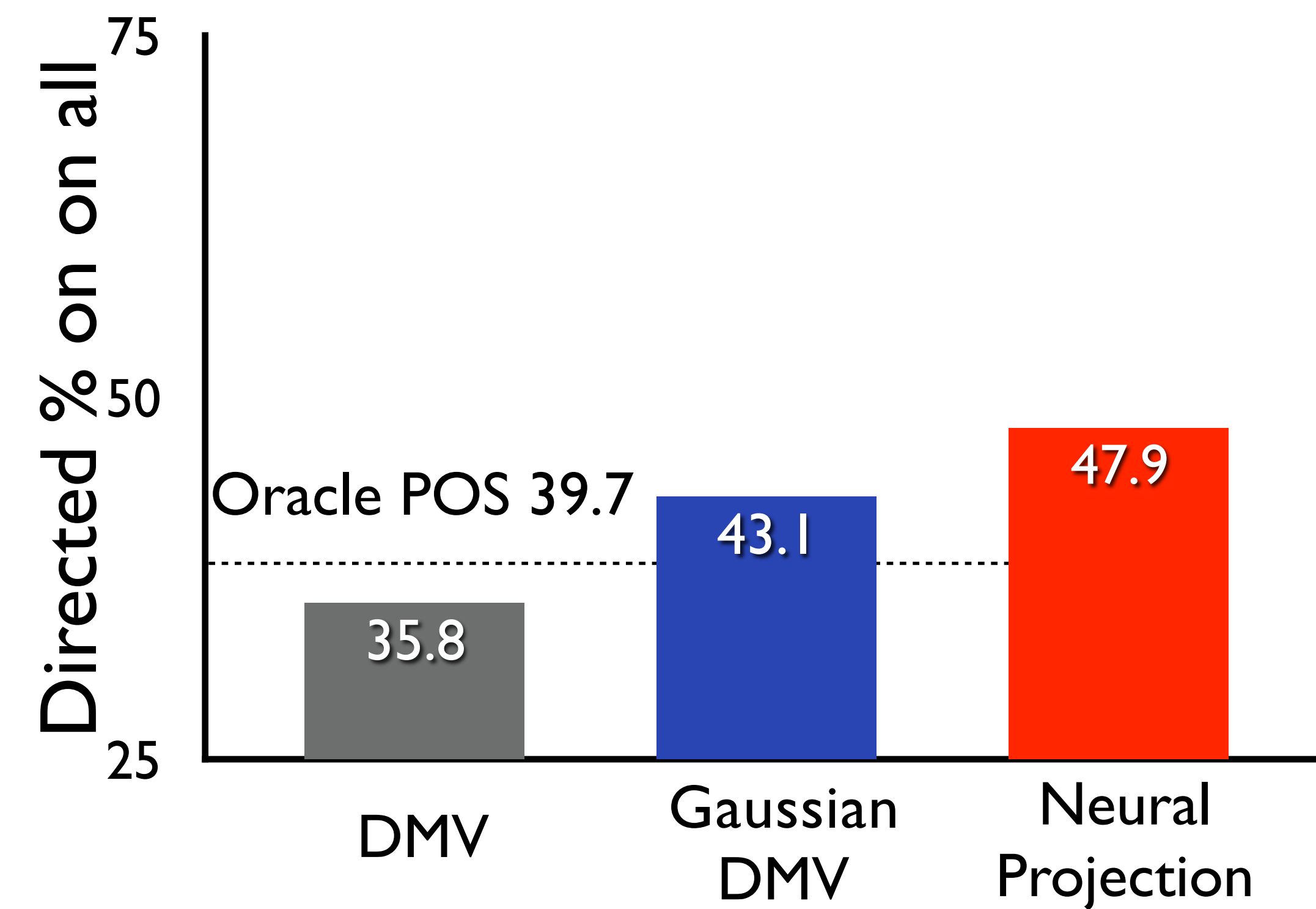
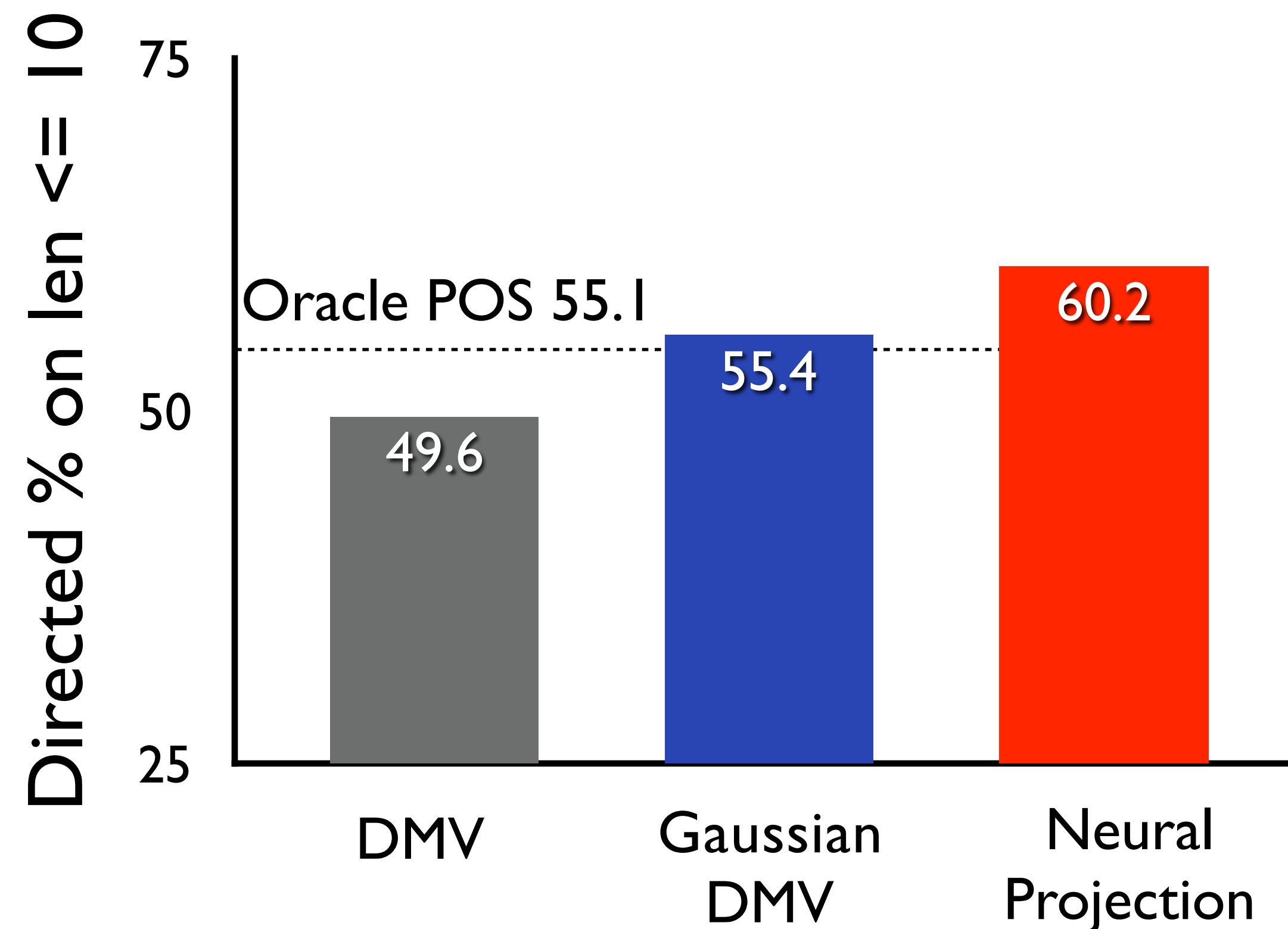


Part-of-speech Induction

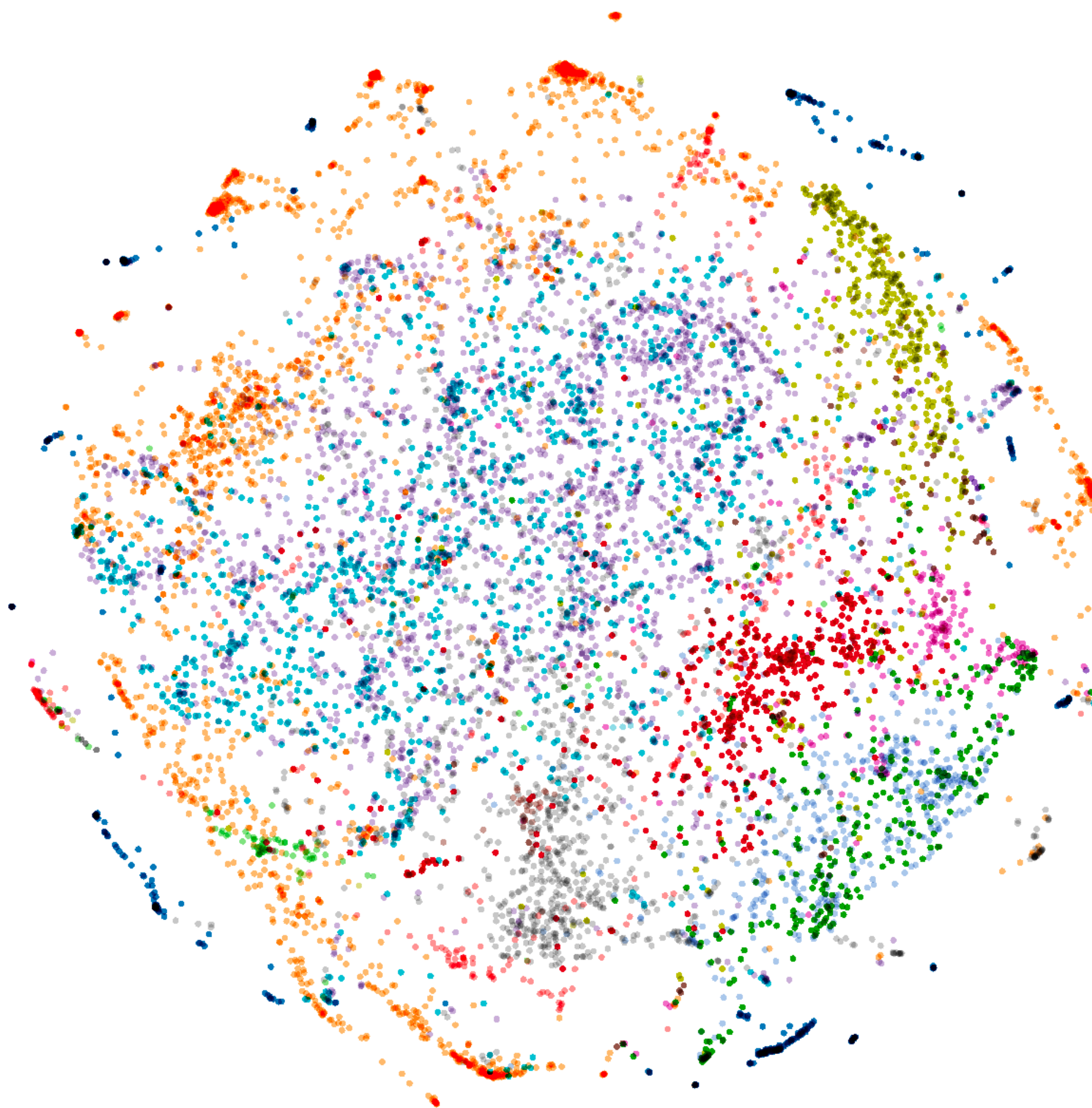


Outperform feature-based SOTA

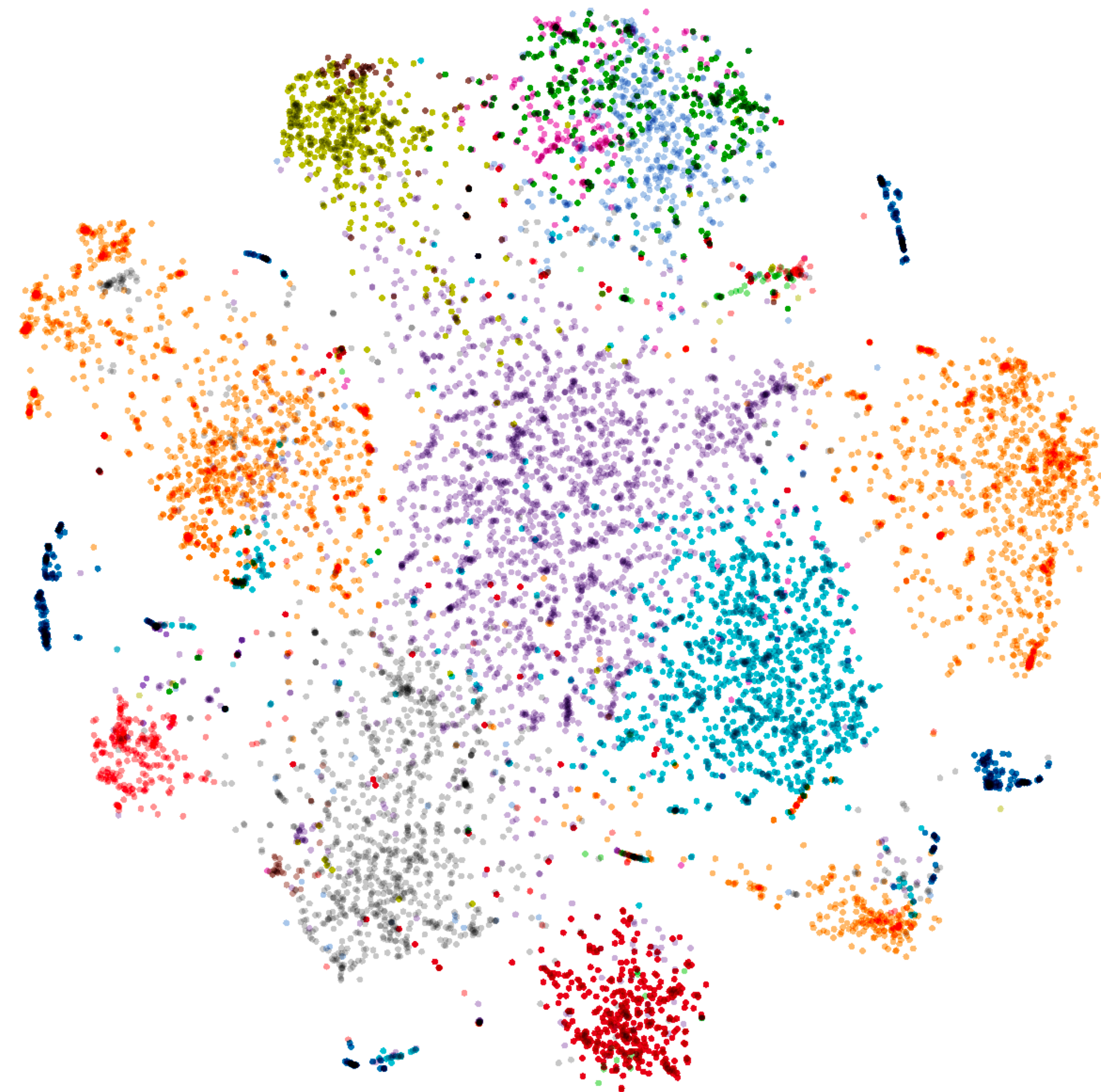
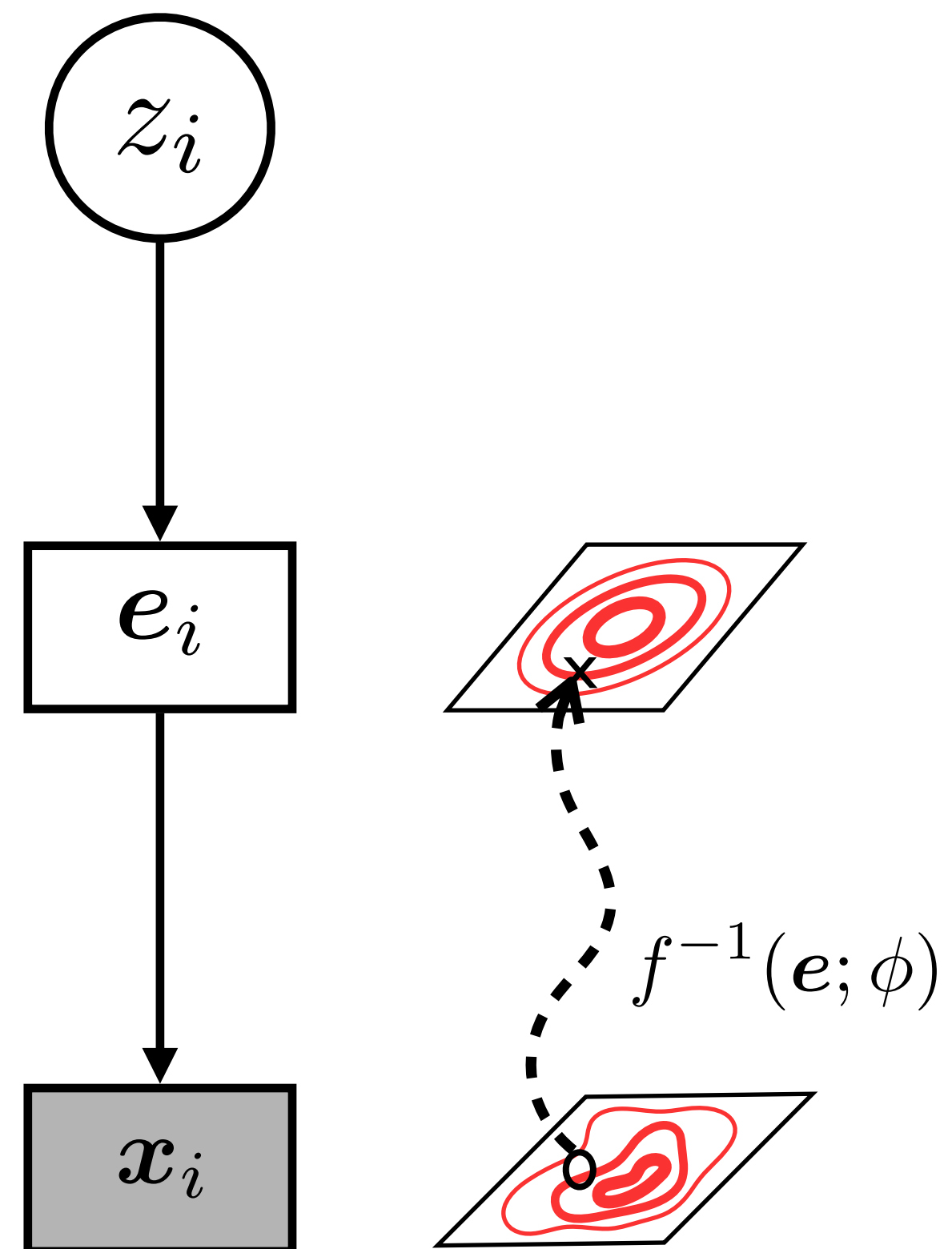
Dependency Parse Induction



Original Embedding Space

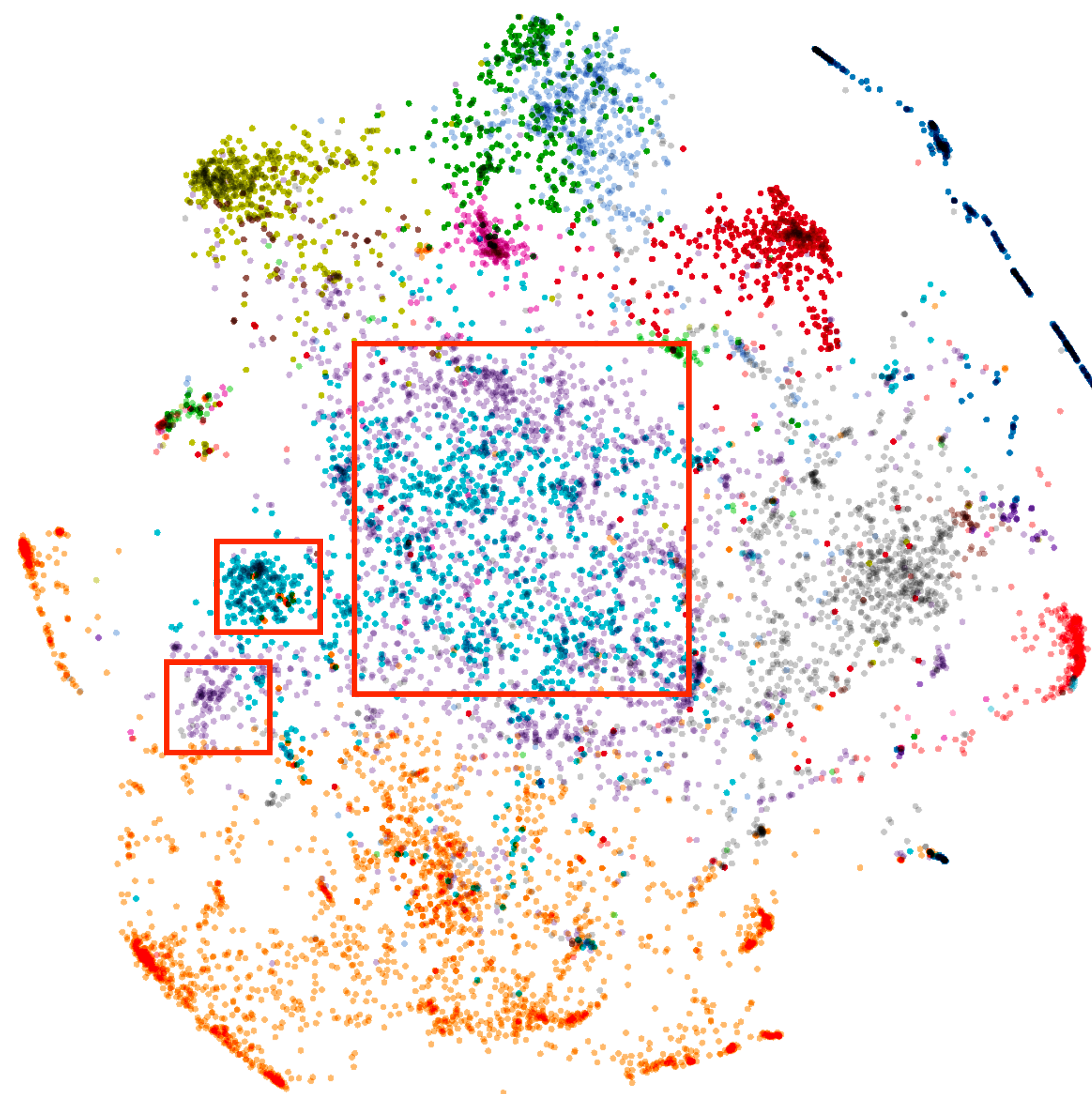


Projected Embedding Space w/ Markov Prior





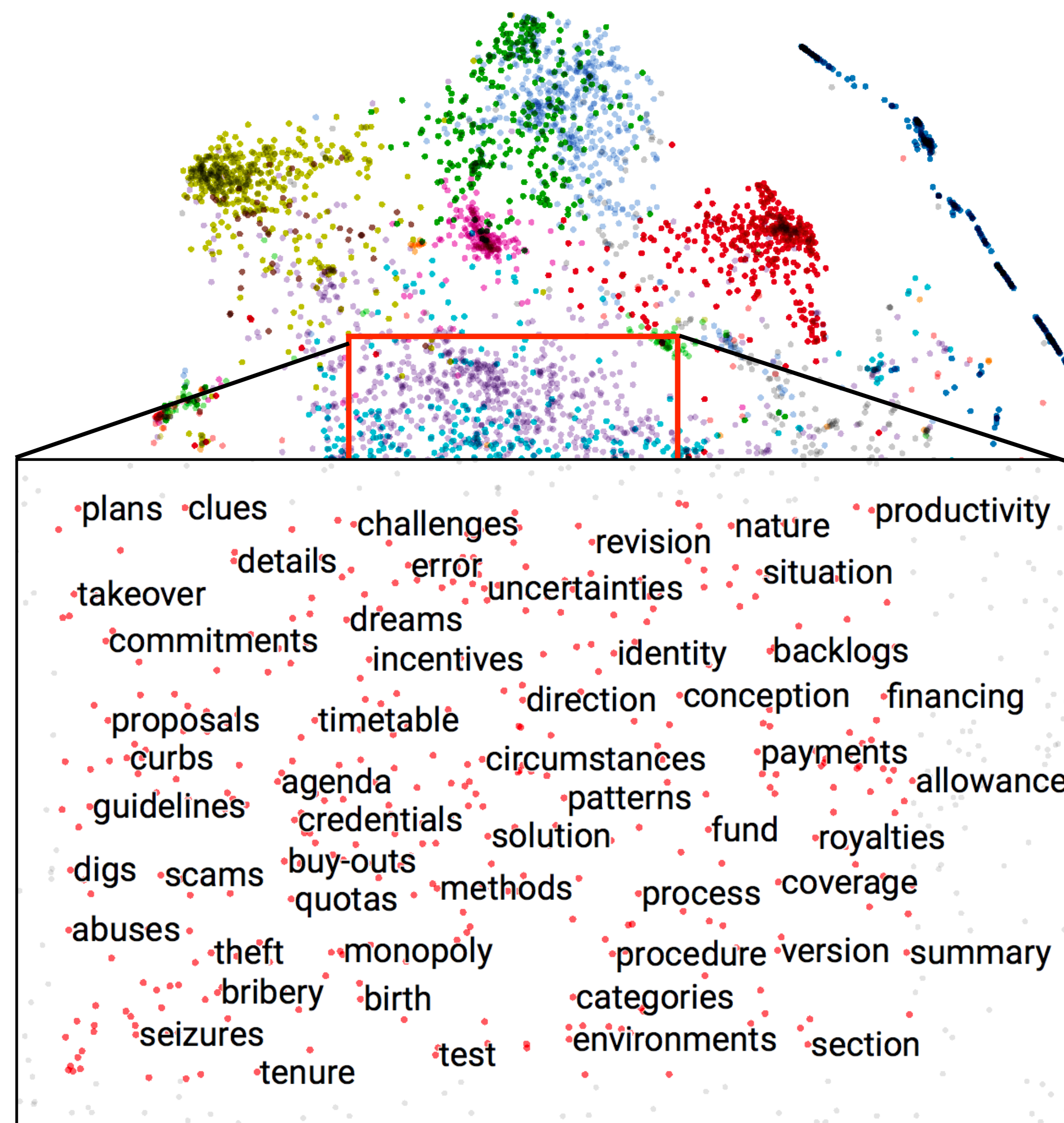
Projected Embedding Space w/ DMV Prior





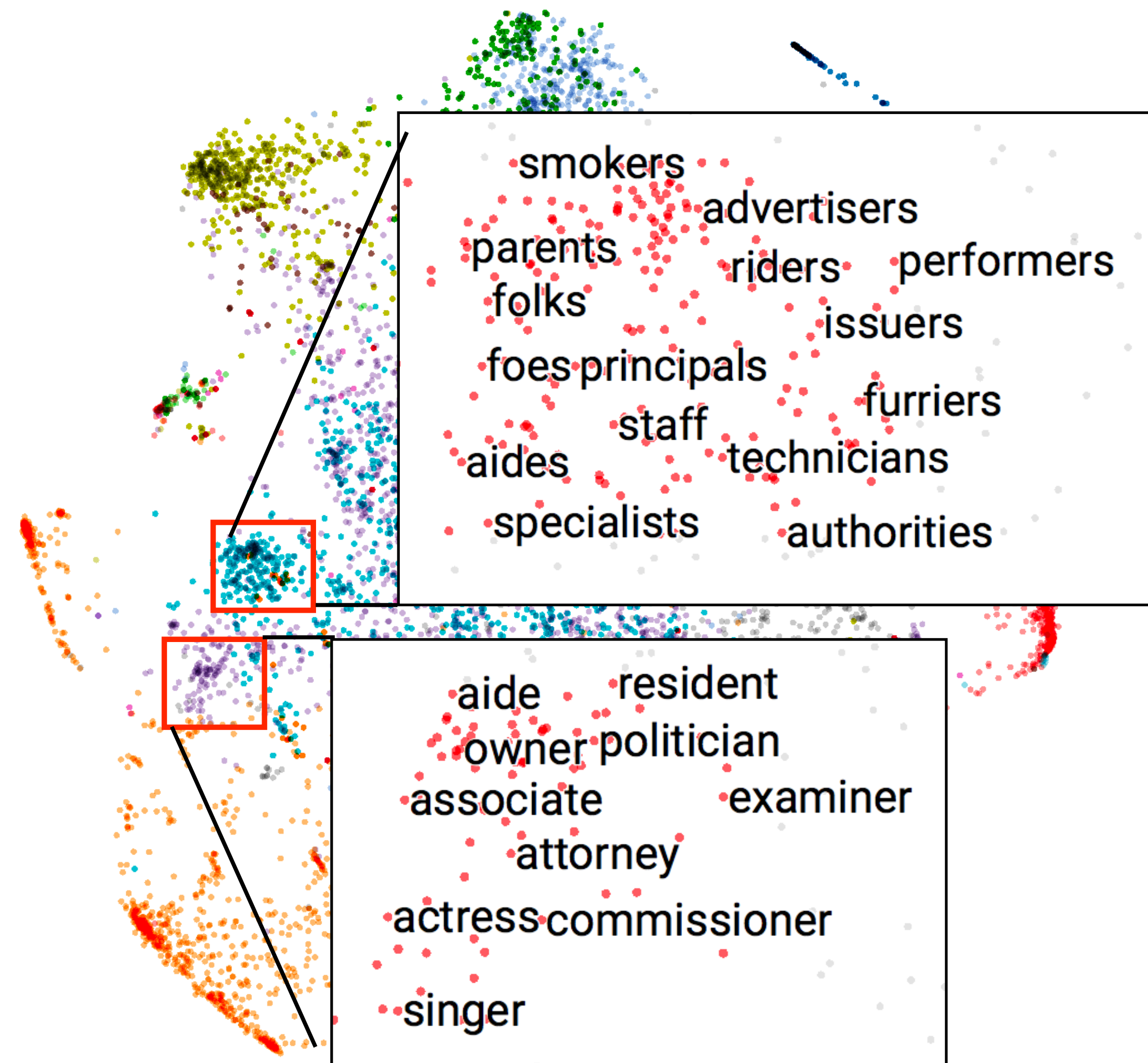
Projected Embedding Space w/ DMV Prior

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



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





Conclusion



Learning with Latent Linguistic Structure

- How can we harness the power of neural networks?
 - NN-based learning on top of latent structured representations
- How can we learn on unlabeled data?
 - Structured variational auto-encoders for semi-supervised learning
 - Structured priors and invertible transformations for unsupervised learning