

# Dialogue State Induction Using Neural Latent Variable Models

Qingkai Min<sup>1;2</sup>, Libo Qin<sup>3</sup>, Zhiyang Teng<sup>1;2</sup>, Xiao Liu<sup>4</sup> and Yue Zhang<sup>1;2</sup>

<sup>1</sup>School of Engineering, Westlake University

<sup>2</sup>Institute of Advanced Technology, Westlake Institute for Advanced Study

<sup>3</sup>Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology

<sup>4</sup>School of Computer Science and Technology, Beijing Institute of Technology

# CONTENTS

## 目录

### CHAPTER 1

Motivation

### CHAPTER 2

Method

### CHAPTER 3

Experiments

### CHAPTER 4

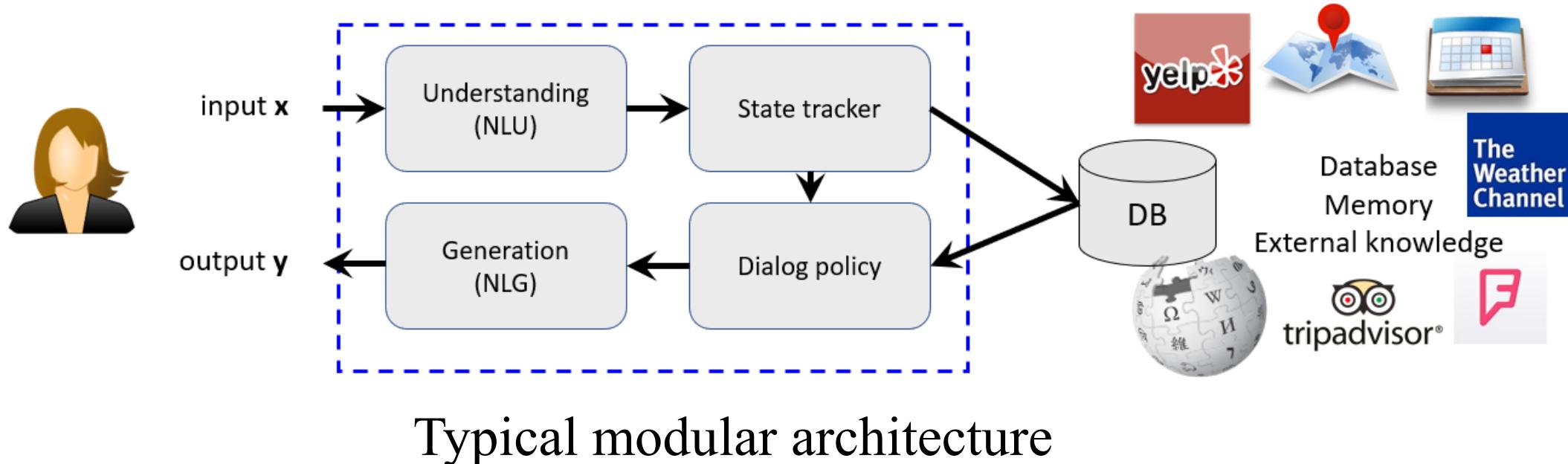
Conclusion



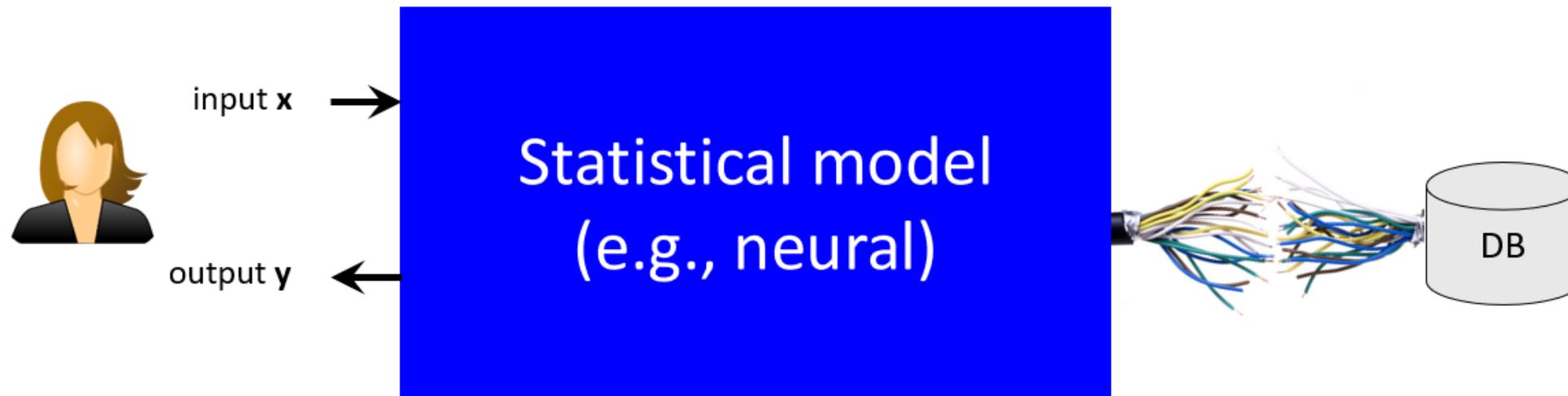
# CHAPTER 1

## Motivation

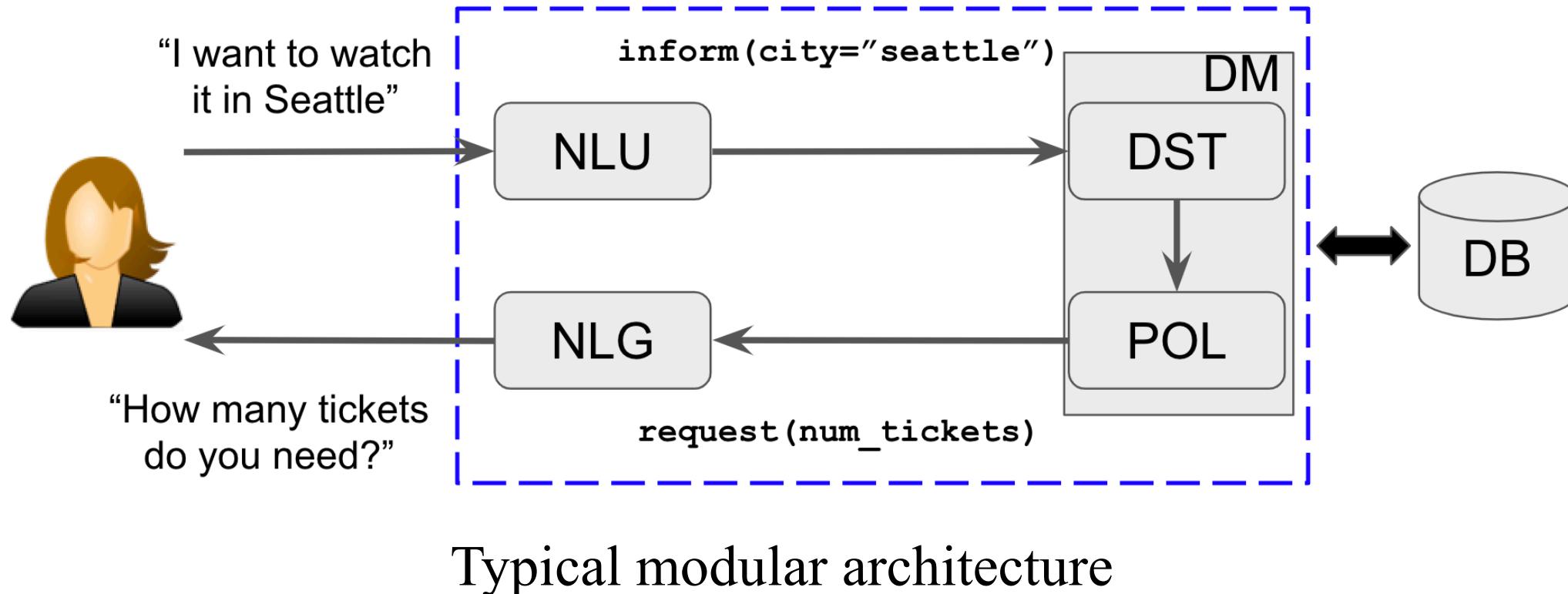
Assist user in solving a task



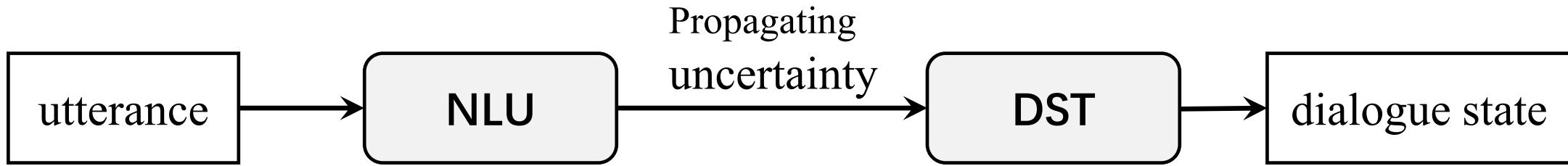
Assist user in solving a task



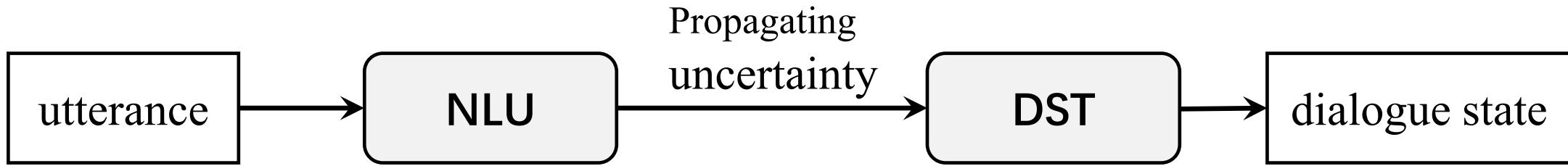
End-to-end architecture



Traditional DST:



Traditional DST:



End-to-end DST:

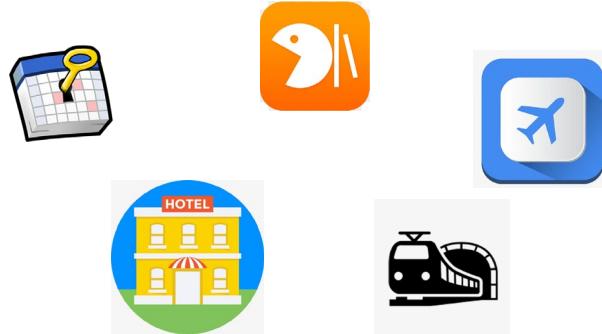


# CHAPTER 1 What is the problem?



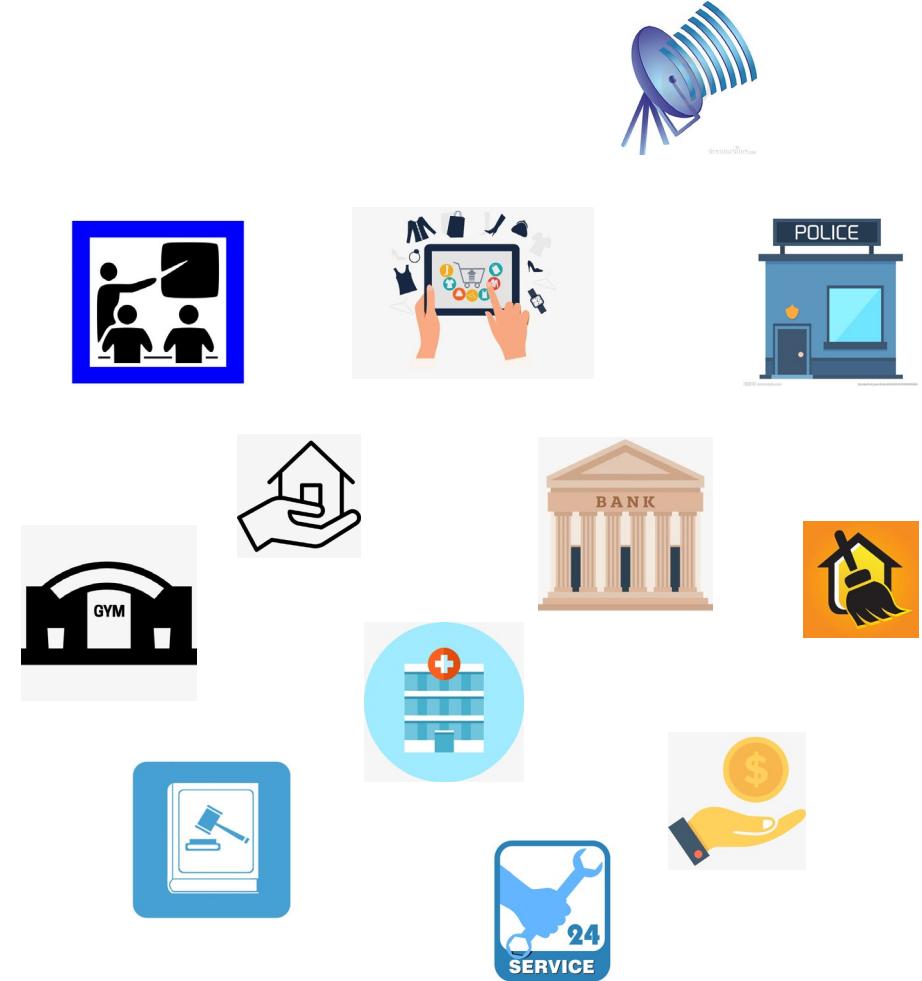
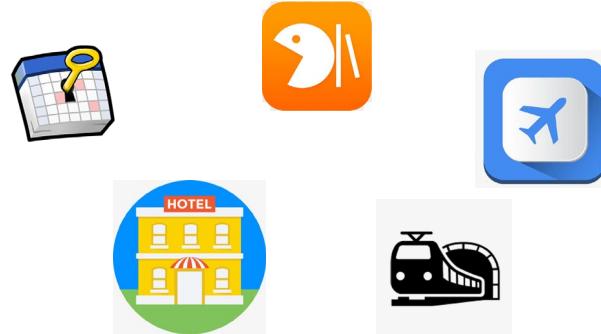
# CHAPTER 1 What is the problem?

Successful in narrow domains with large annotated datasets



# CHAPTER 1 What is the problem?

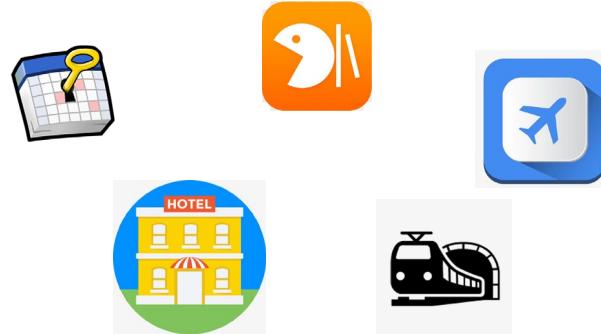
Successful in narrow domains with large annotated datasets



Limited to the domain trained on and do not afford generalization to new domains.

# CHAPTER 1 What is the problem?

Successful in narrow domains with large annotated datasets

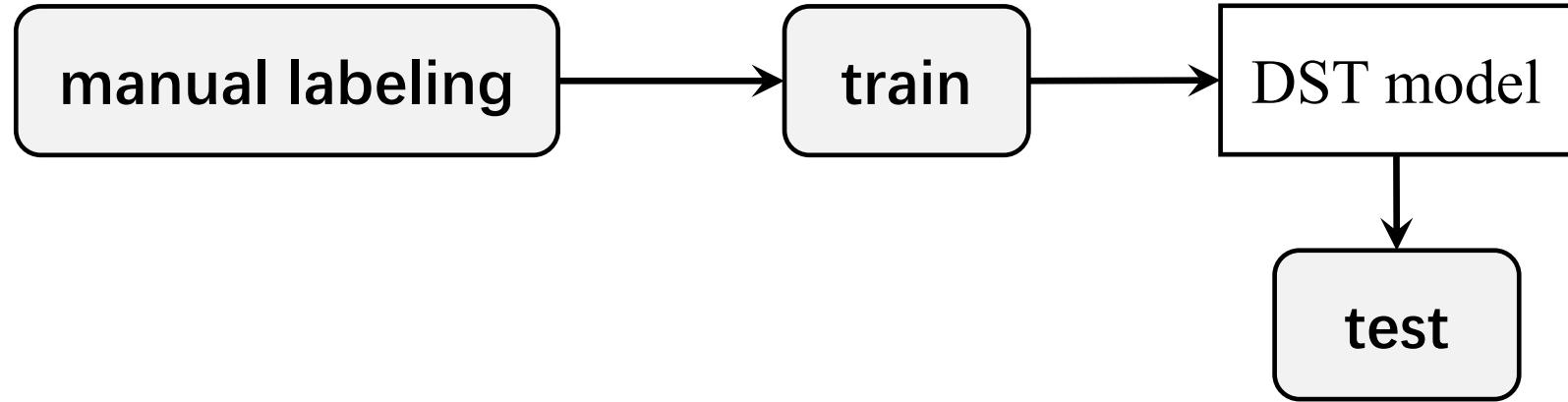


Limited to the domain trained on and do not afford generalization to new domains.



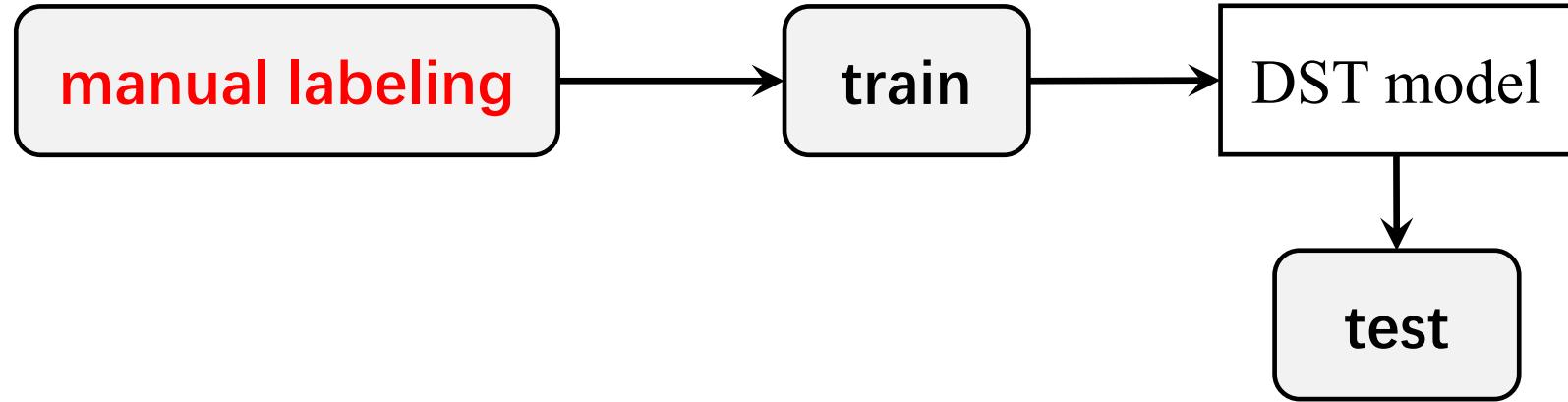
# CHAPTER 1 Limitation of DST

DST paradigm:

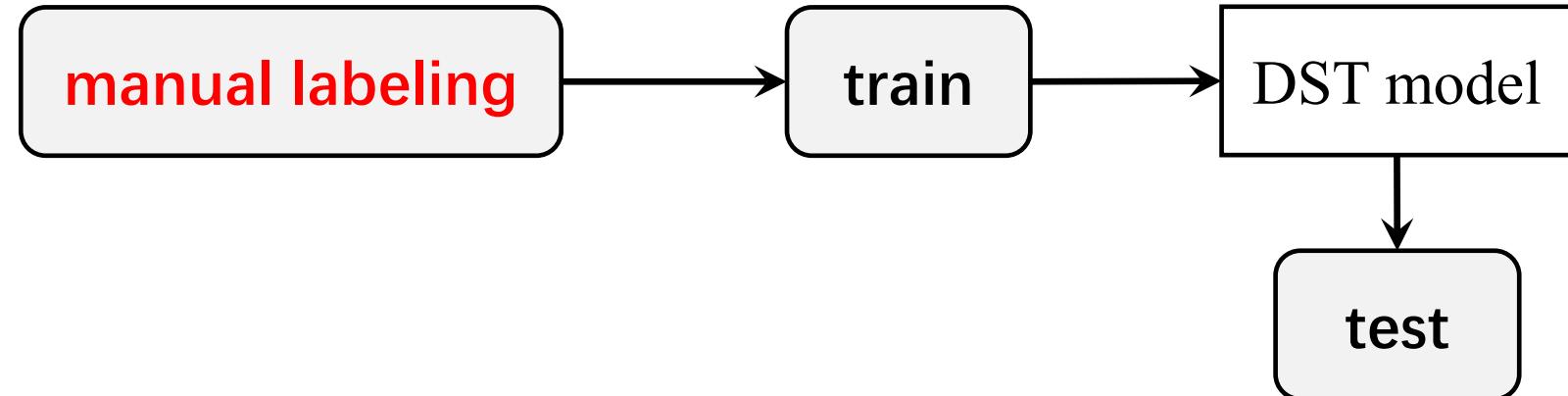


# CHAPTER 1 Limitation of DST

DST paradigm:



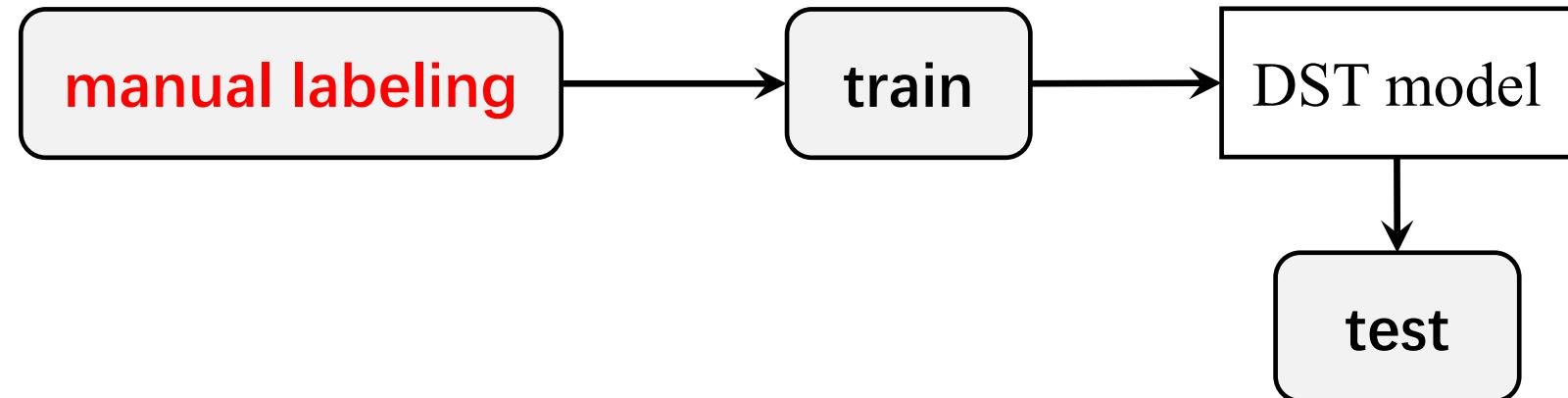
DST paradigm:



Limitations:

- Costly and slow: 8438 dialogues with 1249 workers in MultiWOZ 2.0 dataset

DST paradigm:



Limitations:

- Costly and slow: 8438 dialogues with 1249 workers in MultiWOZ 2.0 dataset

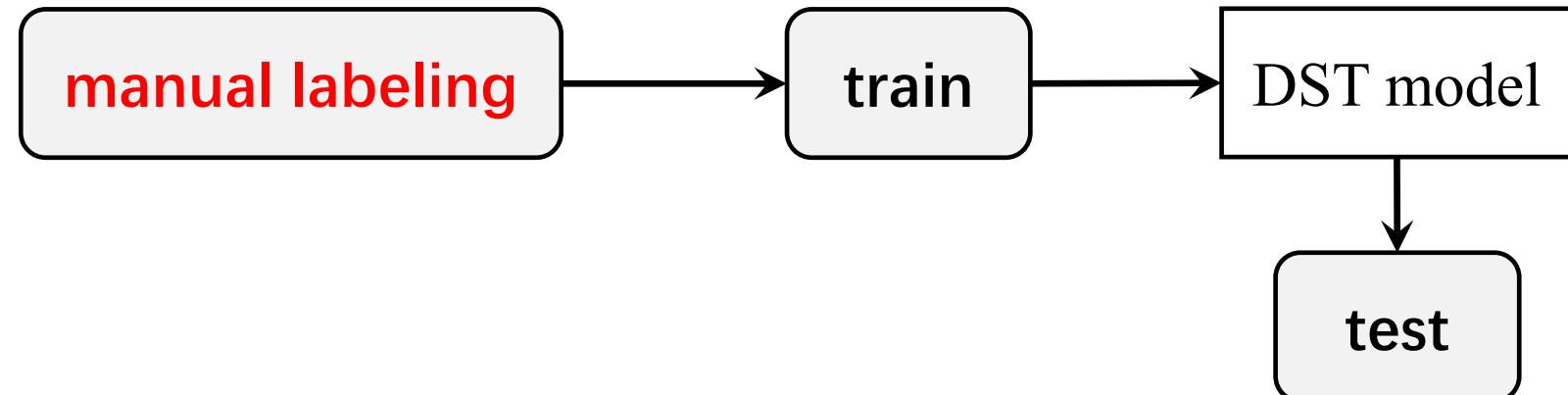
- Error-prone:

|              | Annotation errors              |
|--------------|--------------------------------|
| MultiWOZ 2.0 | around 40% [Eric et al., 2019] |
| MultiWOZ 2.1 | over 30% [Zhang et al., 2019]  |

[Eric et al., 2019] Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. arXiv, 2019.

[Zhang et al., 2019] Jian-Guo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S Yu, Richard Socher, and Caiming Xiong. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. arXiv, 2019.

DST paradigm:



Limitations:

- Costly and slow: 8438 dialogues with 1249 workers in MultiWOZ 2.0 dataset

- Error-prone:

|              | Annotation errors              | now updated to   |
|--------------|--------------------------------|--|
| MultiWOZ 2.0 | around 40% [Eric et al., 2019] |  <b>MultiWOZ 2.2</b> |
| MultiWOZ 2.1 | over 30% [Zhang et al., 2019]  |  |

[Eric et al., 2019] Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyag Gao, and Dilek Hakkani-Tur. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. arXiv, 2019.

[Zhang et al., 2019] Jian-Guo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wan, Philip S Yu, Richard Socher, and Caiming Xiong. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. arXiv, 2019.

## Definition:

What is given?

A set of customer service records  
without annotation.

**User:** I want an expensive restaurant that serves Turkish food.

**System:** Anatolia serves Turkish food.

**User:** What is the area?

## Definition:

What is given?

A set of customer service records  
**without annotation.**

What is the target?

Automatically discover information that the user is looking for at each turn.

User: I want an expensive restaurant that serves Turkish food.  
System: Anatolia serves Turkish food.  
User: What is the area?



inform(price=expensive, food=Turkish)



inform(price=expensive,  
food=Turkish) ; request(area)

# CHAPTER 1 Dialogue State Induction vs DST

**User:** I want an expensive restaurant that serves Turkish food.

**System:** Anatolia serves Turkish food.

**User:** What is the area?

**User:** I want an expensive restaurant that serves Turkish food.

Manual labeling:

```
inform(price=expensive,  
       food=Turkish)
```

**System:** Anatolia serves Turkish food.

**User:** What is the area?

Manual labeling:

```
inform(price=expensive,  
       food=Turkish); request(area)
```

**User:** I want an expensive restaurant that serves Turkish food.

Manual labeling:

```
inform(price=expensive,  
       food=Turkish)
```

**System:** Anatolia serves Turkish food.

**User:** What is the area?

Manual labeling:

```
inform(price=expensive,  
       food=Turkish); request(area)
```

**Ontology(optional):**

price: [cheap, expensive, moderate, ...]

food: [Turkish, Italian, polish, ...]

area: [south, north, center, ...]

**User:** I want an expensive restaurant that serves Turkish food.

Manual labeling:

```
inform(price=expensive,  
       food=Turkish)
```

**System:** Anatolia serves Turkish food.

**User:** What is the area?

Manual labeling:

```
inform(price=expensive,  
       food=Turkish); request(area)
```

**Ontology(optional):**

price: [cheap, expensive, moderate, ...]

food: [Turkish, Italian, polish, ...]

area: [south, north, center, ...]

DST

**Turn 1:**

```
inform(price=expensive,  
       food=Turkish)
```

**Turn 2:**

```
inform(price=expensive,  
       food=Turkish); request(area)
```

**User:** I want an expensive restaurant that serves Turkish food.

Manual labeling:

```
inform(price=expensive,  
       food=Turkish)
```

**System:** Anatolia serves Turkish food.

**User:** What is the area?

Manual labeling:

```
inform(price=expensive,  
       food=Turkish); request(area)
```

**Ontology(optional):**

price: [cheap, expensive, moderate, ...]

food: [Turkish, Italian, polish, ...]

area: [south, north, center, ...]



**Turn 1:**

```
inform(price=expensive,  
       food=Turkish)
```

**Turn 2:**

```
inform(price=expensive,  
       food=Turkish); request(area)
```

**User:** I want an expensive restaurant that serves Turkish food.

**System:** Anatolia serves Turkish food.

**User:** What is the area?

**User:** I want an expensive restaurant that serves Turkish food.

Manual labeling:

```
inform(price=expensive,  
       food=Turkish)
```

**System:** Anatolia serves Turkish food.

**User:** What is the area?

Manual labeling:

```
inform(price=expensive,  
       food=Turkish); request(area)
```

**Ontology(optional):**

price: [cheap, expensive, moderate, ...]

food: [Turkish, Italian, polish, ...]

area: [south, north, center, ...]



**Turn 1:**

```
inform(price=expensive,  
       food=Turkish)
```

**Turn 2:**

```
inform(price=expensive,  
       food=Turkish); request(area)
```

**User:** I want an expensive restaurant that serves Turkish food.

Manual labeling:

```
inform(price=expensive,  
       food=Turkish)
```

**System:** Anatolia serves Turkish food.

**User:** What is the area?

Manual labeling:

```
inform(price=expensive,  
       food=Turkish); request(area)
```

**Ontology(optional):**

price: [cheap, expensive, moderate, ...]

food: [Turkish, Italian, polish, ...]

area: [south, north, center, ...]

**User:** I want an expensive restaurant that serves Turkish food.

Manual labeling:

```
inform(price=expensive,  
       food=Turkish)
```

**System:** Anatolia serves Turkish food.

**User:** What is the area?

Manual labeling:

```
inform(price=expensive,  
       food=Turkish); request(area)
```

**Ontology(optional):**

price: [cheap, expensive, moderate, ...]

food: [Turkish, Italian, polish, ...]

area: [south, north, center, ...]



**Turn 1:**

```
inform(price=expensive,  
       food=Turkish)
```

**Turn 2:**

```
inform(price=expensive,  
       food=Turkish); request(area)
```

**User:** I want an expensive restaurant that serves Turkish food.

Manual labeling:

```
inform(price=expensive,  
       food=Turkish)
```

**System:** Anatolia serves Turkish food.

**User:** What is the area?

Manual labeling:

```
inform(price=expensive,  
       food=Turkish); request(area)
```

**Ontology(optional):**

price: [cheap, expensive, moderate, ...]

food: [Turkish, Italian, polish, ...]

area: [south, north, center, ...]



**Turn 1:**

```
inform(price=expensive,  
       food=Turkish)
```

**Turn 2:**

```
inform(price=expensive,  
       food=Turkish); request(area)
```

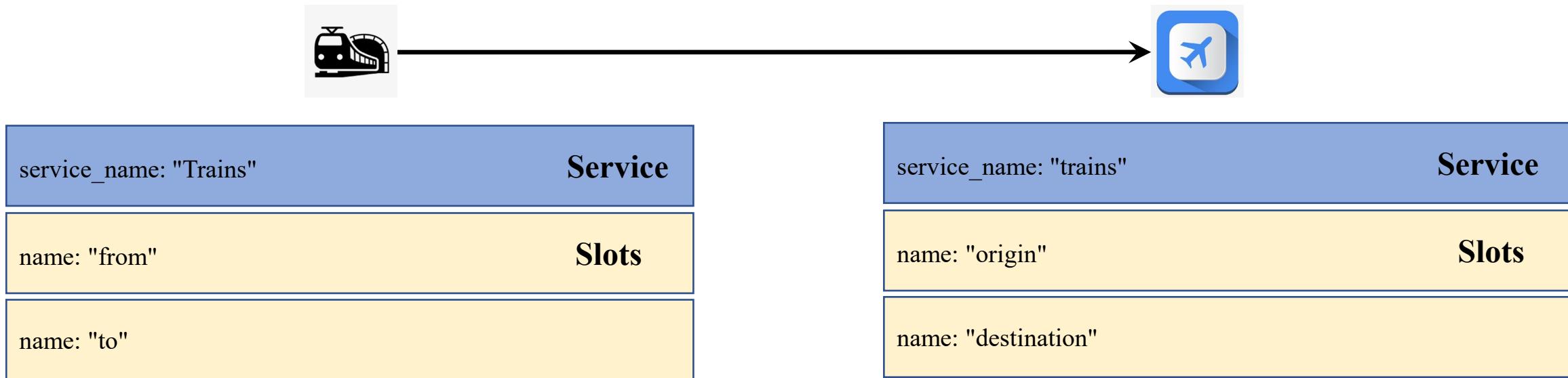
**Zero-shot DST:** support unseen domains (services)

**Zero-shot DST:** support unseen domains (services)

**Motivation:** Different domains (services) with similar schemas

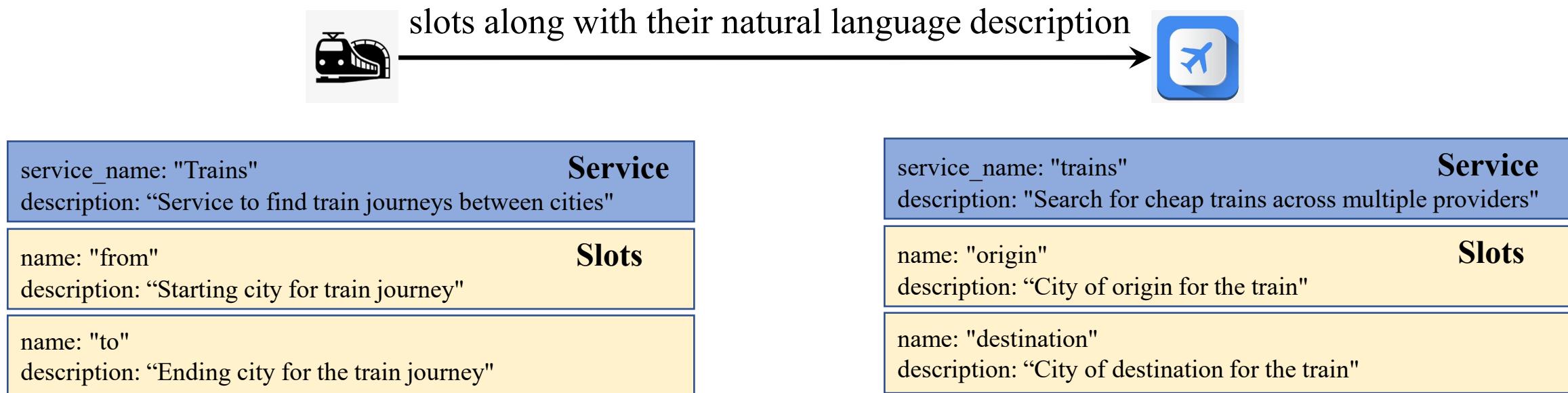
**Zero-shot DST:** support unseen domains (services)

**Motivation:** Different domains (services) with similar schemas



**Zero-shot DST:** support unseen domains (services)

**Motivation:** Different domains (services) with similar schemas



Example from the SGD dataset [Rastogi et al., 2019].

# CHAPTER 1 DSI VS zero-shot DST

| Service | service_name: "Trains"<br>description: "Service to find train journeys between cities" |
|---------|--|
| Slots   | name: "from"<br>description: "Starting city for train journey"                         |
| Slots   | name: "to"<br>description: "Ending city for the train journey"                         |

| Service | service_name: "trains"<br>description: "Search for cheap trains across multiple providers" |
|---------|--|
| Slots   | name: "origin"<br>description: "City of origin for the train"                              |
| Slots   | name: "destination"<br>description: "City of destination for the train"                    |

# CHAPTER 1 DSI VS zero-shot DST

| Service | service_name: "Trains"<br>description: "Service to find train journeys between cities" |
|---------|--|
| Slots   | name: "from"<br>description: "Starting city for train journey"                         |
| Slots   | name: "to"<br>description: "Ending city for the train journey"                         |

| Service | service_name: "trains"<br>description: "Search for cheap trains across multiple providers" |
|---------|--|
| Slots   | name: "origin"<br>description: "City of origin for the train"                              |
| Slots   | name: "destination"<br>description: "City of destination for the train"                    |

Zero-shot DST Limitations:

# CHAPTER 1 DSI VS zero-shot DST

| Service | service_name: "Trains"<br>description: "Service to find train journeys between cities" |
|---------|--|
| Slots   | name: "from"<br>description: "Starting city for train journey"                         |
| Slots   | name: "to"<br>description: "Ending city for the train journey"                         |

| Service | service_name: "trains"<br>description: "Search for cheap trains across multiple providers" |
|---------|--|
| Slots   | name: "origin"<br>description: "City of origin for the train"                              |
| Slots   | name: "destination"<br>description: "City of destination for the train"                    |

## Zero-shot DST Limitations:

- High qualified (**consistent**) human annotation

# CHAPTER 1 DSI VS zero-shot DST

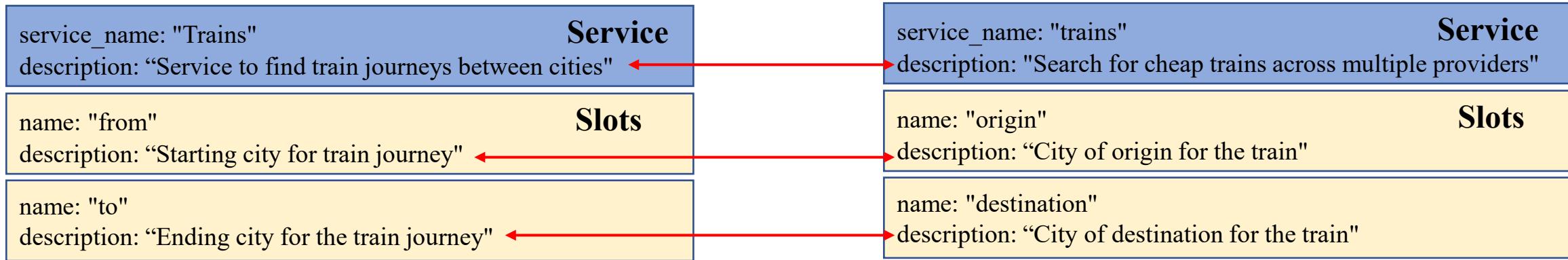
| Service | service_name: "Trains"<br>description: "Service to find train journeys between cities" |
|---------|--|
| Slots   | name: "from"<br>description: "Starting city for train journey"                         |
| Slots   | name: "to"<br>description: "Ending city for the train journey"                         |

| Service | service_name: "trains"<br>description: "Search for cheap trains across multiple providers" |
|---------|--|
| Slots   | name: "origin"<br>description: "City of origin for the train"                              |
| Slots   | name: "destination"<br>description: "City of destination for the train"                    |

## Zero-shot DST Limitations:

- High qualified (**consistent**) human annotation

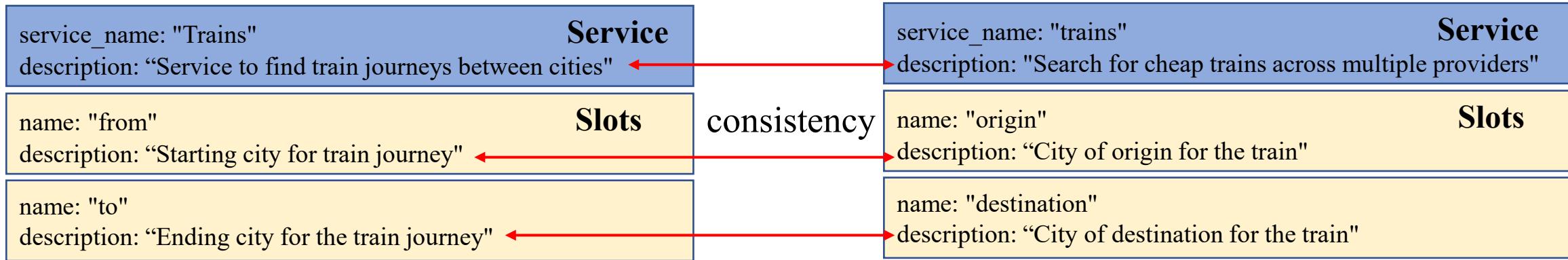
# CHAPTER 1 DSI VS zero-shot DST



## Zero-shot DST Limitations:

- High qualified (**consistent**) human annotation

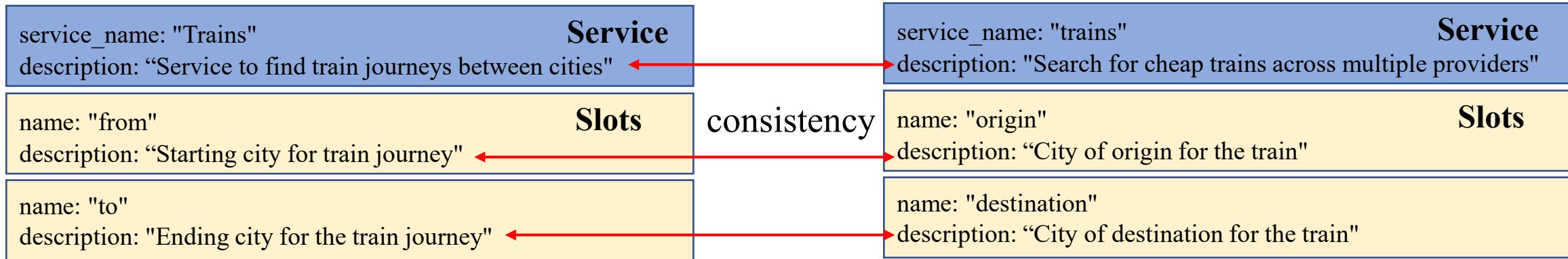
# CHAPTER 1 DSI VS zero-shot DST



## Zero-shot DST Limitations:

- High qualified (**consistent**) human annotation

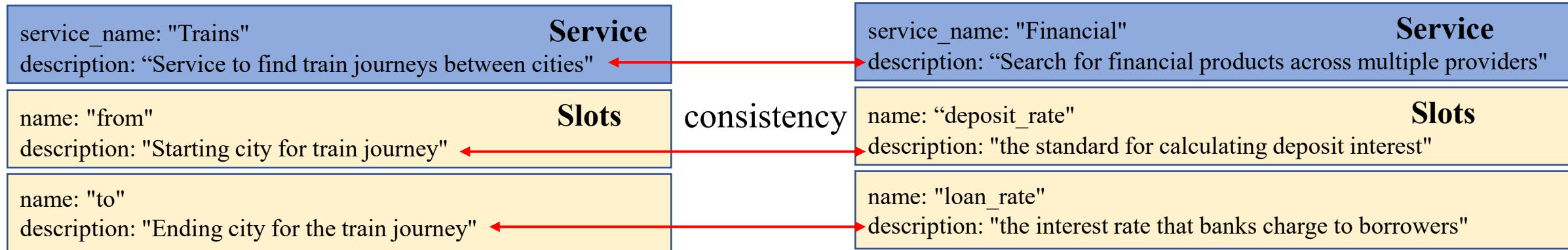
# CHAPTER 1 DSI VS zero-shot DST



## Zero-shot DST Limitations:

- High qualified (**consistent**) human annotation
- Transfer to **distant** domain (service)

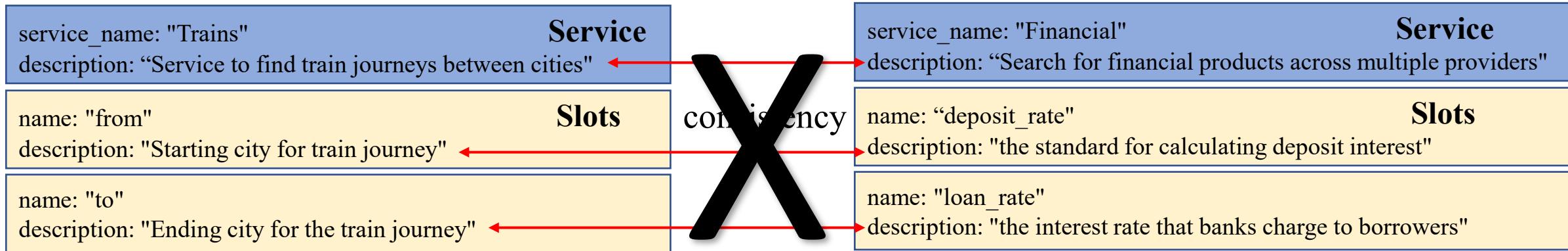
# CHAPTER 1 DSI VS zero-shot DST



## Zero-shot DST Limitations:

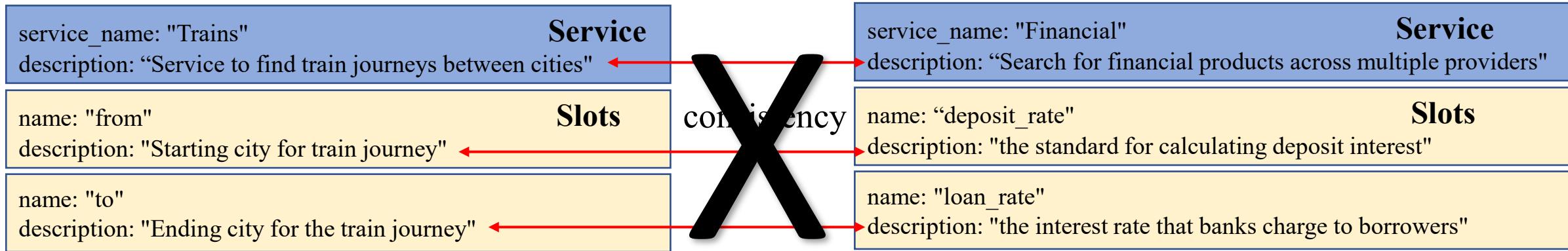
- High qualified (**consistent**) human annotation
- Transfer to **distant** domain (service)

# CHAPTER 1 DSI VS zero-shot DST



## Zero-shot DST Limitations:

- High qualified (**consistent**) human annotation
- Transfer to **distant** domain (service)



## Zero-shot DST Limitations:

- High qualified (**consistent**) human annotation
- Transfer to **distant** domain (service)



## DSI features:

- Release human burden
- Data-driven: automatically discover



# CHAPTER 2

## Method

## CHAPTER 2 How we solve DSI?

Two steps:

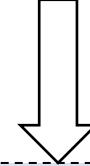
**Utterance:** I need to take a train out of Chicago,  
I will be leaving Dallas on Wednesday.

## CHAPTER 2 How we solve DSI?

### Two steps:

- Candidates (values) extraction  
(POS tag, NER, coreference)

**Utterance:** I need to take a train out of Chicago,  
I will be leaving Dallas on Wednesday.



train, Chicago, Dallas, Wednesday

## CHAPTER 2 How we solve DSI?

### Two steps:

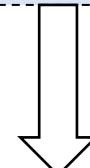
- Candidates (values) extraction (POS tag, NER, coreference)
- Slot assignment: two neural latent variable models (*DSI-base* and *DSI-GM*)

~~train=None~~

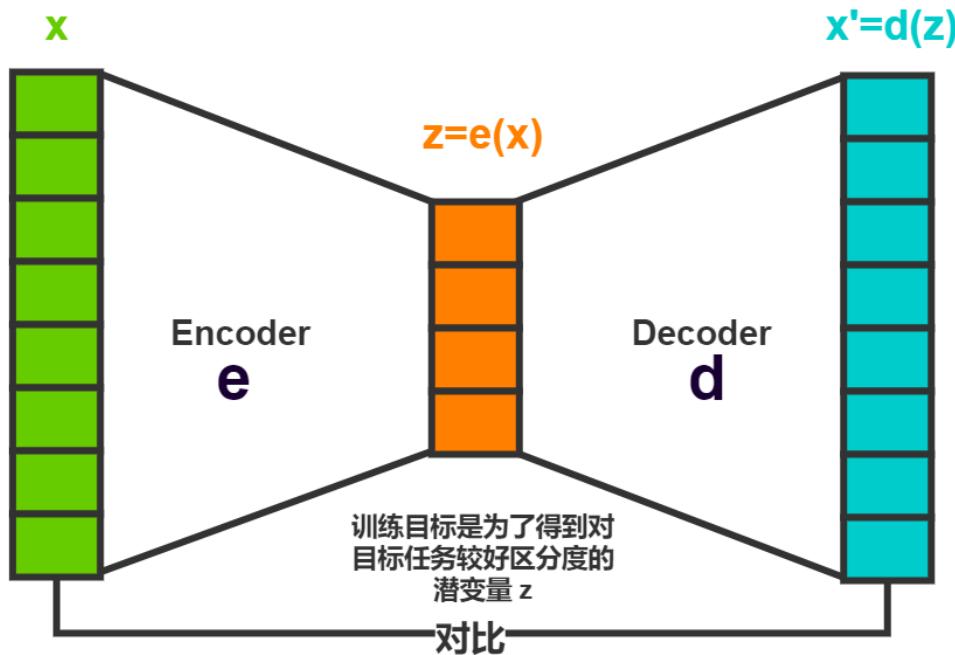
**Utterance:** I need to take a train out of Chicago,  
I will be leaving Dallas on Wednesday.



train, Chicago, Dallas, Wednesday

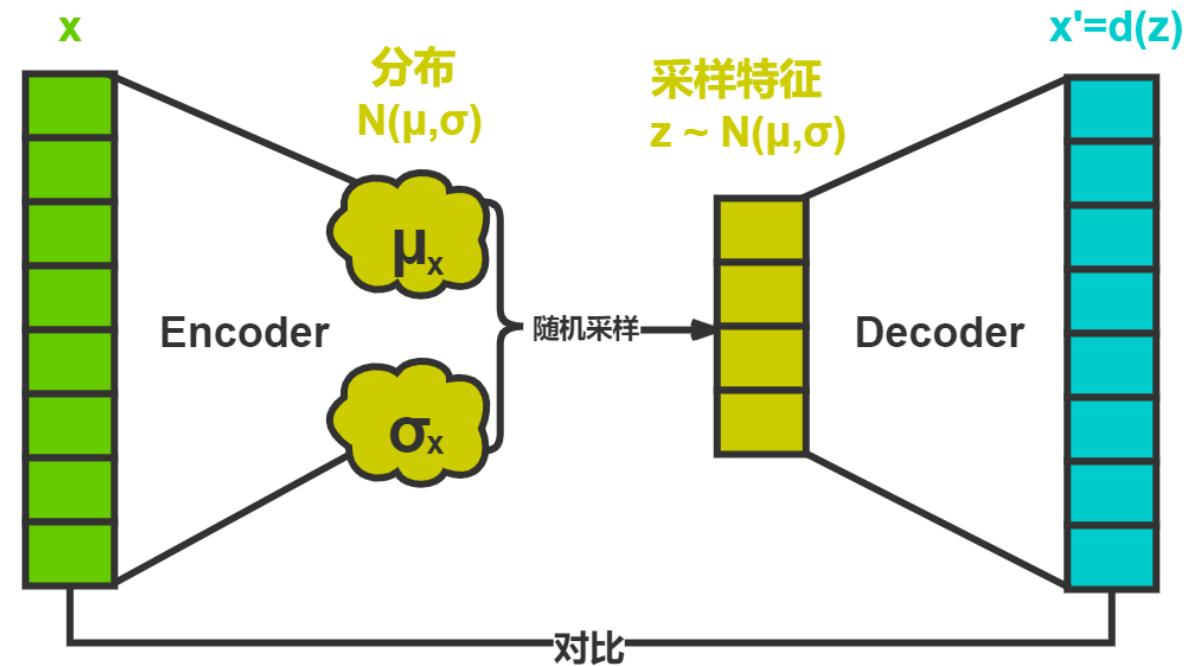


Inform{train-departure=Chicago,  
train-destination=Dallas,  
train-leave at=Wednesday}



$$\text{loss} = ||x - x'||^2$$

AutoEncoder



$$\text{loss} = ||x - x'|| + \text{KL}(N(\mu, \sigma), N(0, 1))$$

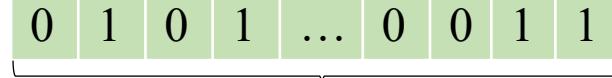
Variational AutoEncoder

I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.

Encoder

I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.

one-hot

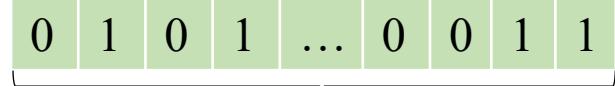


vocab length (all candidates)

Encoder

I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.

one-hot

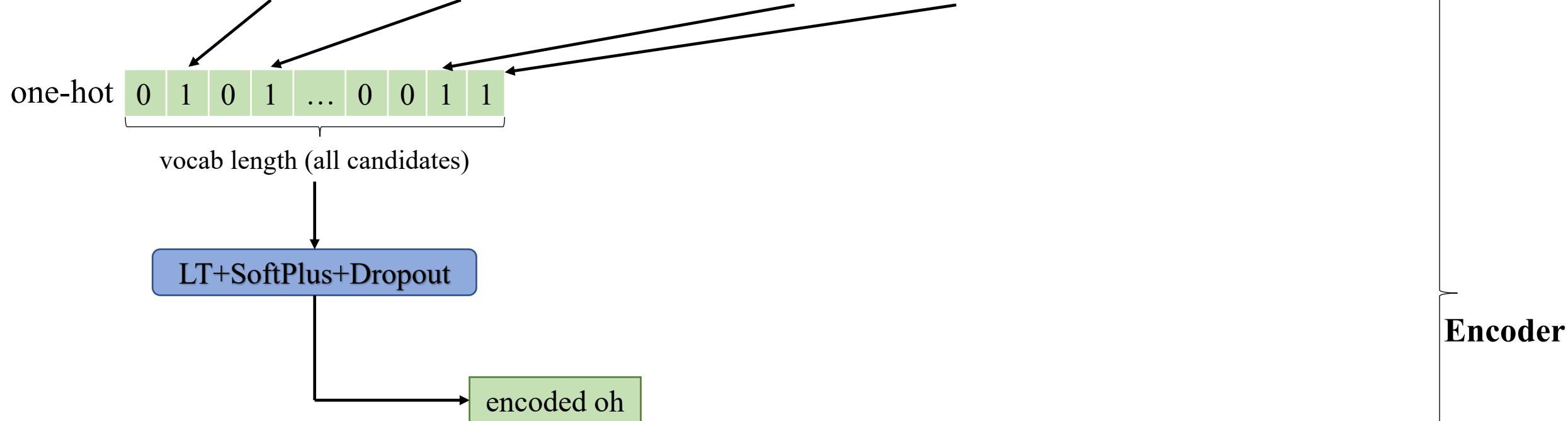


vocab length (all candidates)

LT+SoftPlus+Dropout

Encoder

I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.

one-hot



vocab length (all candidates)

LT+SoftPlus+Dropout

encoded oh

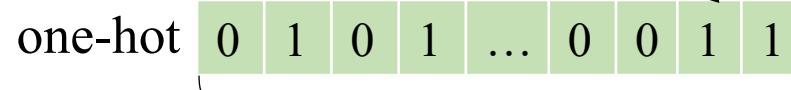
contextualized  
embedding

Pre-trained ELMo

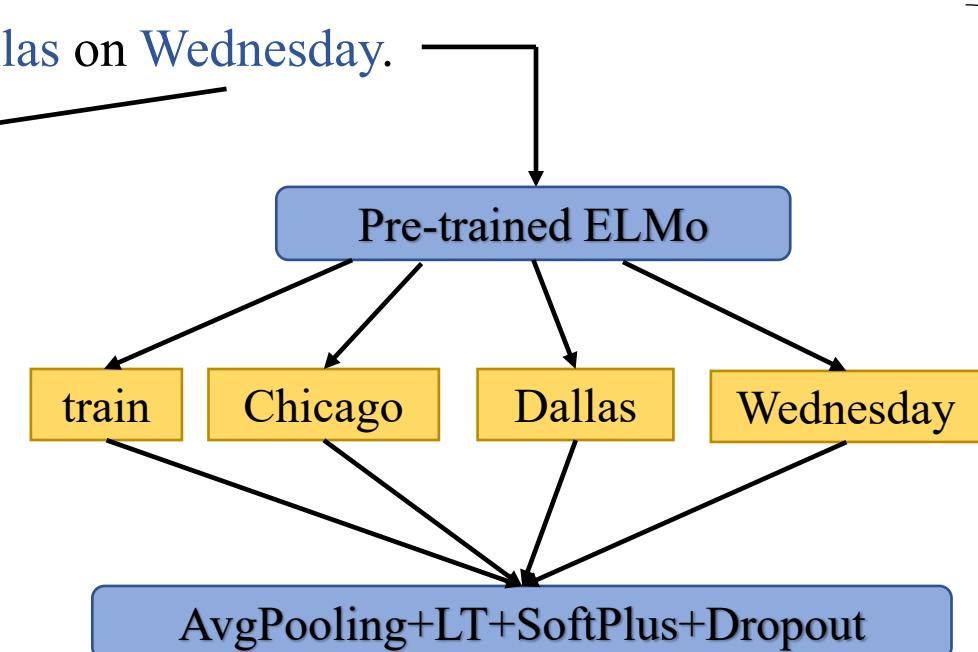


Encoder

I need to take a **train** out of **Chicago**, I will be leaving **Dallas** on **Wednesday**.



contextualized  
embedding



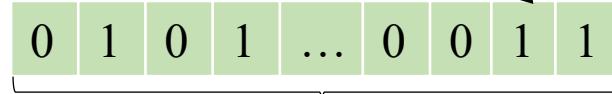
LT+SoftPlus+Dropout

encoded oh

Encoder

I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.

one-hot



vocab length (all candidates)

LT+SoftPlus+Dropout

contextualized  
embedding

Pre-trained ELMo

train

Chicago

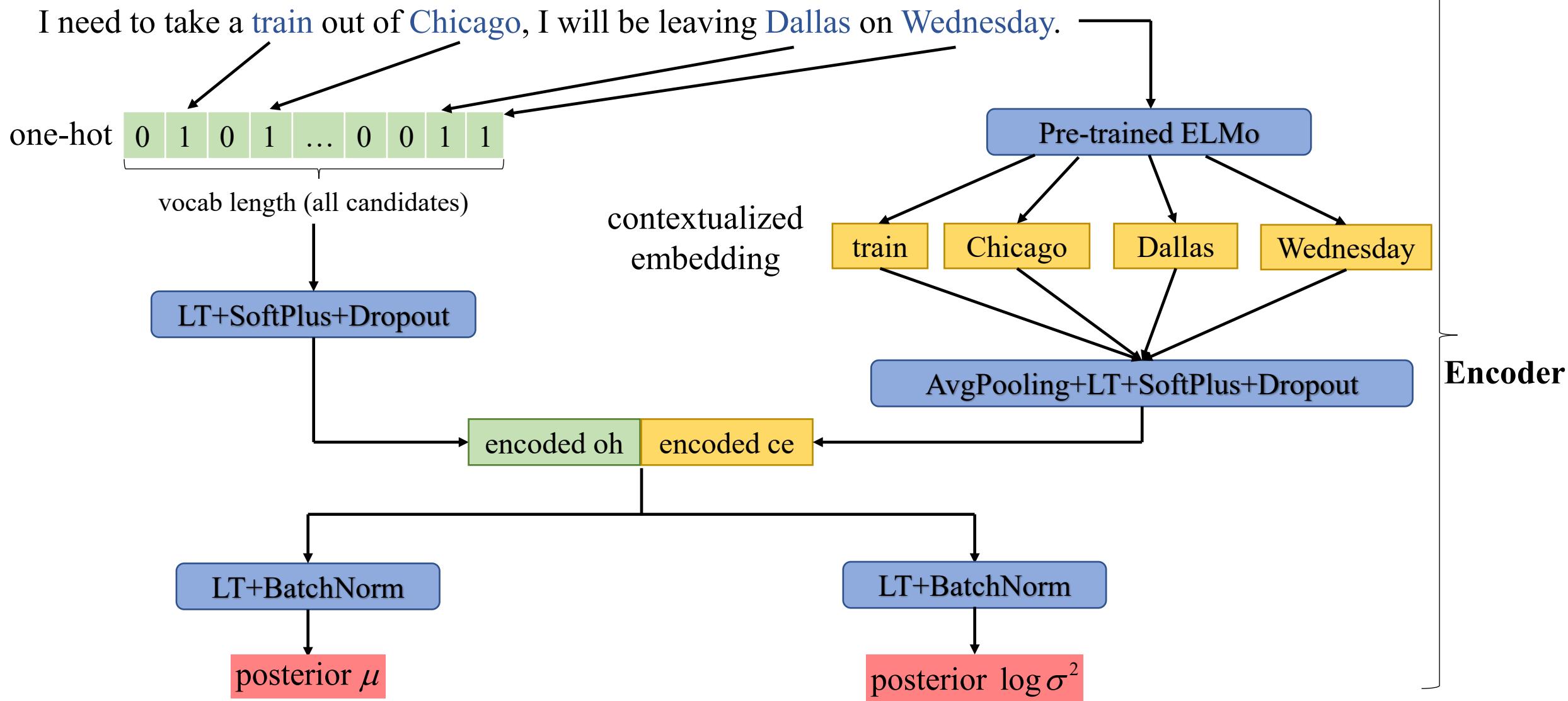
Dallas

Wednesday

AvgPooling+LT+SoftPlus+Dropout

encoded oh | encoded ce

Encoder



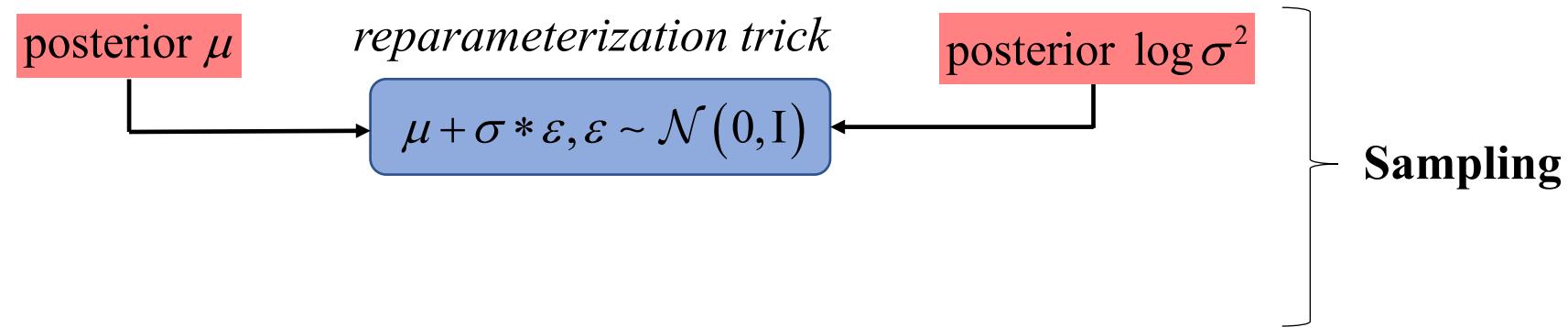
posterior  $\mu$

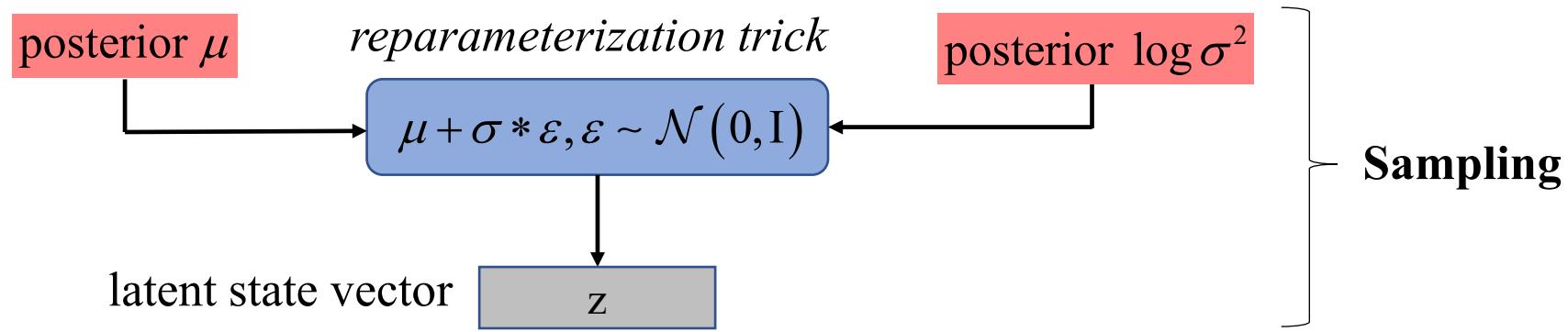
posterior  $\log \sigma^2$

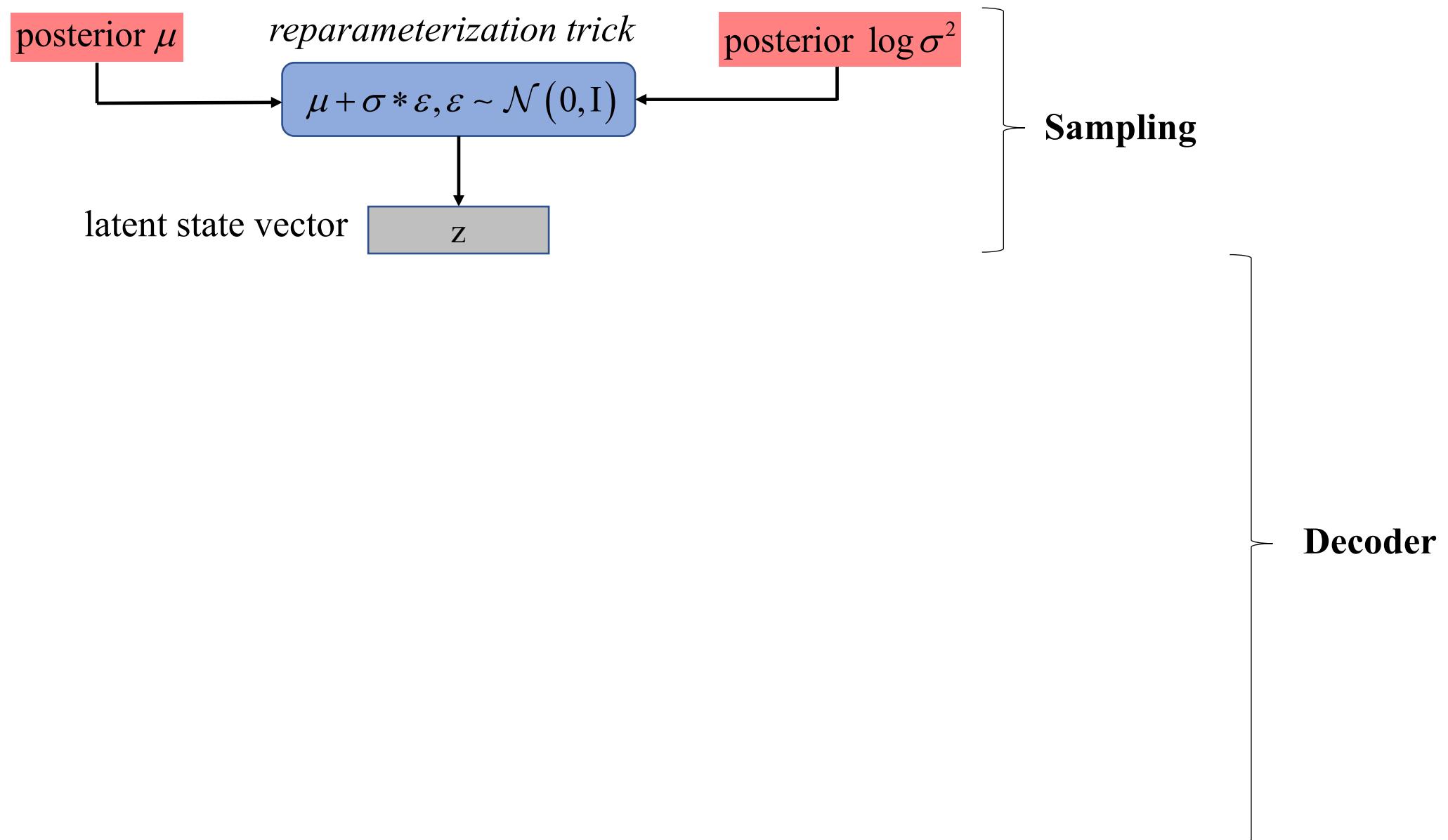
posterior  $\mu$

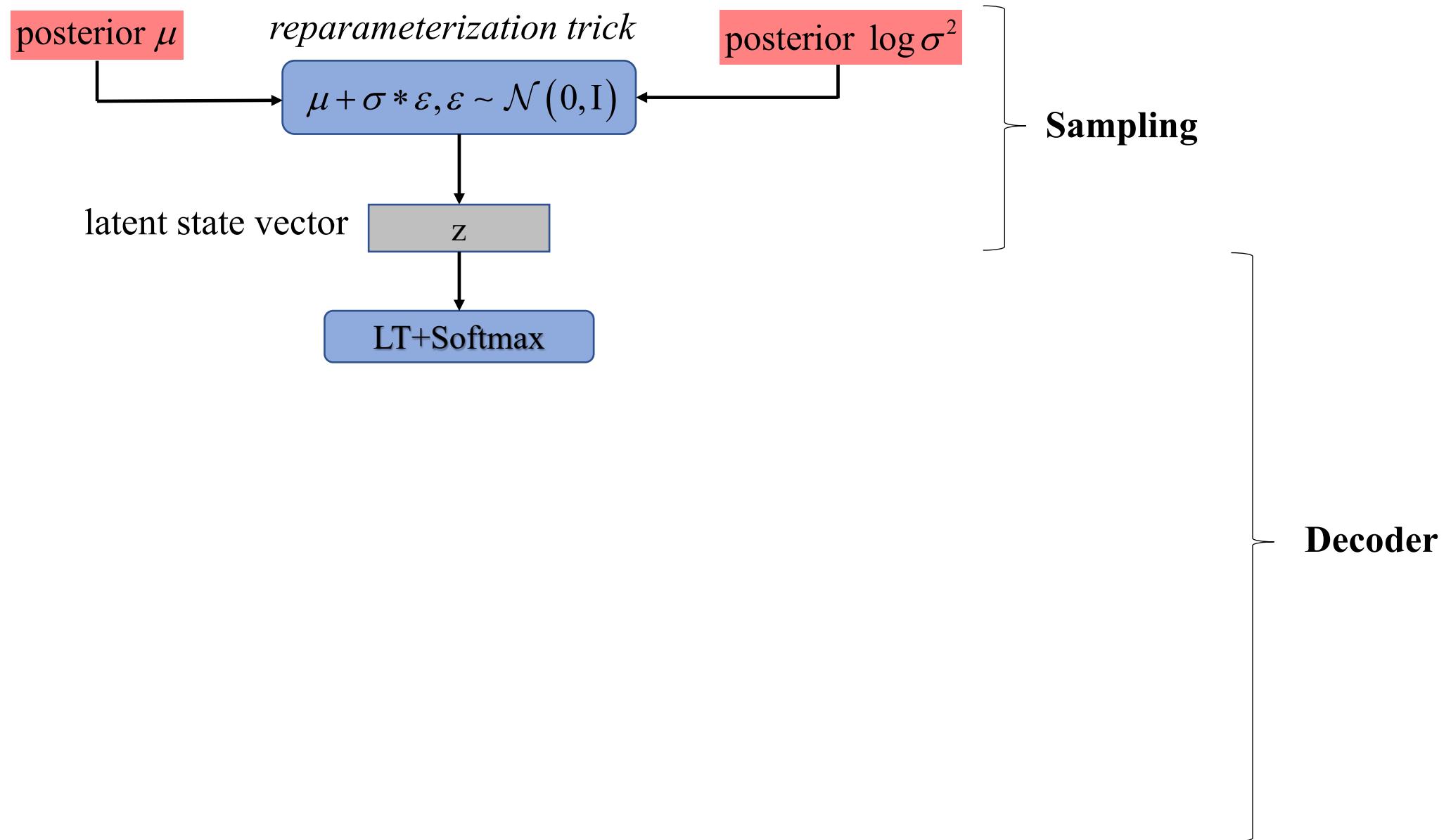
posterior  $\log \sigma^2$

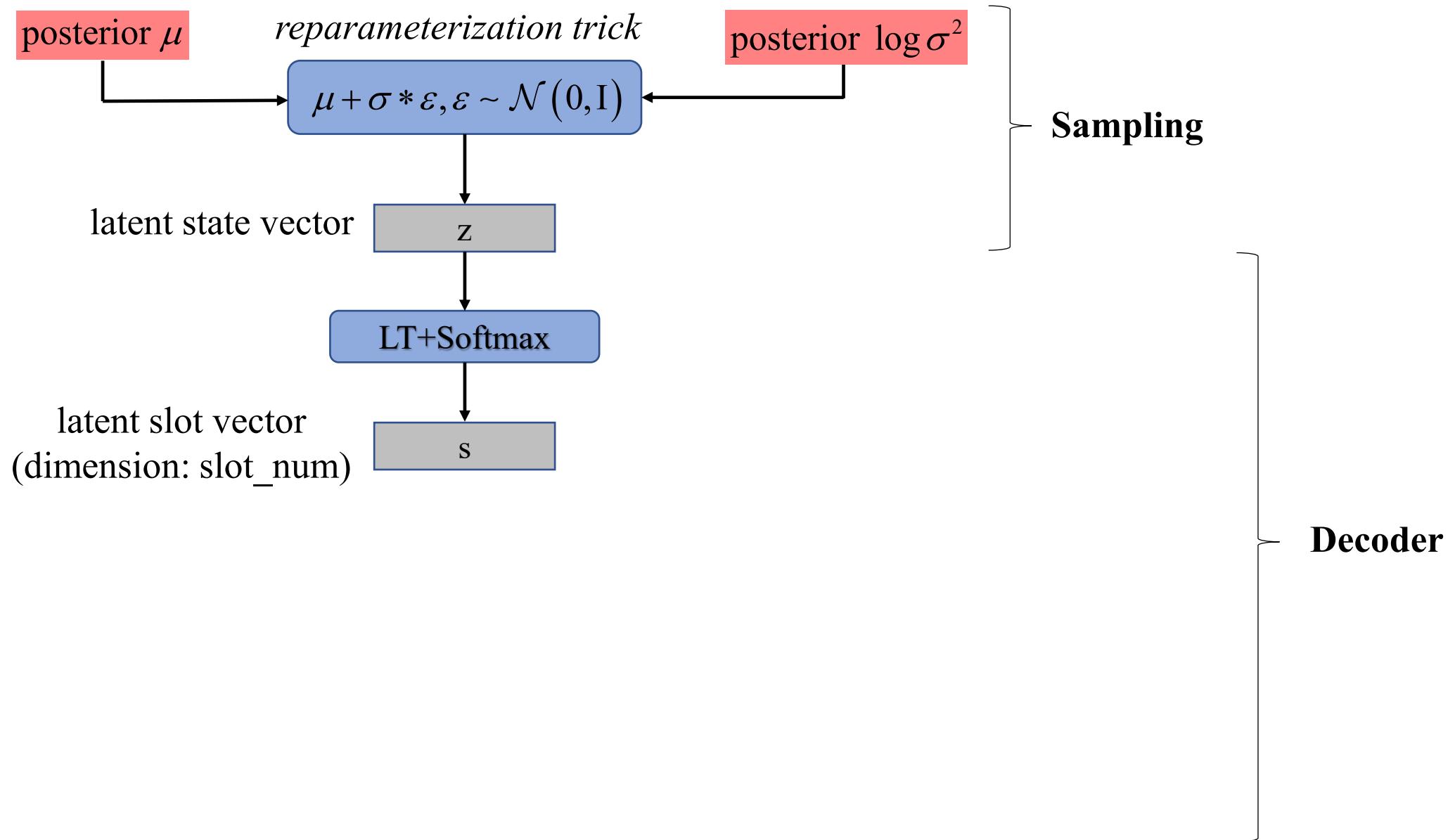
Sampling

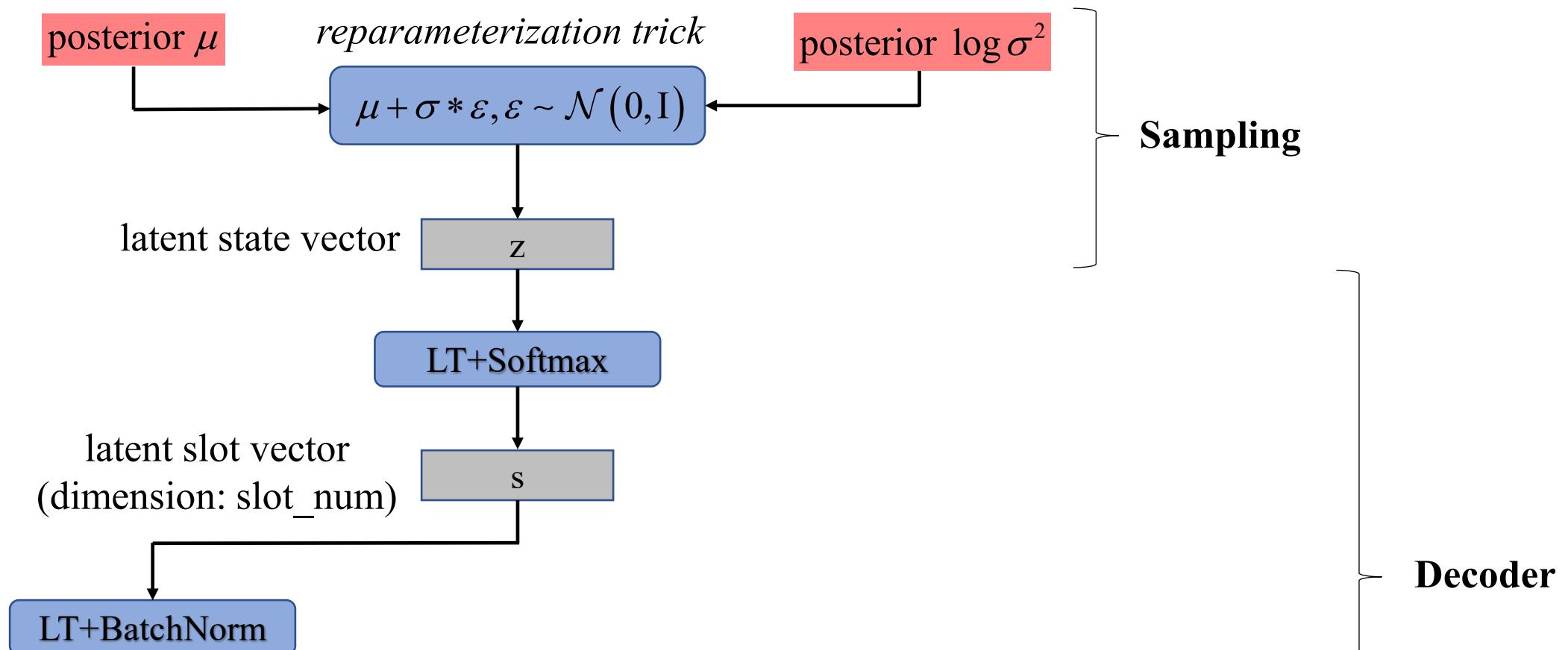


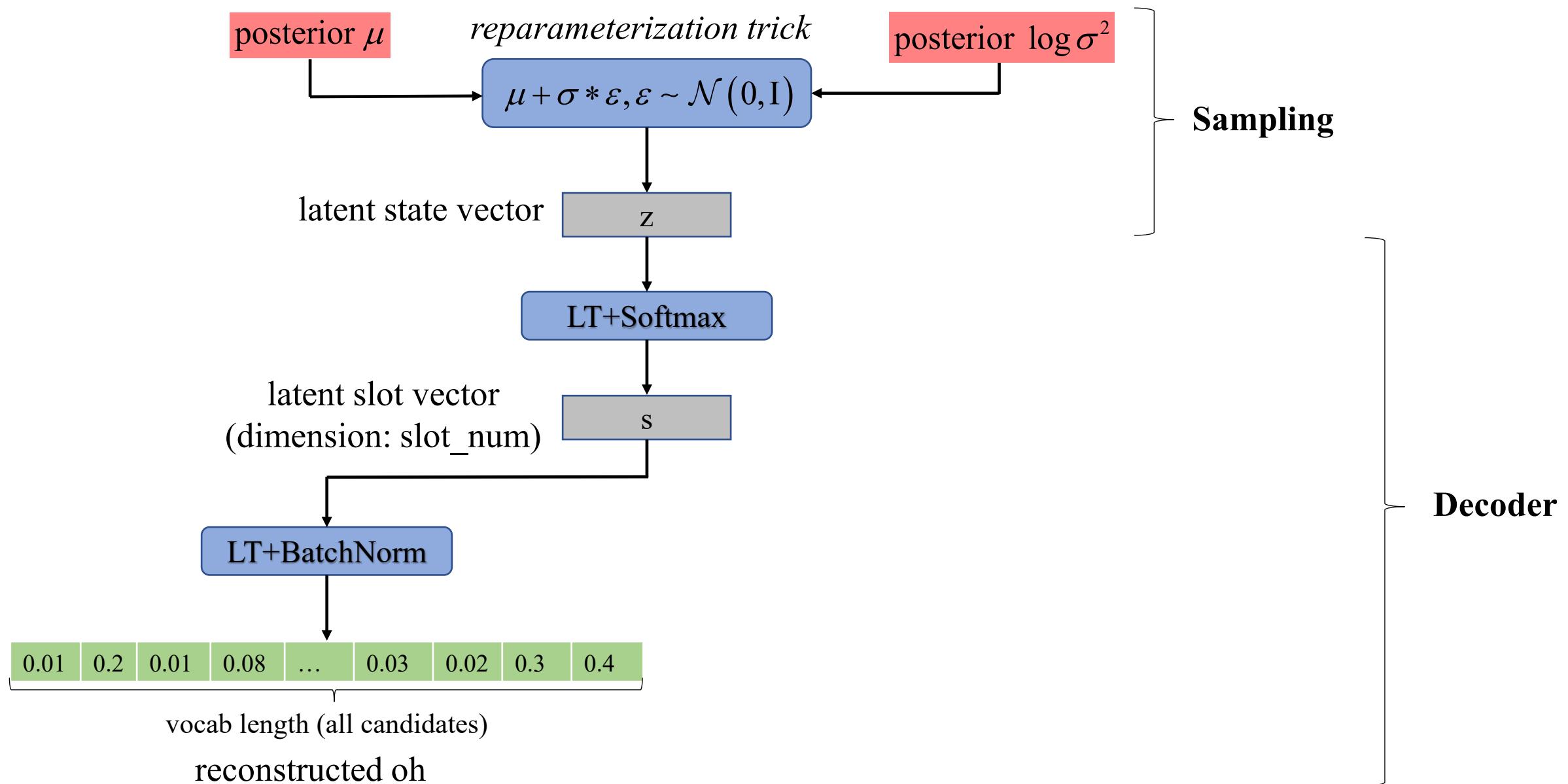


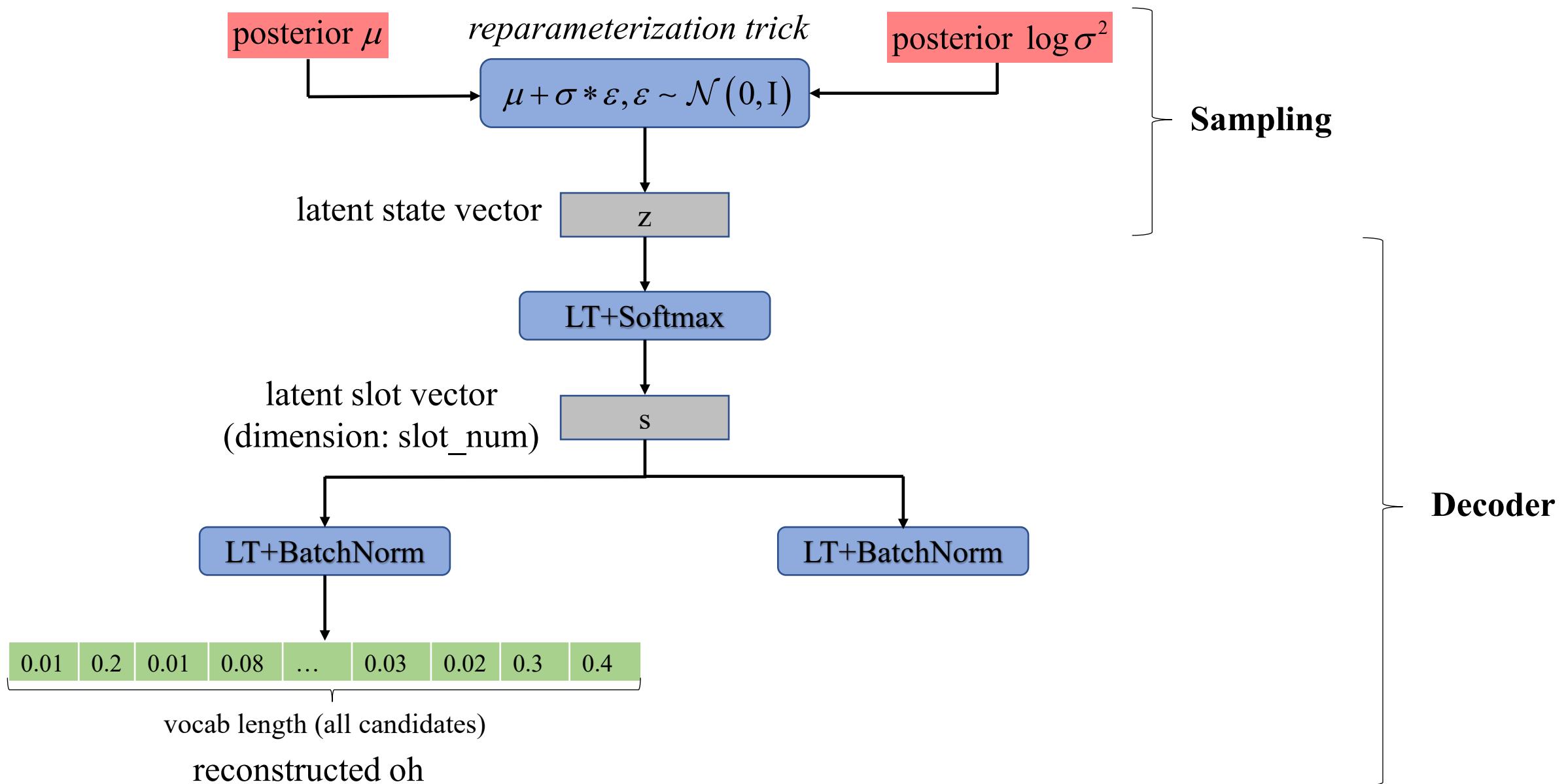


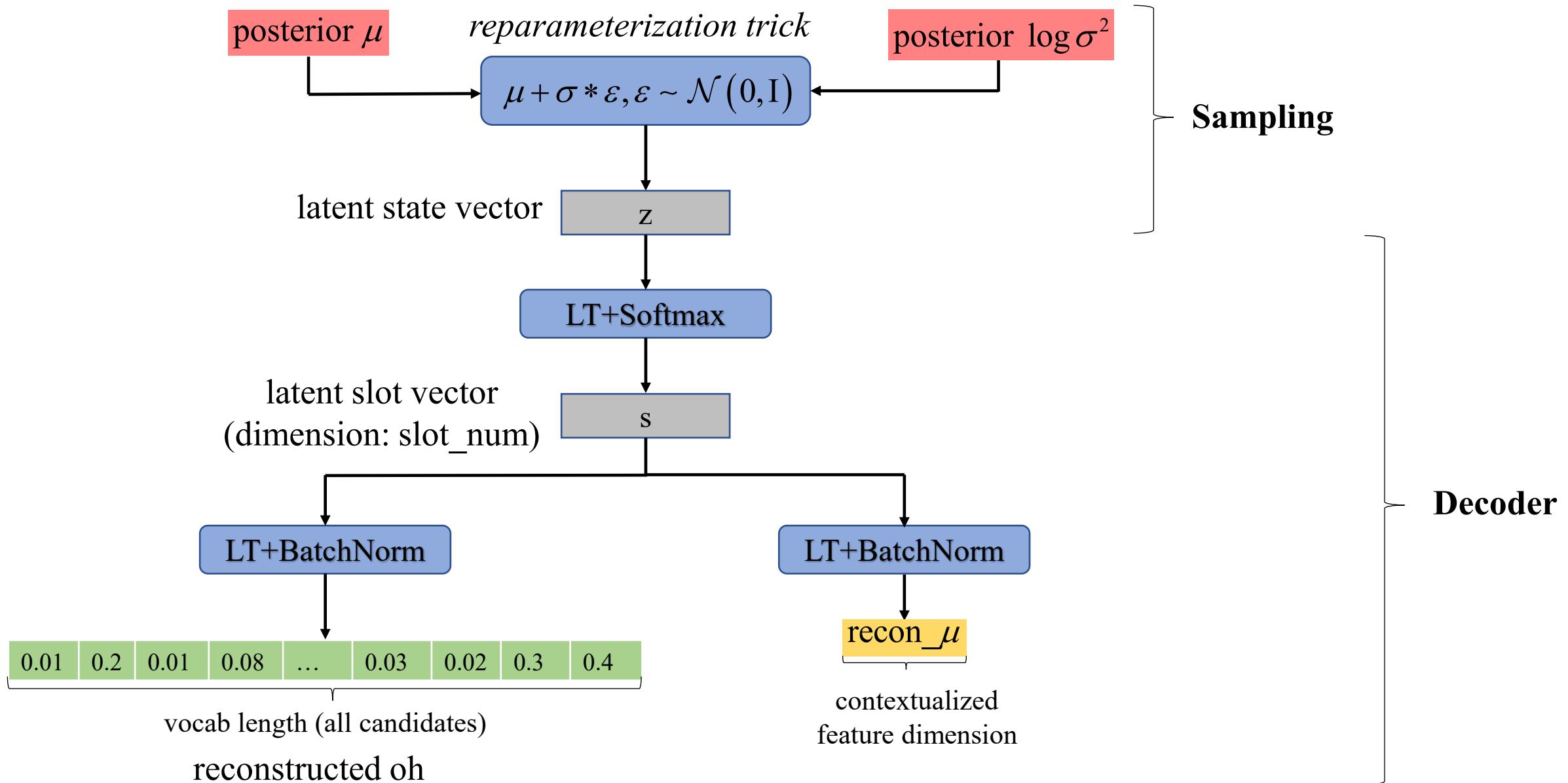


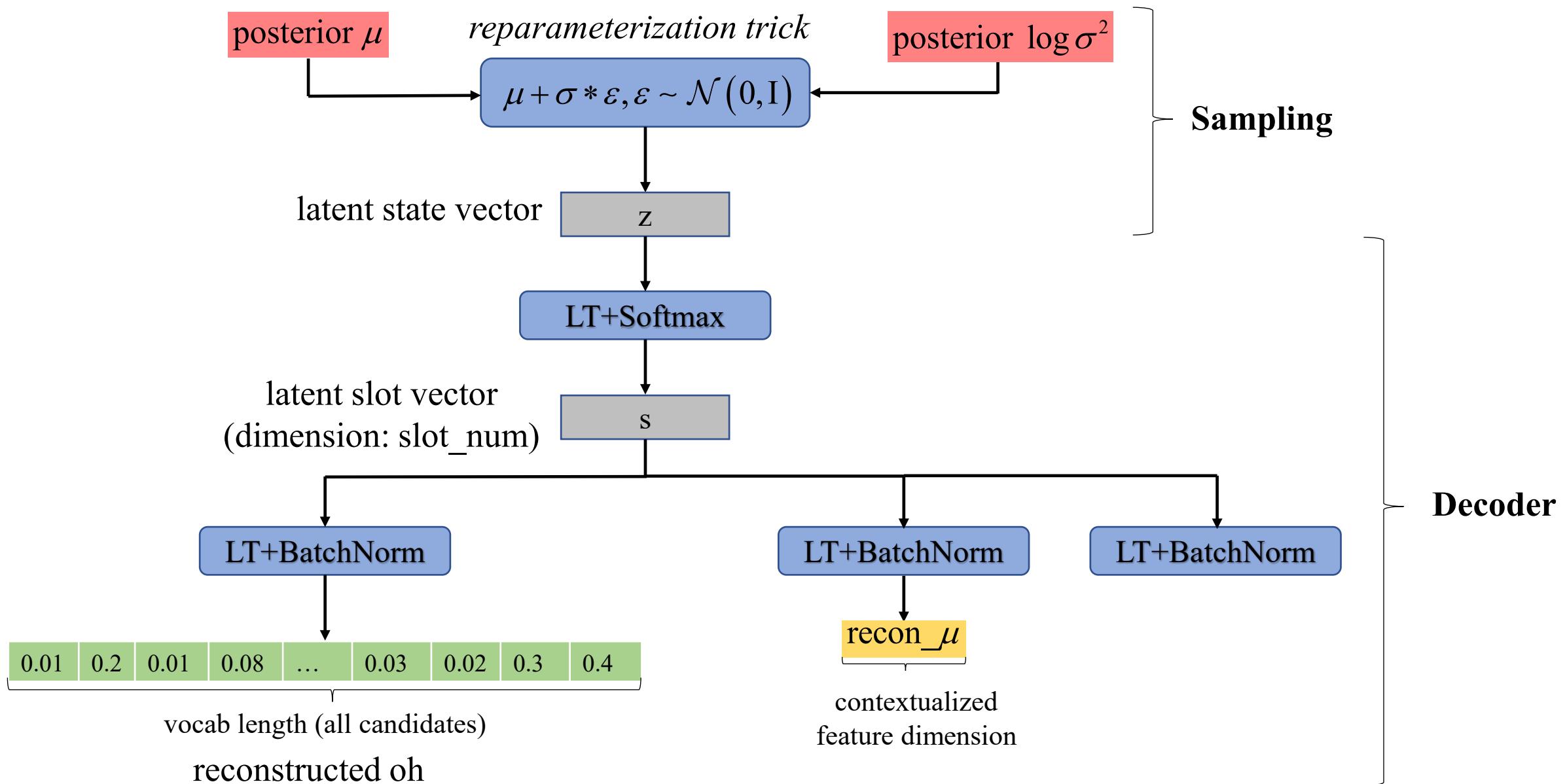


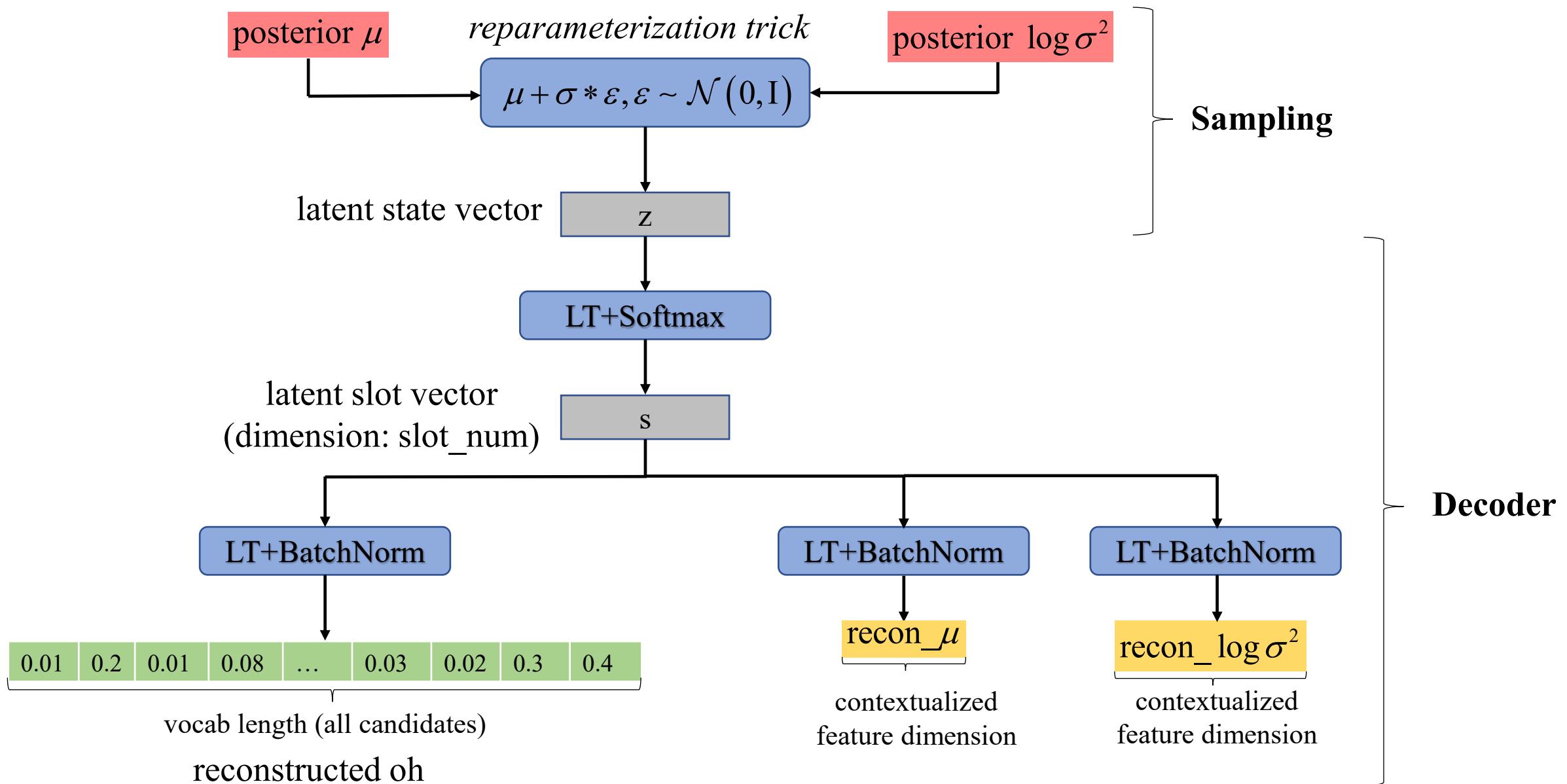


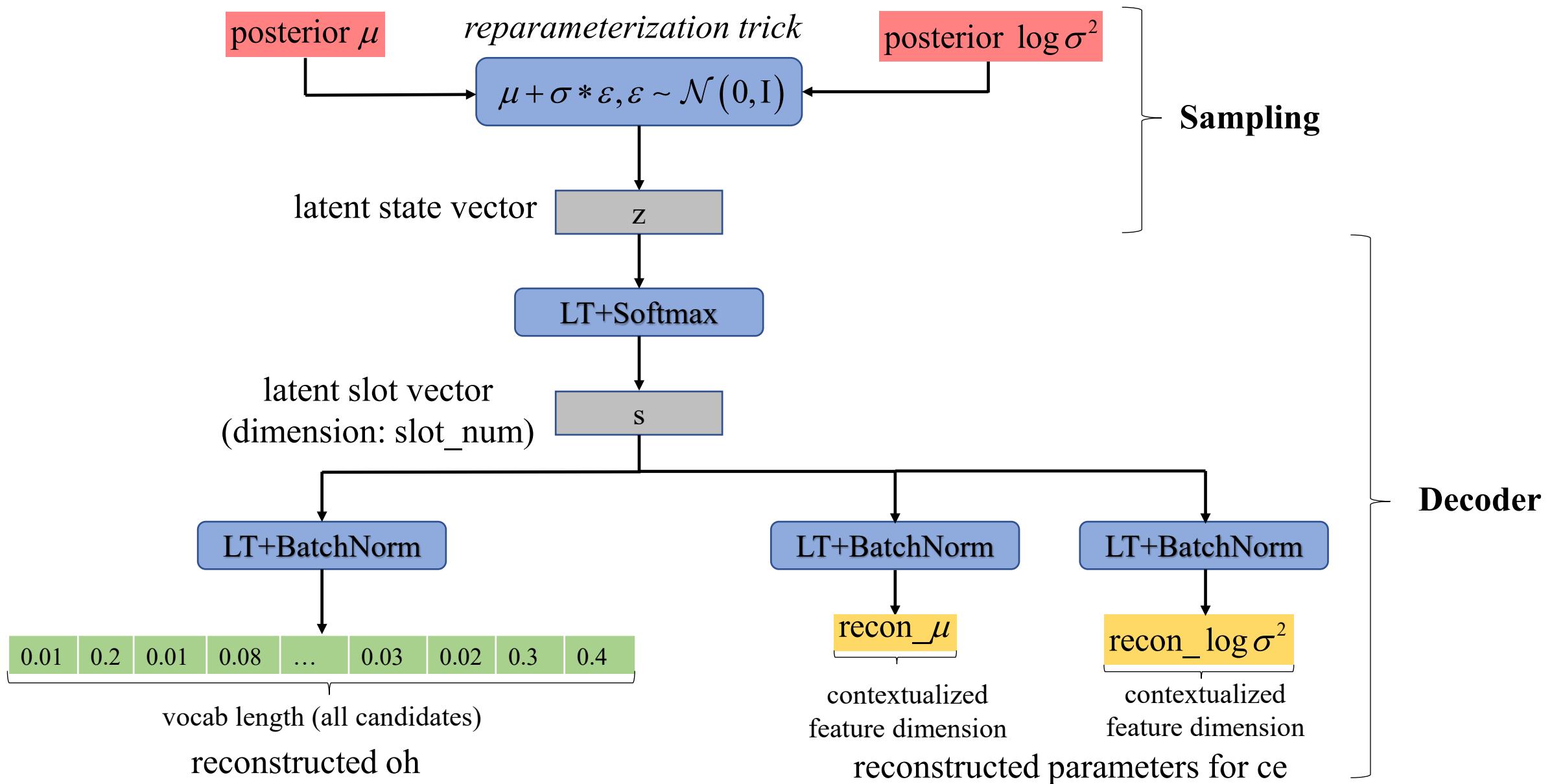










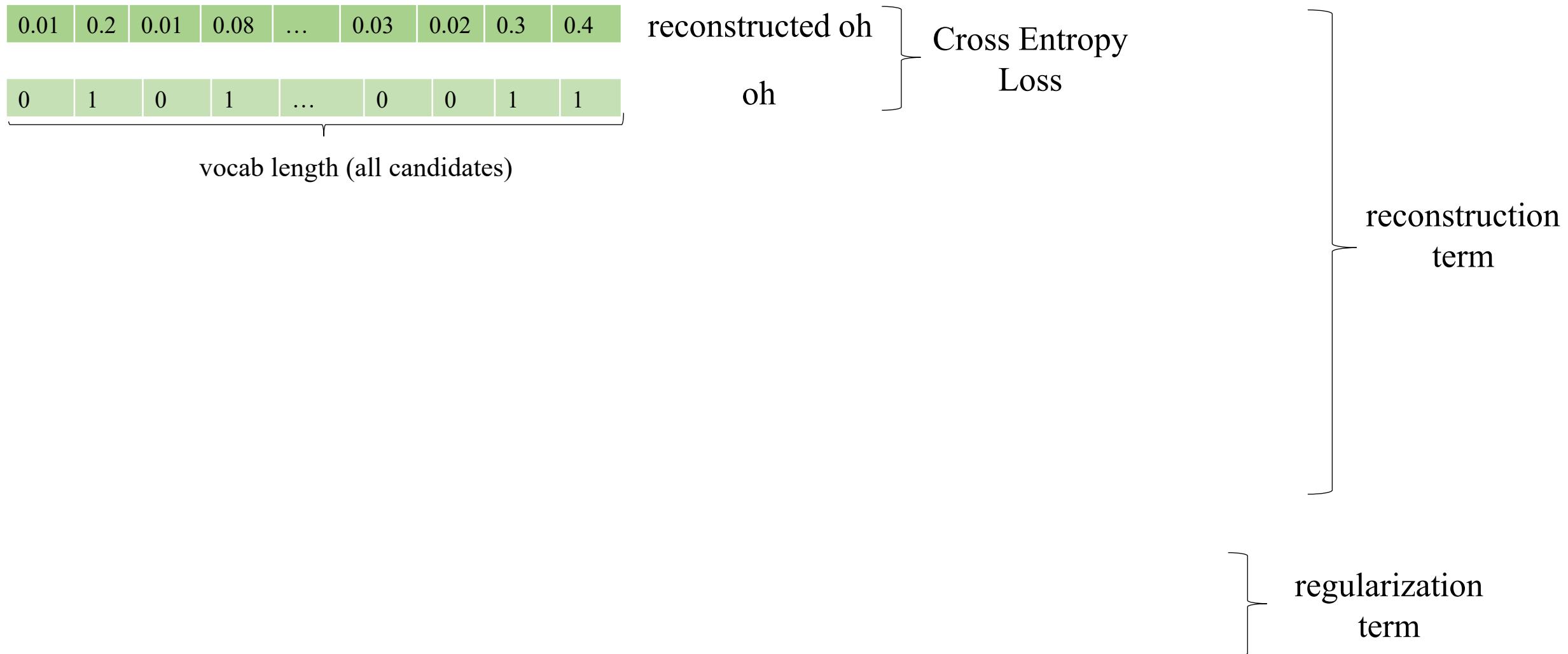


# CHAPTER 2 Loss

reconstruction  
term

regularization  
term

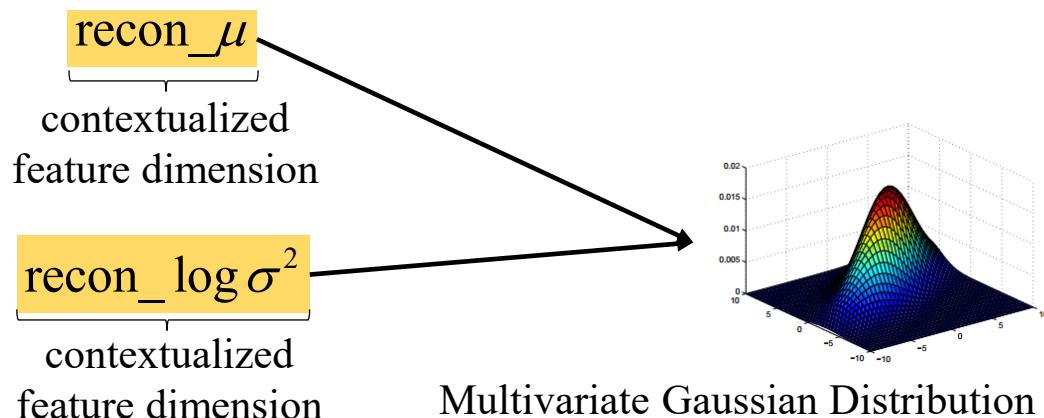
# CHAPTER 2 Loss

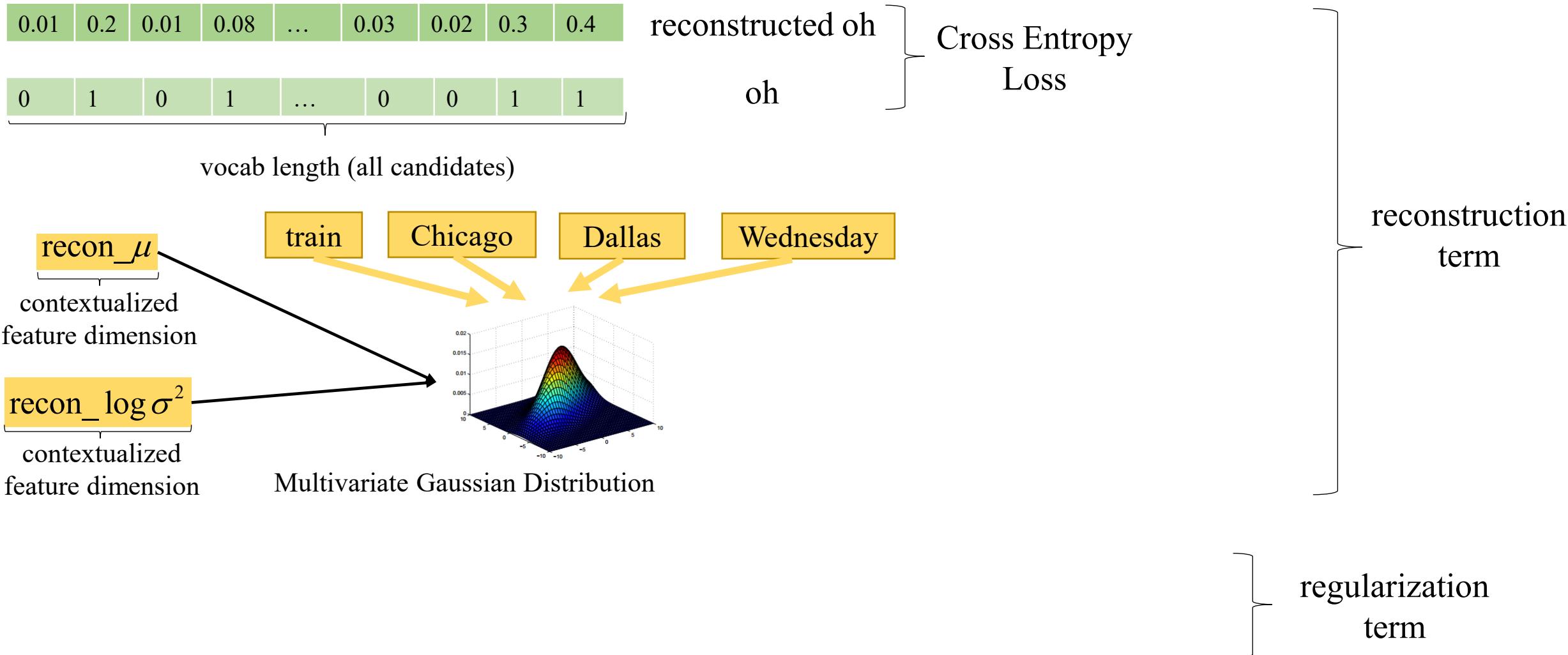


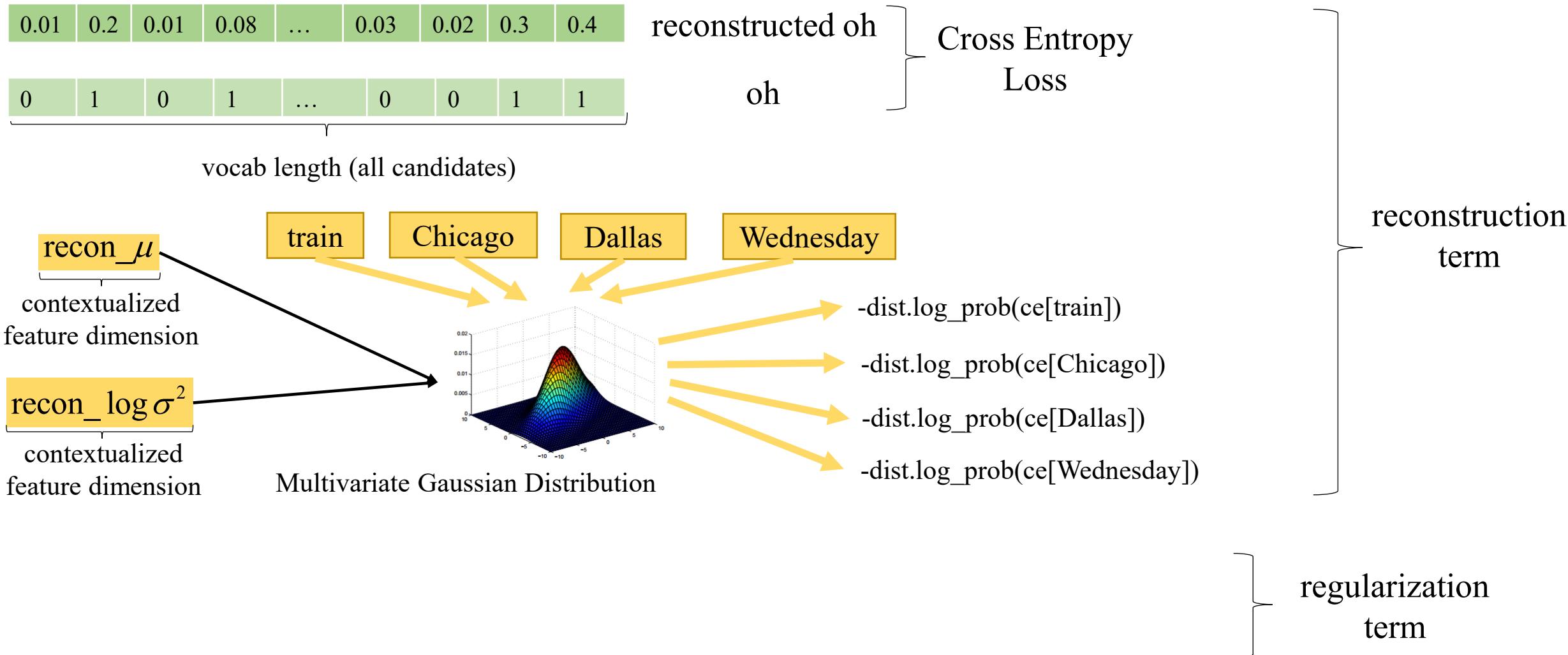
|      |     |      |      |     |      |      |     |     |
|------|-----|------|------|-----|------|------|-----|-----|
| 0.01 | 0.2 | 0.01 | 0.08 | ... | 0.03 | 0.02 | 0.3 | 0.4 |
| 0    | 1   | 0    | 1    | ... | 0    | 0    | 1   | 1   |

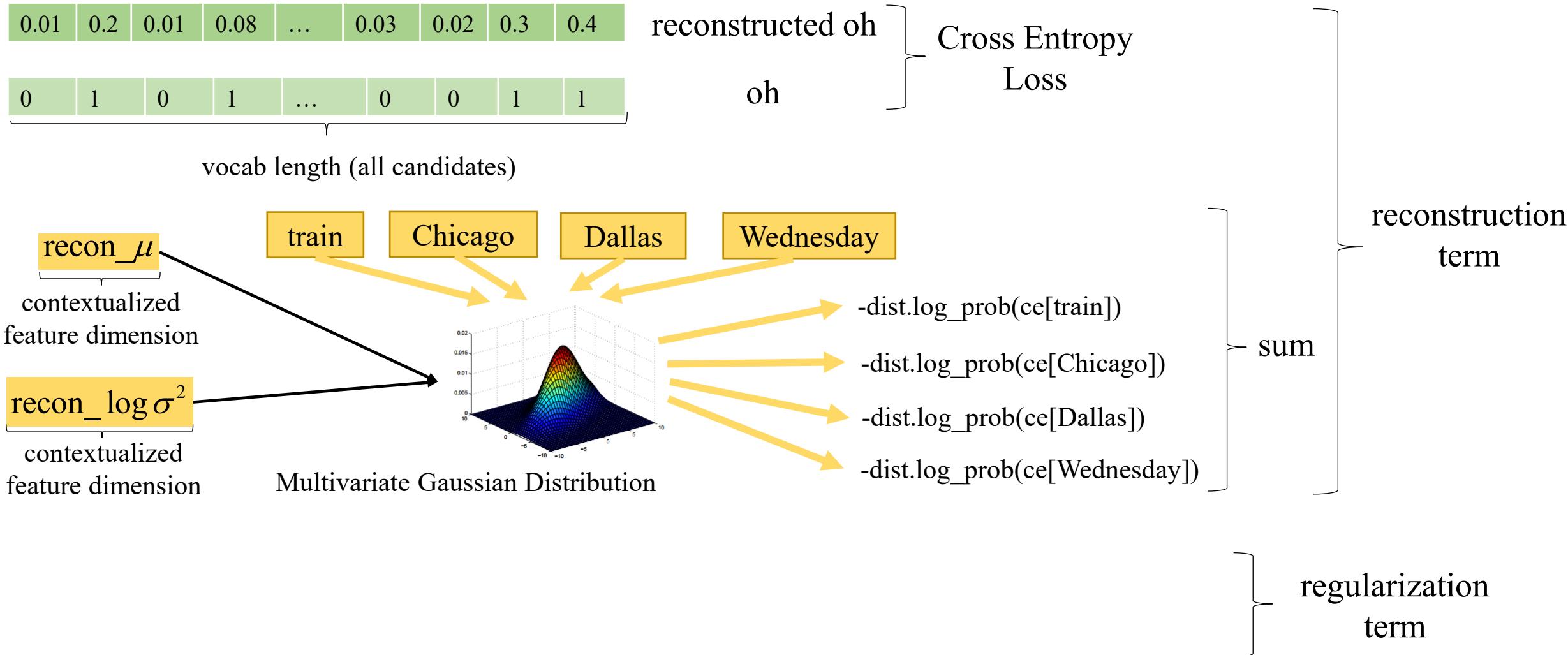
vocab length (all candidates)

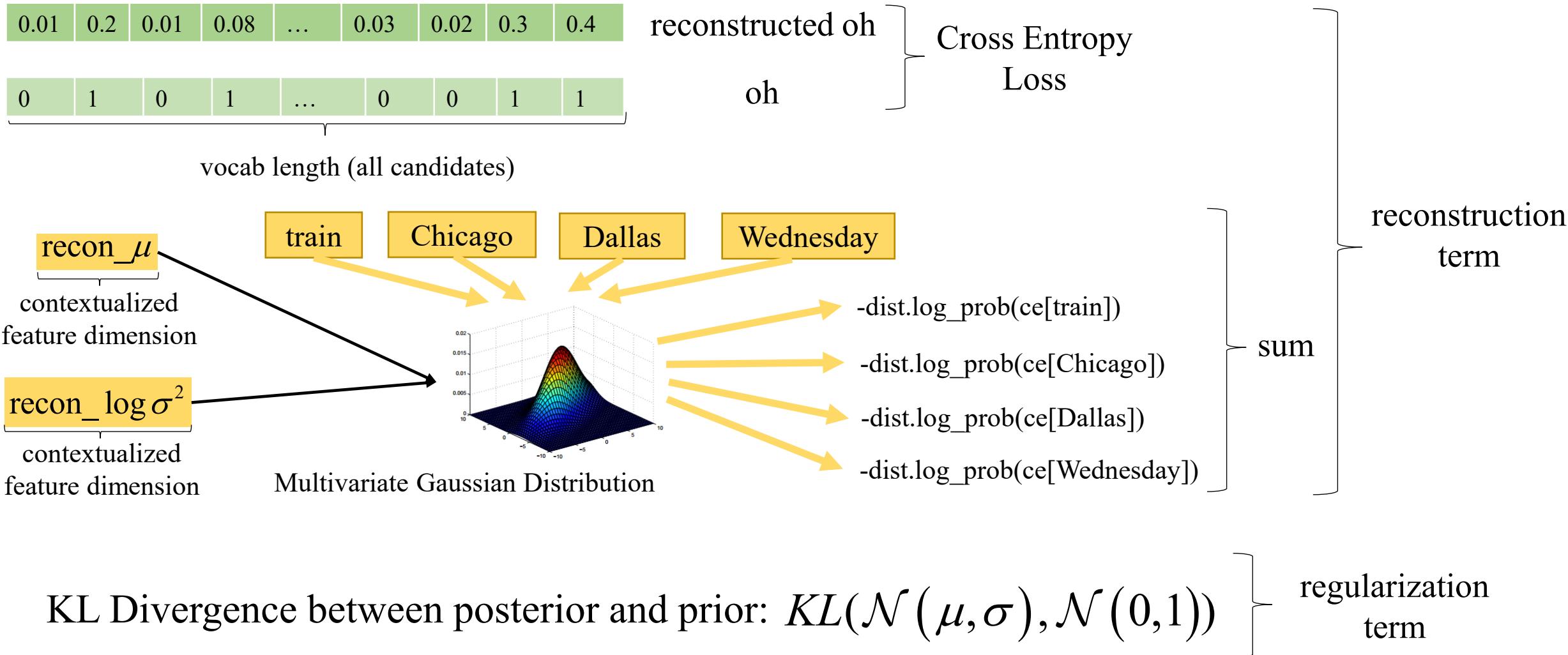
reconstructed oh  
oh

Cross Entropy  
Lossreconstruction  
termregularization  
term









## CHAPTER 2 *DSI-base inference*

I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.

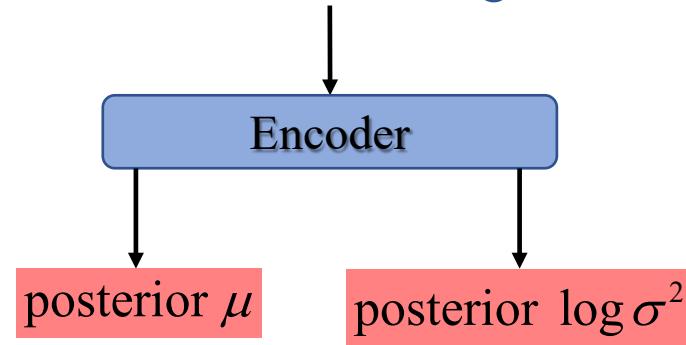
## CHAPTER 2 *DSI-base* inference

I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.

Encoder

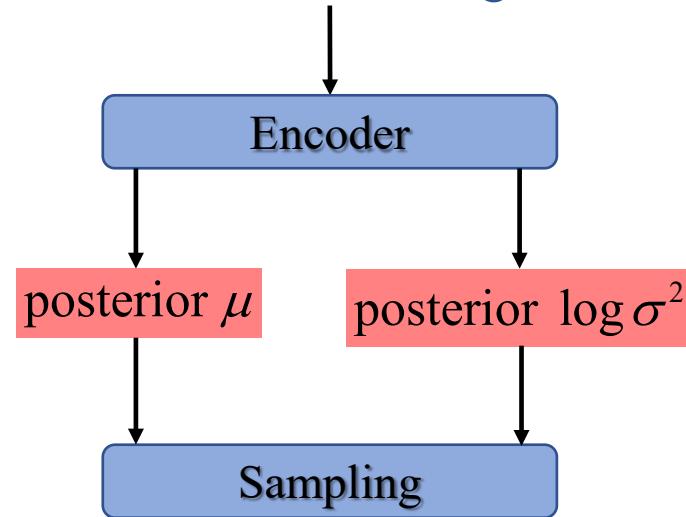
## CHAPTER 2 *DSI-base* inference

I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



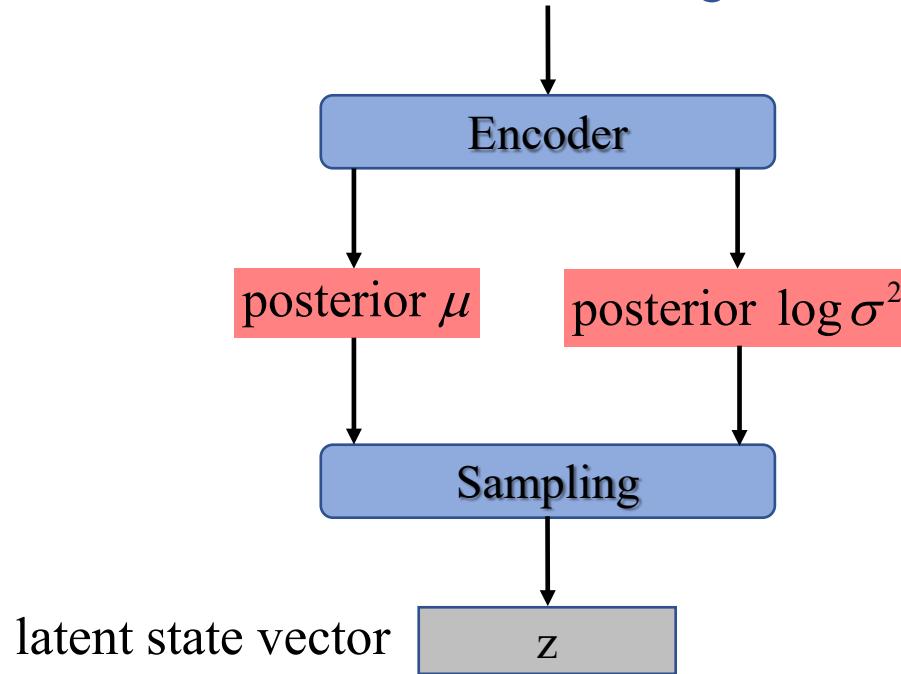
## CHAPTER 2 *DSI-base inference*

I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



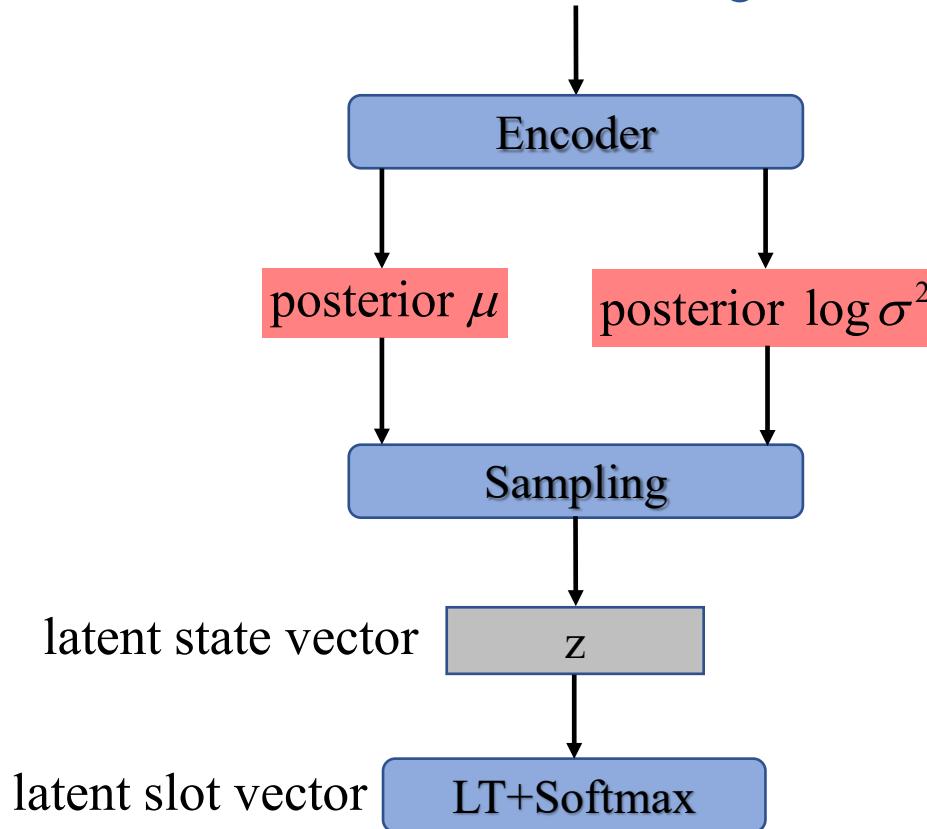
CHAPTER 2 *DSI-base* inference

I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



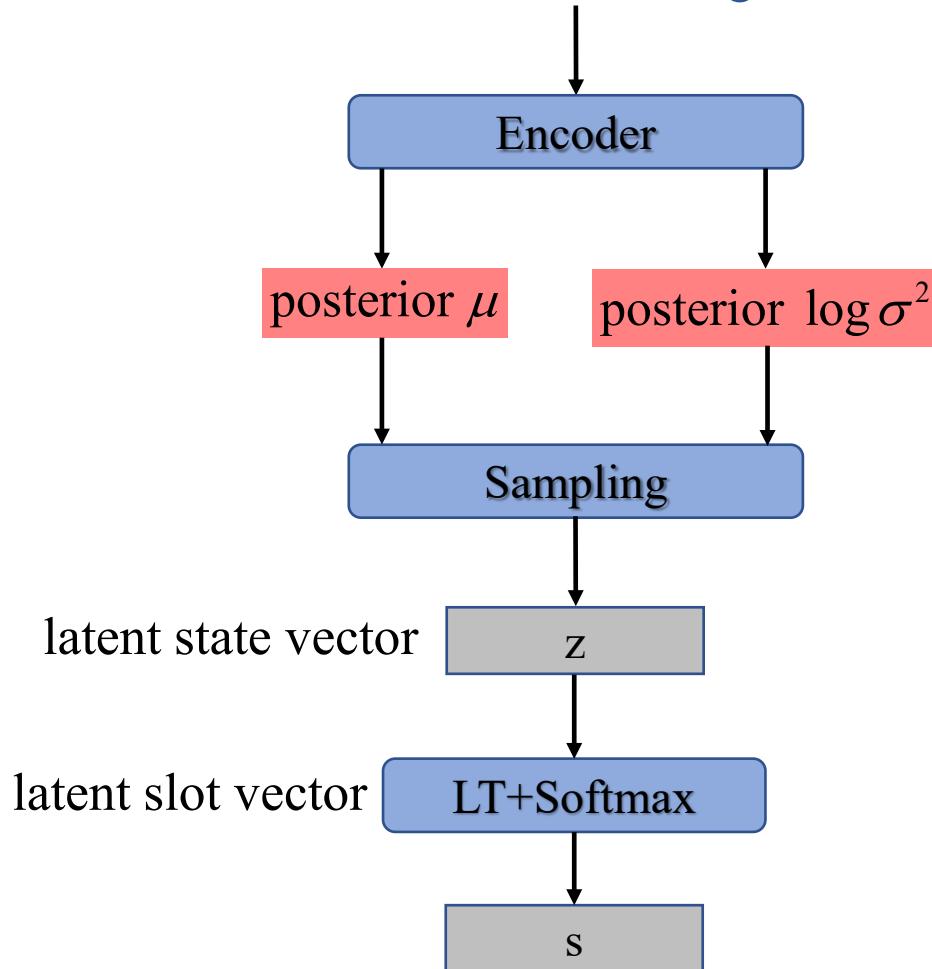
CHAPTER 2 *DSI-base* inference

I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.

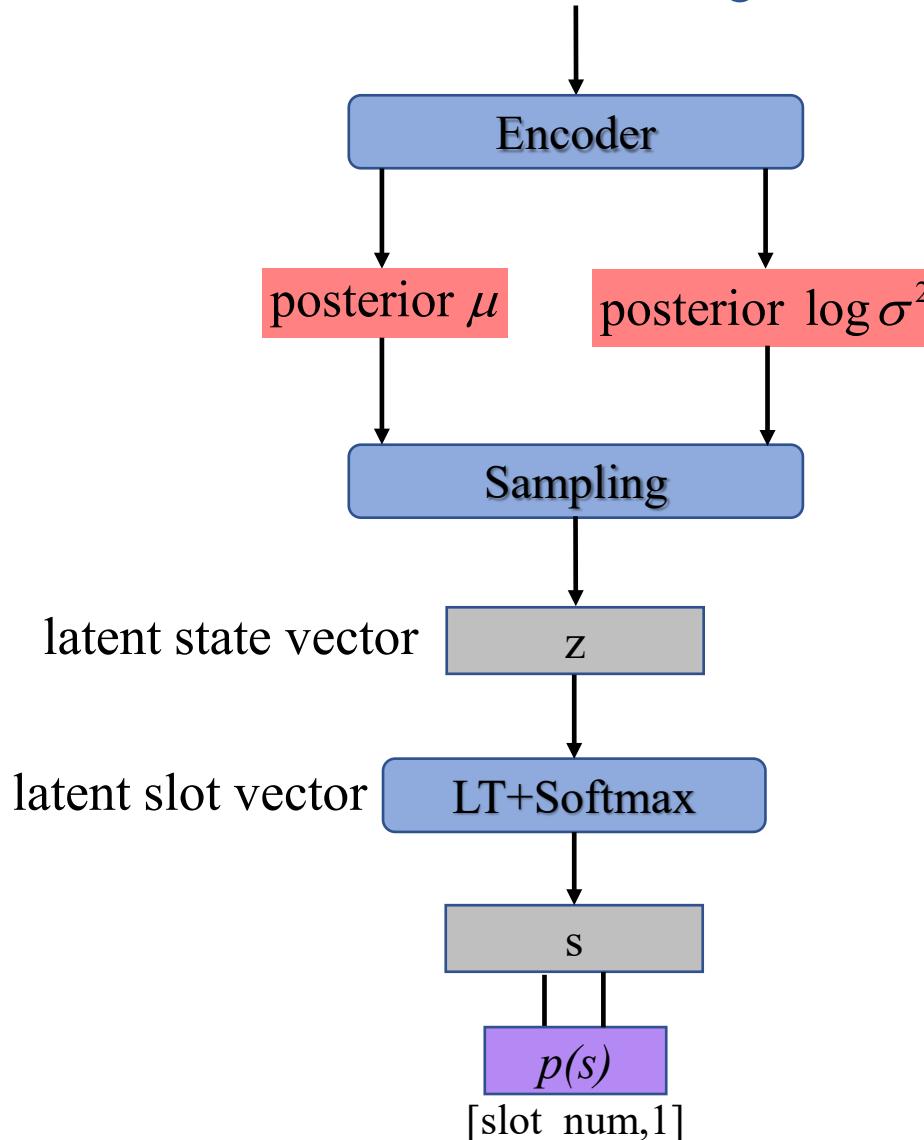


## CHAPTER 2 *DSI-base inference*

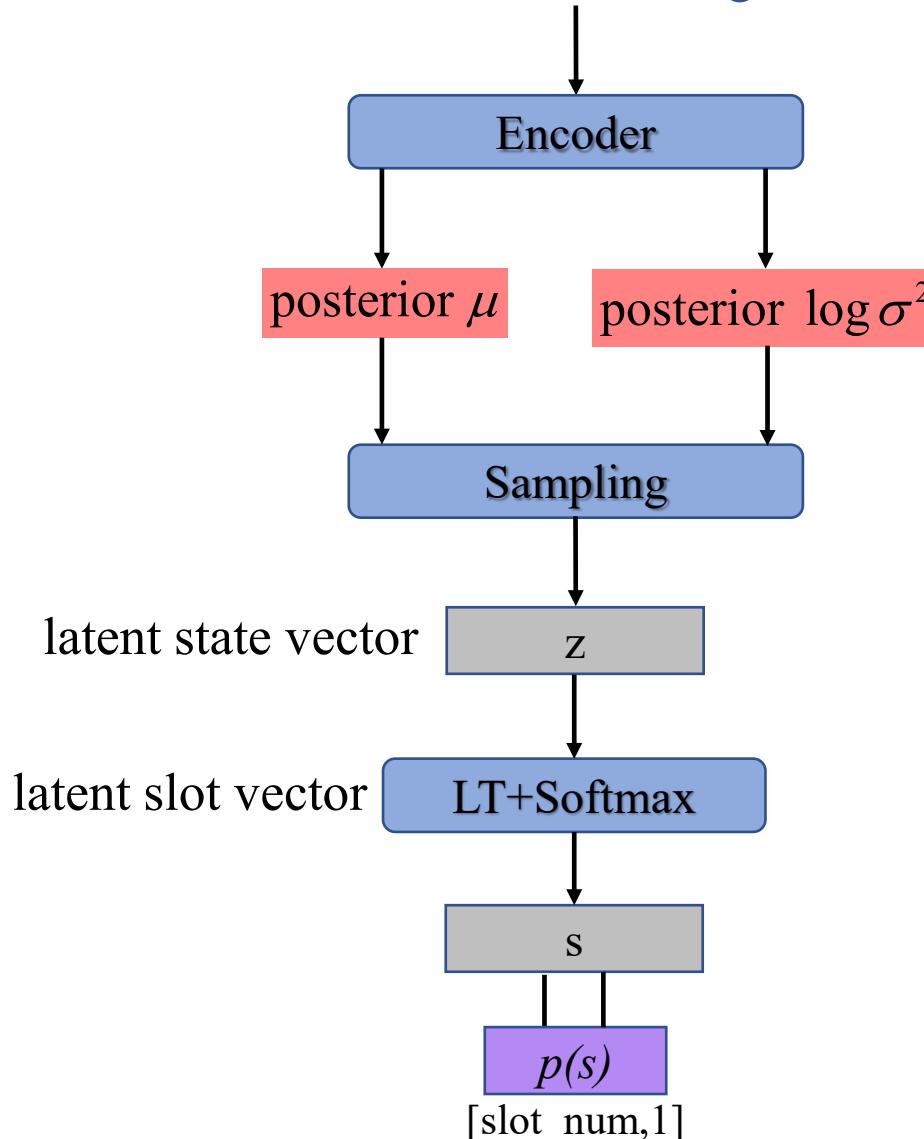
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.

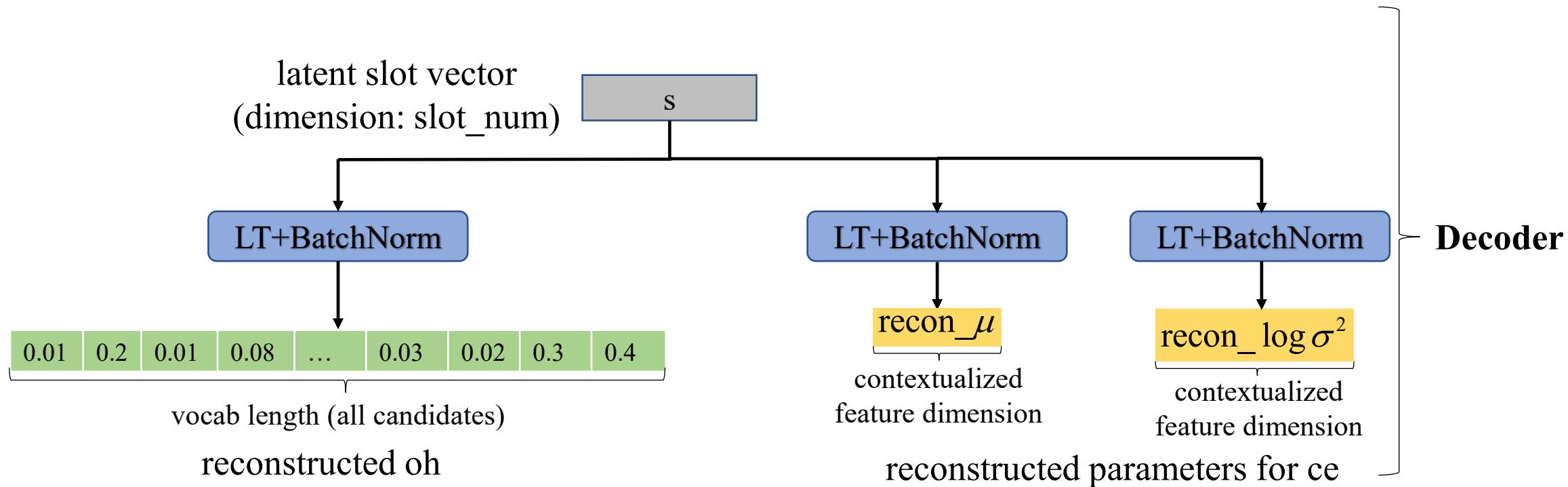


I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.

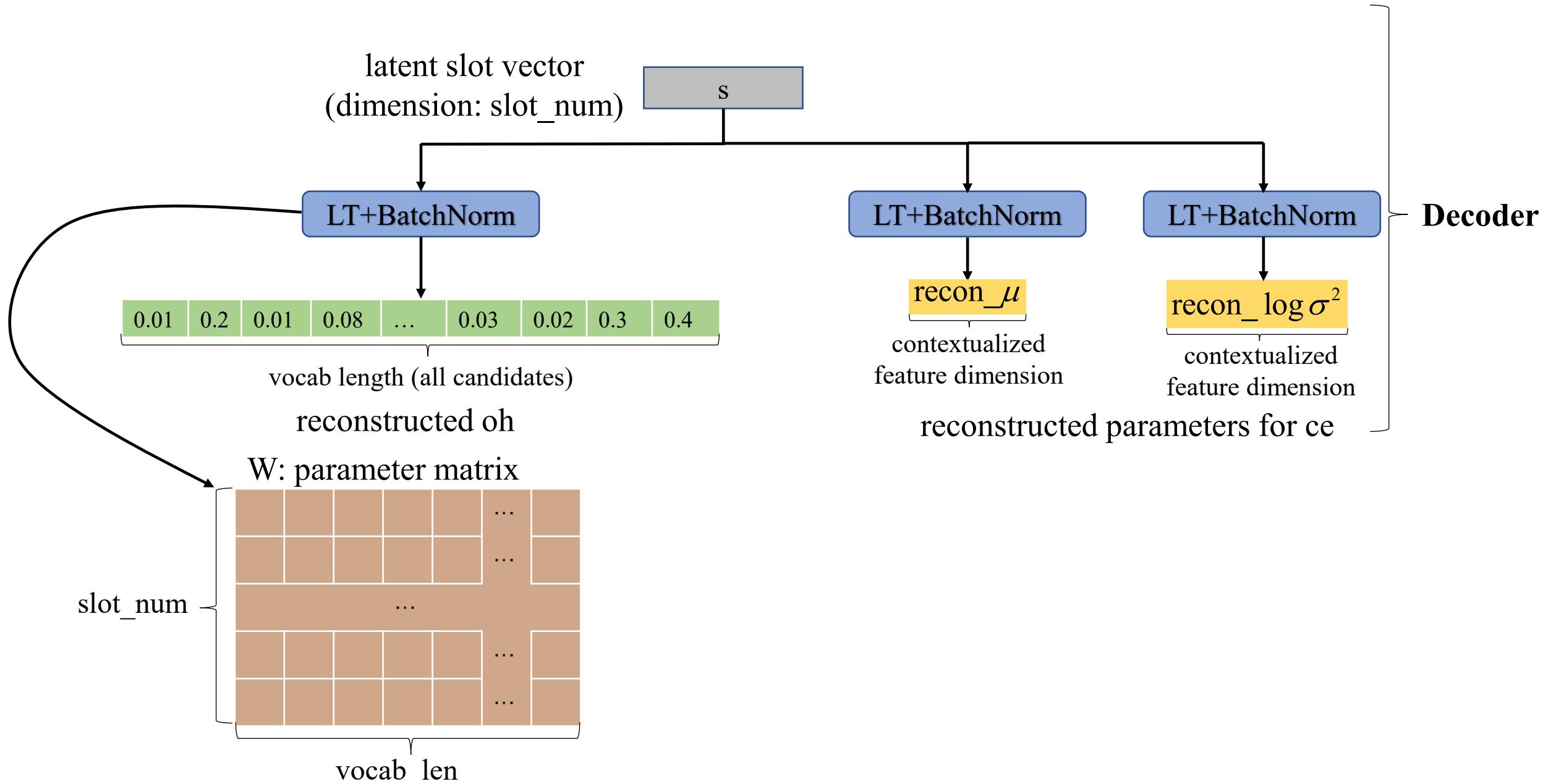


For each candidate in {train, Chicago, Dallas, Wednesday}

## CHAPTER 2 What does the model learn ?



## CHAPTER 2 What does the model learn ?



# CHAPTER 2 What does the model learn ?

s: slot vector

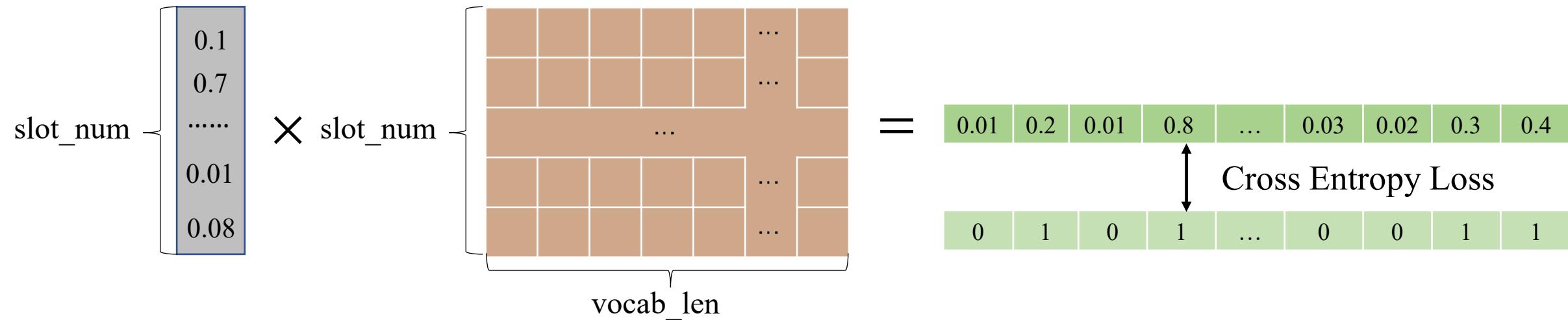
W: parameter matrix

$$\begin{array}{c} \text{slot\_num} \left\{ \begin{array}{l} 0.1 \\ 0.7 \\ \cdots \\ 0.01 \\ 0.08 \end{array} \right. \times \text{slot\_num} \left\{ \begin{array}{c} \times \text{slot\_num} \left\{ \begin{array}{c} \text{vocab\_len} \left\{ \begin{array}{c} \cdots \\ \cdots \end{array} \right. \end{array} \right. \end{array} \right. \end{array} = \begin{array}{cccccccccc} 0.01 & 0.2 & 0.01 & 0.8 & \cdots & 0.03 & 0.02 & 0.3 & 0.4 \end{array}$$

## CHAPTER 2 What does the model learn ?

s: slot vector

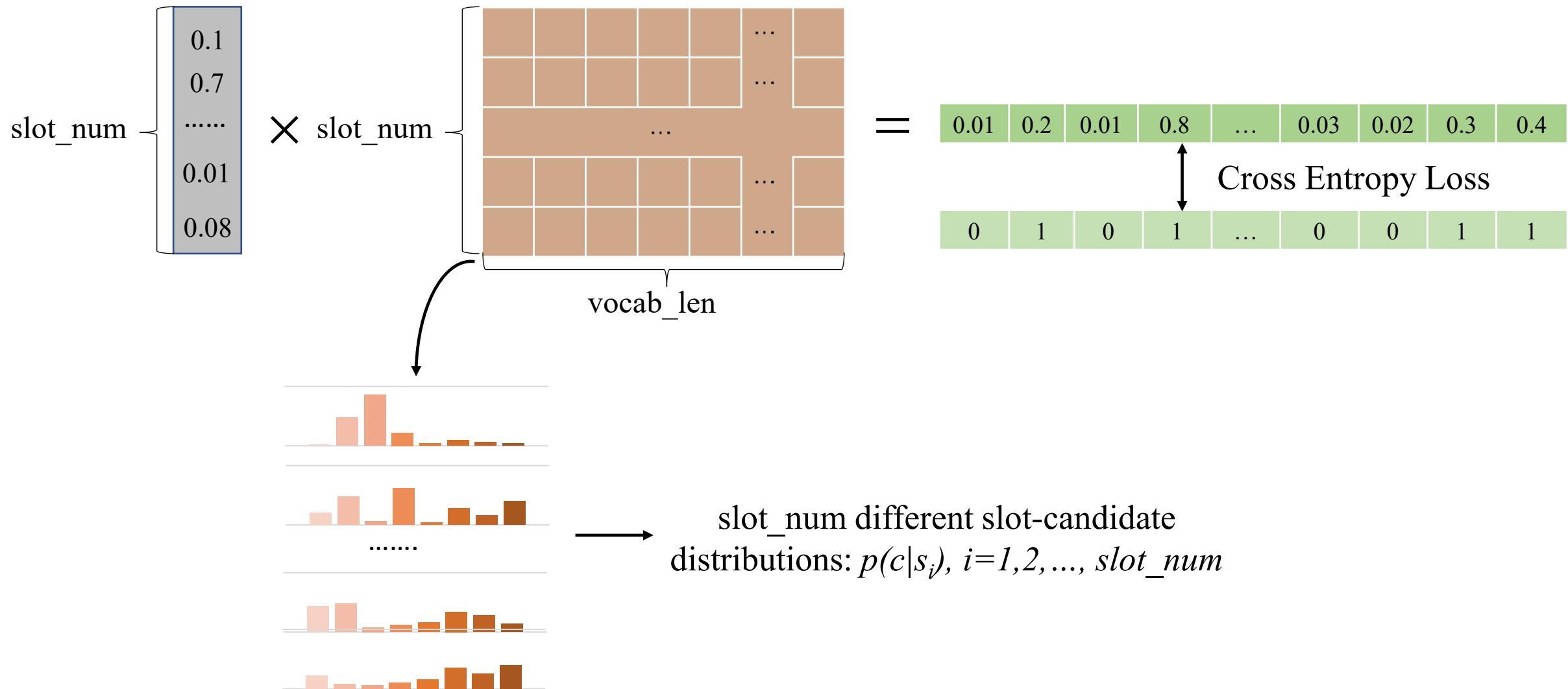
## W: parameter matrix



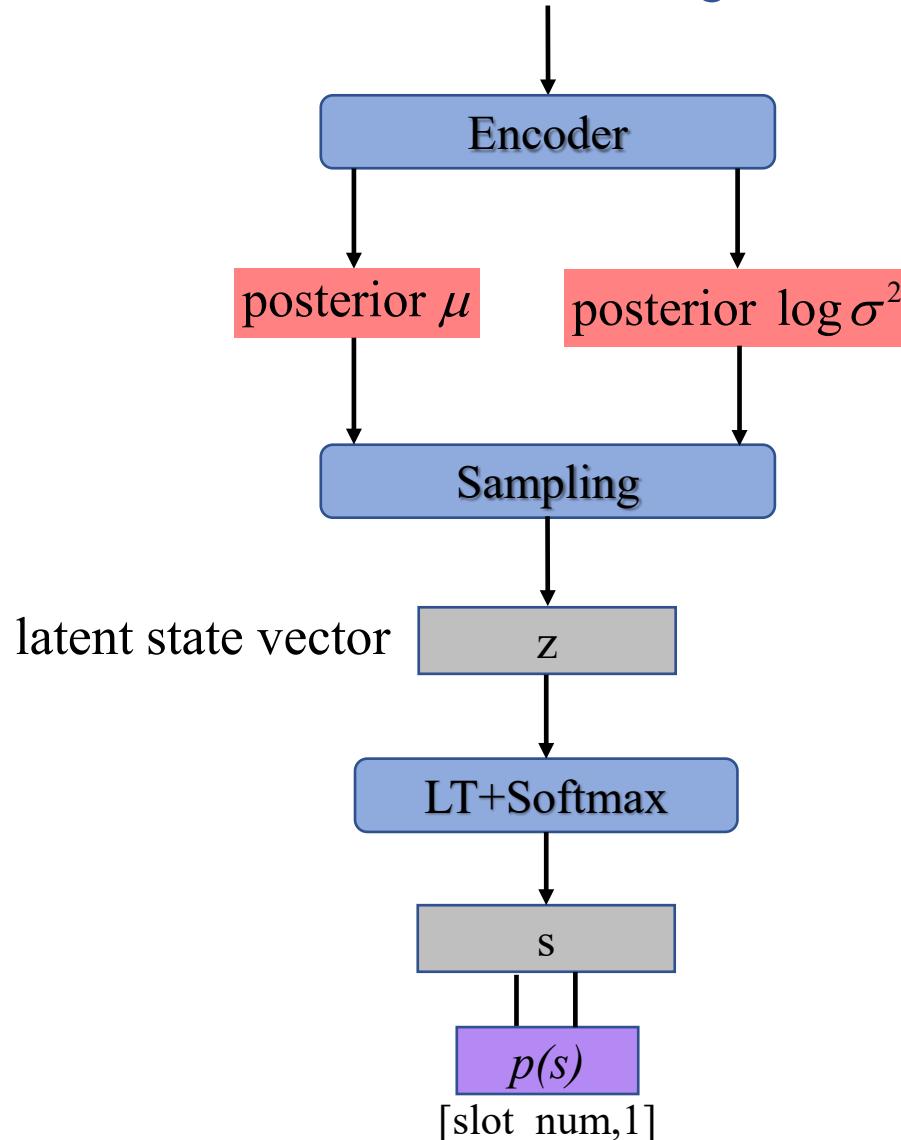
# CHAPTER 2 What does the model learn ?

s: slot vector

W: parameter matrix

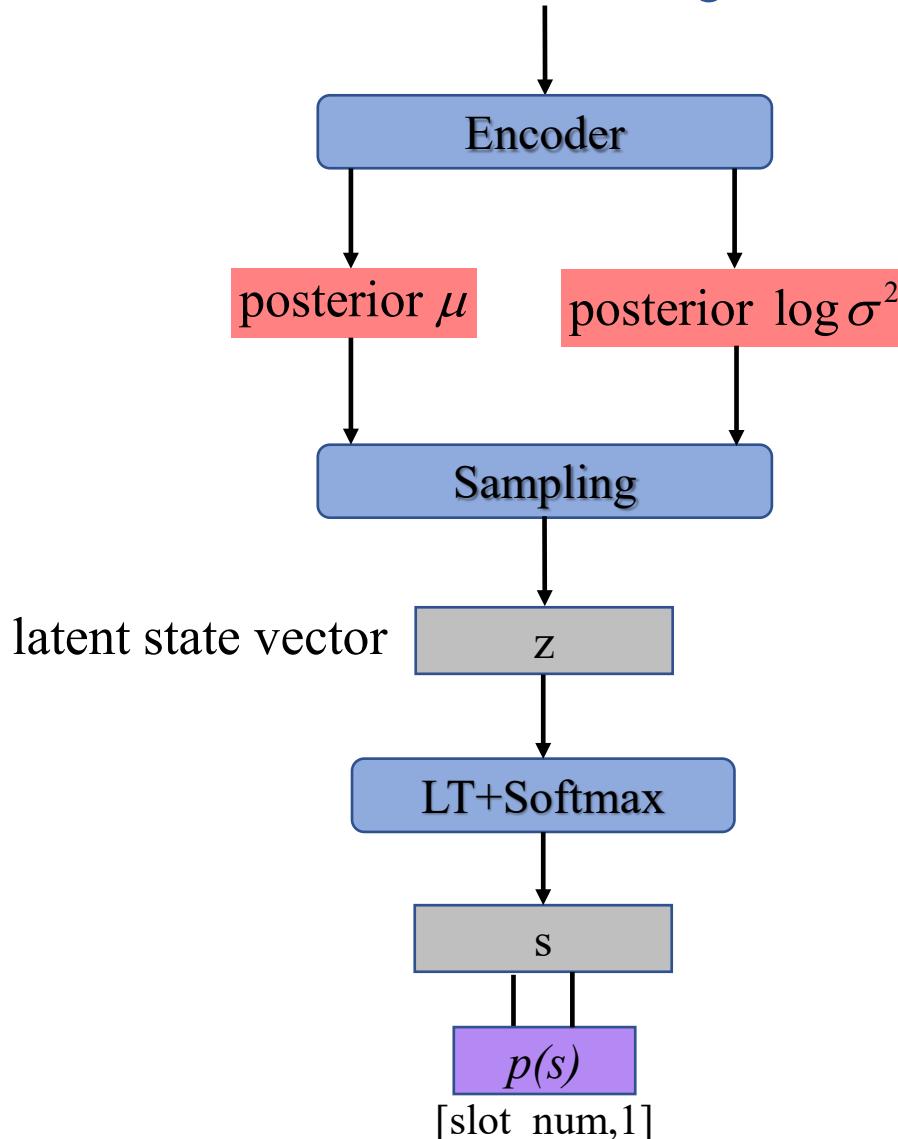


I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



For each candidate in {train, Chicago, Dallas, Wednesday}

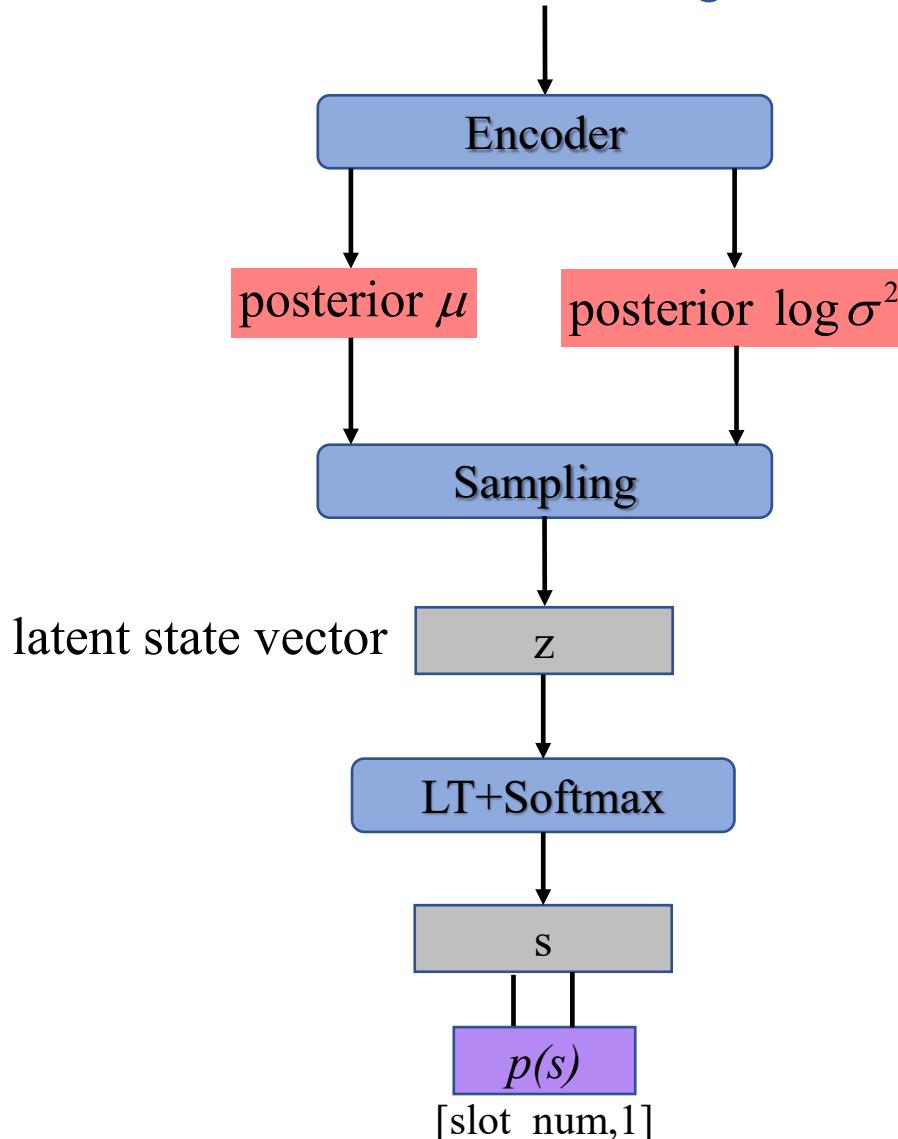
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



For each candidate in {train, Chicago, Dallas, Wednesday}

oh 0 0 0 1 ... 0 0 0 0

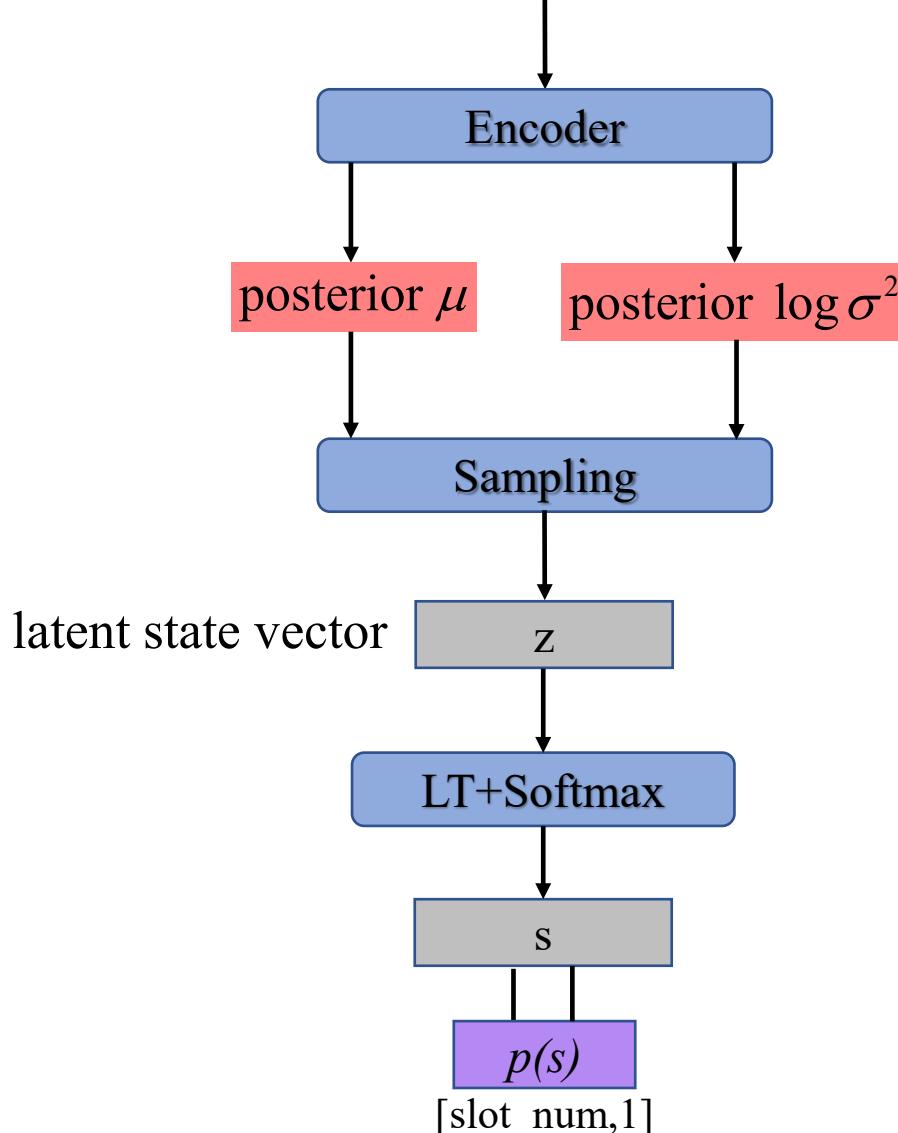
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



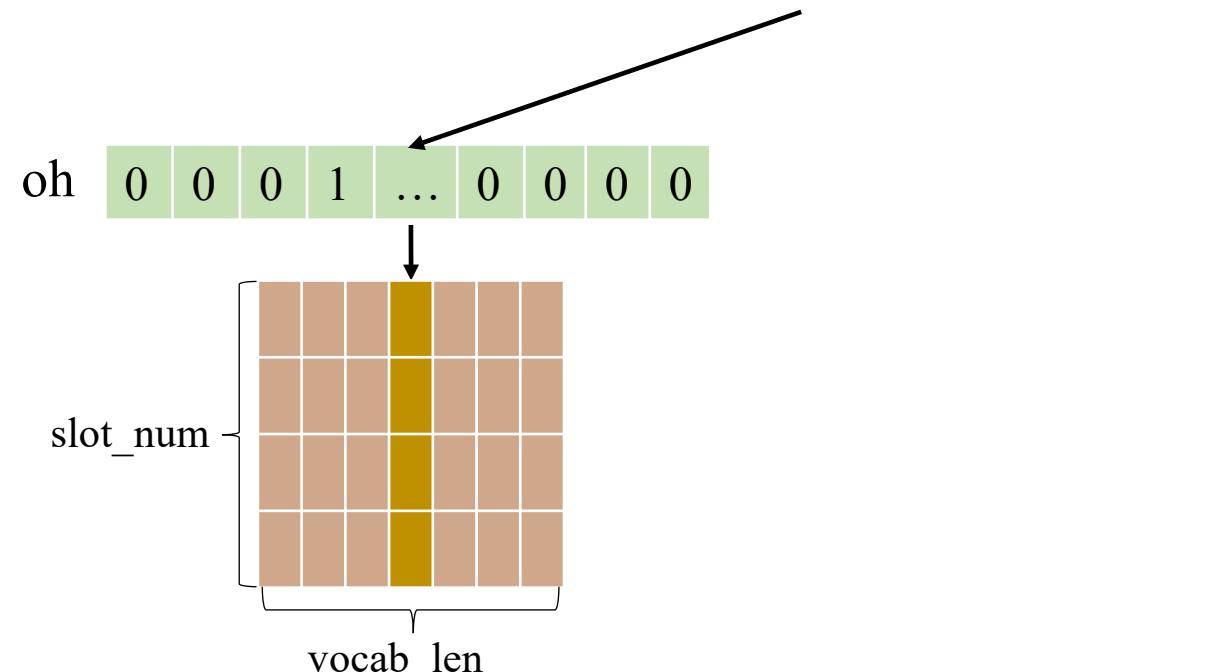
For each candidate in {train, Chicago, Dallas, Wednesday}

oh 0 0 0 1 ... 0 0 0 0

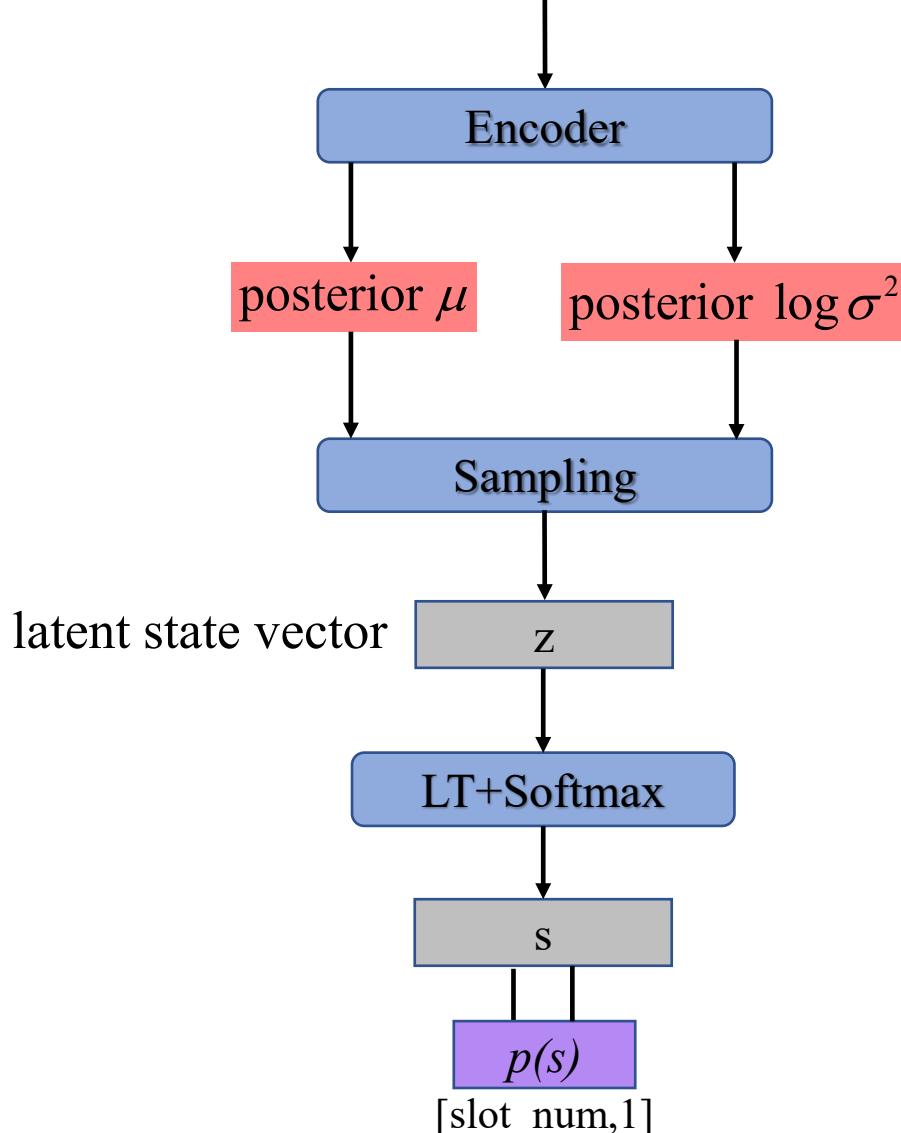
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



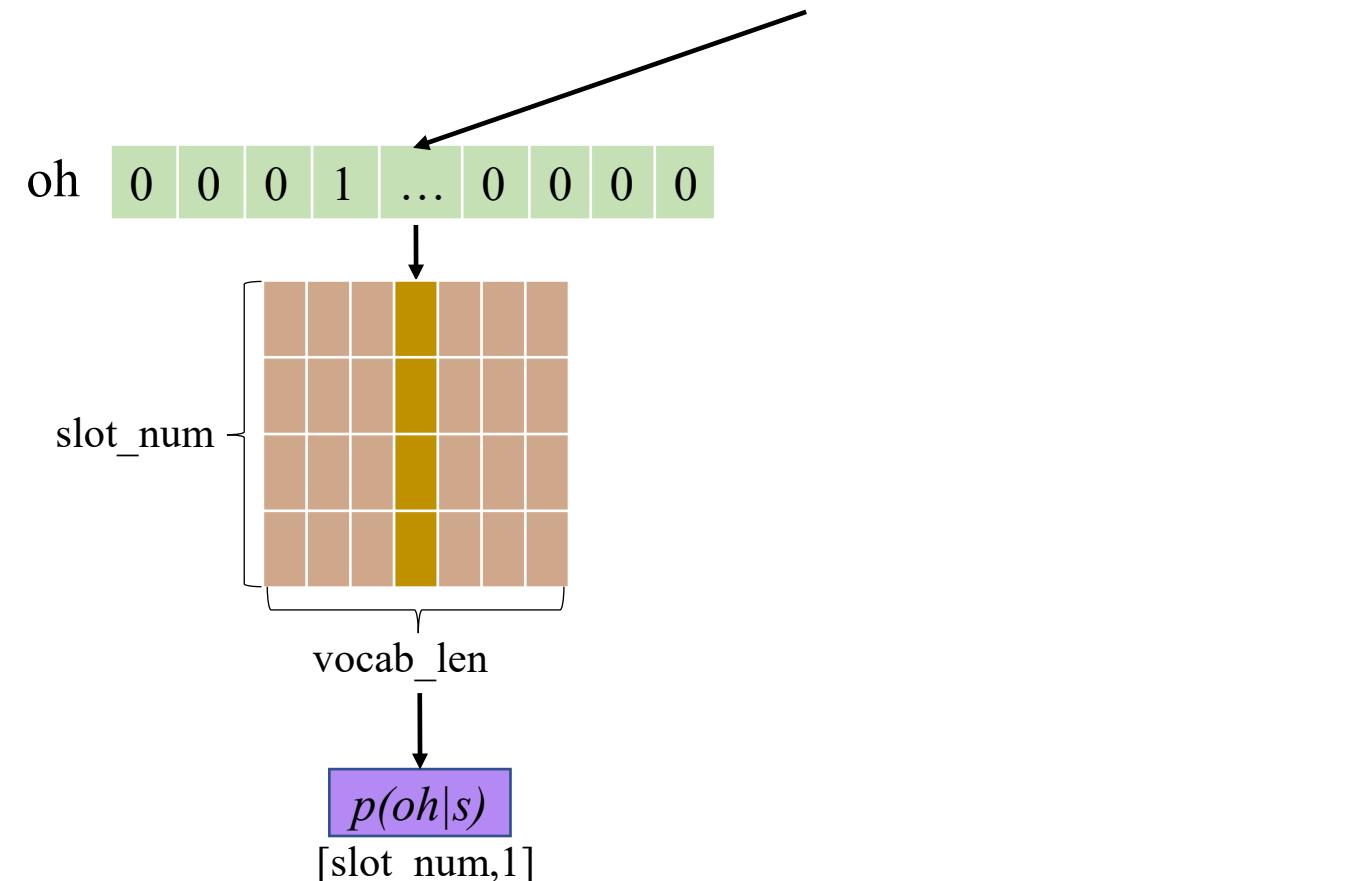
For each candidate in {train, Chicago, Dallas, Wednesday}



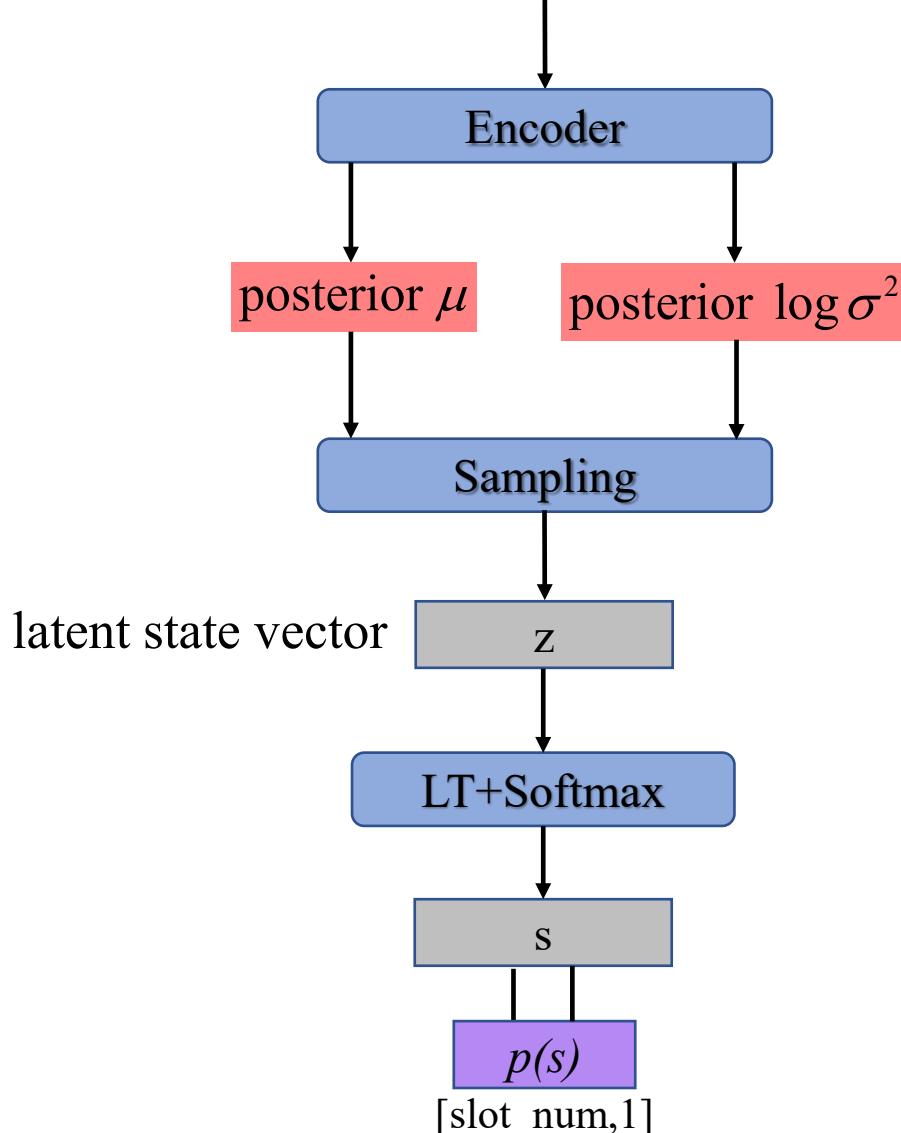
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



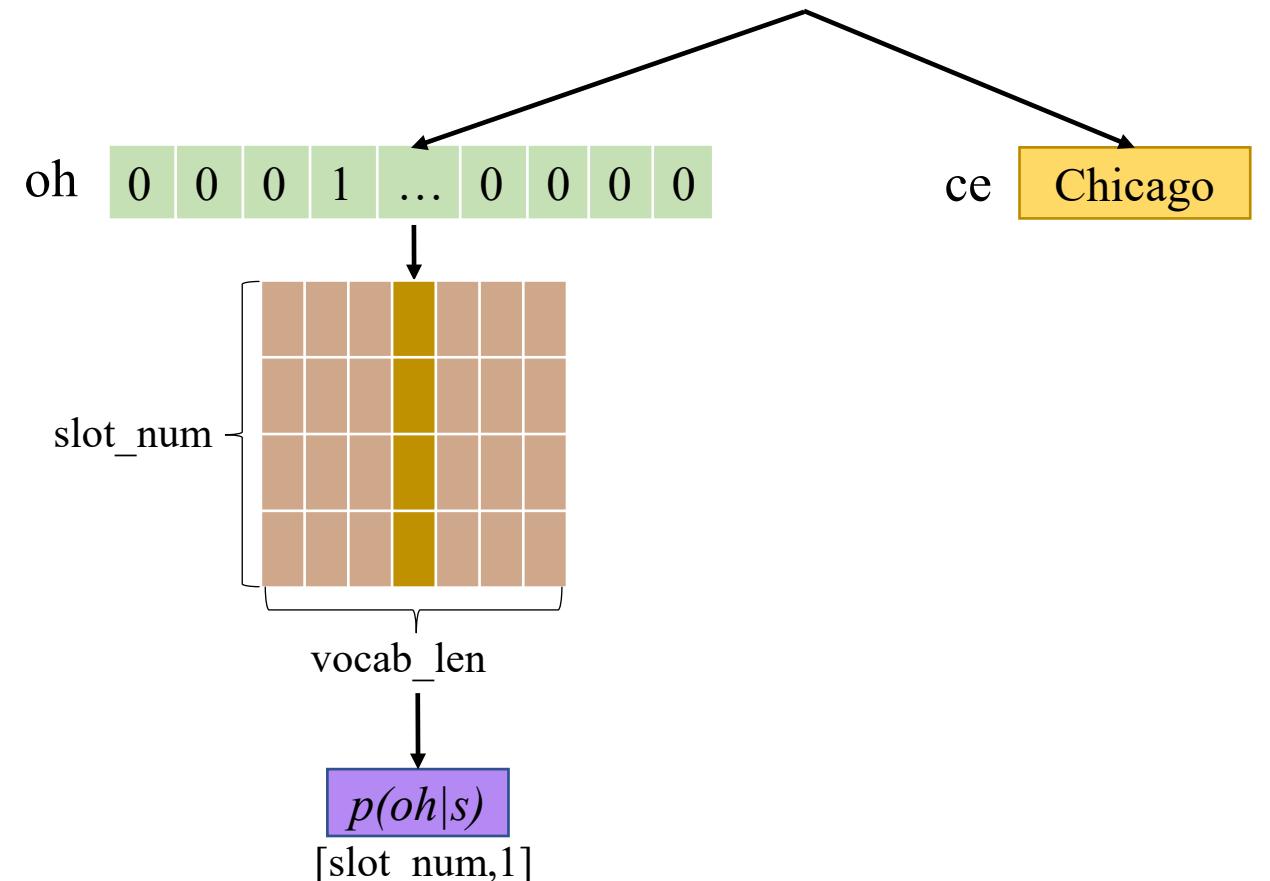
For each candidate in {train, Chicago, Dallas, Wednesday}

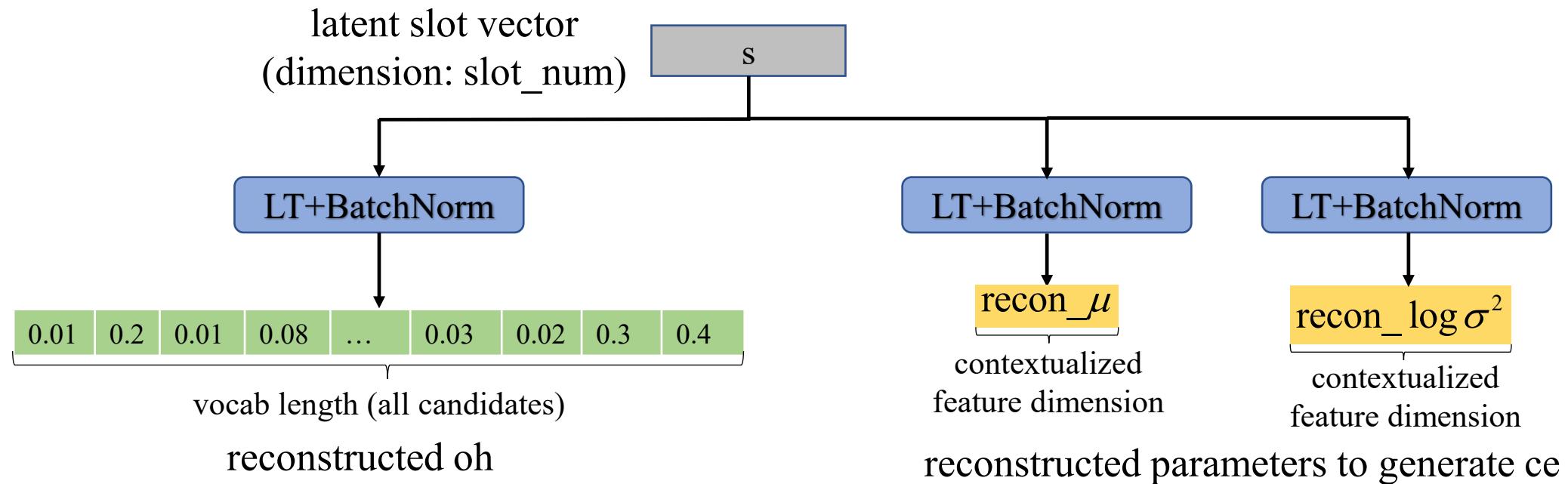


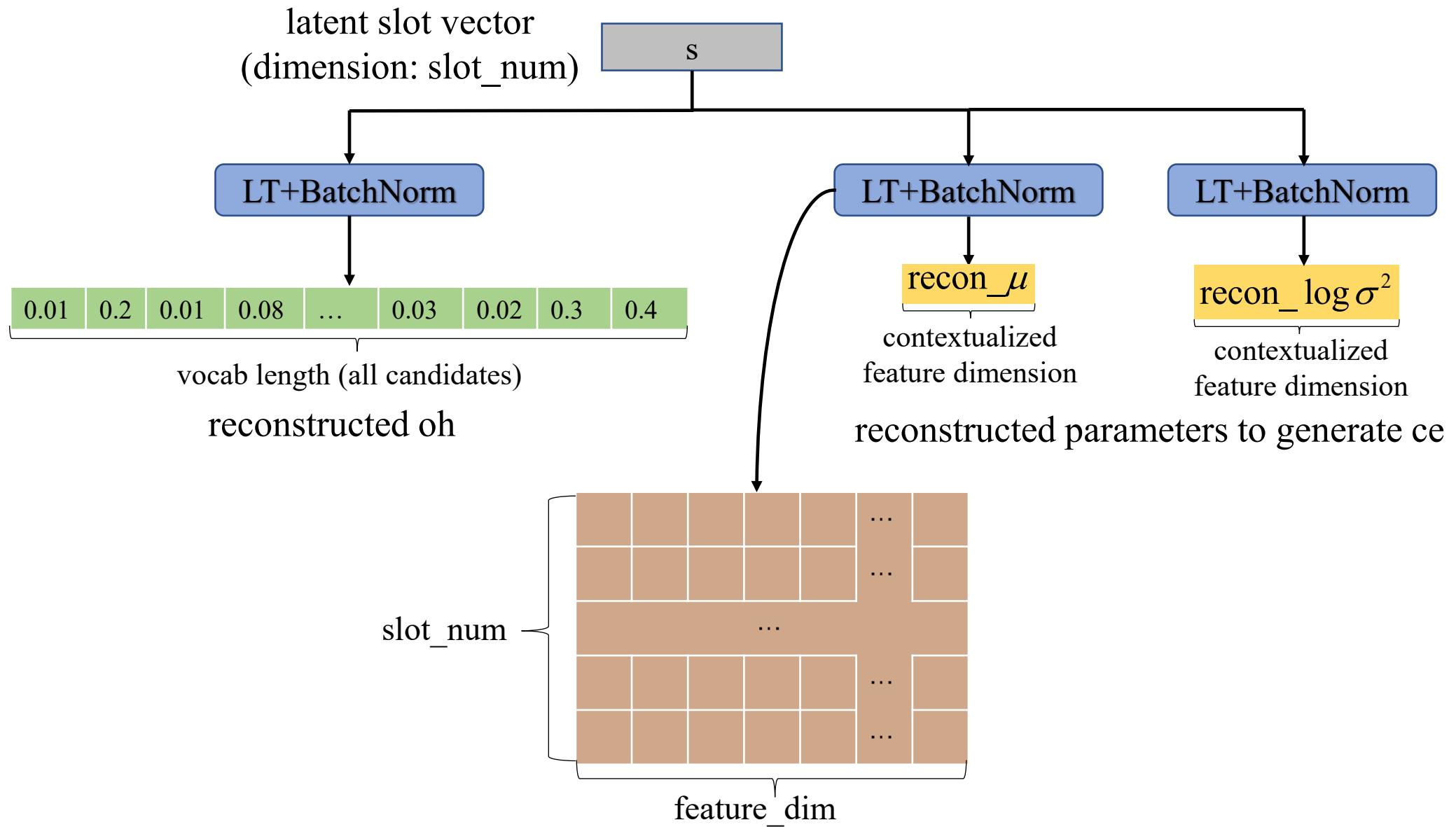
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.

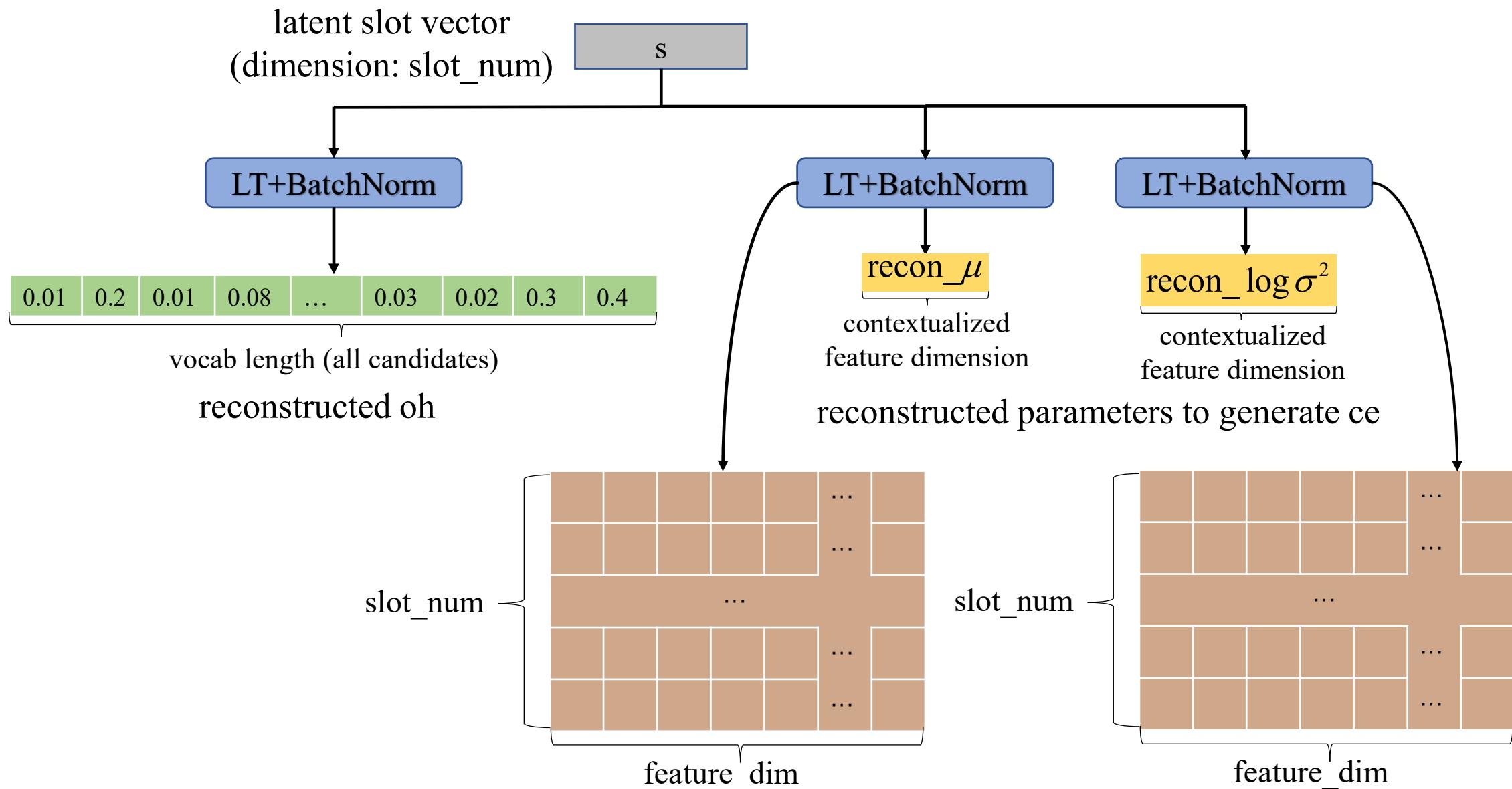


For each candidate in {train, Chicago, Dallas, Wednesday}









# CHAPTER 2 What does the model learn ?

s: slot vector

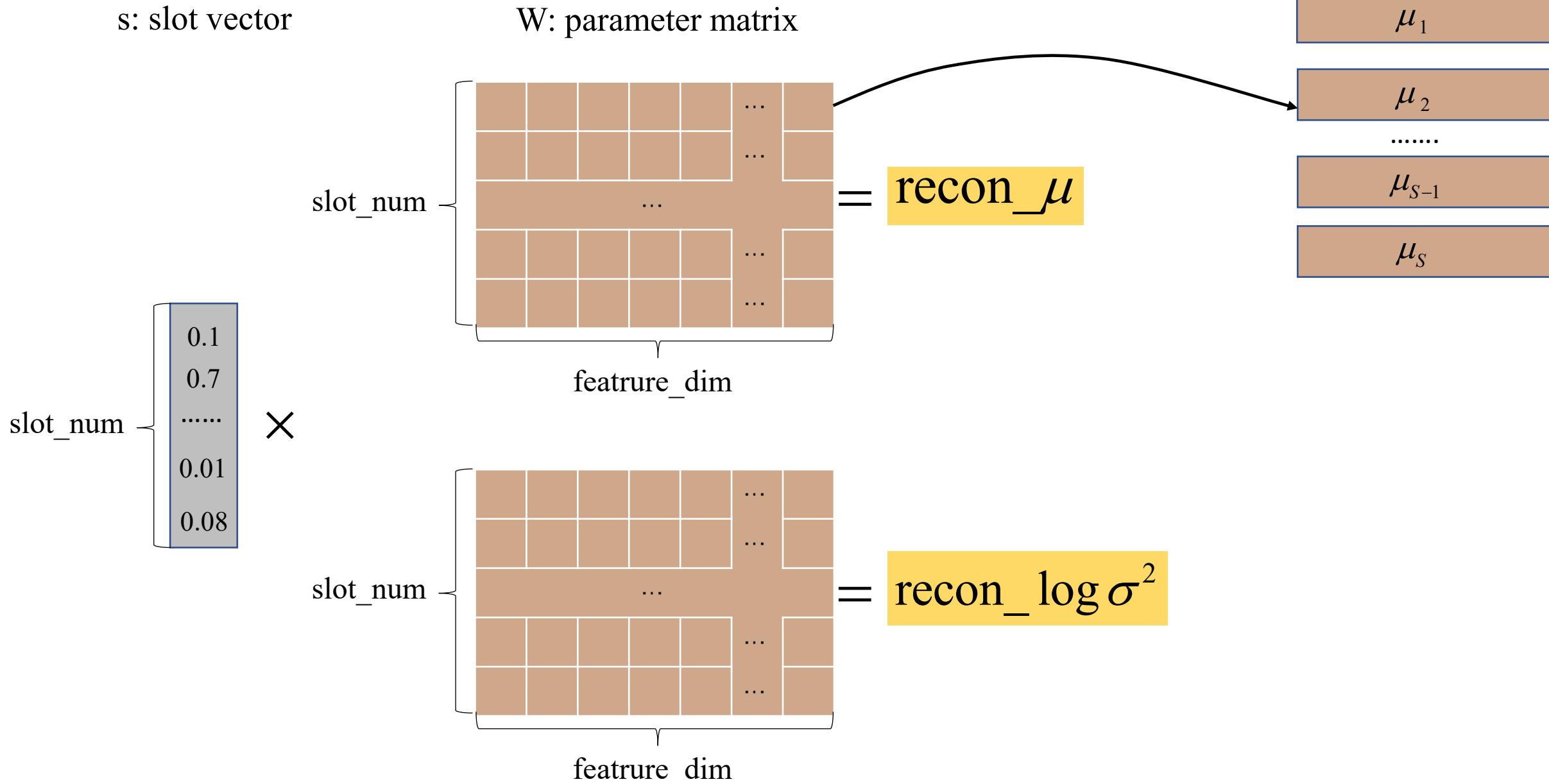
W: parameter matrix

$$\begin{array}{c} \text{slot\_num} \\ \left\{ \begin{array}{l} 0.1 \\ 0.7 \\ \cdots \\ 0.01 \\ 0.08 \end{array} \right. \times \end{array} \quad \begin{array}{c} \text{slot\_num} \\ \left\{ \begin{array}{l} \text{feature\_dim} \\ \cdots \\ \text{feature\_dim} \end{array} \right. = \text{recon\_}\mu \end{array}$$
$$\begin{array}{c} \text{slot\_num} \\ \left\{ \begin{array}{l} \text{feature\_dim} \\ \cdots \\ \text{feature\_dim} \end{array} \right. = \text{recon\_log } \sigma^2 \end{array}$$

## CHAPTER 2 What does the model learn ?

s: slot vector

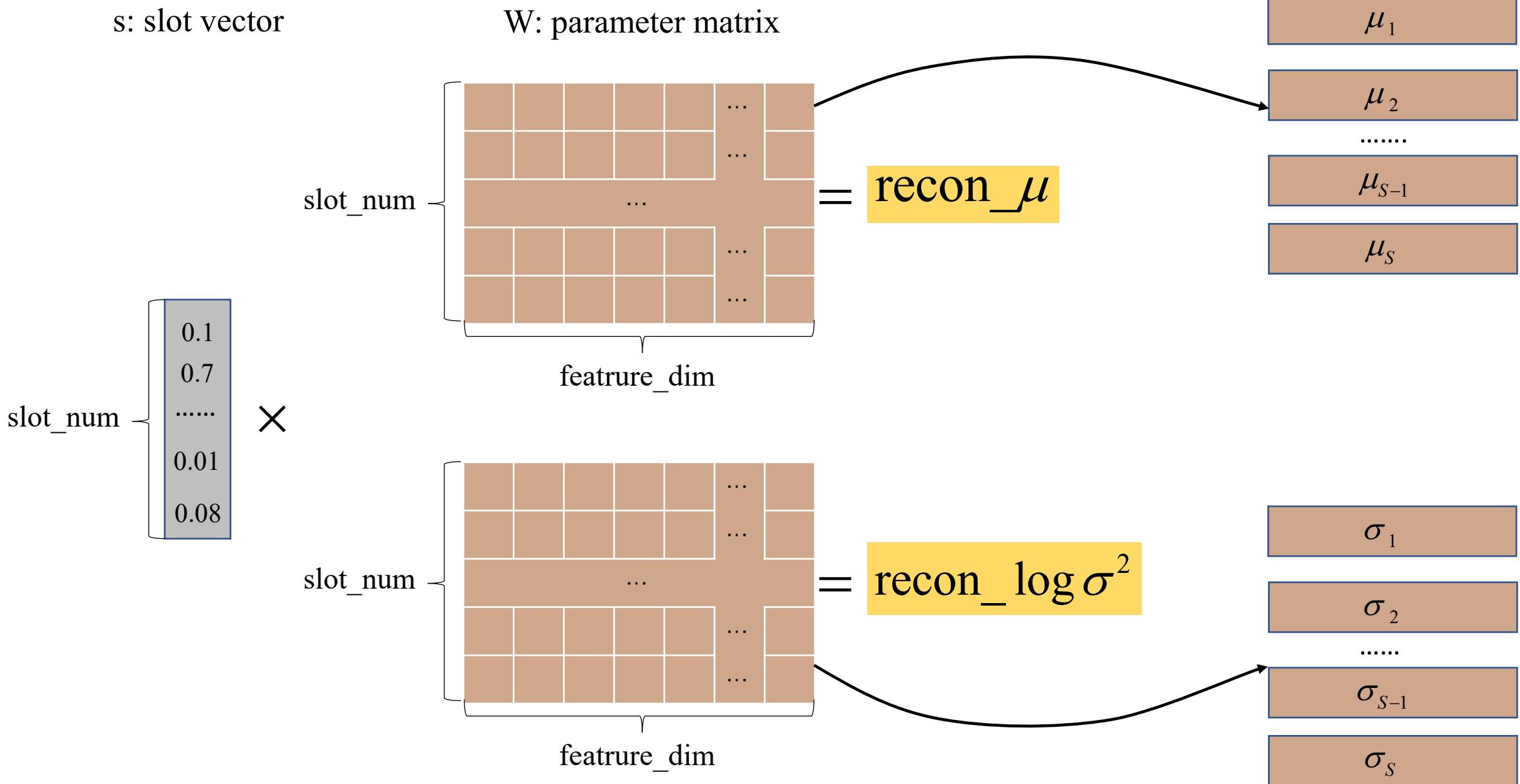
## W: parameter matrix



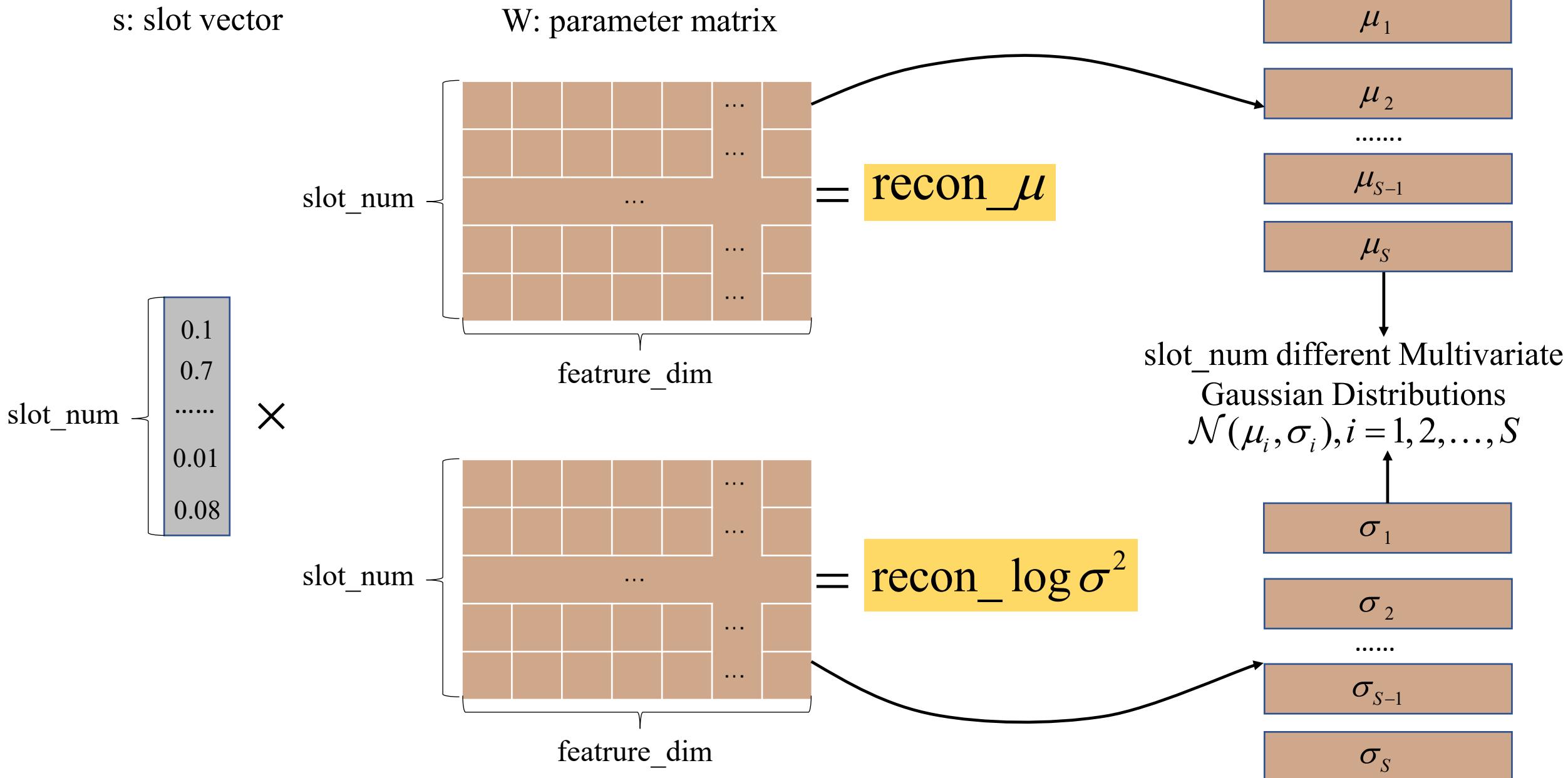
## CHAPTER 2 What does the model learn ?

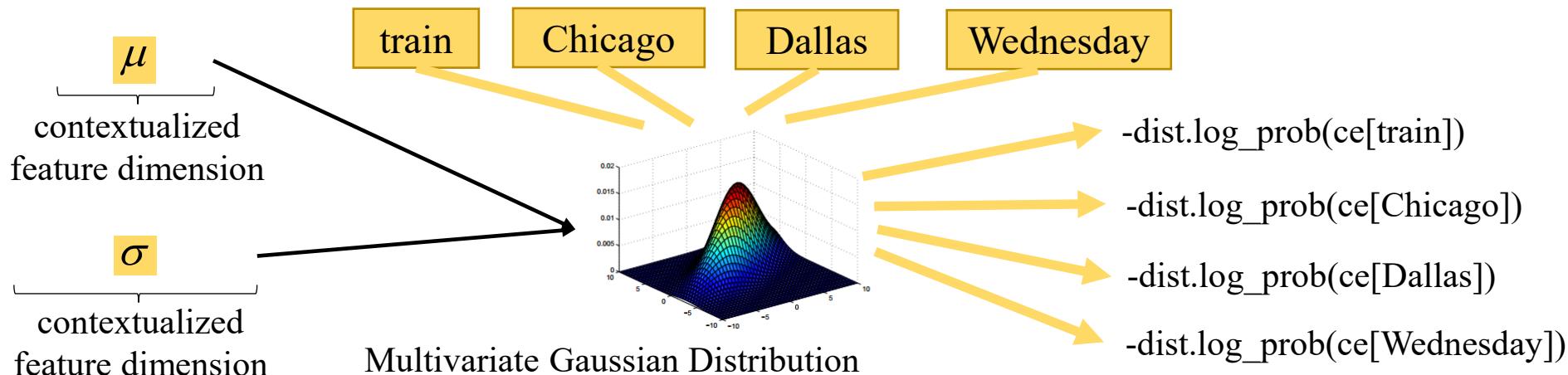
s: slot vector

## W: parameter matrix

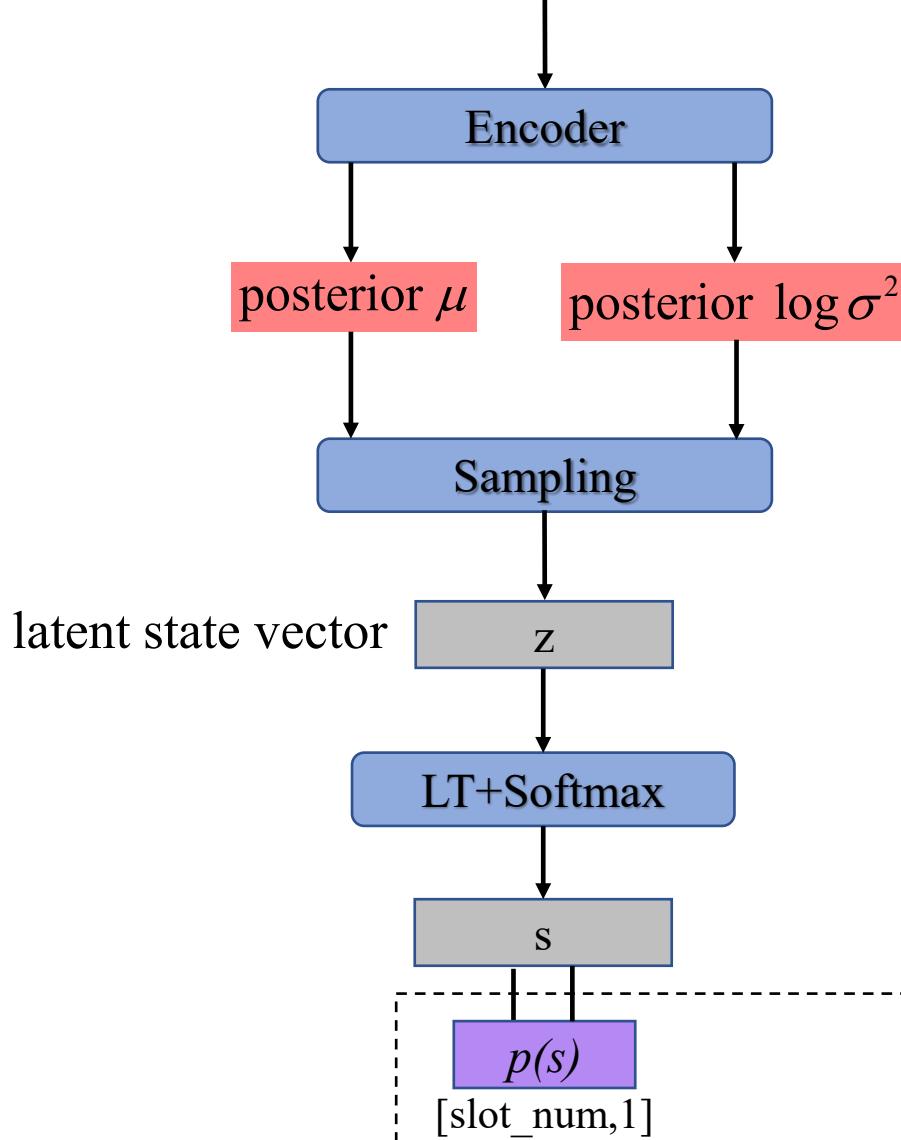


# CHAPTER 2 What does the model learn ?

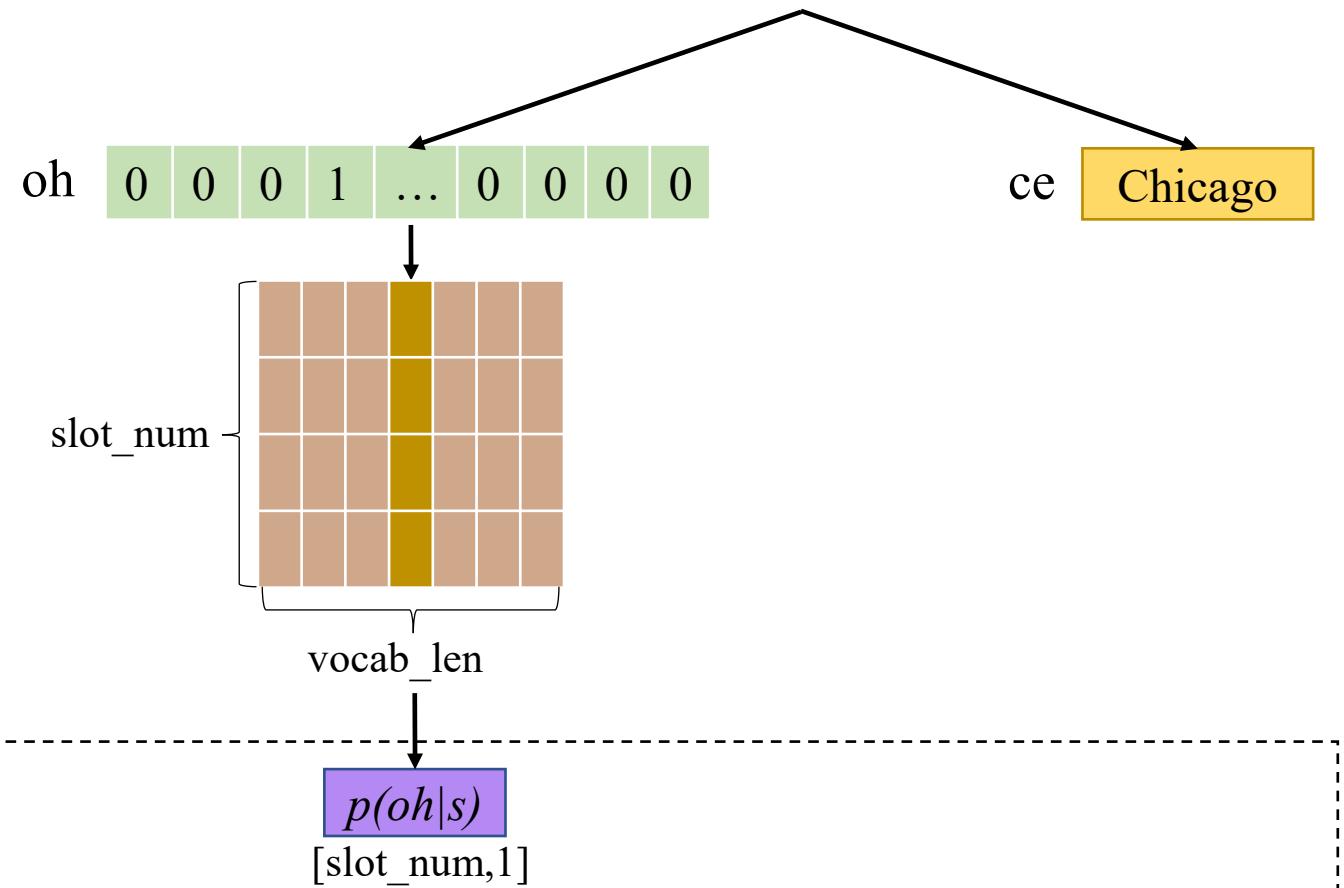




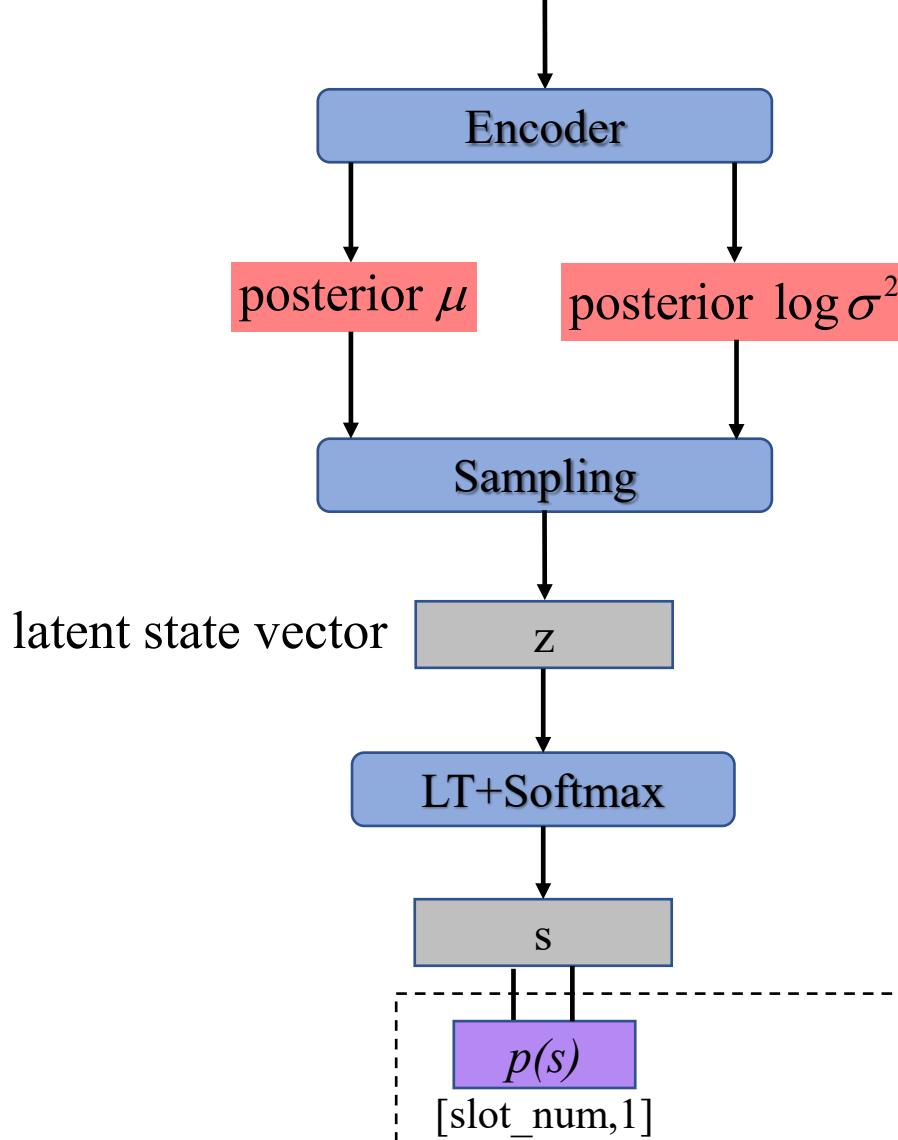
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



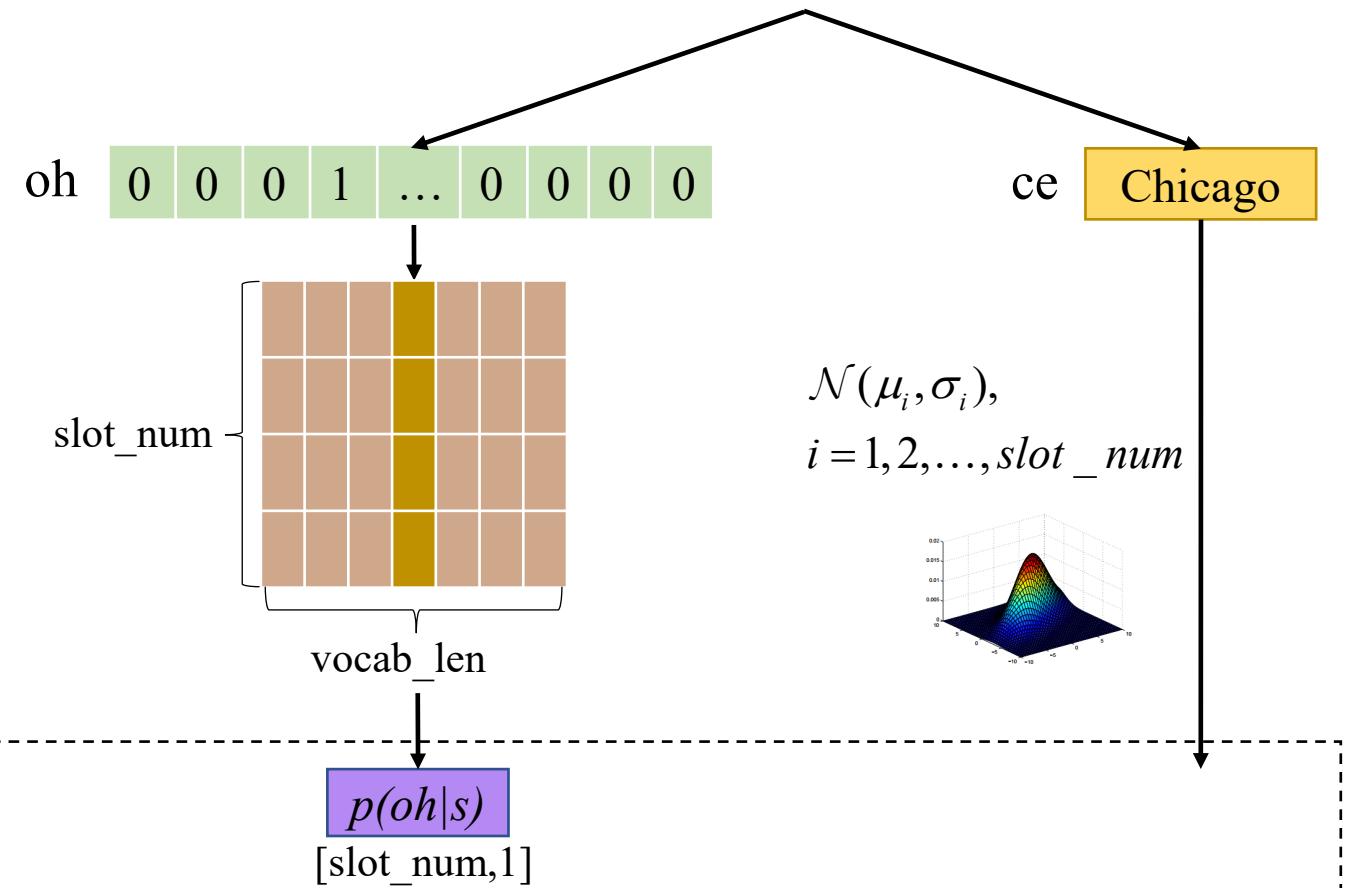
For each candidate in {train, Chicago, Dallas, Wednesday}



I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.

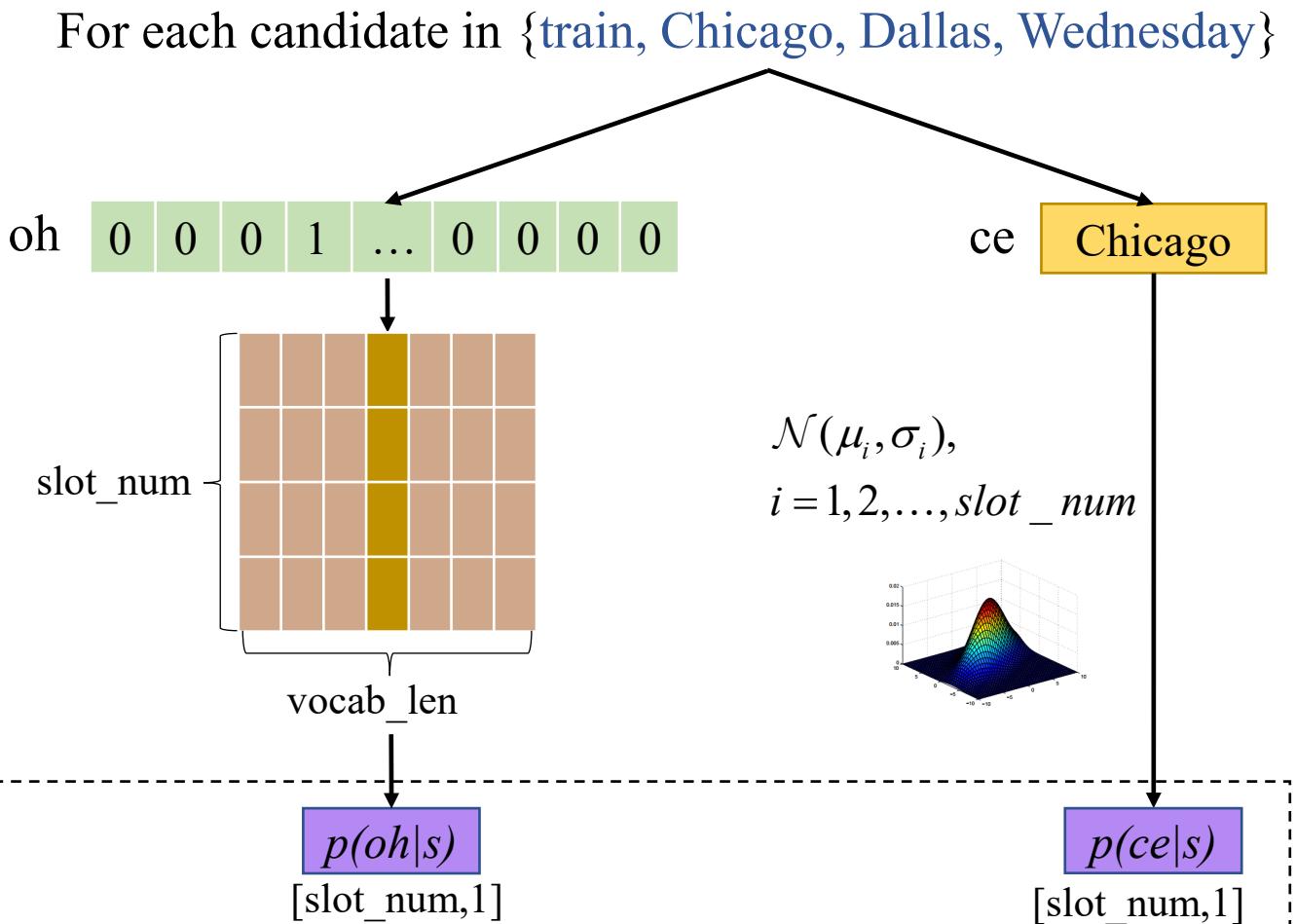
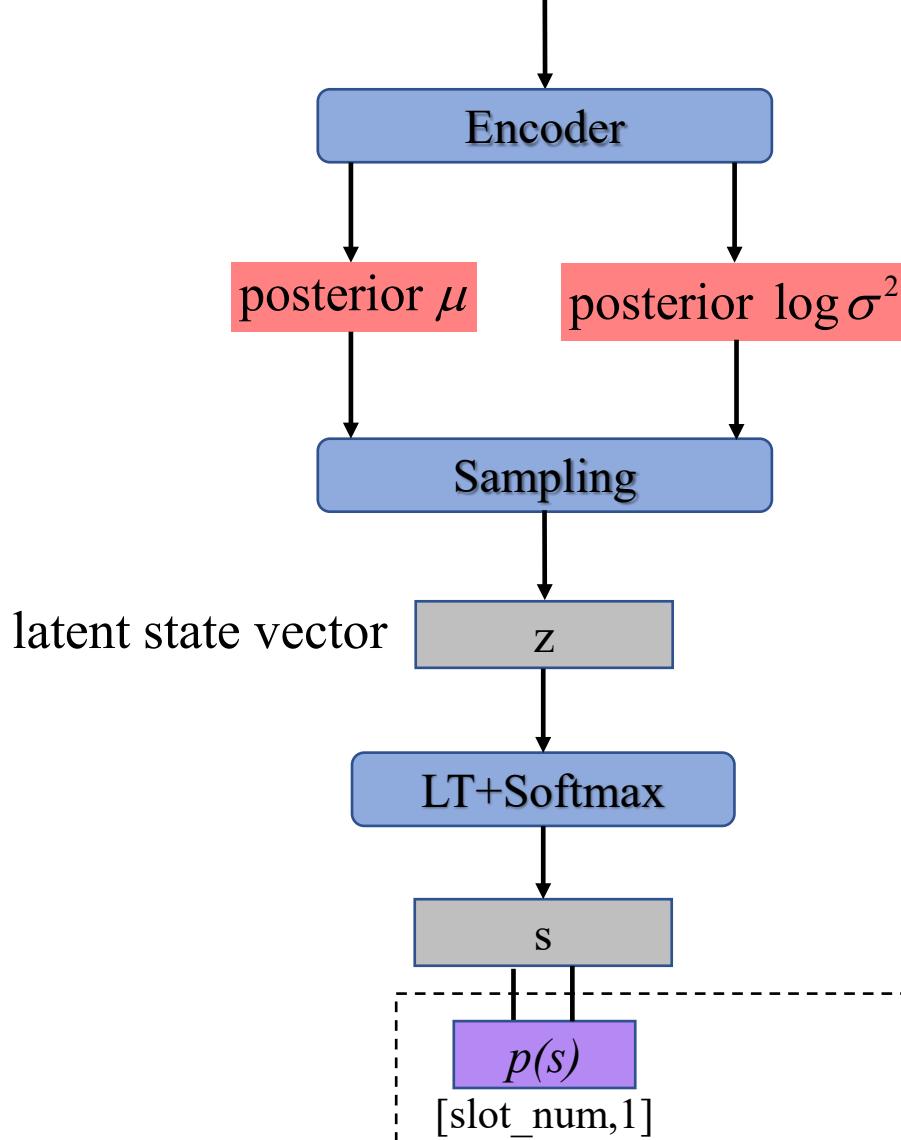


For each candidate in {train, Chicago, Dallas, Wednesday}

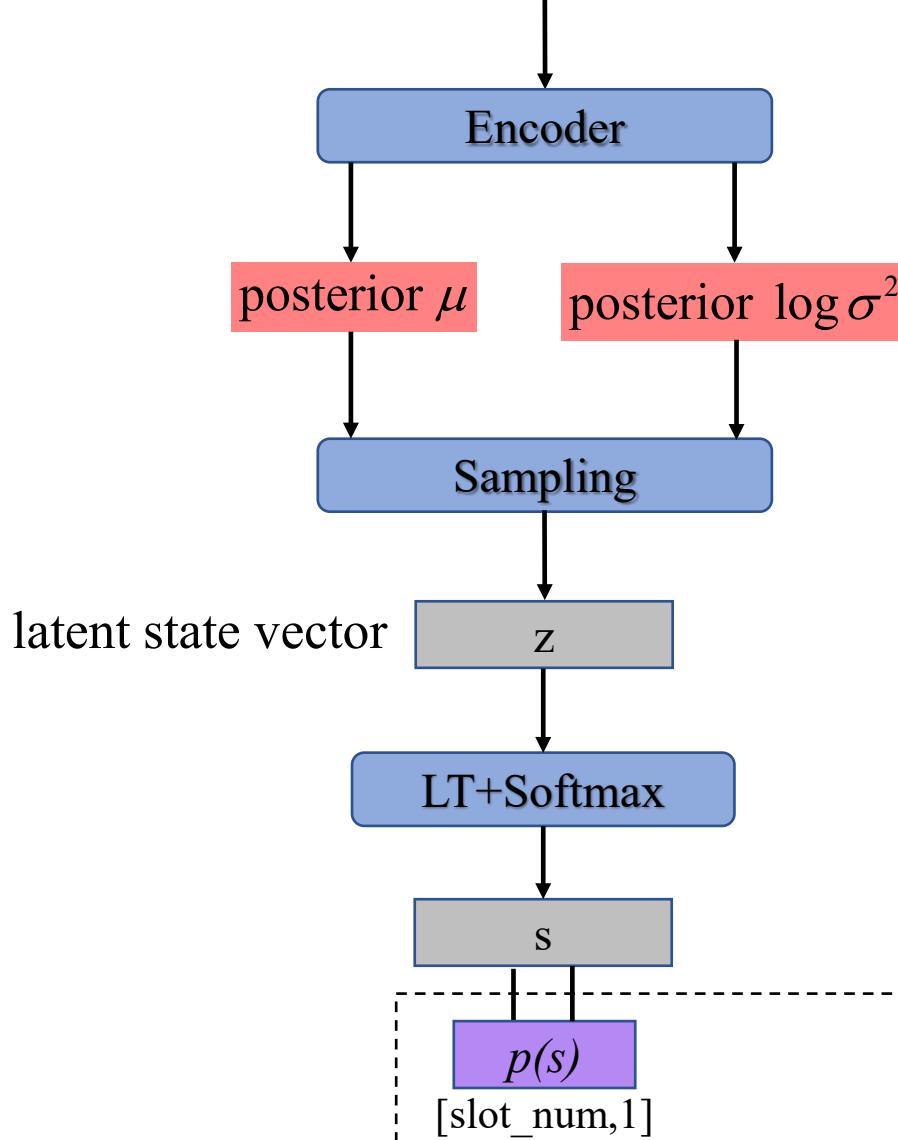


## CHAPTER 2 *DSI-base inference*

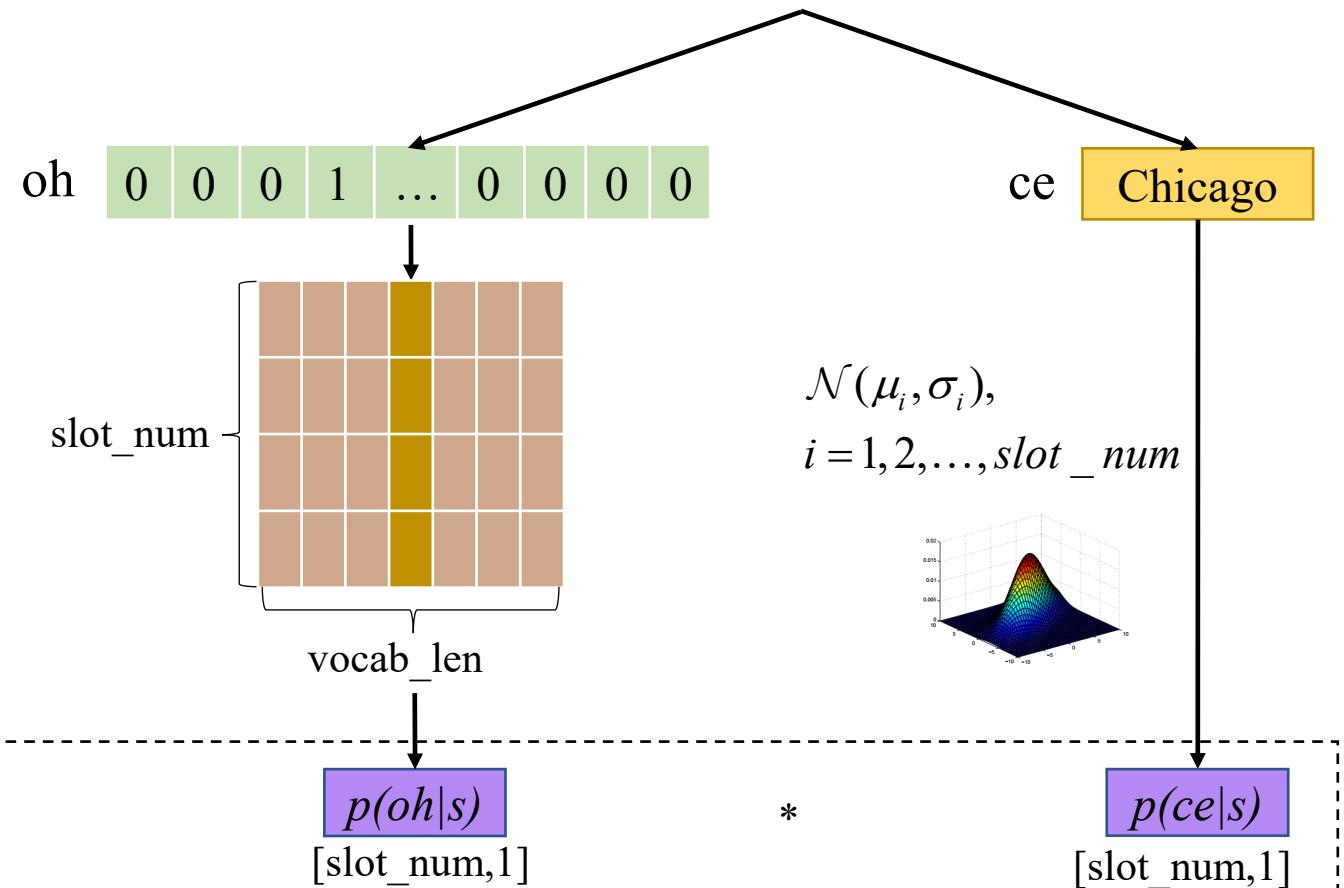
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



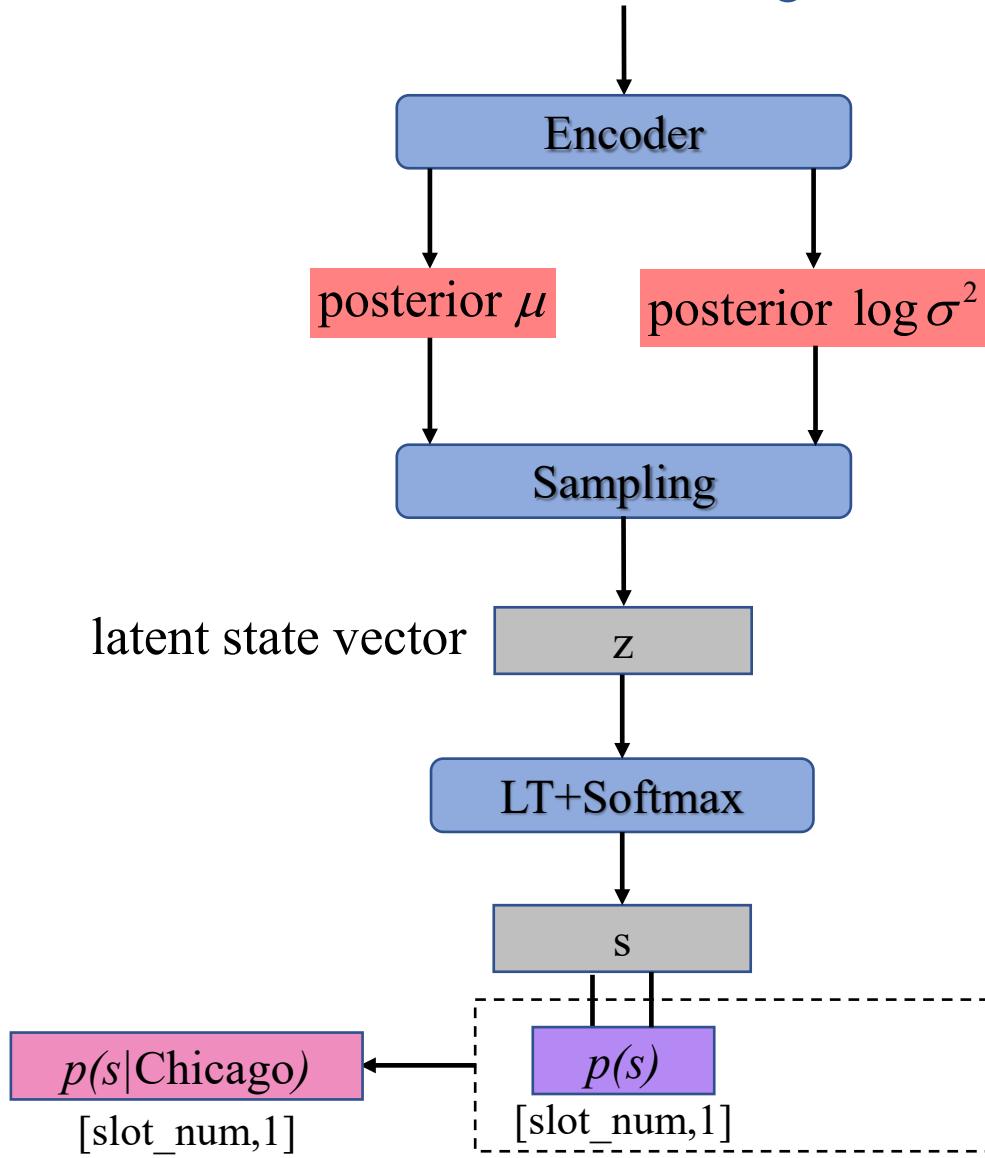
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



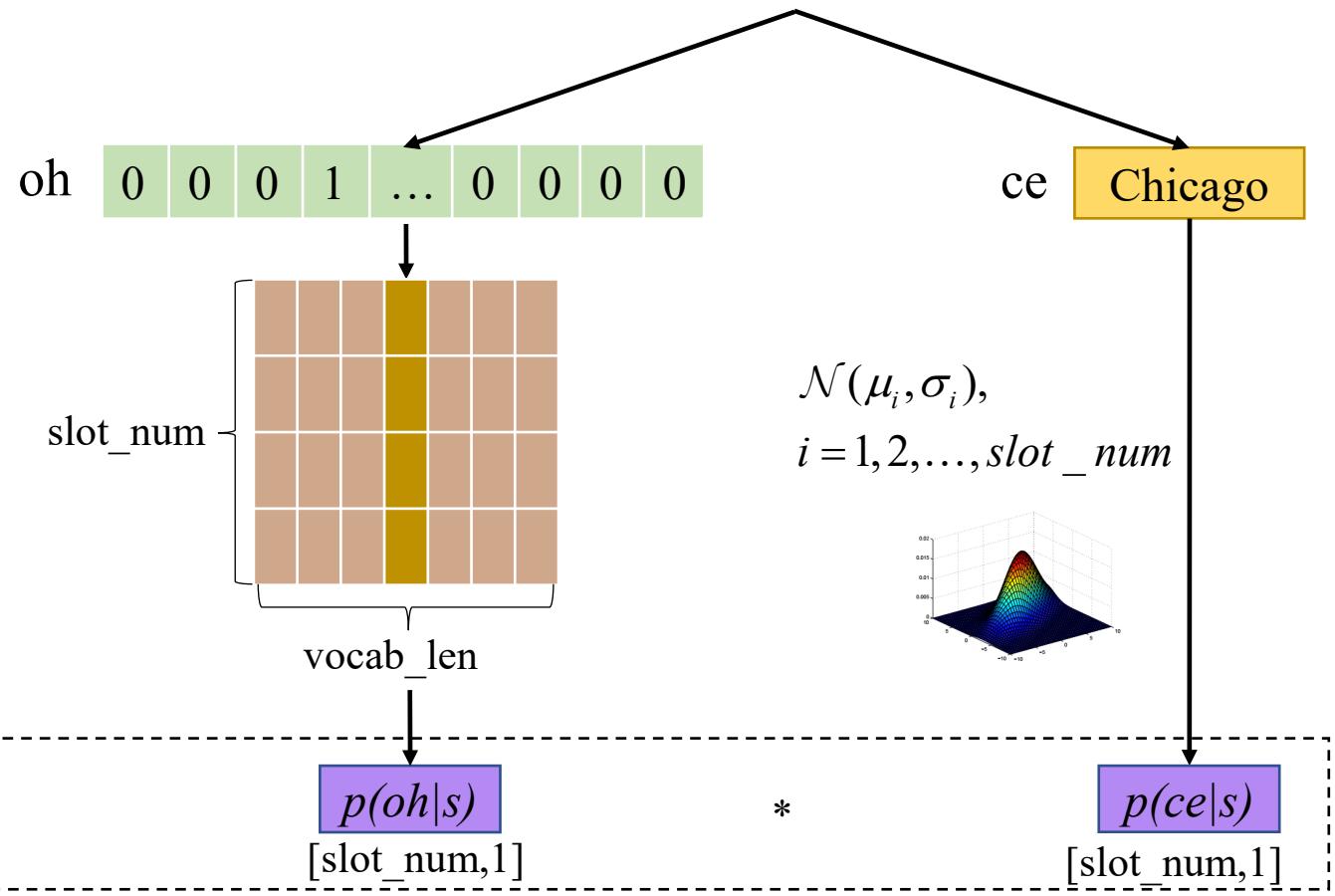
For each candidate in {train, Chicago, Dallas, Wednesday}



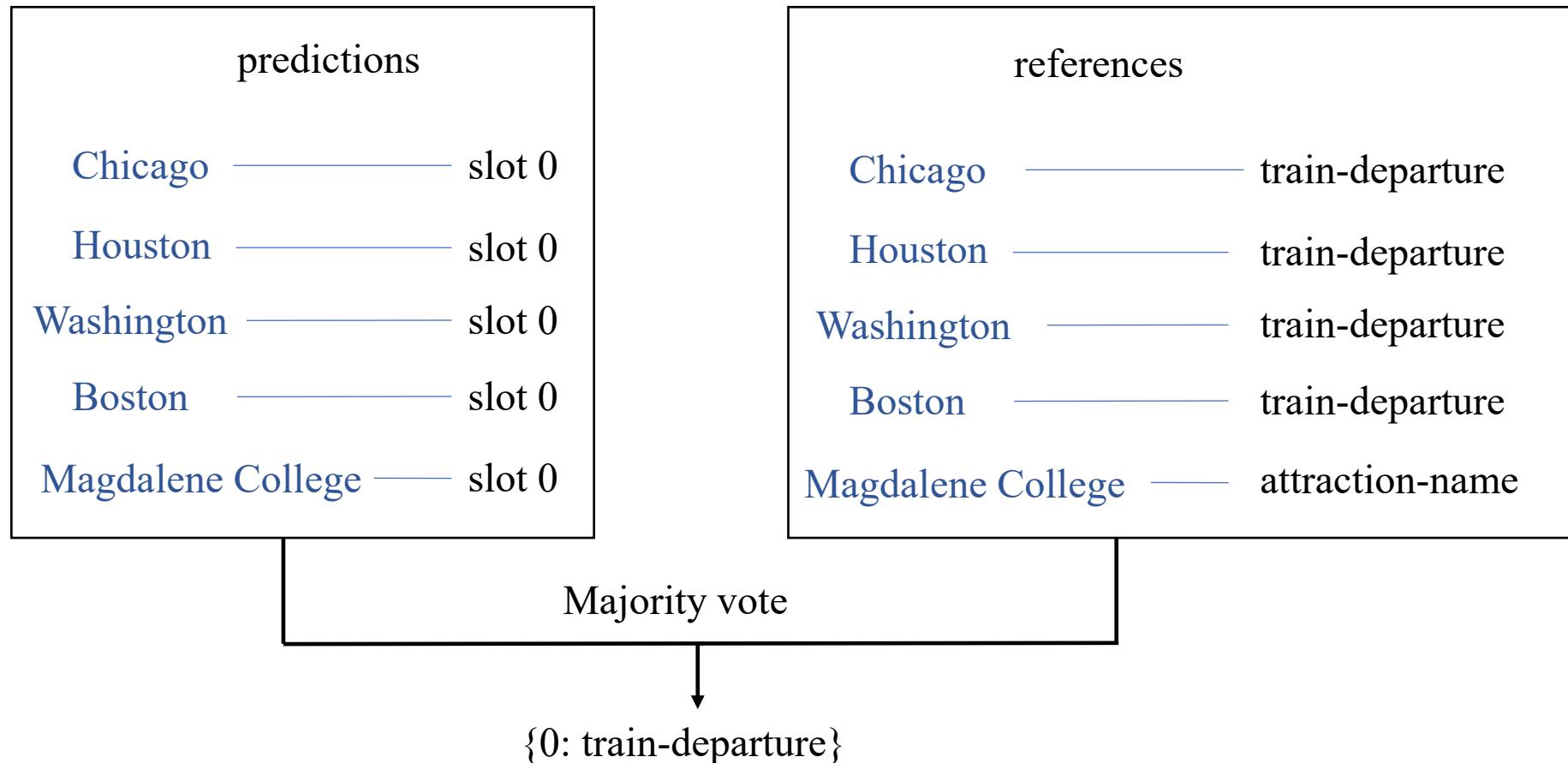
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



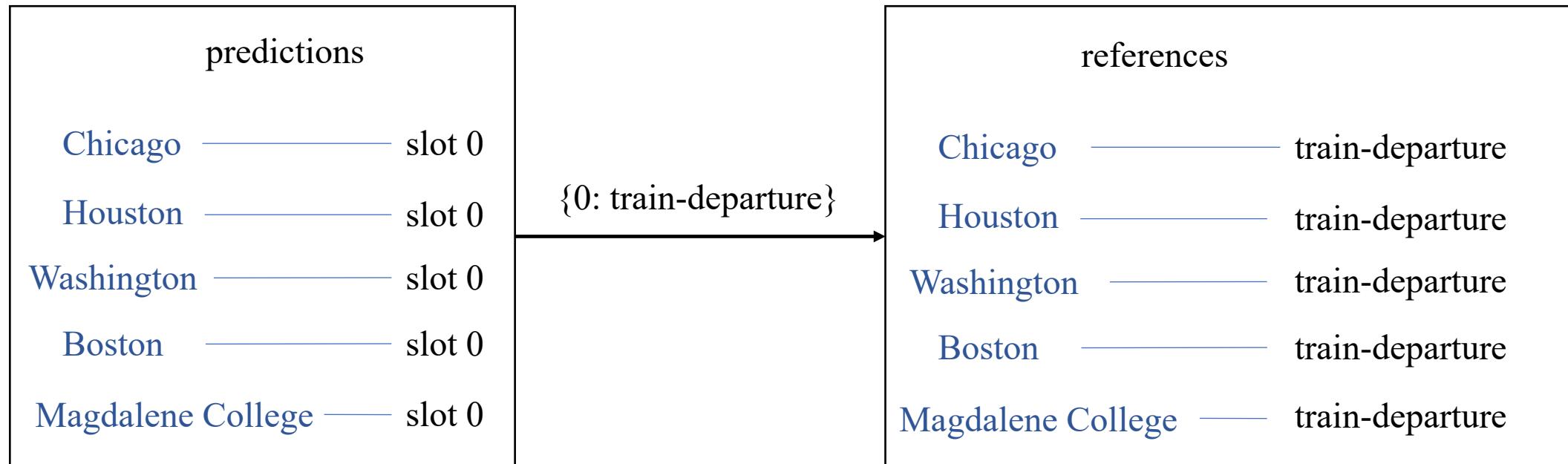
For each candidate in `{train, Chicago, Dallas, Wednesday}`



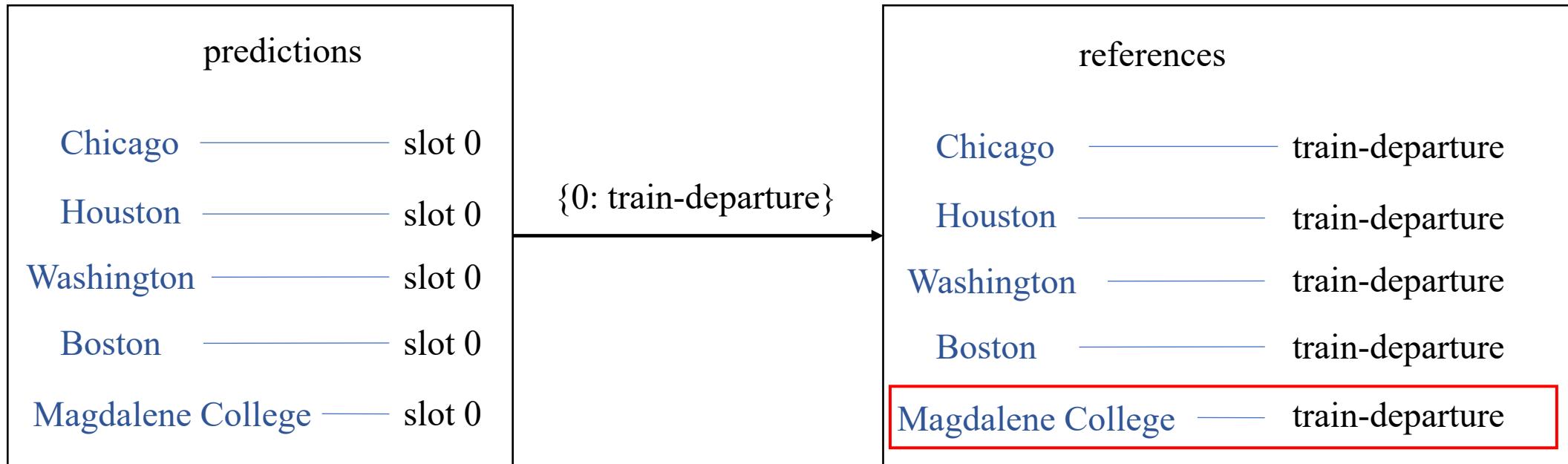
Mapping from slot indexes to labels?

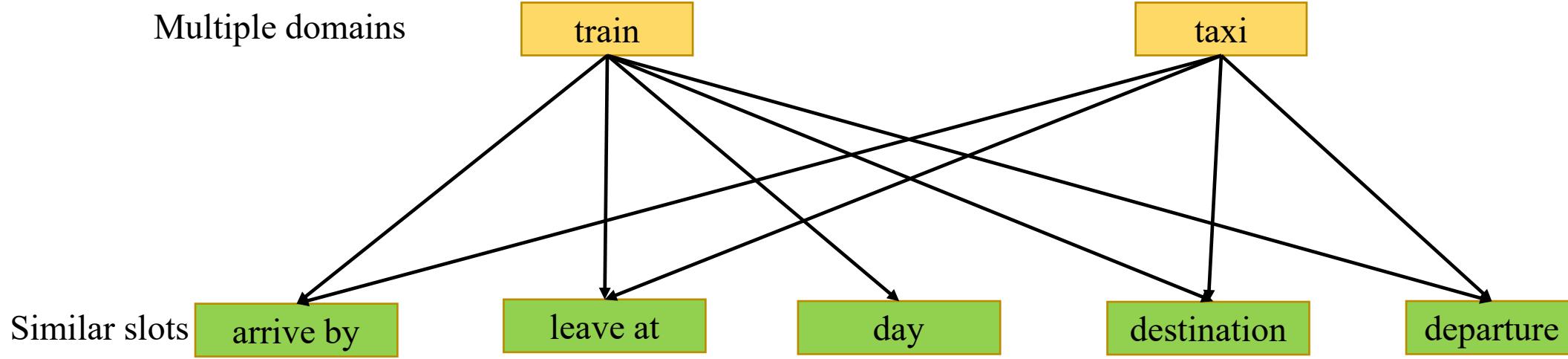


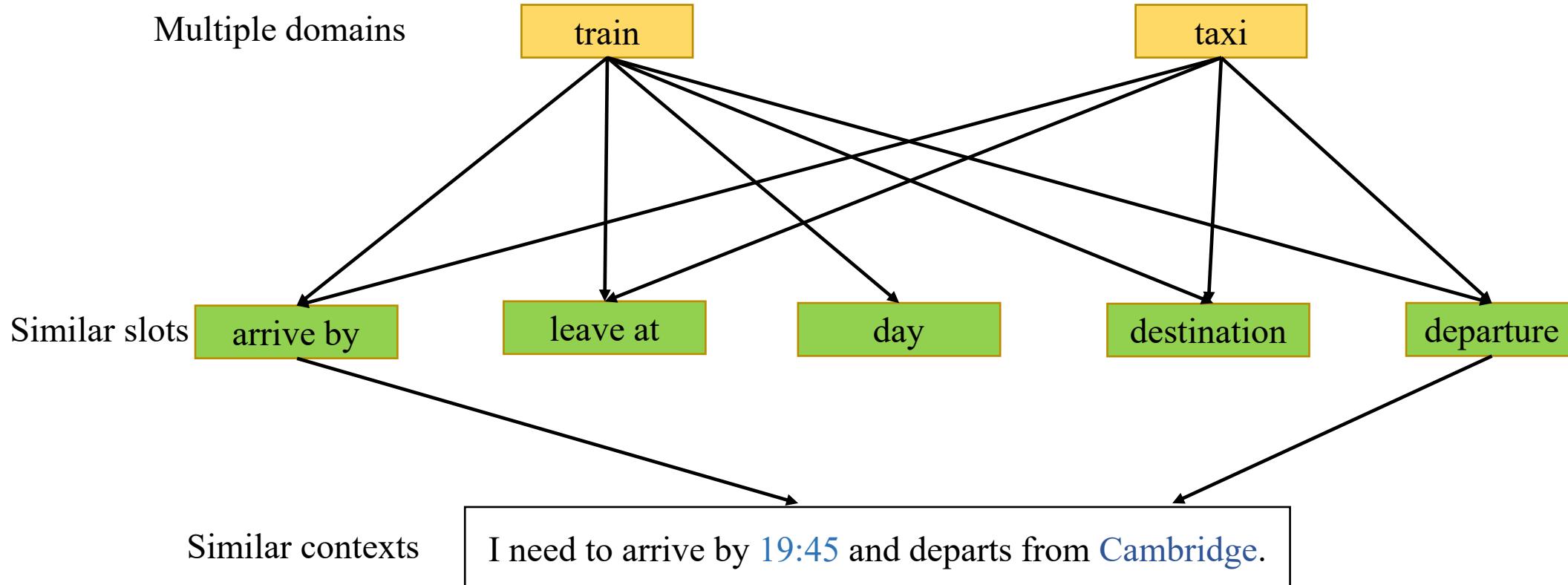
Mapping from slot indexes to labels?



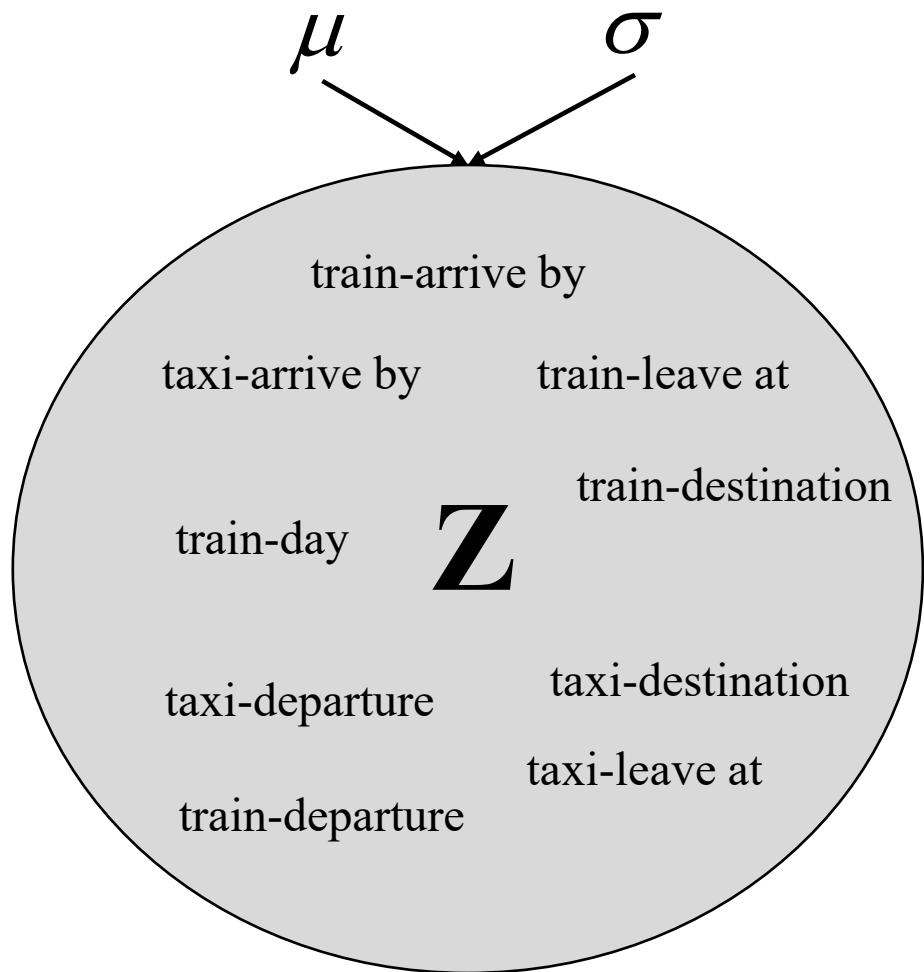
Mapping from slot indexes to labels?



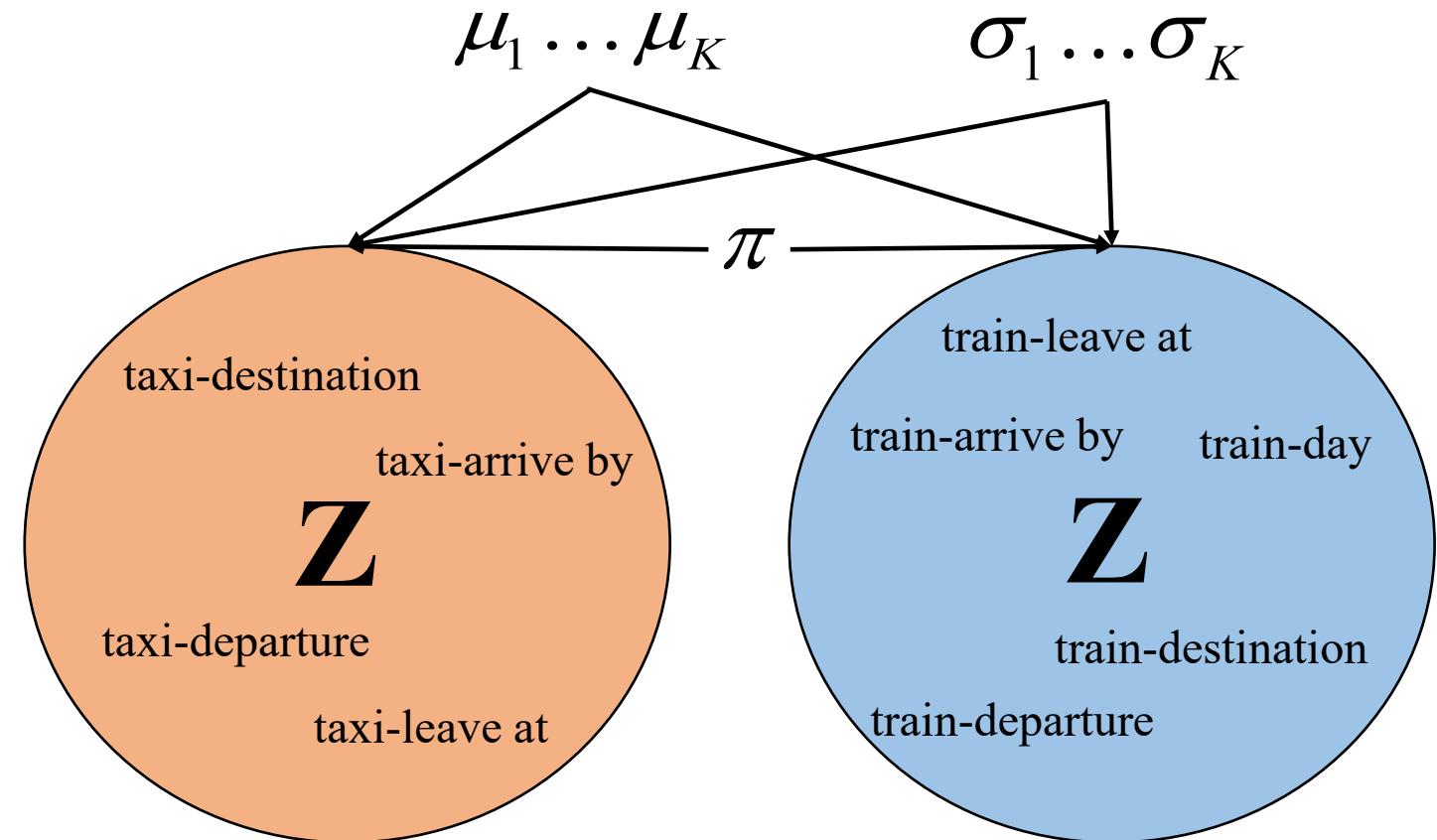




A single Gaussian prior

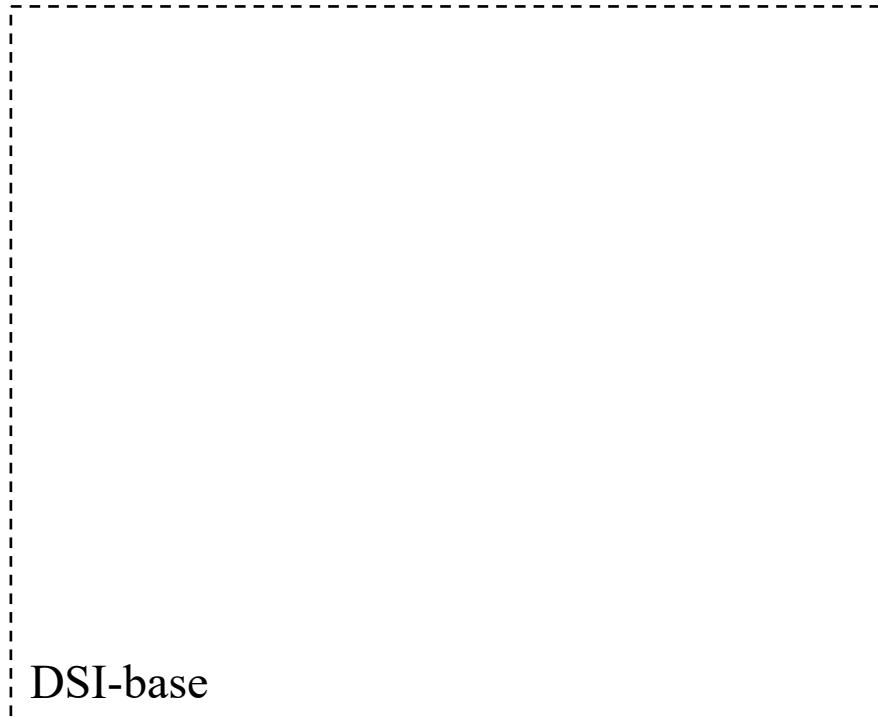


A Mixture-of-Gaussians prior

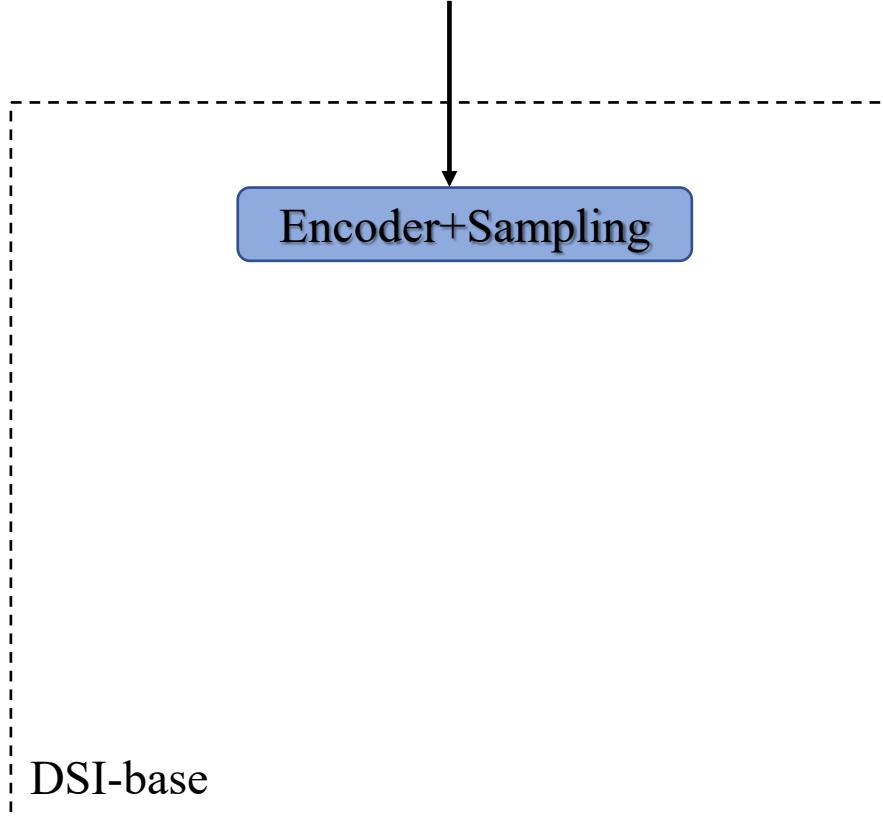


I need to take a [train](#) out of [Chicago](#), I will be leaving [Dallas](#) on [Wednesday](#).

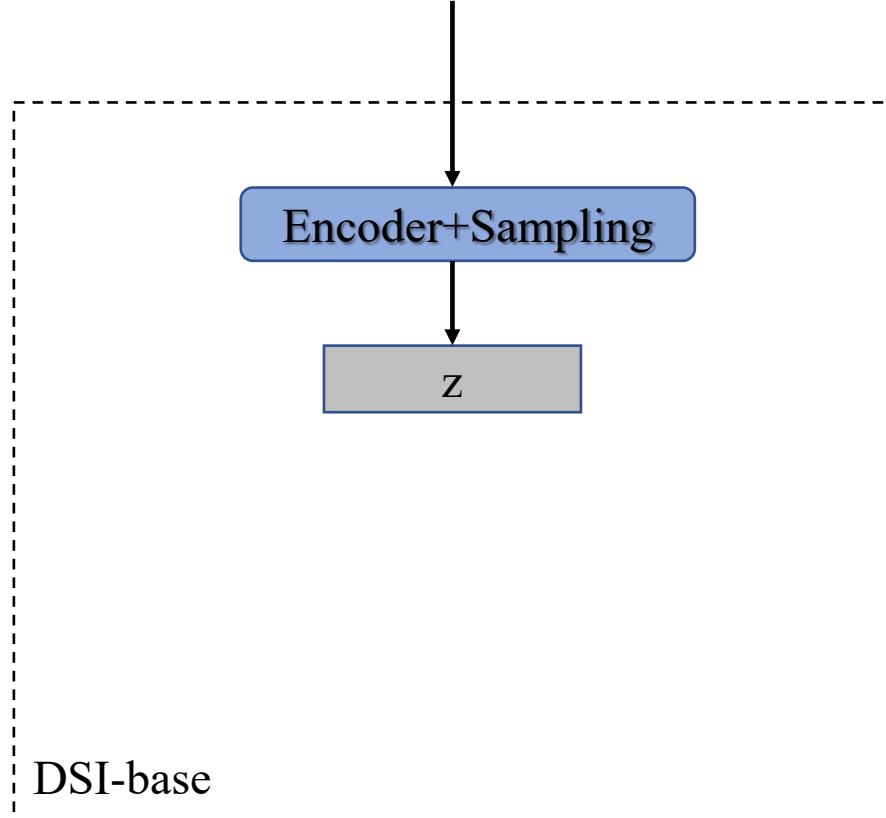
I need to take a [train](#) out of [Chicago](#), I will be leaving [Dallas](#) on [Wednesday](#).



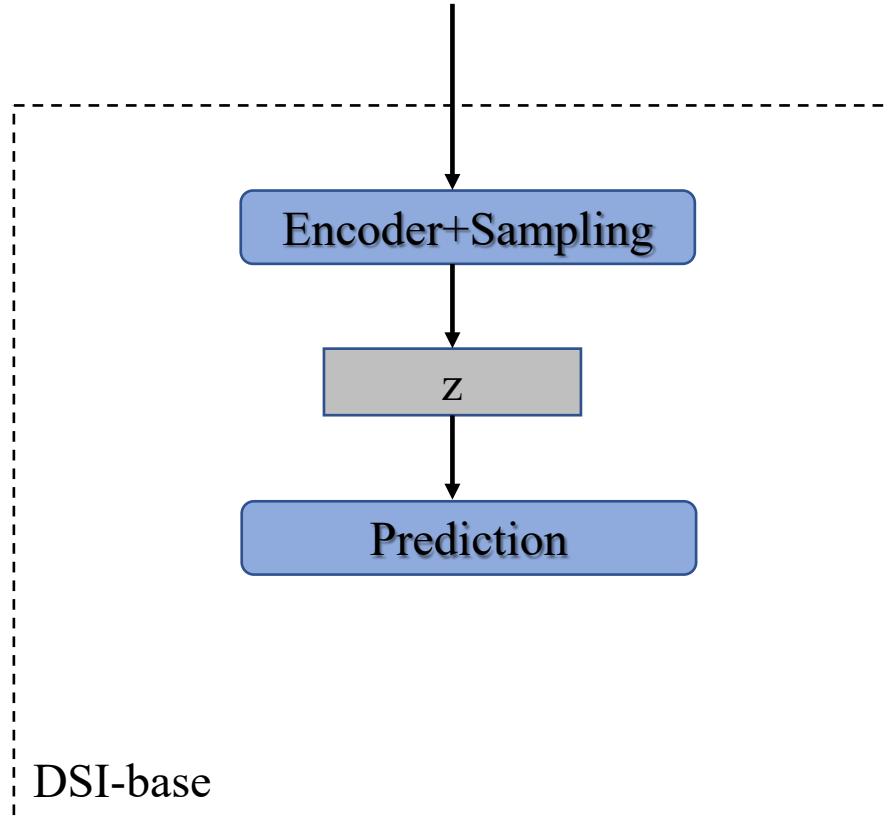
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



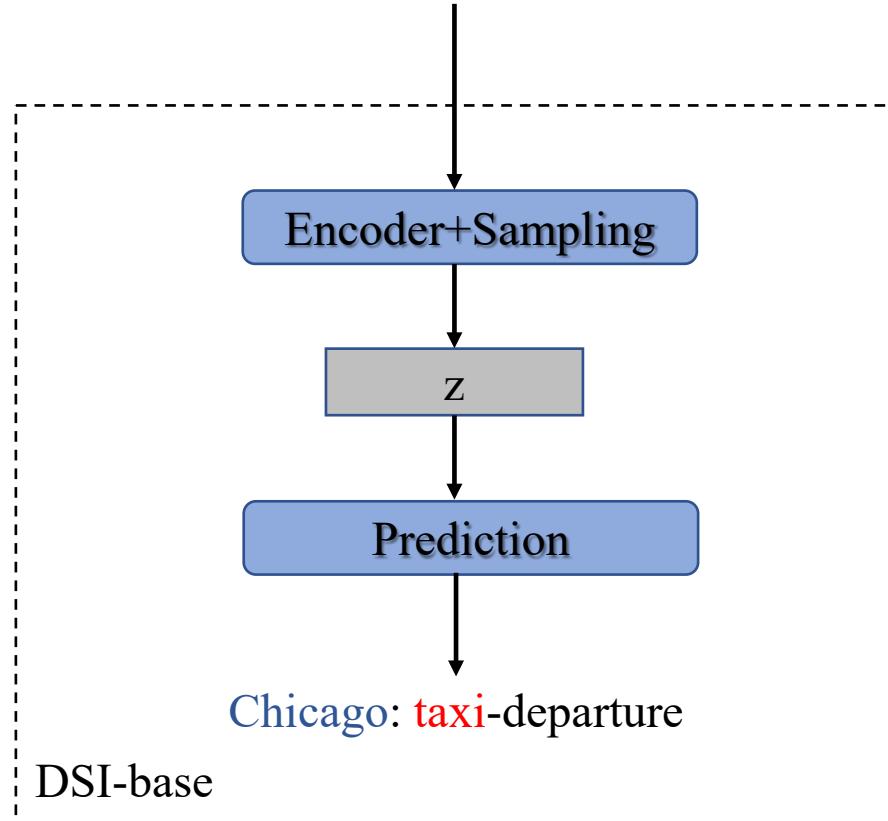
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



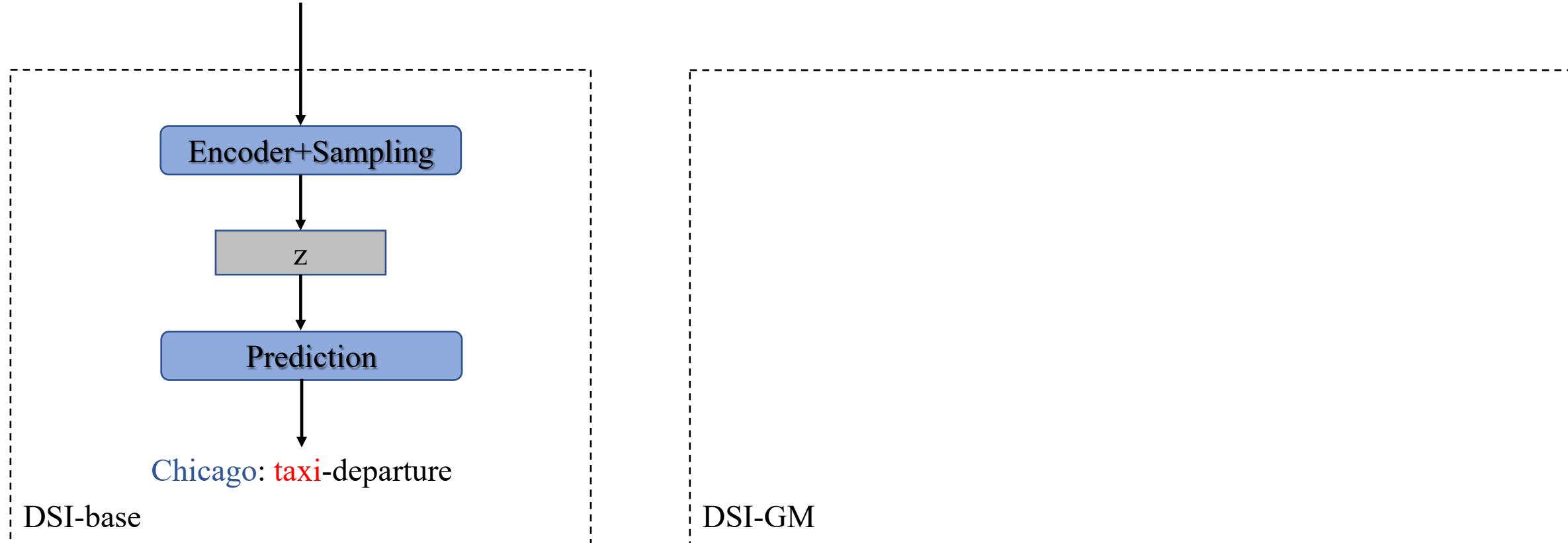
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



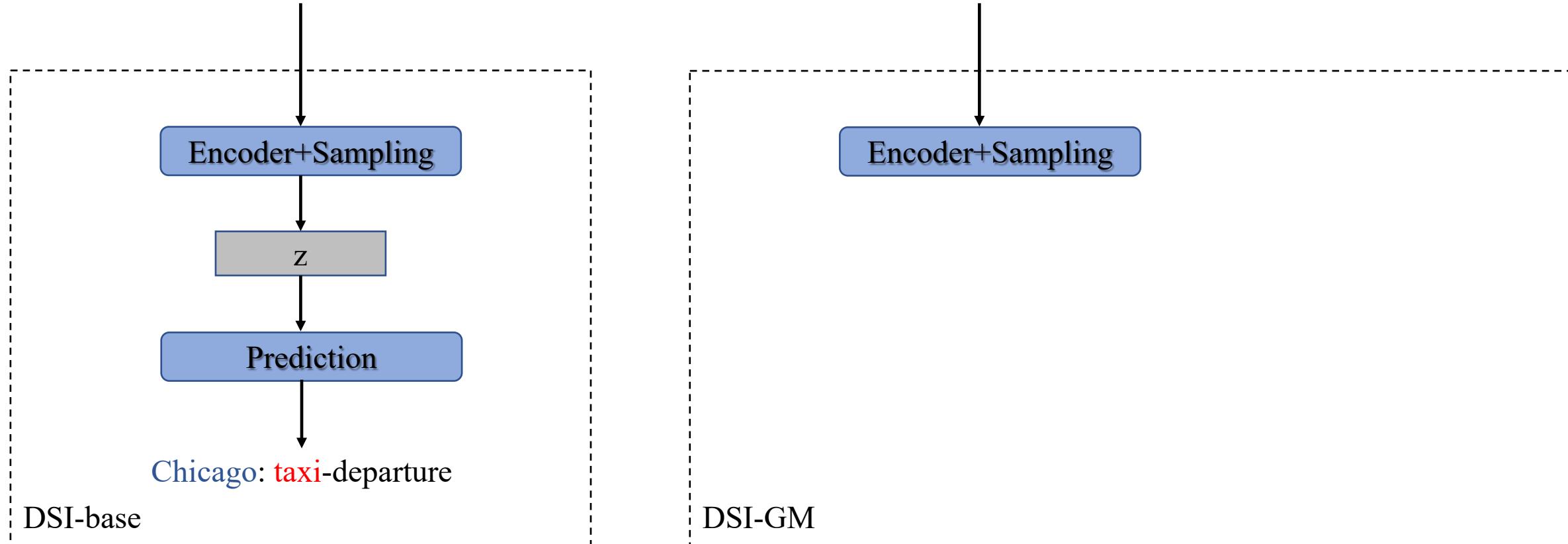
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



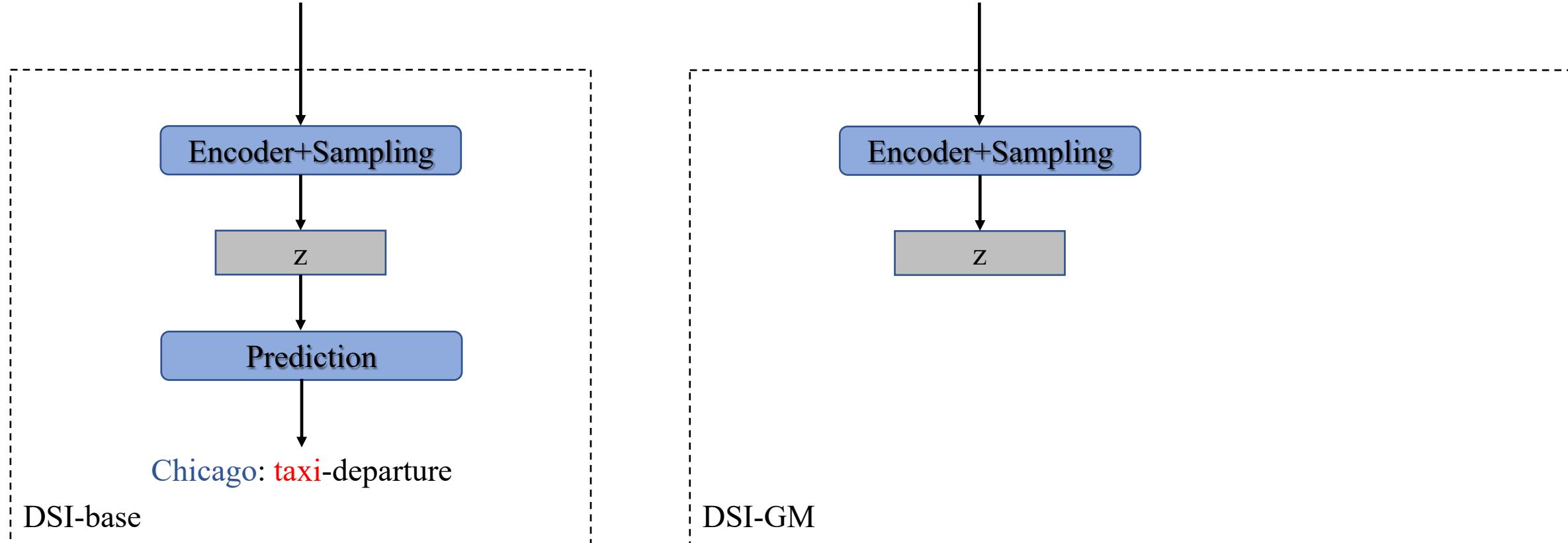
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



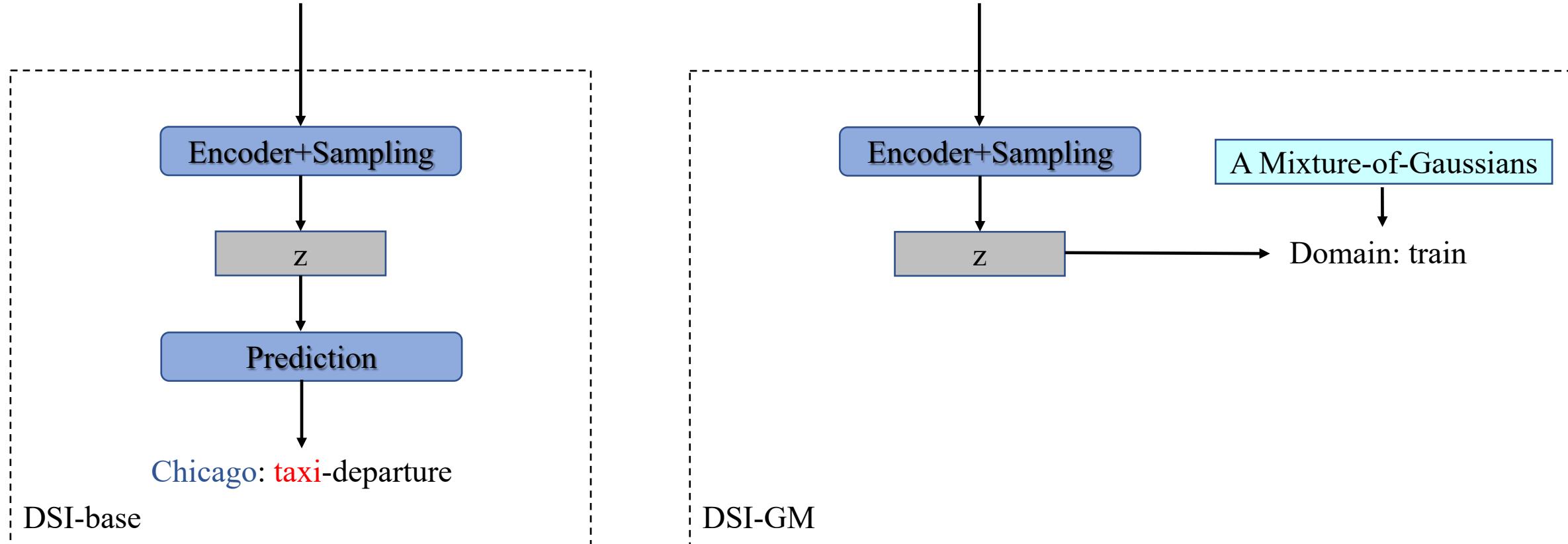
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



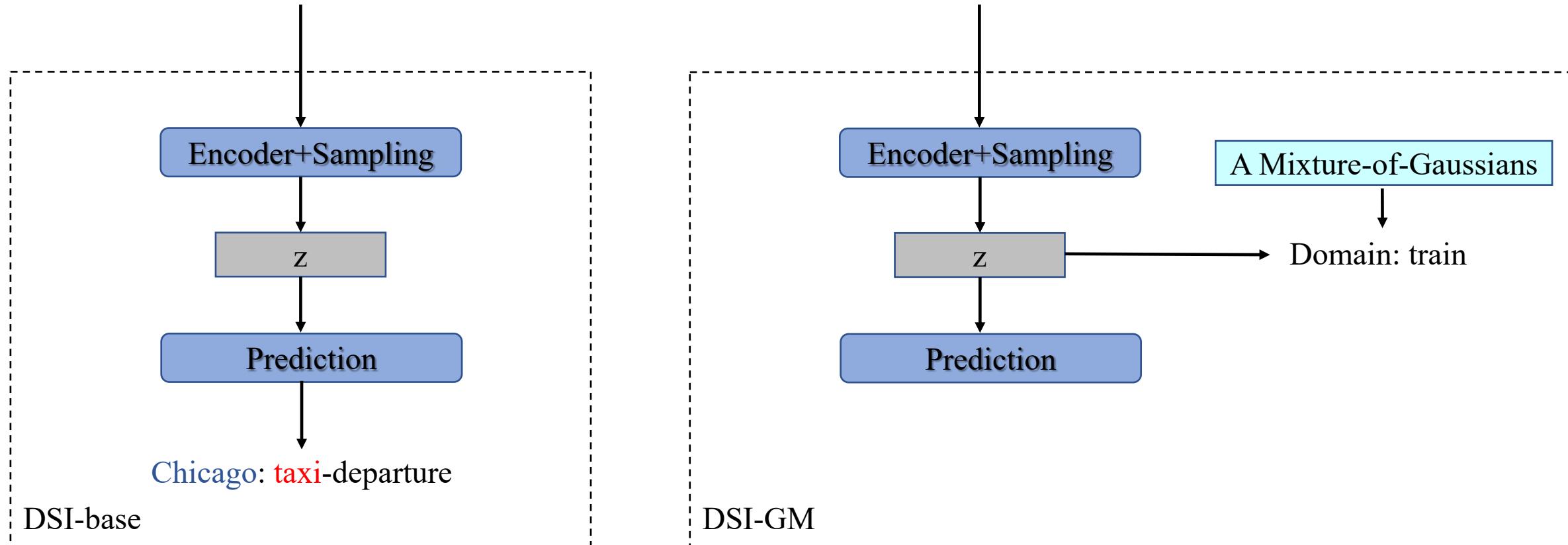
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



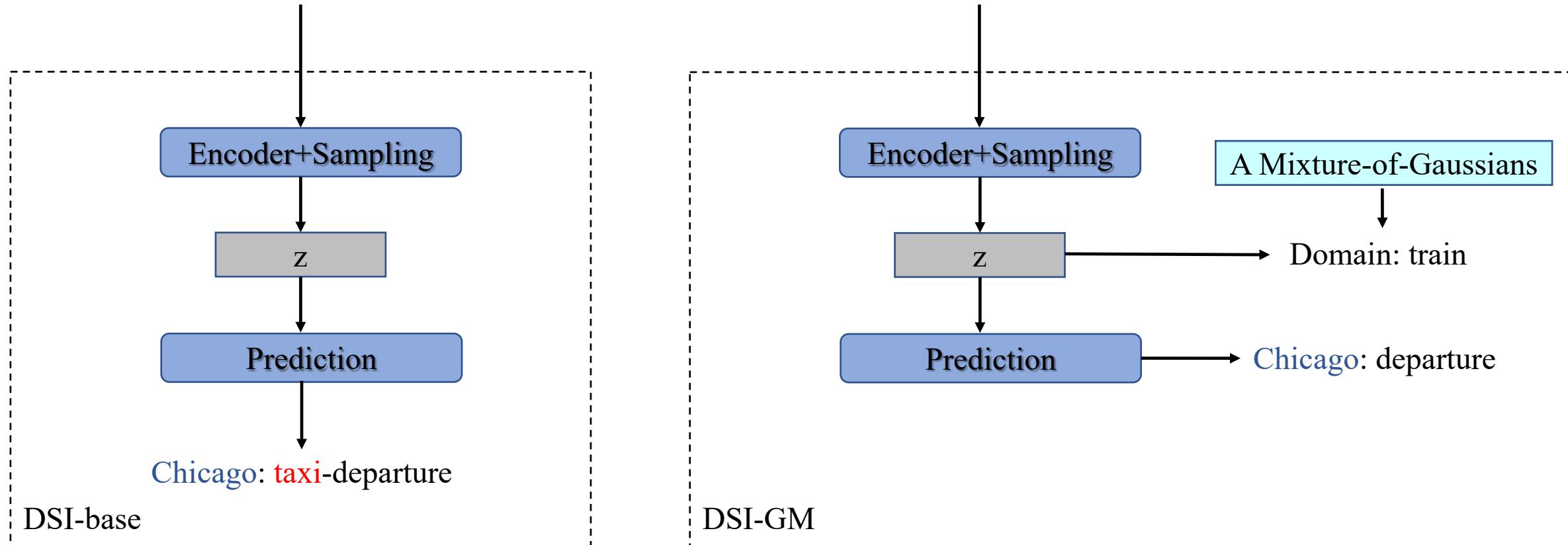
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



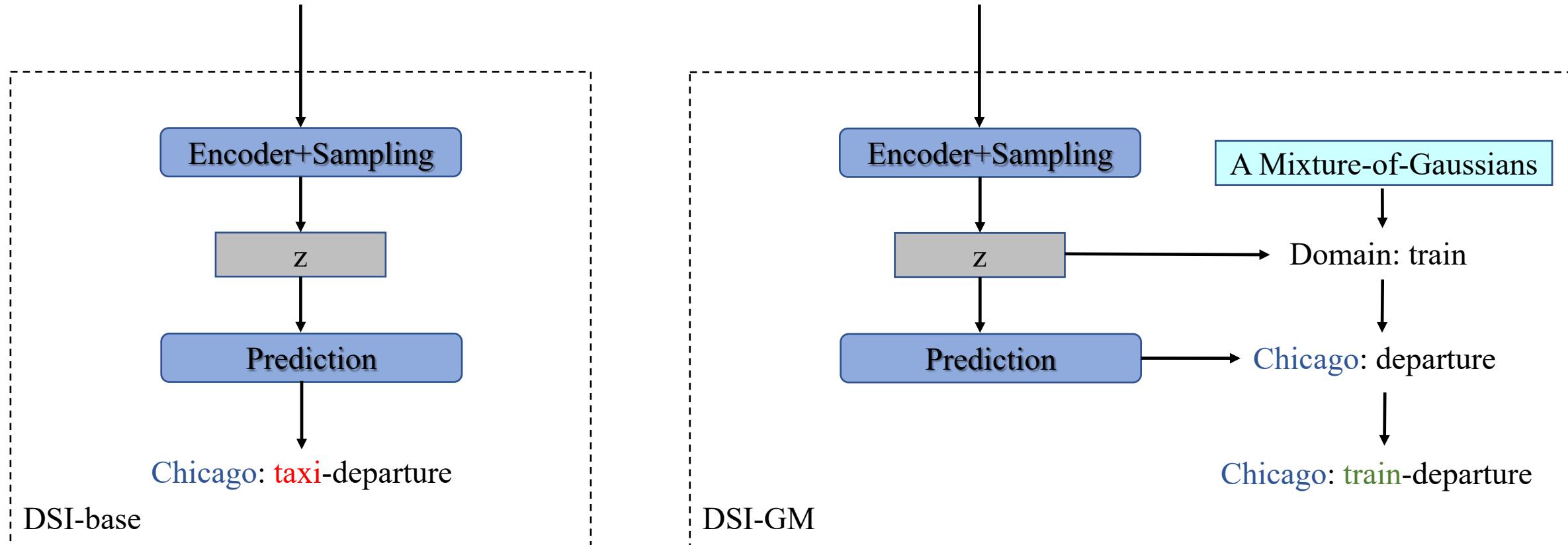
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



# CHAPTER 2 Model: generative *DSI-GM*

# CHAPTER 2 Model: generative *DSI-GM*

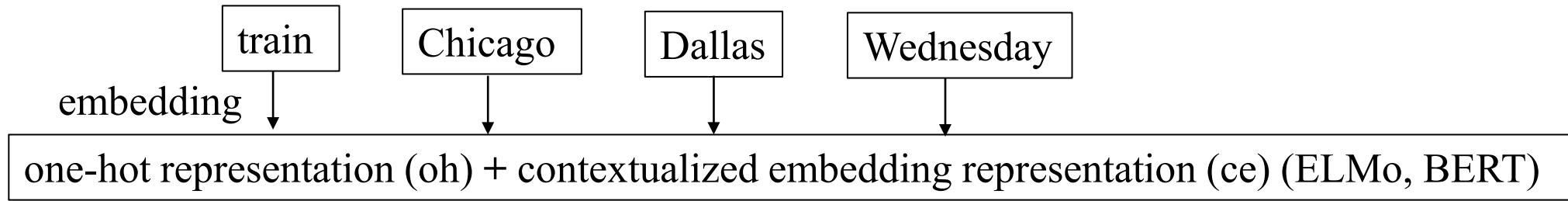
train

Chicago

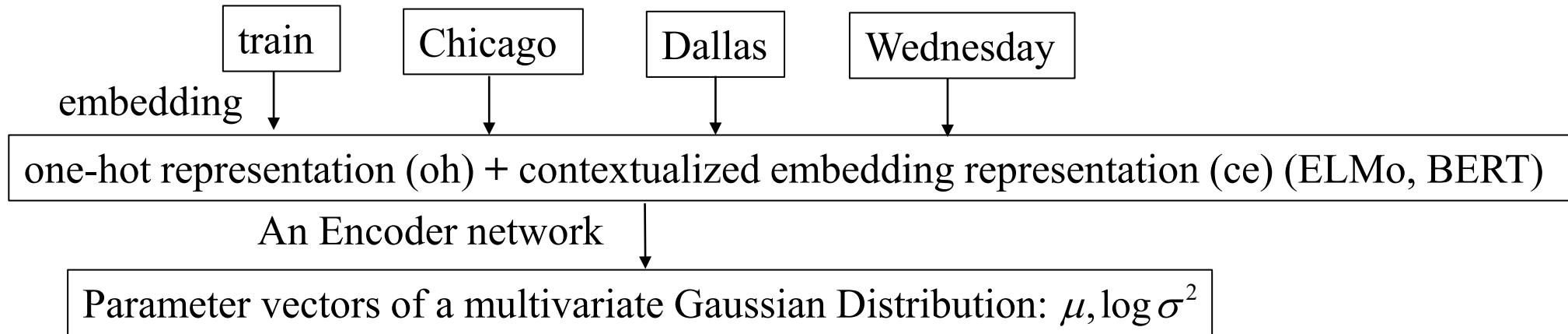
Dallas

Wednesday

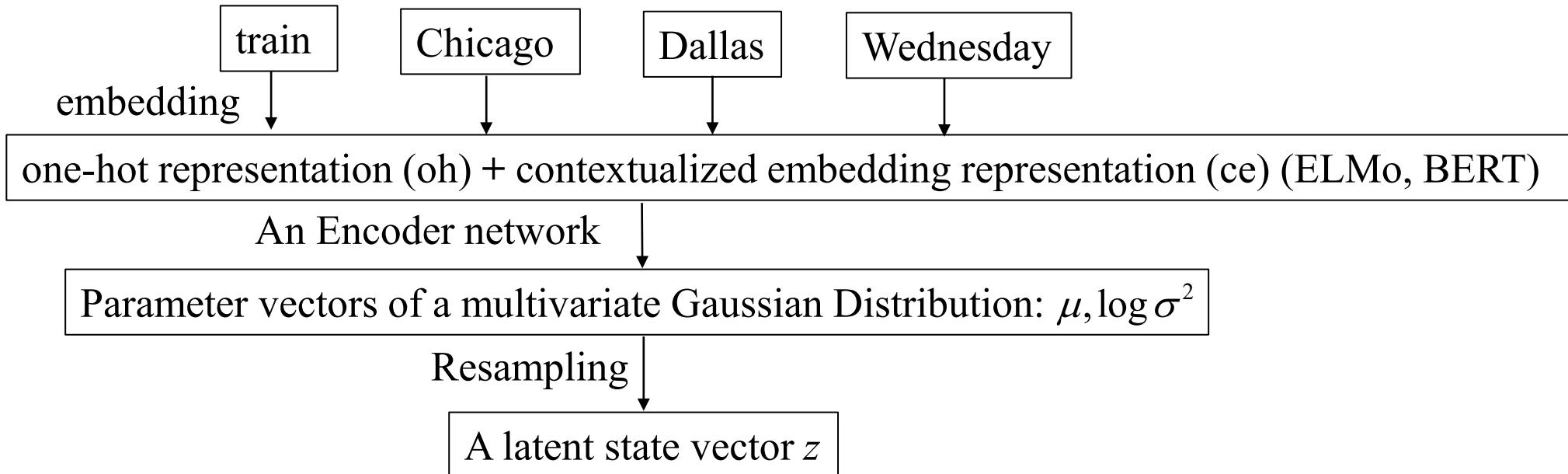
## CHAPTER 2 Model: generative *DSI-GM*



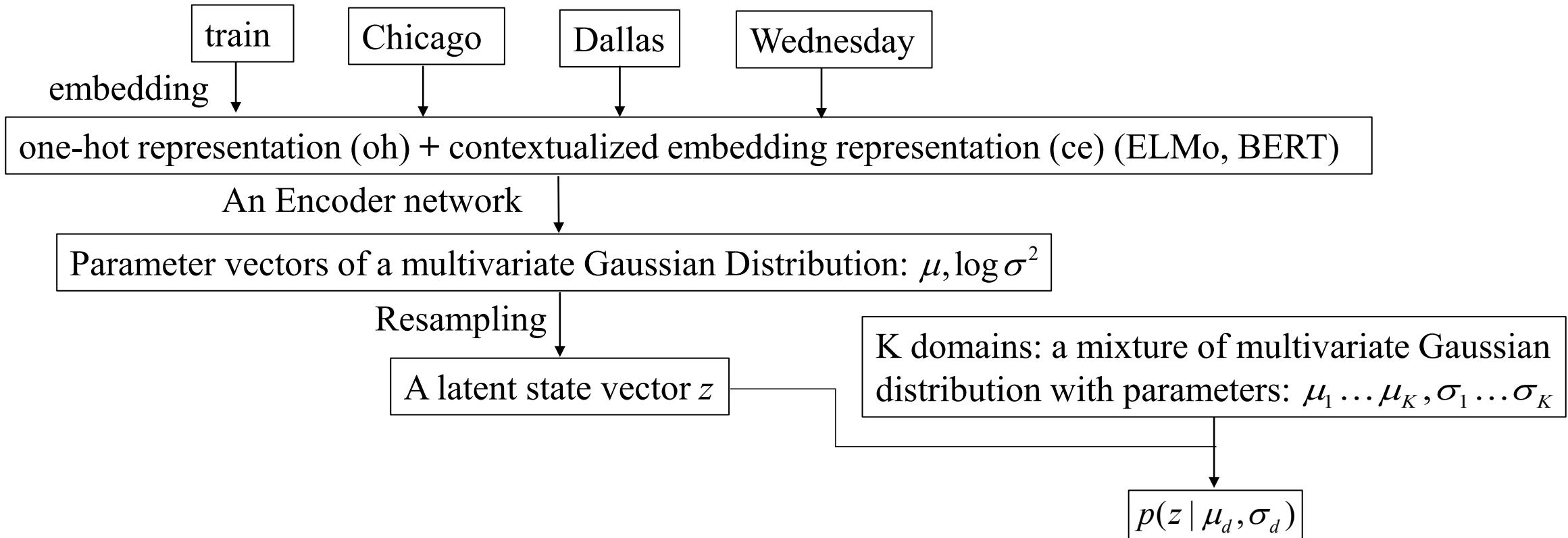
## CHAPTER 2 Model: generative *DSI-GM*



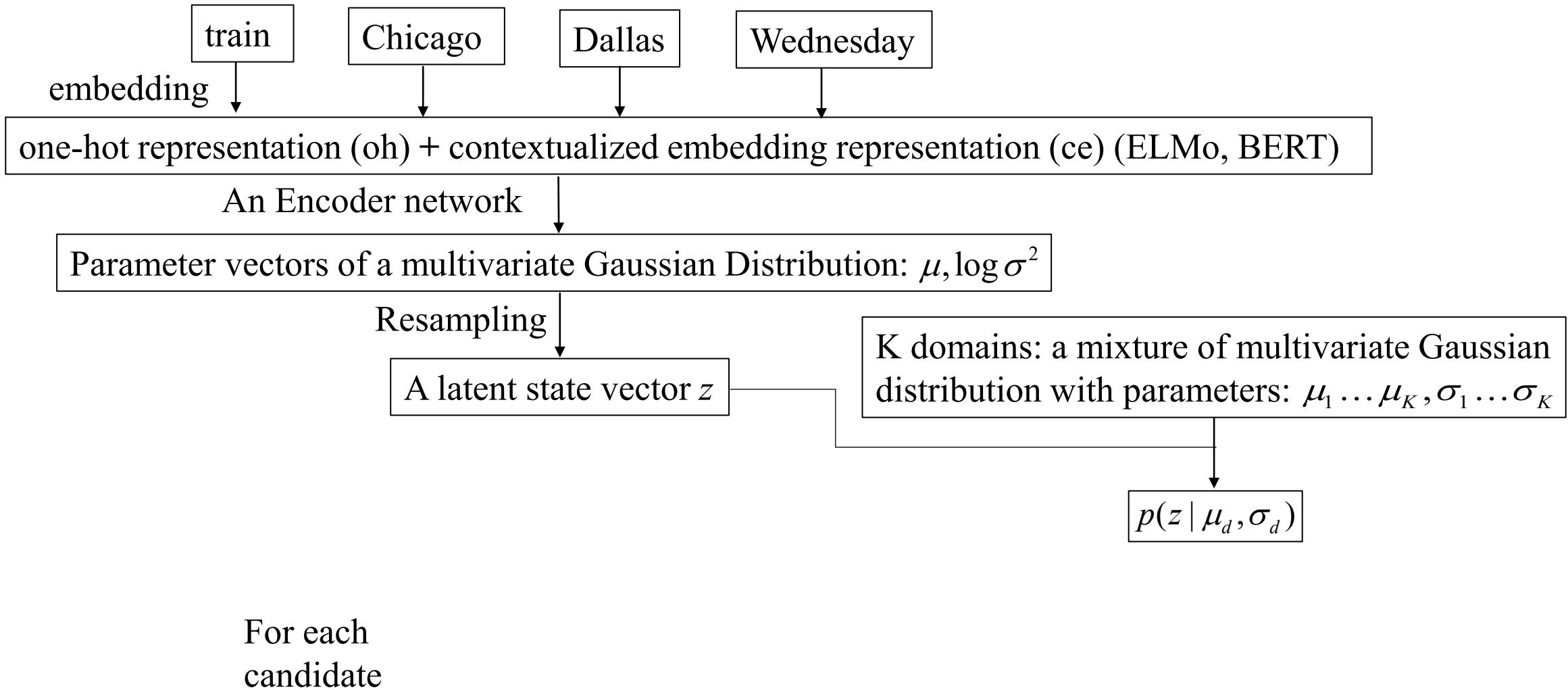
## CHAPTER 2 Model: generative DSI-GM



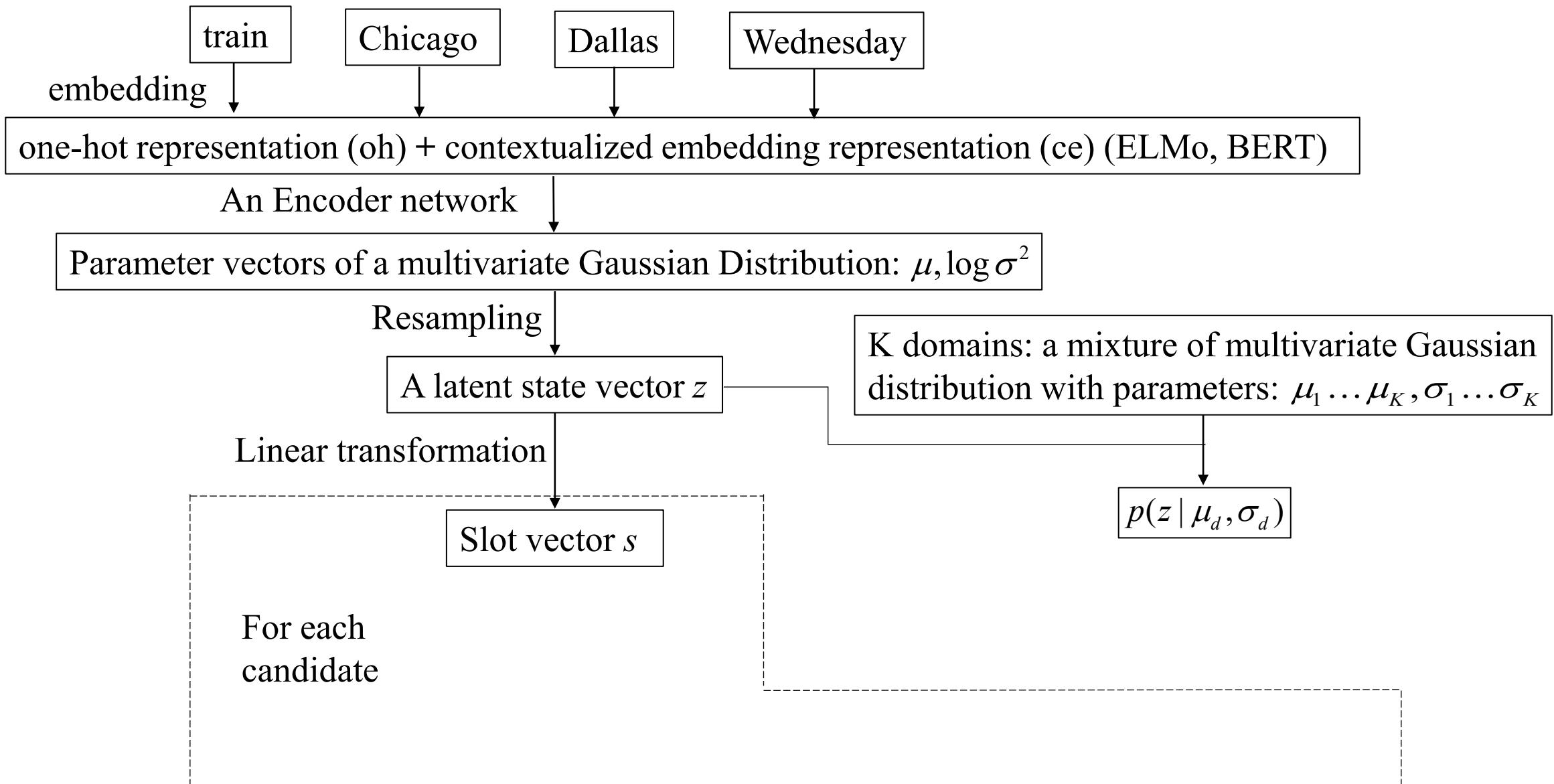
## CHAPTER 2 Model: generative DSI-GM



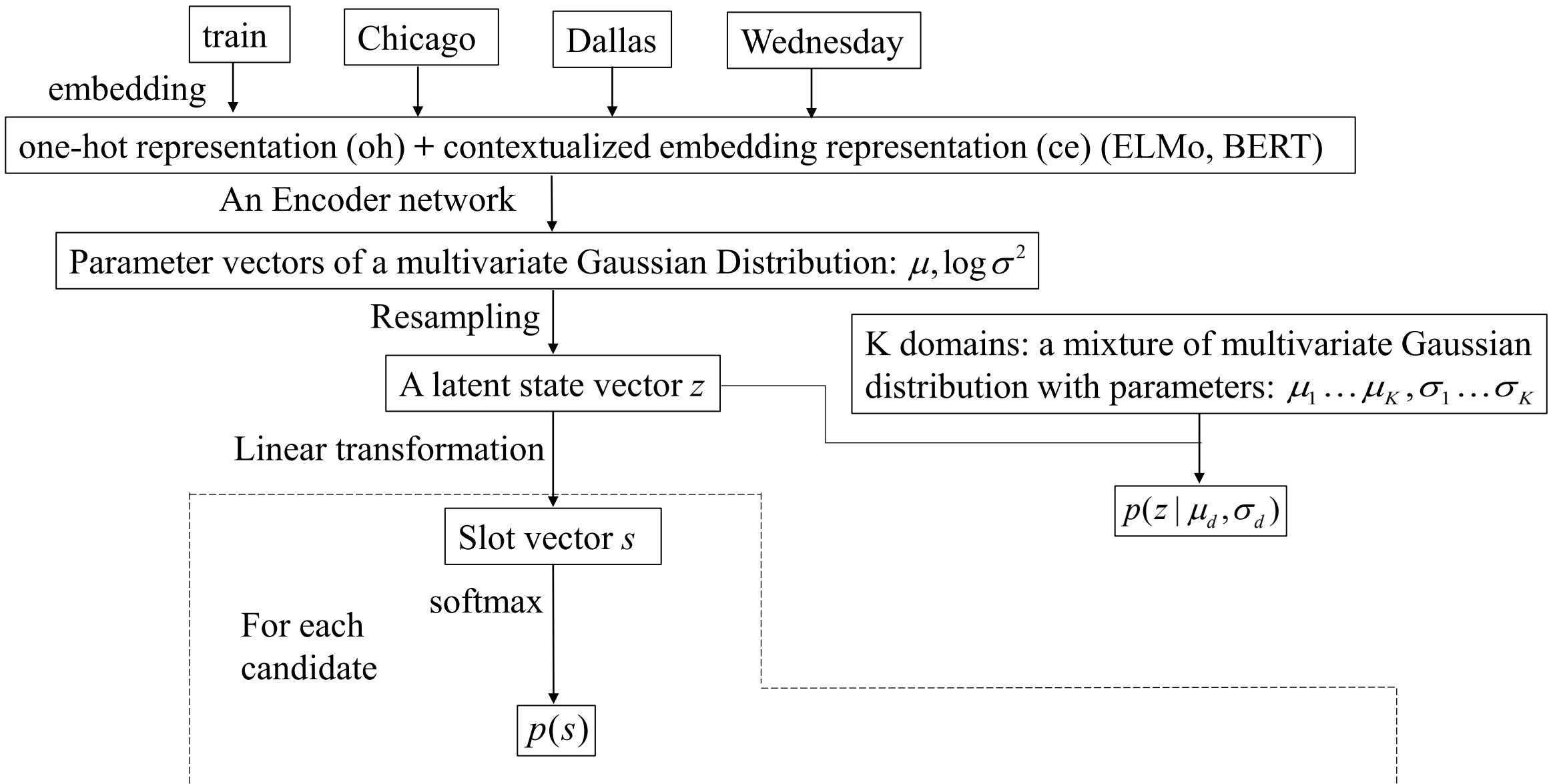
## CHAPTER 2 Model: generative DSI-GM



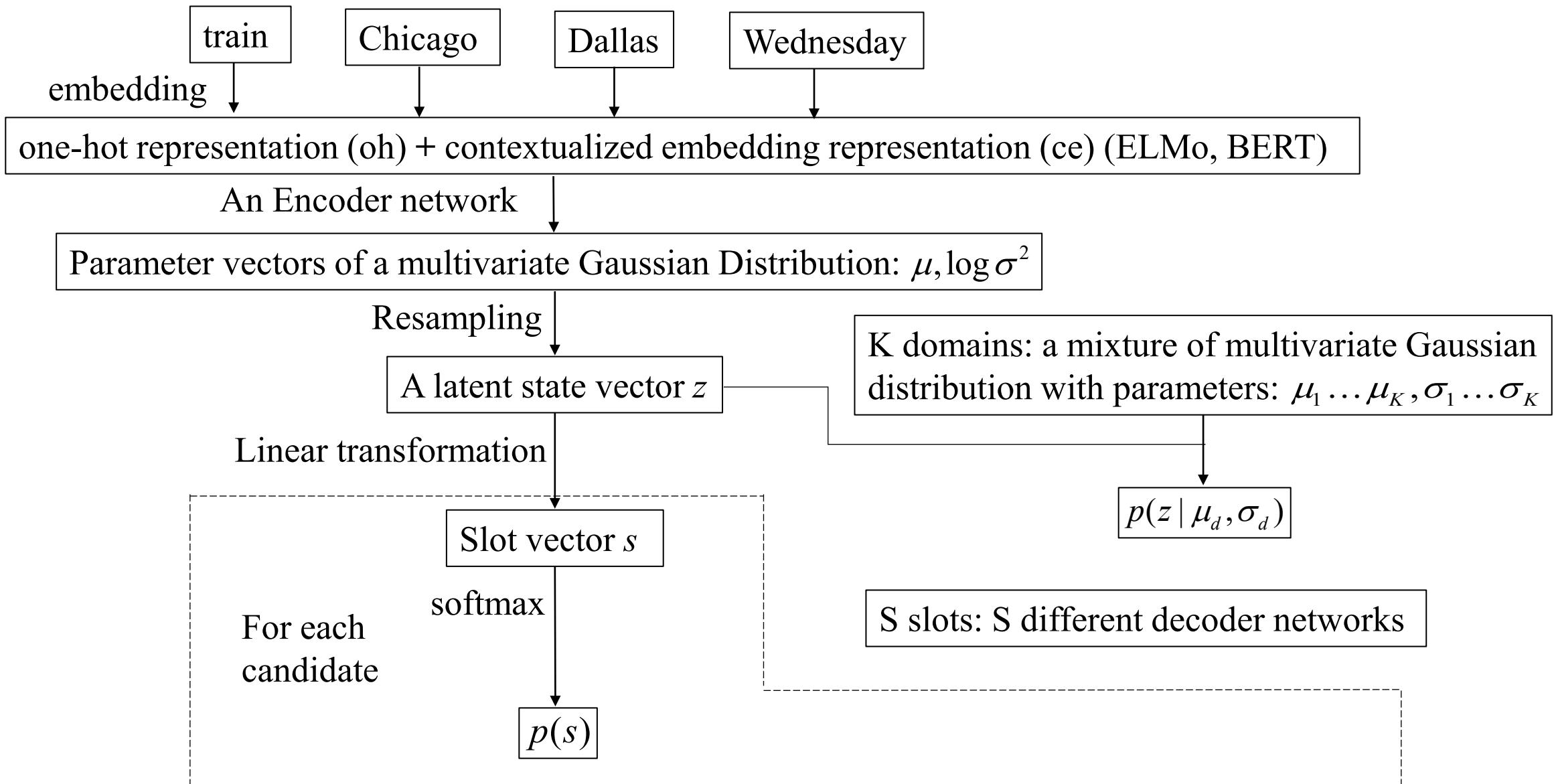
# CHAPTER 2 Model: generative DSI-GM



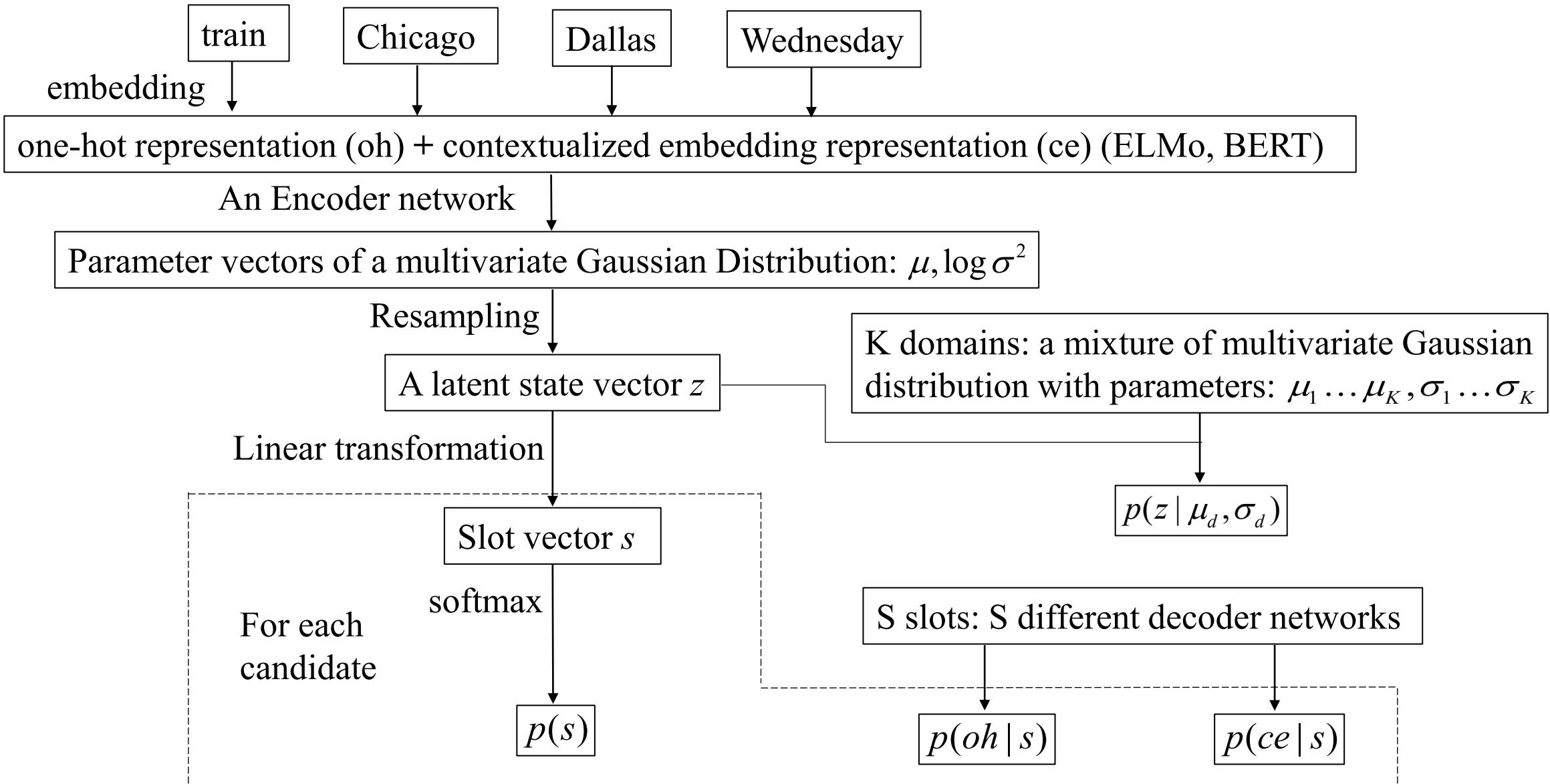
# CHAPTER 2 Model: generative DSI-GM



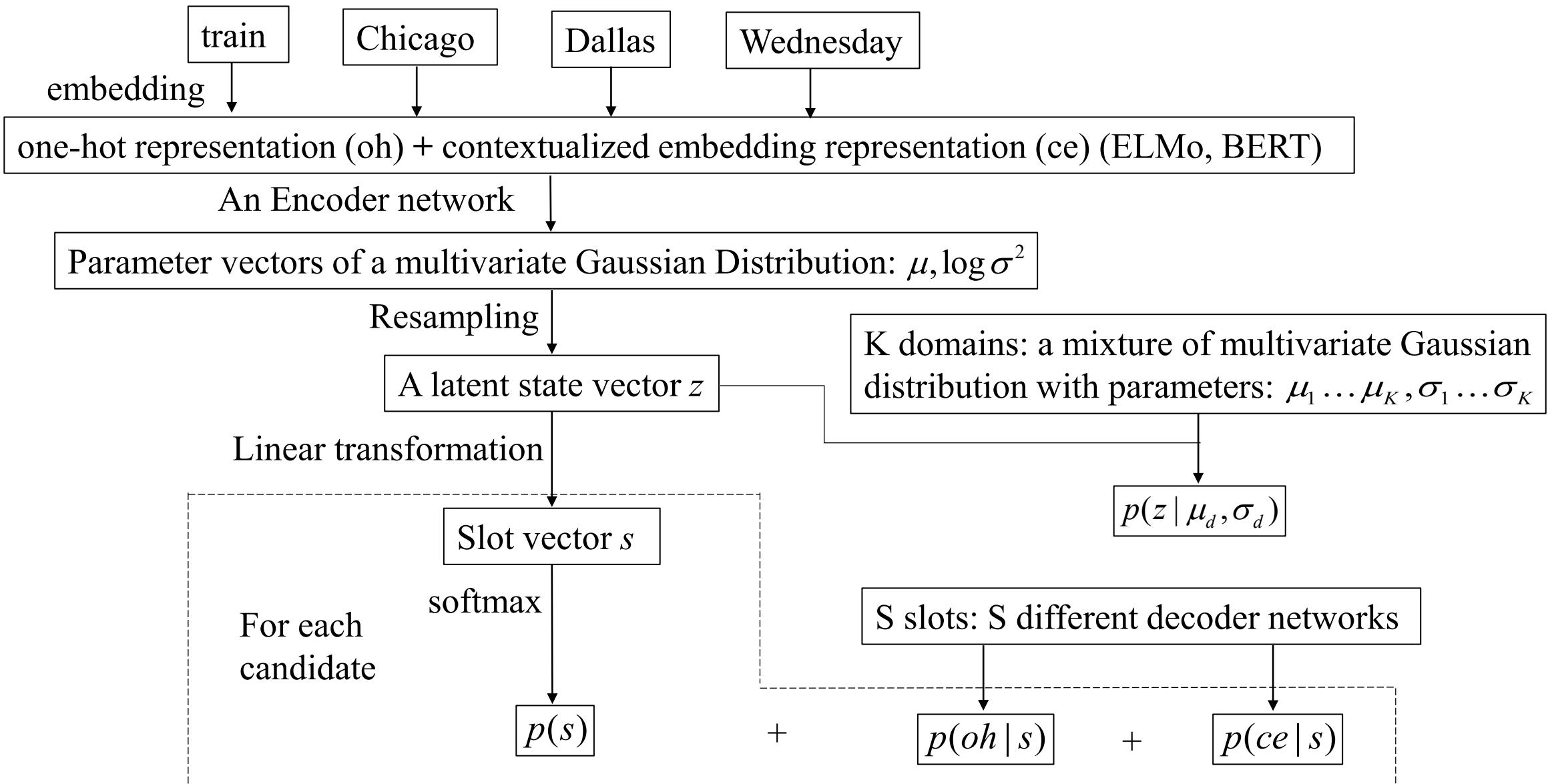
# CHAPTER 2 Model: generative DSI-GM



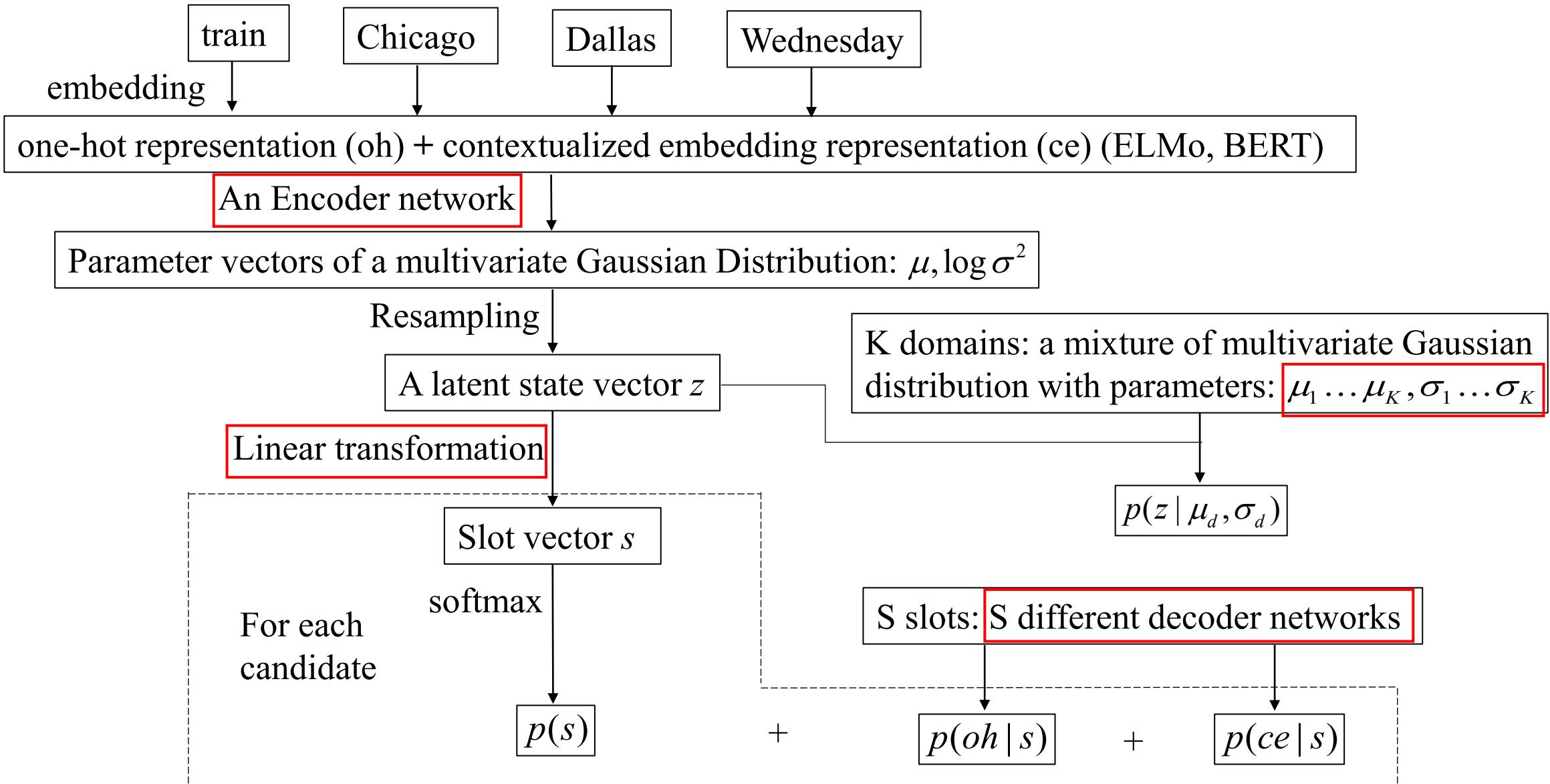
# CHAPTER 2 Model: generative DSI-GM



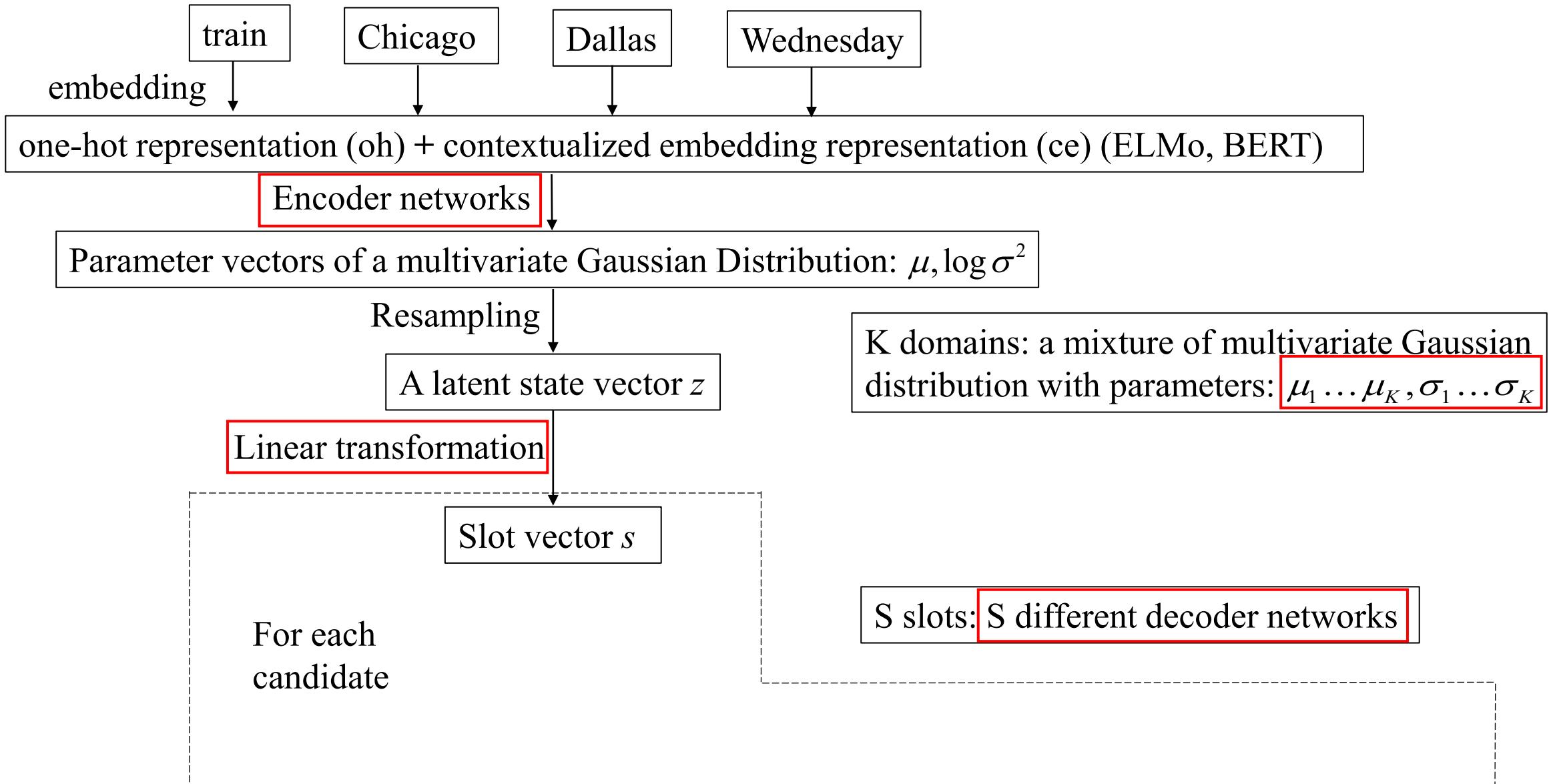
## CHAPTER 2 Model: generative DSI-GM



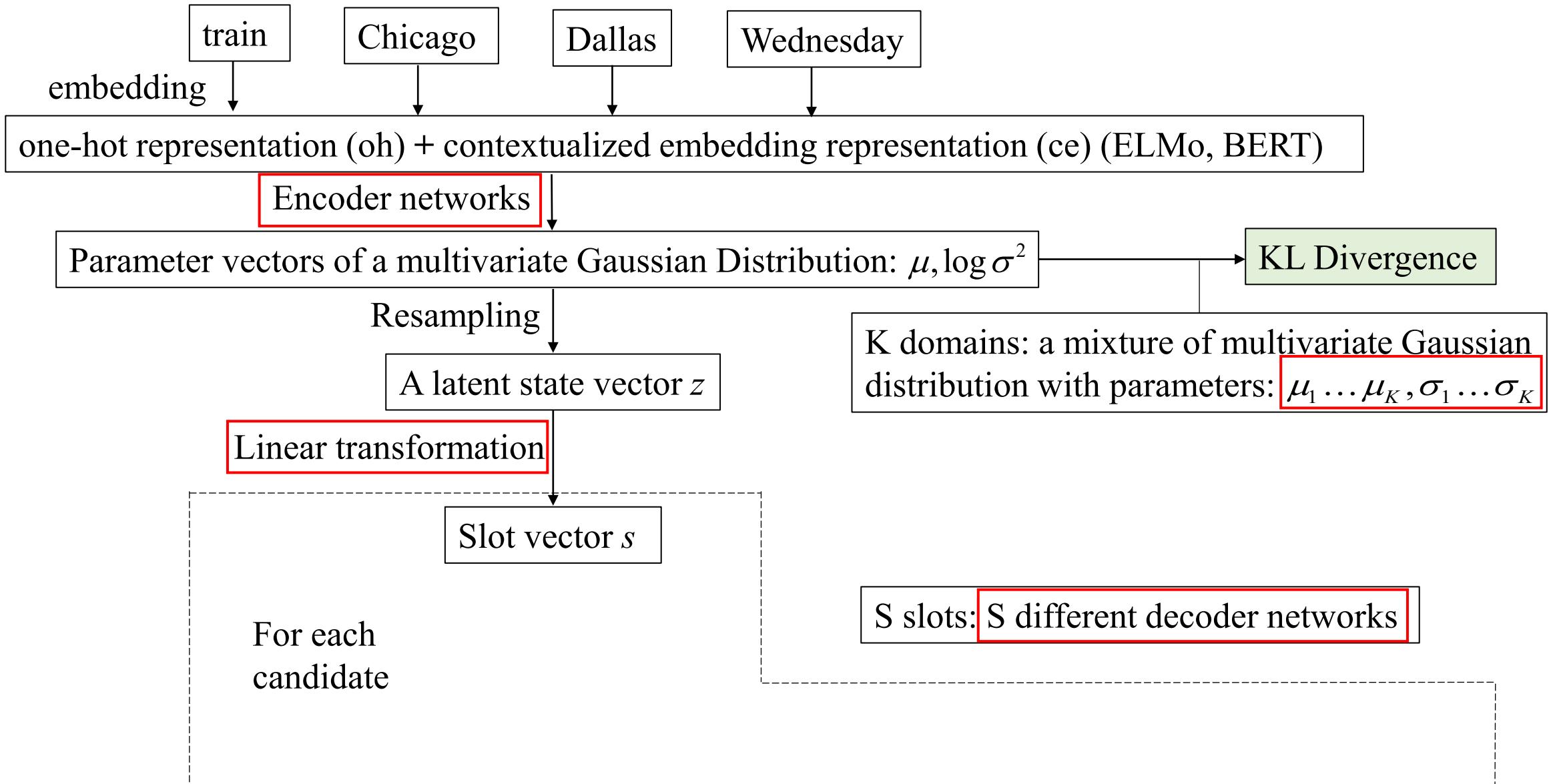
# CHAPTER 2 Model: generative DSI-GM



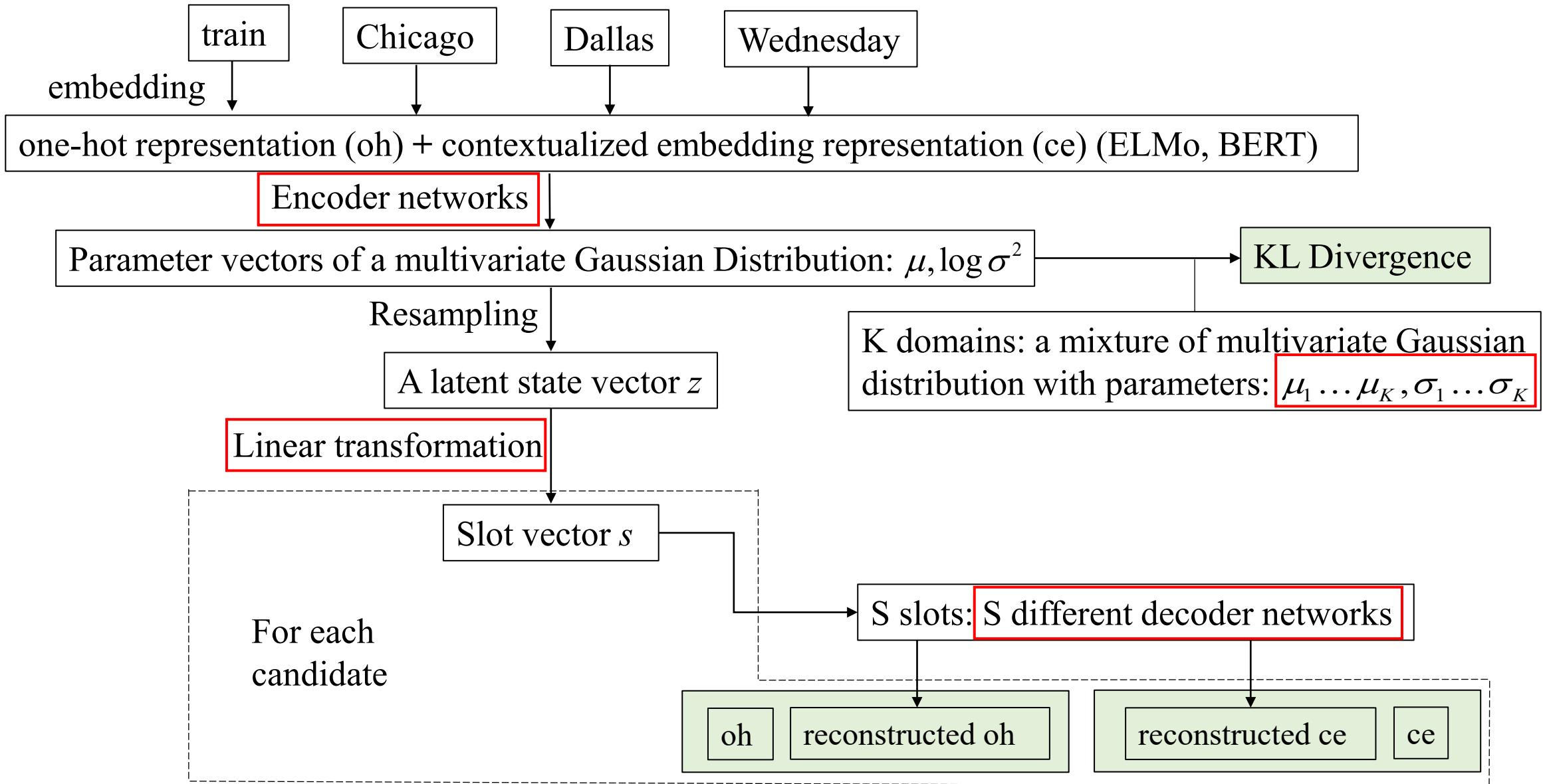
# CHAPTER 2 Model: generative DSI-GM



# CHAPTER 2 Model: generative DSI-GM



# CHAPTER 2 Model: generative DSI-GM





# CHAPTER 3

## Experiments

# CHAPTER 3 DSI results

| Models          | MultiWOZ 2.1 |        |      |          |             |        |      |          | SGD        |        |      |          |             |        |      |          |
|-----------------|--------------|--------|------|----------|-------------|--------|------|----------|------------|--------|------|----------|-------------|--------|------|----------|
|                 | Turn level   |        |      |          | Joint level |        |      |          | Turn level |        |      |          | Joint level |        |      |          |
|                 | Precision    | Recall | F1   | Accuracy | Precision   | Recall | F1   | Accuracy | Precision  | Recall | F1   | Accuracy | Precision   | Recall | F1   | Accuracy |
| <i>Random</i>   | 1.49         | 1.51   | 1.49 | 1.39     | 0.21        | 0.28   | 0.23 | 0.02     | 0.94       | 0.95   | 0.94 | 0.92     | 0.05        | 0.08   | 0.06 | 0.02     |
| <i>DSI-base</i> | 38.8         | 37.7   | 37.3 | 25.7     | 33.9        | 32.1   | 32.1 | 2.3      | 27.0       | 26.0   | 26.0 | 21.1     | 13.9        | 17.5   | 14.5 | 2.3      |
| <i>DSI-GM</i>   | 52.5         | 39.3   | 49.6 | 36.1     | 49.2        | 43.2   | 44.8 | 5.0      | 34.7       | 33.4   | 33.5 | 27.5     | 19.0        | 22.9   | 19.5 | 3.1      |

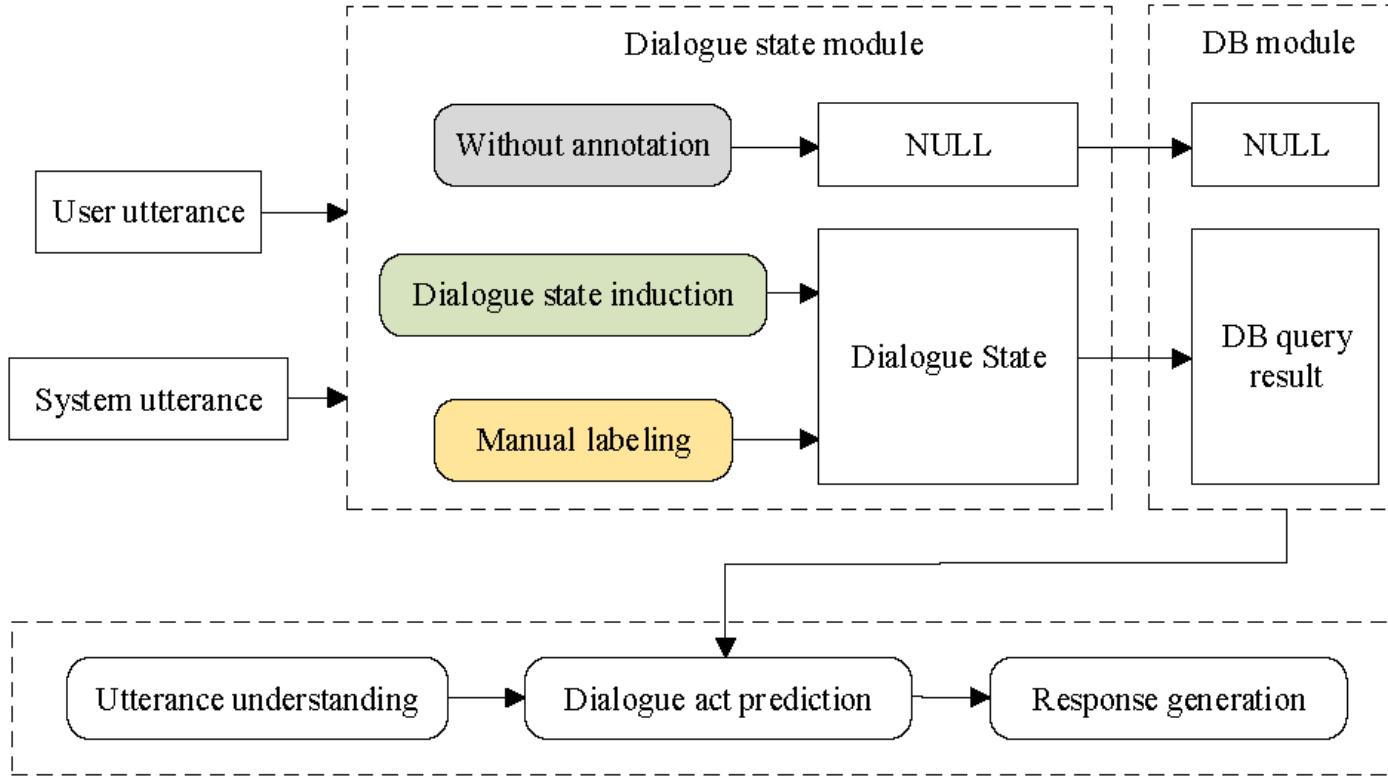
Table 1: Overall results of DSI.

# CHAPTER 3 DSI results

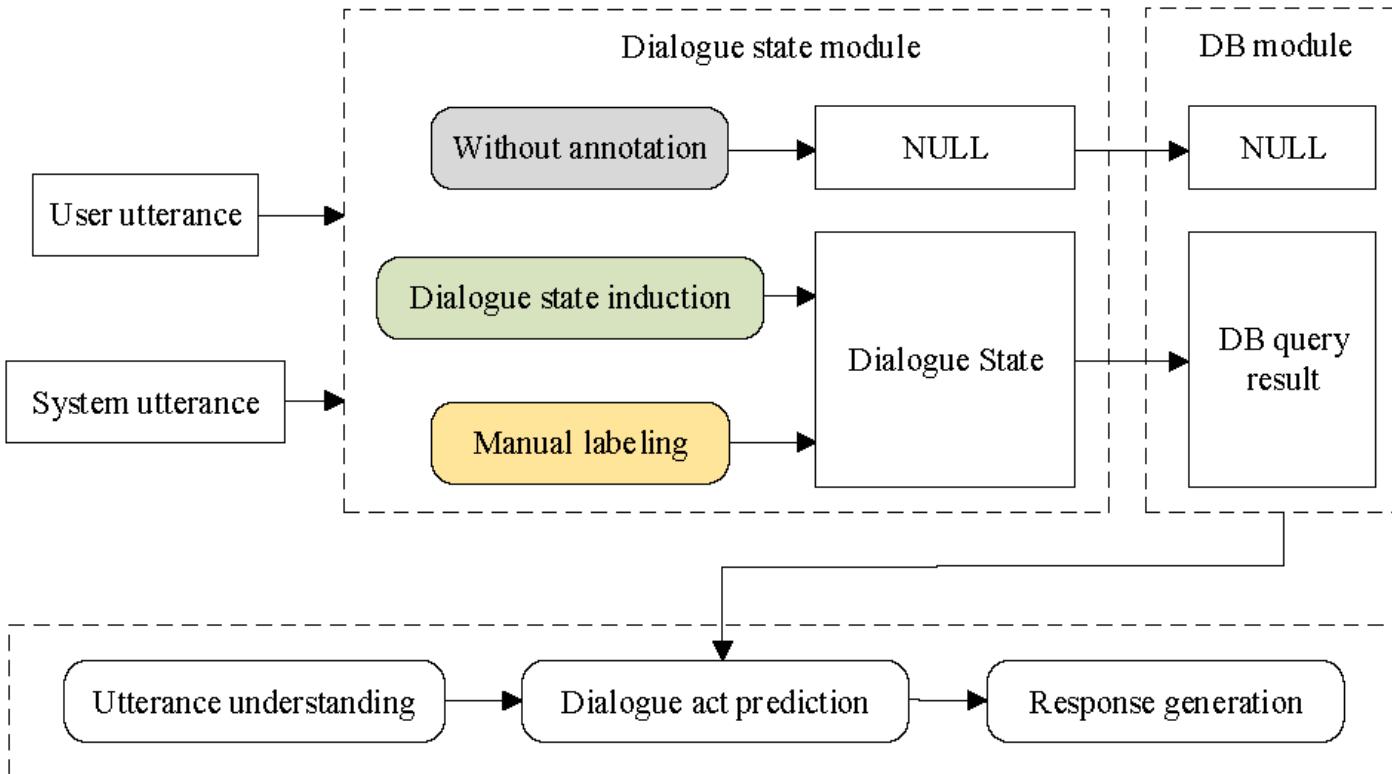
| Models          | MultiWOZ 2.1 |        |      |          |             |        |      |          | SGD        |        |      |          |             |        |      |          |
|-----------------|--------------|--------|------|----------|-------------|--------|------|----------|------------|--------|------|----------|-------------|--------|------|----------|
|                 | Turn level   |        |      |          | Joint level |        |      |          | Turn level |        |      |          | Joint level |        |      |          |
|                 | Precision    | Recall | F1   | Accuracy | Precision   | Recall | F1   | Accuracy | Precision  | Recall | F1   | Accuracy | Precision   | Recall | F1   | Accuracy |
| <i>Random</i>   | 1.49         | 1.51   | 1.49 | 1.39     | 0.21        | 0.28   | 0.23 | 0.02     | 0.94       | 0.95   | 0.94 | 0.92     | 0.05        | 0.08   | 0.06 | 0.02     |
| <i>DSI-base</i> | 38.8         | 37.7   | 37.3 | 25.7     | 33.9        | 32.1   | 32.1 | 2.3      | 27.0       | 26.0   | 26.0 | 21.1     | 13.9        | 17.5   | 14.5 | 2.3      |
| <i>DSI-GM</i>   | 52.5         | 39.3   | 49.6 | 36.1     | 49.2        | 43.2   | 44.8 | 5.0      | 34.7       | 33.4   | 33.5 | 27.5     | 19.0        | 22.9   | 19.5 | 3.1      |

Table 1: Overall results of DSI.

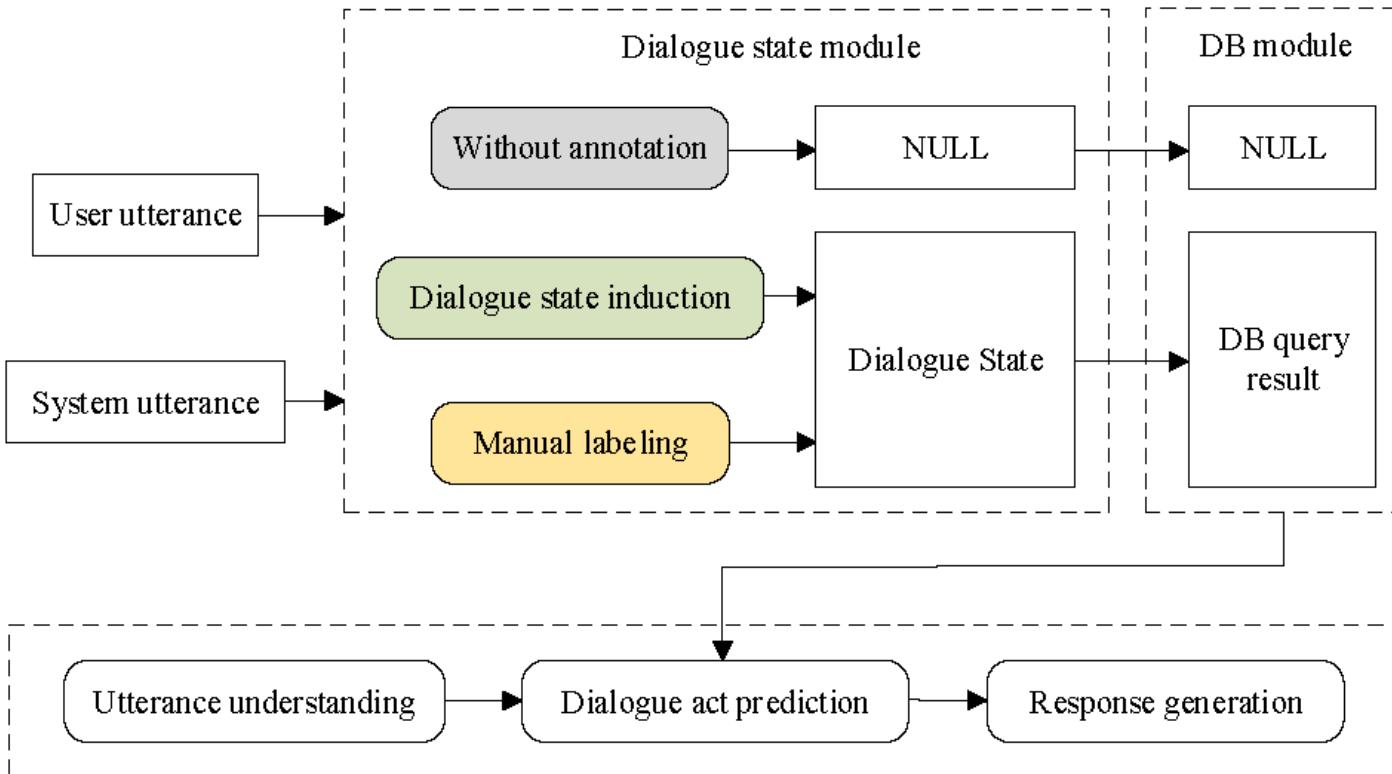
# CHAPTER 3 DSI-Based Response Generation



[Chen et al., 2019] Wenhui Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. In ACL, 2019.



| Dialogue State         | Dialog Act Prediction |        |      | Delexicalized |           |
|------------------------|-----------------------|--------|------|---------------|-----------|
|                        | Precision             | Recall | F1   | BLEU          | Entity F1 |
| <i>None</i>            | 71.0                  | 67.4   | 69.1 | 18.7          | 54.6      |
| <i>DSI-GM</i>          | 72.0                  | 70.5   | 71.2 | 20.8          | 56.5      |
| <i>Manual labeling</i> | 75.6                  | 73.0   | 74.2 | 21.6          | 61.3      |

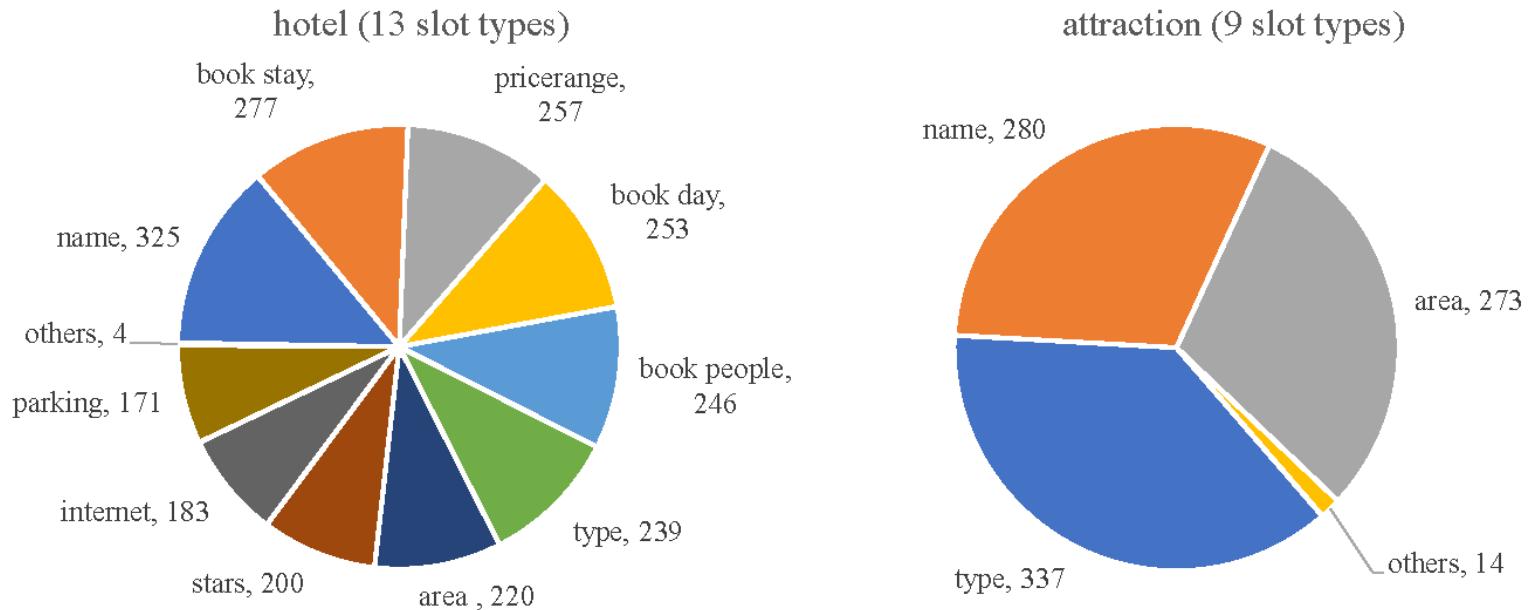


| Dialogue State  | Dialog Act Prediction |        |      | Delexicalized |           |
|-----------------|-----------------------|--------|------|---------------|-----------|
|                 | Precision             | Recall | F1   | BLEU          | Entity F1 |
| None            | 71.0                  | 67.4   | 69.1 | 18.7          | 54.6      |
| DSI-GM          | 72.0                  | 70.5   | 71.2 | 20.8          | 56.5      |
| Manual labeling | 75.6                  | 73.0   | 74.2 | 21.6          | 61.3      |

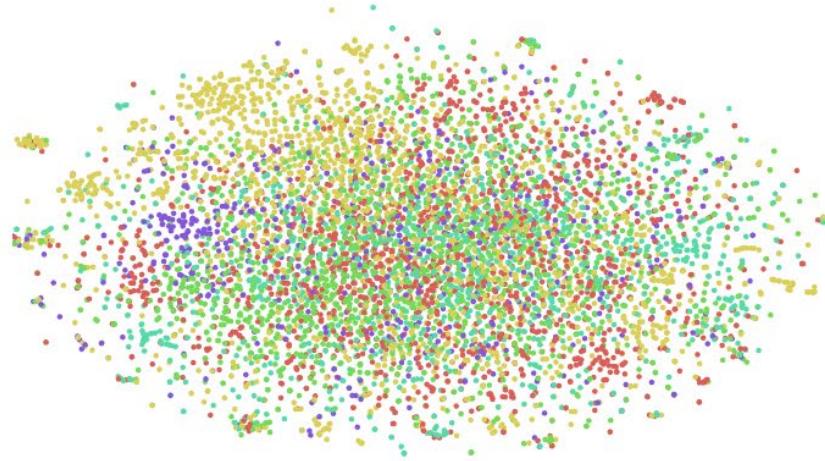
# CHAPTER 3 Analysis

|          | attraction | hotel | restaurant | taxi | train |
|----------|------------|-------|------------|------|-------|
| DSI-base | 27.9       | 21.7  | 26.1       | 30.7 | 26.0  |
| DSI-GM   | 40.3       | 31.4  | 35.6       | 39.9 | 36.8  |

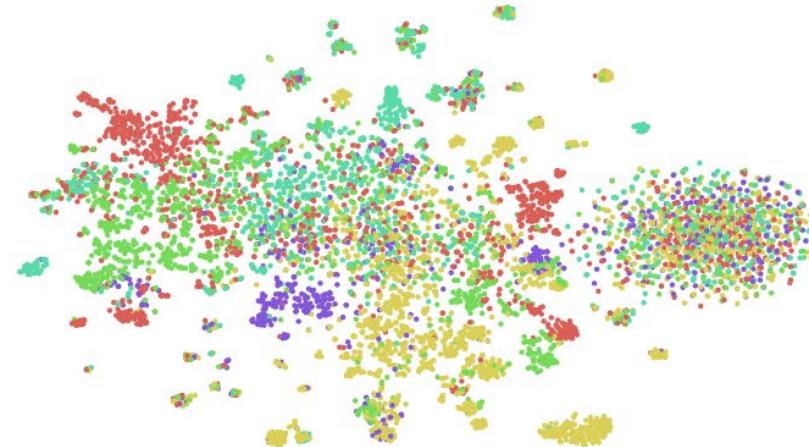
Table 4: Turn goal accuracy per domain.



# CHAPTER 3 Analysis



(a) *DSI-base*



(b) *DSI-GM*

Domain level comparison of the latent representation z.



# CHAPTER 4

## Conclusion



- **Dialogue state induction**: a novel task to automatically identify dialogue states
- *DSI-base/DSI-GM*: two neural generative models with **latent variables**
- Challenging and promising: **unsupervised** setting is very **practical**
- IJCAI review: this problem is important and interesting, this area should attract more attention. This work has great potential of **motivating follow-up research**.

- **Dialogue state induction**: a novel task to automatically identify dialogue states
- *DSI-base/DSI-GM*: two neural generative models with **latent variables**
- Challenging and promising: **unsupervised** setting is very **practical**
- IJCAI review: this problem is important and interesting, this area should attract more attention. This work has great potential of **motivating follow-up research**.

- **Dialogue state induction**: a novel task to automatically identify dialogue states
- *DSI-base/DSI-GM*: two neural generative models with **latent variables**
- Challenging and promising: **unsupervised** setting is very **practical**
- IJCAI review: this problem is important and interesting, this area should attract more attention. This work has great potential of **motivating follow-up research**.

- **Dialogue state induction**: a novel task to automatically identify dialogue states
- *DSI-base/DSI-GM*: two neural generative models with **latent variables**
- Challenging and promising: **unsupervised** setting is very **practical**
- IJCAI review: this problem is important and interesting, this area should attract more attention. This work has great potential of **motivating follow-up research**.

# THANK YOU

Contact:  
[minqingkai@westlake.edu.cn](mailto:minqingkai@westlake.edu.cn)

Paper:  
<https://www.ijcai.org/Proceedings/2020/0532.pdf>

GitHub:  
<https://github.com/taolusi/dialogue-state-induction>

paper



GitHub

