Dialogue State Induction Using Neural Latent Variable Models

Qingkai Min[1;2] , Libo Qin[3] , Zhiyang Teng[1;2] , Xiao Liu[4] and Yue Zhang[1;2]

[1]School of Engineering, Westlake University

[2]Institute of Advanced Technology, Westlake Institute for Advanced Study

[3]Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology

[4]School of Computer Science and Technology, Beijing Institute of Technology

西湖大學
WESTLAKE UNIVERSITY

CONTENTS
目录

CHAPTER 1
Motivation

CHAPTER 2
Method

CHAPTER 3
Experiments

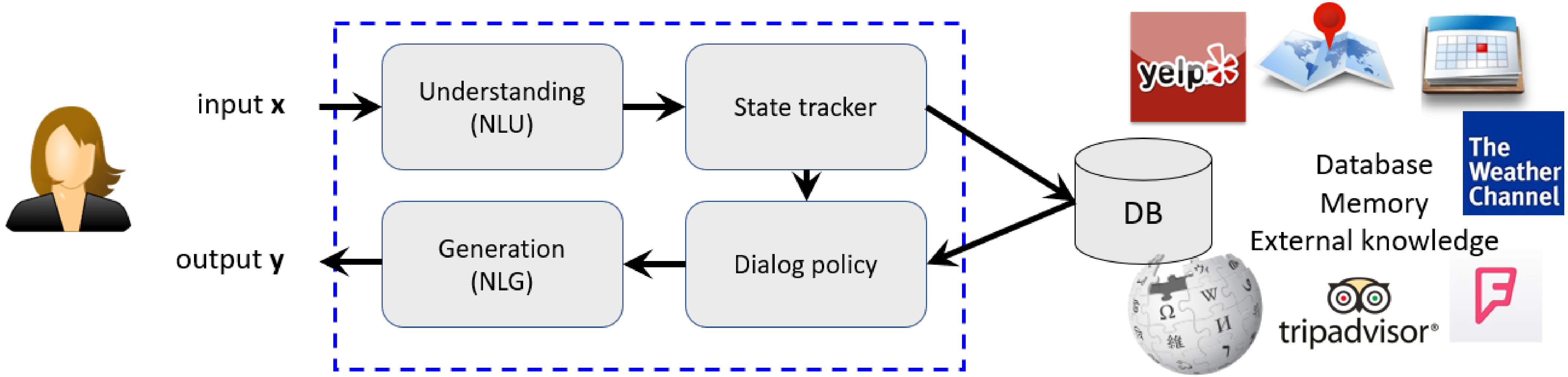
CHAPTER 4
Conclusion

CHAPTER 1

Motivation

What is a task-oriented dialogue system?

Assist user in solving a task



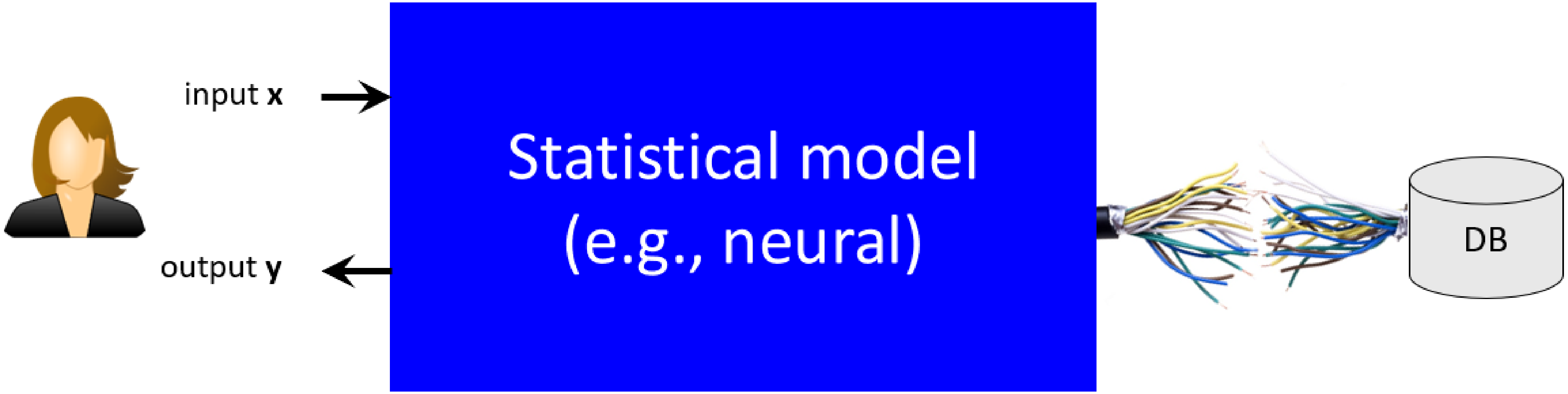
Typical modular architecture

Pic from: Gao J, Galley M, Li L. Neural approaches to conversational AI[C]//The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. 2018: 1371-1374.

What is a task-oriented dialogue system?

西湖大學 WESTLAKE UNIVERSITY

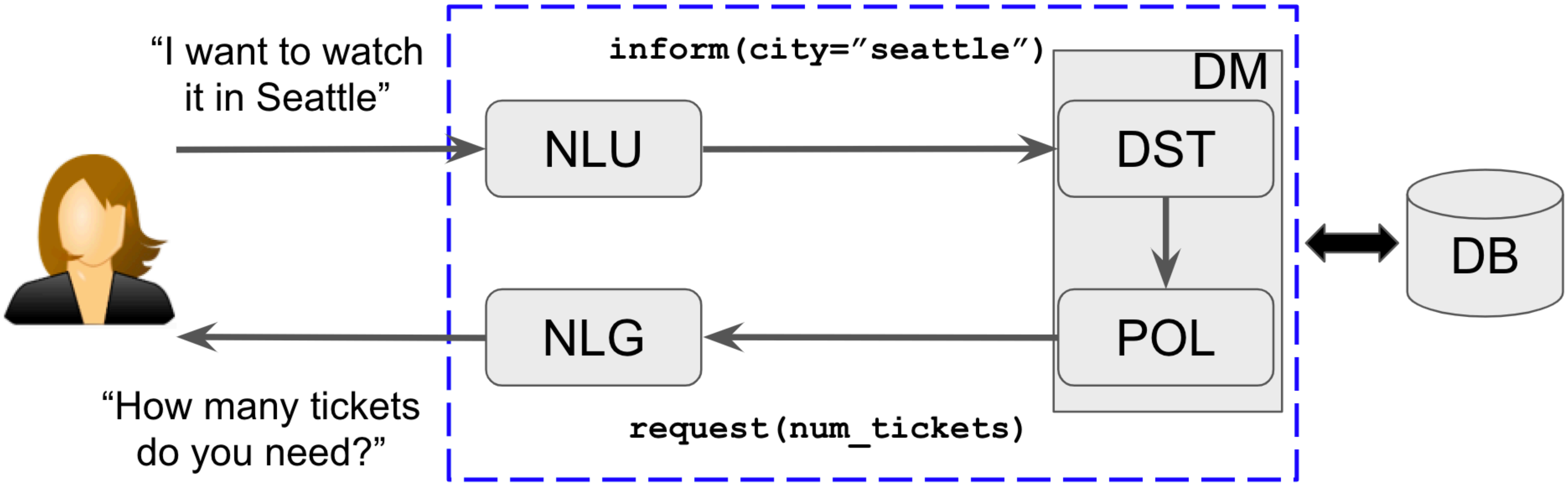
Assist user in solving a task



End-to-end architecture

Pic from: Gao J, Galley M, Li L. Neural approaches to conversational AI[C]//The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. 2018: 1371-1374.

What is a task-oriented dialogue system?



Typical modular architecture

Pic from: Gao J, Galley M, Li L. Neural approaches to conversational AI[C]//The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. 2018: 1371-1374.

What is dialogue state tracking (DST)?

The dialogue state represents what the user is looking for at the current turn of the conversation.

What is dialogue state tracking (DST)?

The dialogue state represents what the user is looking for at the current turn of the conversation.

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

Turn 1:
`inform(price=expensive, food=Turkish)`

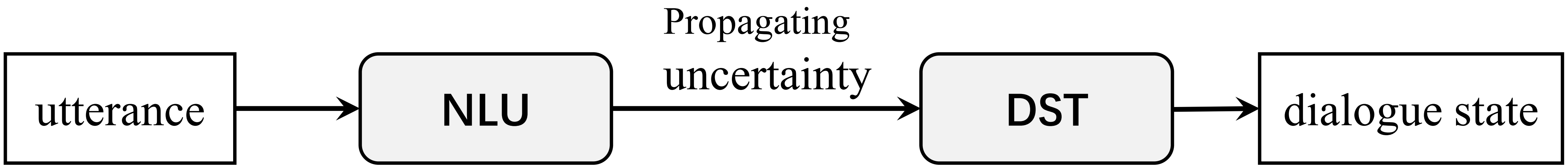
System: Anatolia serves Turkish food.
User: What is the area?

Turn 2:
`inform(price=expensive, food=Turkish); request(area)`

Current DST scenarios

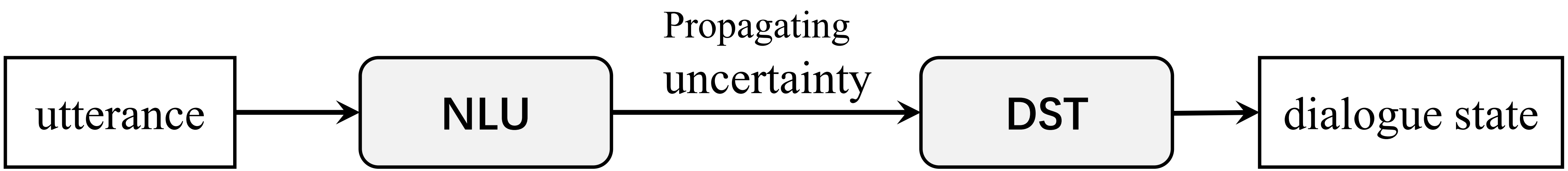
Current DST scenarios

Traditional DST:



Current DST scenarios

Traditional DST:



End-to-end DST:

End-to-end DST

End-to-end DST

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

System: Anatolia serves Turkish food.
User: What is the area?

End-to-end DST

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, ...]

End-to-end DST

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

End-to-end DST

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)`

Limitation of end-to-end DST

Limitation of end-to-end DST

End-to-end DST paradigm:

manual labeling → corpus → **train** → DST model → **test**

Limitation of end-to-end DST

End-to-end DST paradigm:

manual labeling → corpus → train → DST model → test

Limitations of end-to-end DST:

Limitation of end-to-end DST

End-to-end DST paradigm:

manual labeling → corpus → train → DST model → test

Limitations of end-to-end DST:

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

Limitation of end-to-end DST

End-to-end DST paradigm:

manual labeling → corpus → **train** → DST model → **test**

Limitations of end-to-end DST:

- 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.

Limitation of end-to-end DST

End-to-end DST paradigm:

manual labeling → corpus → **train** → DST model → **test**

Limitations of end-to-end DST:

- 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]

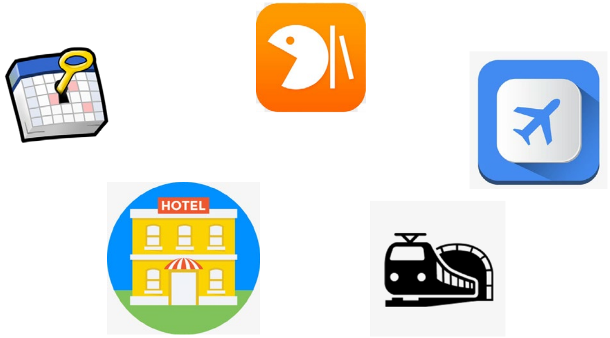
now updated to → MultiWOZ 2.2

[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.

What is the problem?

What is the problem?

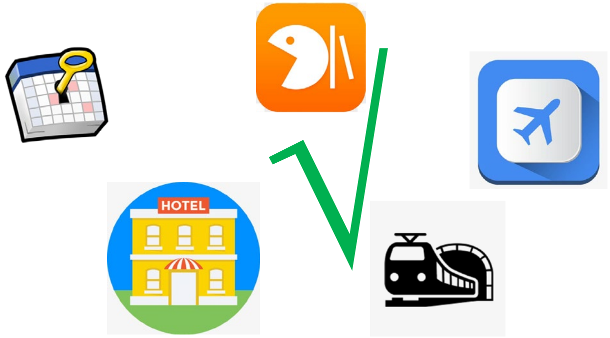
Successful in narrow domains with large annotated datasets



CHAPTER 1

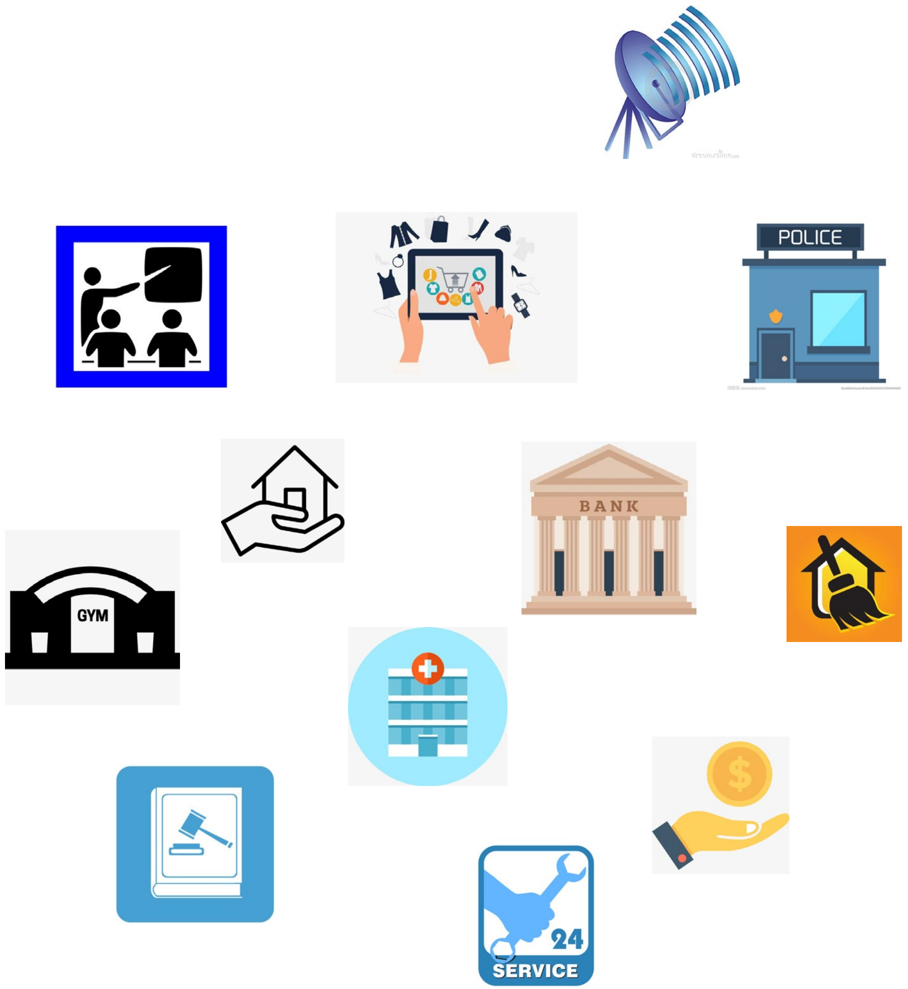
What is the problem?

Successful in narrow domains with large annotated datasets



What is the problem?

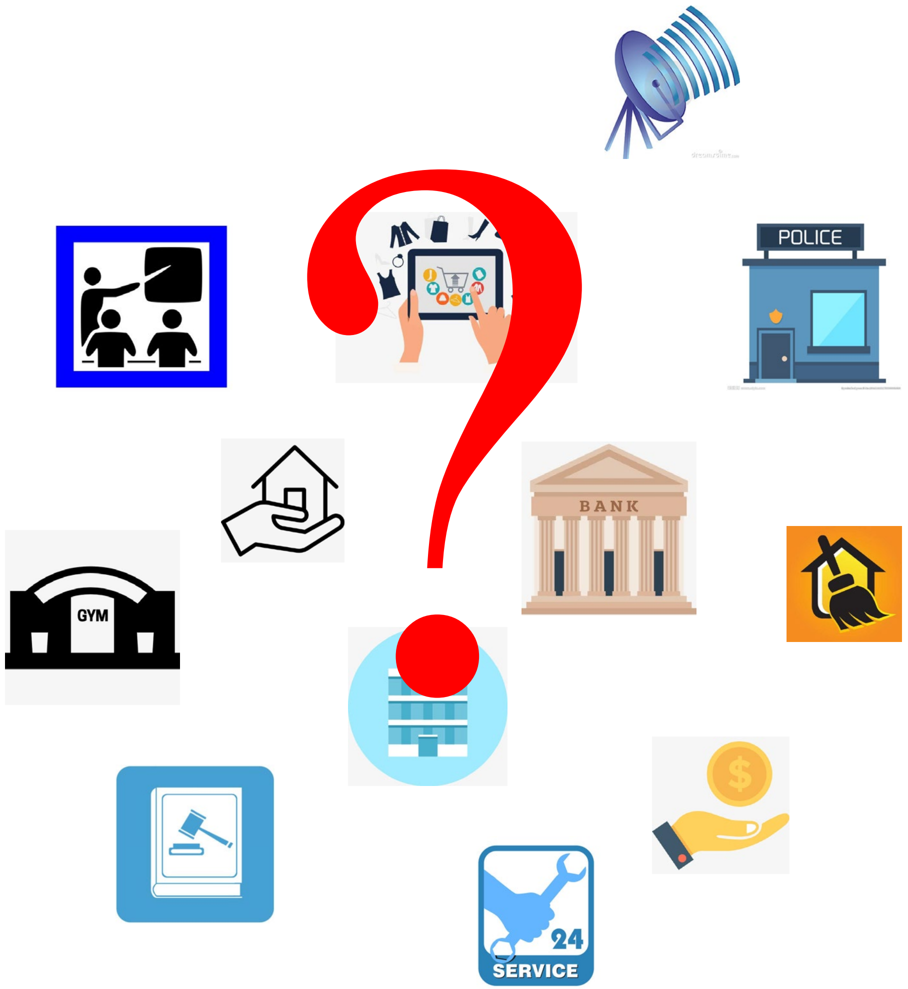
Successful in narrow domains with large annotated datasets



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

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.

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)
```

Dialogue State Induction vs DST

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?

Dialogue State Induction vs DST

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)`

Dialogue State Induction vs DST

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, ...]

Dialogue State Induction vs DST

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)`

Dialogue State Induction vs DST

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.

System: Anatolia serves Turkish food.
User: What is the area?

Dialogue State Induction vs DST

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, ...]~~

Dialogue State Induction vs DST

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, ...]~~

DSI

Turn 1:
`inform(price=expensive, food=Turkish)`

Turn 2:
`inform(price=expensive, food=Turkish); request(area)`

DSI VS zero-shot DST

Zero-shot DST: support unseen domains (services)

DSI VS zero-shot DST

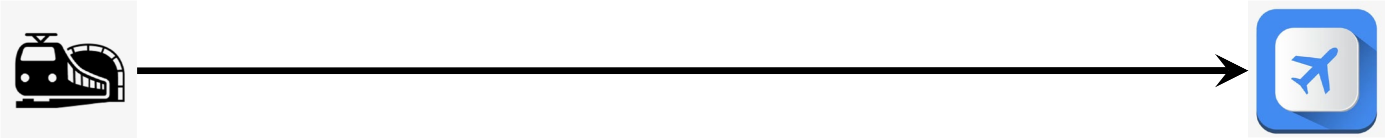
Zero-shot DST: support unseen domains (services)

Motivation: Different domains (services) with similar schemas

DSI VS zero-shot DST

Zero-shot DST: support unseen domains (services)

Motivation: Different domains (services) with similar schemas



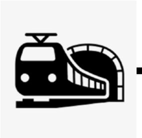
service_name: "Trains"	**Service**
name: "from"	**Slots**
name: "to"	

service_name: “Flights"	**Service**
name: "origin"	**Slots**
name: "destination"	

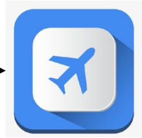
DSI VS zero-shot DST

Zero-shot DST: support unseen domains (services)

Motivation: Different domains (services) with similar schemas



slots along with their natural language description



Service service_name: "Trains" description: “Service to find train journeys between cities"	**Service** service_name: “Flights" description: "Search for cheap flights across multiple providers"
Slots name: "from" description: “Starting city for train journey"	**Slots** name: "origin" description: “City of origin for the flight"
name: "to" description: “Ending city for the train journey"	name: "destination" description: “City of destination for the flight"

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

[Rastogi et al., 2019]Rastogi A, Zang X, Sunkara S, et al. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset[J]. arXiv preprint arXiv:1909.05855, 2019.

DSI VS zero-shot DST

service_name: "Trains" description: “Service to find train journeys between cities"	**Service**
name: "from" description: “Starting city for train journey"	**Slots**
name: "to" description: “Ending city for the train journey"	

service_name: “Flights" description: "Search for cheap trains across multiple providers"	**Service**
name: "origin" description: “City of origin for the flight"	**Slots**
name: "destination" description: “City of destination for the flight"	

DSI VS zero-shot DST

service_name: "Trains" description: “Service to find train journeys between cities"	**Service**
name: "from" description: “Starting city for train journey"	**Slots**
name: "to" description: “Ending city for the train journey"	

service_name: “Flights" description: "Search for cheap trains across multiple providers"	**Service**
name: "origin" description: “City of origin for the flight"	**Slots**
name: "destination" description: “City of destination for the flight"	

Zero-shot DST Limitations:

DSI VS zero-shot DST

service_name: "Trains" description: “Service to find train journeys between cities"	**Service**
name: "from" description: “Starting city for train journey"	**Slots**
name: "to" description: “Ending city for the train journey"	

service_name: “Flights" description: "Search for cheap trains across multiple providers"	**Service**
name: "origin" description: “City of origin for the flight"	**Slots**
name: "destination" description: “City of destination for the flight"	

Zero-shot DST Limitations:

- High qualified (consistent) human annotation

DSI VS zero-shot DST

service_name: "Trains" description: “Service to find train journeys between cities"	**Service**
name: "from" description: “Starting city for train journey"	**Slots**
name: "to" description: “Ending city for the train journey"	

service_name: “Flights" description: "Search for cheap trains across multiple providers"	**Service**
name: "origin" description: “City of origin for the flight"	**Slots**
name: "destination" description: “City of destination for the flight"	

Zero-shot DST Limitations:

- High qualified (consistent) human annotation

DSI VS zero-shot DST

Service service_name: "Trains" description: “Service to find train journeys between cities"	**Service** service_name: “Flights" description: "Search for cheap trains across multiple providers"
Slots name: "from" description: “Starting city for train journey"	**Slots** name: "origin" description: “City of origin for the flight"
name: "to" description: “Ending city for the train journey"	name: "destination" description: “City of destination for the flight"

Zero-shot DST Limitations:

- High qualified (consistent) human annotation

DSI VS zero-shot DST

Service
service_name: "Trains"
description: “Service to find train journeys between cities"

Slots
name: "from"
description: “Starting city for train journey"

name: "to"
description: “Ending city for the train journey"

consistency

Service
service_name: “Flights"
description: "Search for cheap trains across multiple providers"

Slots
name: "origin"
description: “City of origin for the flight"

name: "destination"
description: “City of destination for the flight"

Zero-shot DST Limitations:

- High qualified (consistent) human annotation

DSI VS zero-shot DST

Service service_name: "Trains" description: “Service to find train journeys between cities"		**Service** service_name: “Flights" description: "Search for cheap trains across multiple providers"
Slots name: "from" description: “Starting city for train journey"	consistency	**Slots** name: "origin" description: “City of origin for the flight"
name: "to" description: "Ending city for the train journey"		name: "destination" description: “City of destination for the flight"

Zero-shot DST Limitations:

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

DSI VS zero-shot DST

Service
service_name: "Trains"
description: “Service to find train journeys between cities"

Slots
name: "from"
description: "Starting city for train journey"

name: "to"
description: "Ending city for the train journey"

consistency

Service
service_name: "Financial"
description: “Search for financial products across multiple providers"

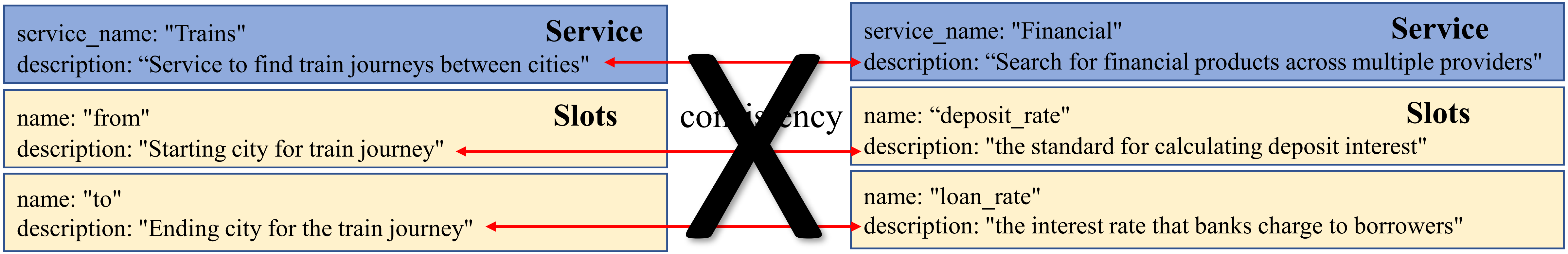
Slots
name: “deposit_rate"
description: "the standard for calculating deposit interest"

name: "loan_rate"
description: "the interest rate that banks charge to borrowers"

Zero-shot DST Limitations:

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

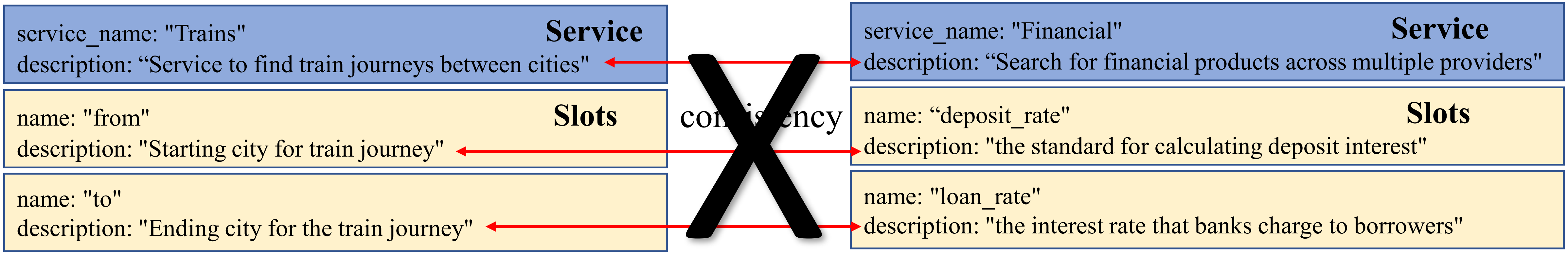
DSI VS zero-shot DST



Zero-shot DST Limitations:

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

DSI VS zero-shot DST



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

WESTLAKE UNIVERSITY

Two steps:

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

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

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*)

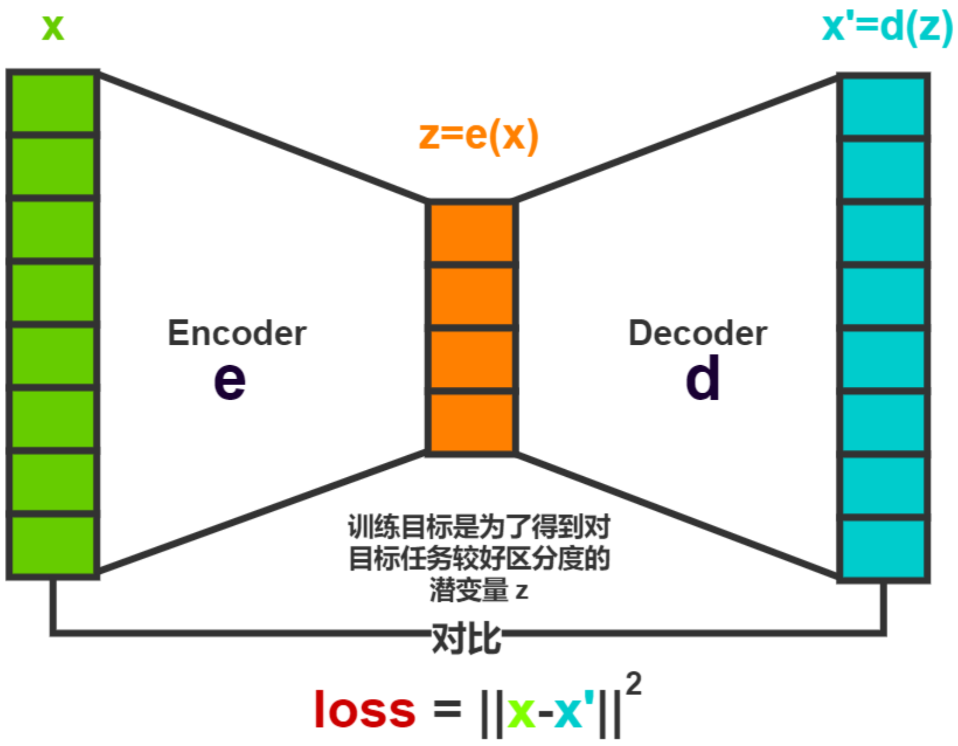
Utterance: I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.

train, Chicago, Dallas, Wednesday

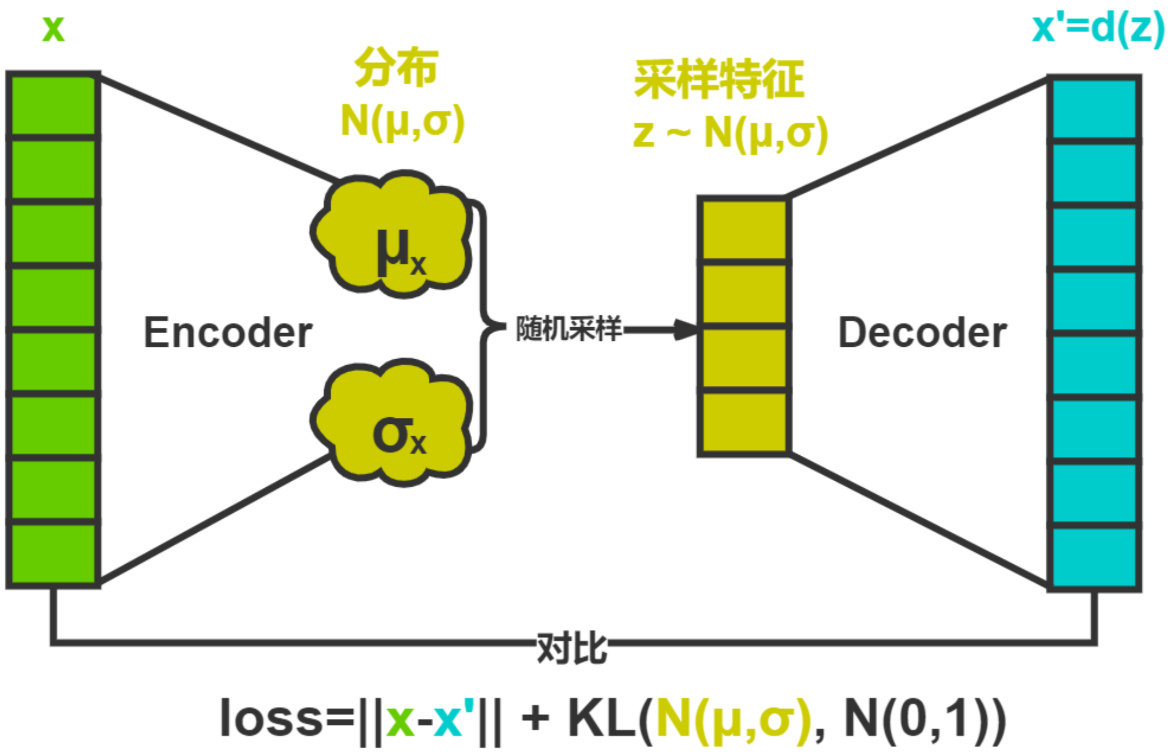
```
Inform{train-departure=Chicago,
train-destination=Dallas,
train-leave at=Wednesday}
```

~~train=None~~

What is VAE?



AutoEncoder



Variational AutoEncoder

Pics from https://www.jianshu.com/p/ffd493e10751

Model: generative *DSI-base*

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

$\left.\vphantom{\begin{matrix}a\\b\\c\\d\\e\\f\end{matrix}}\right\}$ **Encoder**

Model: generative *DSI-base*

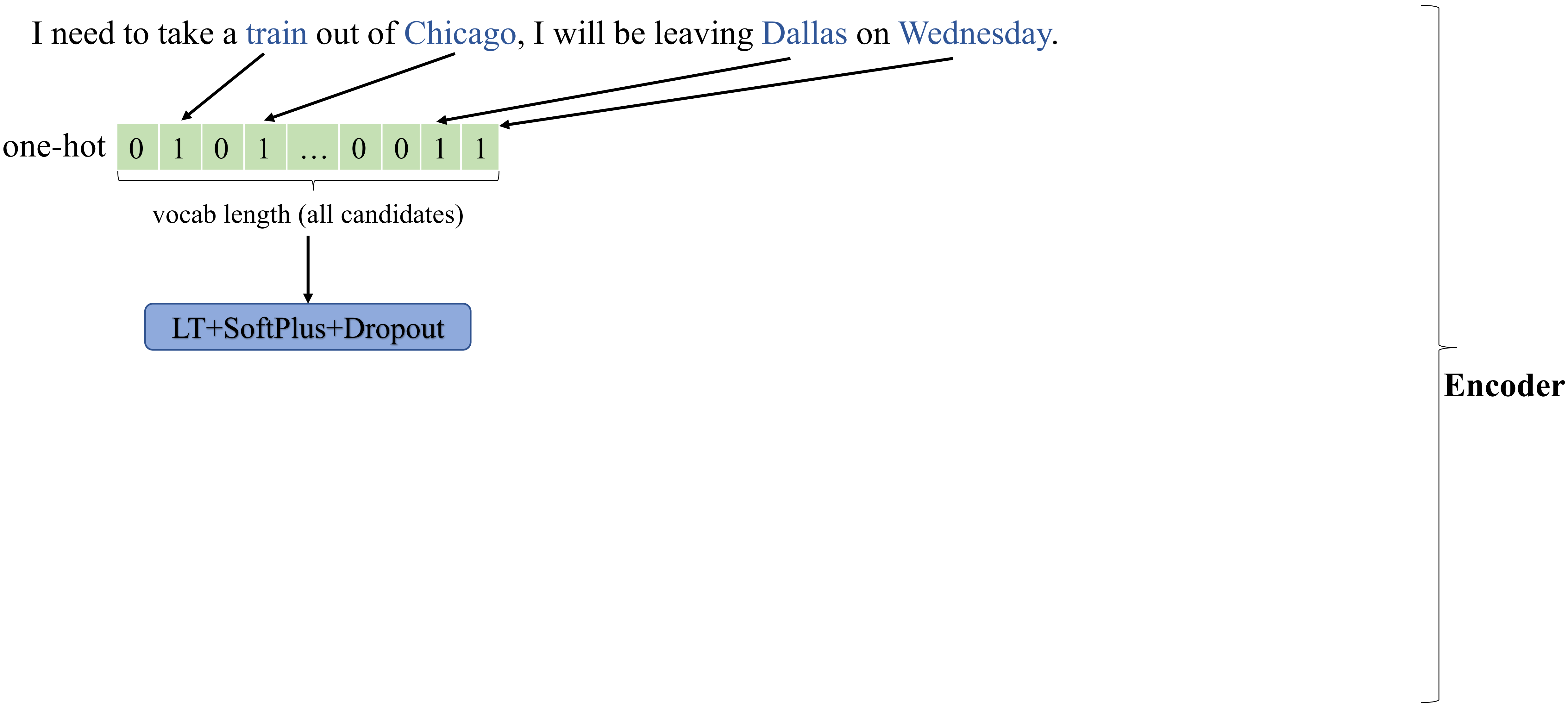
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.

one-hot									
	0	1	0	1	…	0	0	1	1

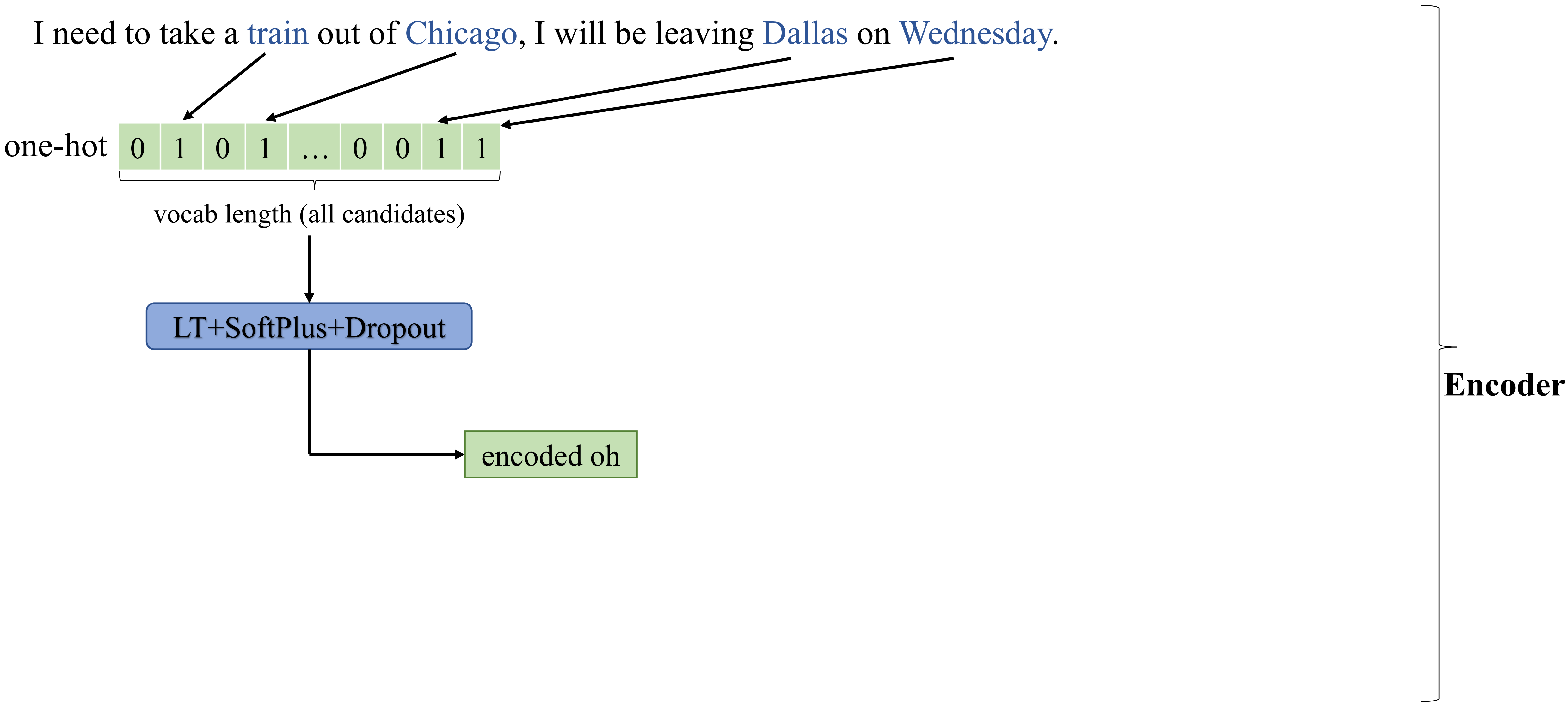
vocab length (all candidates)

Encoder

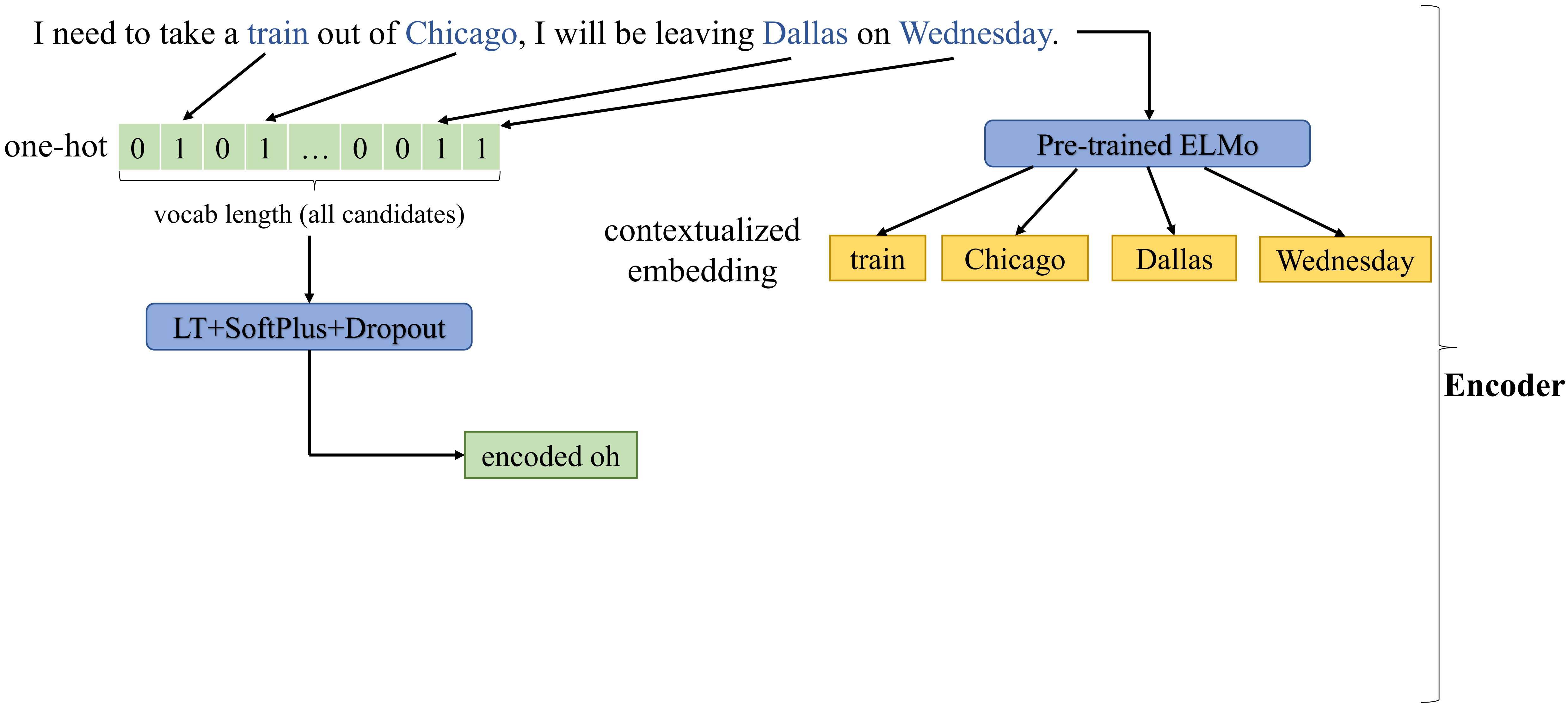
Model: generative *DSI-base*



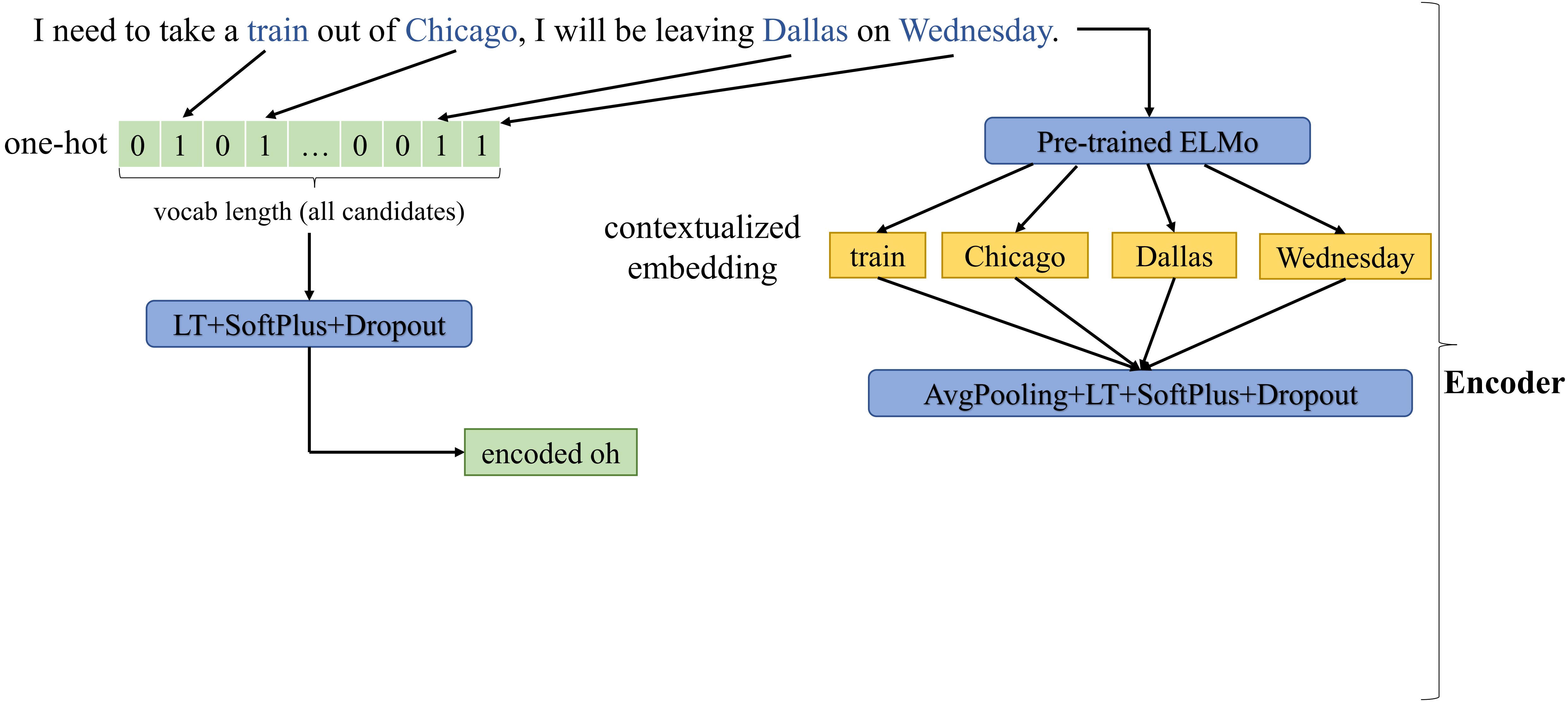
Model: generative *DSI-base*



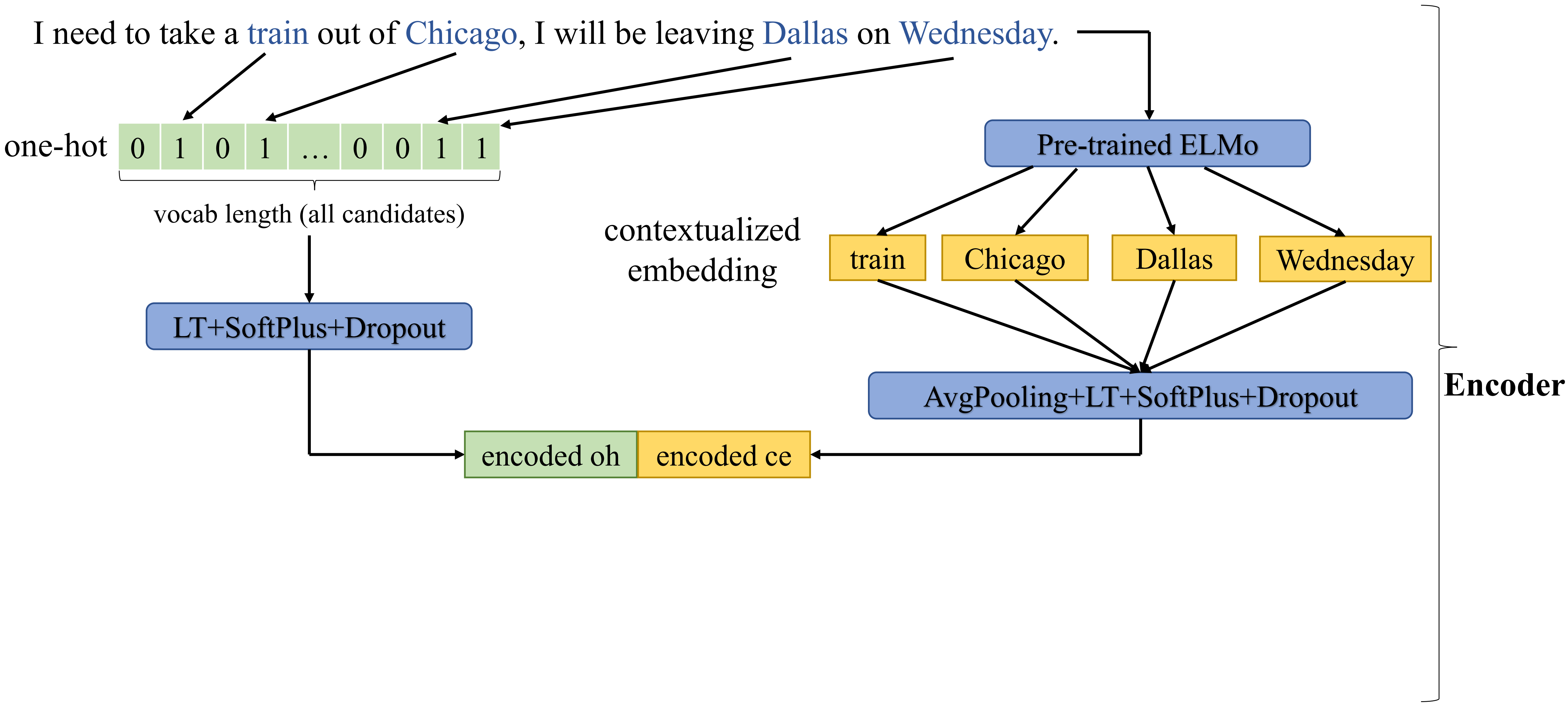
Model: generative *DSI-base*



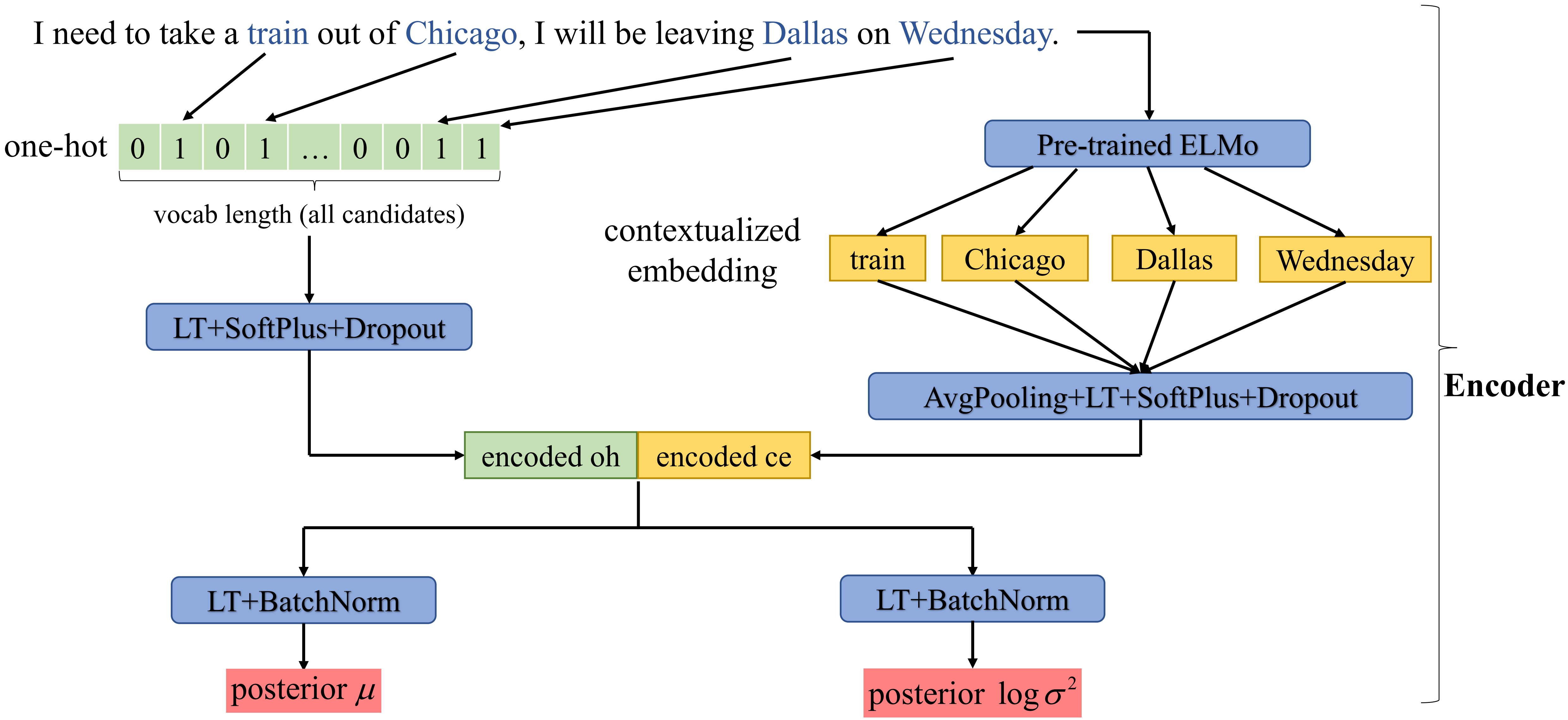
Model: generative *DSI-base*



Model: generative *DSI-base*



Model: generative *DSI-base*



Model: generative *DSI-base*

posterior μ

posterior $\log \sigma^2$

Model: generative *DSI-base*

posterior μ

posterior $\log \sigma^2$

Sampling

Model: generative *DSI-base*

posterior μ

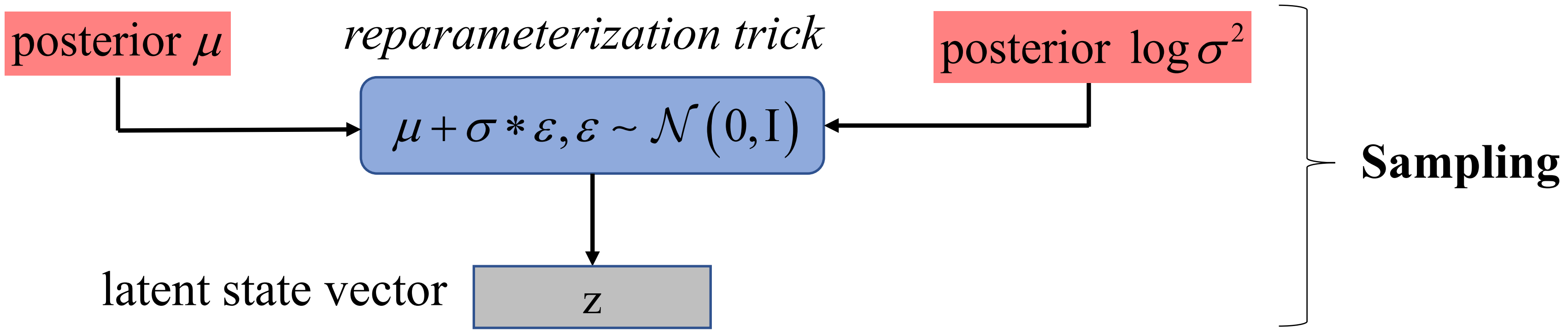
reparameterization trick

$$\mu + \sigma * \varepsilon, \varepsilon \sim \mathcal{N}(0, \mathrm{I})$$

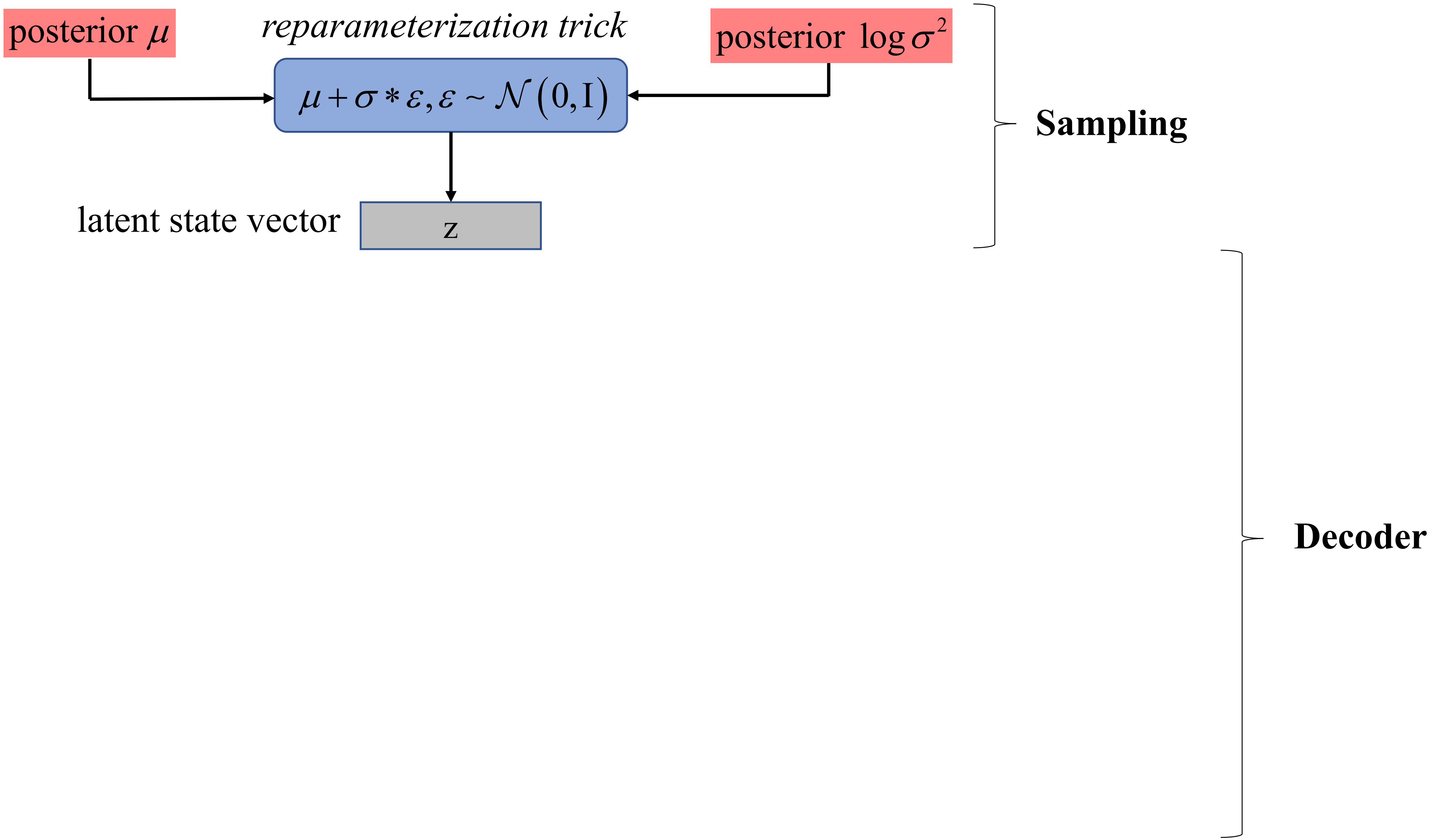
posterior $\log \sigma^2$

Sampling

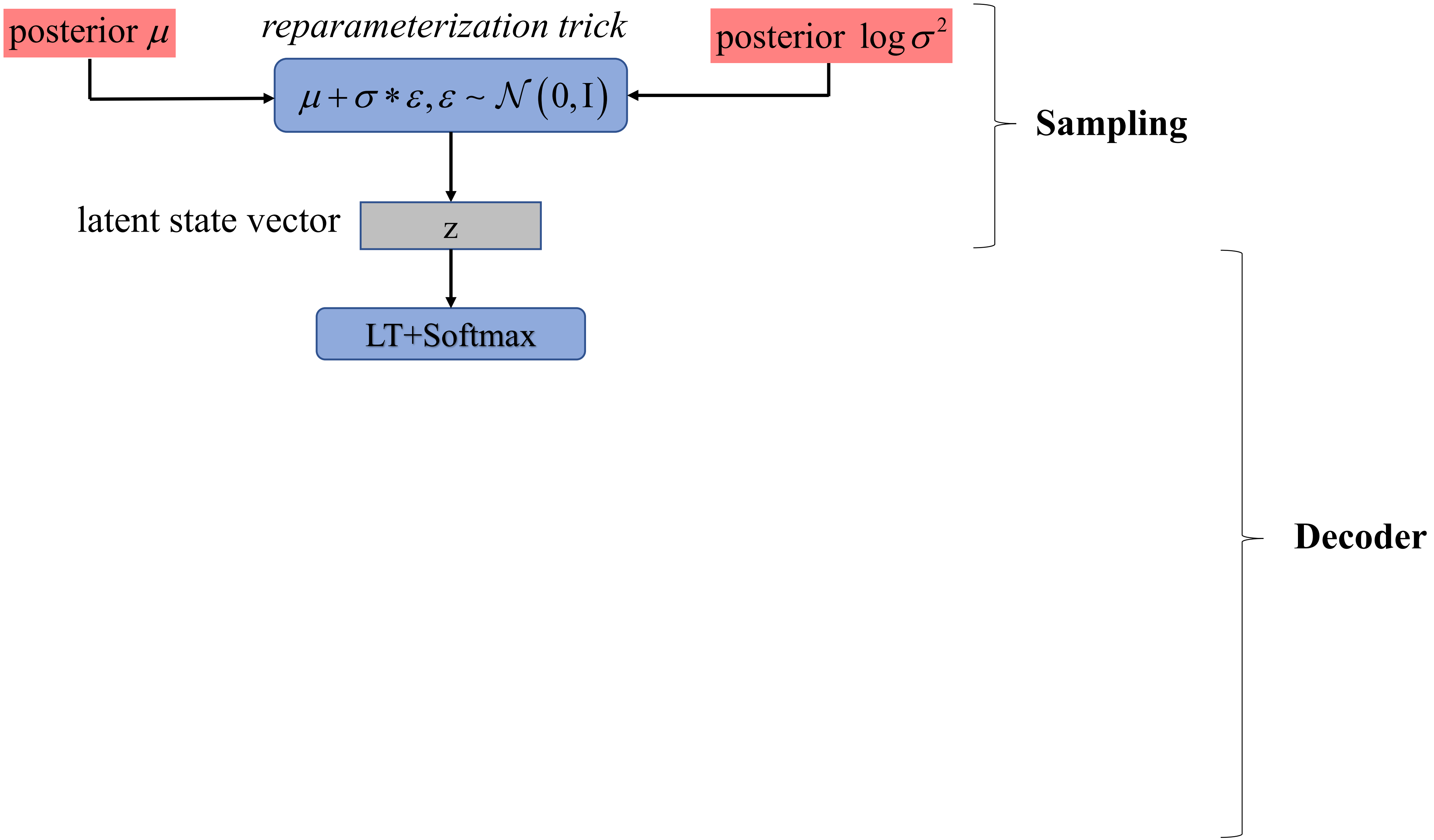
Model: generative *DSI-base*



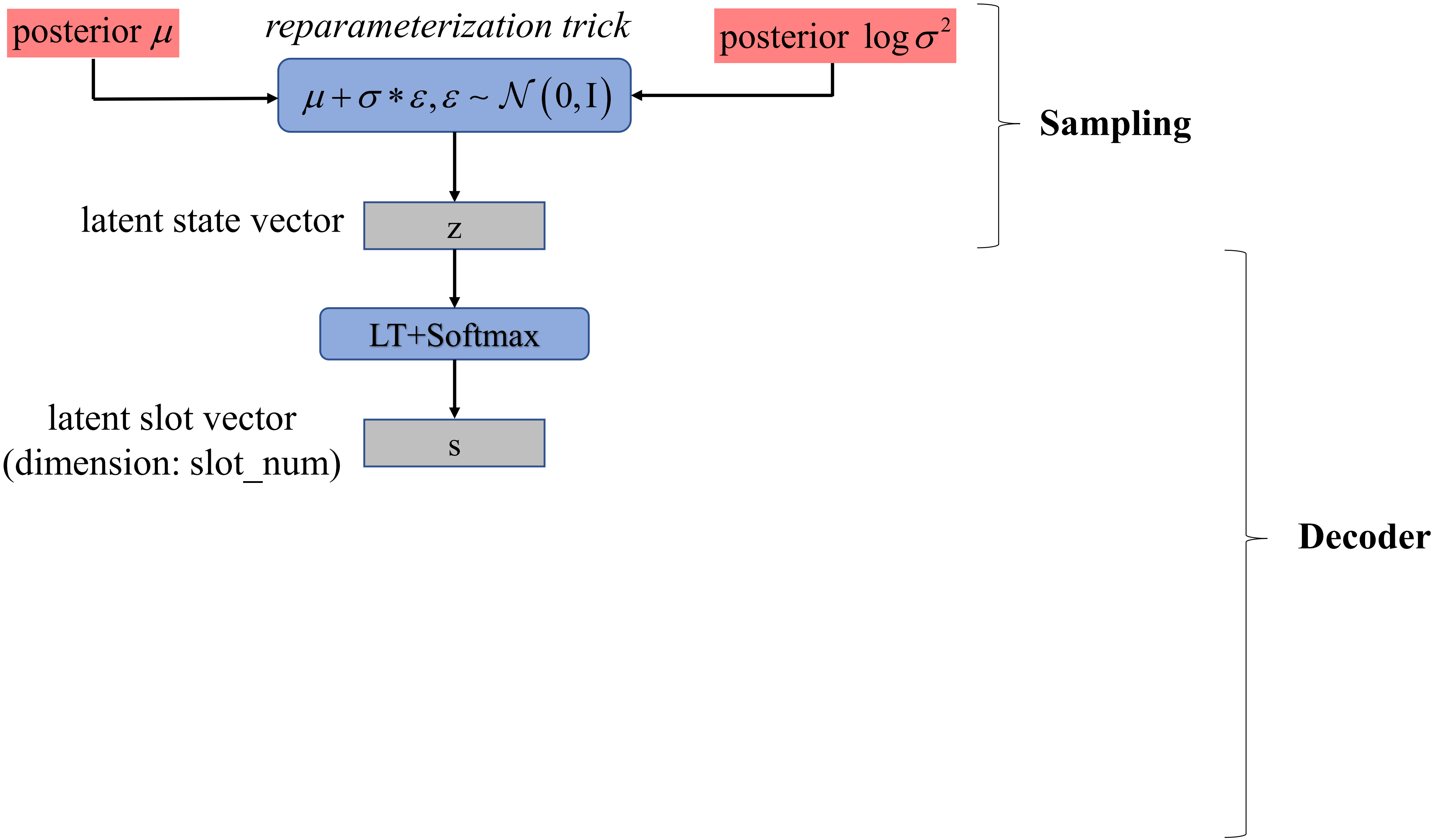
Model: generative *DSI-base*



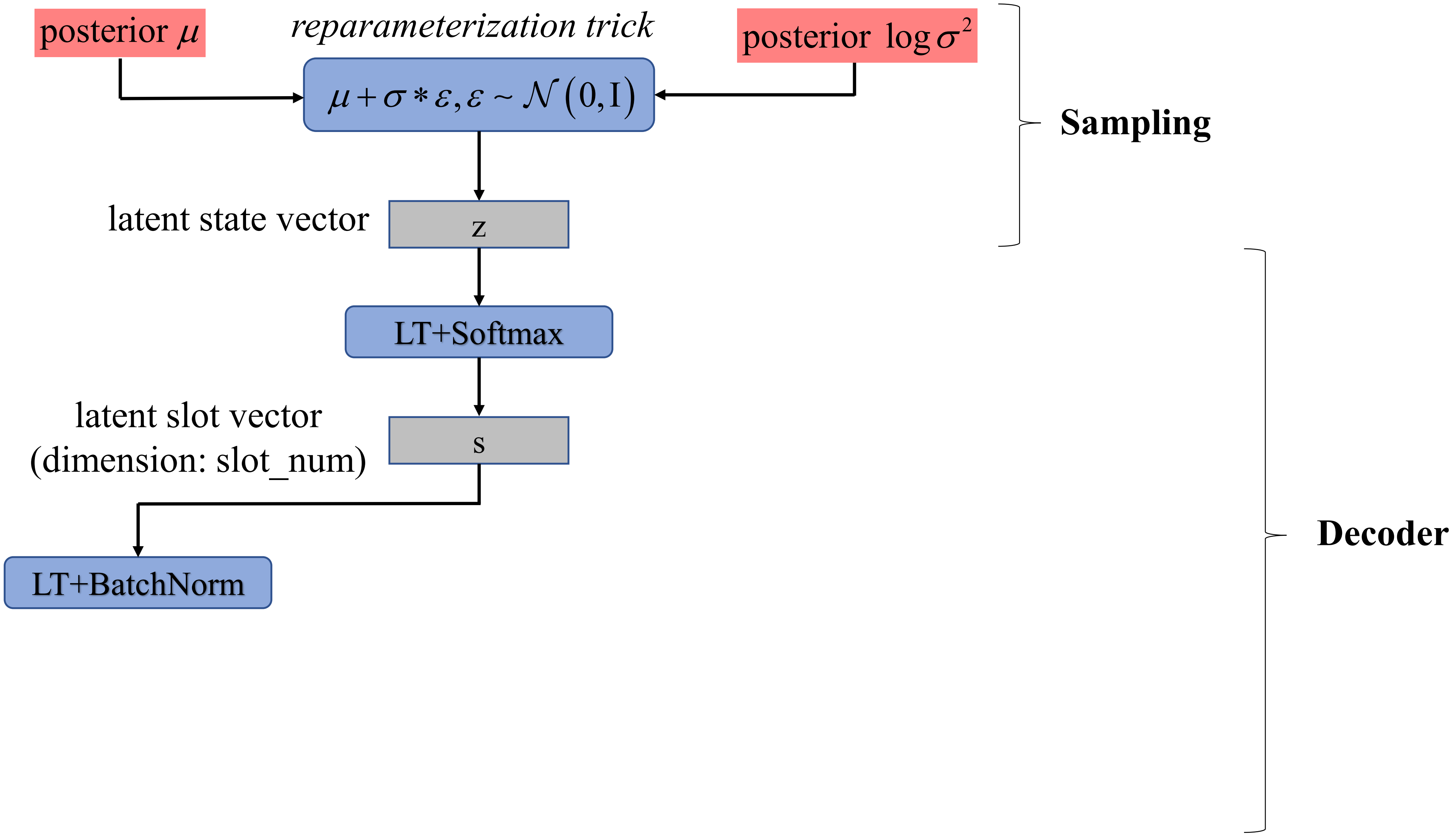
Model: generative *DSI-base*



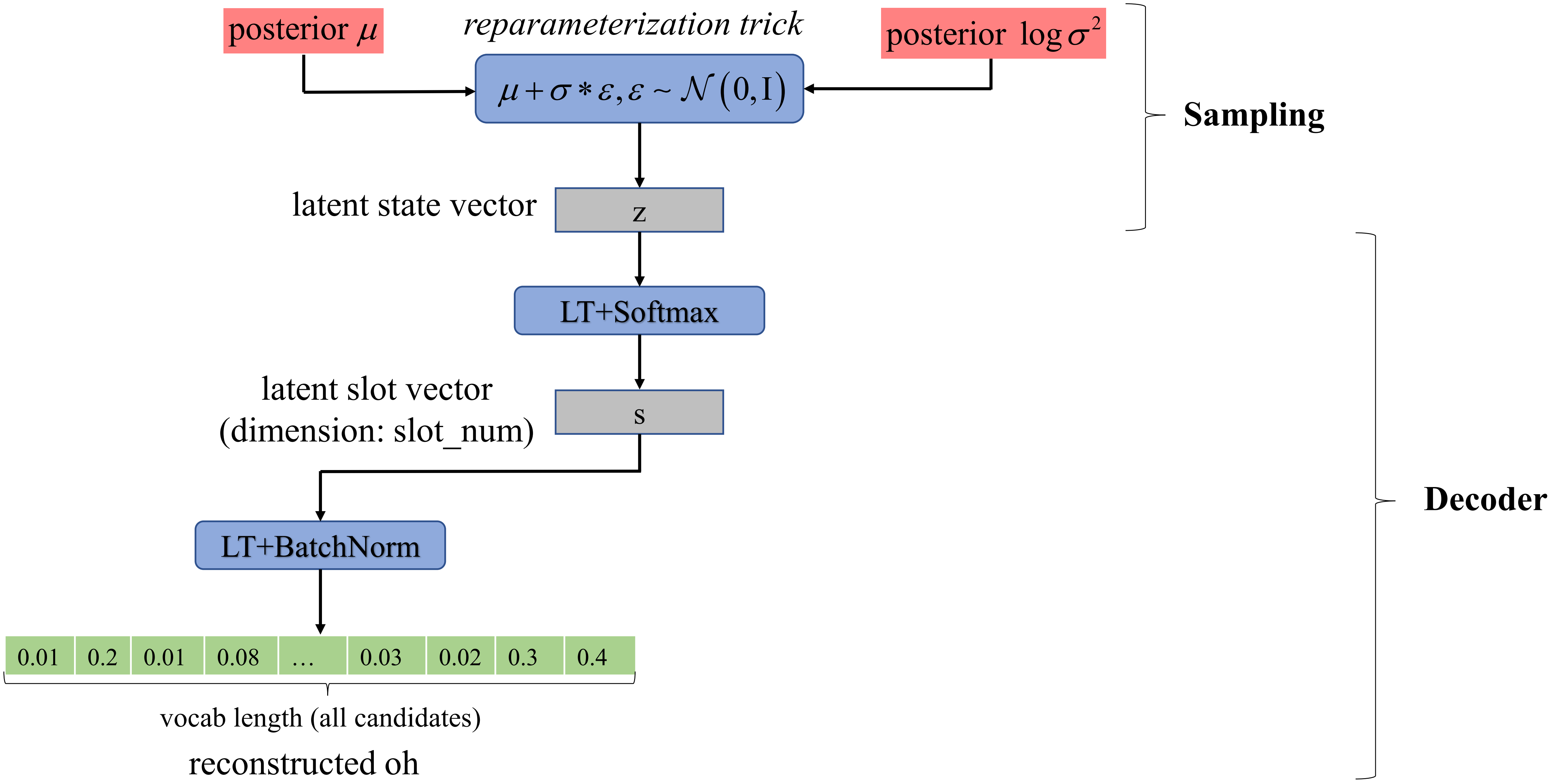
Model: generative *DSI-base*



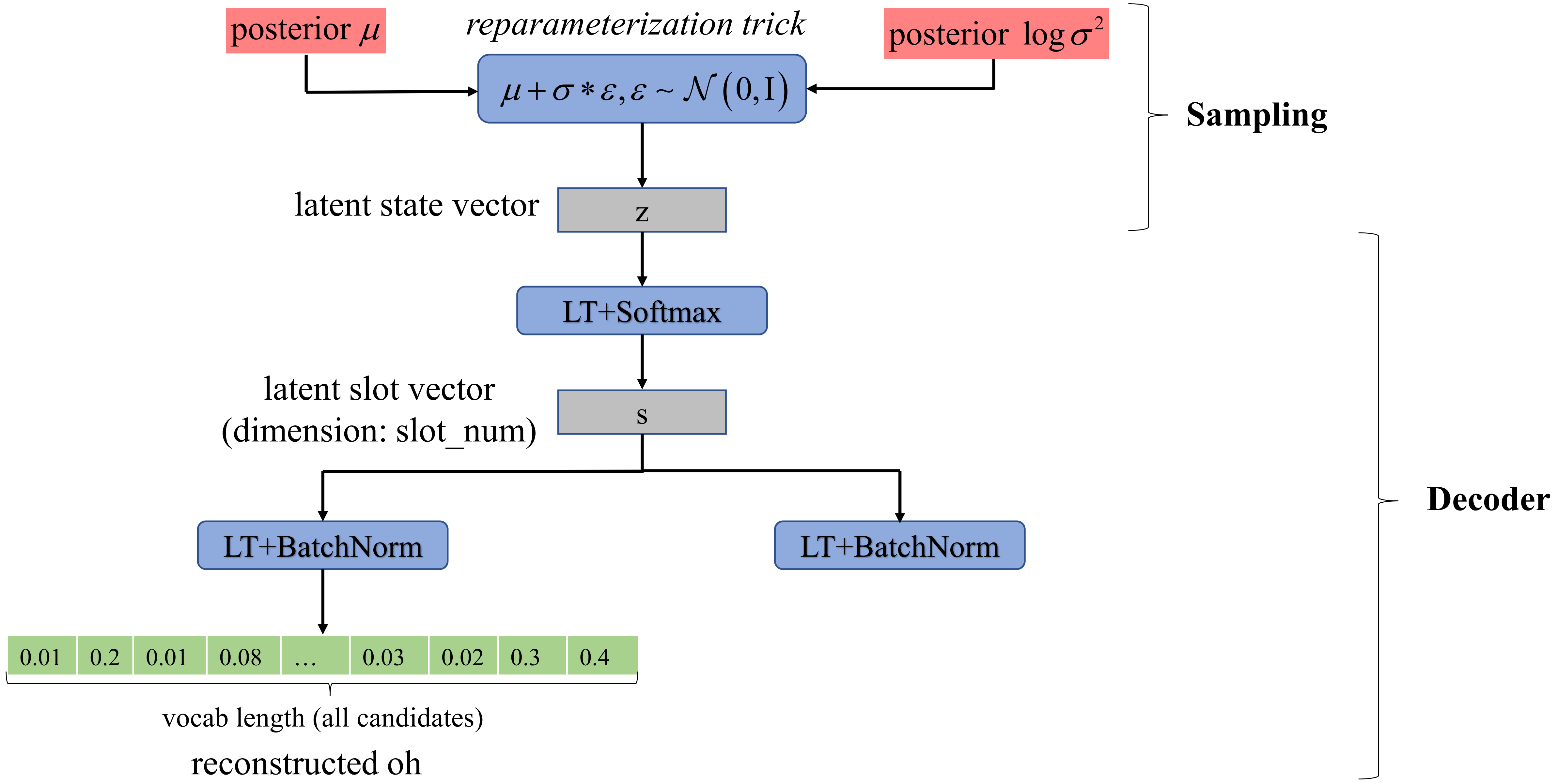
Model: generative *DSI-base*



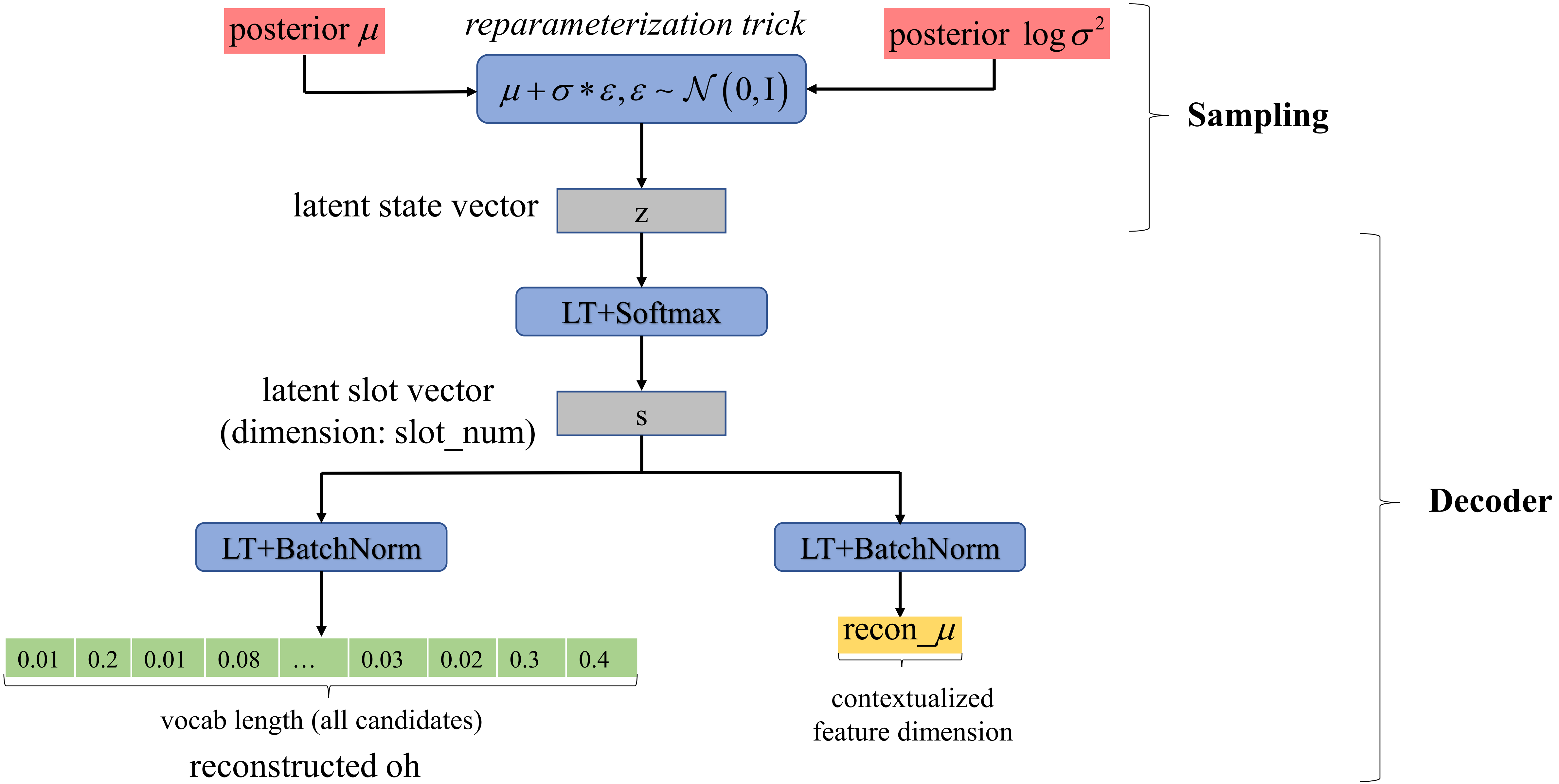
Model: generative *DSI-base*



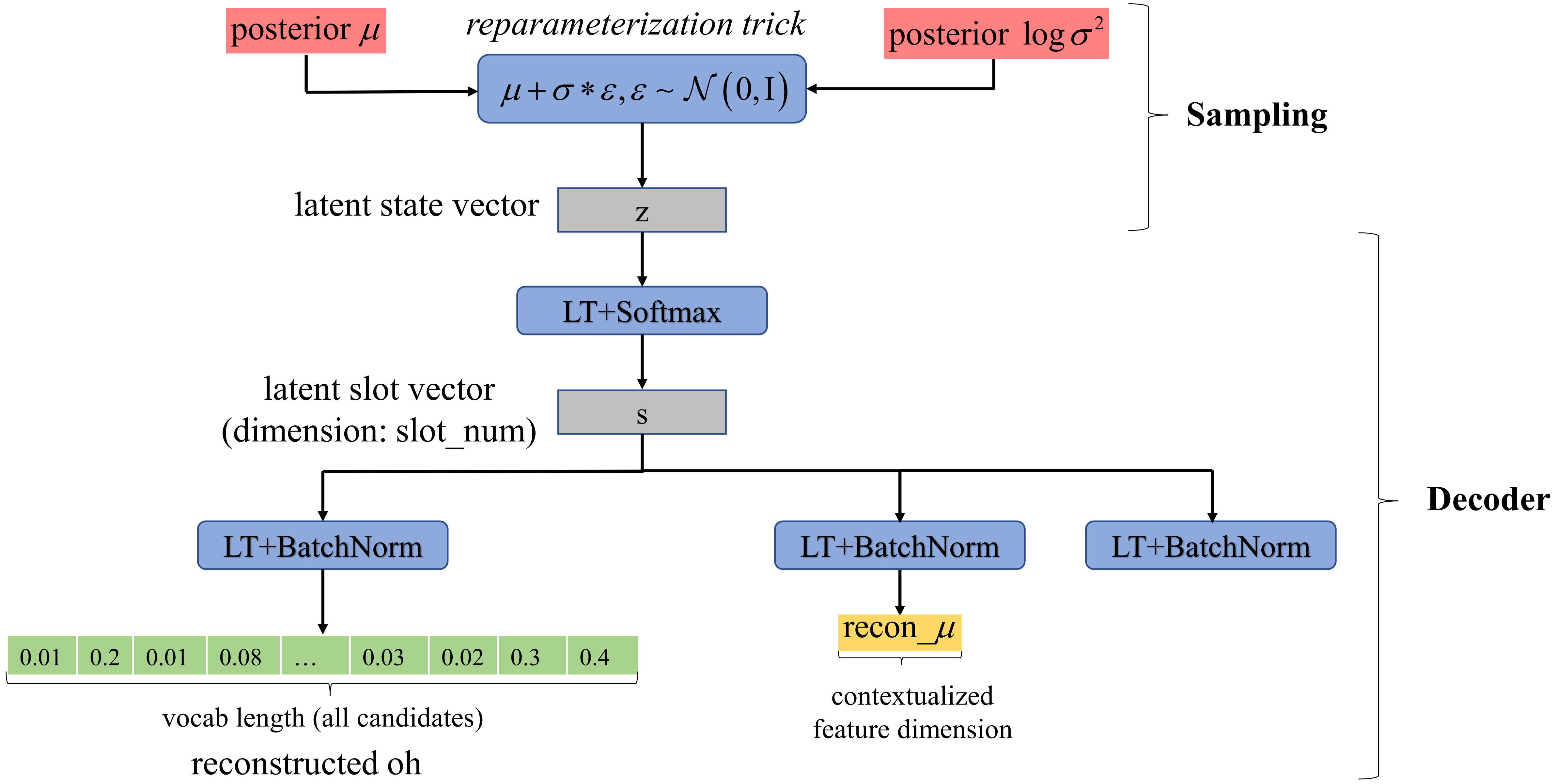
Model: generative *DSI-base*



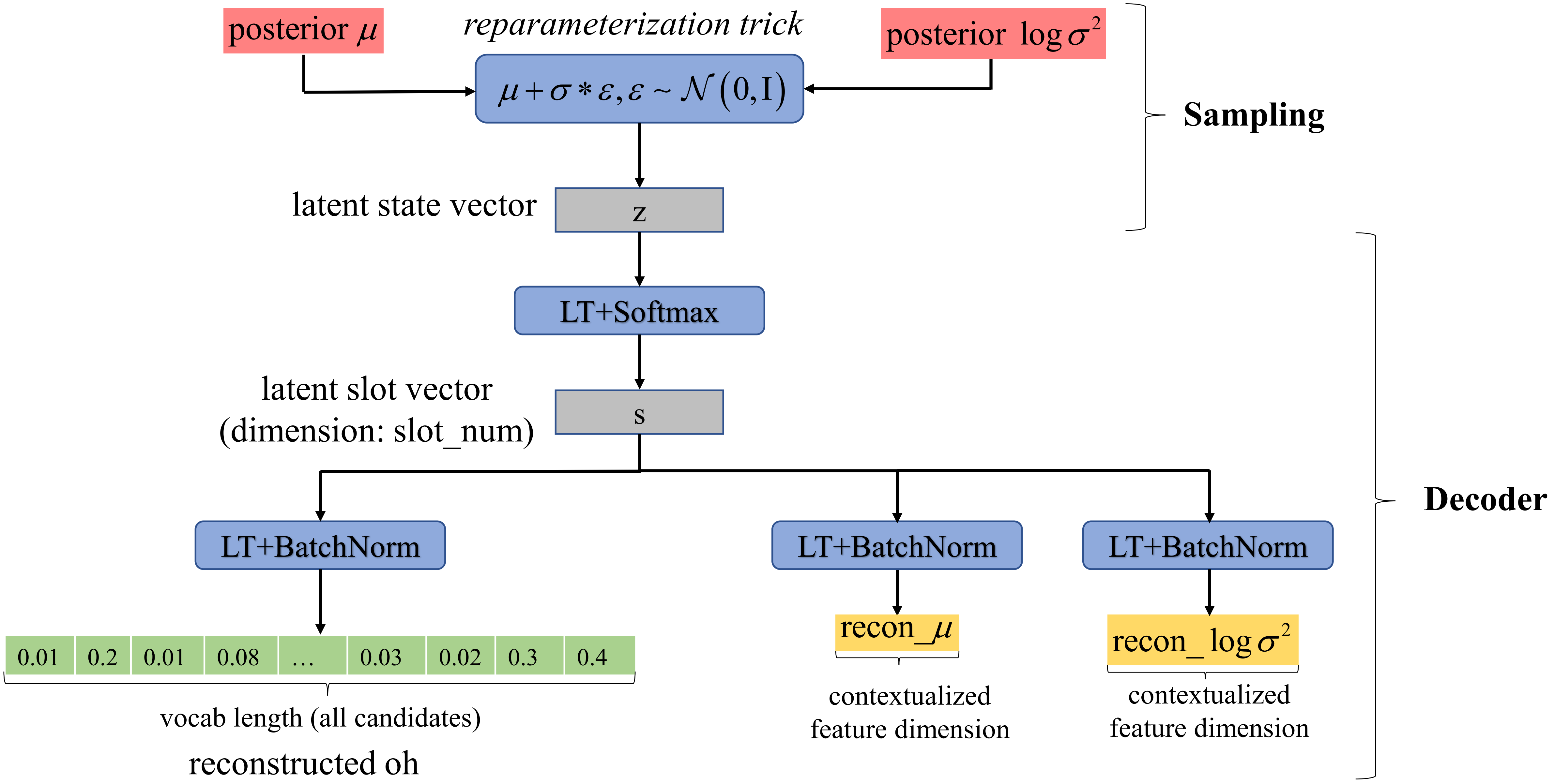
Model: generative *DSI-base*



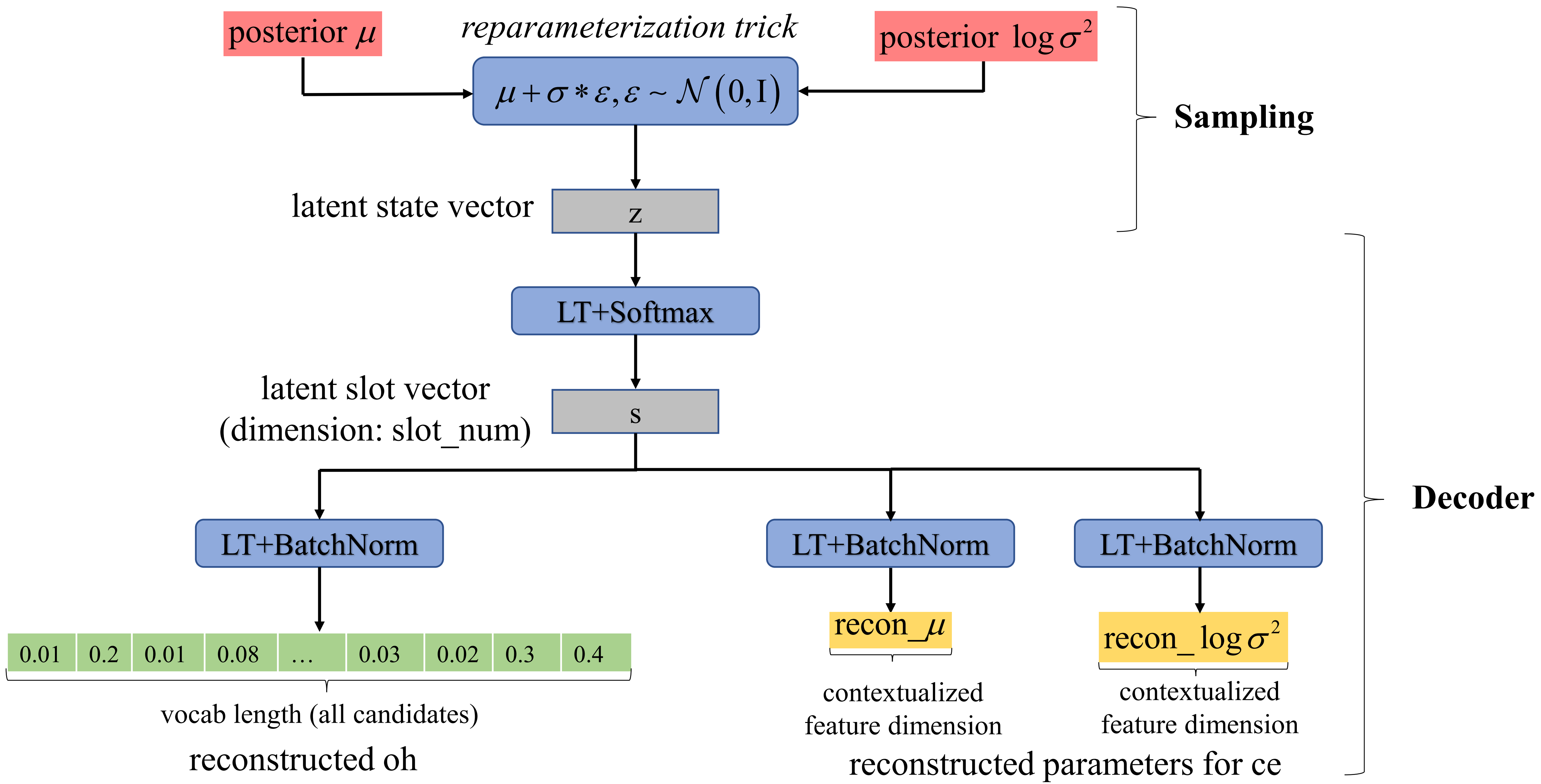
Model: generative *DSI-base*



Model: generative *DSI-base*



Model: generative *DSI-base*



reconstruction term

regularization term

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 Loss

reconstruction term

regularization term

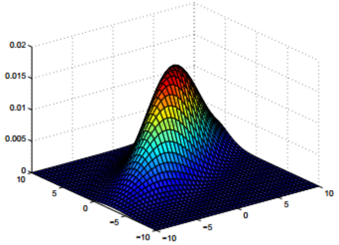
0.01	0.2	0.01	0.08	…	0.03	0.02	0.3	0.4	reconstructed oh
0	1	0	1	…	0	0	1	1	oh

vocab length (all candidates)

Cross Entropy Loss

$\underbrace{\text{recon\_}\mu}_{\text{contextualized feature dimension}}$

$\underbrace{\text{recon\_}\log\sigma^2}_{\text{contextualized feature dimension}}$



Multivariate Gaussian Distribution

reconstruction term

regularization term

Loss

0.01	0.2	0.01	0.08	…	0.03	0.02	0.3	0.4	reconstructed oh
0	1	0	1	…	0	0	1	1	oh

vocab length (all candidates)

Cross Entropy Loss

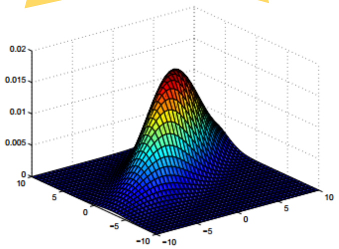
train | Chicago | Dallas | Wednesday

recon_μ

contextualized feature dimension

recon_ $\log\sigma^2$

contextualized feature dimension



Multivariate Gaussian Distribution

reconstruction term

regularization term

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 Loss

train

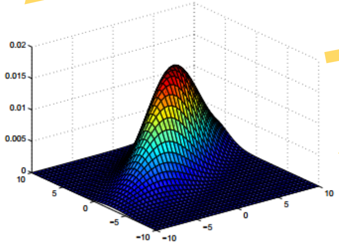
Chicago

Dallas

Wednesday

$\underbrace{\text{recon\_}\mu}_{\text{contextualized feature dimension}}$

$\underbrace{\text{recon\_}\log\sigma^2}_{\text{contextualized feature dimension}}$



Multivariate Gaussian Distribution

-dist.log_prob(ce[train])

-dist.log_prob(ce[Chicago])

-dist.log_prob(ce[Dallas])

-dist.log_prob(ce[Wednesday])

reconstruction term

regularization term

Pic from https://www.datalearner.com/blog/1051485590815771

Loss

0.01	0.2	0.01	0.08	…	0.03	0.02	0.3	0.4	reconstructed oh
0	1	0	1	…	0	0	1	1	oh

vocab length (all candidates)

Cross Entropy Loss

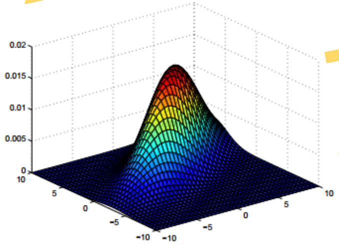
recon_μ

contextualized feature dimension

recon_ $\log\sigma^2$

contextualized feature dimension

train | Chicago | Dallas | Wednesday



Multivariate Gaussian Distribution

-dist.log_prob(ce[train])

-dist.log_prob(ce[Chicago])

-dist.log_prob(ce[Dallas])

-dist.log_prob(ce[Wednesday])

sum

reconstruction term

regularization term

Pic from https://www.datalearner.com/blog/1051485590815771

0.01	0.2	0.01	0.08	…	0.03	0.02	0.3	0.4	reconstructed oh
0	1	0	1	…	0	0	1	1	oh

vocab length (all candidates)

Cross Entropy Loss

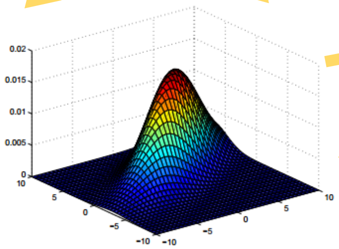
recon_μ

contextualized feature dimension

recon_ $\log\sigma^2$

contextualized feature dimension

train Chicago Dallas Wednesday



Multivariate Gaussian Distribution

-dist.log_prob(ce[train])

-dist.log_prob(ce[Chicago])

-dist.log_prob(ce[Dallas])

-dist.log_prob(ce[Wednesday])

sum

reconstruction term

KL Divergence between posterior and prior: $KL(\mathcal{N}(\mu,\sigma),\mathcal{N}(0,1))$

regularization term

Pic from https://www.datalearner.com/blog/1051485590815771

DSI-base inference

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

DSI-base inference

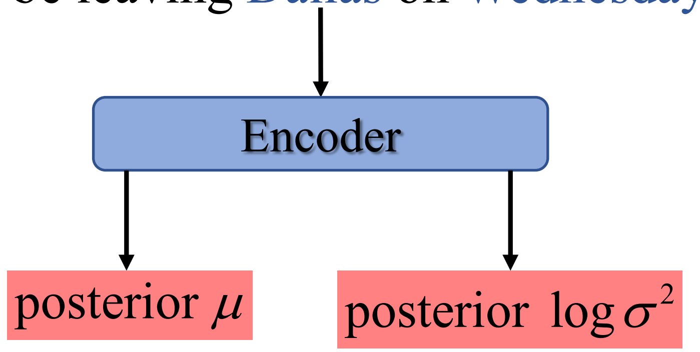
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.

↓

Encoder

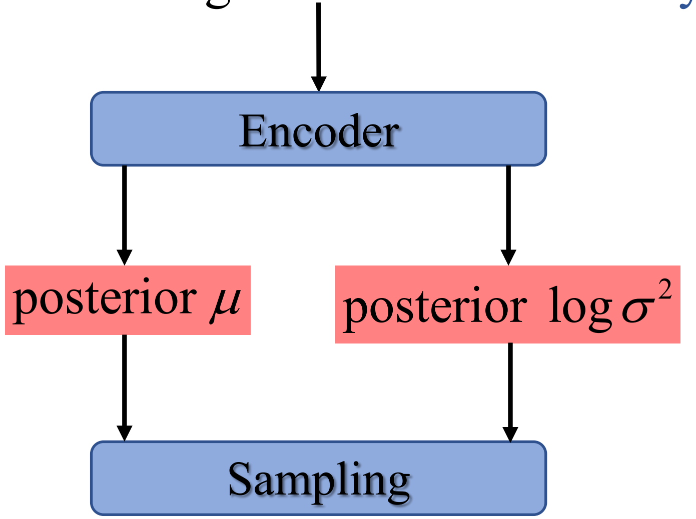
DSI-base inference

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

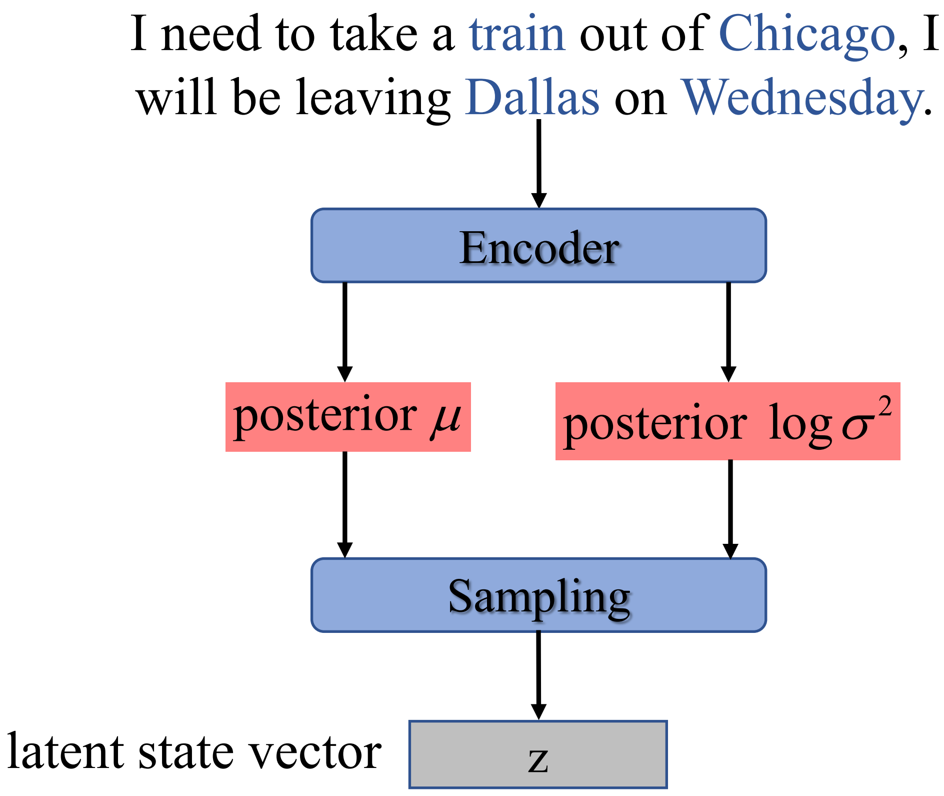


DSI-base inference

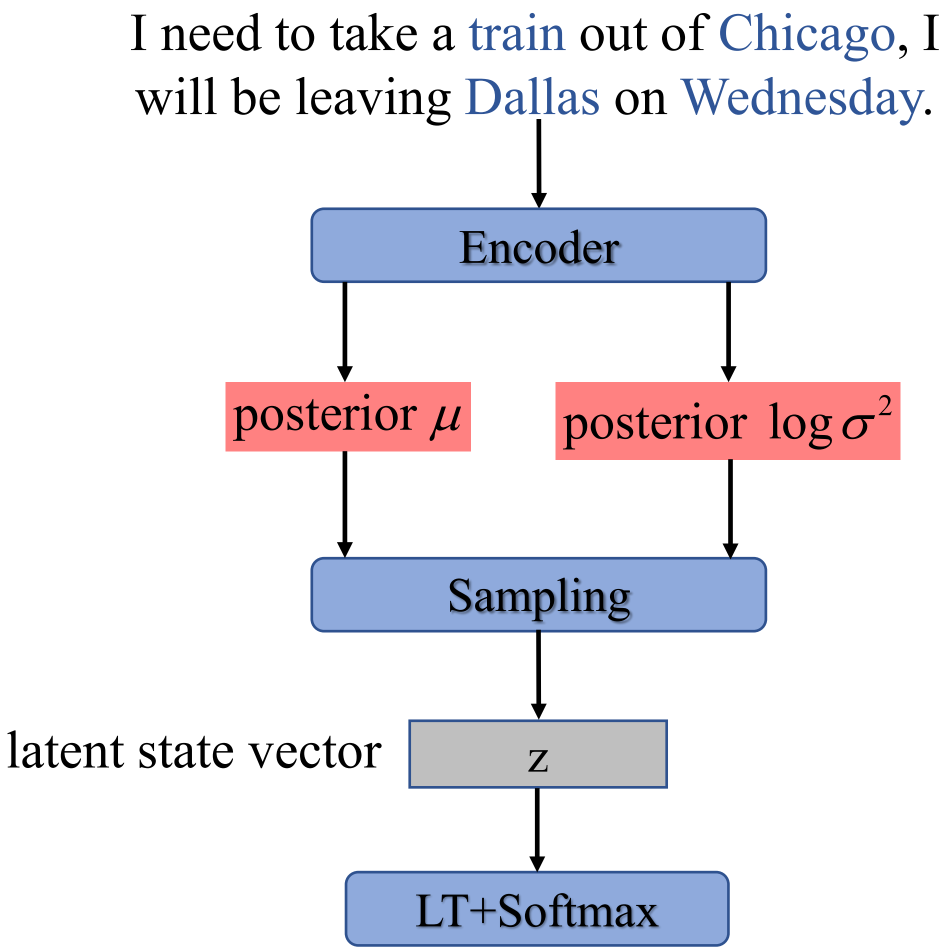
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



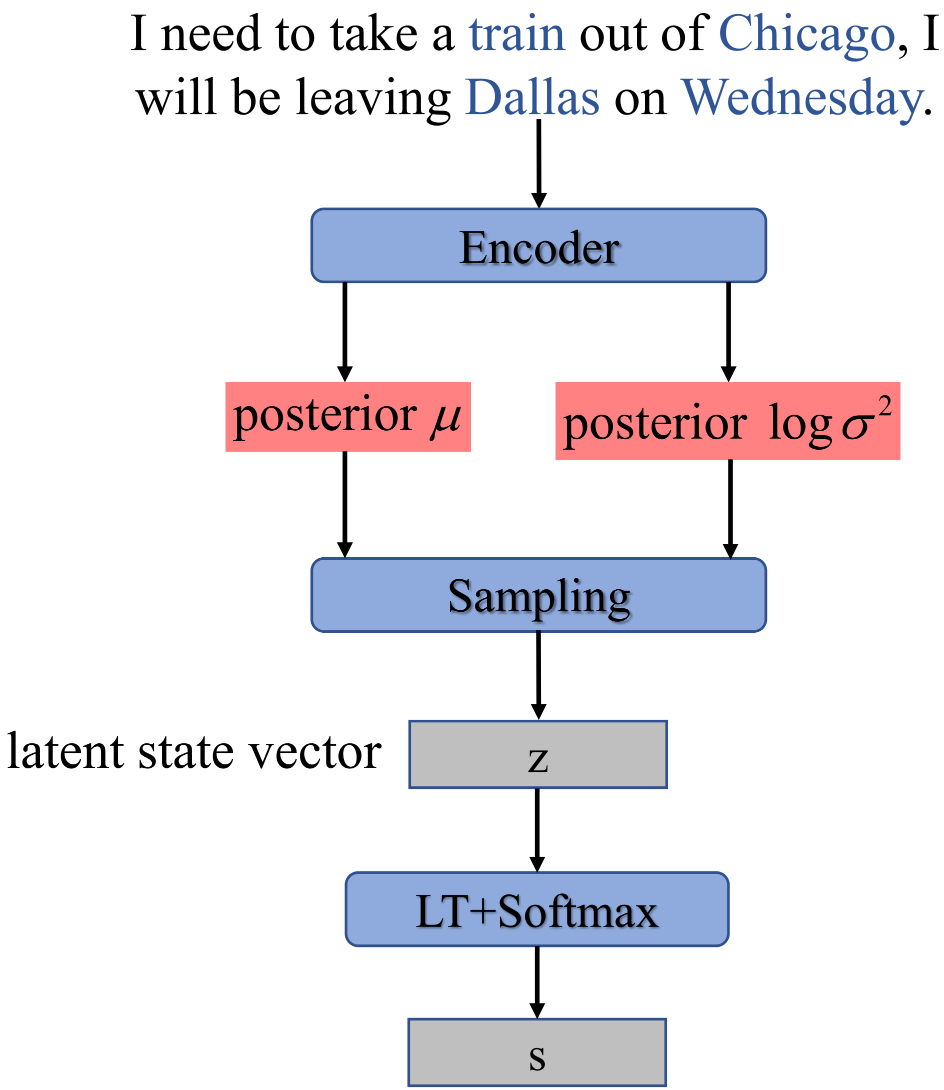
DSI-base inference



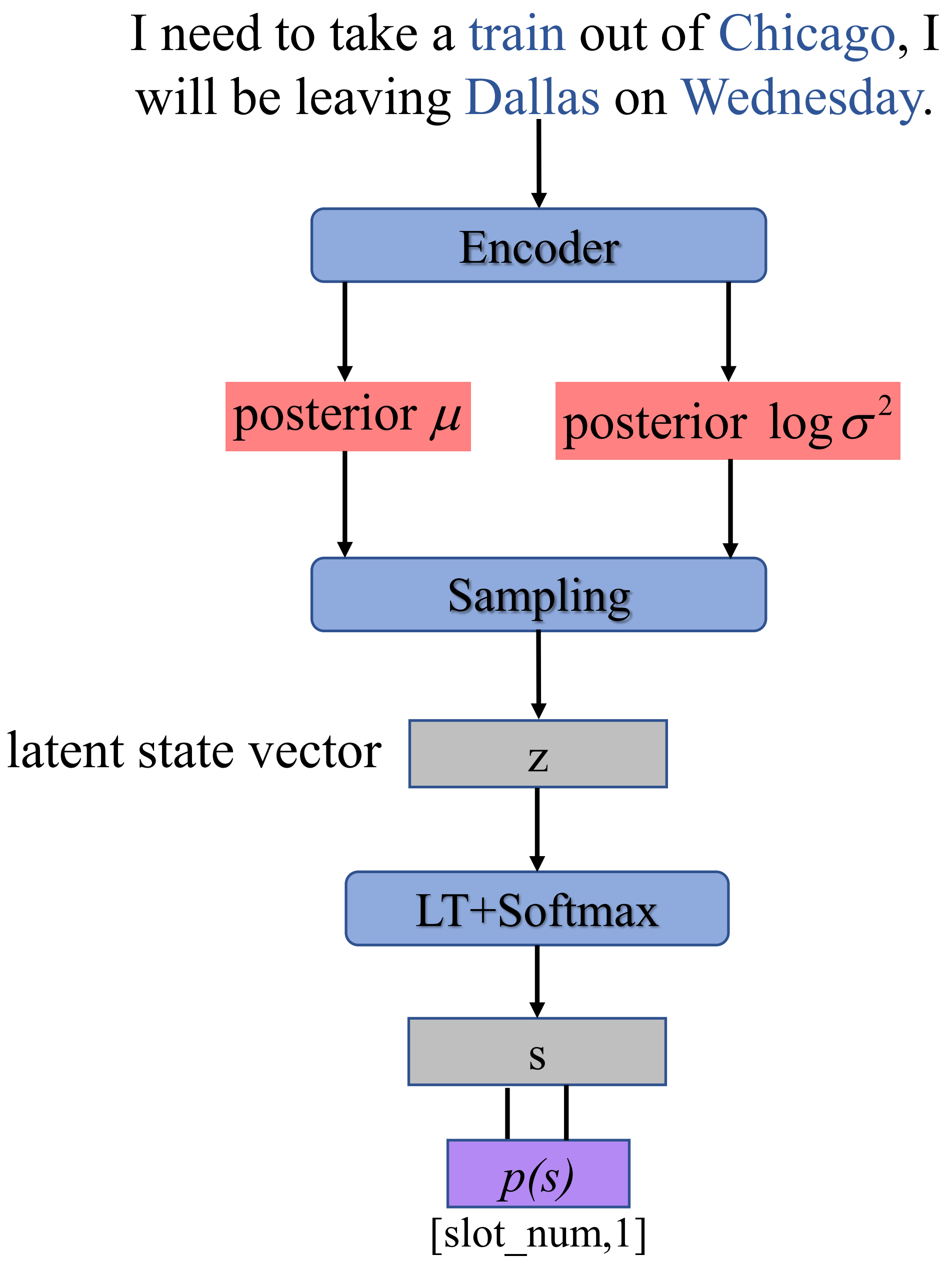
DSI-base inference



DSI-base inference

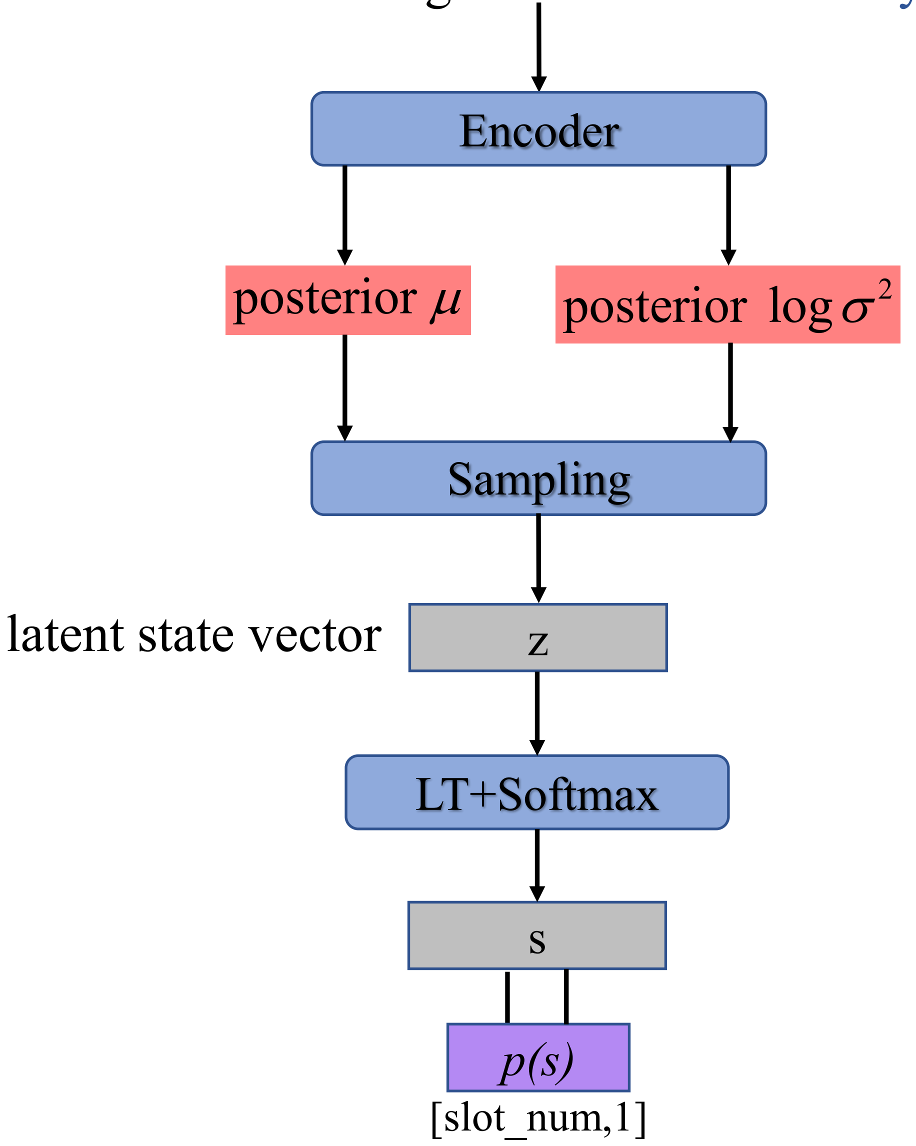


DSI-base inference



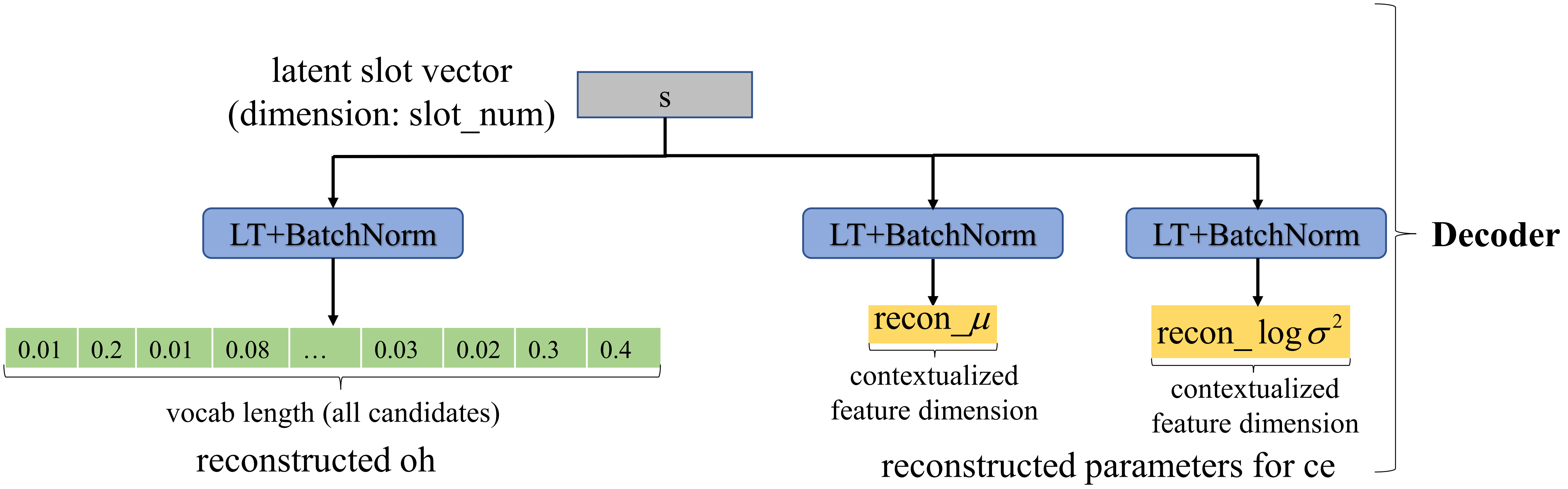
DSI-base inference

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

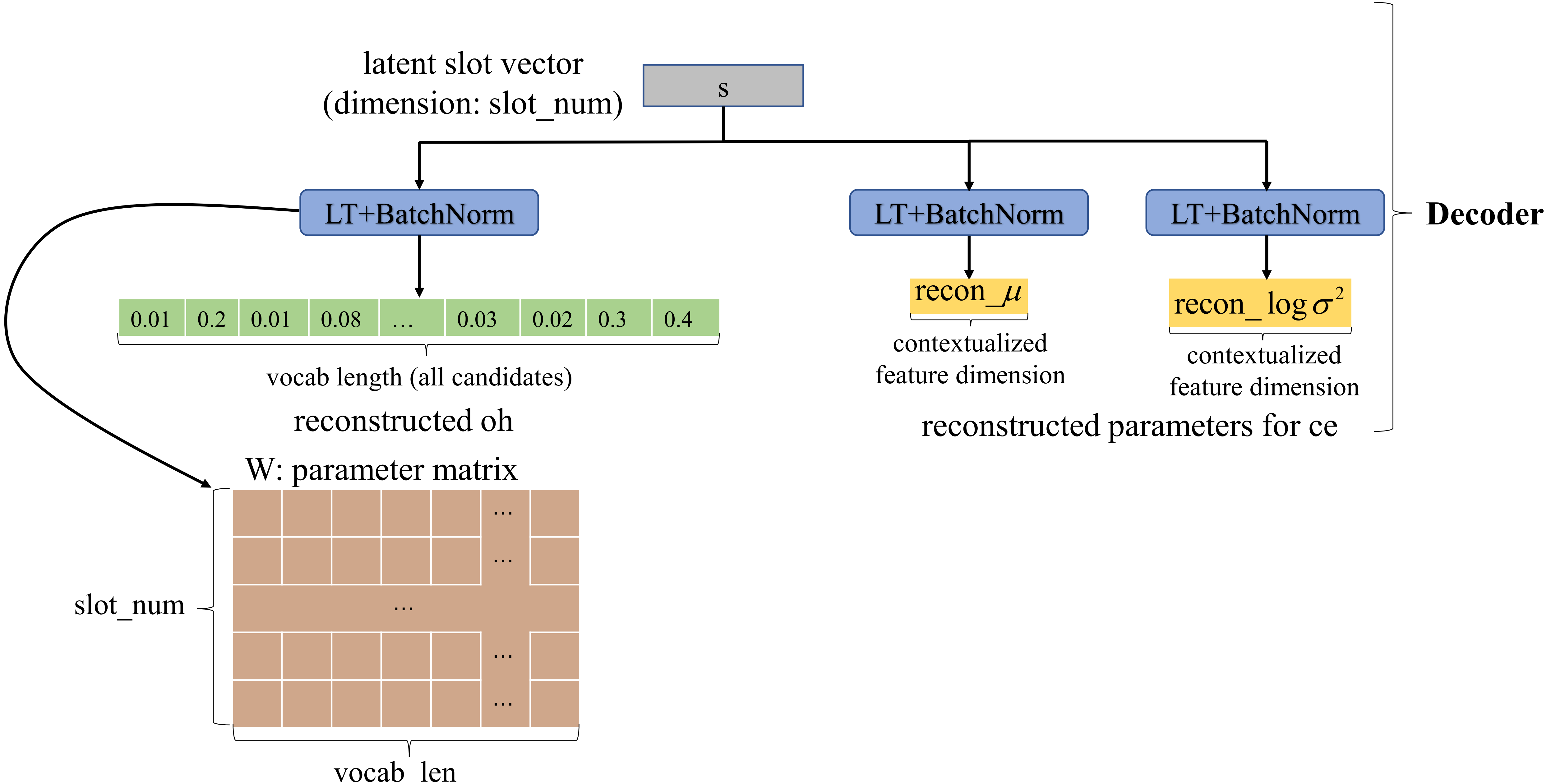


For each candidate in {train, Chicago, Dallas, Wednesday}

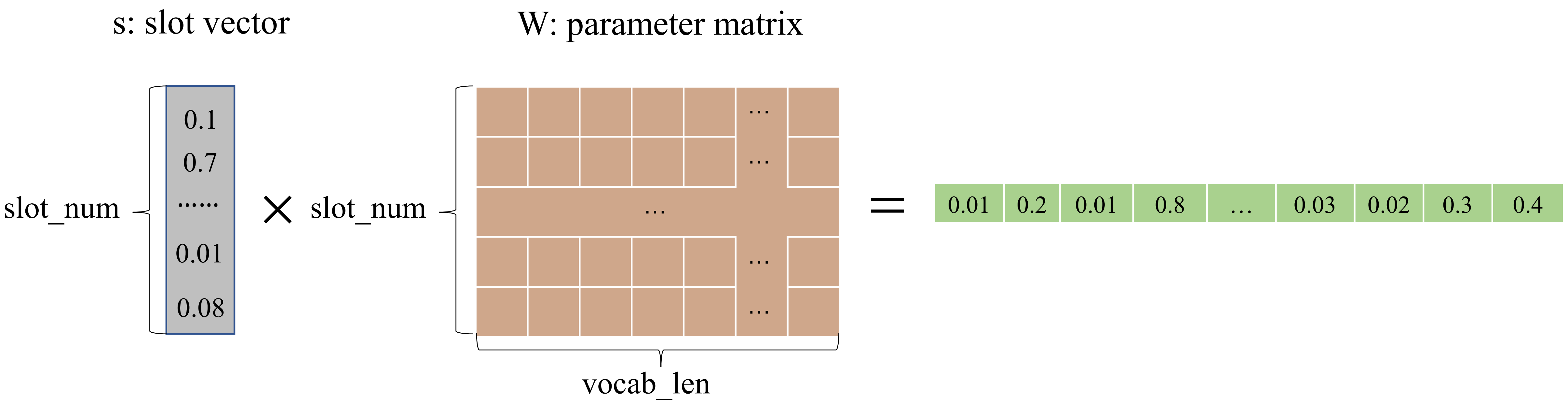
What does the model learn ?



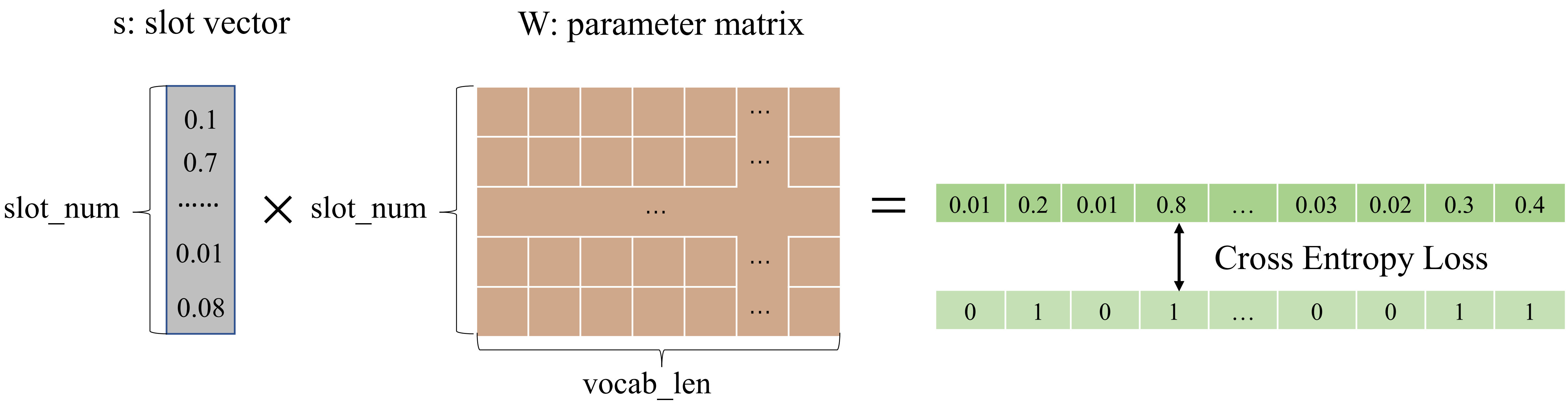
What does the model learn ?



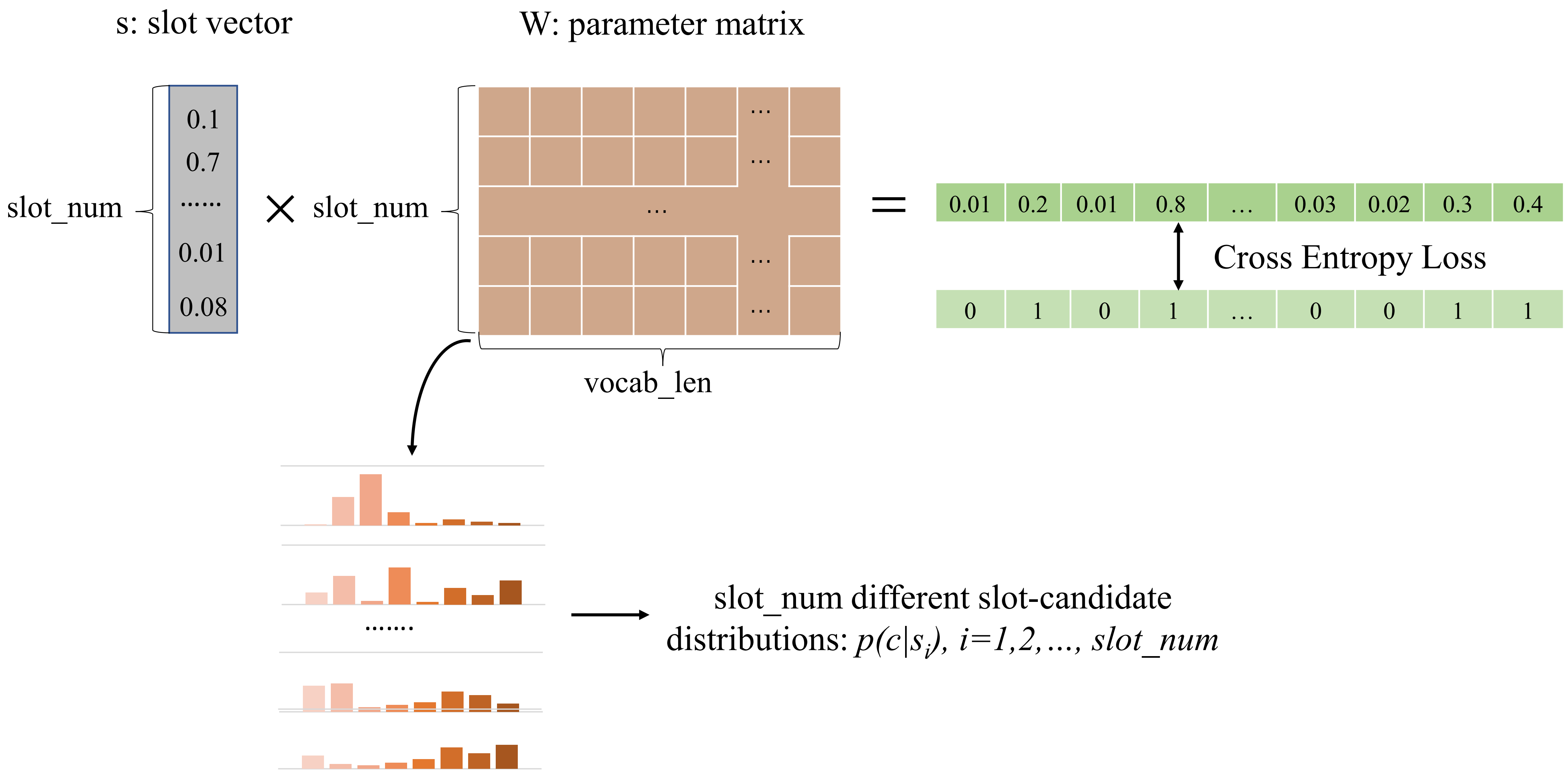
What does the model learn ?



What does the model learn ?

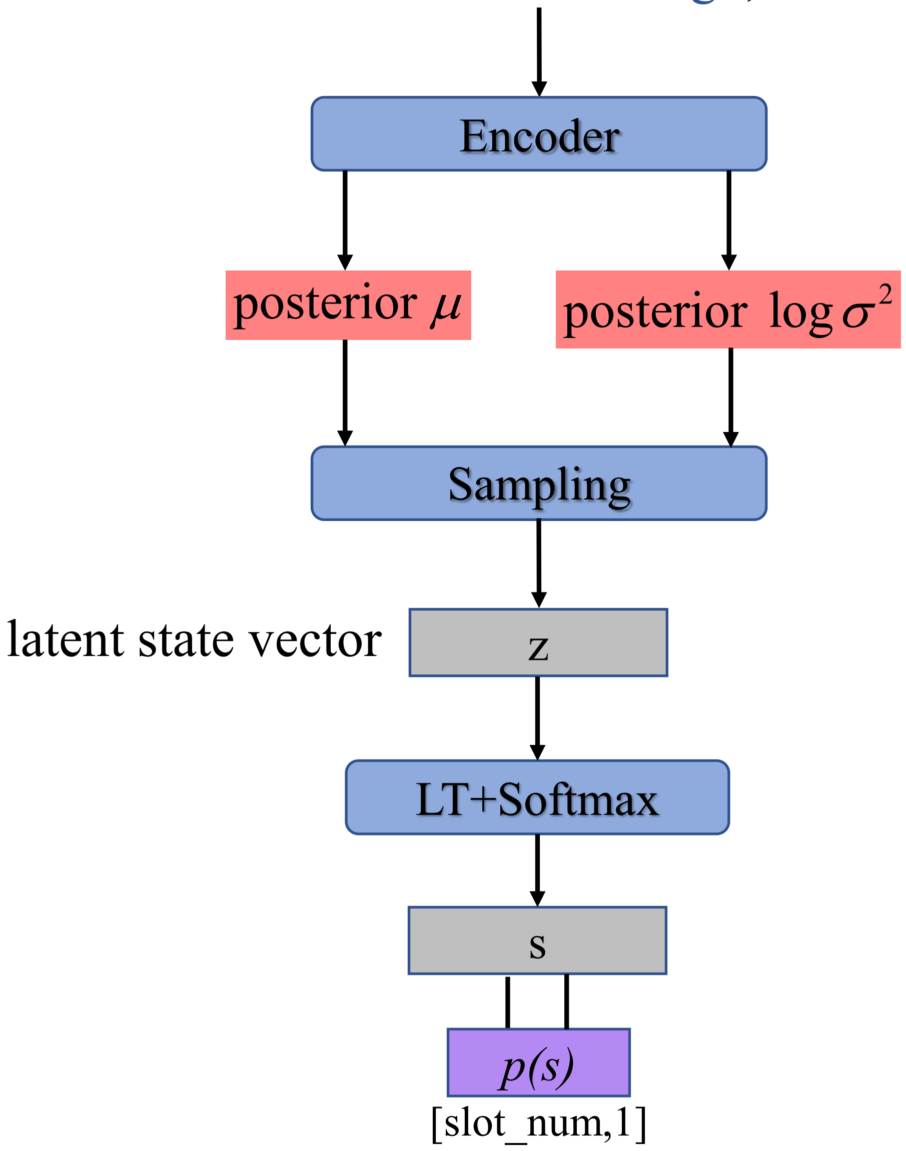


What does the model learn ?



DSI-base inference

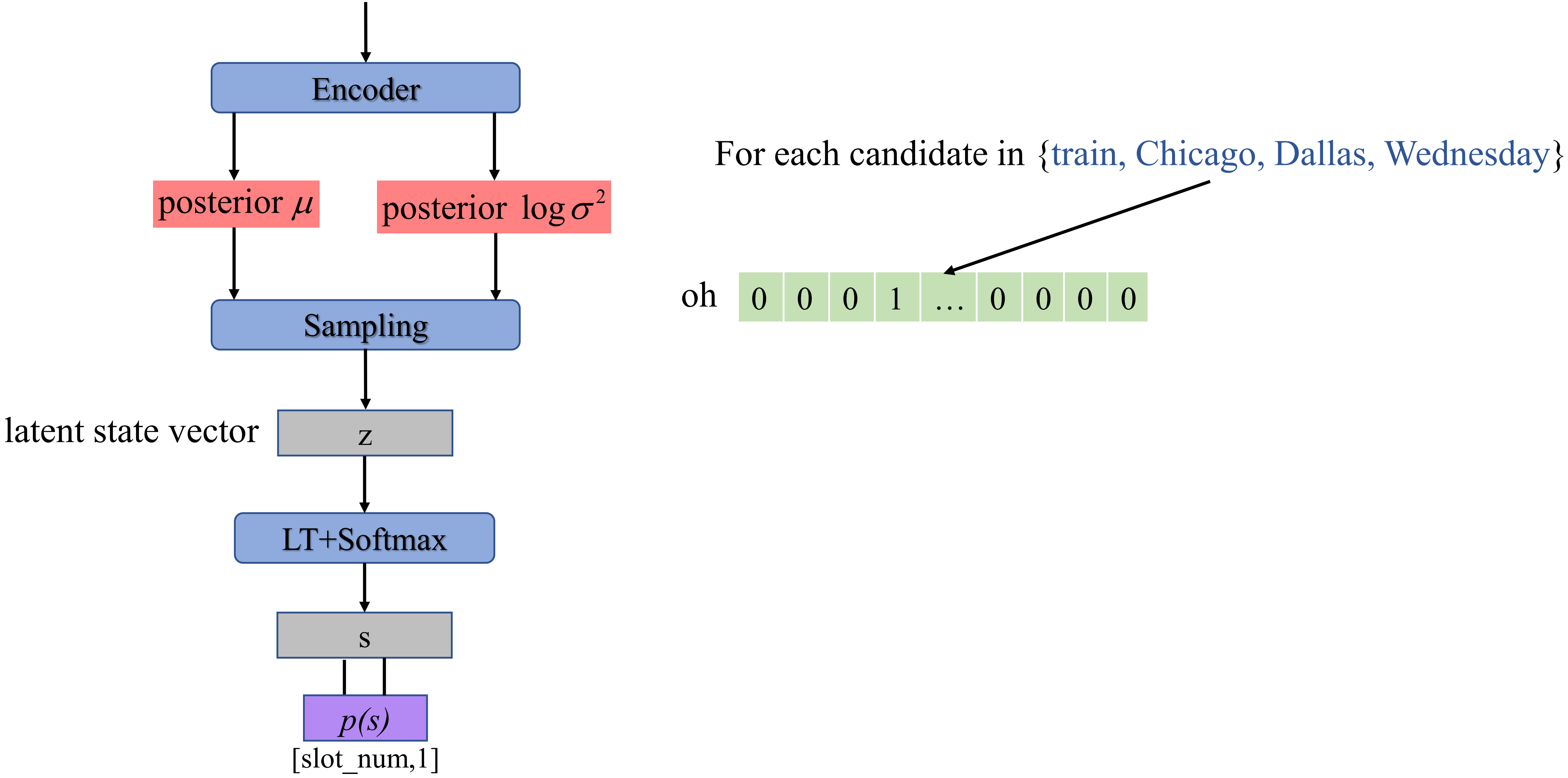
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



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

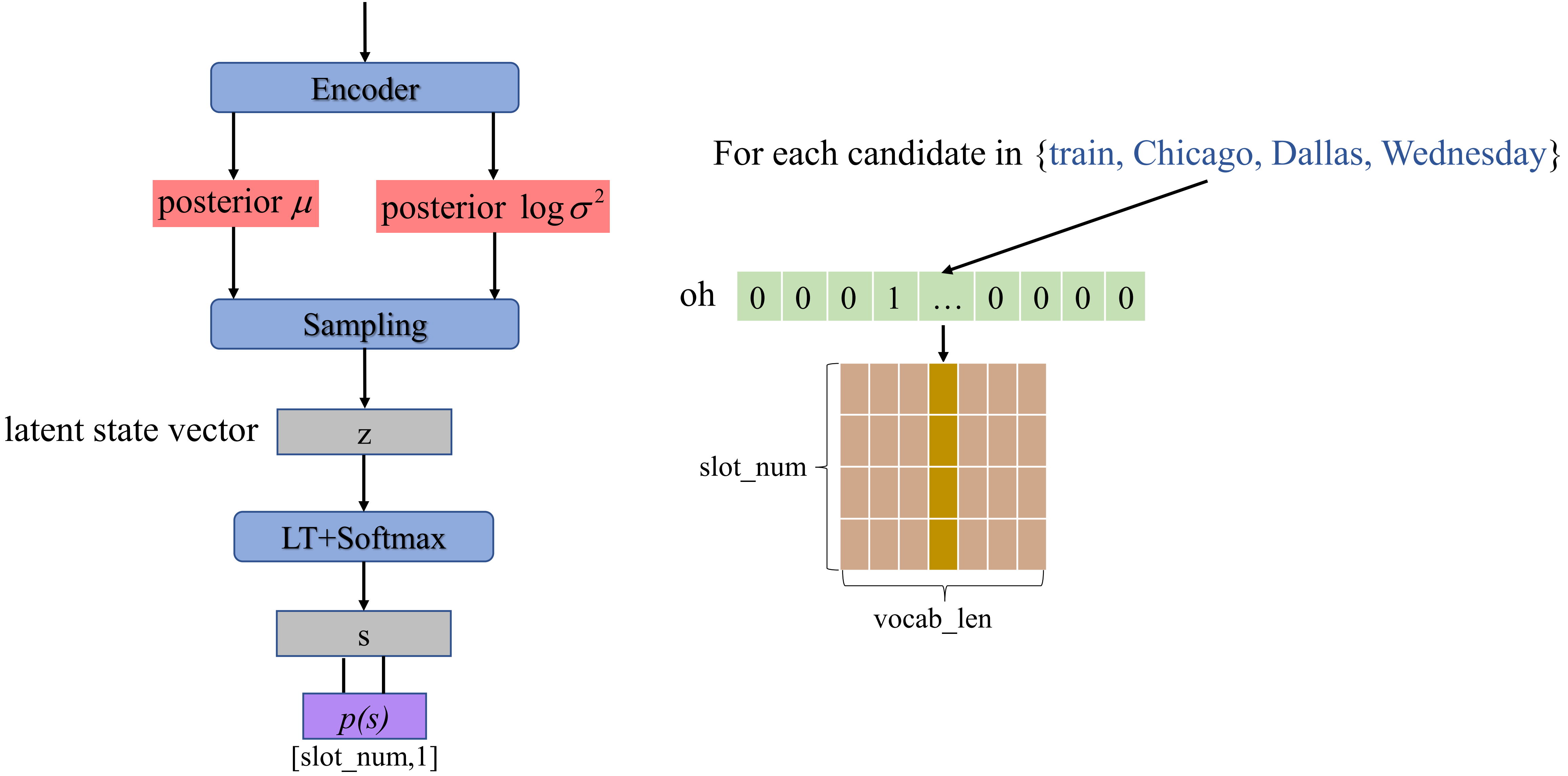
DSI-base inference

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



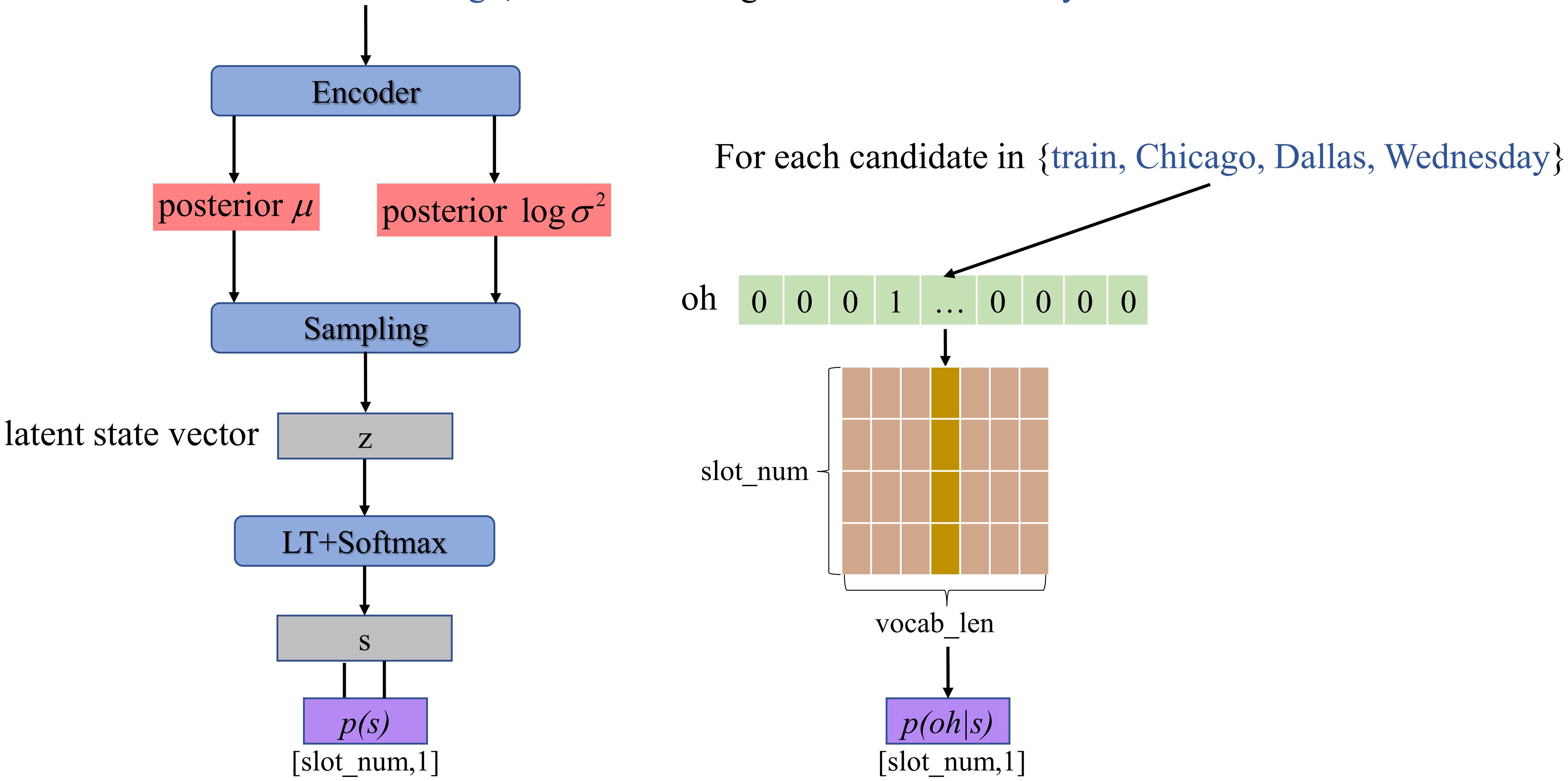
DSI-base inference

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

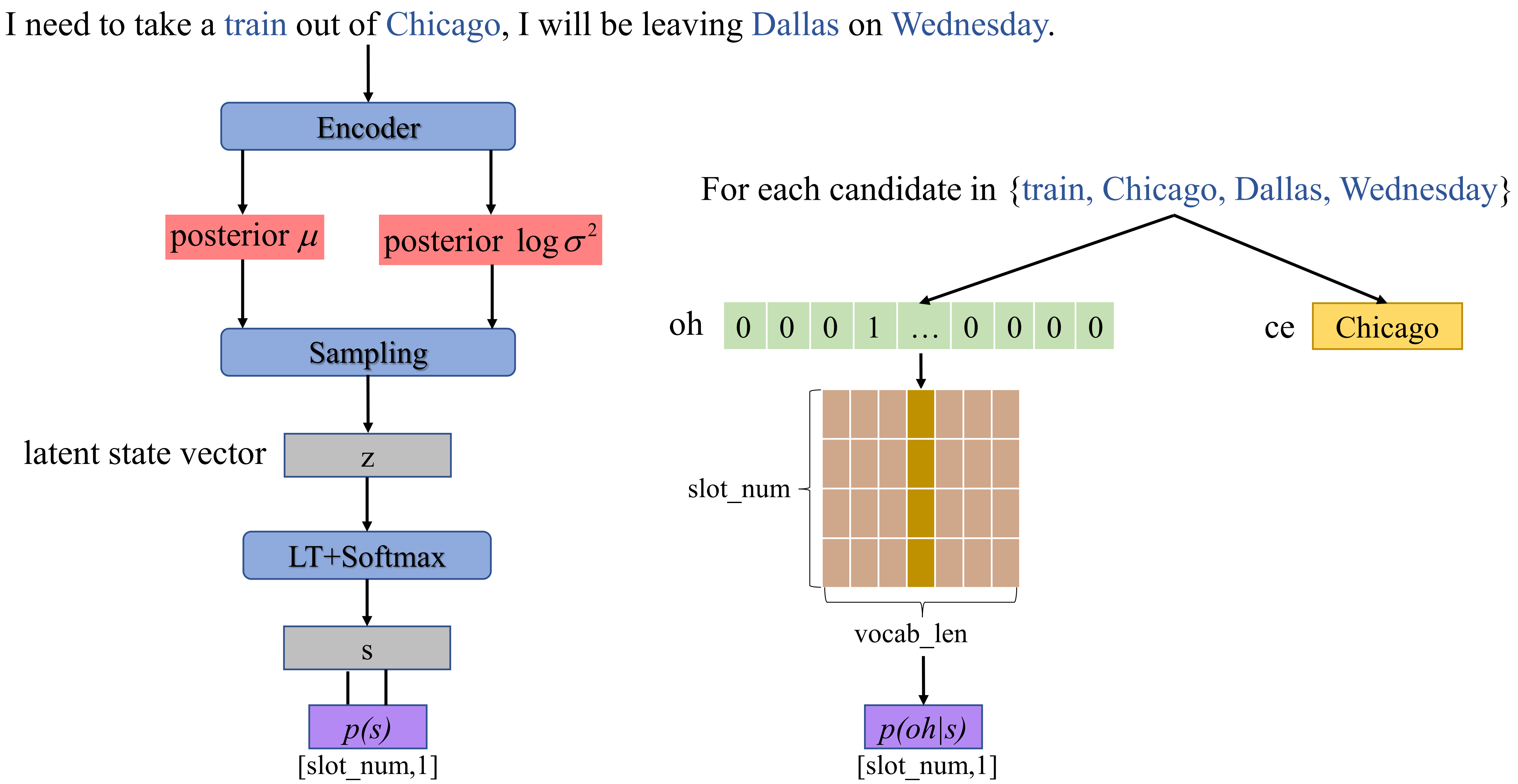


DSI-base inference

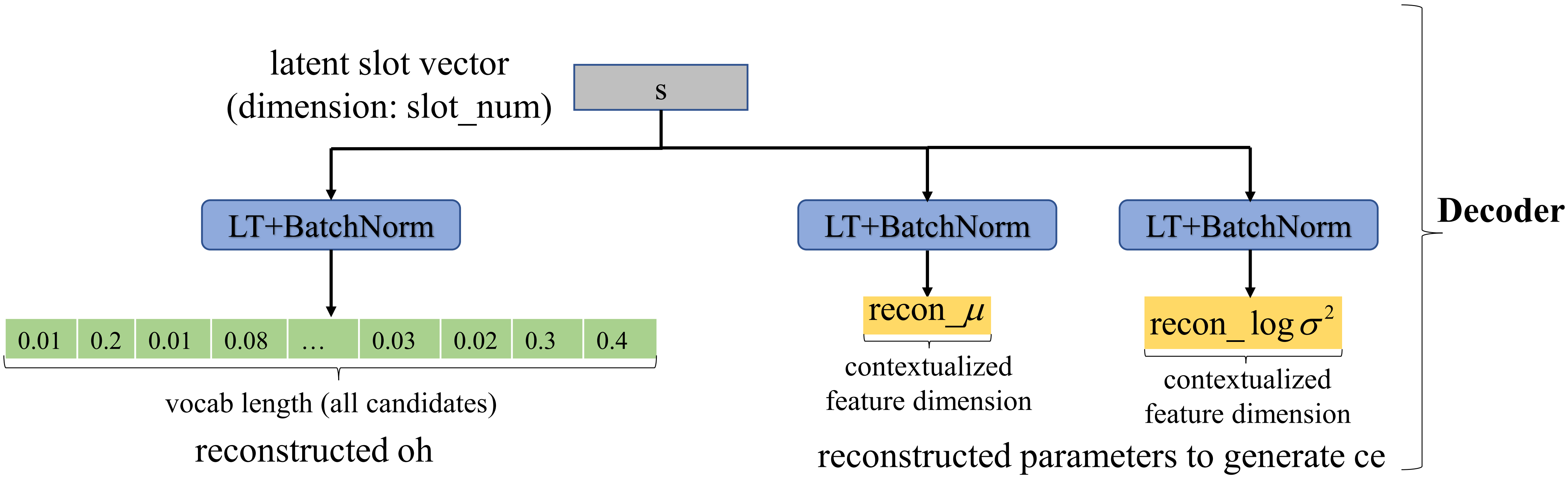
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



DSI-base inference

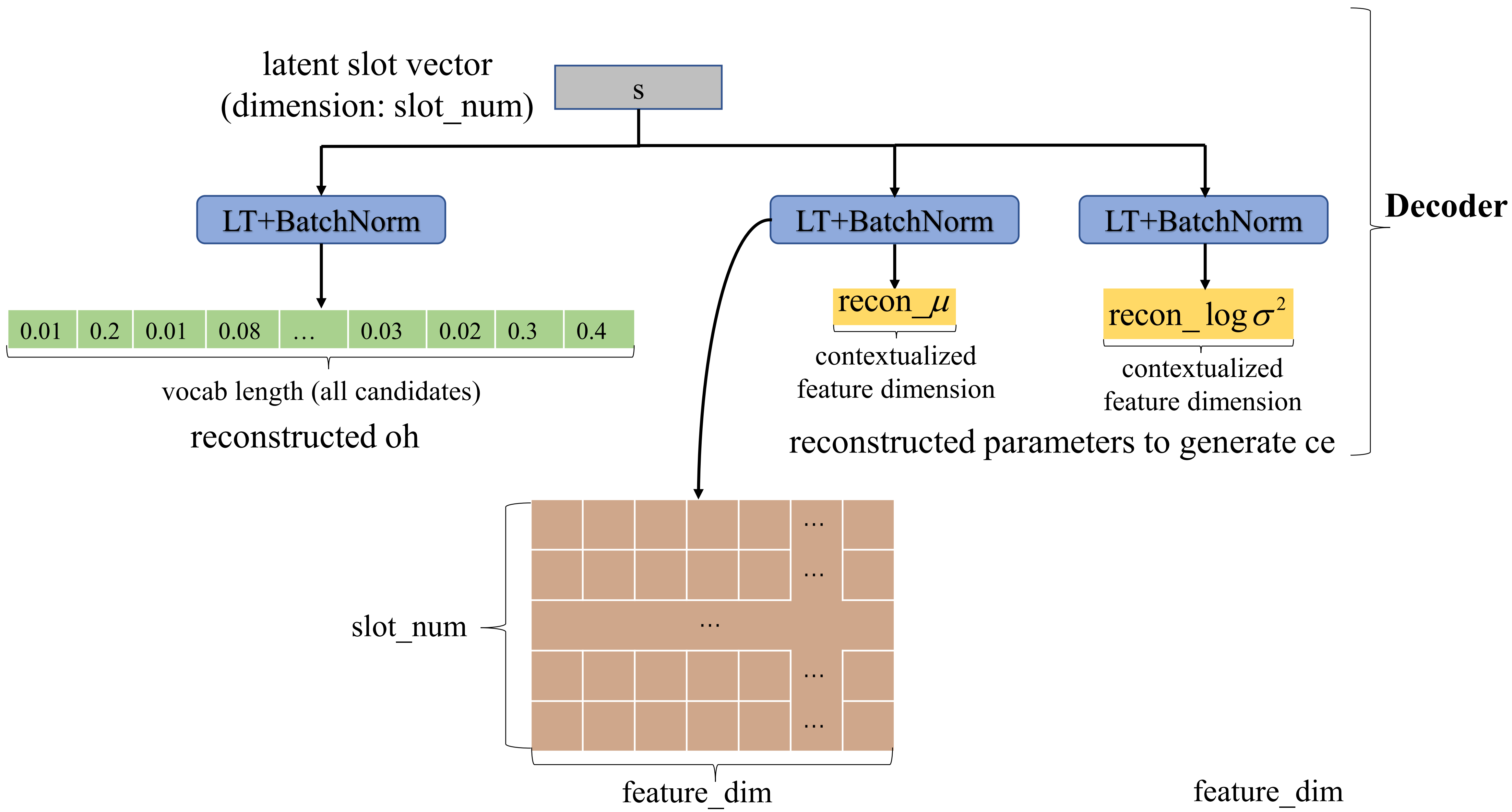


DSI-base inference

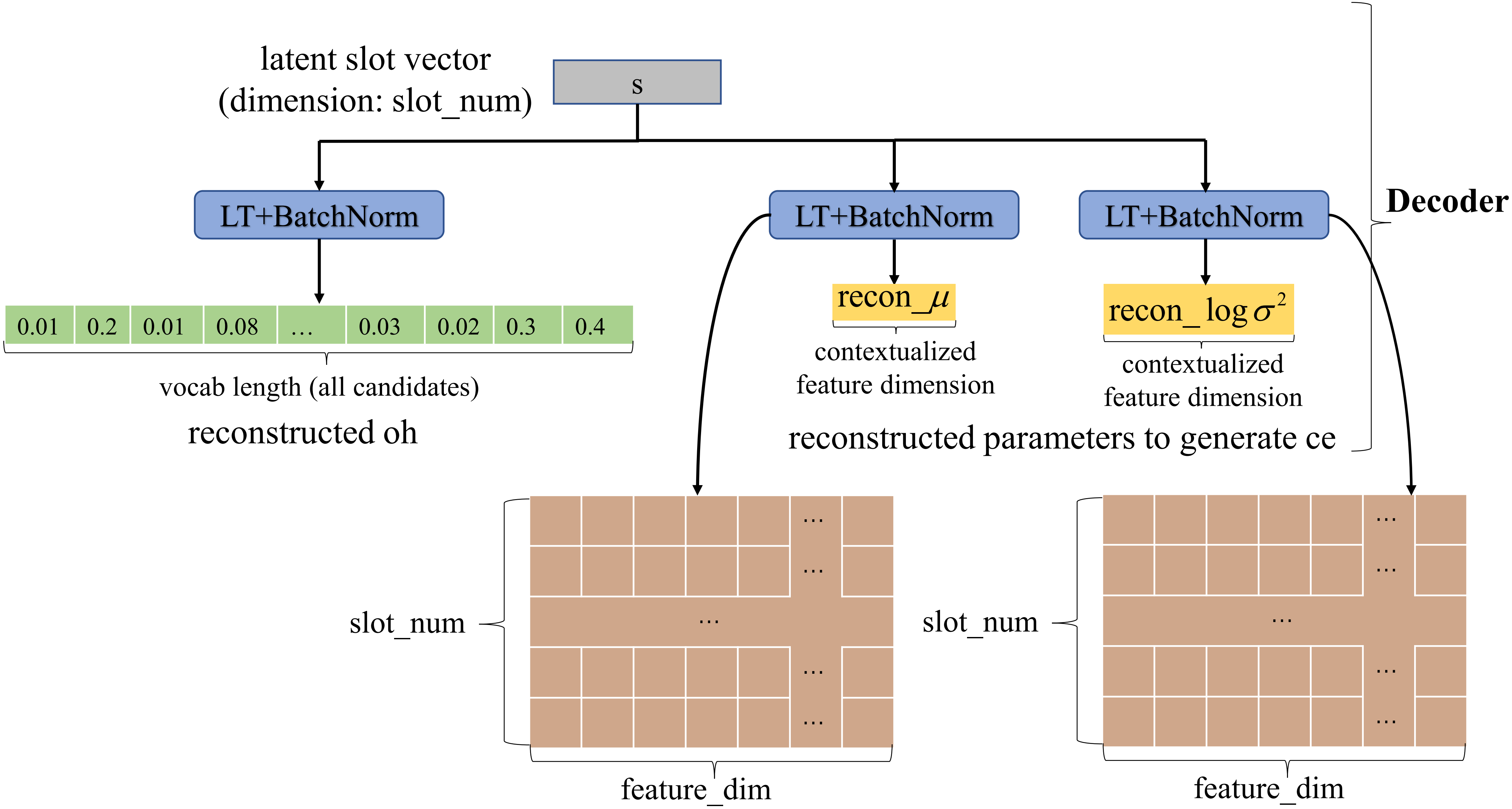


feature_dim

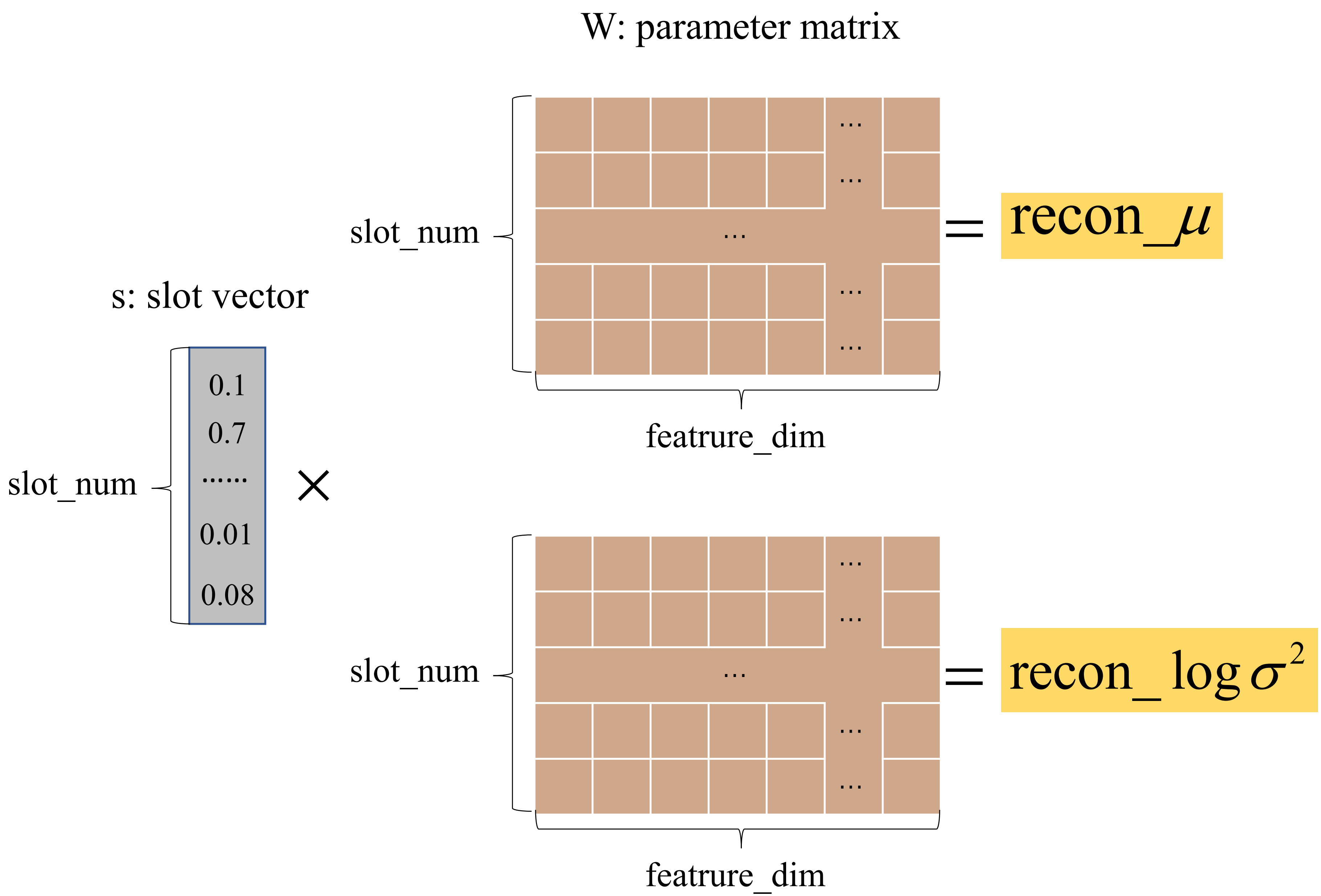
DSI-base inference



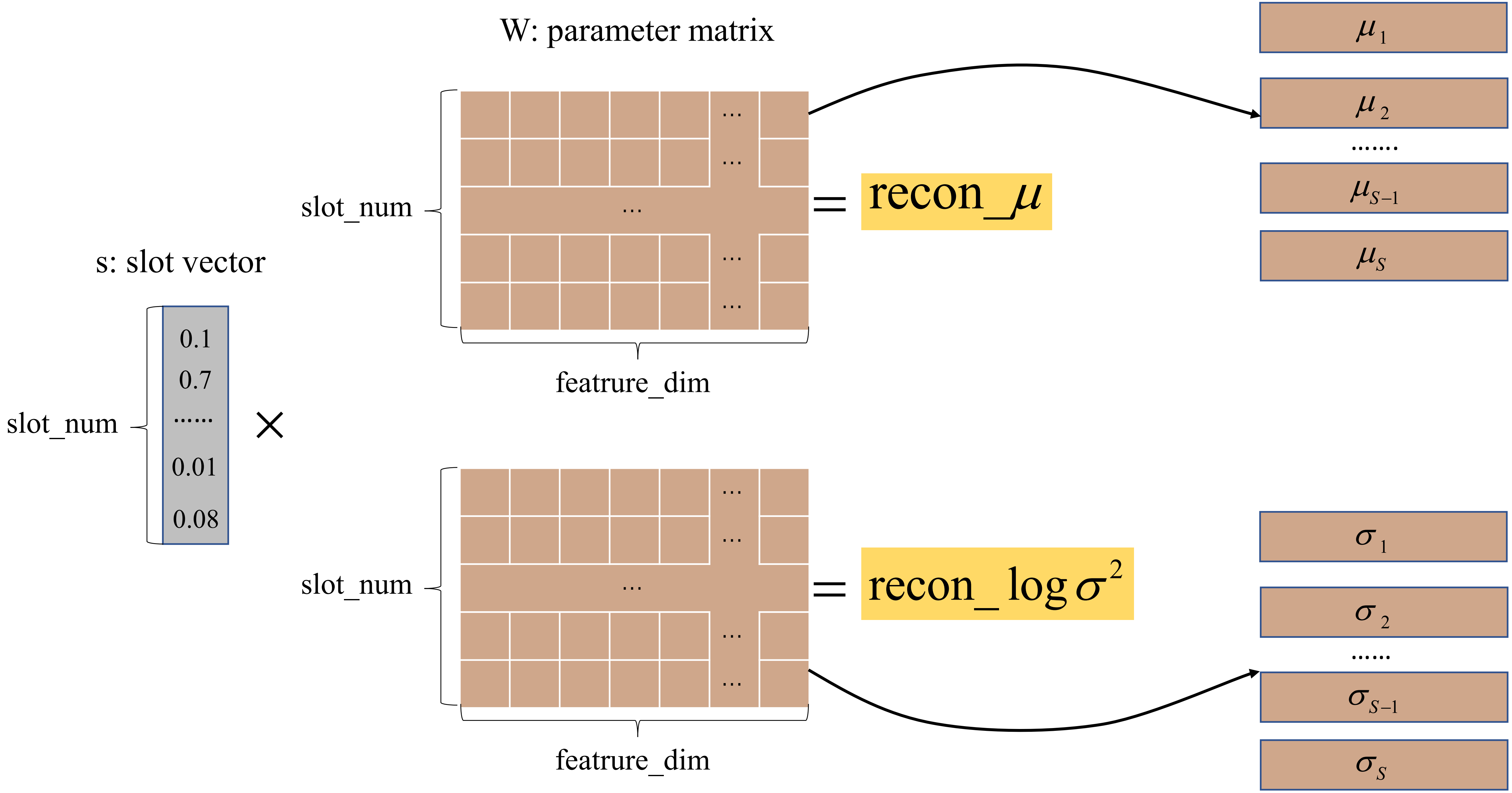
DSI-base inference



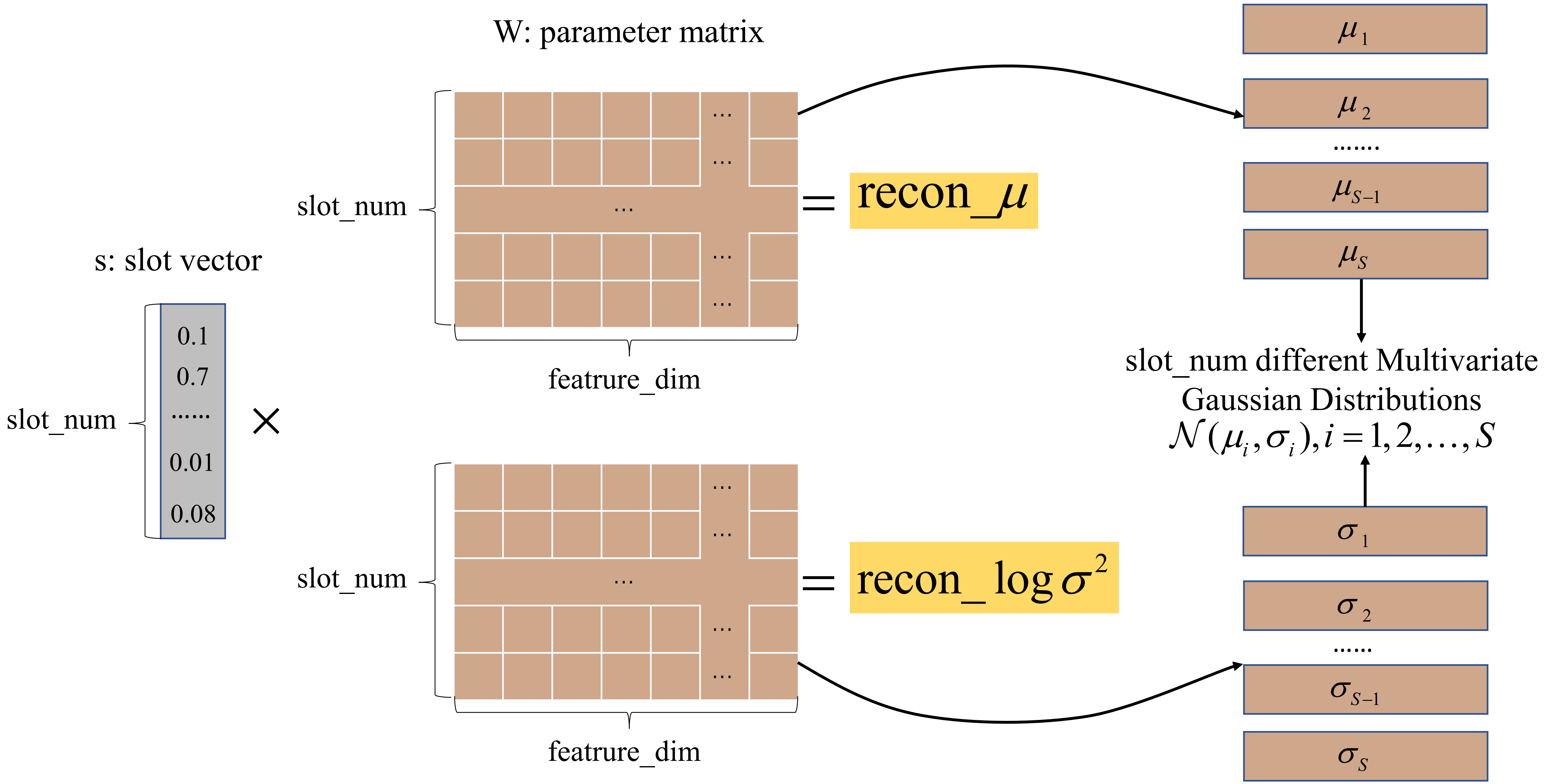
What does the model learn ?



What does the model learn ?



What does the model learn ?

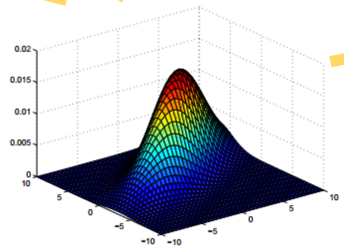


Multivariate Gaussian probability density

$\underbrace{\mu}_{\text{contextualized feature dimension}}$

$\underbrace{\sigma}_{\text{contextualized feature dimension}}$

train Chicago Dallas Wednesday



Multivariate Gaussian Distribution

-dist.log_prob(ce[train])

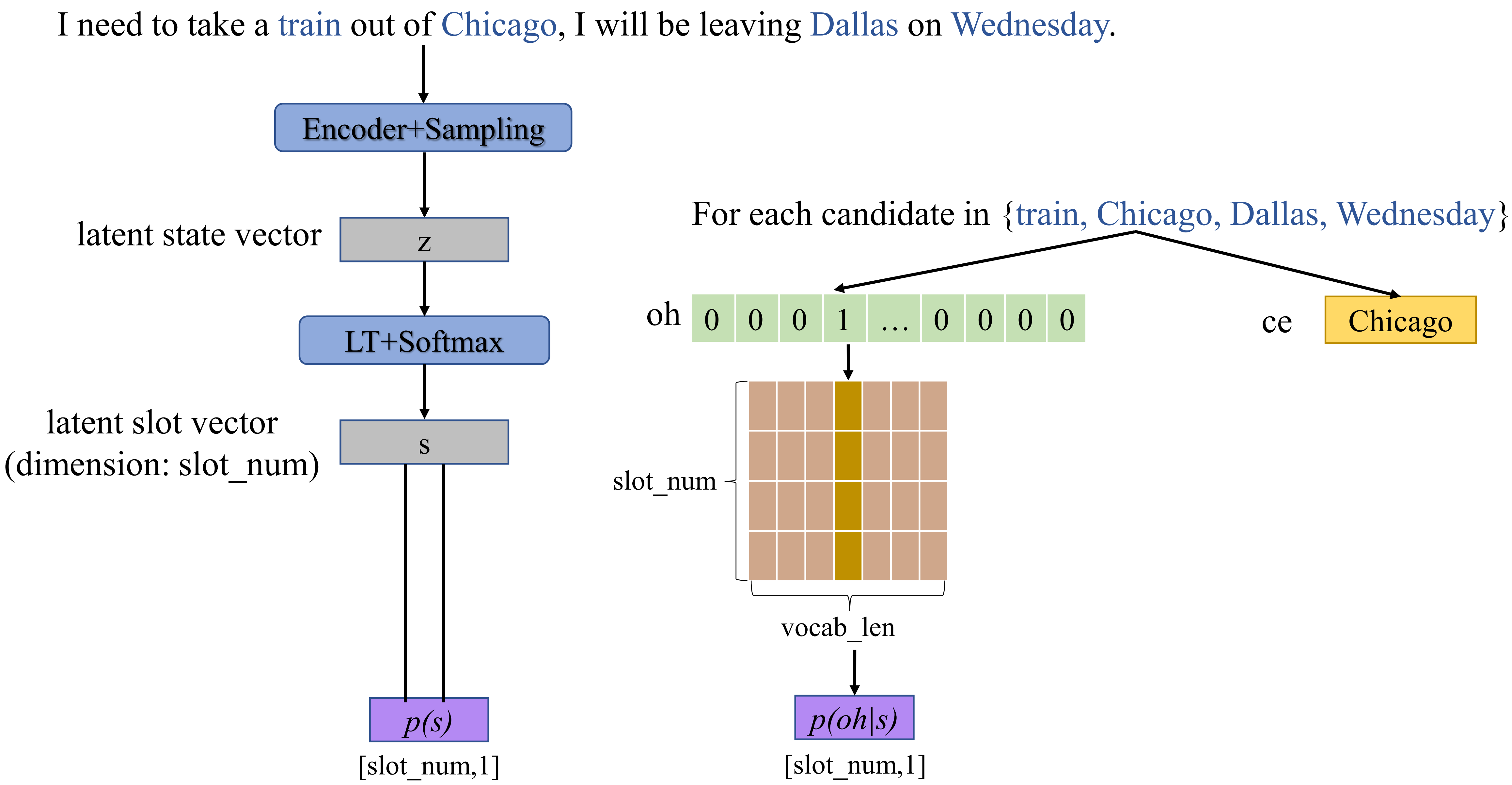
-dist.log_prob(ce[Chicago])

-dist.log_prob(ce[Dallas])

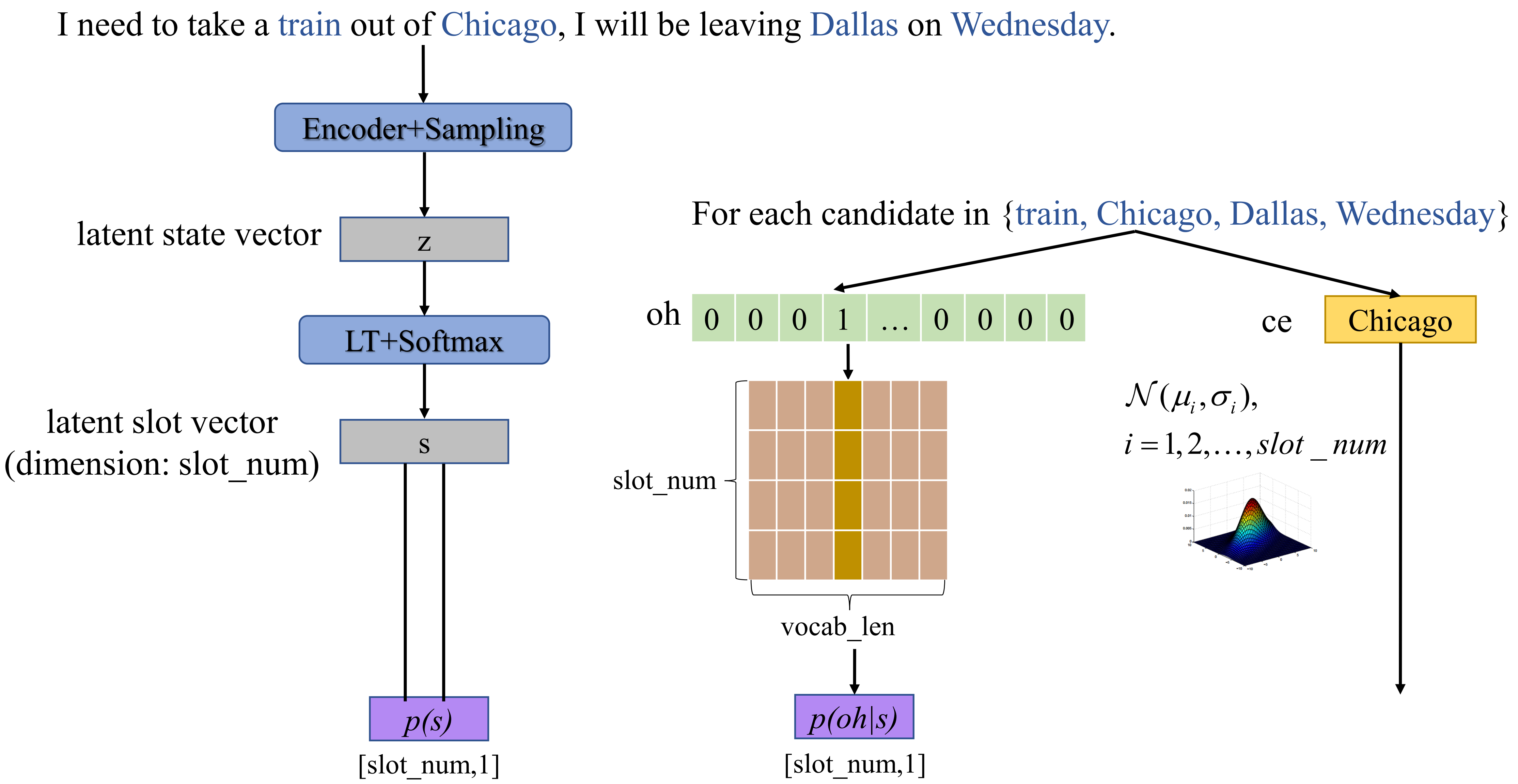
-dist.log_prob(ce[Wednesday])

Pic from https://www.datalearner.com/blog/1051485590815771

DSI-base inference



DSI-base inference



DSI-base inference

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

Encoder+Sampling

latent state vector: z

LT+Softmax

latent slot vector (dimension: slot_num): s

$p(s)$

[slot_num,1]

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

oh: 0 0 0 1 … 0 0 0 0

slot_num

vocab_len

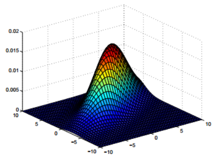
$p(oh|s)$

[slot_num,1]

ce: Chicago

$\mathcal{N}(\mu_i, \sigma_i)$,

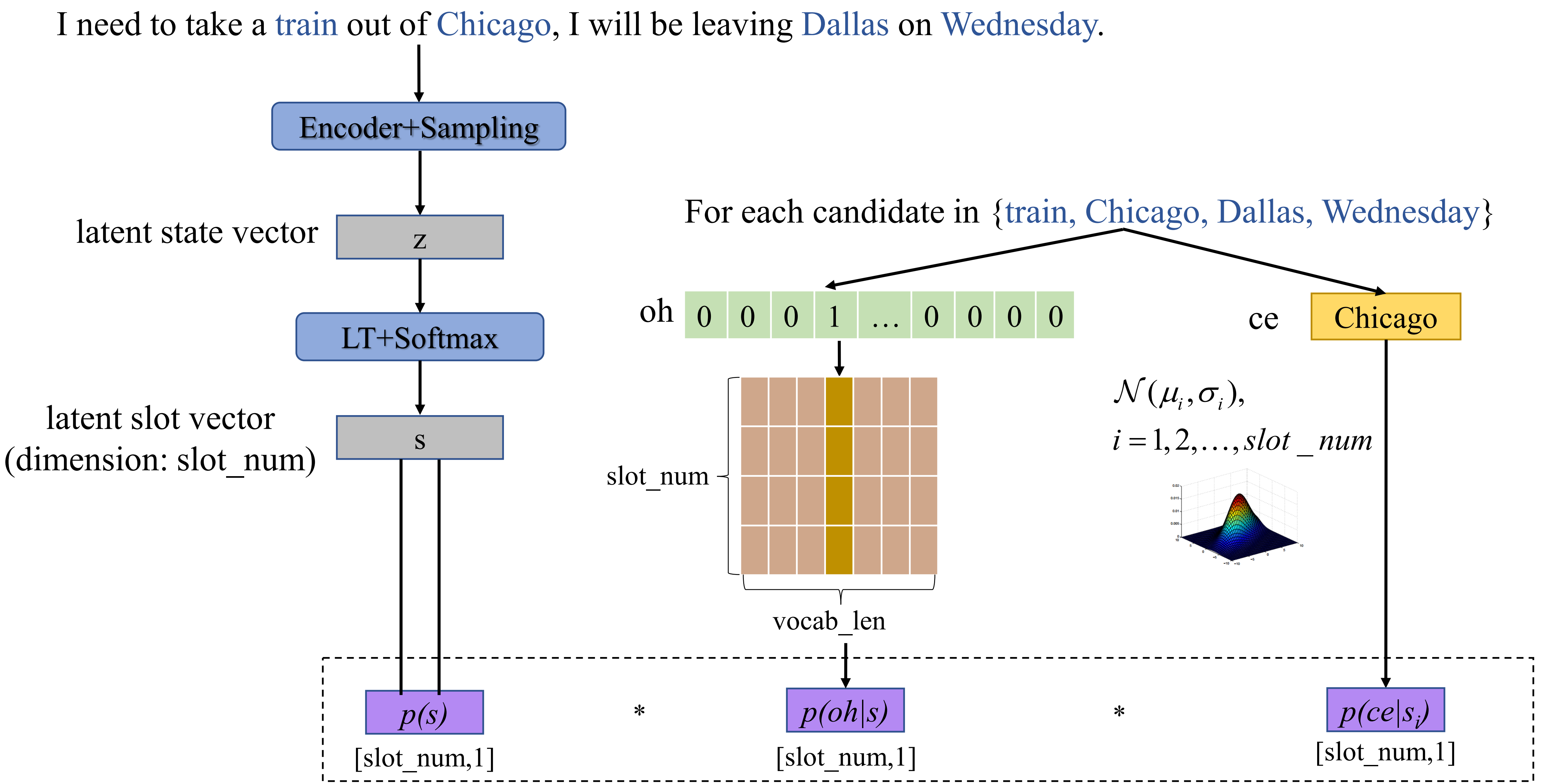
$i = 1, 2, \ldots, slot\_num$



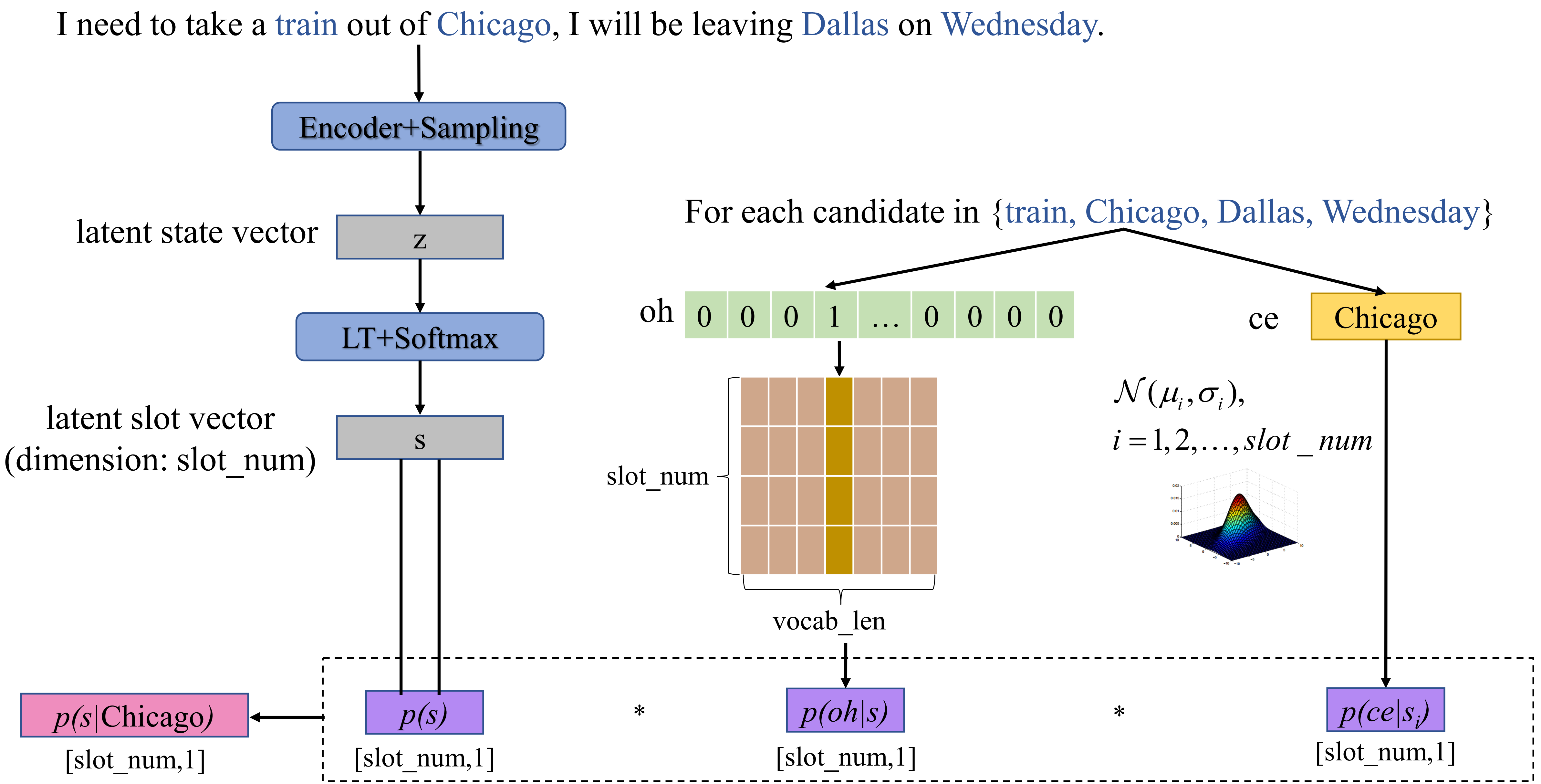
$p(ce|s)$

[slot_num,1]

DSI-base inference

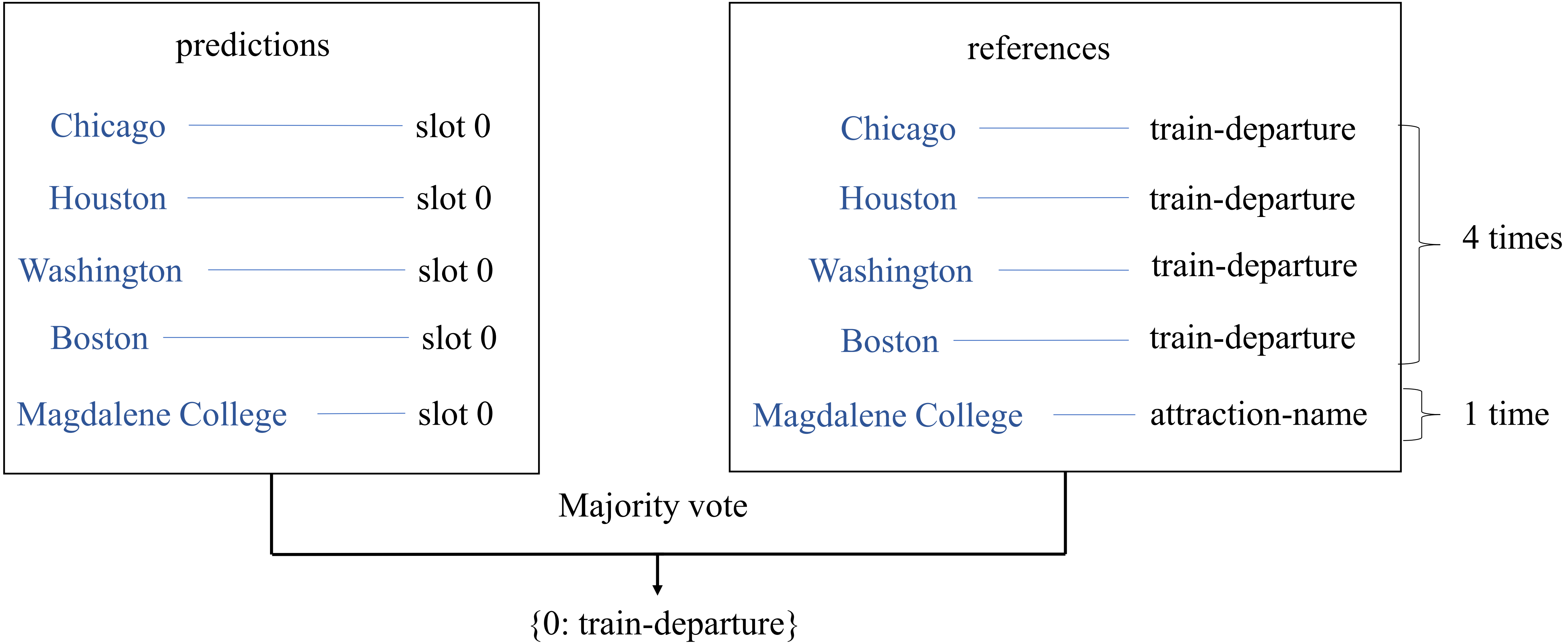


DSI-base inference



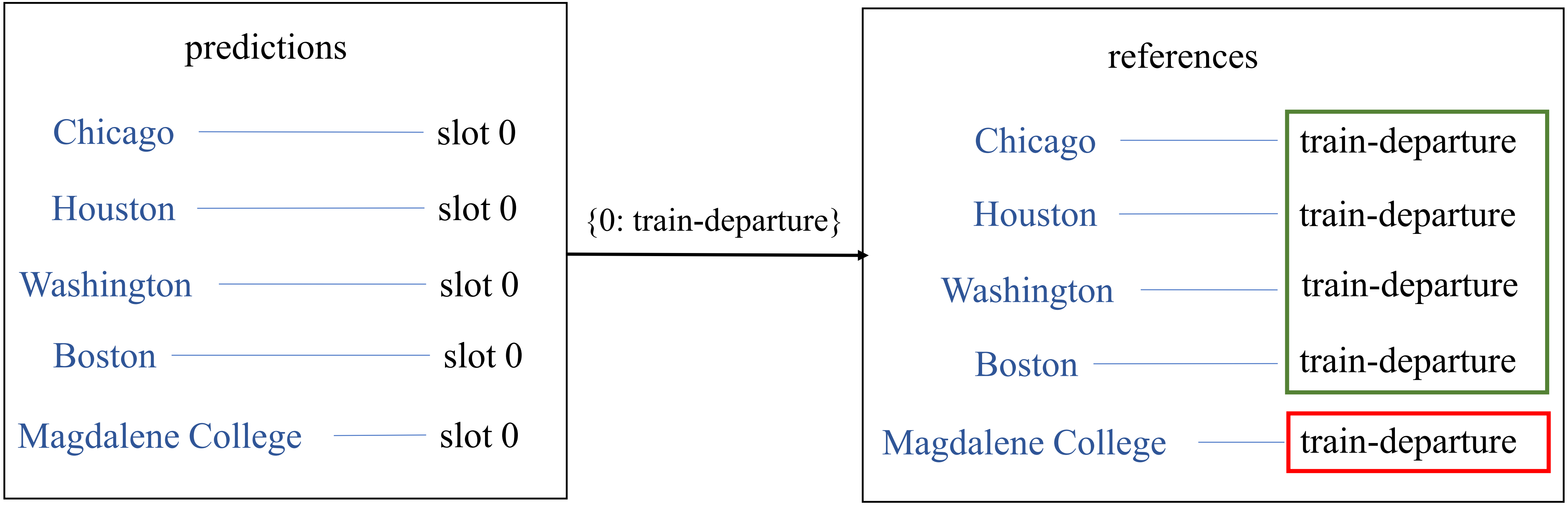
Post-processing: slot mapping

Mapping from slot indexes to labels?

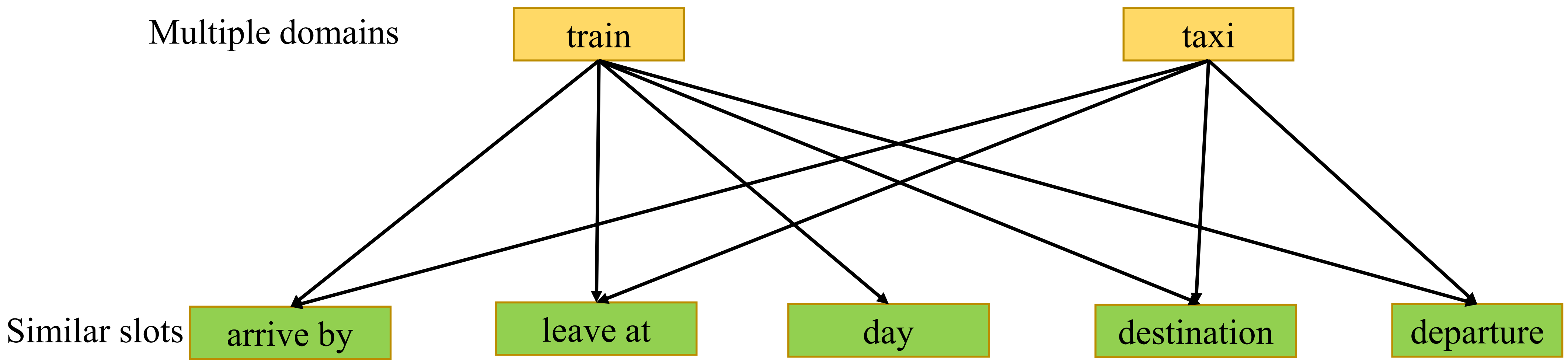


Post-processing: slot mapping

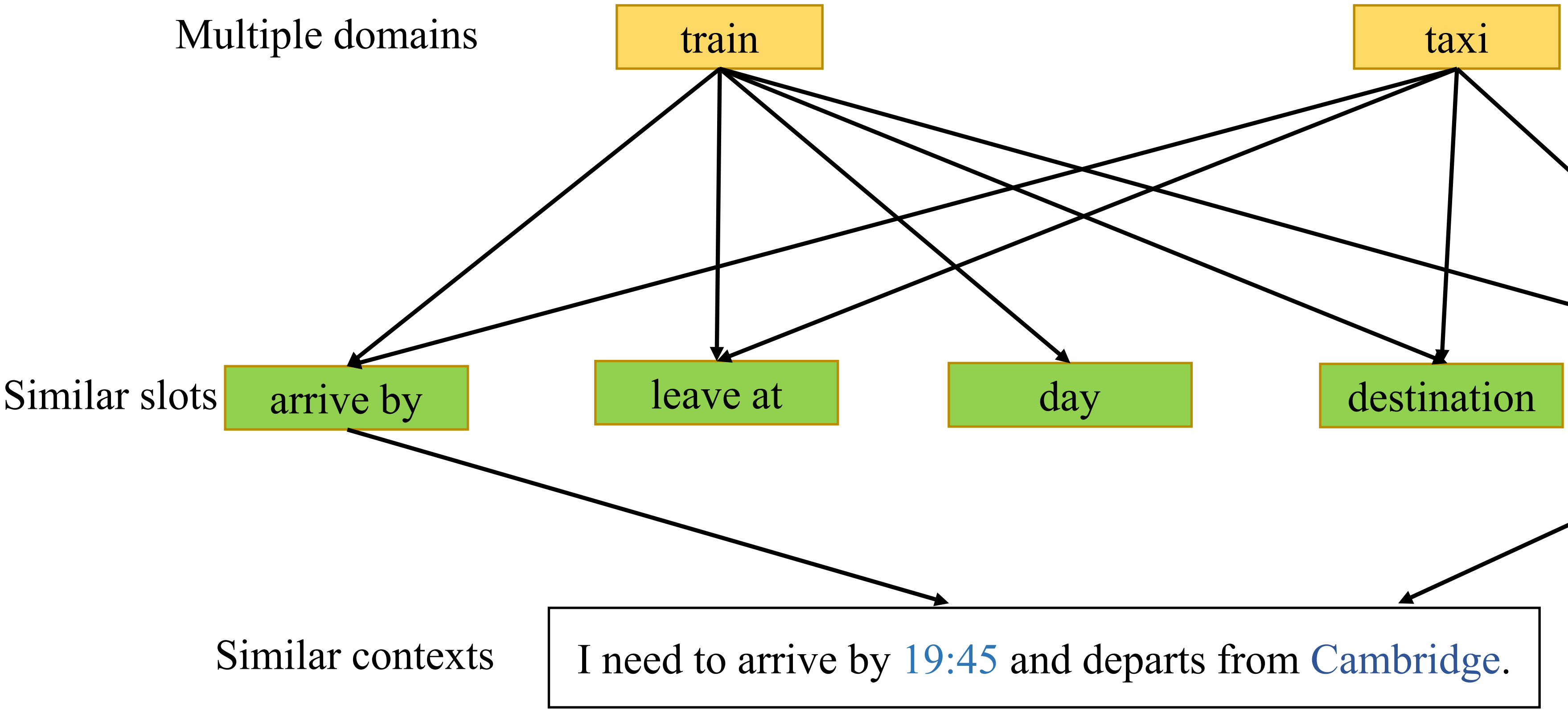
Mapping from slot indexes to labels?



What is the problem with *DSI-base*?

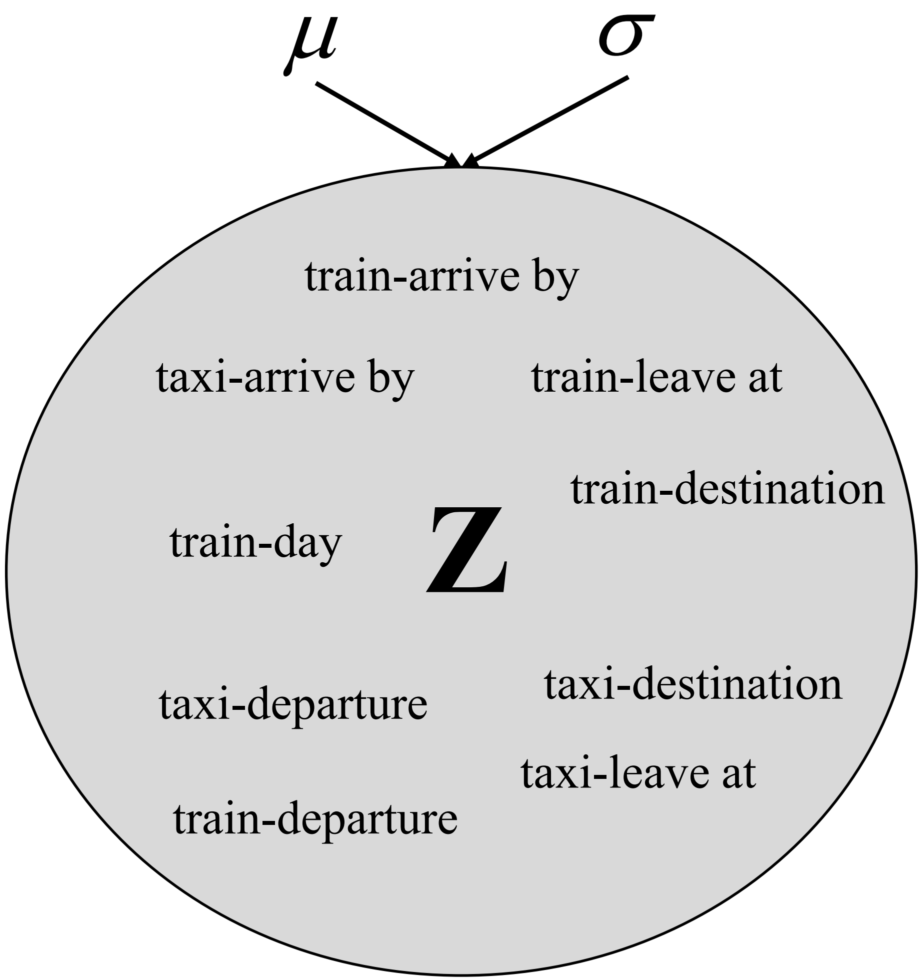


What is the problem with *DSI-base*?

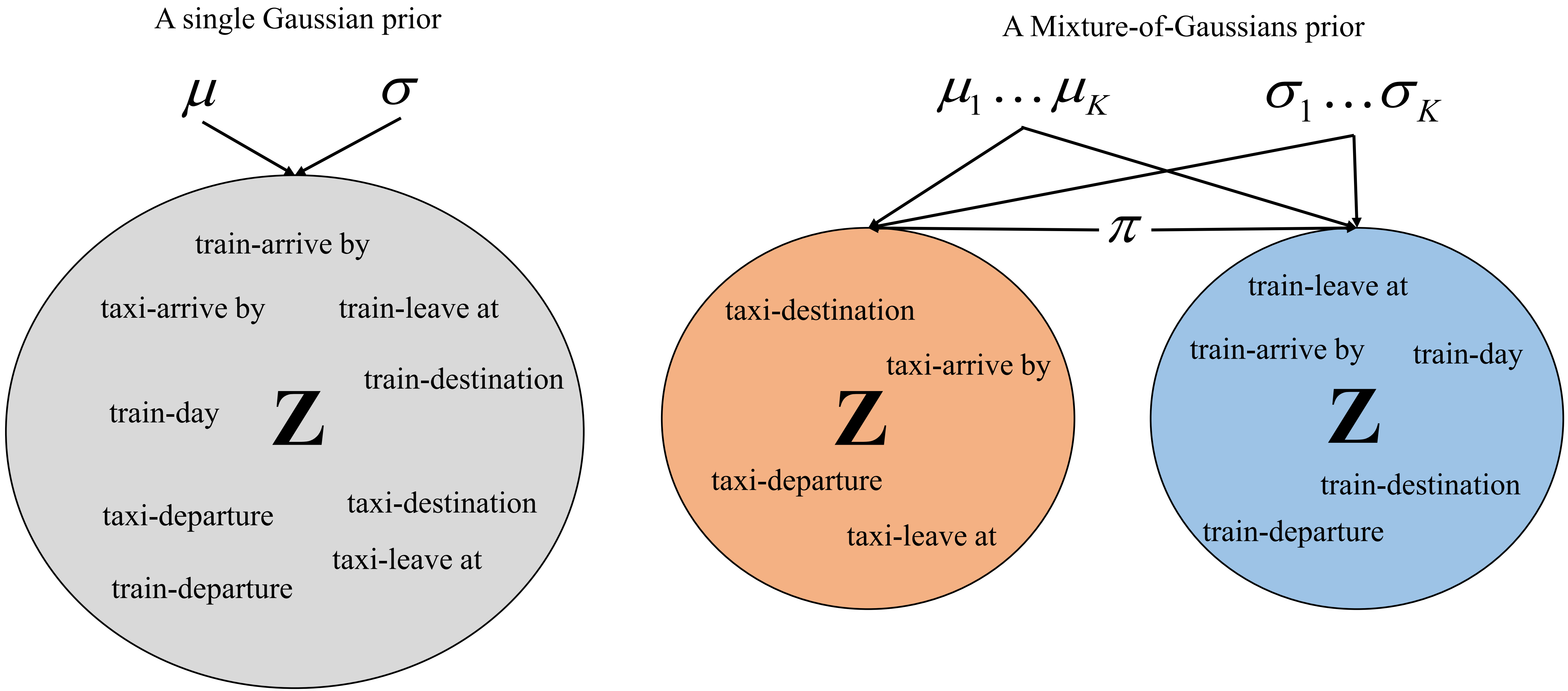


How does *DSI-GM* work?

A single Gaussian prior



How does *DSI-GM* work?



How does *DSI-GM* work?

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

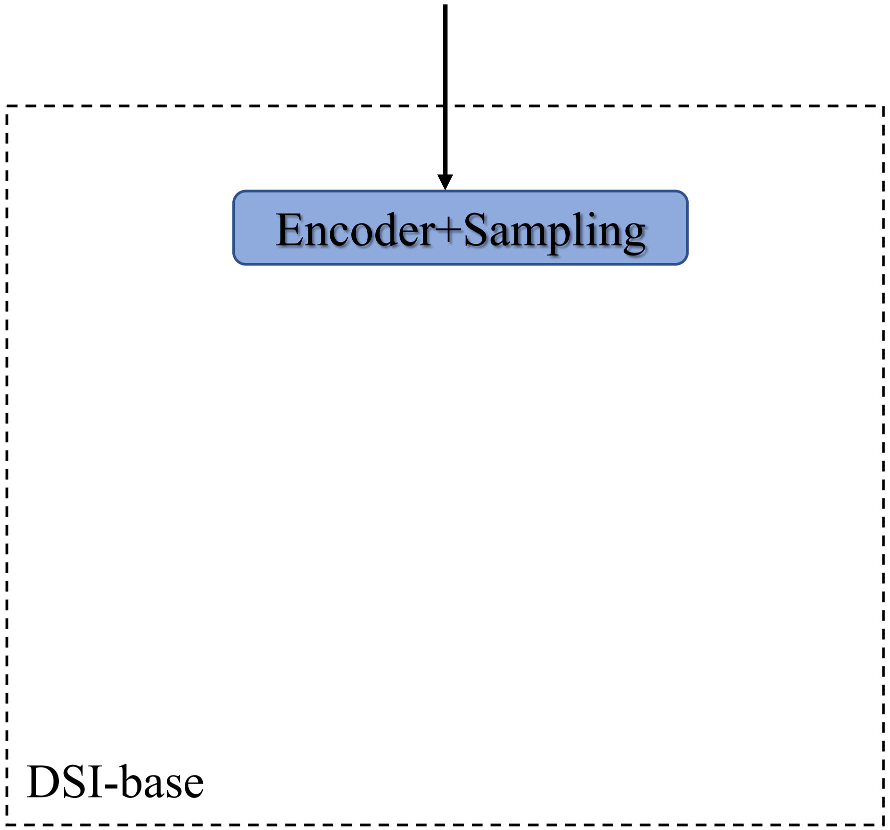
How does *DSI-GM* work?

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

DSI-base

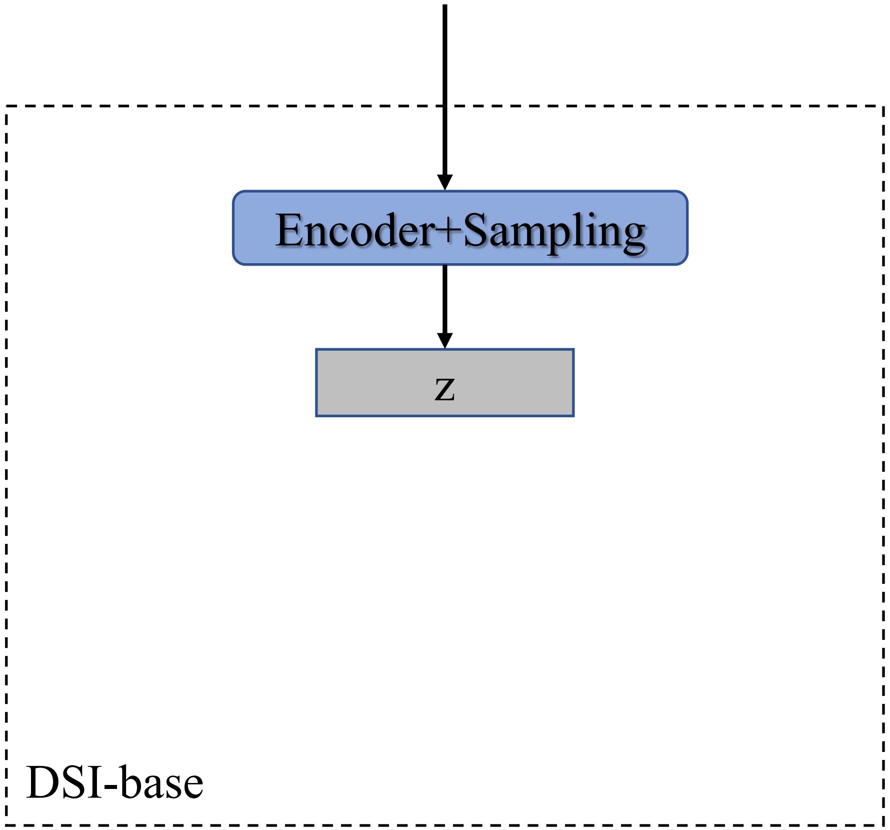
How does *DSI-GM* work?

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



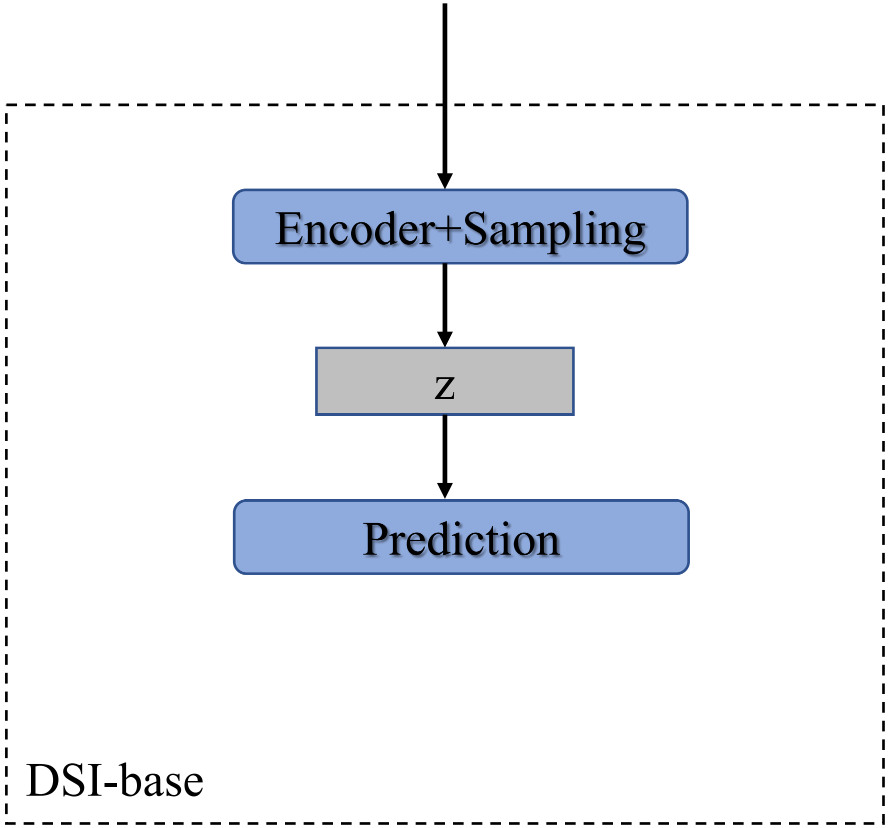
How does *DSI-GM* work?

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



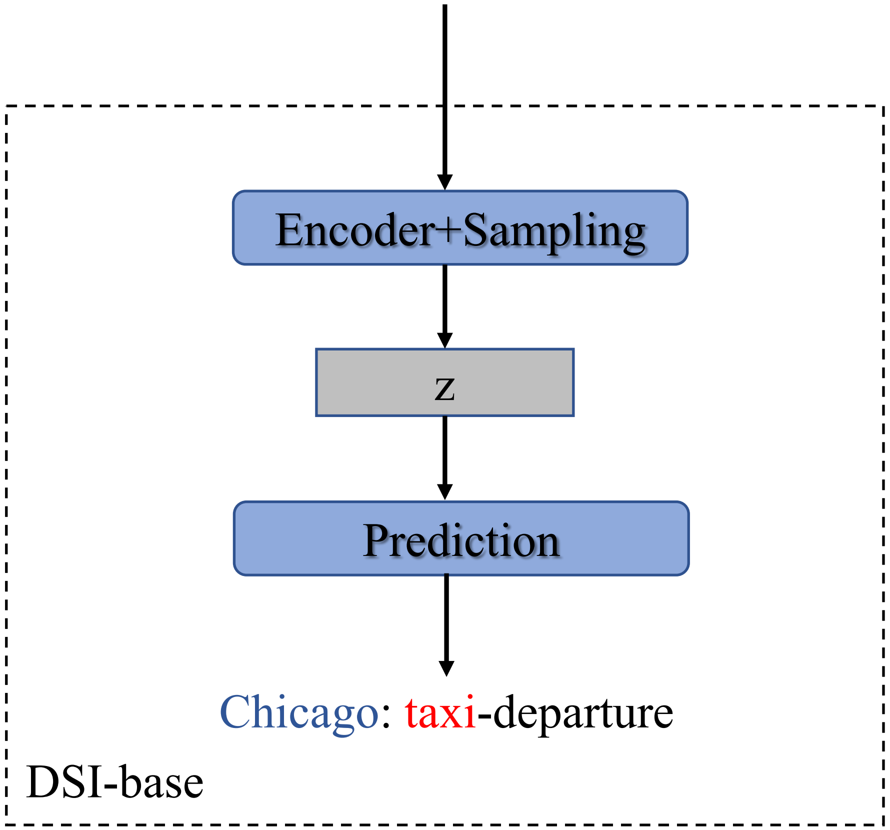
How does *DSI-GM* work?

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



How does *DSI-GM* work?

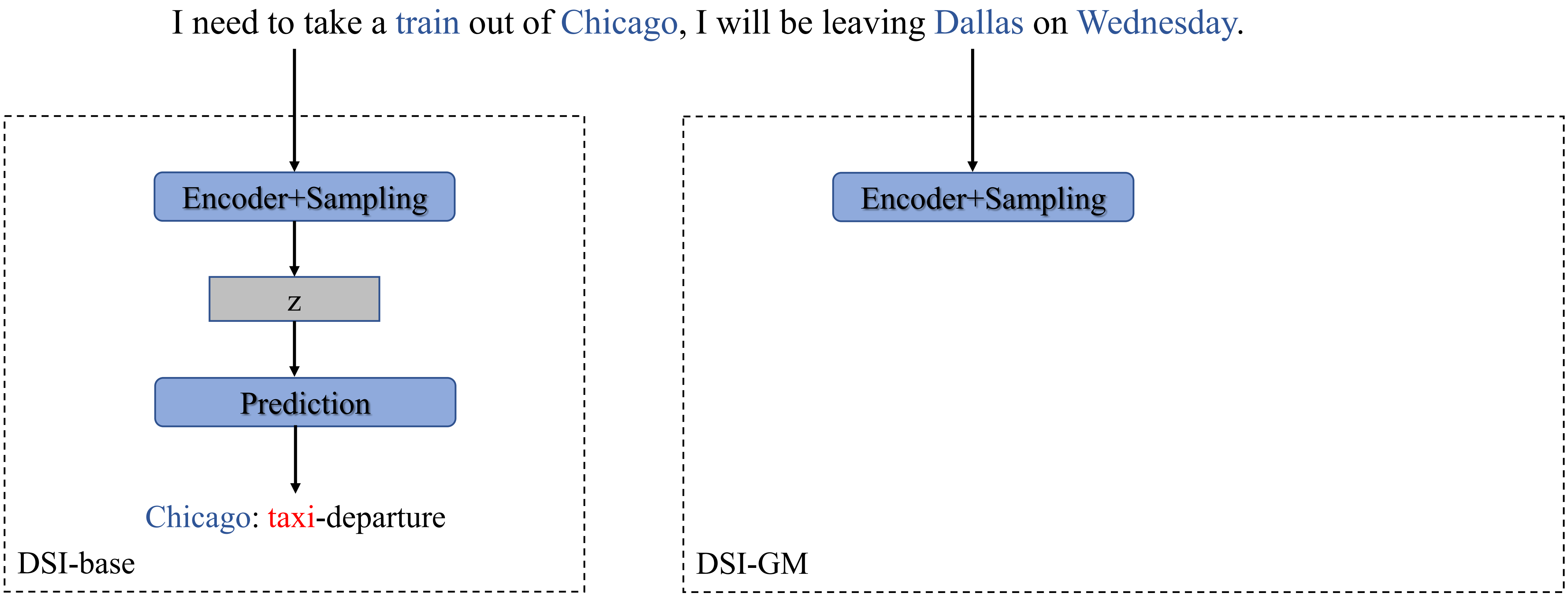
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



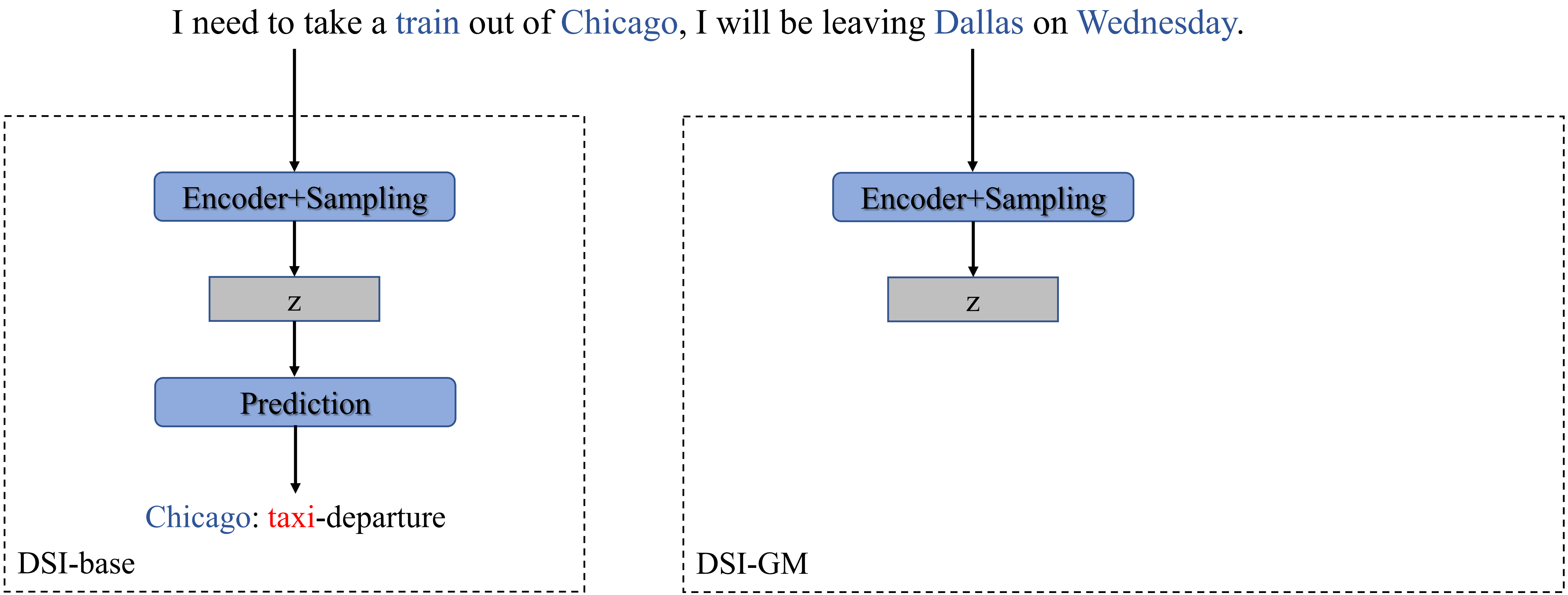
How does *DSI-GM* work?

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

How does *DSI-GM* work?

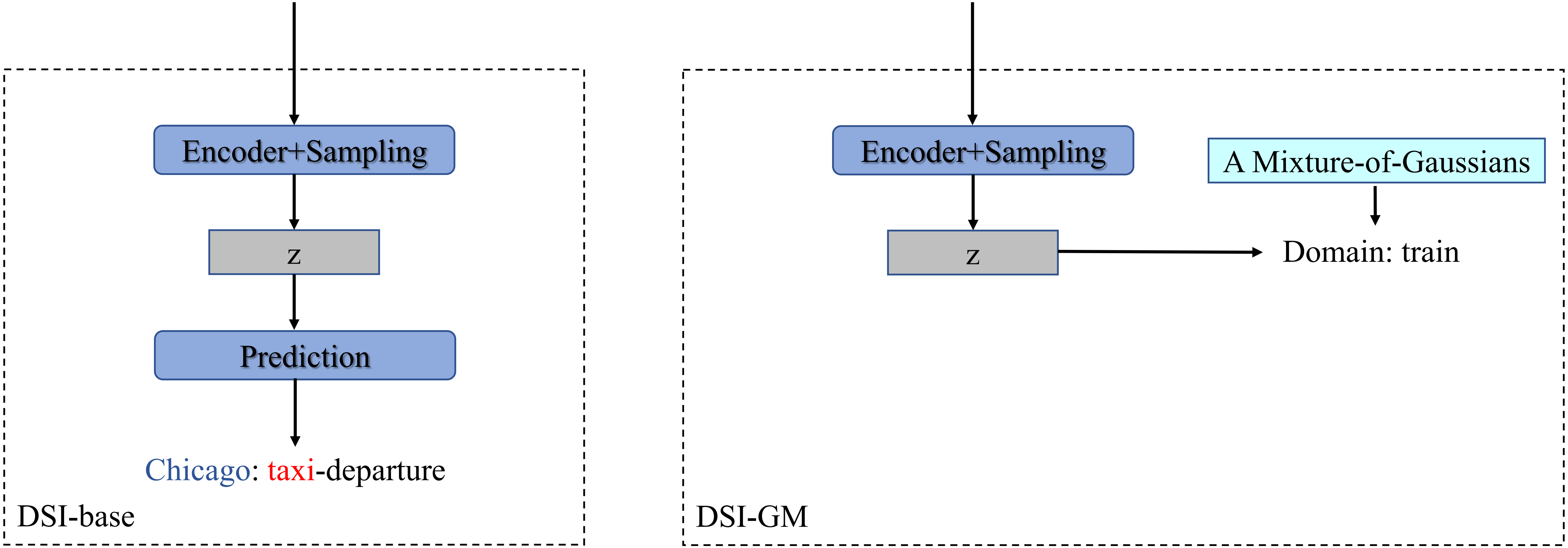


How does *DSI-GM* work?

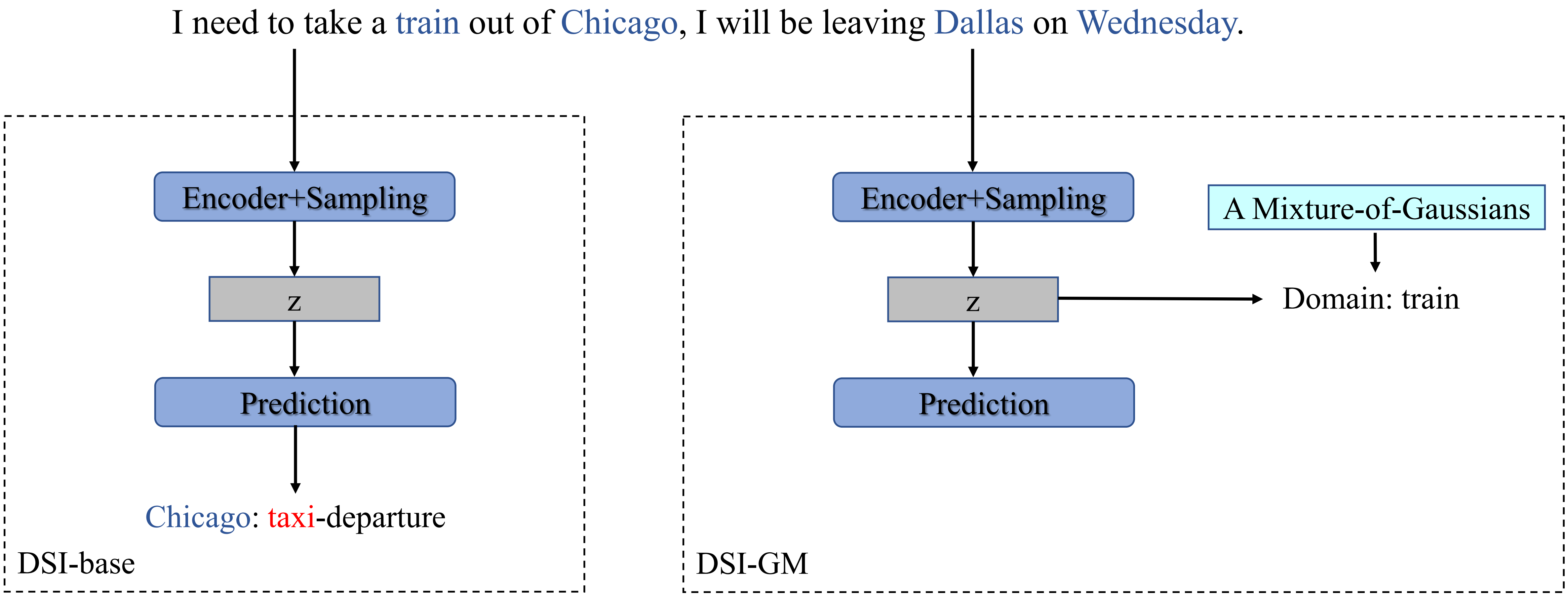


How does *DSI-GM* work?

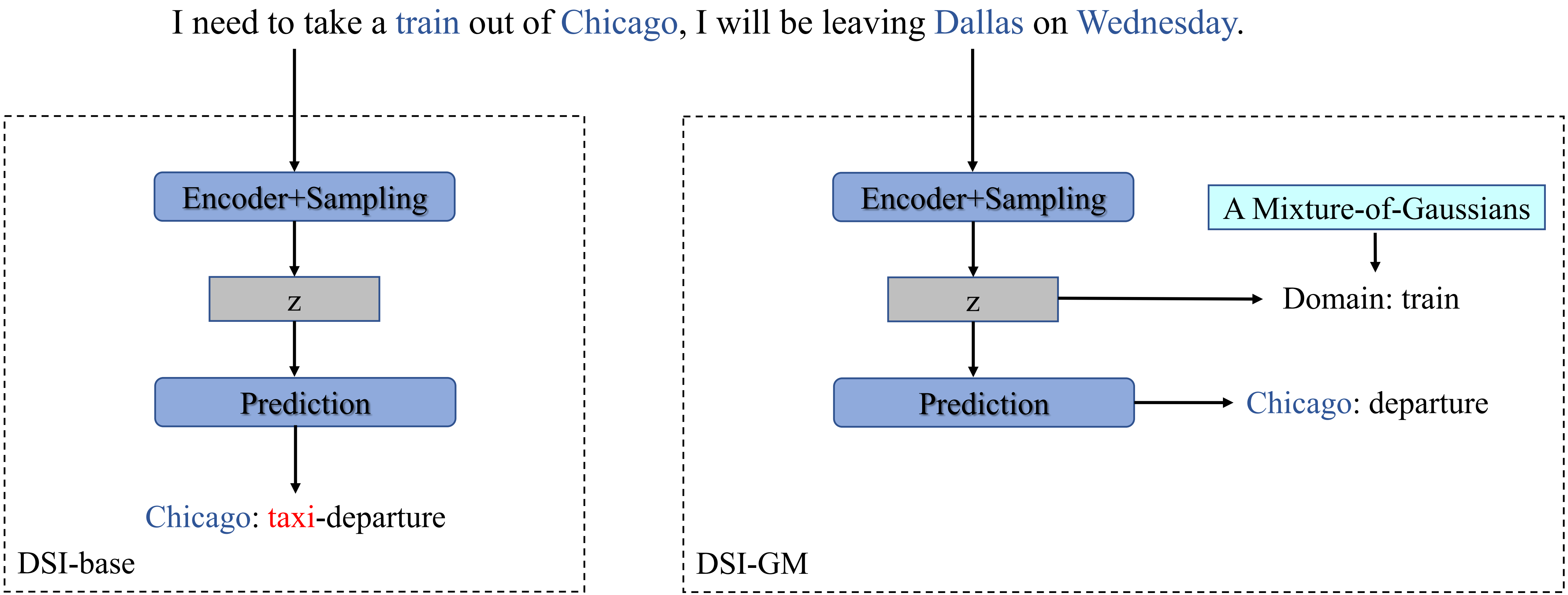
I need to take a train out of Chicago, I will be leaving Dallas on Wednesday.



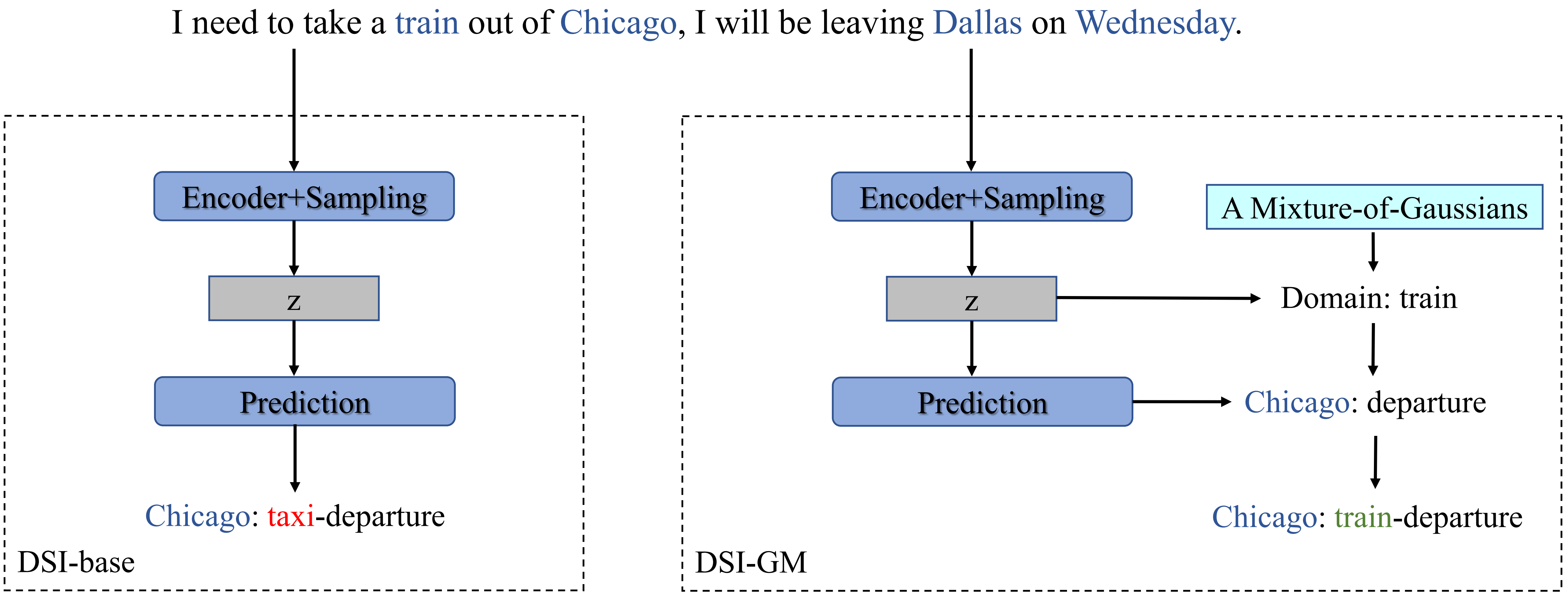
How does *DSI-GM* work?



How does *DSI-GM* work?



How does *DSI-GM* work?



CHAPTER 3

Experiments

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
Random	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
DSI-base	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
DSI-GM	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.

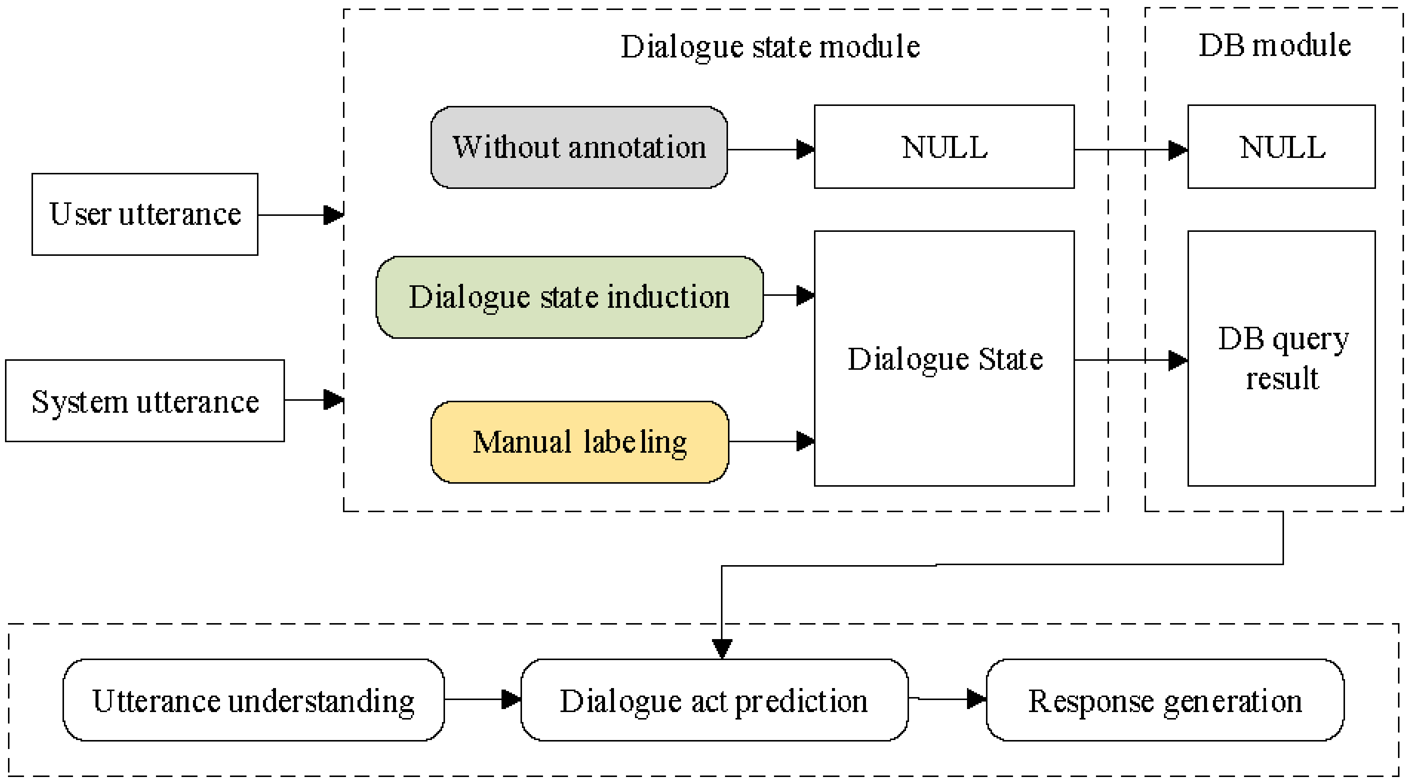
DSI results

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
Random	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
DSI-base	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
DSI-GM	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.

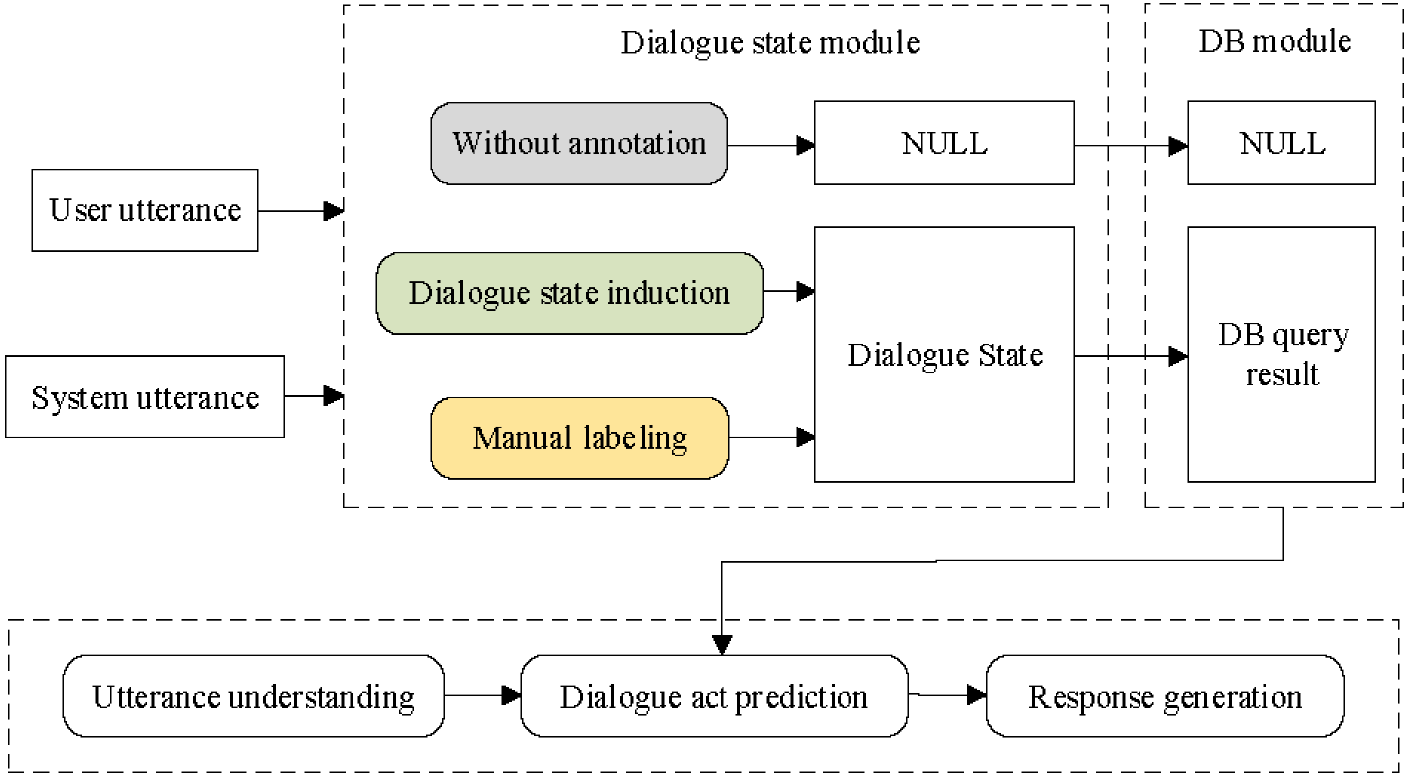
DSI-Based Response Generation

DSI-Based Response Generation



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

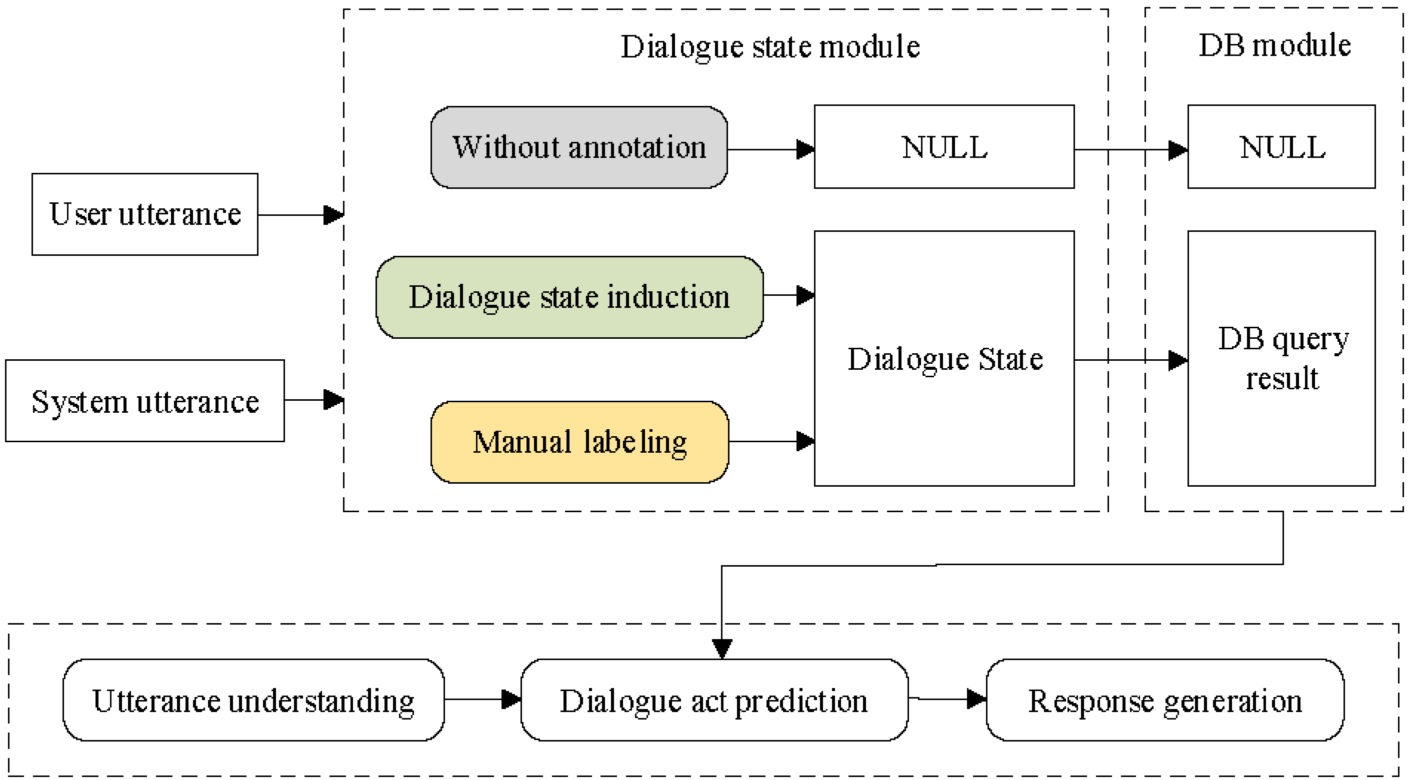
DSI-Based Response Generation



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

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

DSI-Based Response Generation



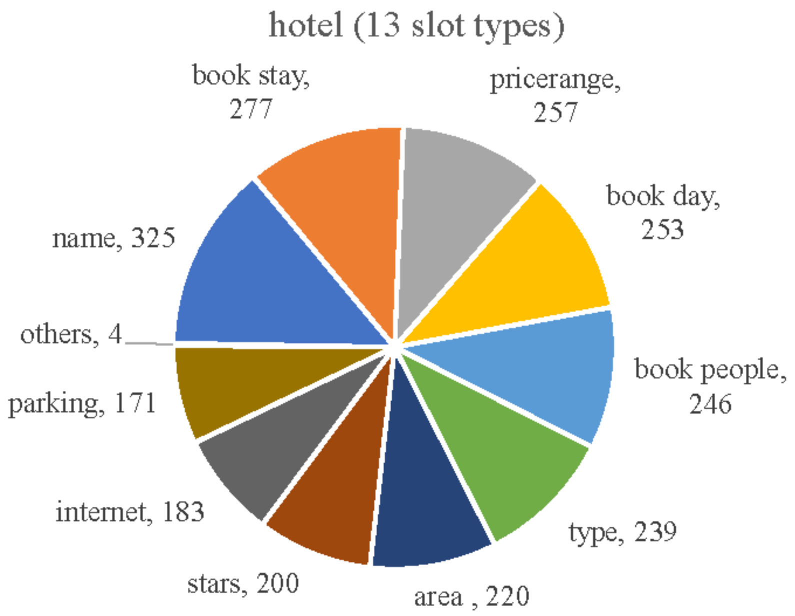
Dialogue State	Dialog Act Prediction			Delexicalized	
	Precision	Recall	F1	BLEU	Entity F1
None	71.0	67.4	69.1	18.7	54.6
DSI-GM	72.0	70.5	71.2	20.8	56.5
Manual labeling	75.6	73.0	74.2	21.6	61.3

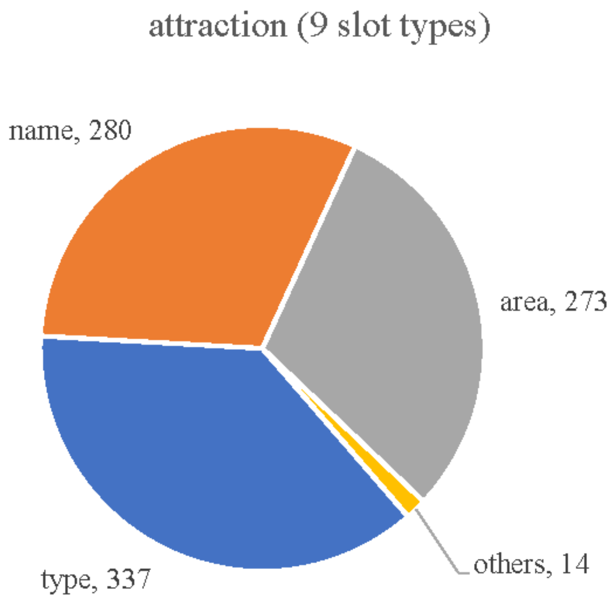
[Chen et al., 2019] Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. In ACL, 2019.

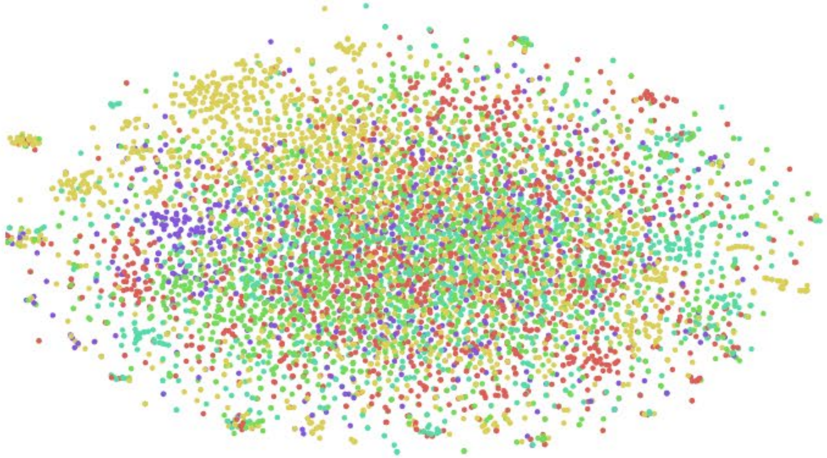
Analysis

	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

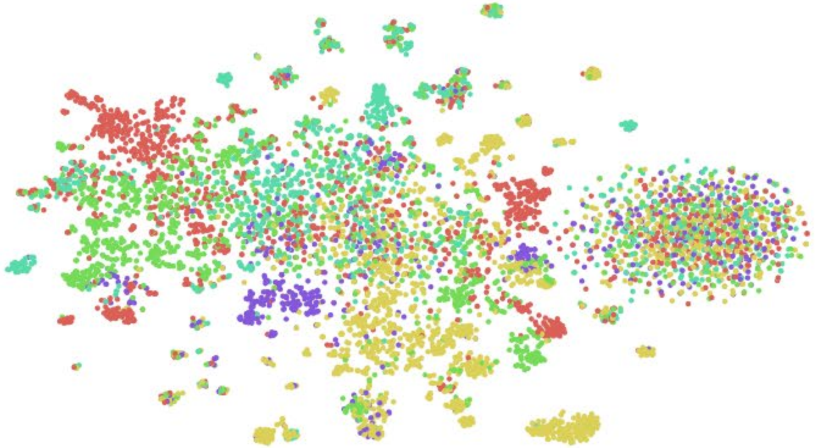
Table 4: Turn goal accuracy per domain.







(a) *DSI-base*



(b) *DSI-GM*

Domain level comparison of the latent representation z.

CHAPTER 4

Conclusion

Conclusion

Conclusion

- Dialogue state induction: a novel task to automatically identify dialogue states

Conclusion

- Dialogue state induction: a novel task to automatically identify dialogue states
- *DSI-base/DSI-GM*: two neural generative models with latent variables

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

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.

西湖大學
WESTLAKE UNIVERSITY

THANK YOU

Contact:
minqingkai@westlake.edu.cn

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

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

paper

GitHub