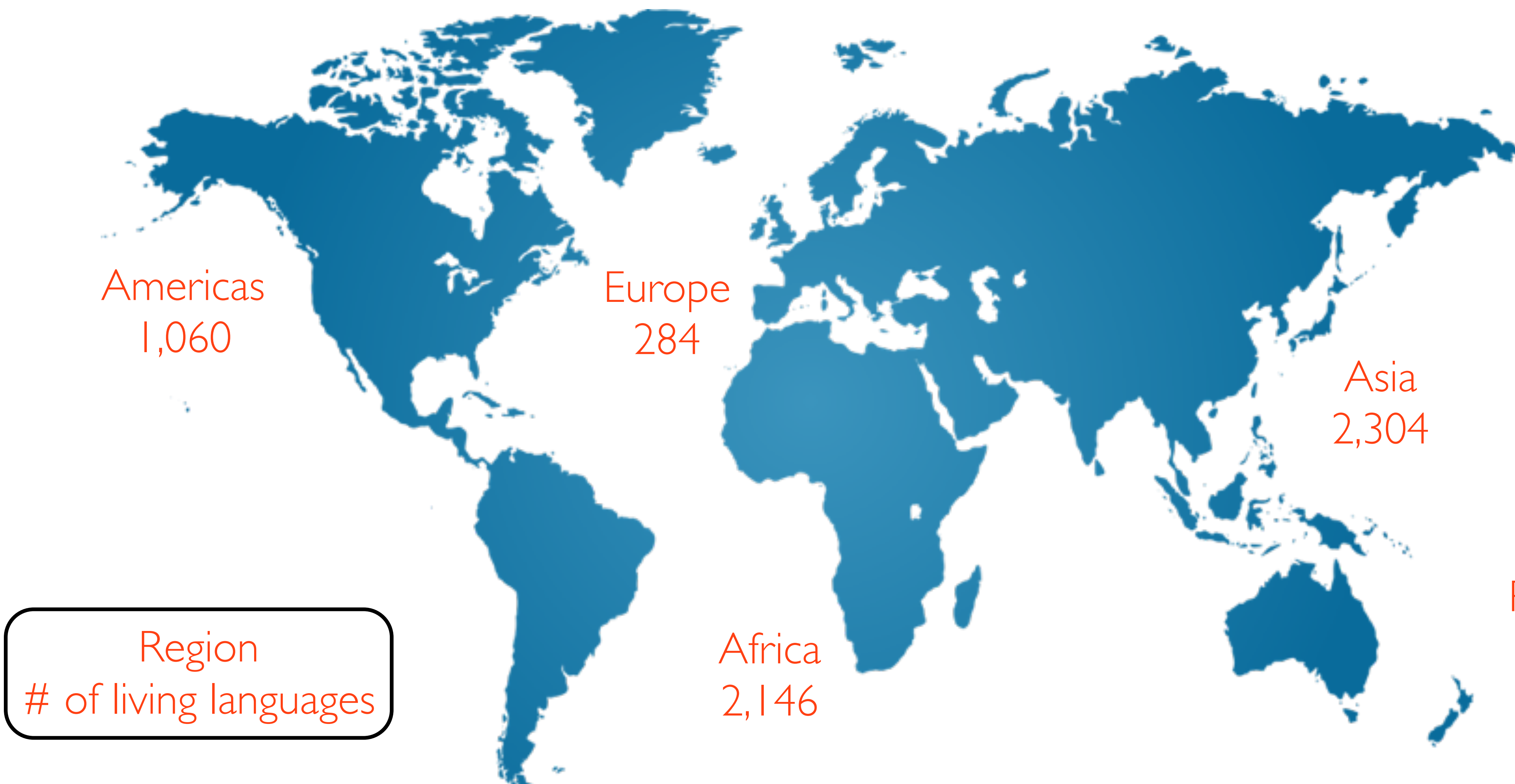
Joint Learning of Phonetic Units and Word Pronunciations for ASR

Chia-ying (Jackie) Lee, Yu Zhang and James Glass

Spoken Language Systems Group
MIT Computer Science and Artificial Intelligence Lab
Cambridge, MA

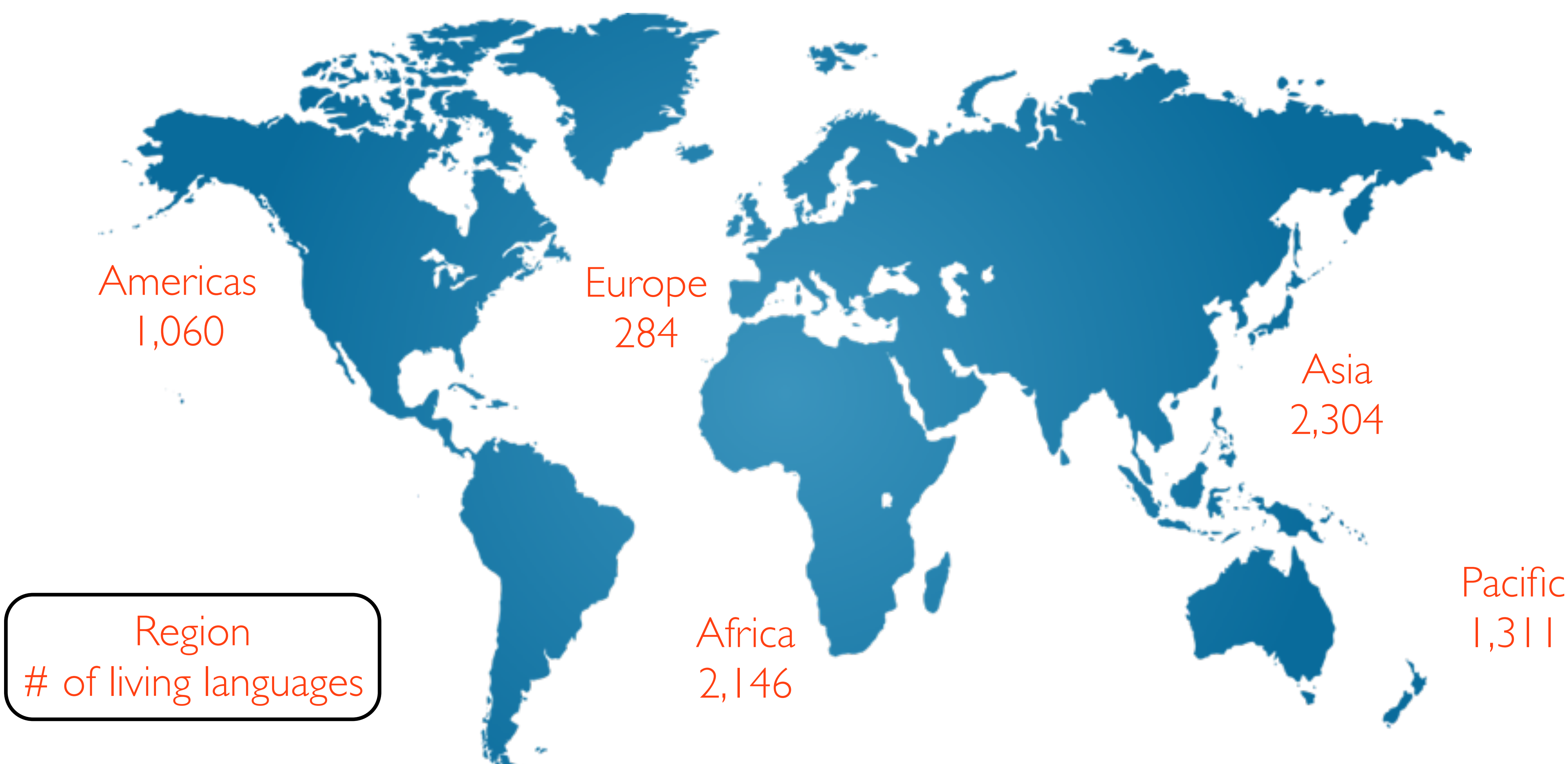
World Language Map



Data source: http://www.ethnologue.com/

World Language Map

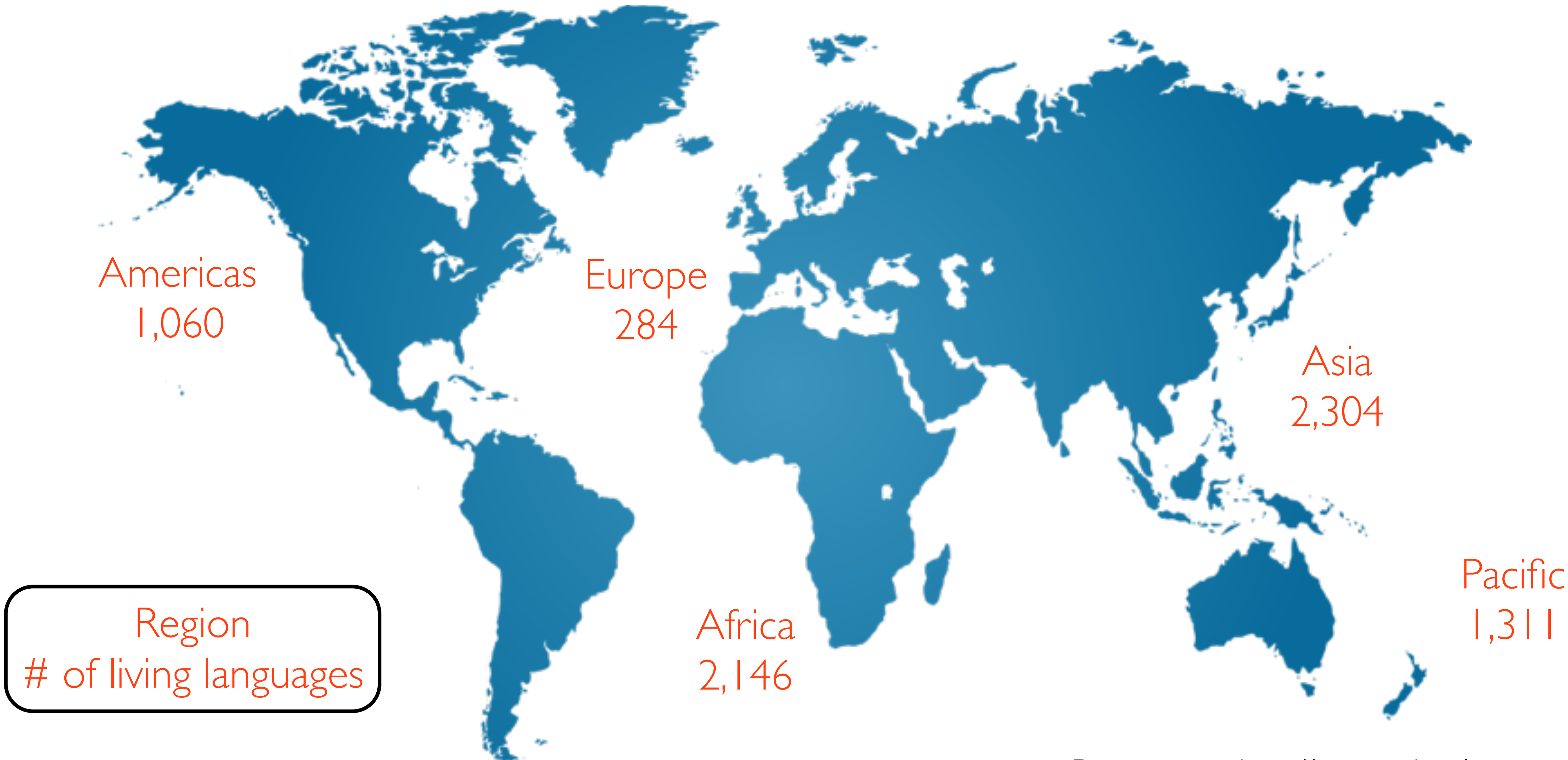
- Roughly 7,000 living languages all around the world



Data source: http://www.ethnologue.com/

World Language Map

- **Roughly 7,000 living languages all around the world**
  - Only 2% are supported by automatic speech recognition (ASR) technology



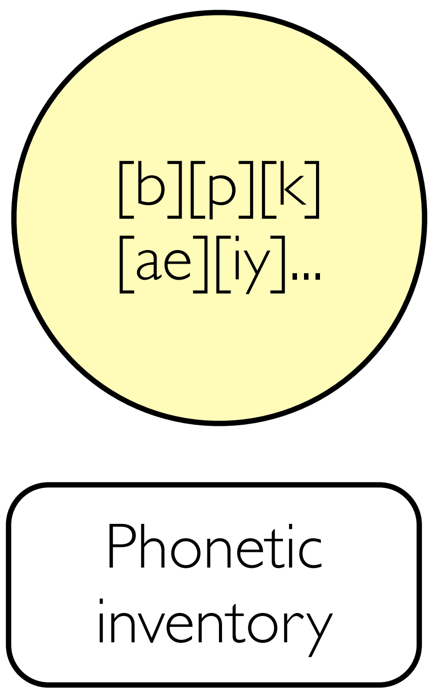
Data source: http://www.ethnologue.com/

2% Language Barrier

- Conventional ASR training is expensive
  - Requires a lot of expert knowledge

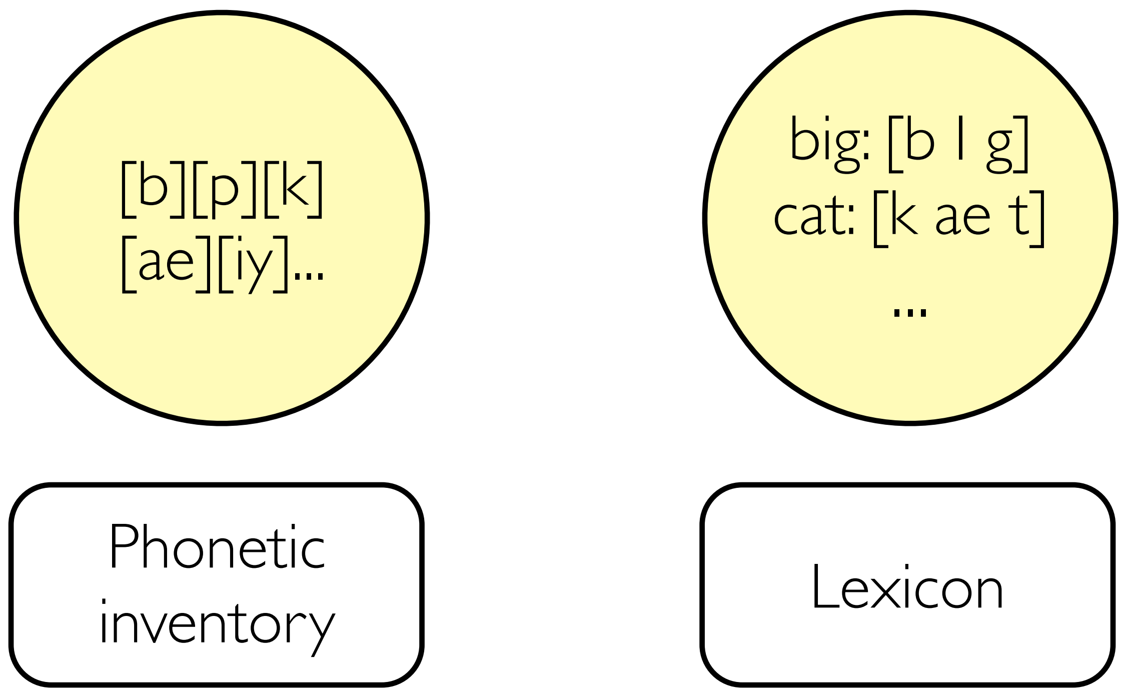
2% Language Barrier

- Conventional ASR training is expensive
  - Requires a lot of expert knowledge



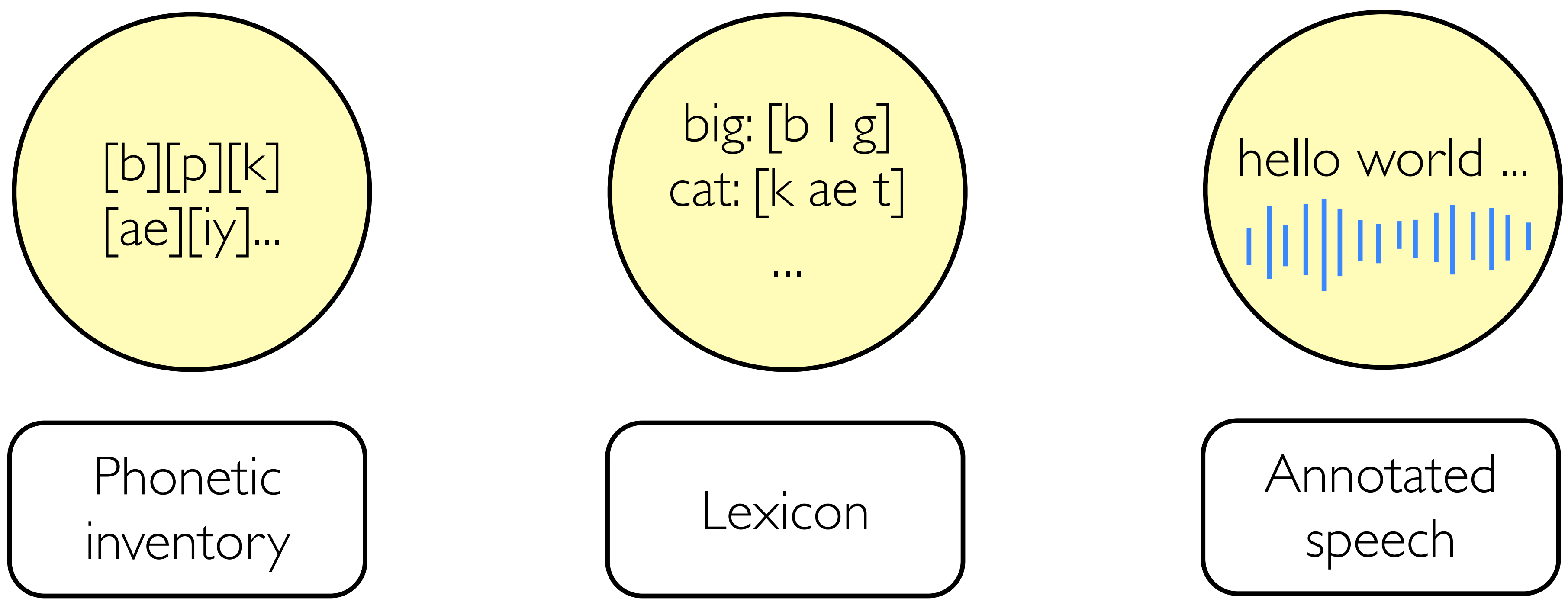
2% Language Barrier

- Conventional ASR training is expensive
  - Requires a lot of expert knowledge



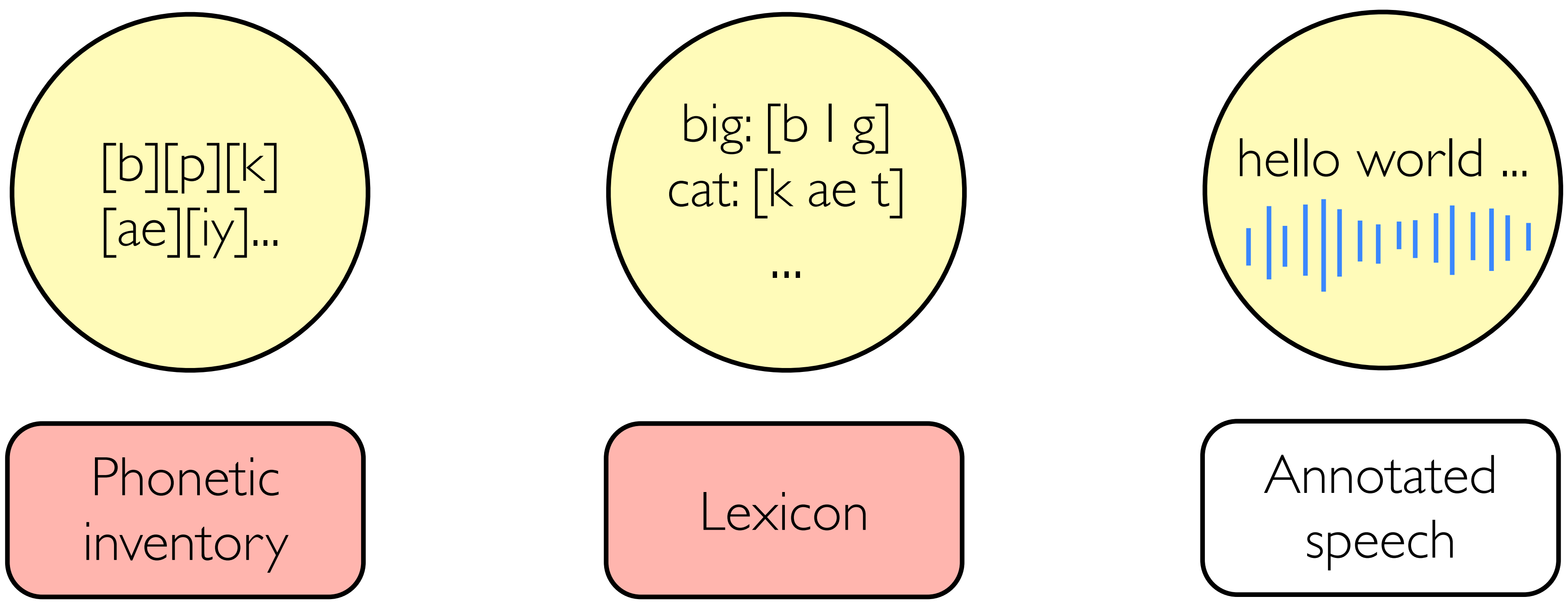
2% Language Barrier

- Conventional ASR training is expensive
  - Requires a lot of expert knowledge



2% Language Barrier

- Conventional ASR training is expensive
  - Requires a lot of expert knowledge

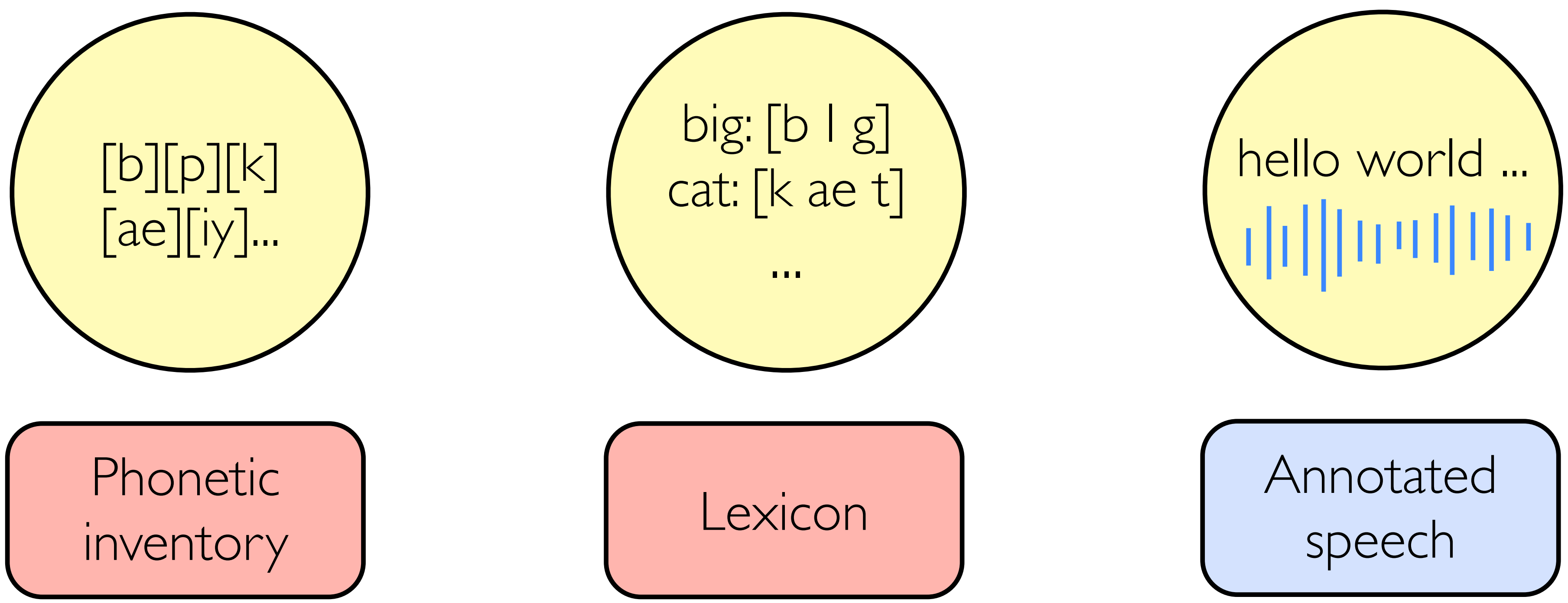


difficult to collect

require linguistic expert knowledge

2% Language Barrier

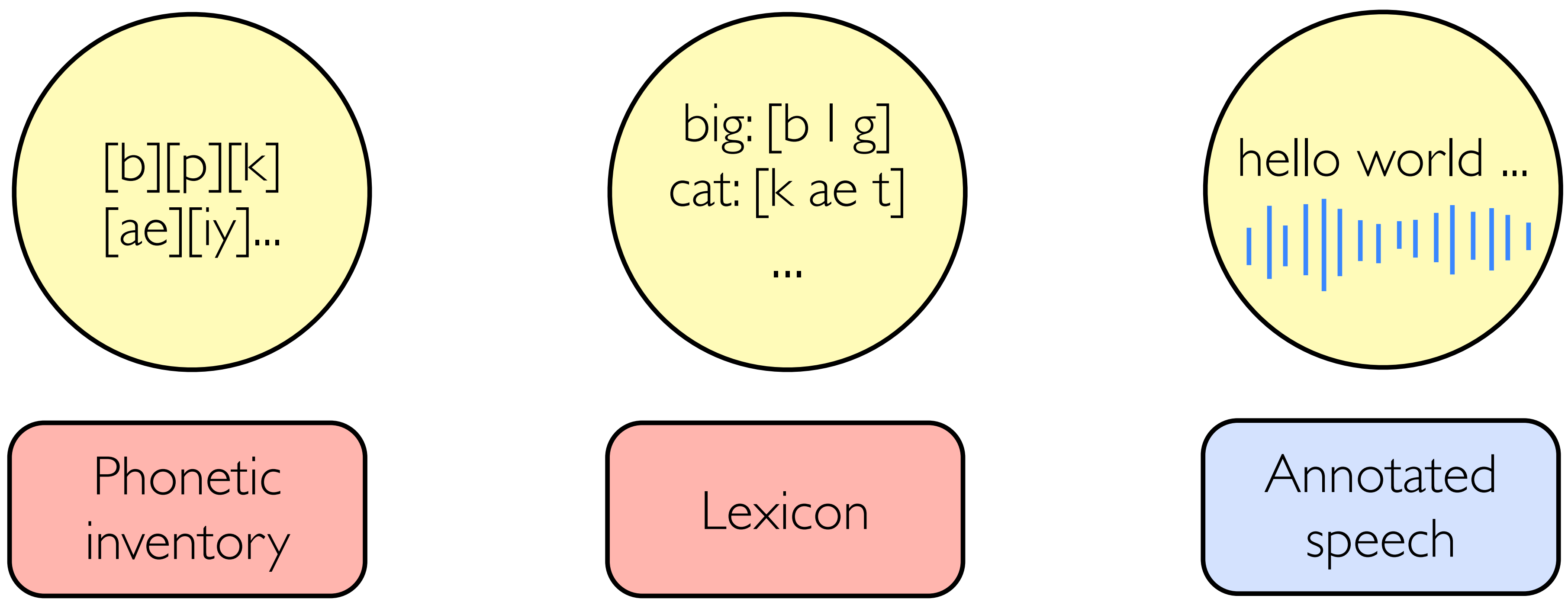
- Conventional ASR training is expensive
  - Requires a lot of expert knowledge



difficult to collect
require linguistic expert knowledge

easier to generate
by non-experts

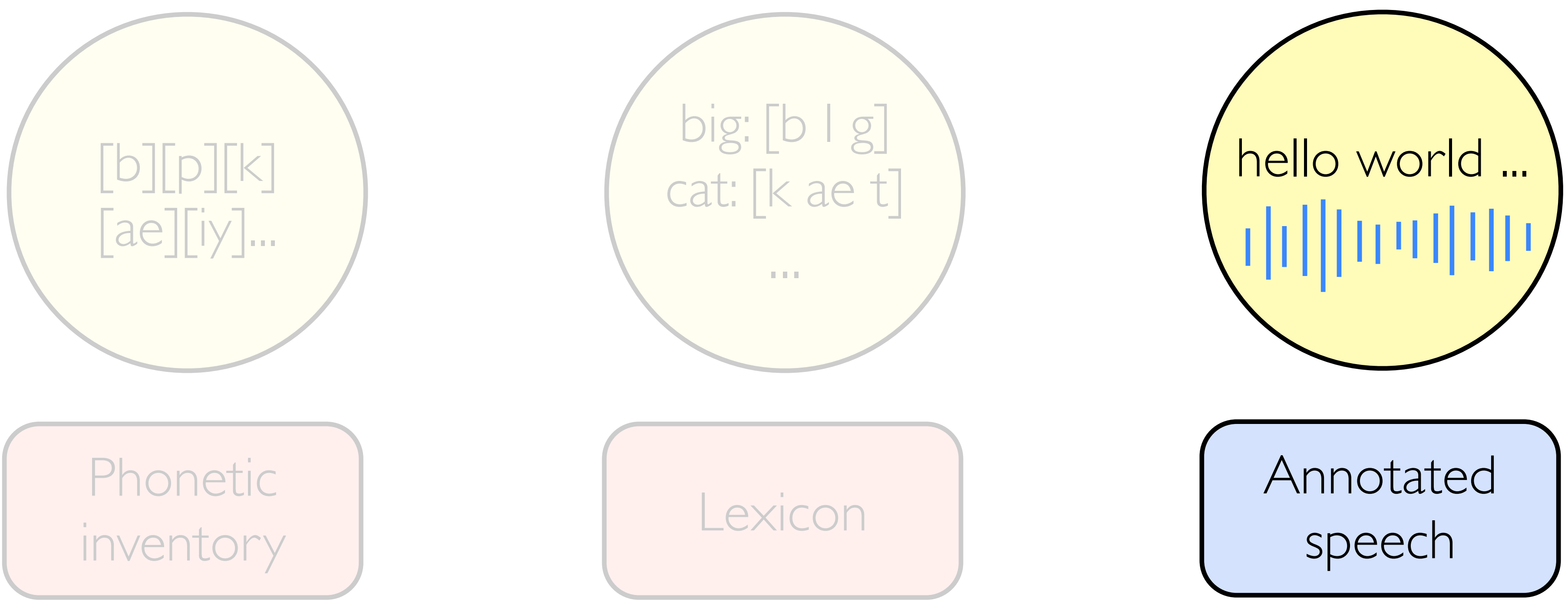
Towards ASR Training without Experts



difficult to collect
require linguistic expert knowledge

easier to generate
by non-experts

Towards ASR Training without Experts

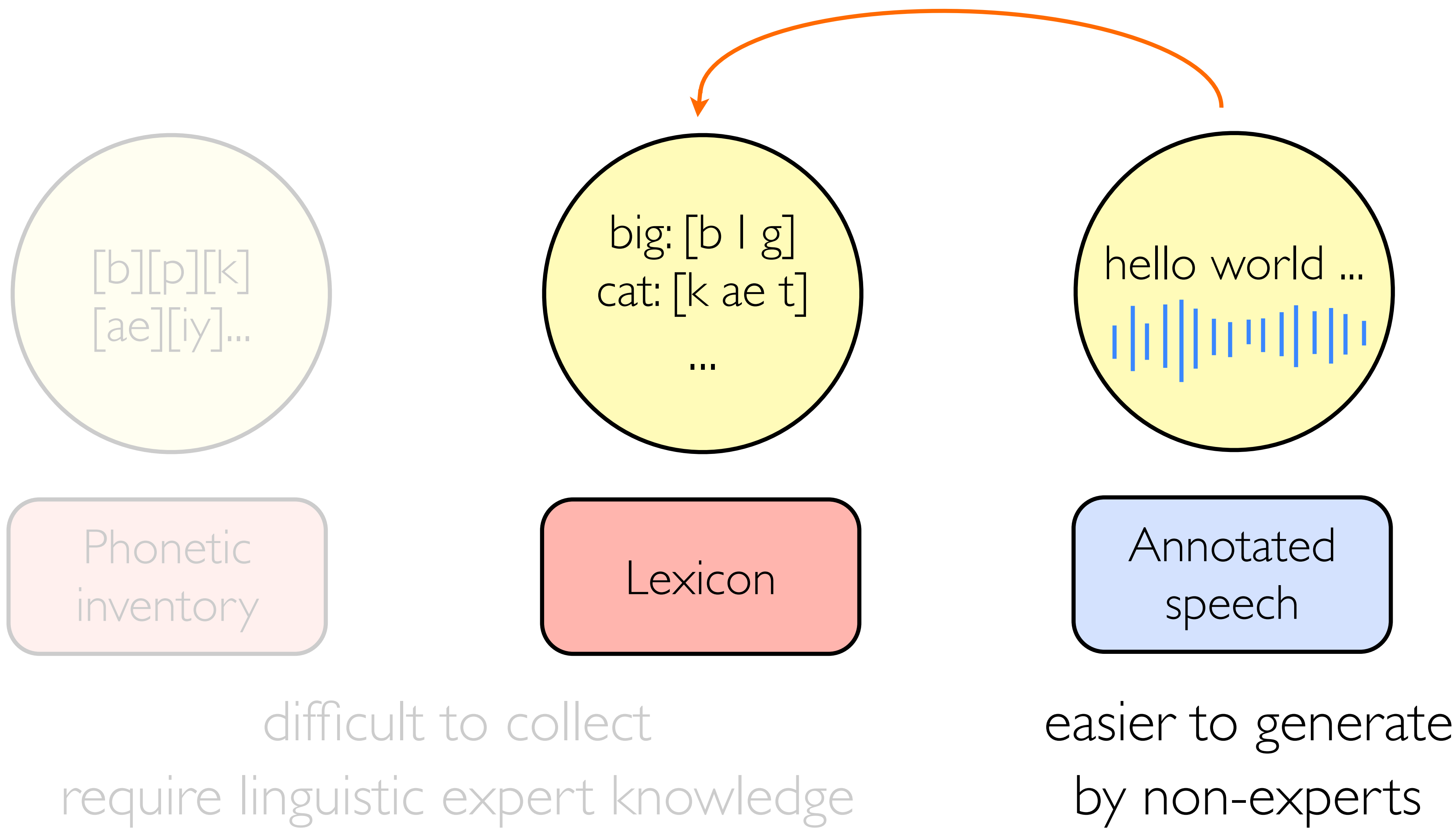


difficult to collect
require linguistic expert knowledge

easier to generate
by non-experts

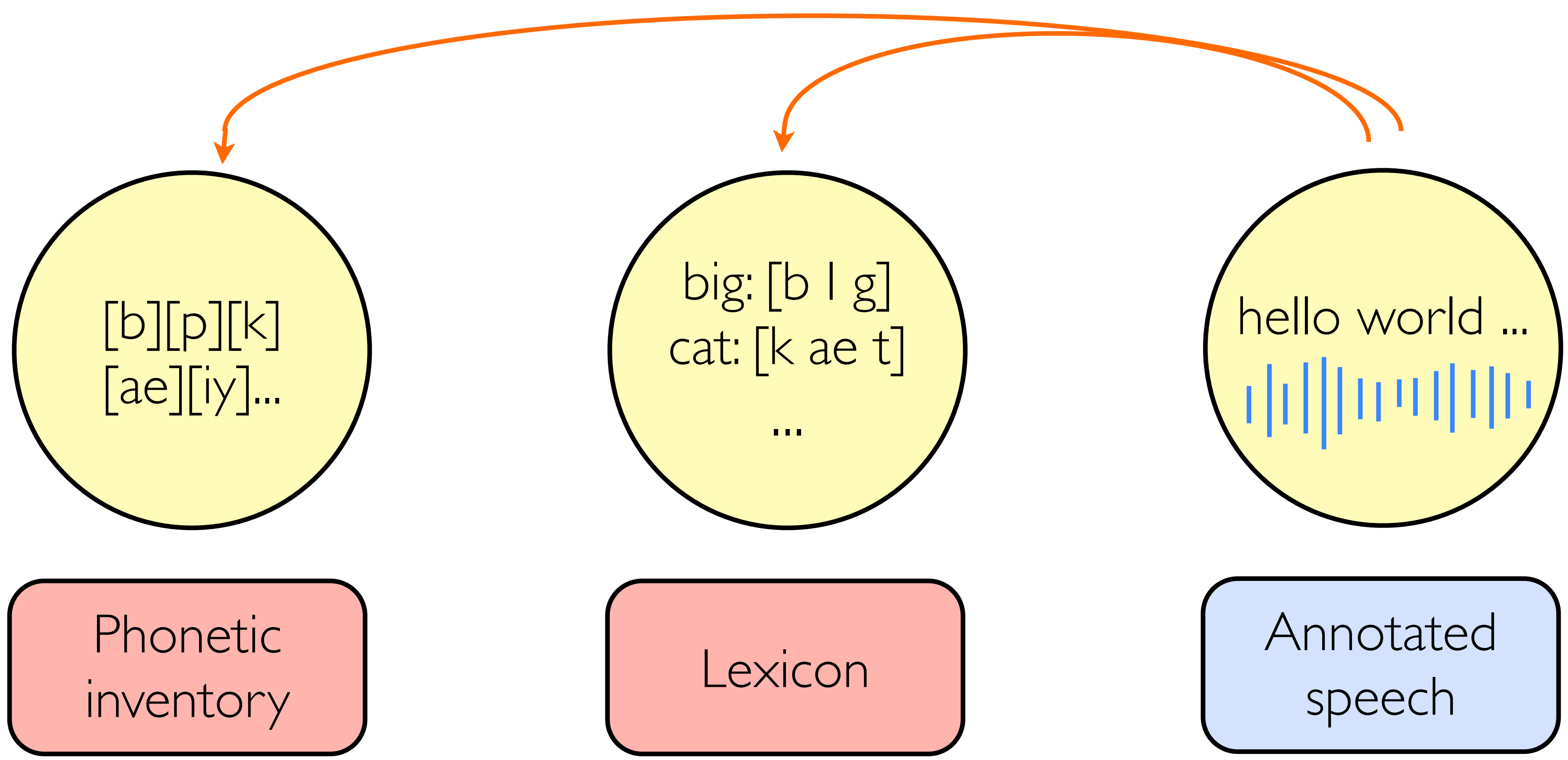
Towards ASR Training without Experts

- Infer lexicon and phonetic units from transcribed speech



Towards ASR Training without Experts

- Infer lexicon and phonetic units from transcribed speech

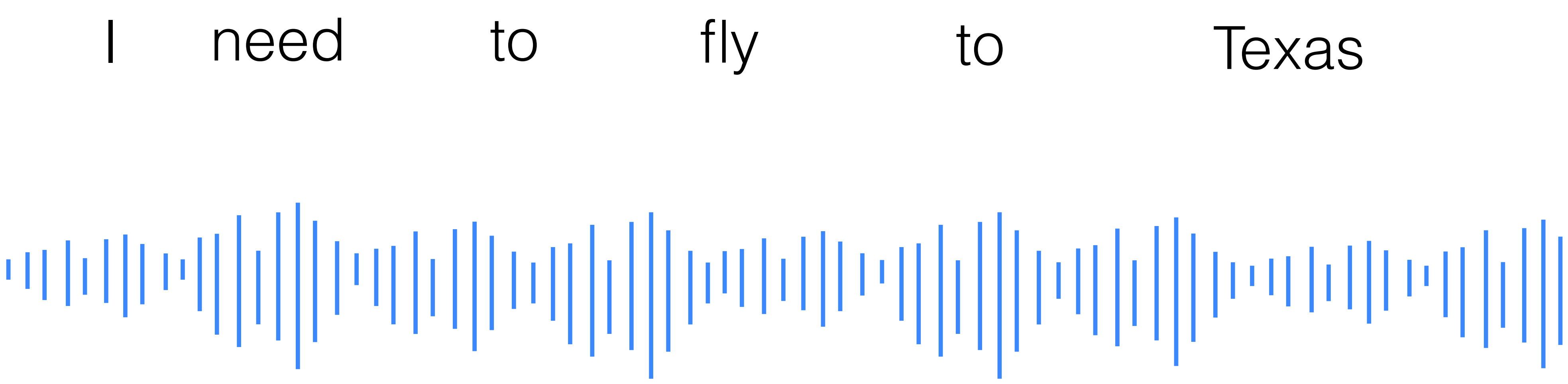


difficult to collect
require linguistic expert knowledge

easier to generate
by non-experts

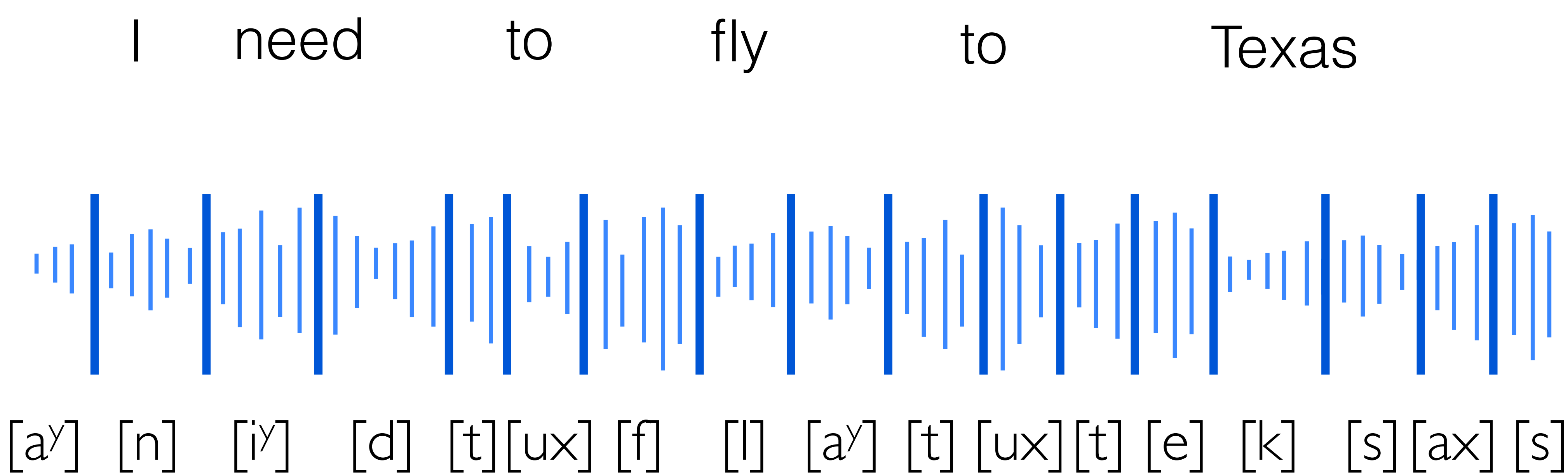
Discover Pronunciation Lexicon

- Learn word pronunciations from transcribed speech



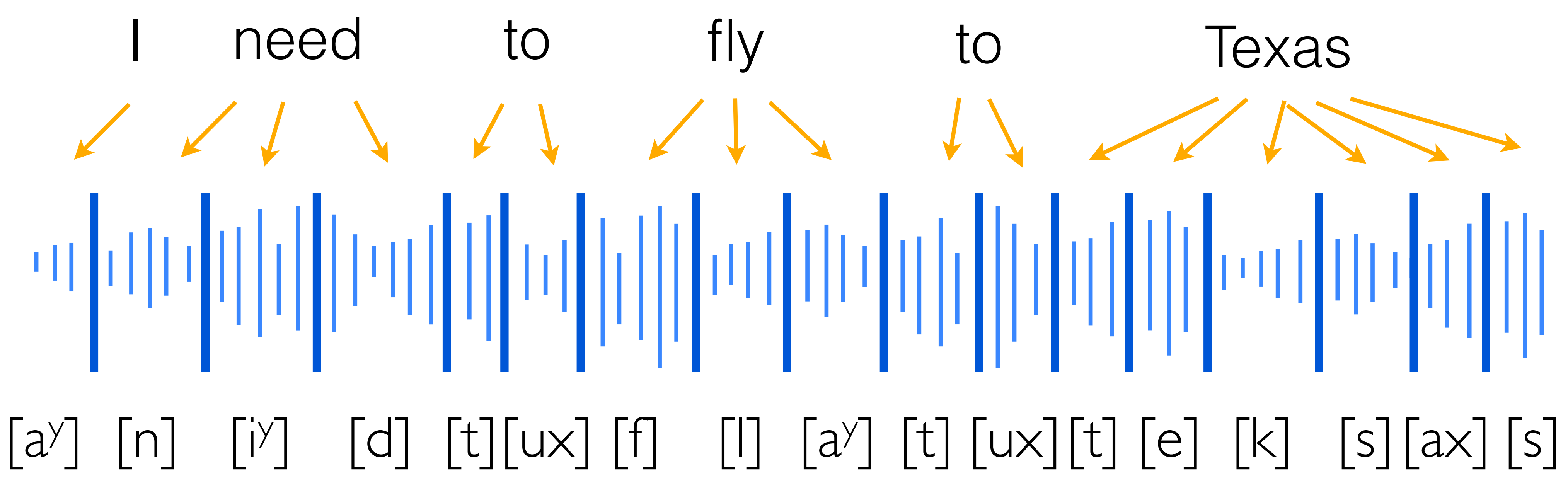
Discover Pronunciation Lexicon

- Learn word pronunciations from transcribed speech



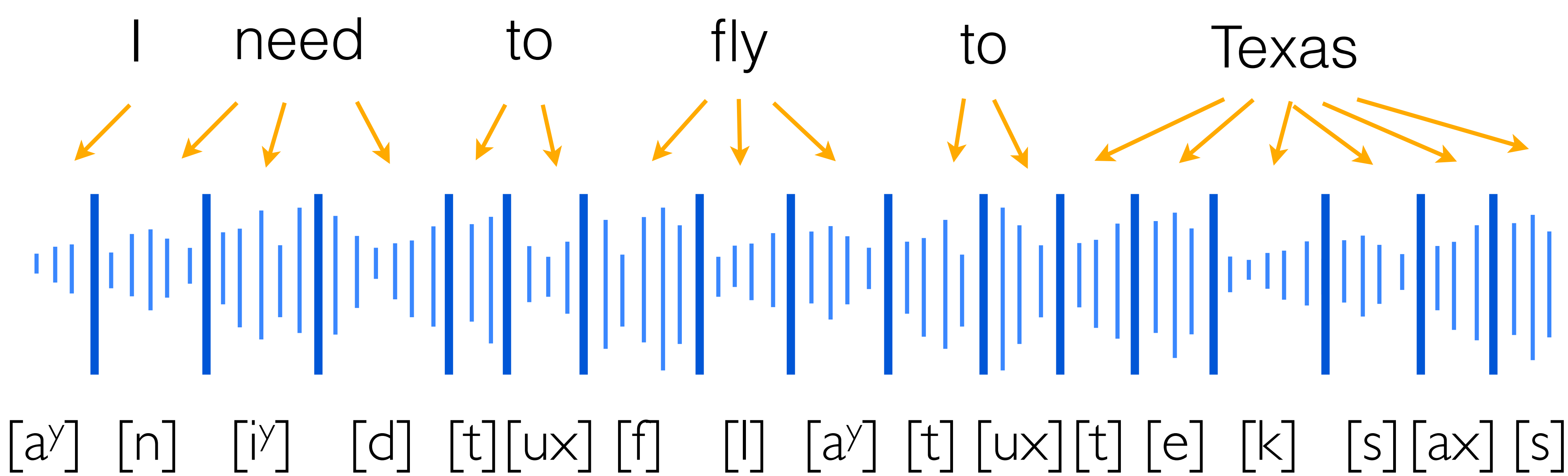
Discover Pronunciation Lexicon

- Learn word pronunciations from transcribed speech



Discover Pronunciation Lexicon

- Learn word pronunciations from transcribed speech



I : [a^y]

need : [n i^y d]

to : [t ux]

fly : [f l a^y]

...

Without Linguistic Knowledge

- Can we discover the word pronunciations?

ང་གཟུགས་པོ་སྣང་ད་གམེད། ཐུགས་རྗེ་ཆེ།

Without Linguistic Knowledge

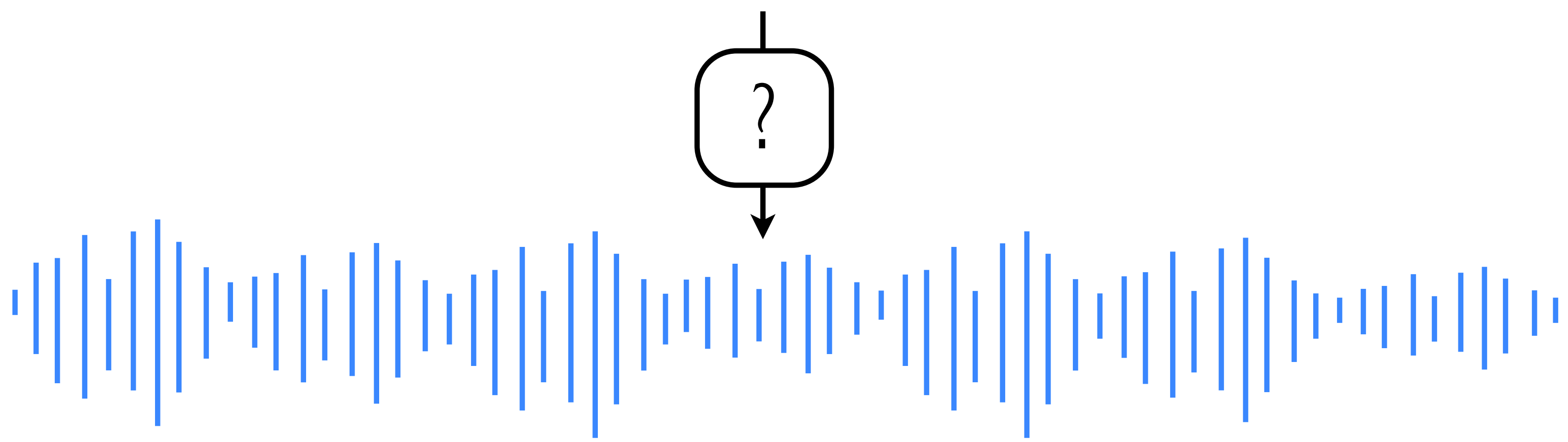
- Can we discover the word pronunciations?

ང་གཟུགས་པོ་སྣང་ད་གསེད། ཐུགས་རྗེ་ཆེ།

Without Linguistic Knowledge

- Can we discover the word pronunciations?

ང་གཟུགས་པོ་སྣང་ད་གམེད། ཐུགས་རྗེ་ཆེ།

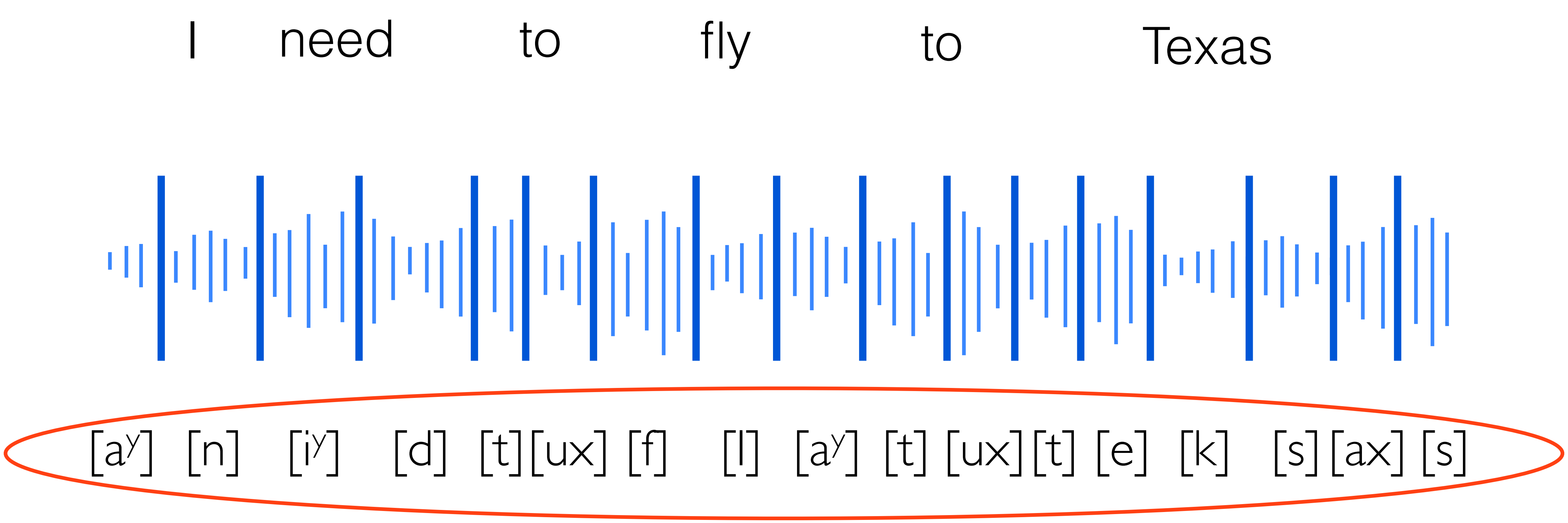


Challenges

I need to fly to Texas

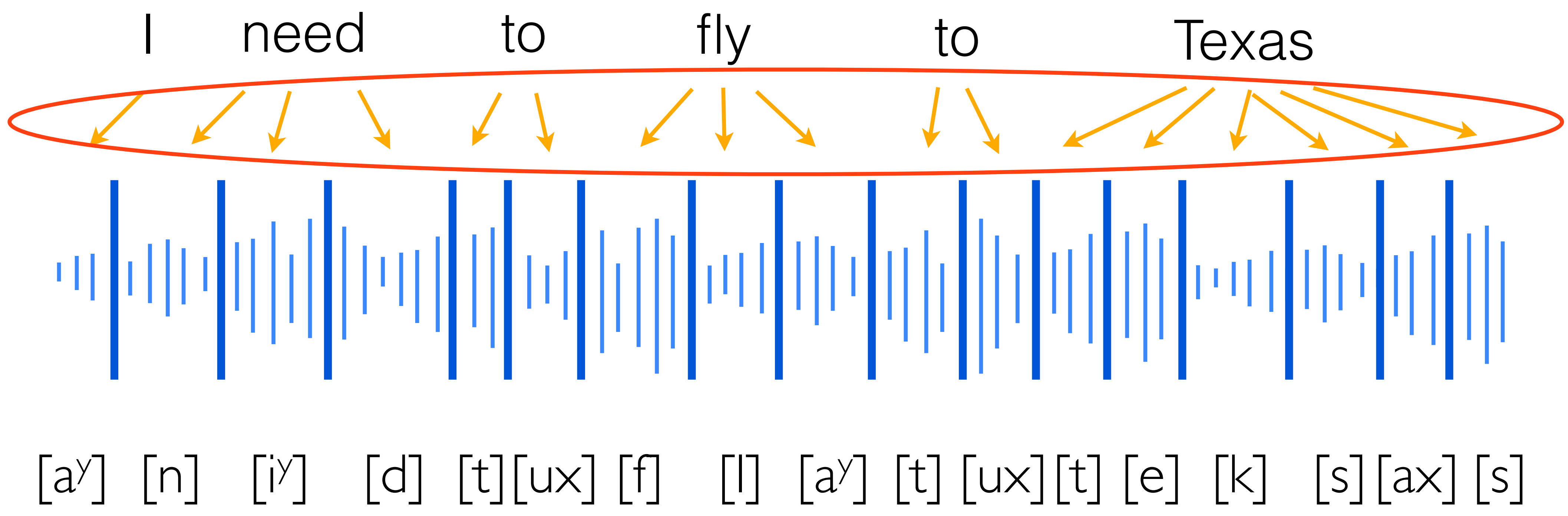
Challenges

- Latent phone sequence



Challenges

- Latent phone sequence
- Latent letter to sound (L2S) mapping rules



Hierarchical Bayesian Model

Hierarchical Bayesian Model

- Unknown phone sequence

Hierarchical Bayesian Model

- Unknown phone sequence
  - Unknown phone inventory

Hierarchical Bayesian Model

- Unknown phone sequence
  - Unknown phone inventory
  - HMM-based mixture model

Hierarchical Bayesian Model

- Unknown phone sequence
  - Unknown phone inventory
  - HMM-based mixture model

θ_k: HMM θ_1 θ_2 θ_3 ⋯ θ_K

[s]   [i^y]   [z]   [k]

Hierarchical Bayesian Model

- **Unknown phone sequence**
  - Unknown phone inventory
  - HMM-based mixture model
- **Unknown L2S rules**
  - Weights over HMMs

θ_k: HMM

θ_1	θ_2	θ_3	⋯	θ_K
[s]	[iʸ]	[z]		[k]

Hierarchical Bayesian Model

- **Unknown phone sequence**
  - Unknown phone inventory
  - HMM-based mixture model
- **Unknown L2S rules**
  - Weights over HMMs
  - Associated with each letter

θ_k: HMM	θ_1	θ_2	θ_3	$\cdots$	θ_K
	[s]	[i^y]	[z]		[k]

Hierarchical Bayesian Model

- Unknown phone sequence
  - Unknown phone inventory
  - HMM-based mixture model
- Unknown L2S rules
  - Weights over HMMs
  - Associated with each letter

θ_k: HMM

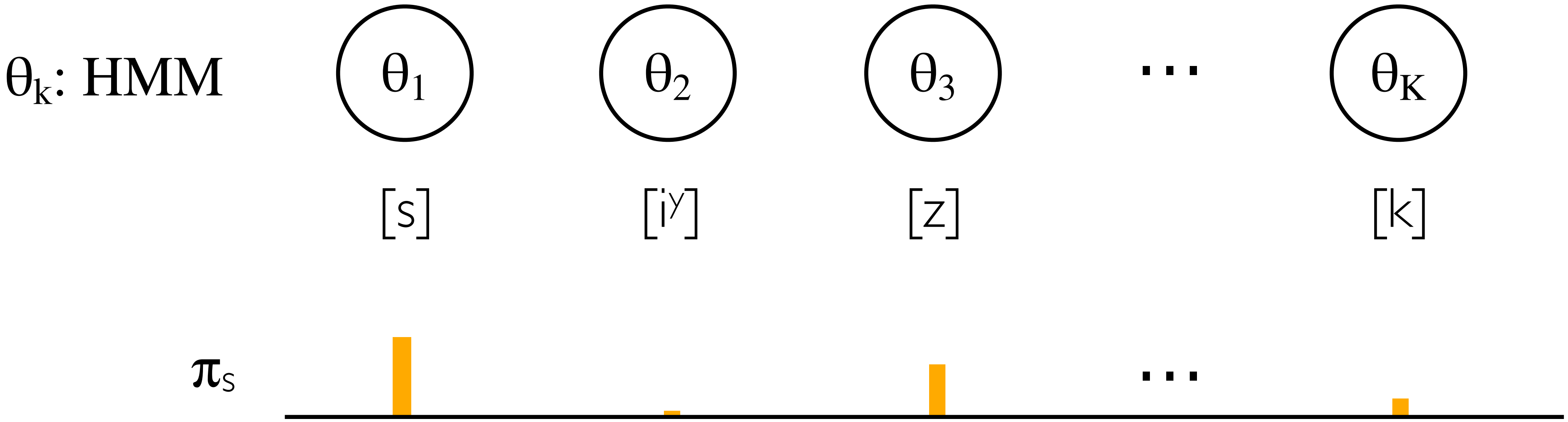
θ_1 θ_2 θ_3 ... θ_K

[s] [iʸ] [z] [k]

π_s

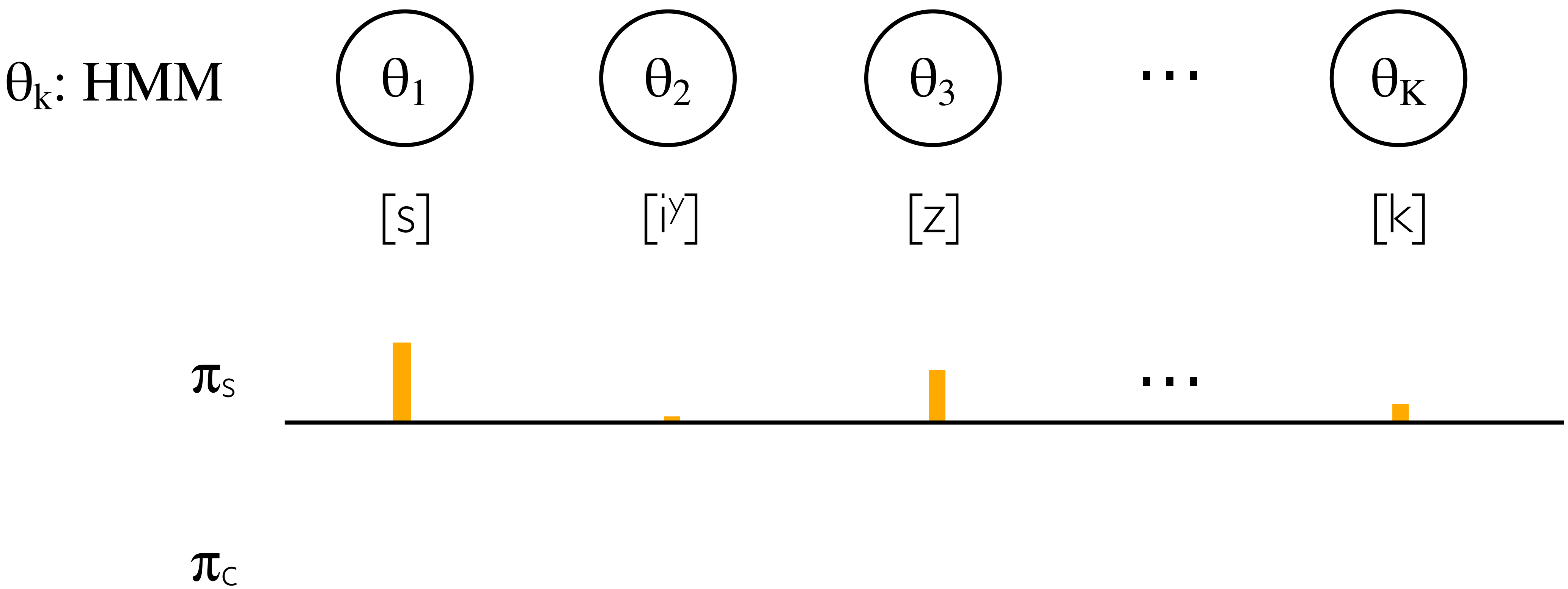
Hierarchical Bayesian Model

- **Unknown phone sequence**
  - Unknown phone inventory
  - HMM-based mixture model
- **Unknown L2S rules**
  - Weights over HMMs
  - Associated with each letter



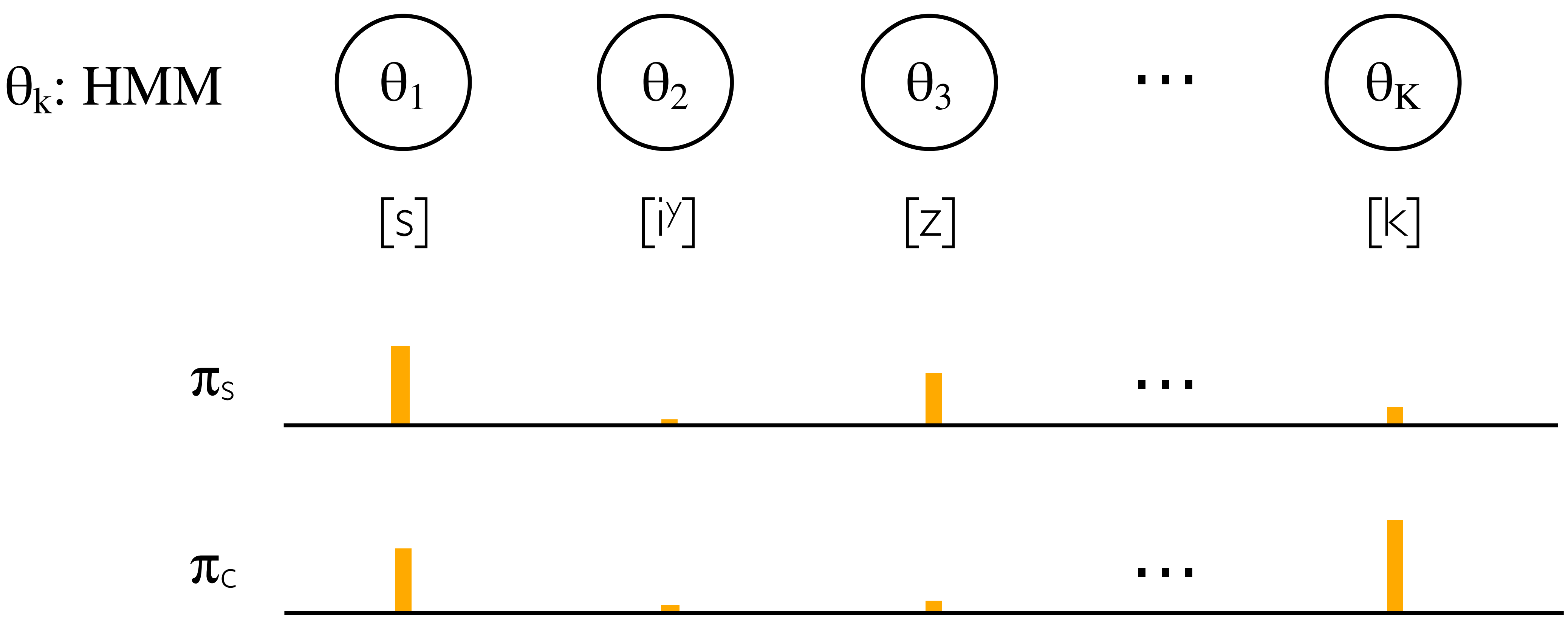
Hierarchical Bayesian Model

- **Unknown phone sequence**
  - Unknown phone inventory
  - HMM-based mixture model
- **Unknown L2S rules**
  - Weights over HMMs
  - Associated with each letter

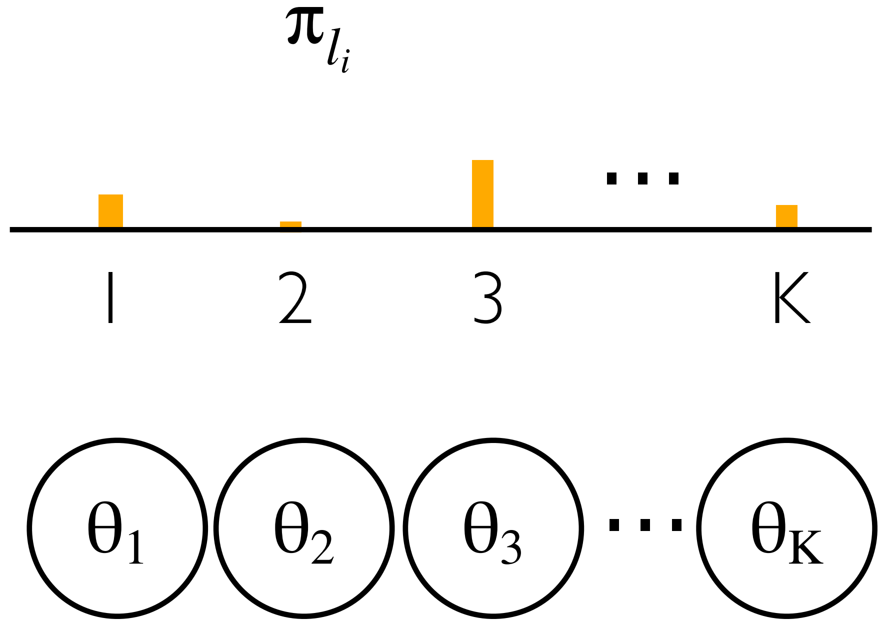


Hierarchical Bayesian Model

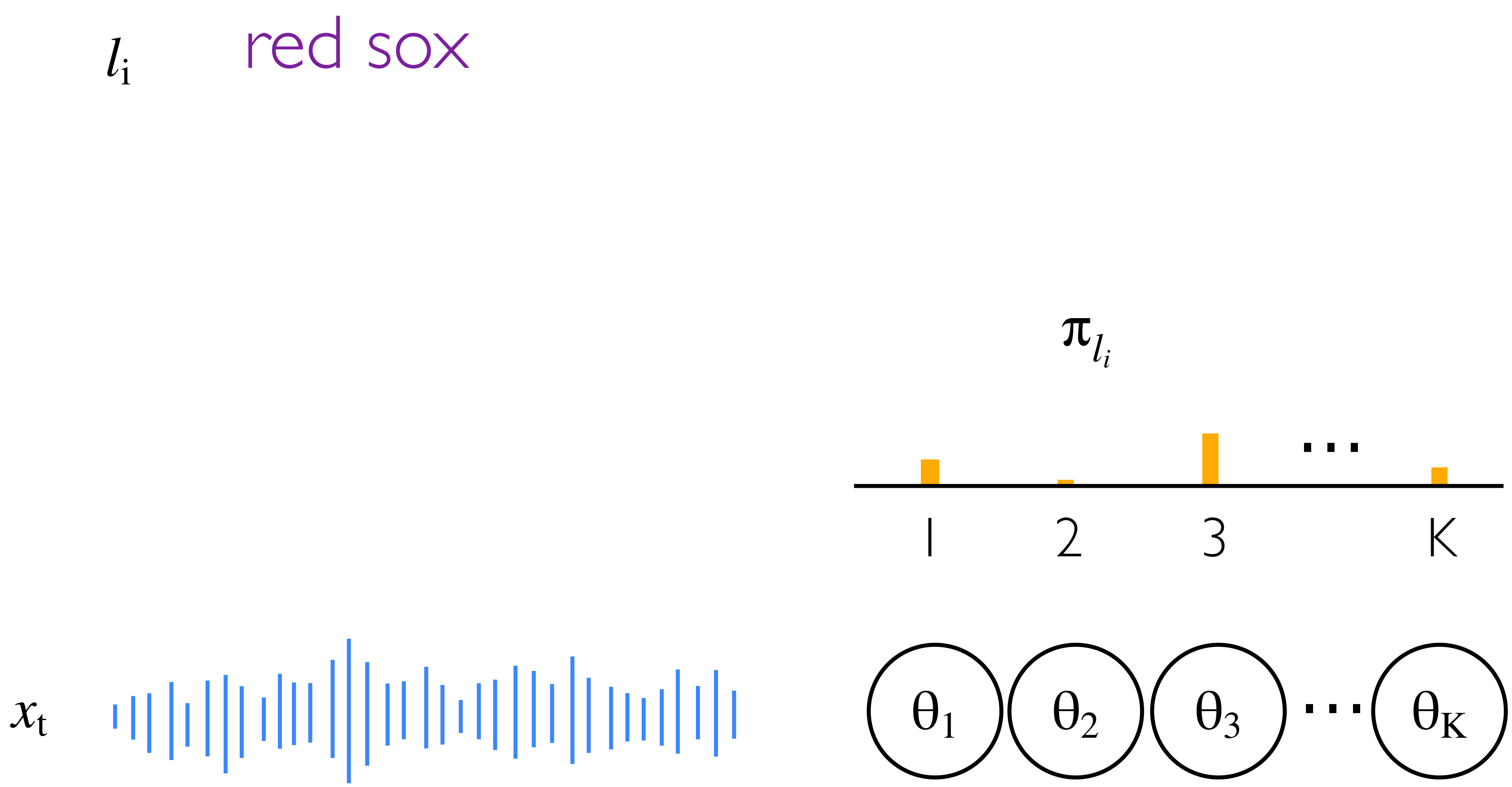
- **Unknown phone sequence**
  - Unknown phone inventory
  - HMM-based mixture model
- **Unknown L2S rules**
  - Weights over HMMs
  - Associated with each letter



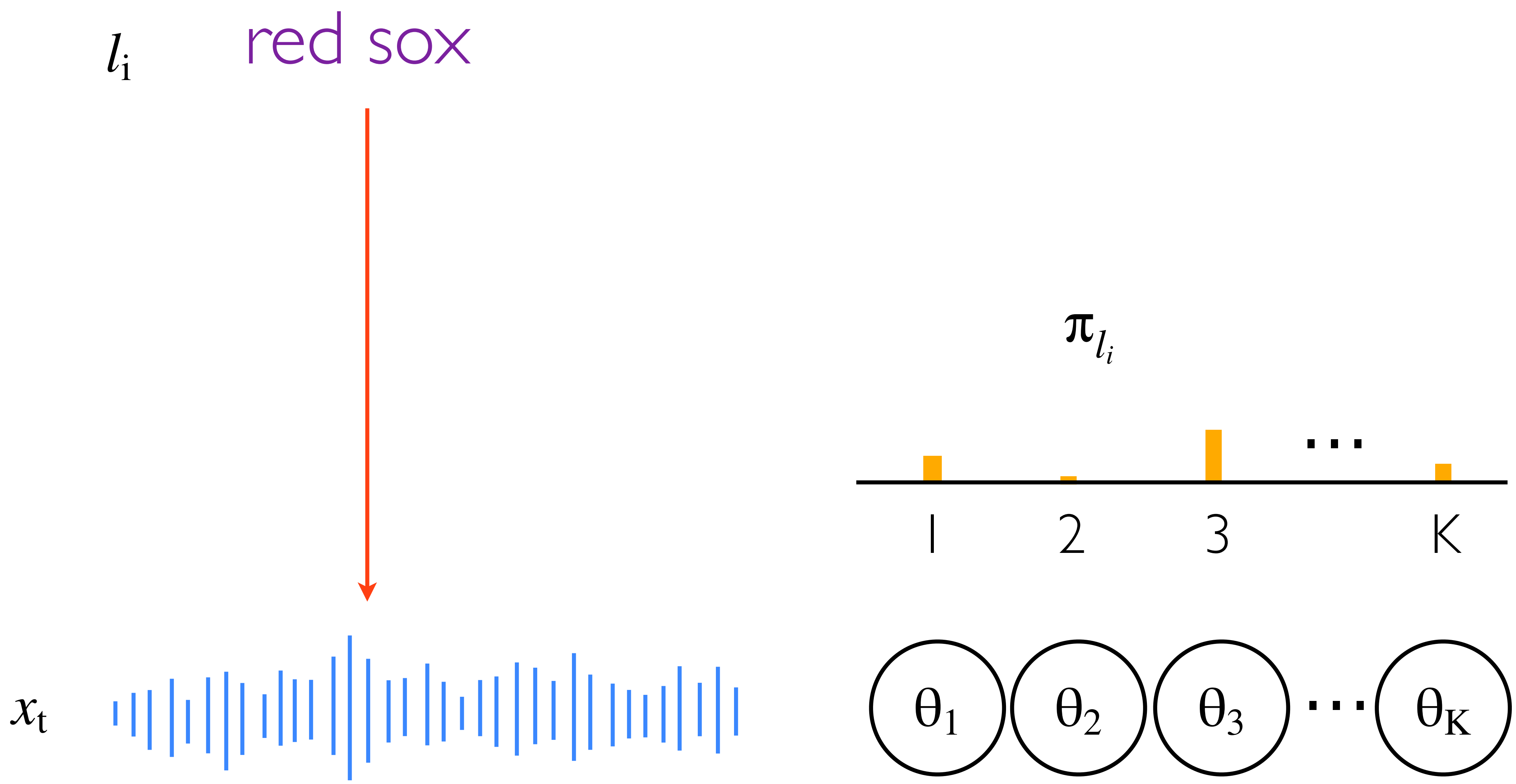
Generative Process



Generative Process

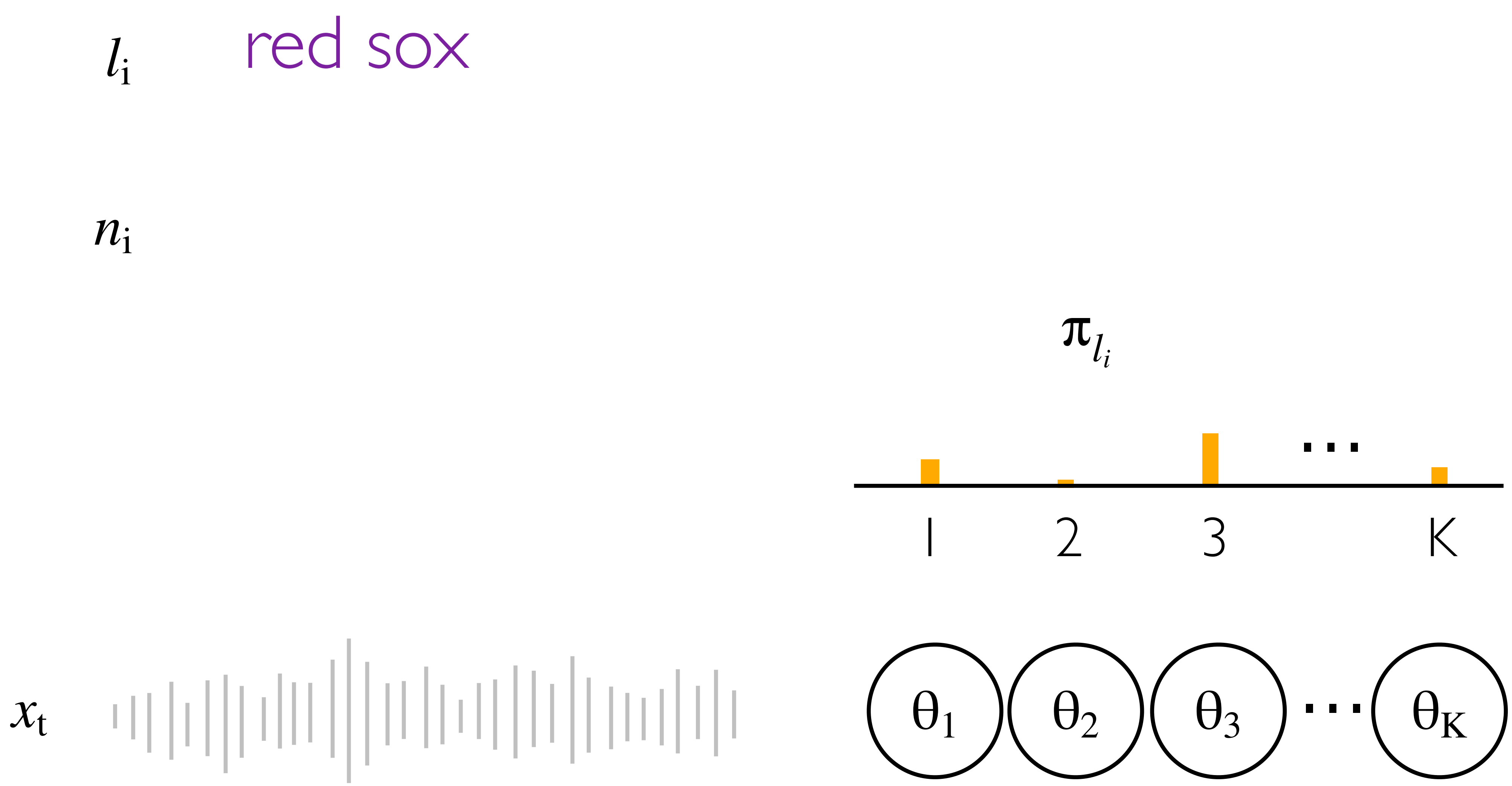


Generative Process



Generative Process

- Step 1
  - Generate the number of phones that each letter maps to (n_i)



Generative Process

- Step 1
  - Generate the number of phones that each letter maps to (n_i)

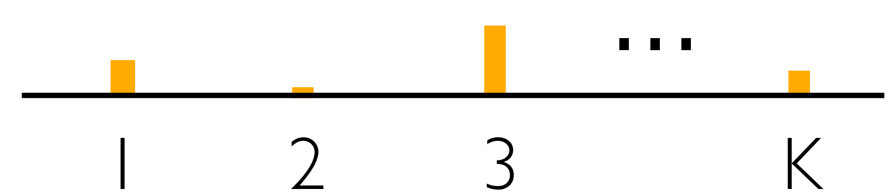
l_i red sox

n_i

$$n_i \sim \phi_{l_i}$$

3-dim categorical distribution

π_{l_i}



x_t

θ_1 θ_2 θ_3 … θ_K

Generative Process

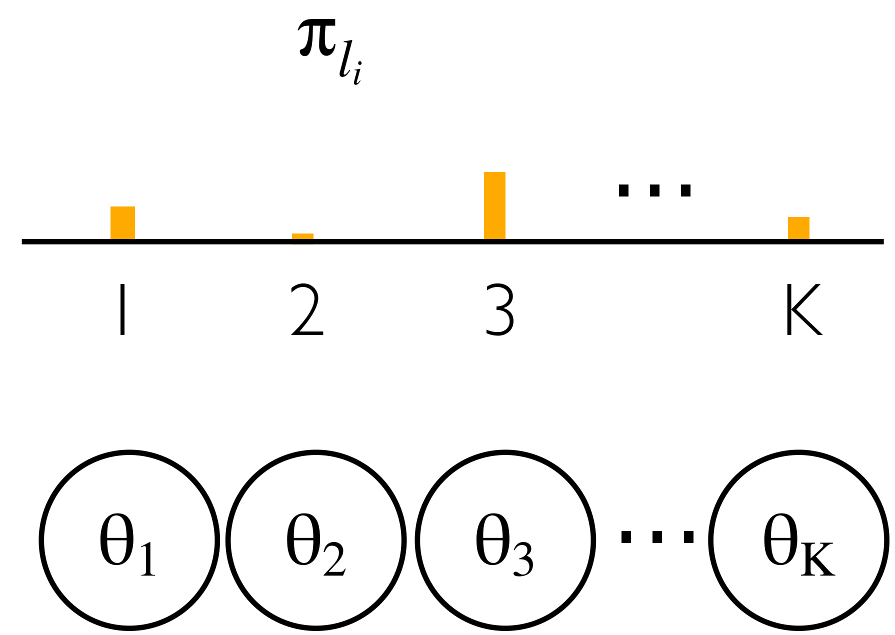
- Step 1
  - Generate the number of phones that each letter maps to (n_i)

l_i red sox

n_i

$$n_i \sim \phi_{l_i} \sim Dir(\eta)$$

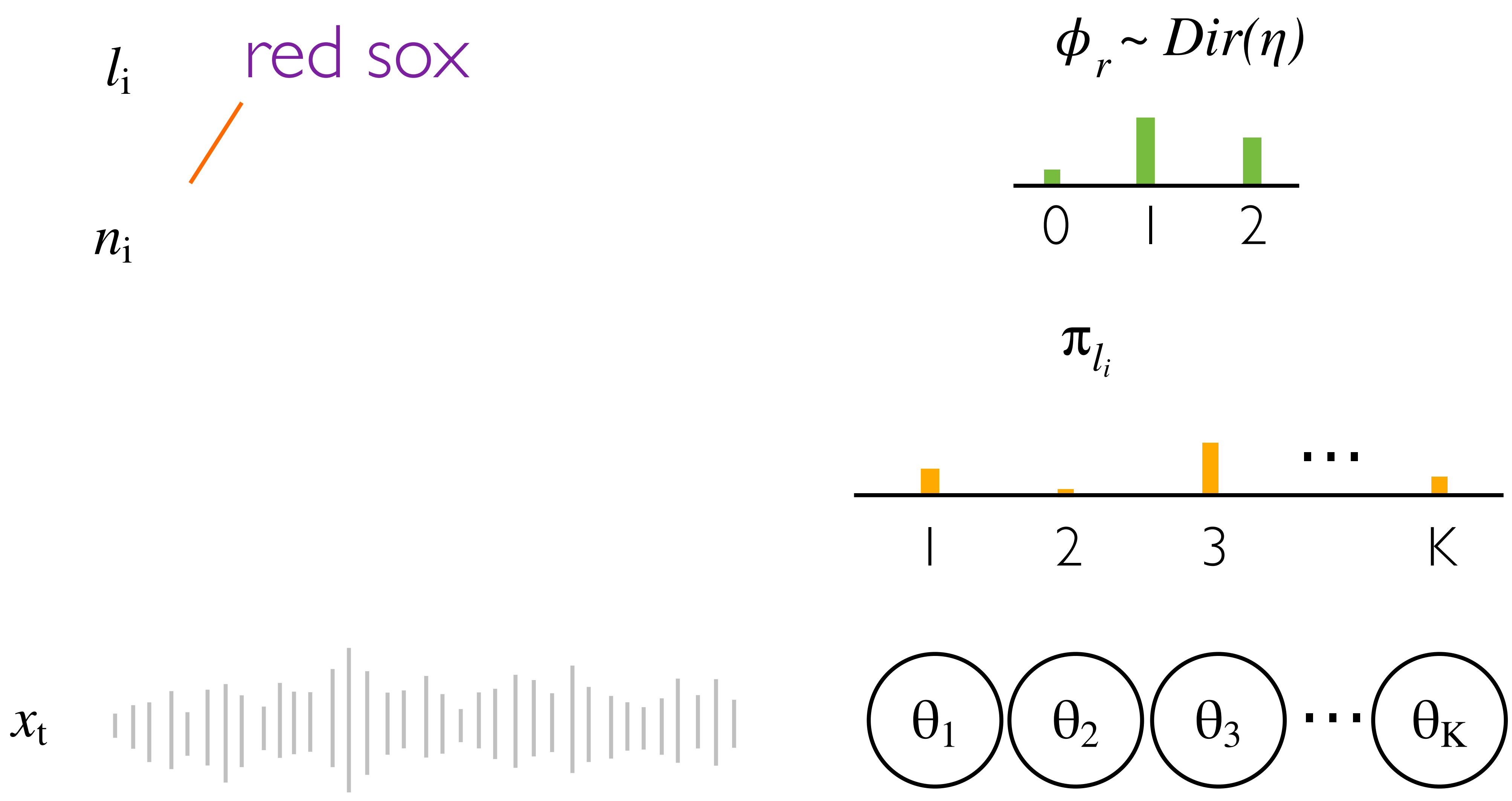
3-dim categorical distribution



x_t

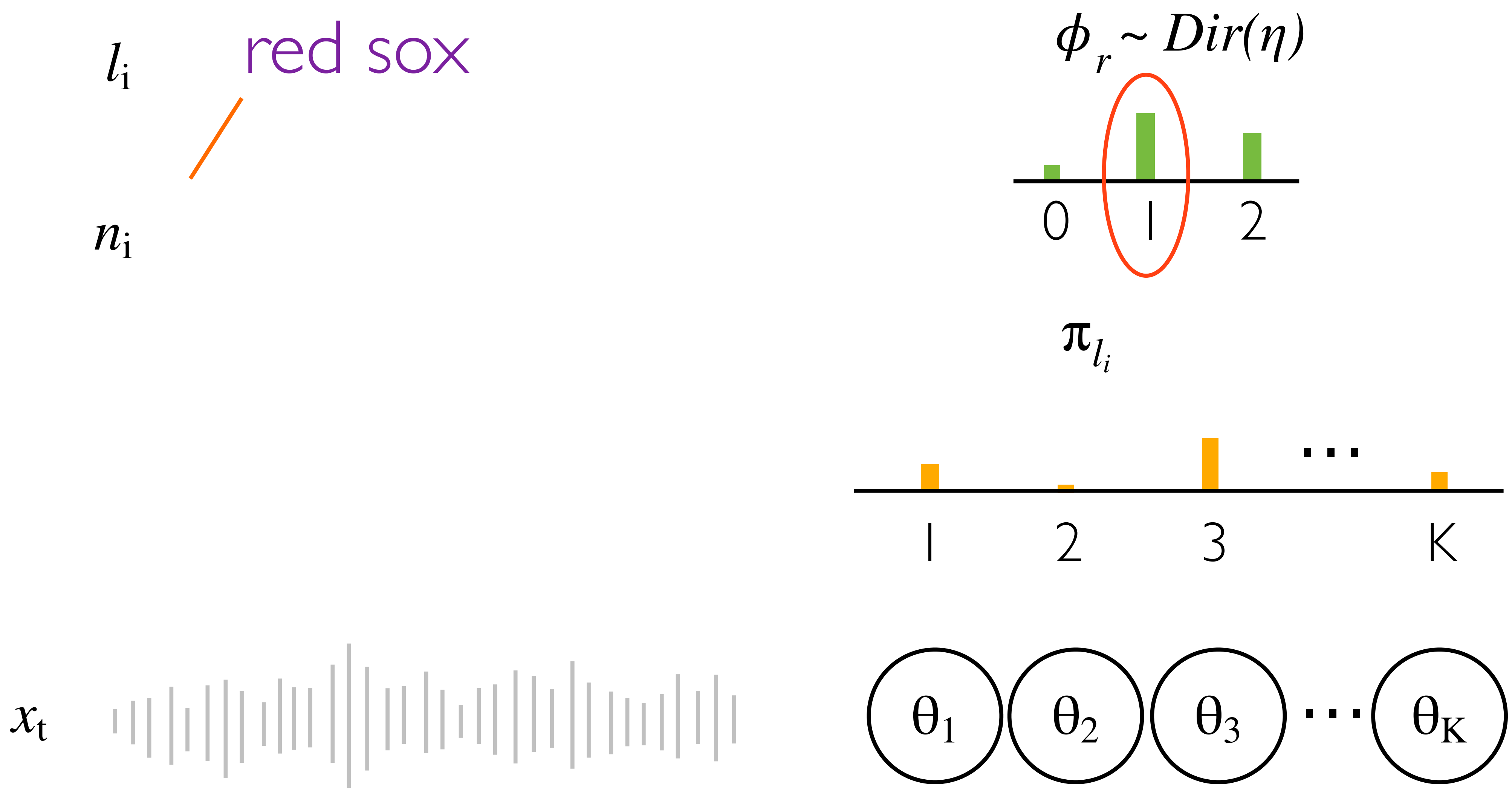
Generative Process

- Step 1
  - Generate the number of phones that each letter maps to (n_i)



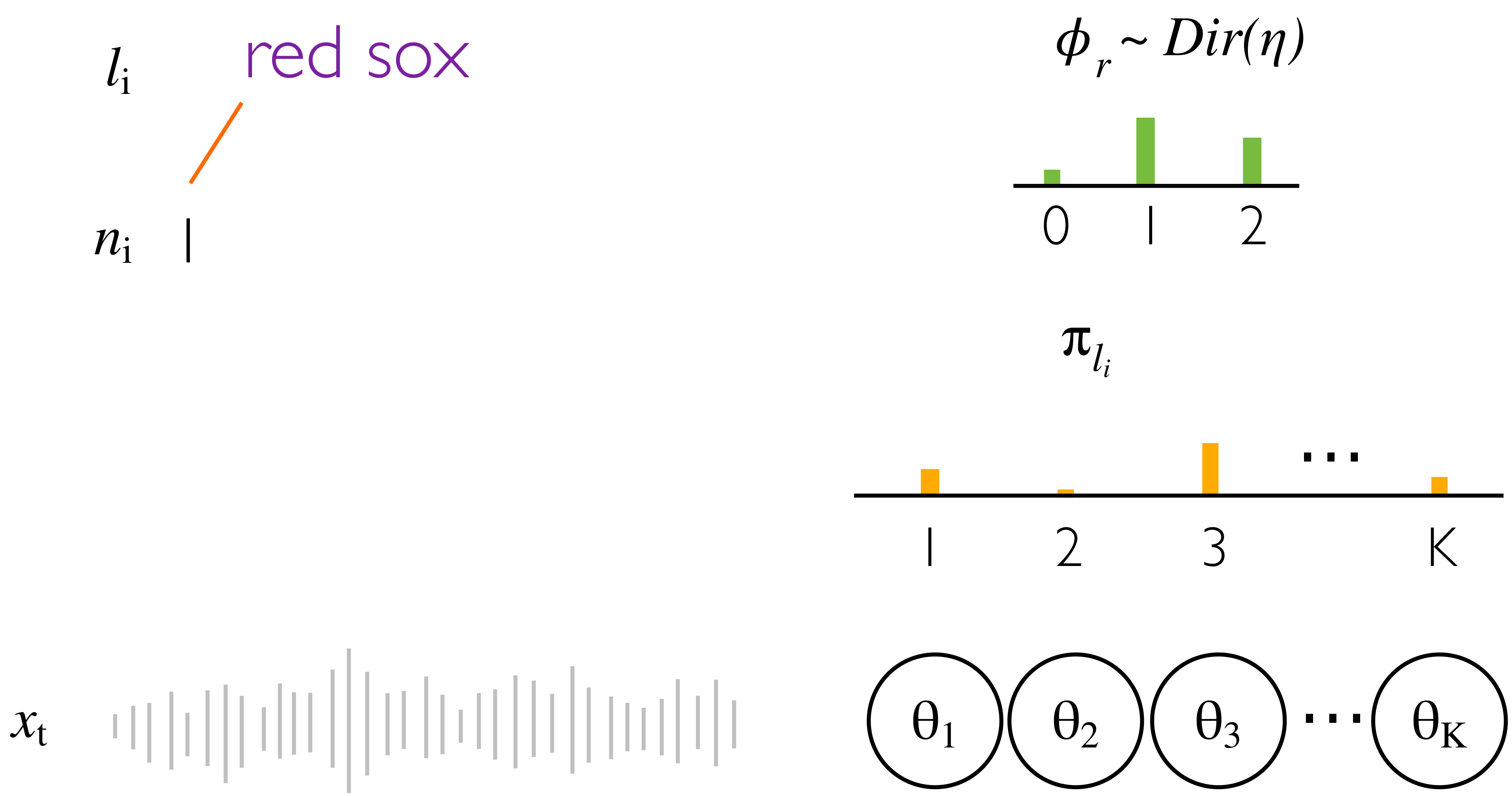
Generative Process

- Step 1
  - Generate the number of phones that each letter maps to (n_i)



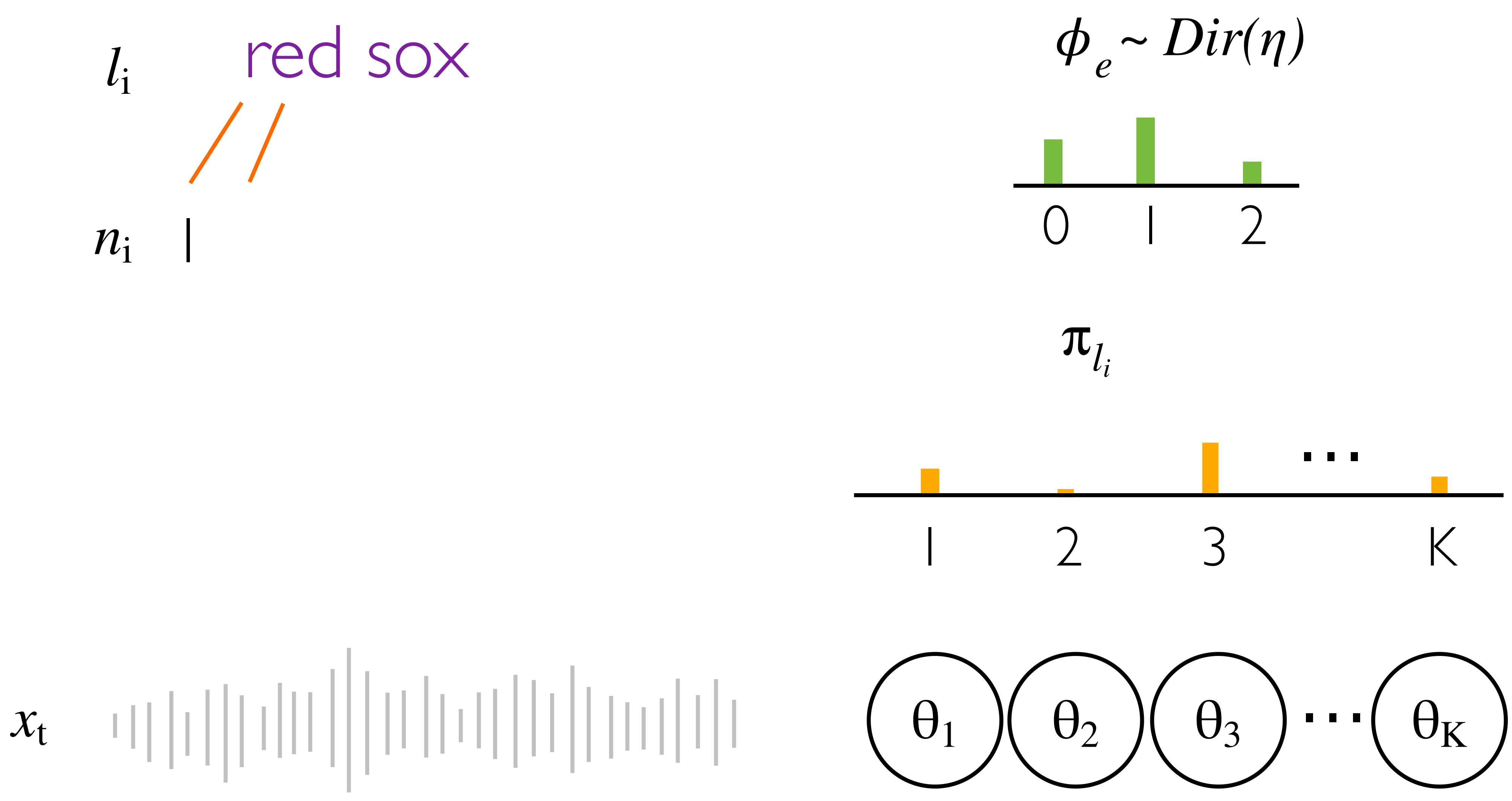
Generative Process

- Step 1
  - Generate the number of phones that each letter maps to (n_i)



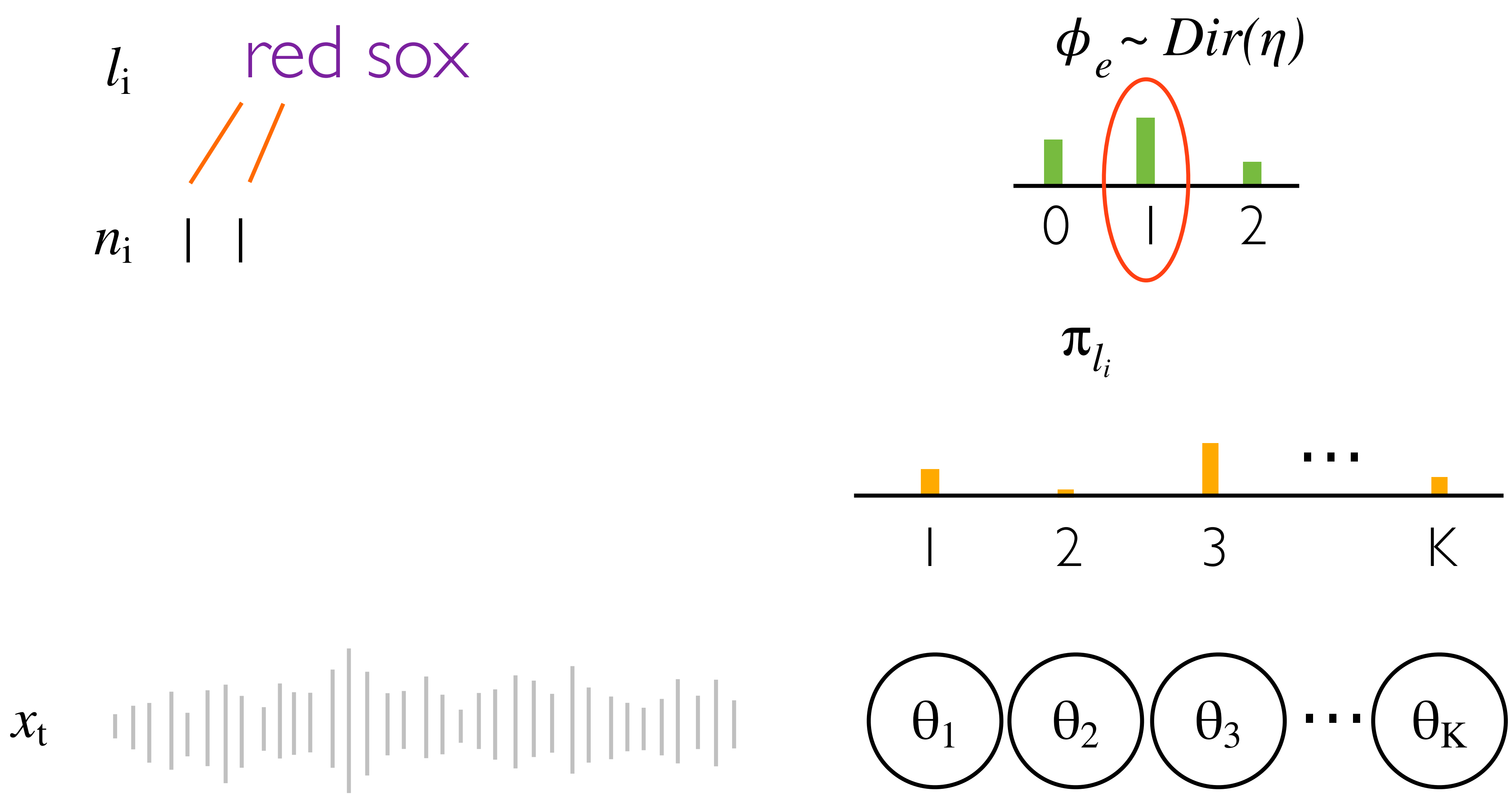
Generative Process

- Step 1
  - Generate the number of phones that each letter maps to (n_i)



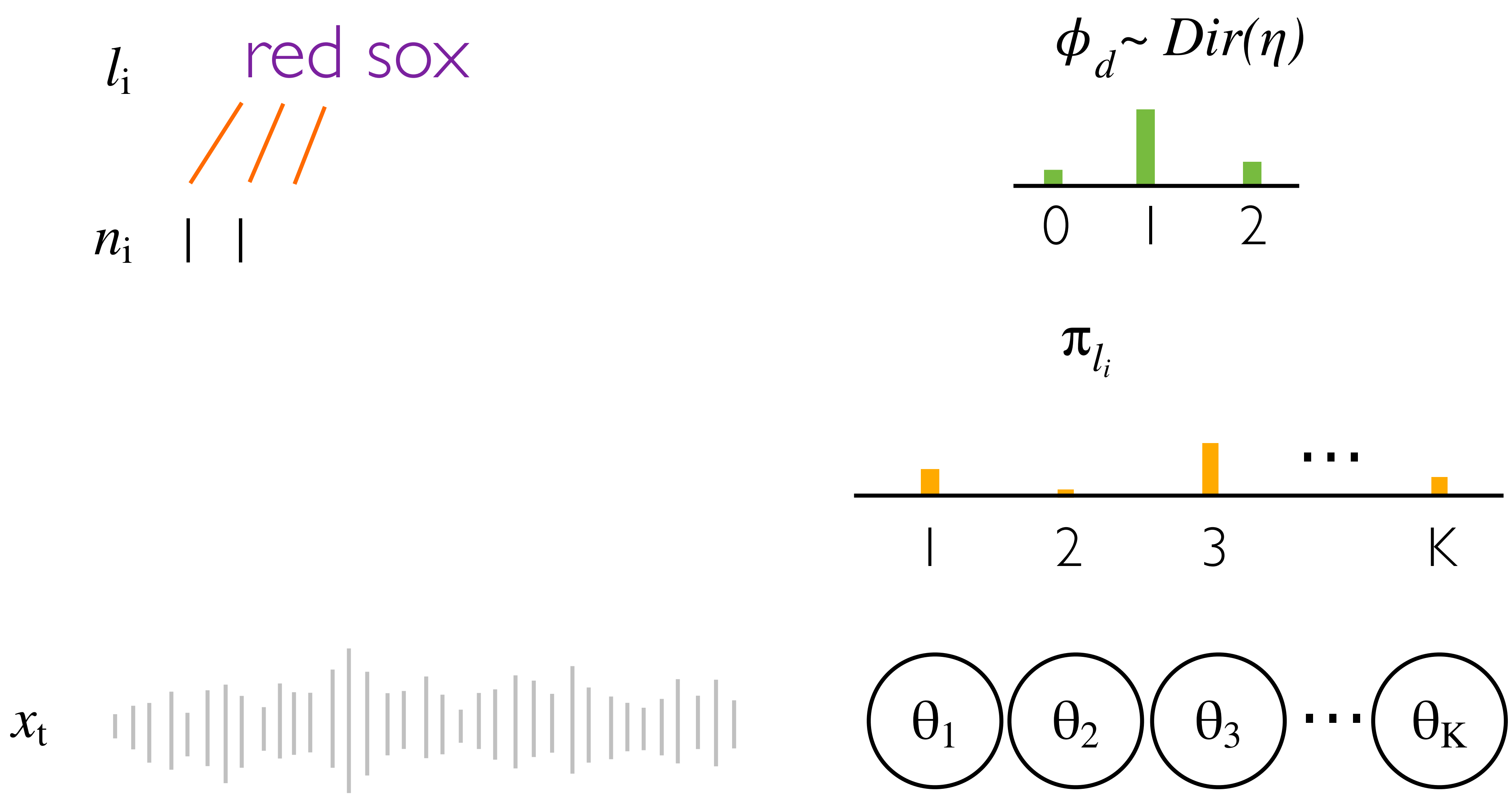
Generative Process

- Step 1
  - Generate the number of phones that each letter maps to (n_i)



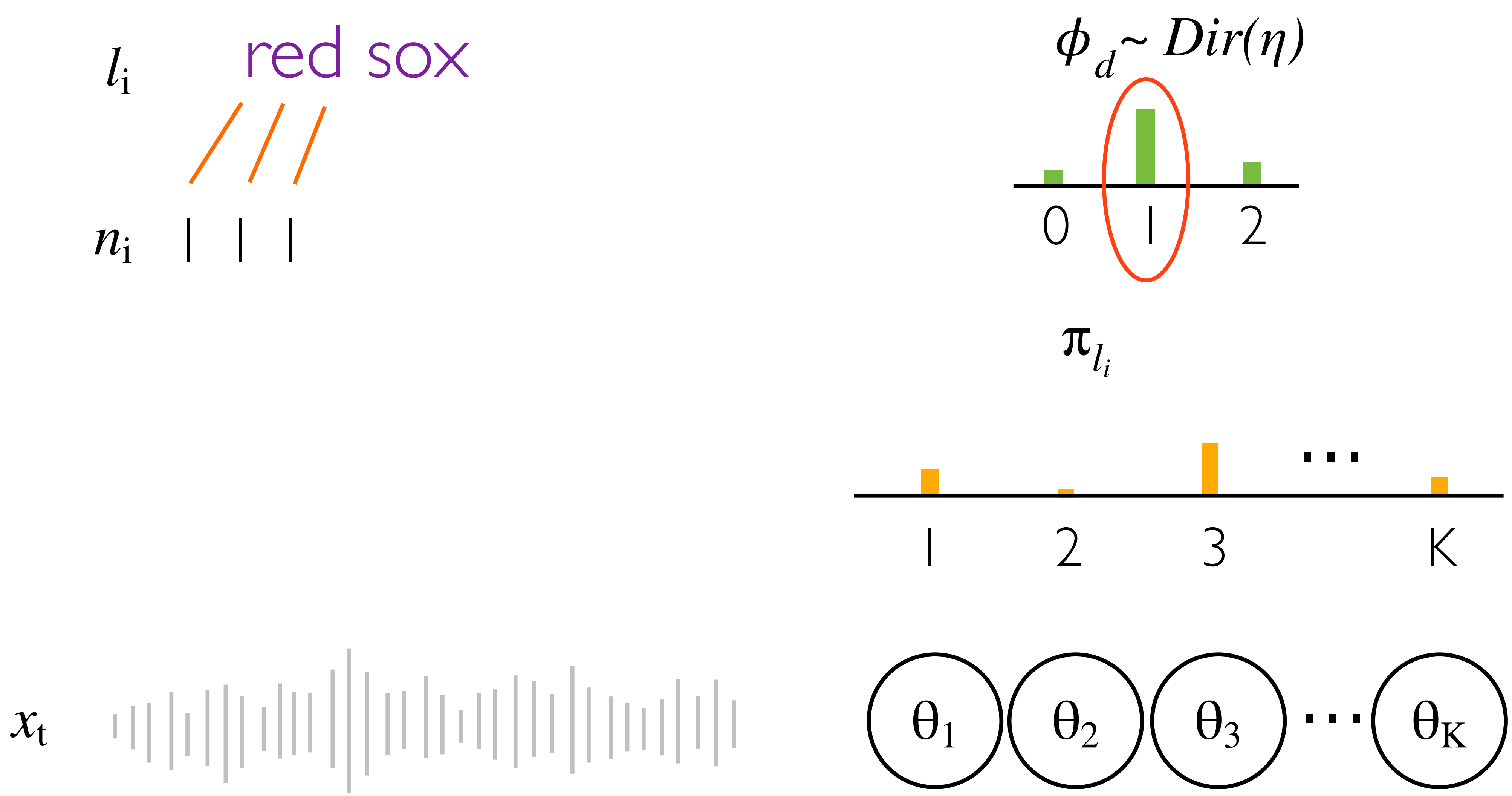
Generative Process

- Step 1
  - Generate the number of phones that each letter maps to (n_i)



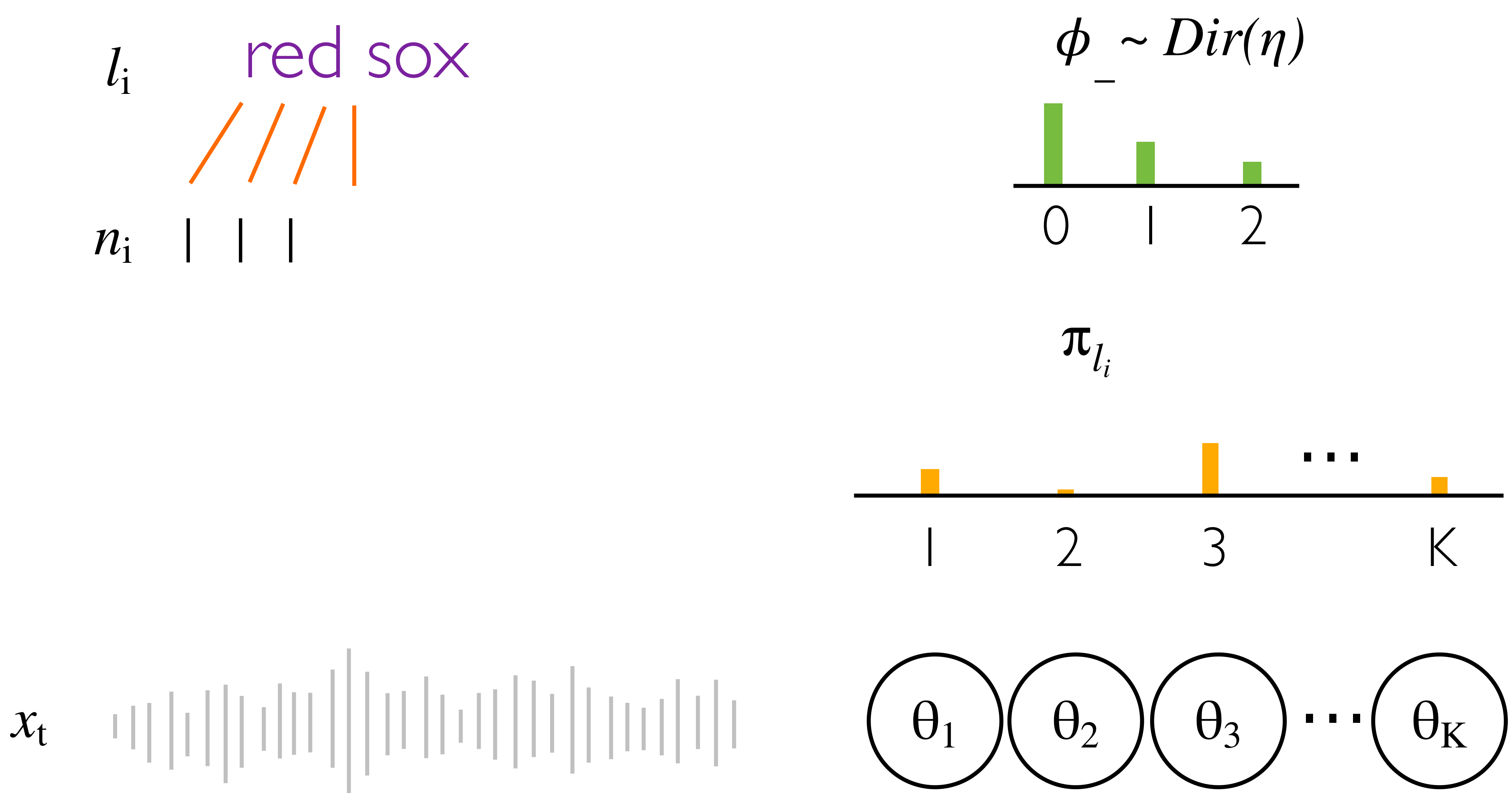
Generative Process

- Step 1
  - Generate the number of phones that each letter maps to (n_i)



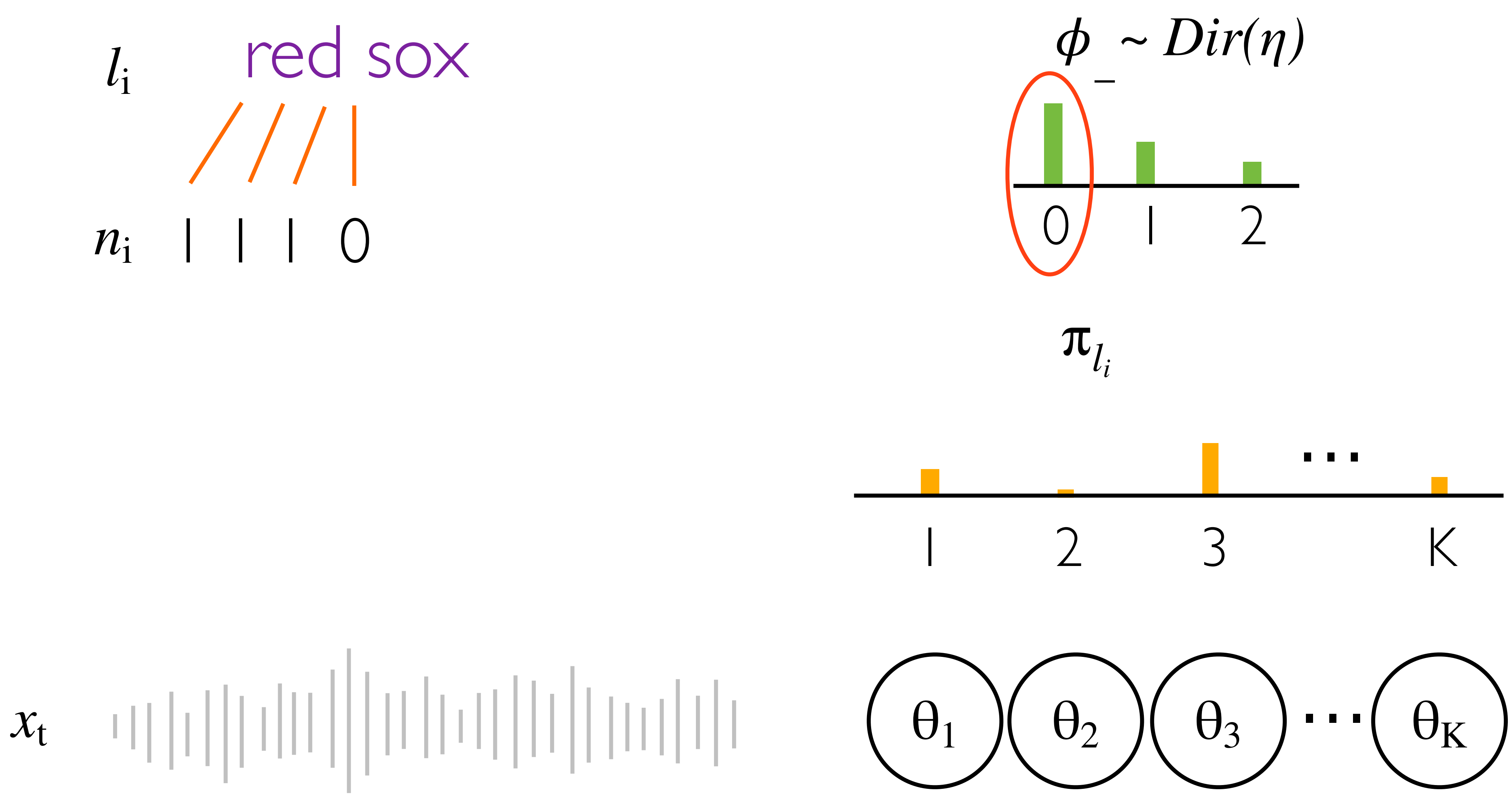
Generative Process

- Step 1
  - Generate the number of phones that each letter maps to (n_i)



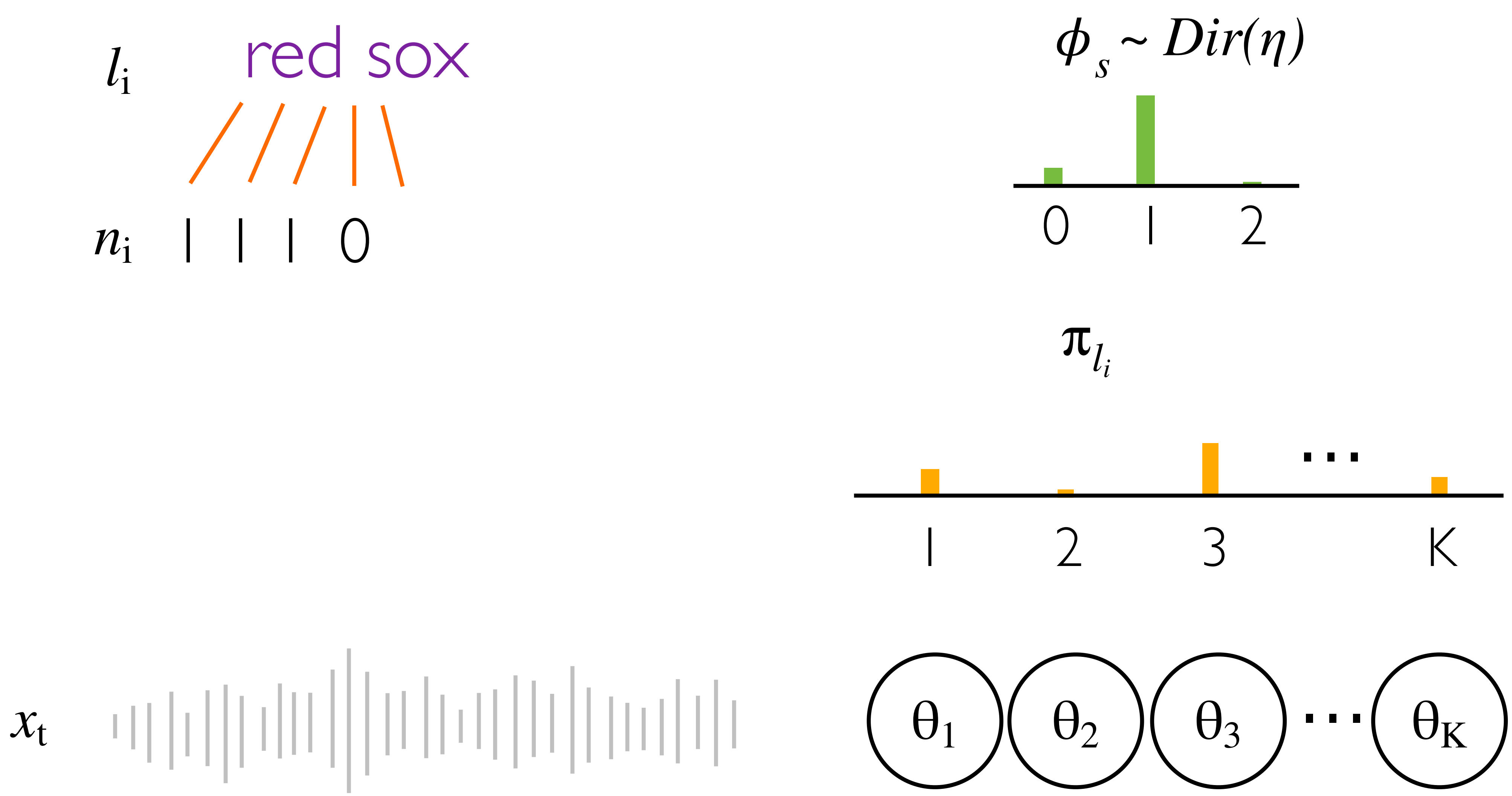
Generative Process

- Step 1
  - Generate the number of phones that each letter maps to (n_i)



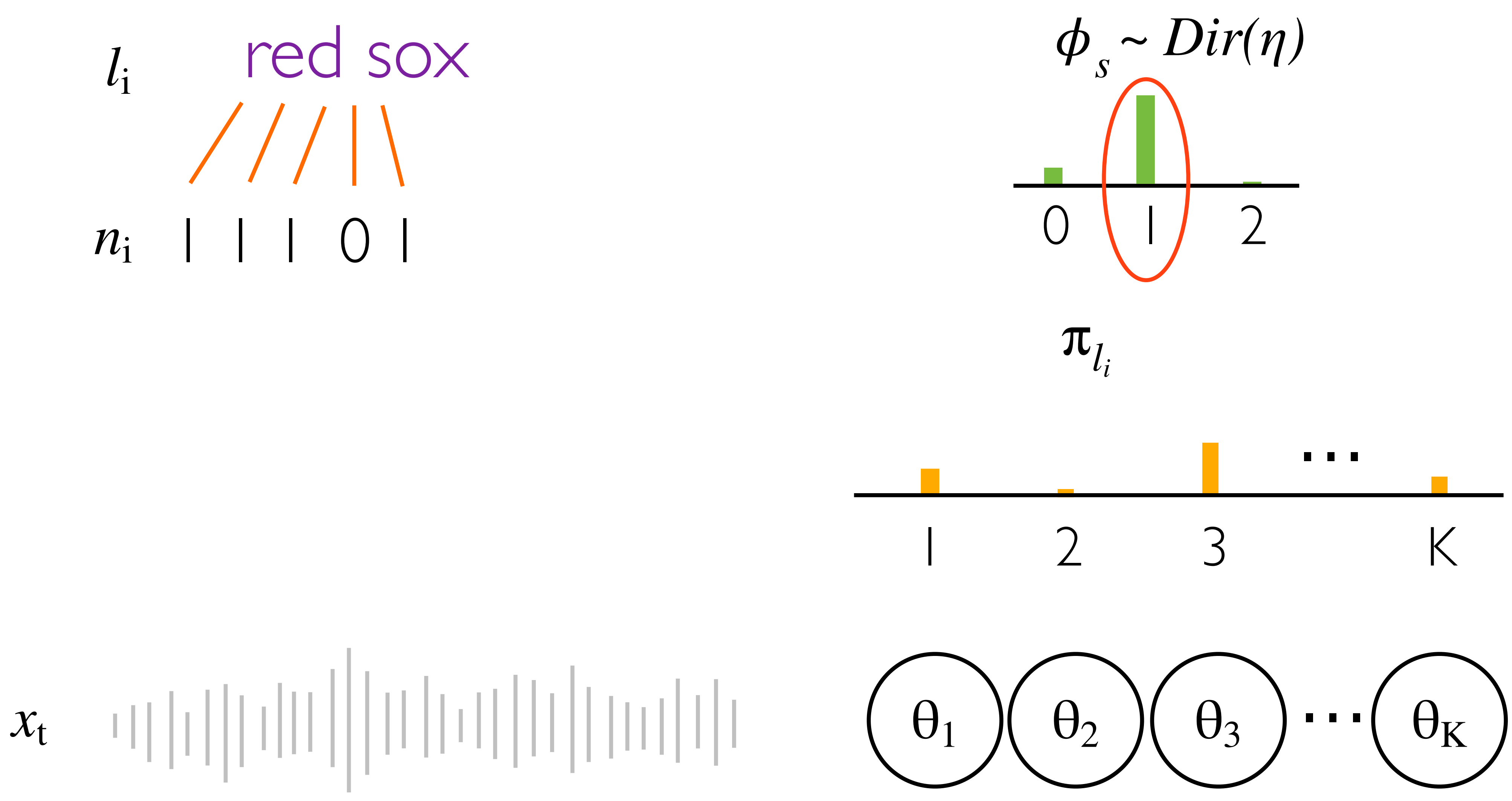
Generative Process

- Step 1
  - Generate the number of phones that each letter maps to (n_i)



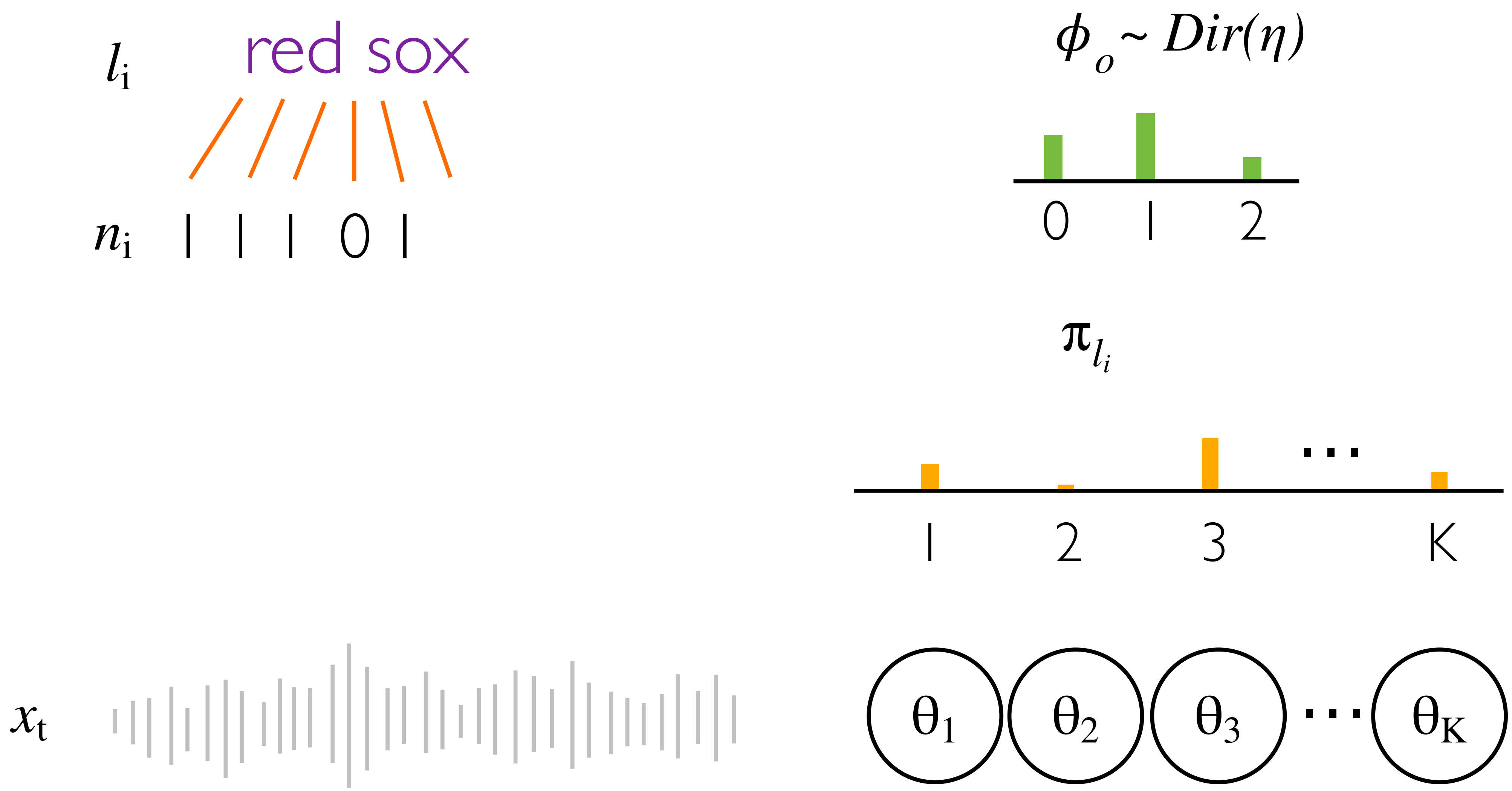
Generative Process

- Step 1
  - Generate the number of phones that each letter maps to (n_i)



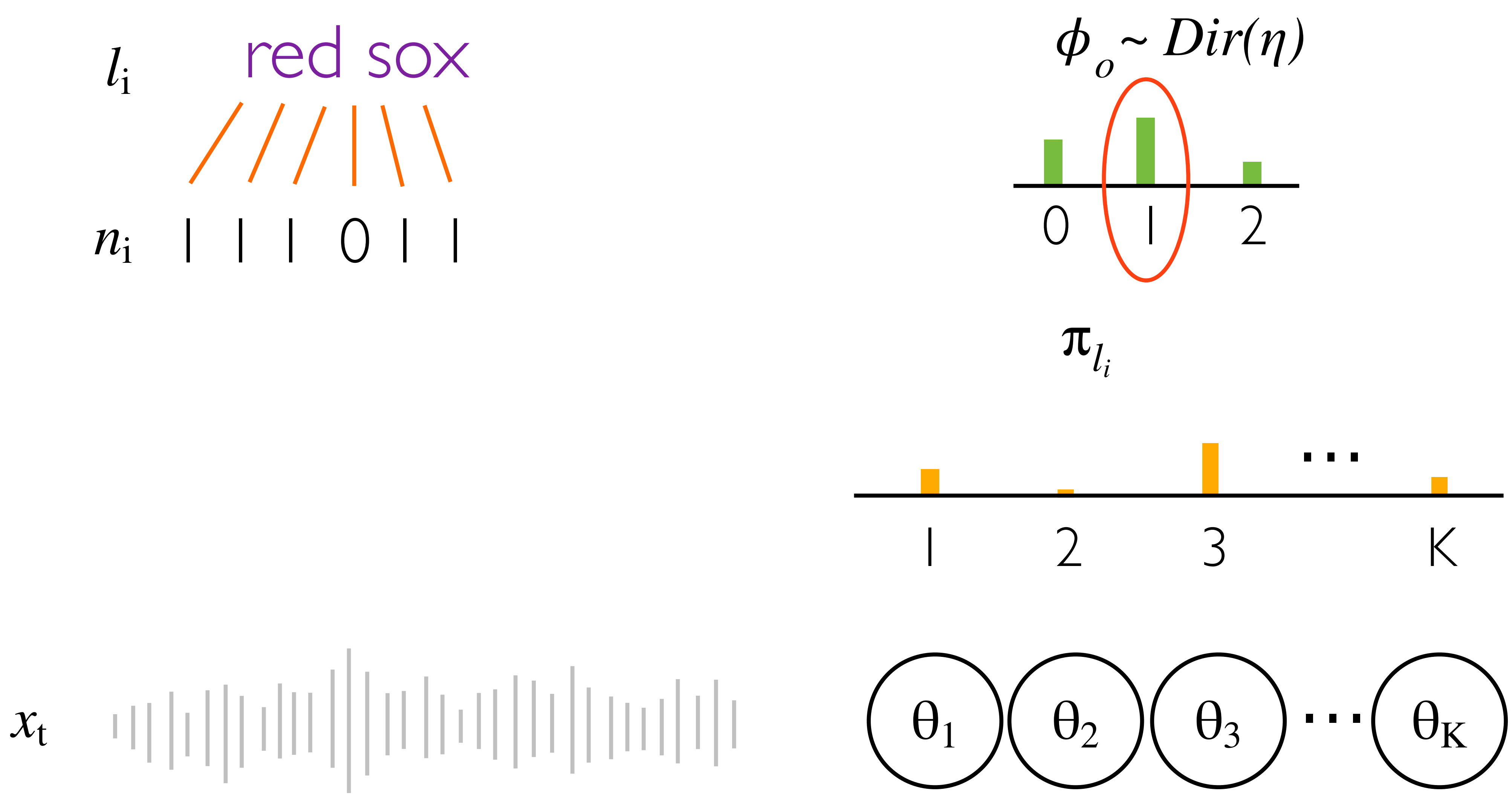
Generative Process

- Step 1
  - Generate the number of phones that each letter maps to (n_i)



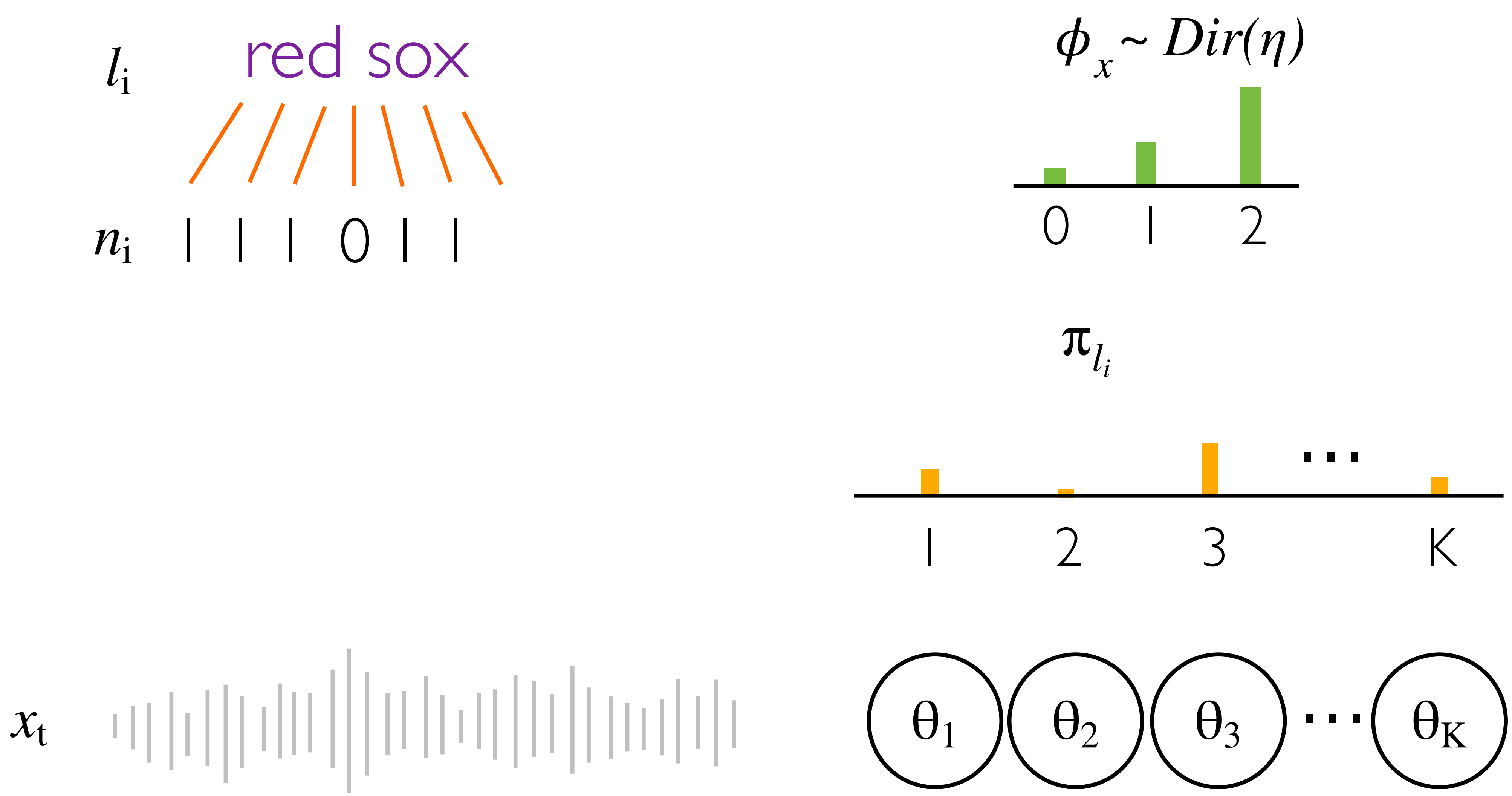
Generative Process

- Step 1
  - Generate the number of phones that each letter maps to (n_i)



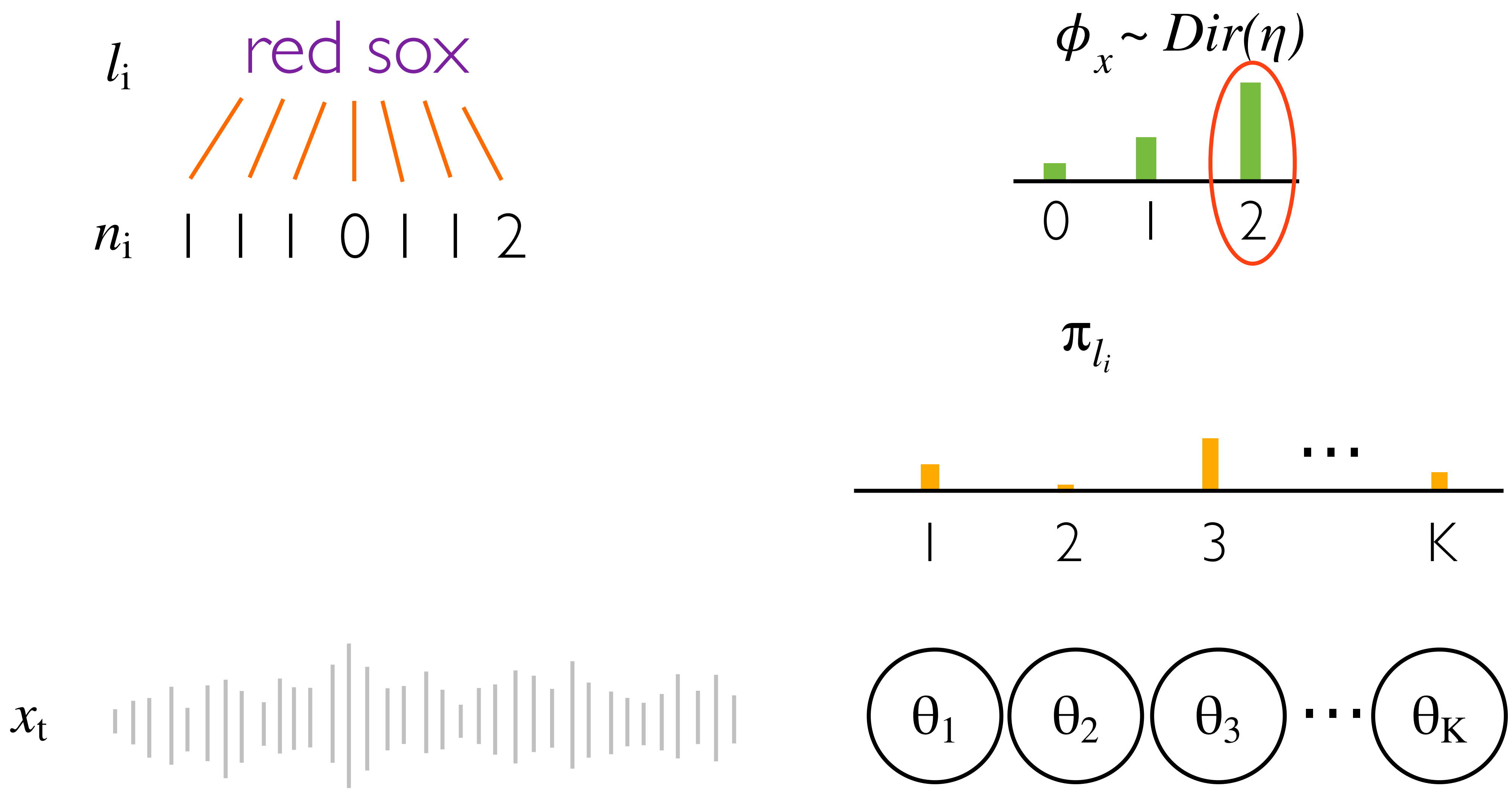
Generative Process

- Step 1
  - Generate the number of phones that each letter maps to (n_i)



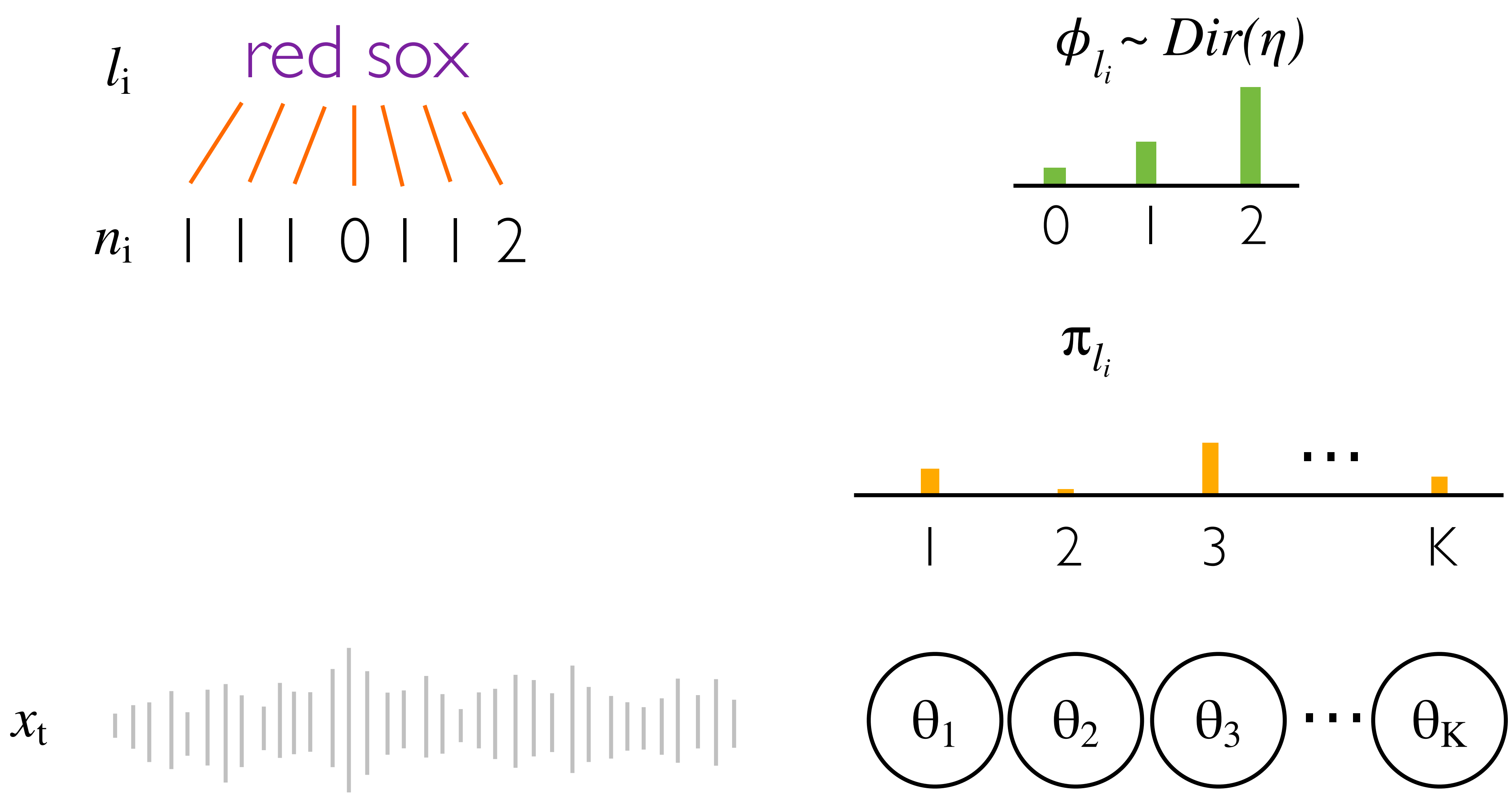
Generative Process

- Step 1
  - Generate the number of phones that each letter maps to (n_i)



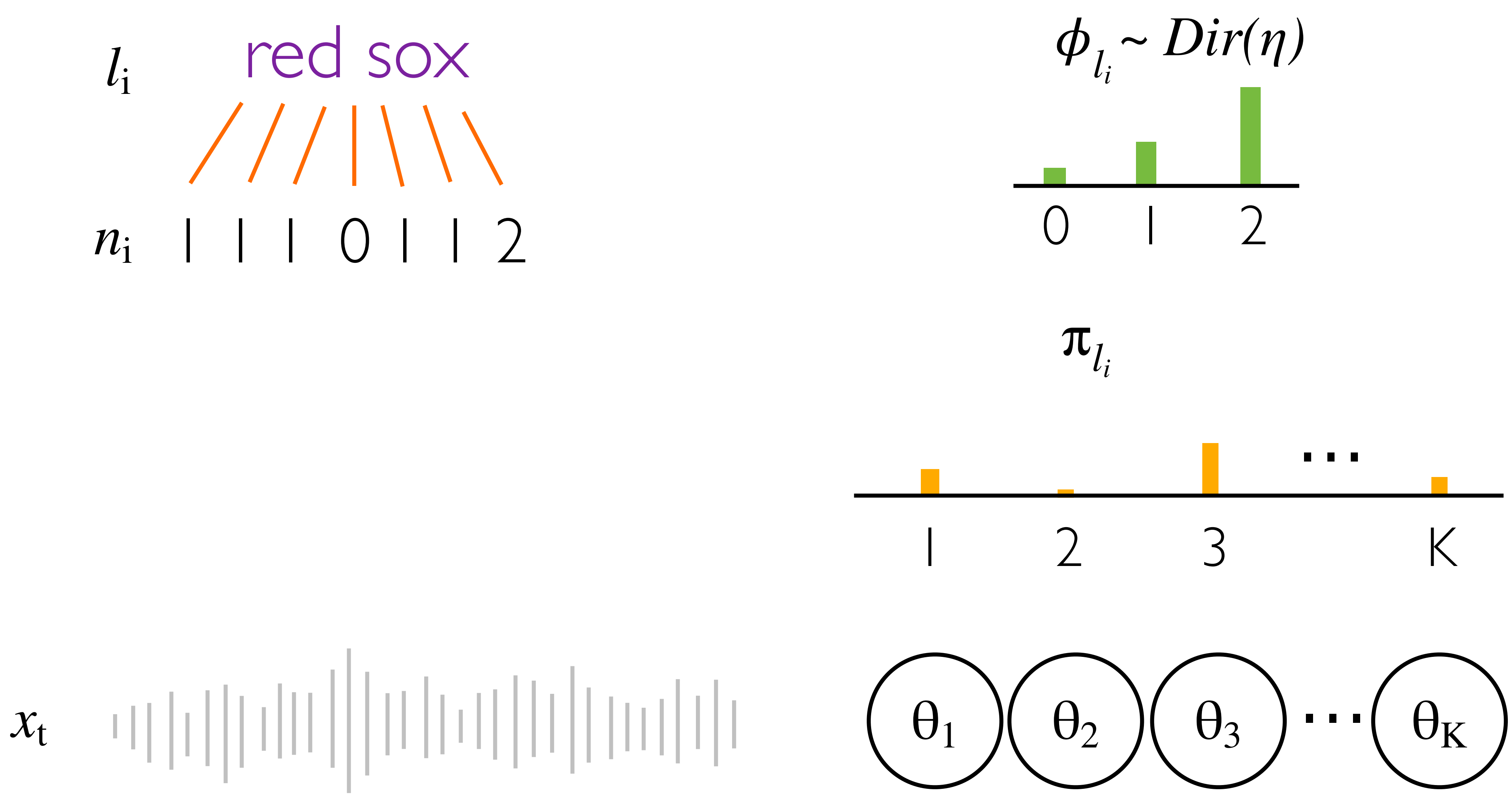
Generative Process

- Step 1
  - Generate the number of phones that each letter maps to (n_i)



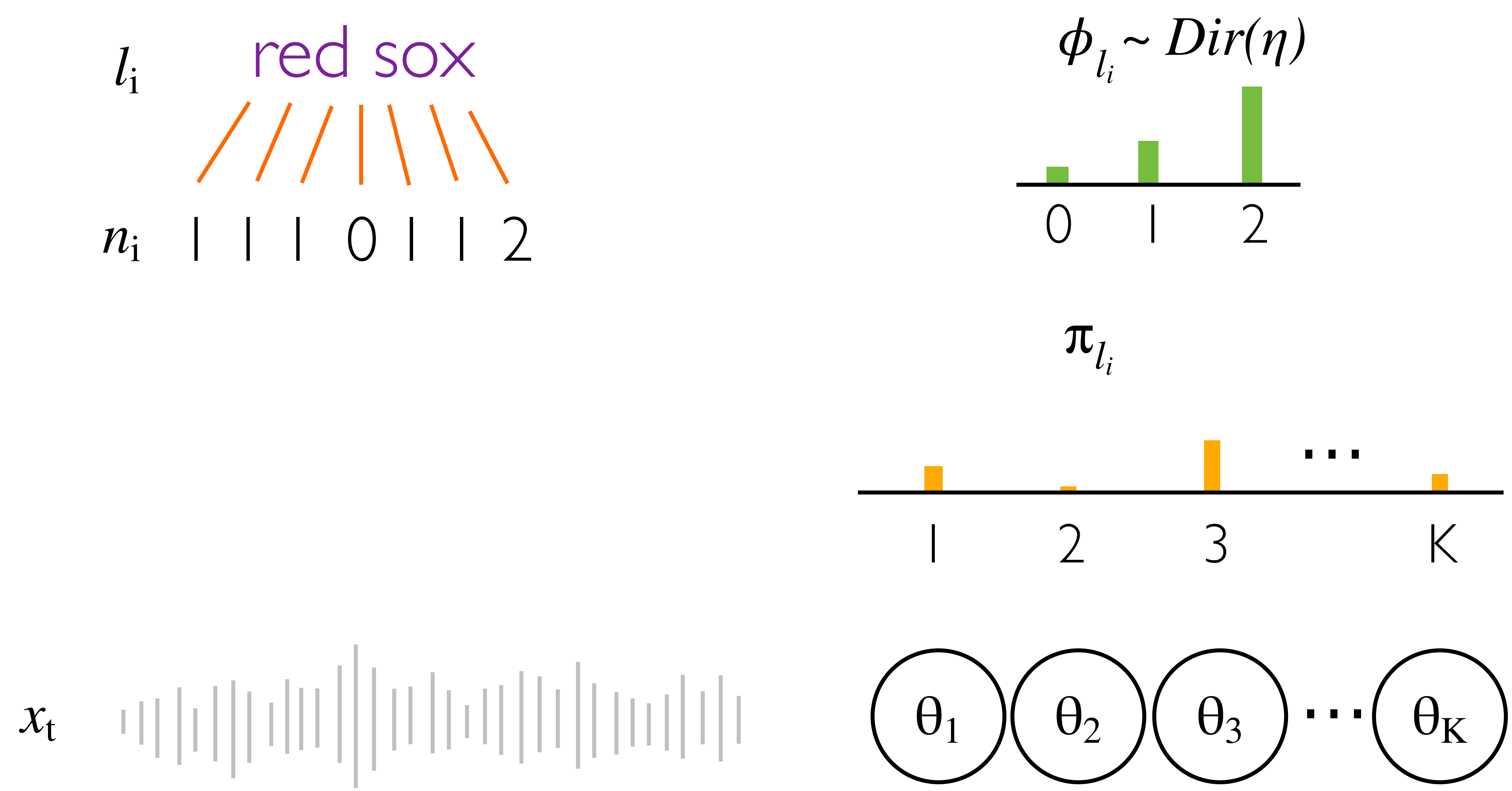
Generative Process

- Step 2



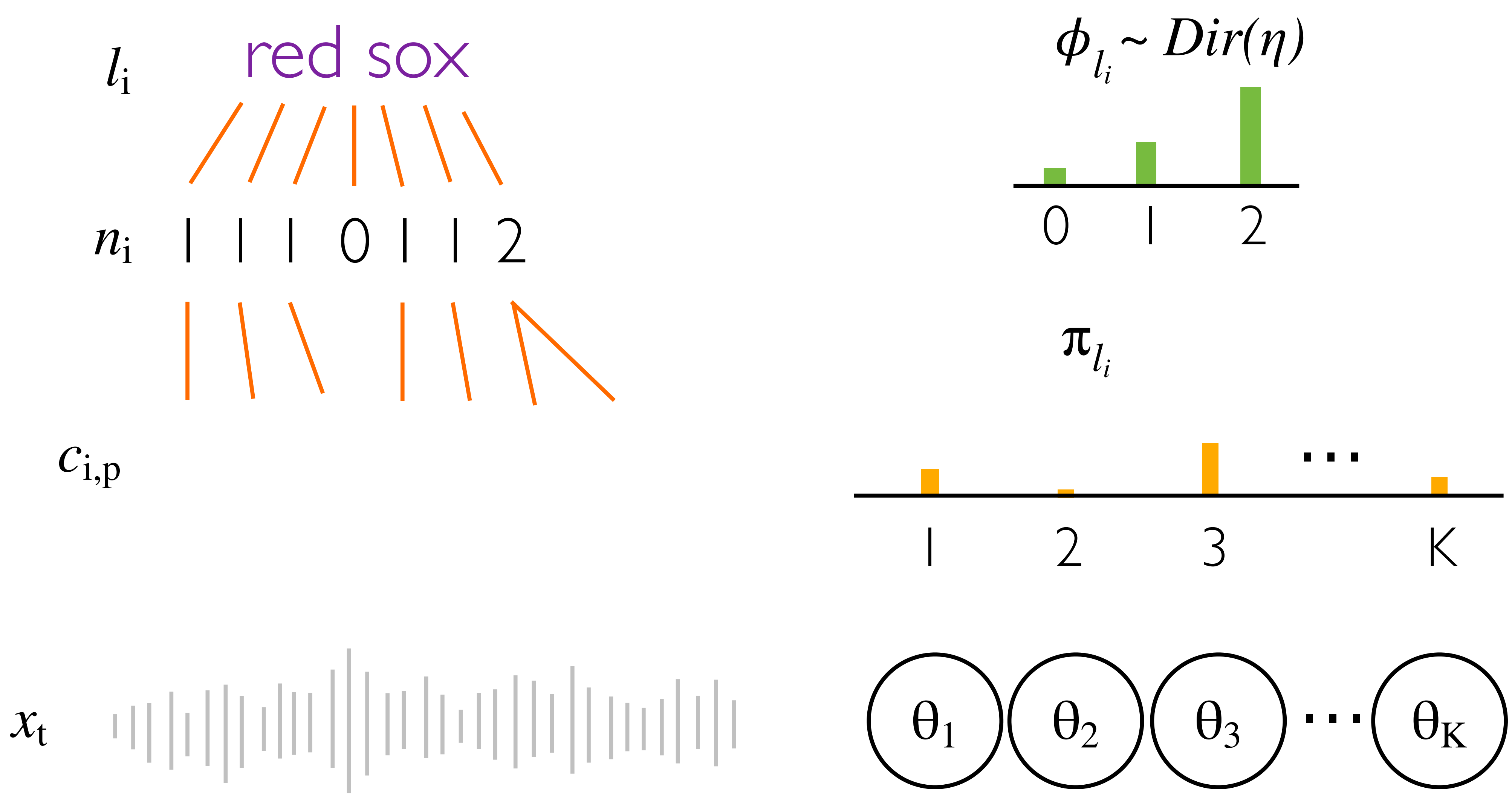
Generative Process

- Step 2
  - Generate the phone label ($c_{i,p}$) for every phone that a letter maps to, $1 \leq p \leq n_i$



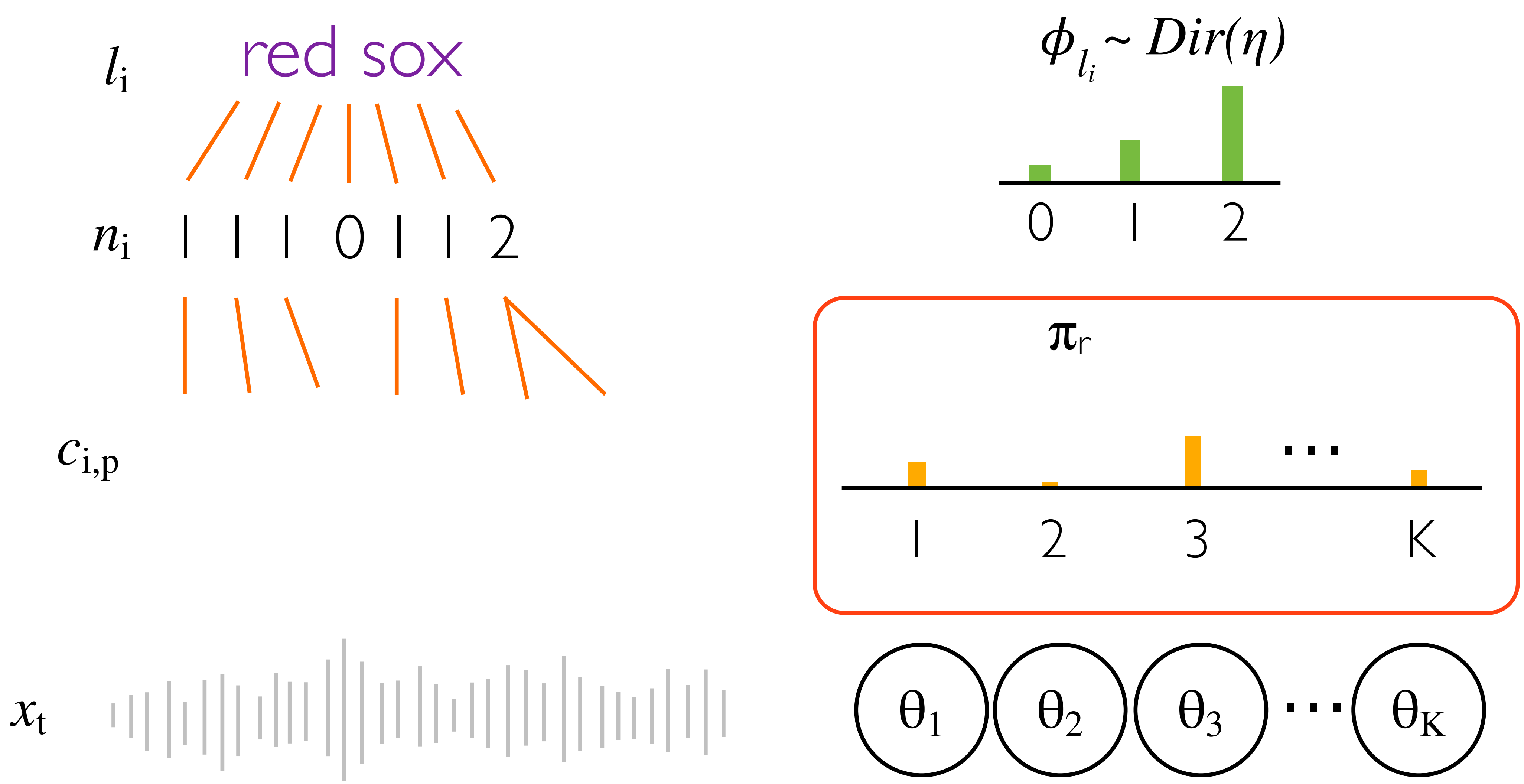
Generative Process

- Step 2
  - Generate the phone label ($c_{i,p}$) for every phone that a letter maps to, $1 \le p \le n_i$



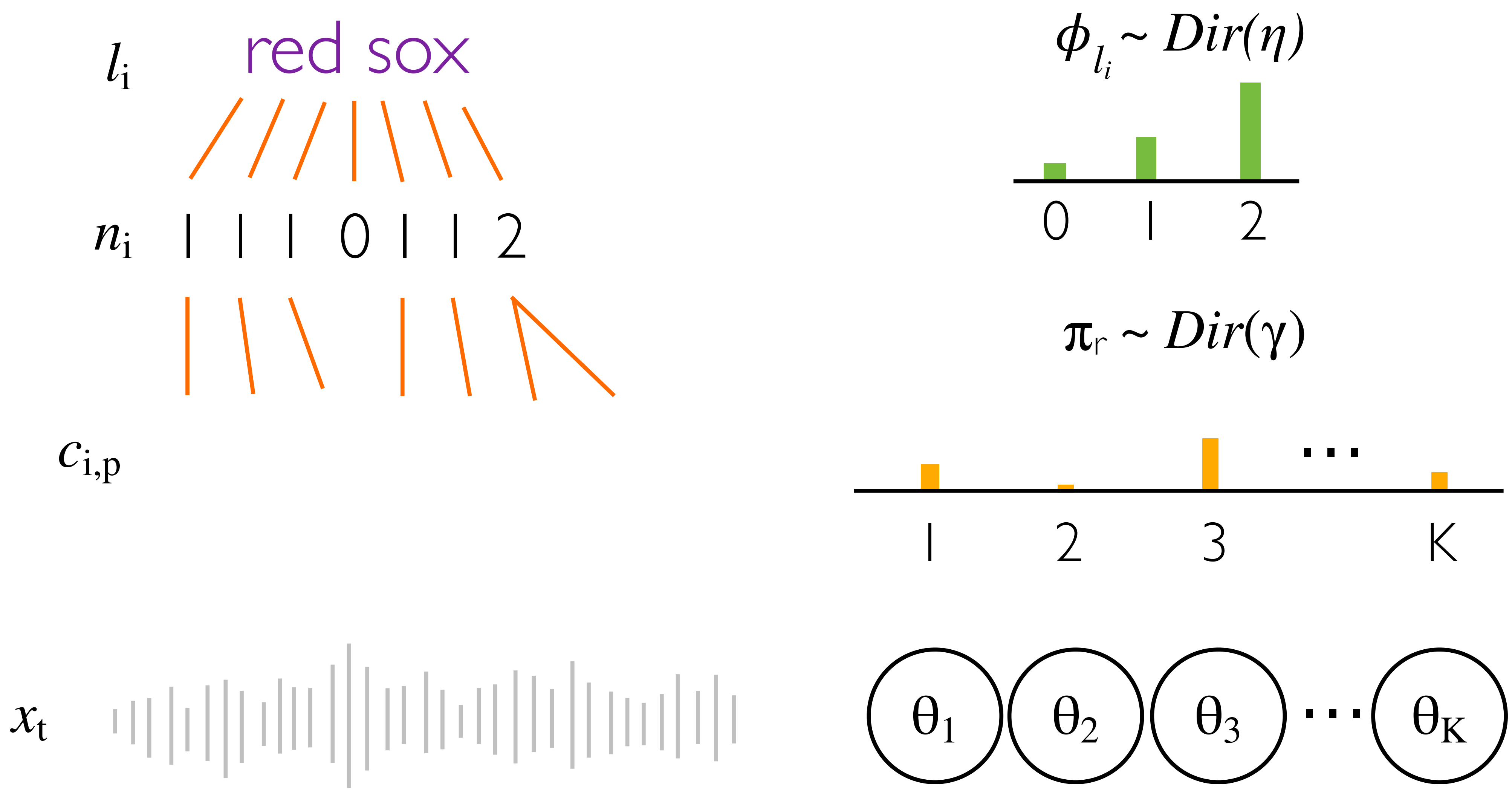
Generative Process

- Step 2
  - Generate the phone label ($c_{i,p}$) for every phone that a letter maps to, $1 \leq p \leq n_i$



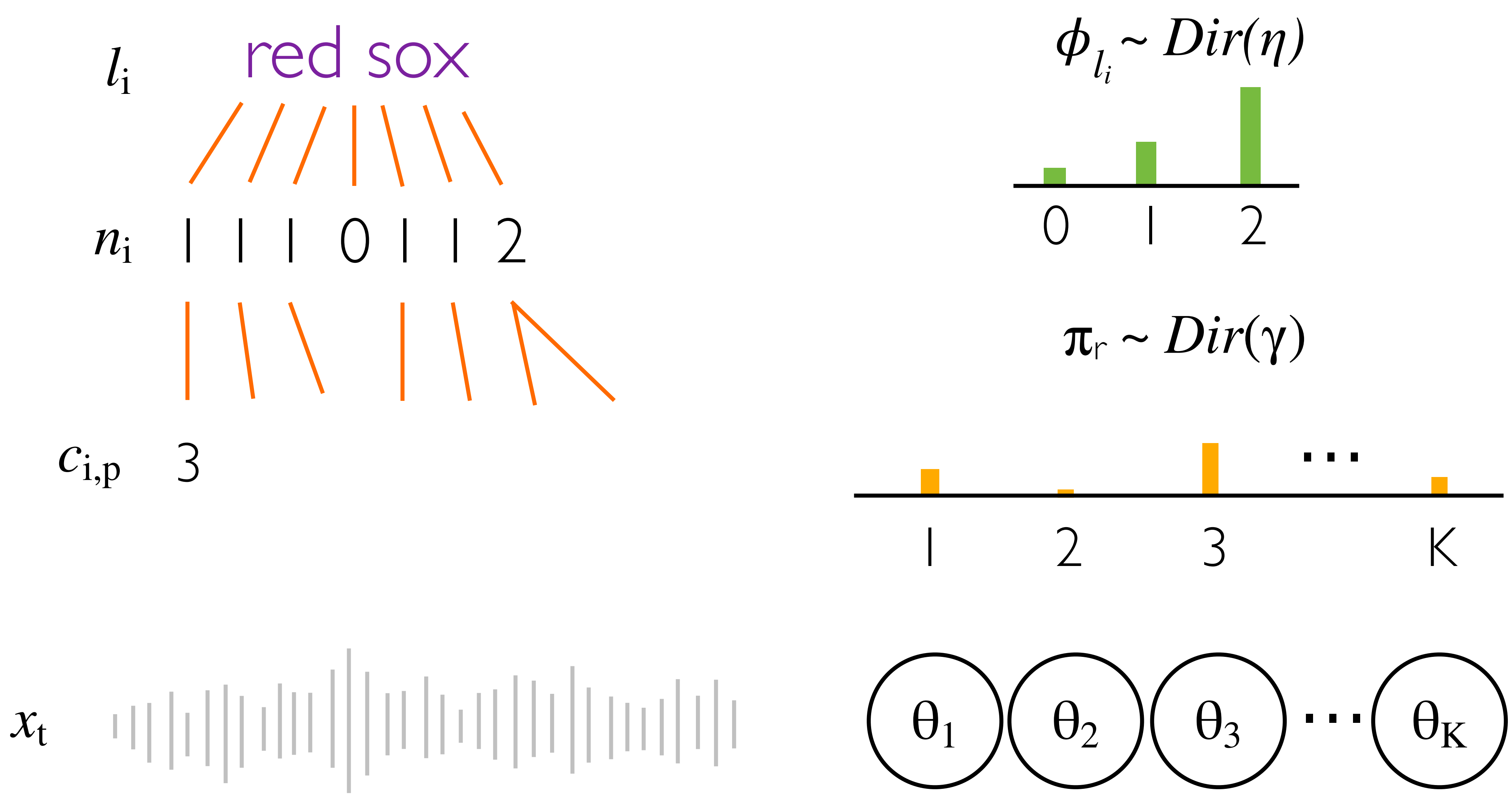
Generative Process

- Step 2
  - Generate the phone label ($c_{i,p}$) for every phone that a letter maps to, $1 \leq p \leq n_i$



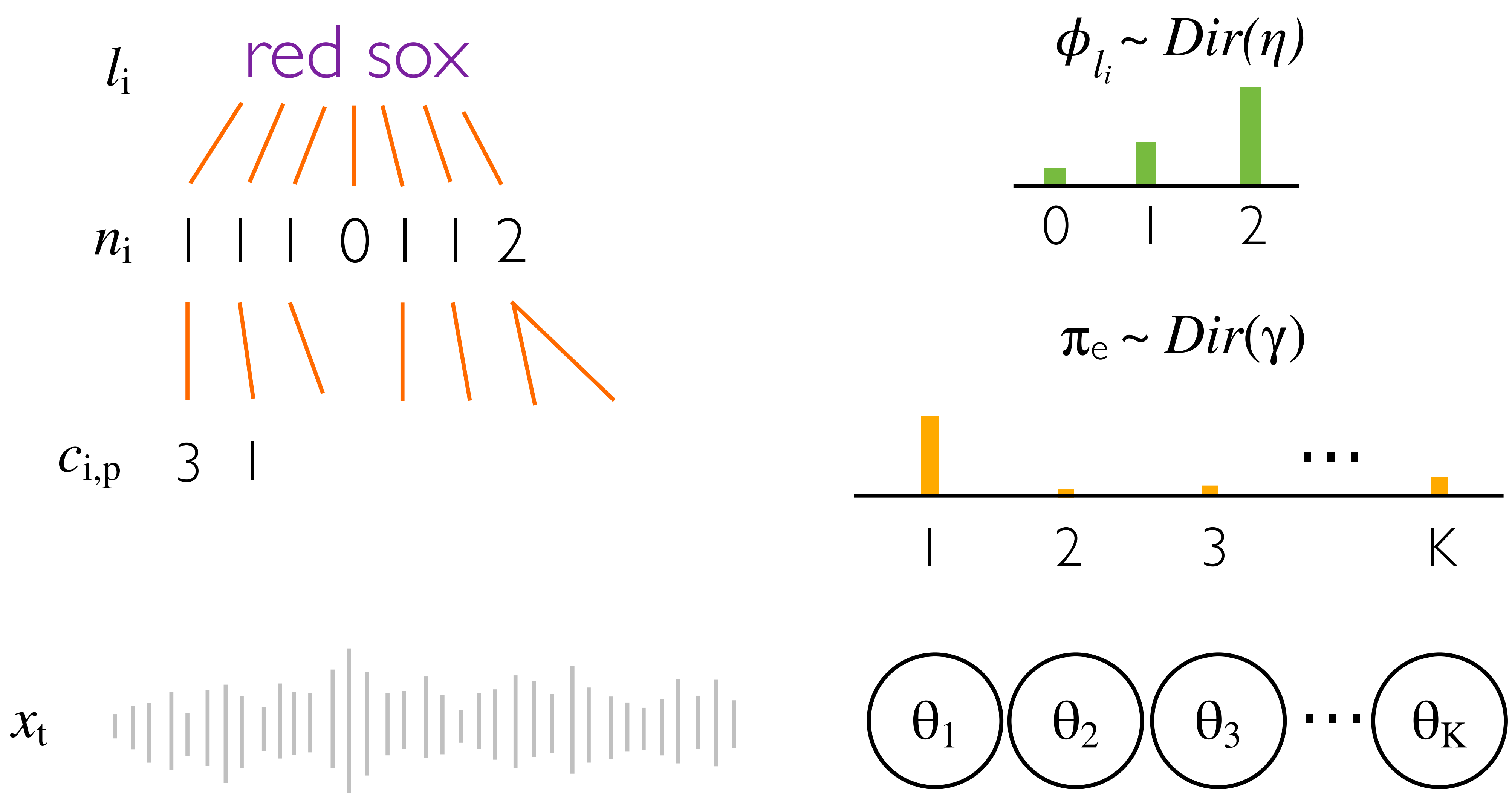
Generative Process

- Step 2
  - Generate the phone label ($c_{i,p}$) for every phone that a letter maps to, $1 \leq p \leq n_i$



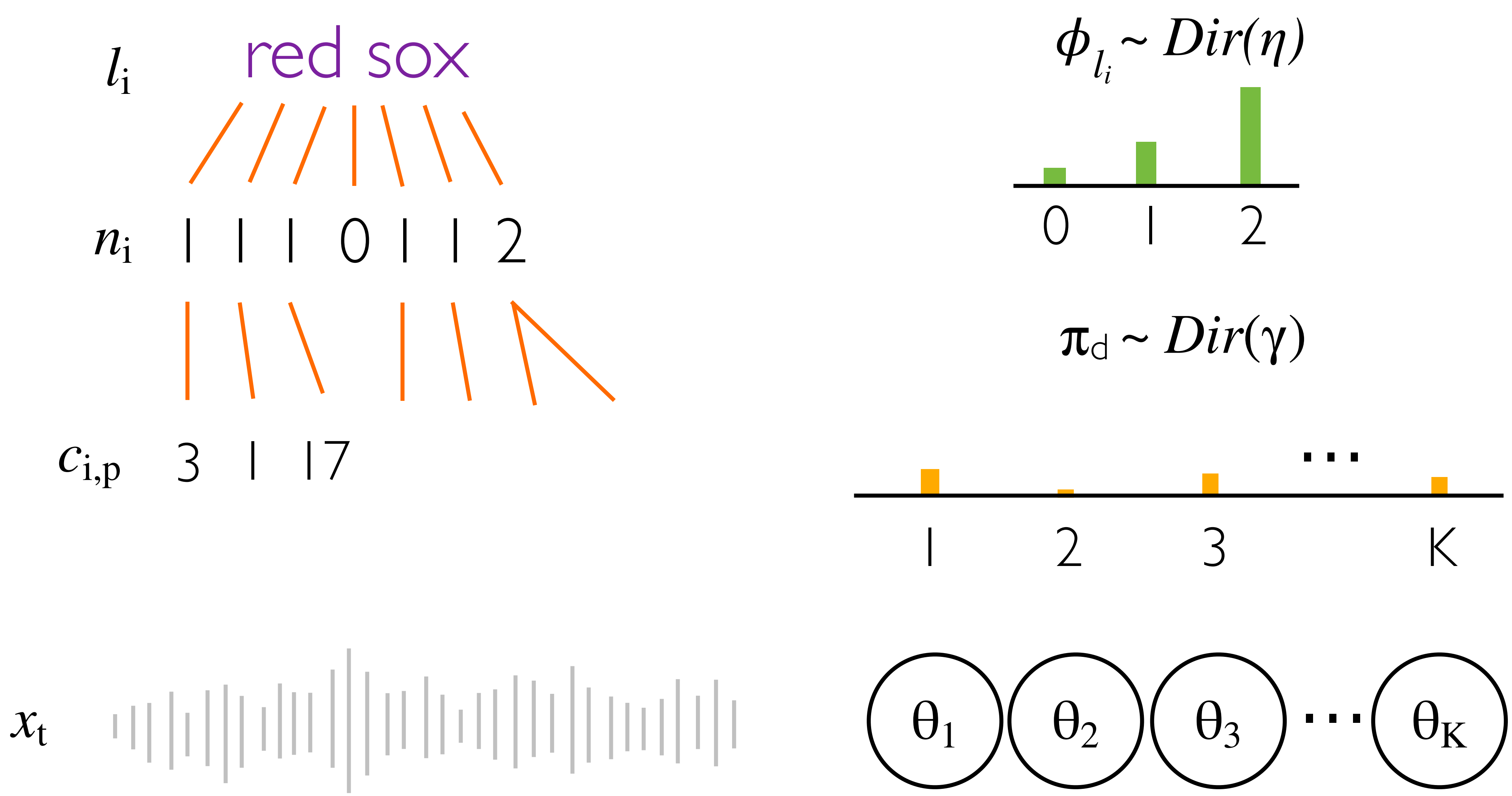
Generative Process

- Step 2
  - Generate the phone label ($c_{i,p}$) for every phone that a letter maps to, $1 \leq p \leq n_i$



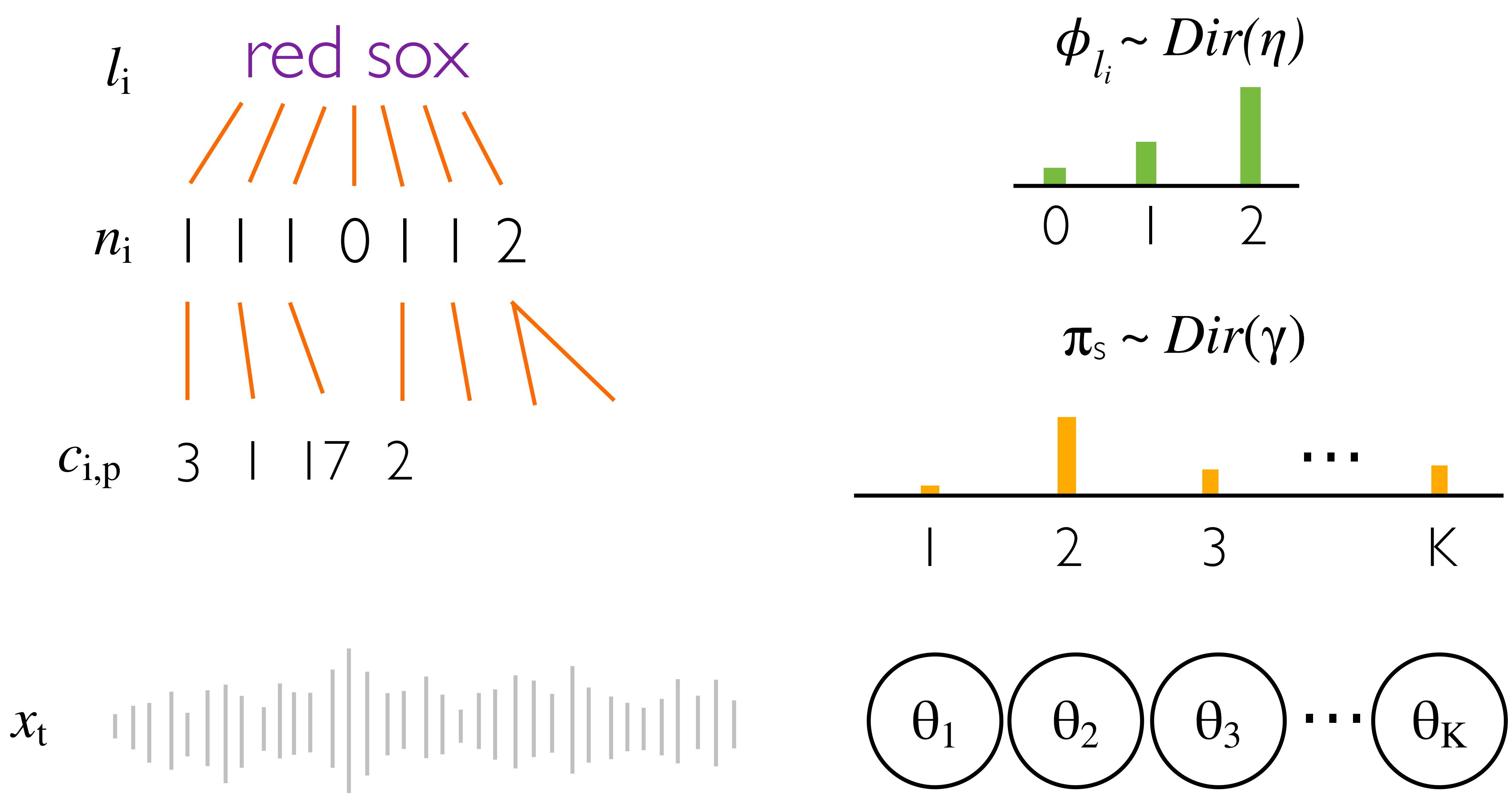
Generative Process

- Step 2
  - Generate the phone label ($c_{i,p}$) for every phone that a letter maps to, $1 \leq p \leq n_i$



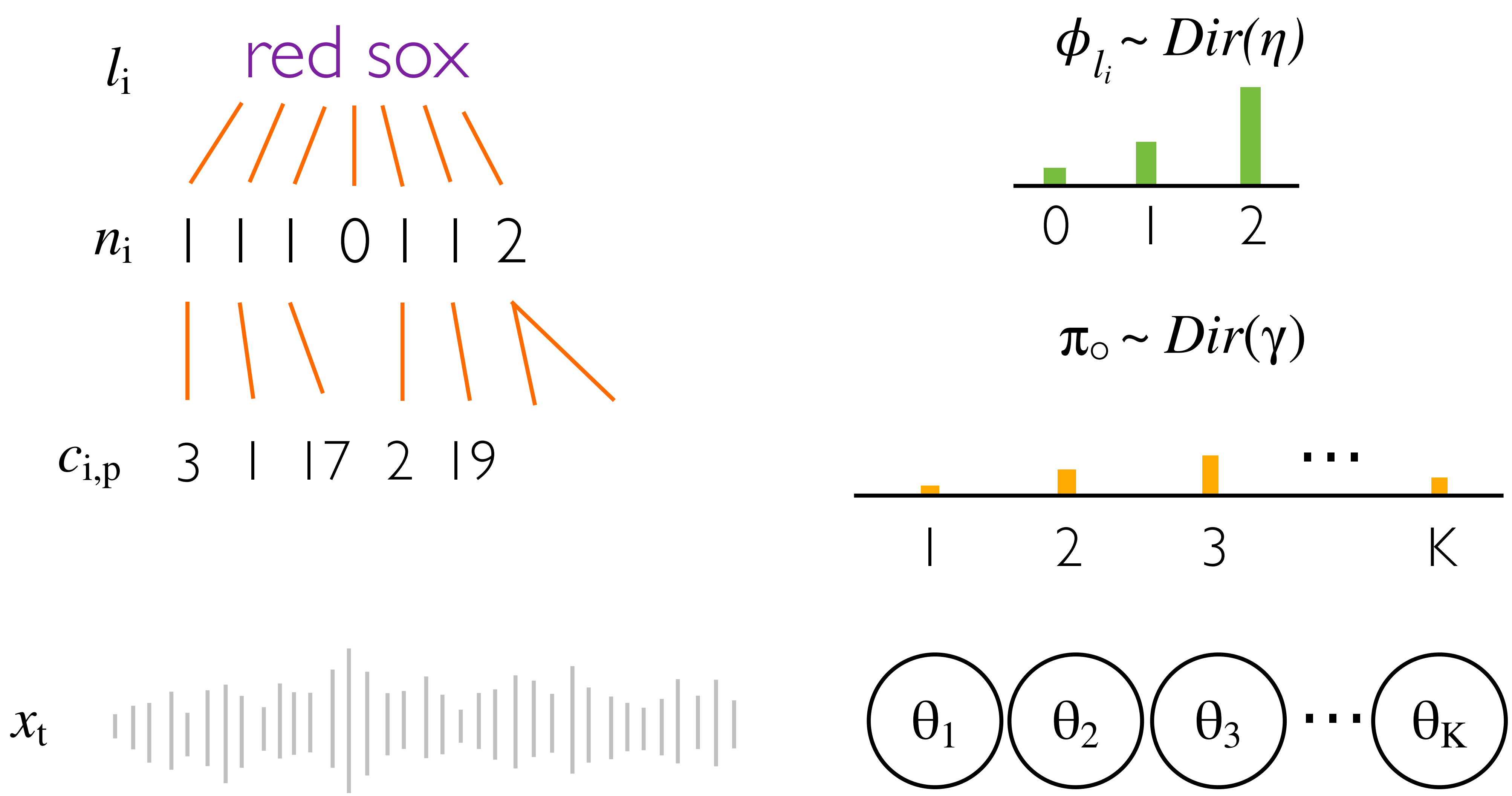
Generative Process

- Step 2
  - Generate the phone label ($c_{i,p}$) for every phone that a letter maps to, $1 \leq p \leq n_i$



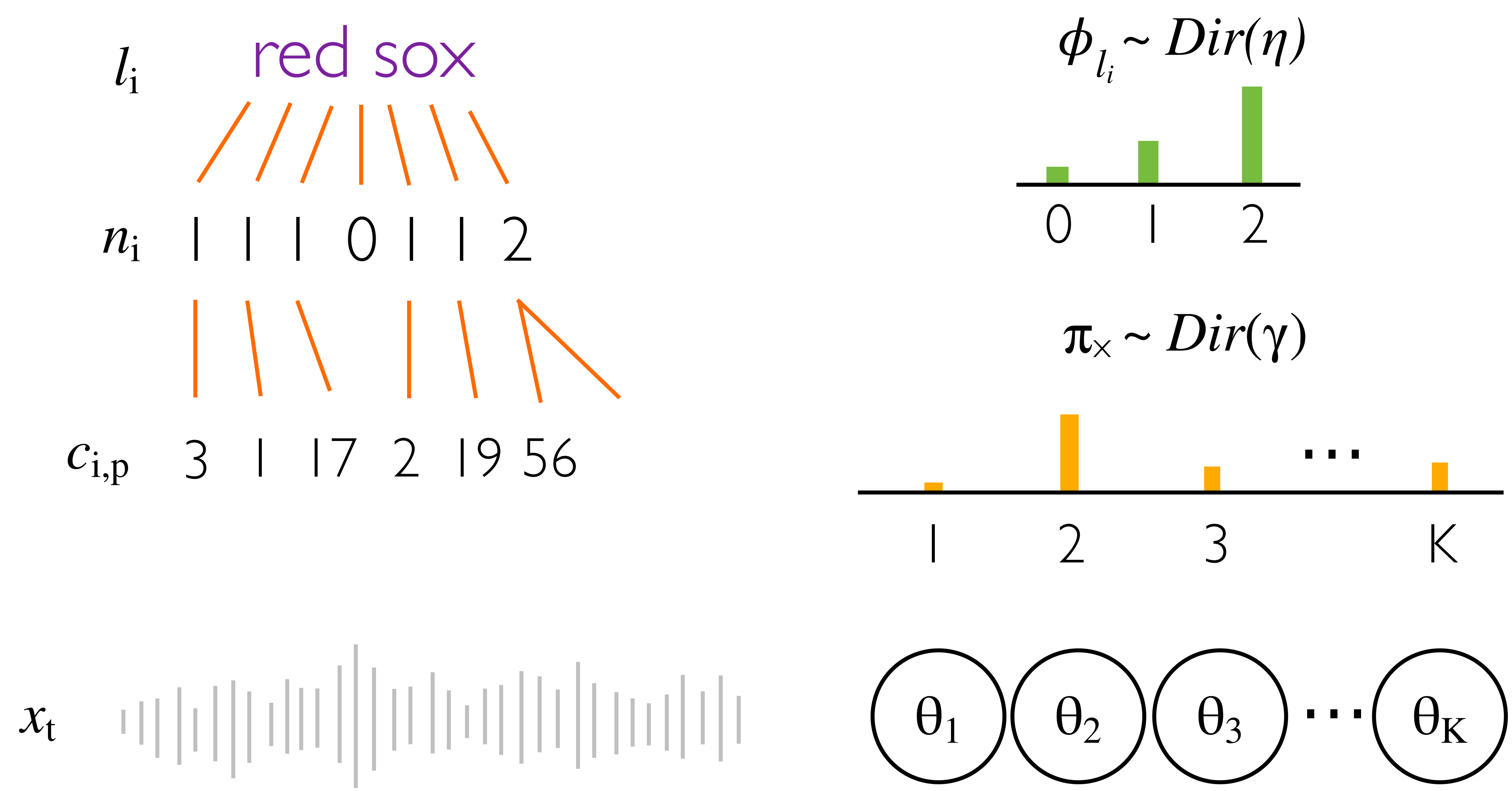
Generative Process

- Step 2
  - Generate the phone label ($c_{i,p}$) for every phone that a letter maps to, $1 \leq p \leq n_i$



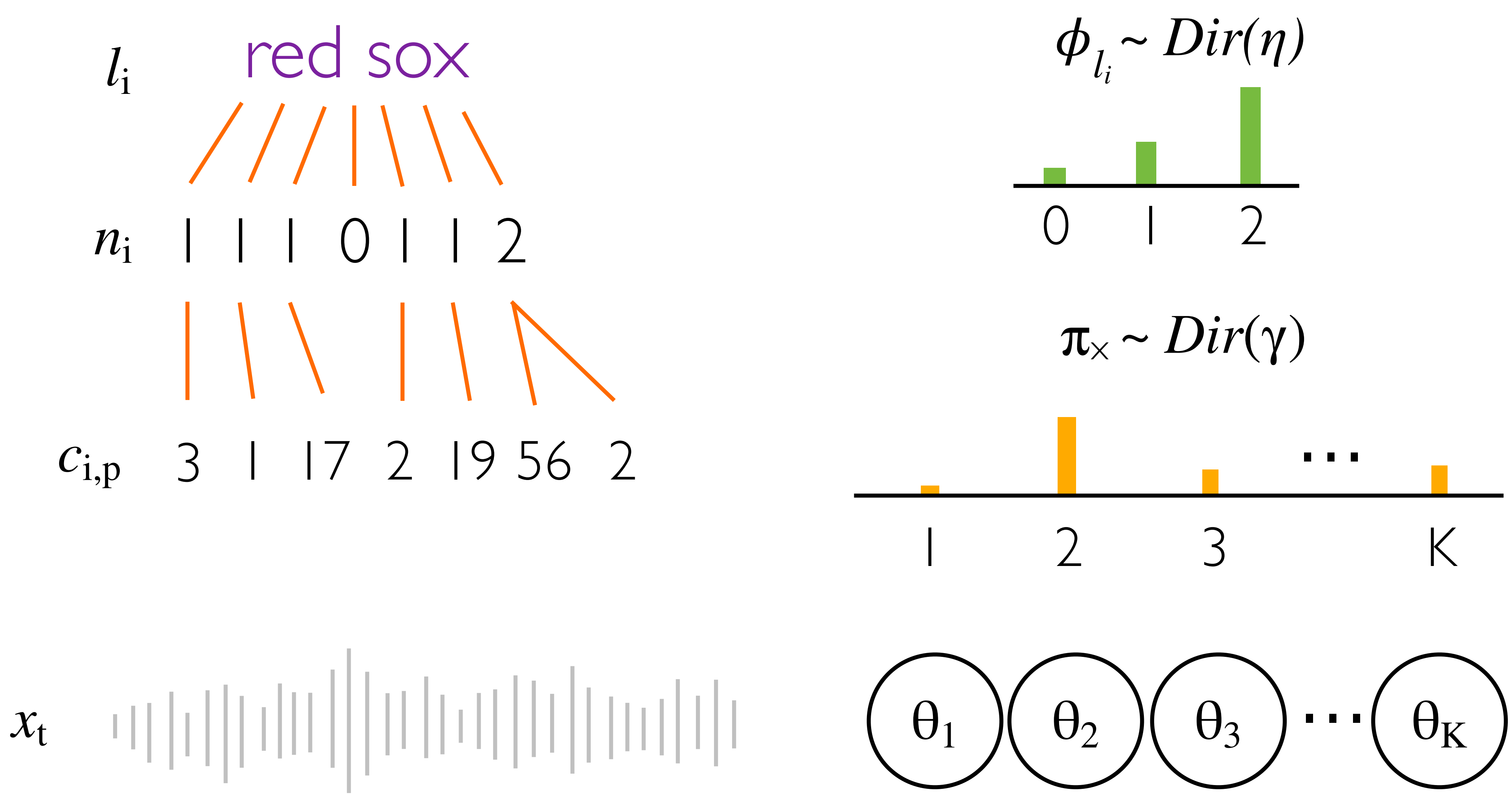
Generative Process

- Step 2
  - Generate the phone label ($c_{i,p}$) for every phone that a letter maps to, $1 \leq p \leq n_i$



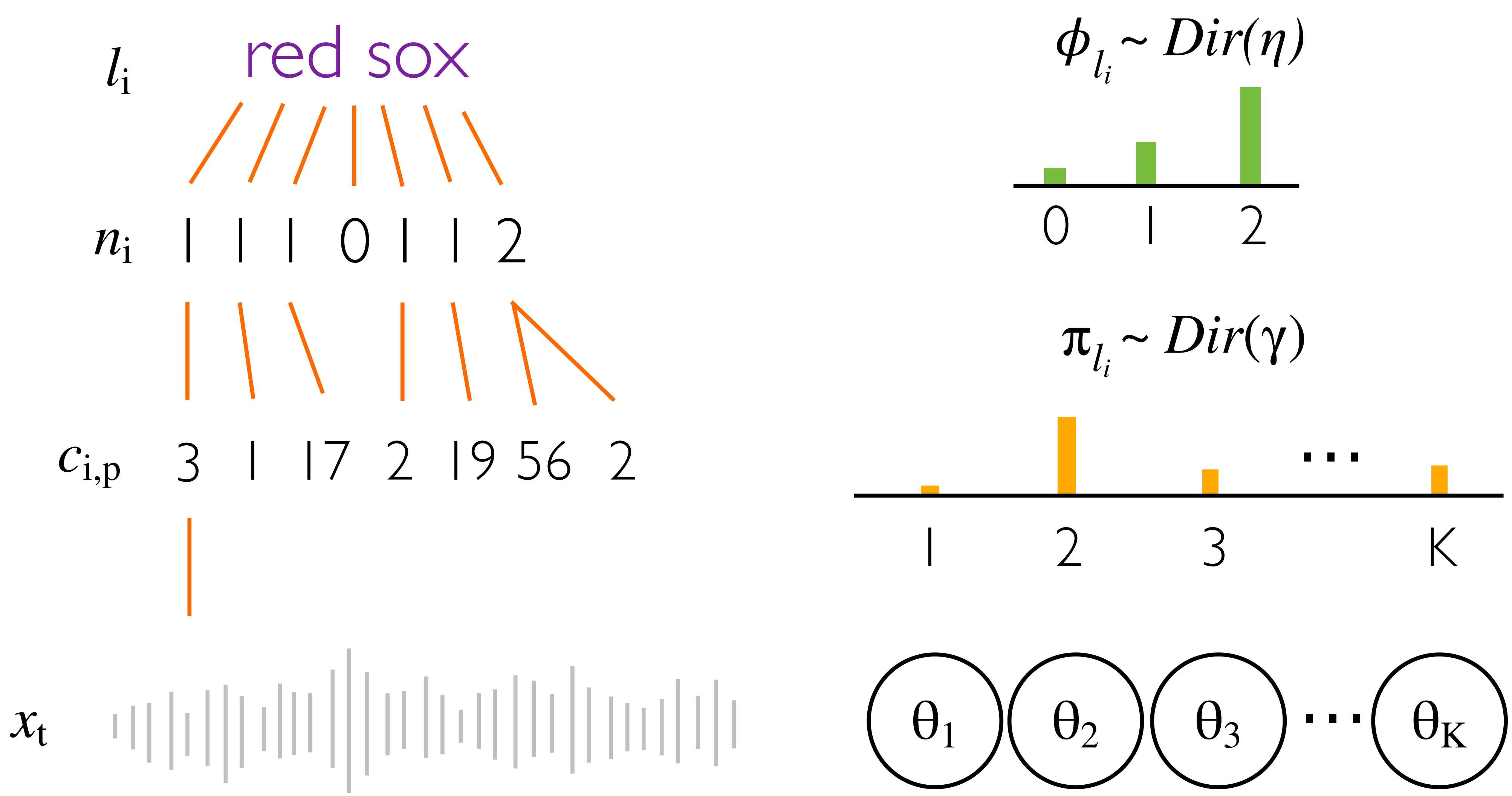
Generative Process

- Step 2
  - Generate the phone label ($c_{i,p}$) for every phone that a letter maps to, $1 \leq p \leq n_i$



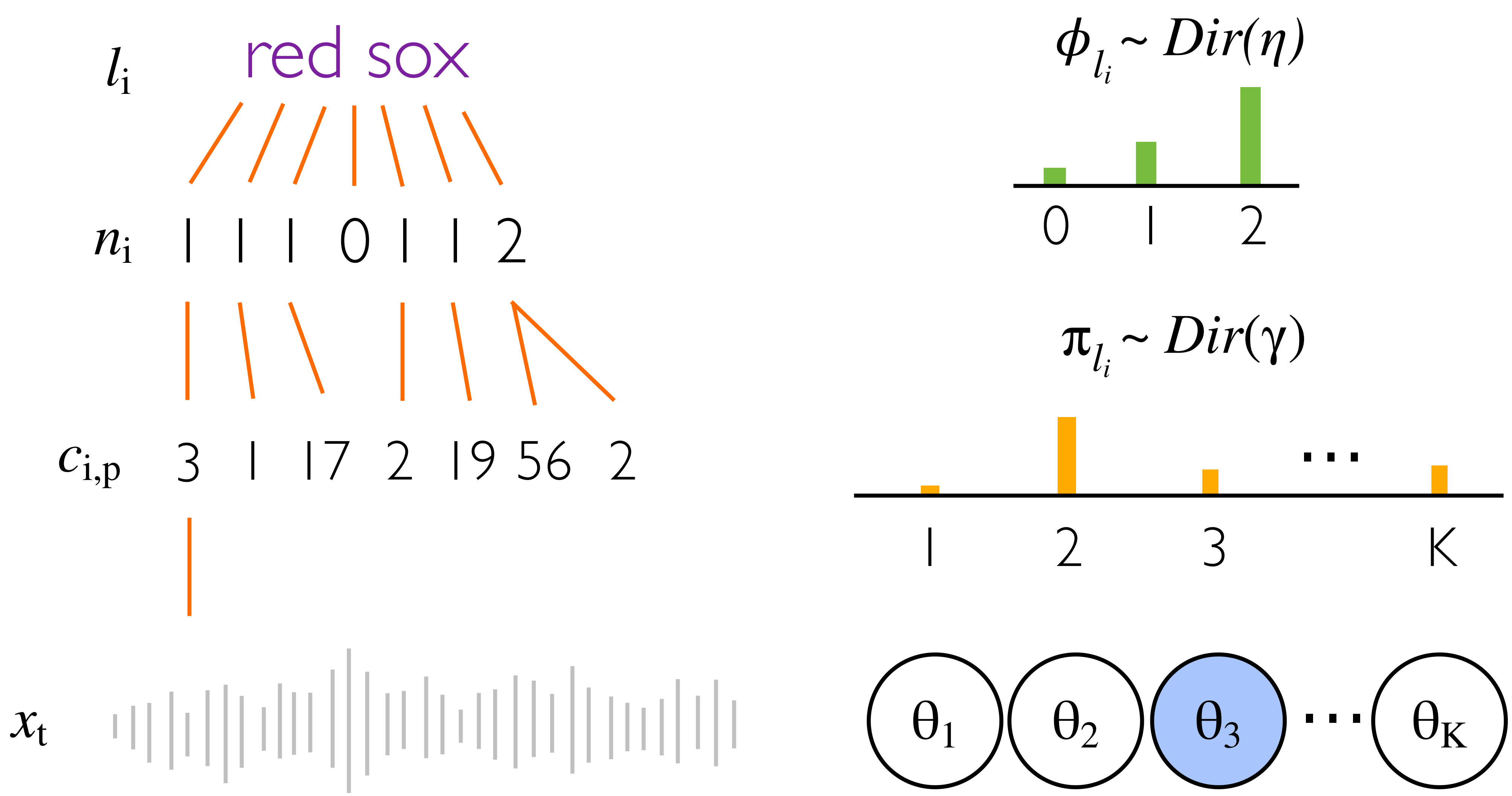
Generative Process

- **Step 3**
  - Generate speech (x_t)



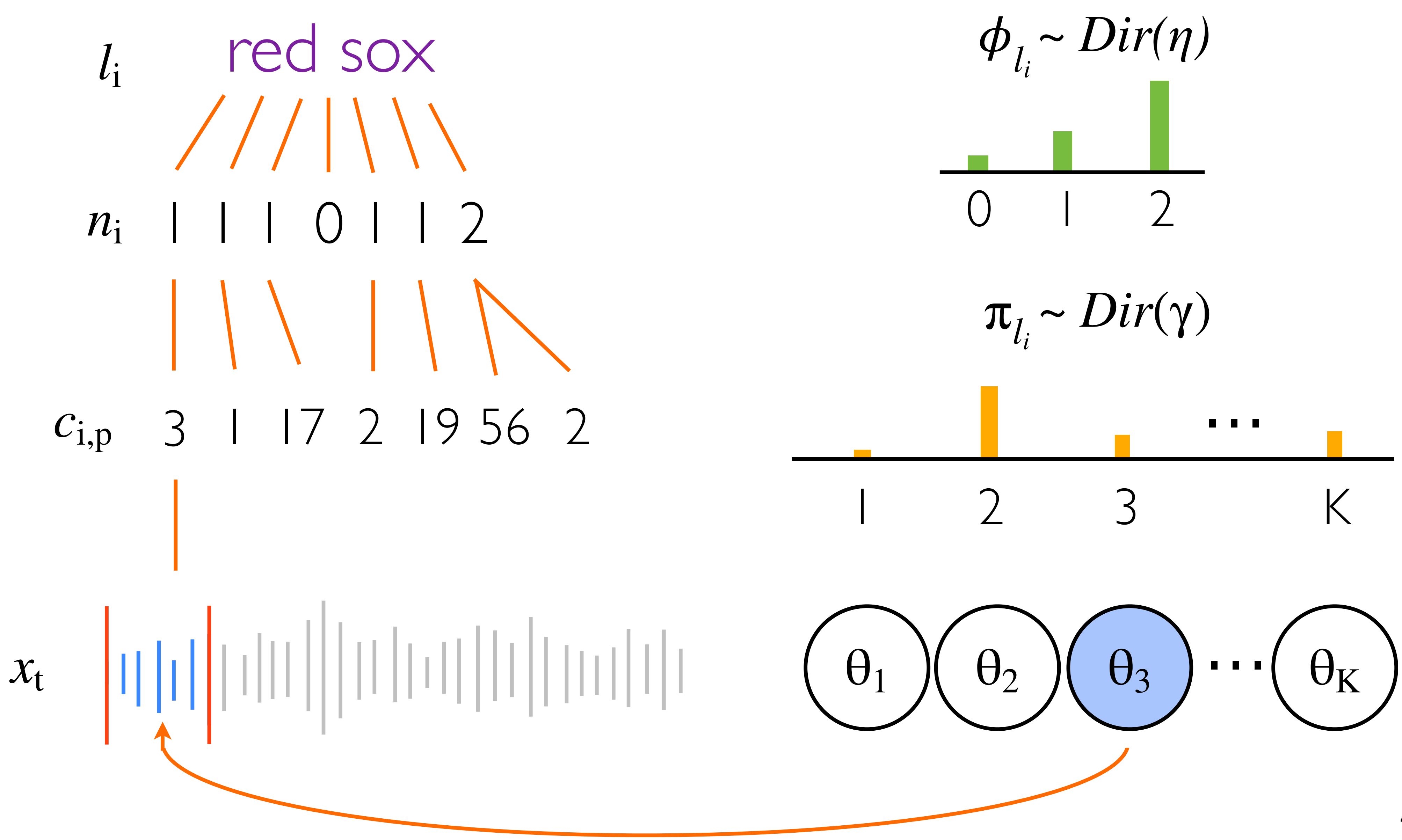
Generative Process

- Step 3
  - Generate speech (x_t)



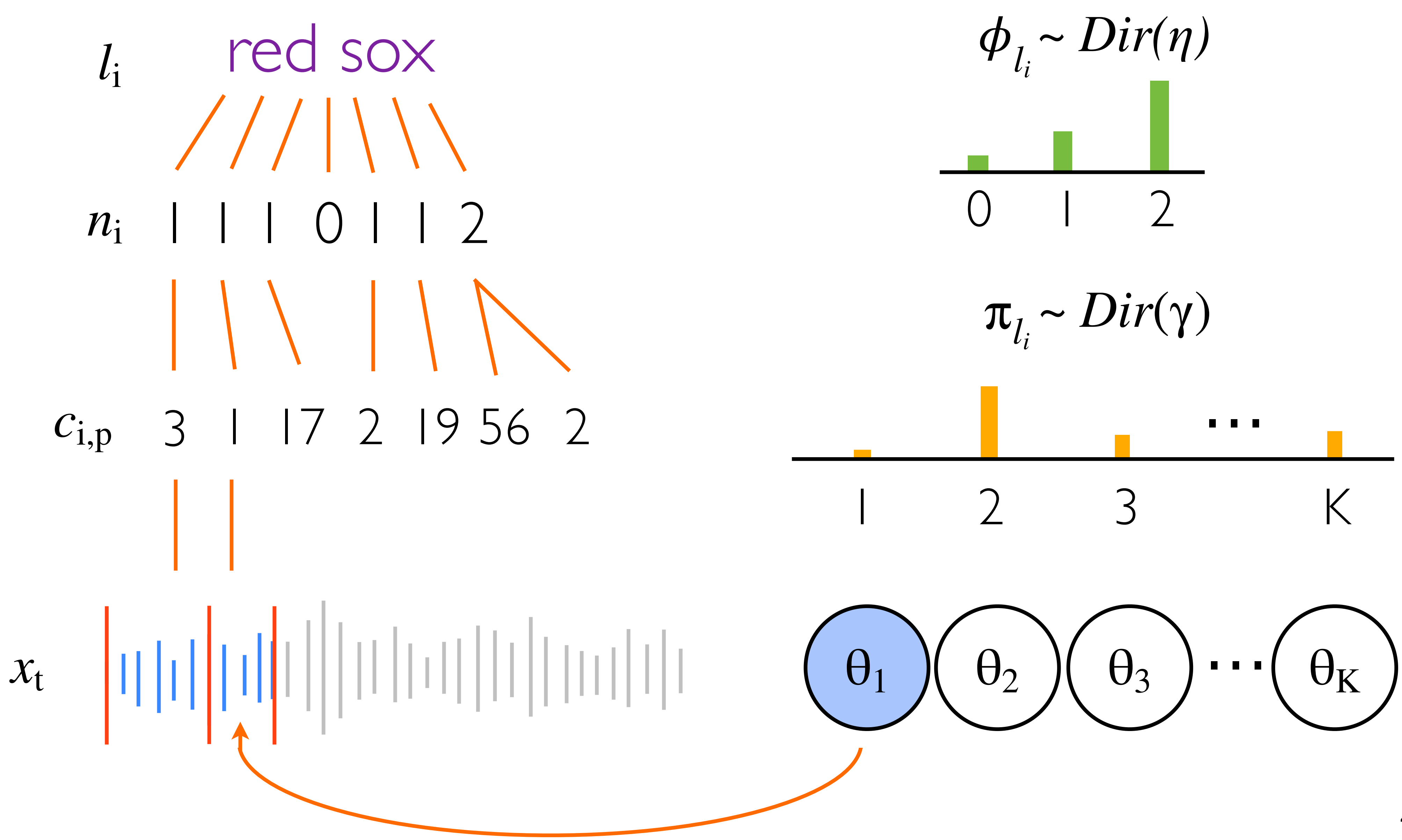
Generative Process

- Step 3
  - Generate speech (x_t)



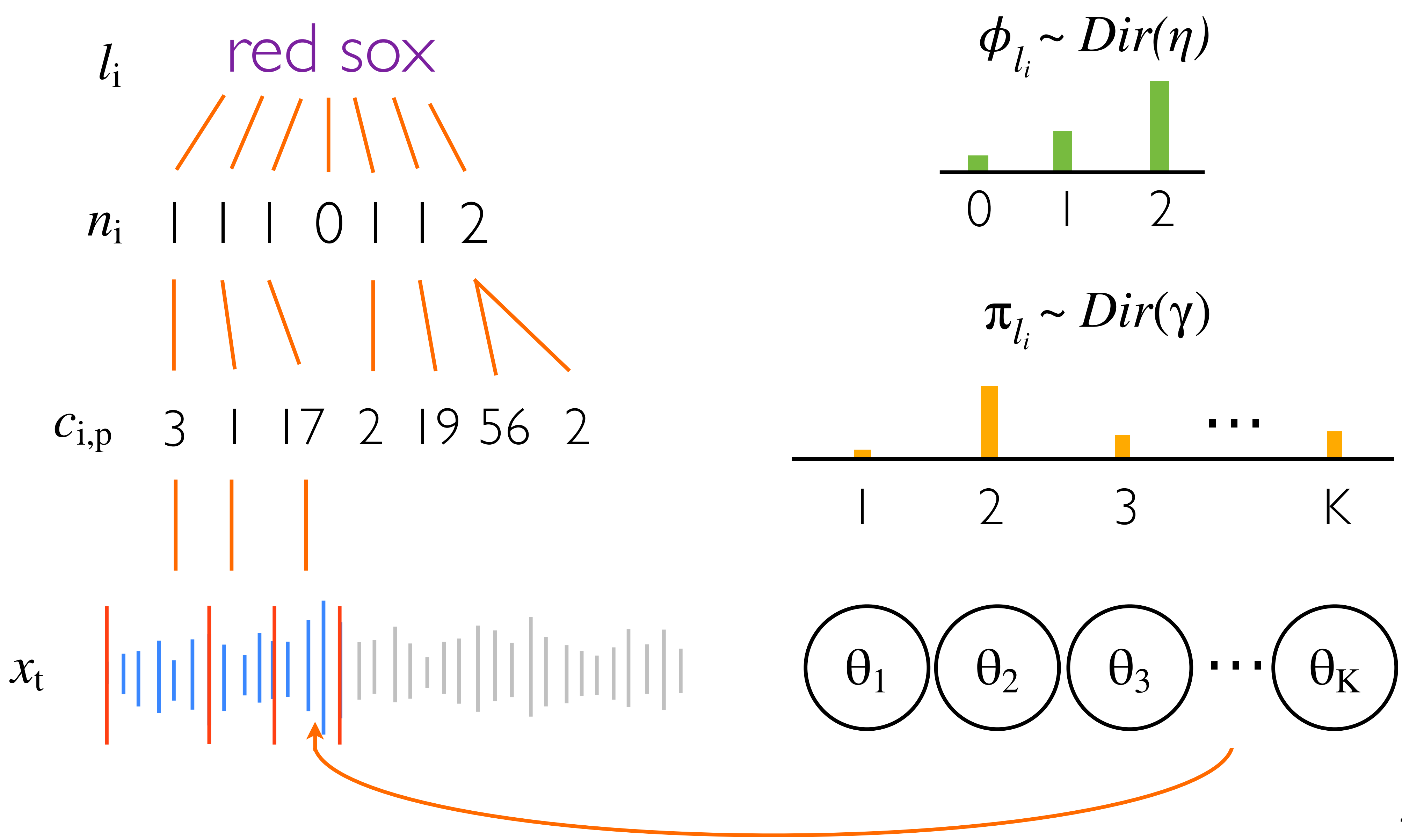
Generative Process

- Step 3
  - Generate speech (x_t)



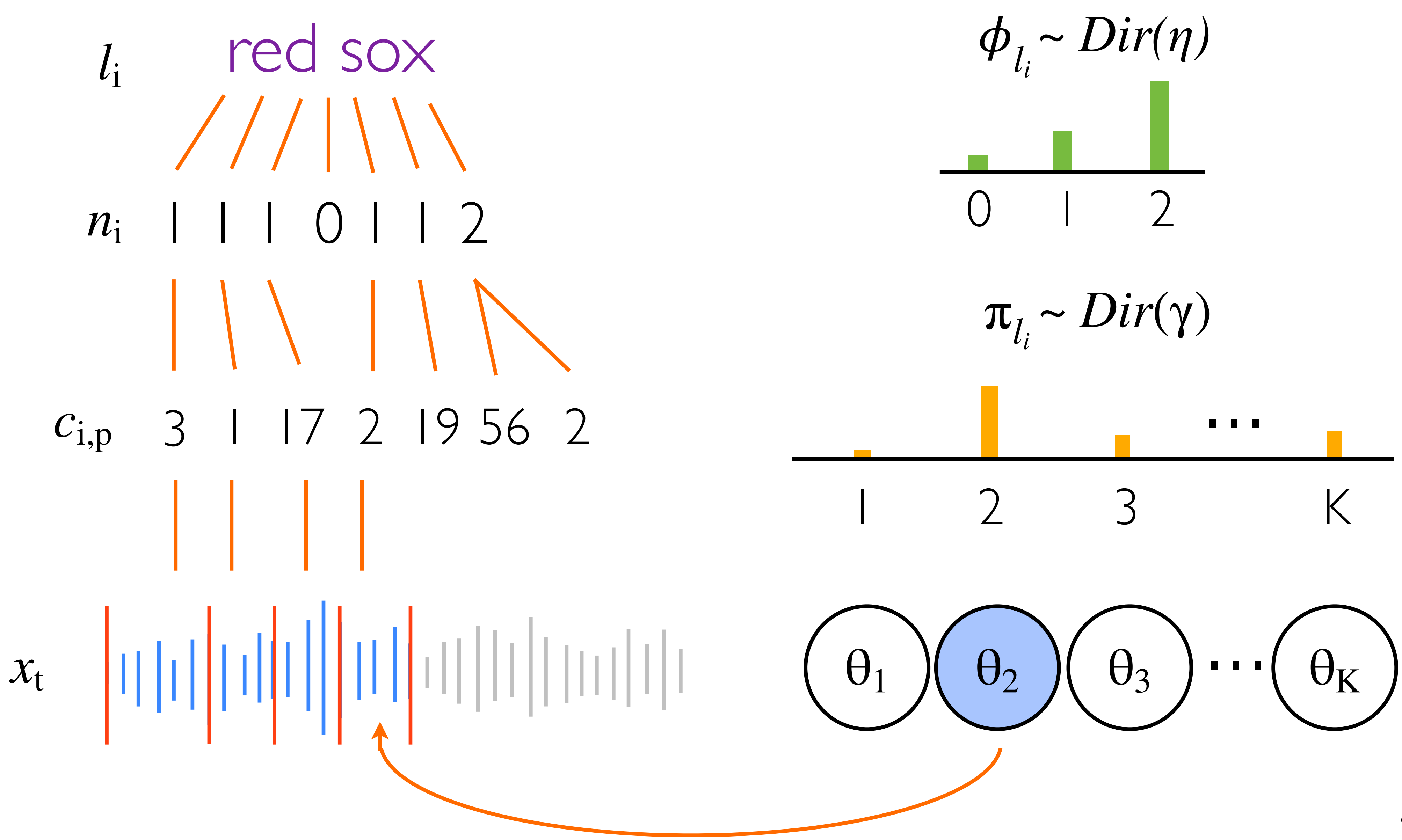
Generative Process

- Step 3
  - Generate speech (x_t)



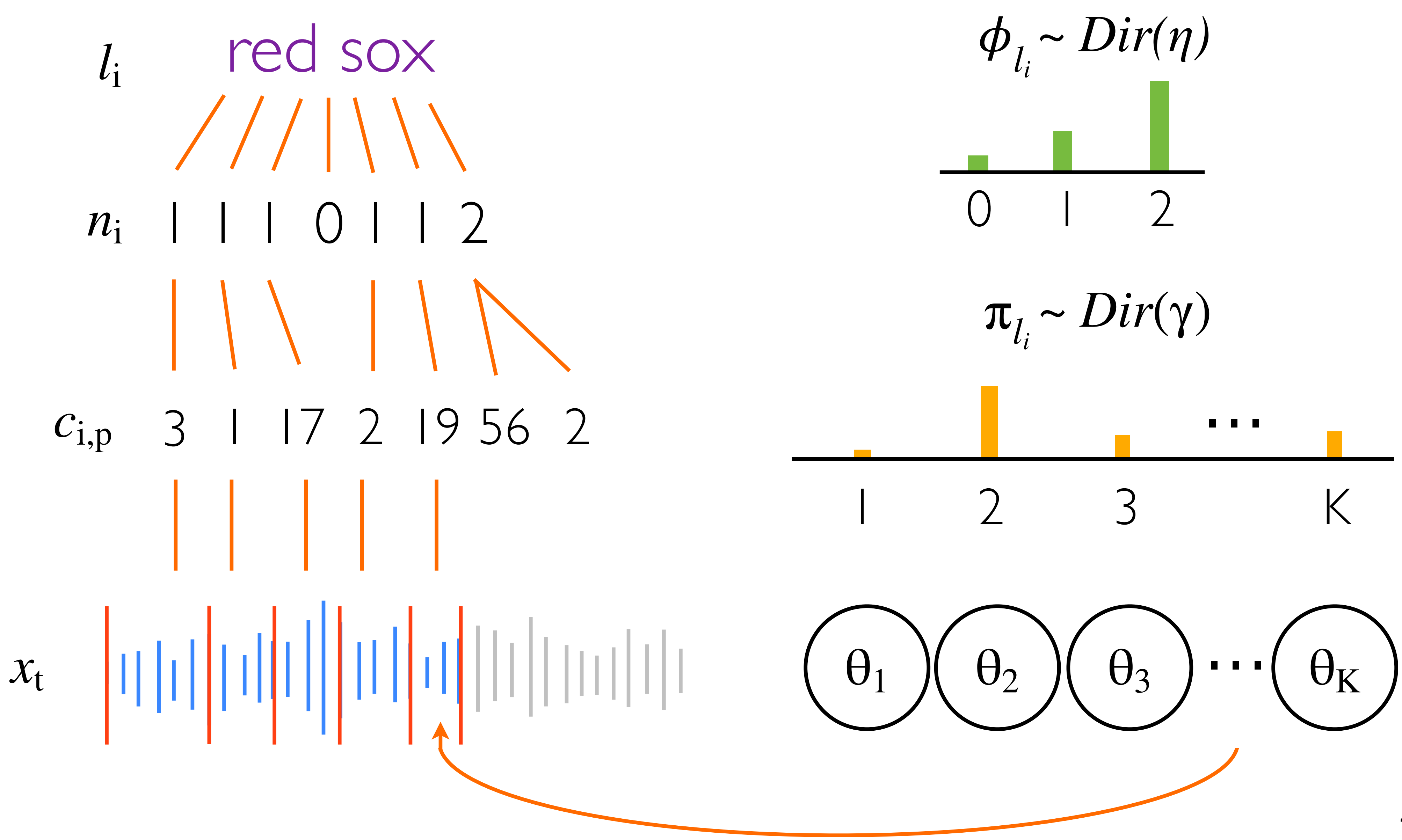
Generative Process

- Step 3
  - Generate speech (x_t)



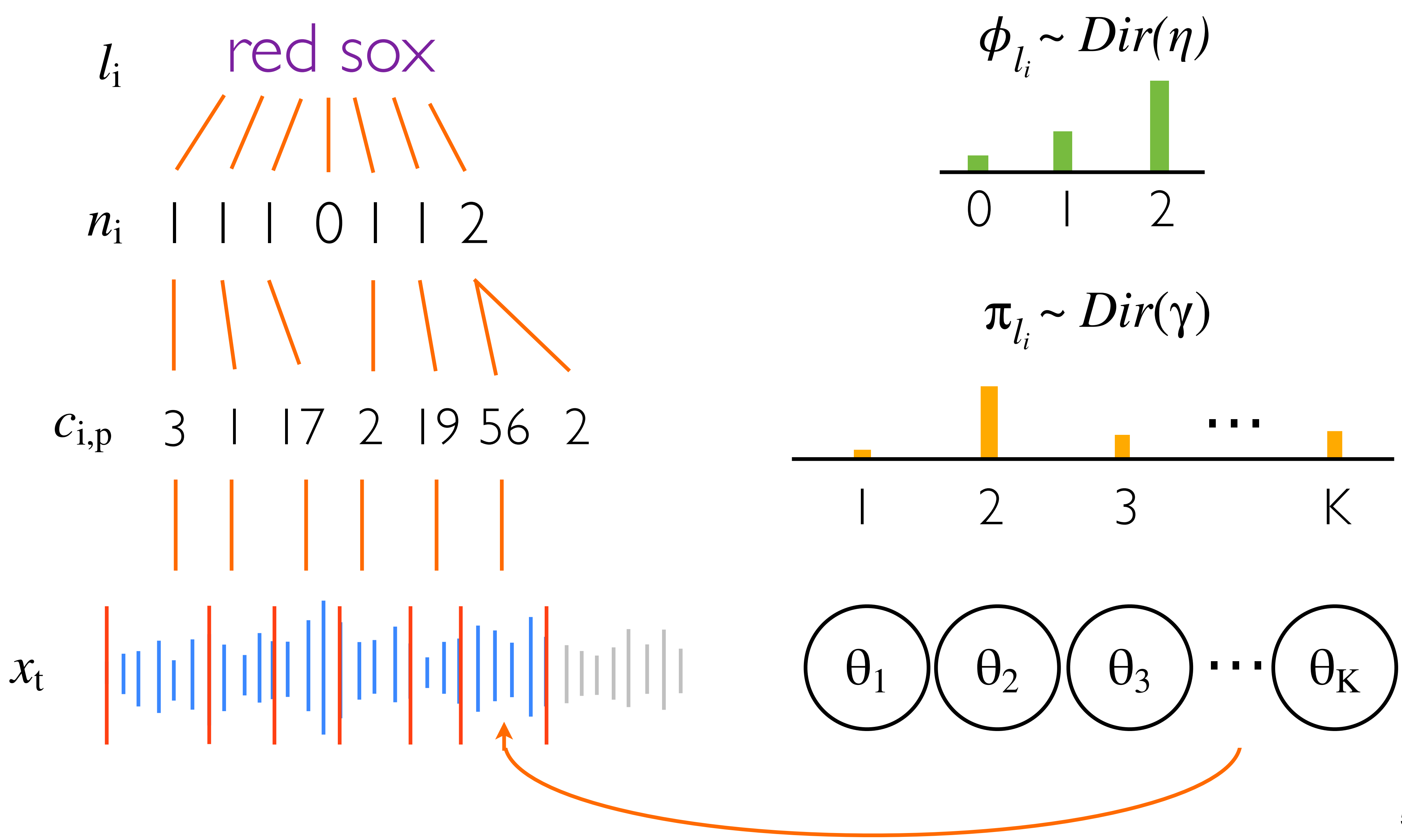
Generative Process

- Step 3
  - Generate speech (x_t)



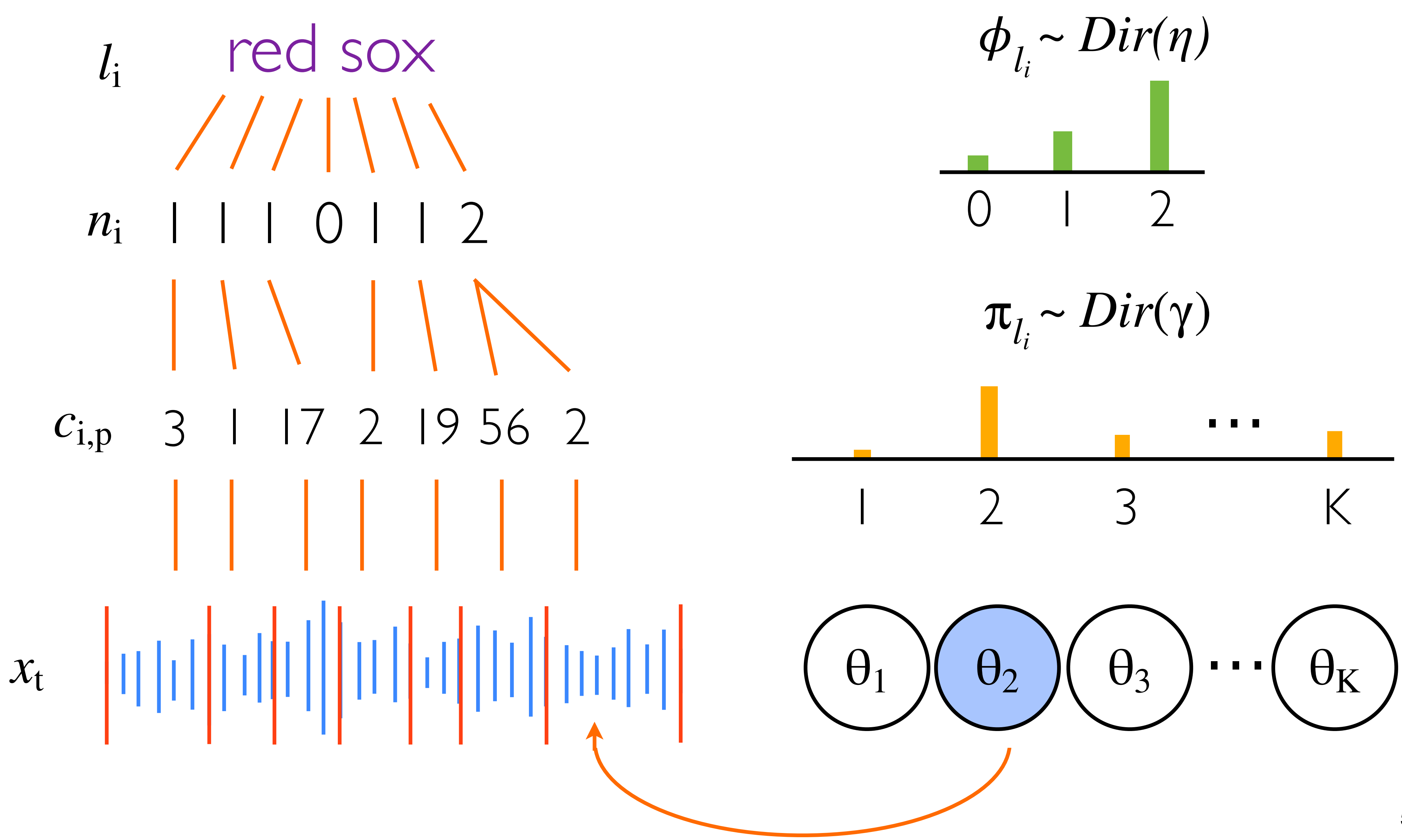
Generative Process

- Step 3
  - Generate speech (x_t)



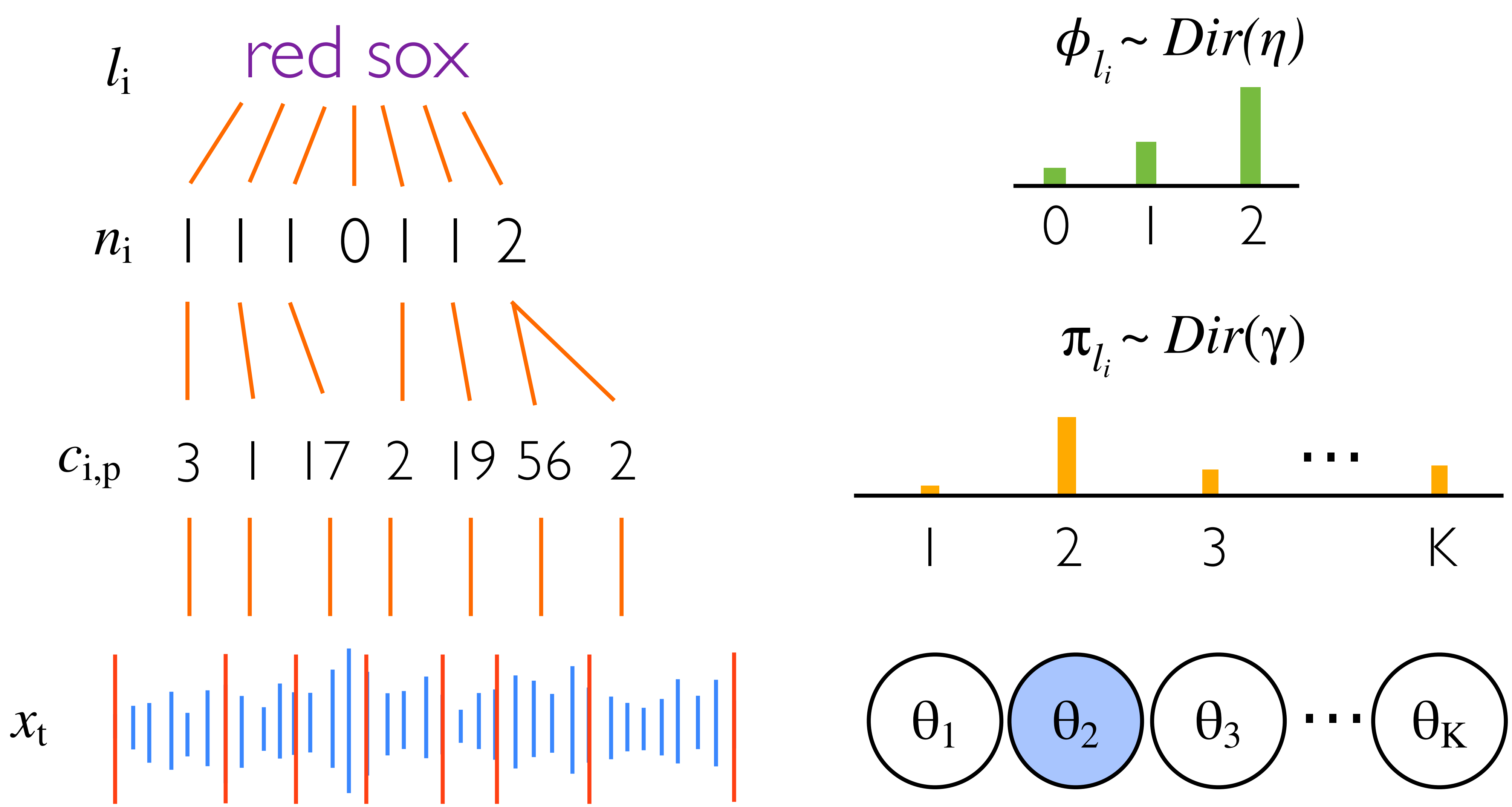
Generative Process

- Step 3
  - Generate speech (x_t)



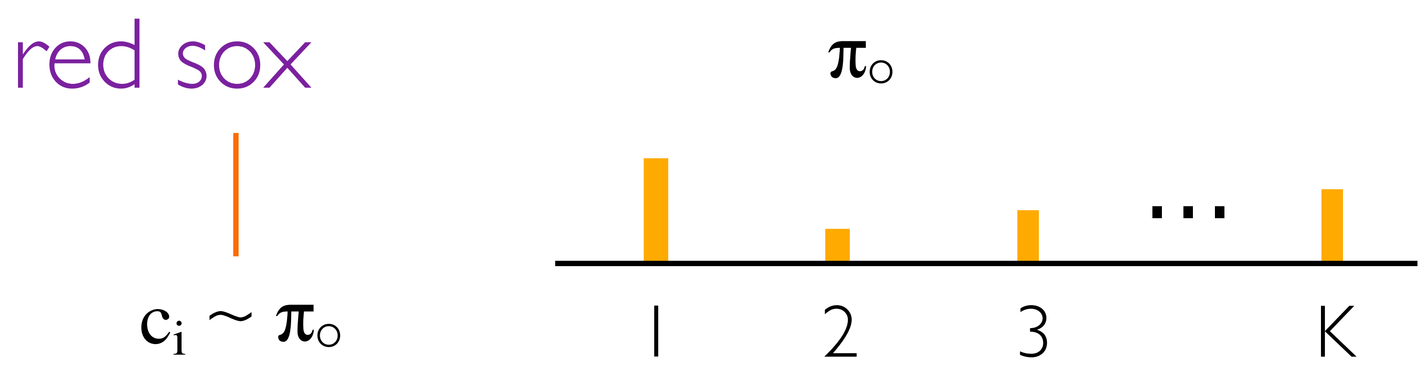
Generative Process

- Step 3
  - Generate speech (x_t)



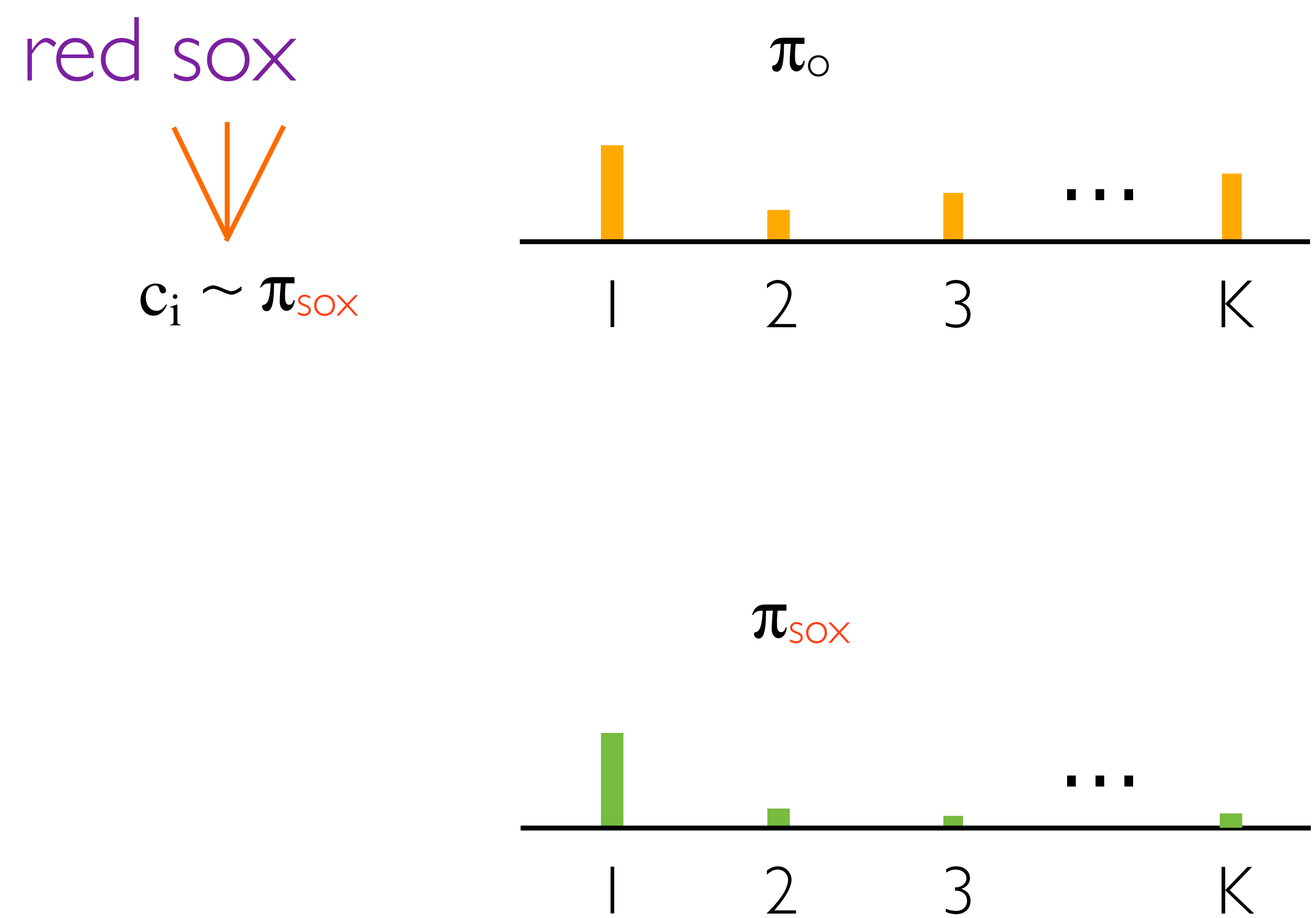
Context-dependent L2S Rules

- Take context into account for learning L2S mapping rules



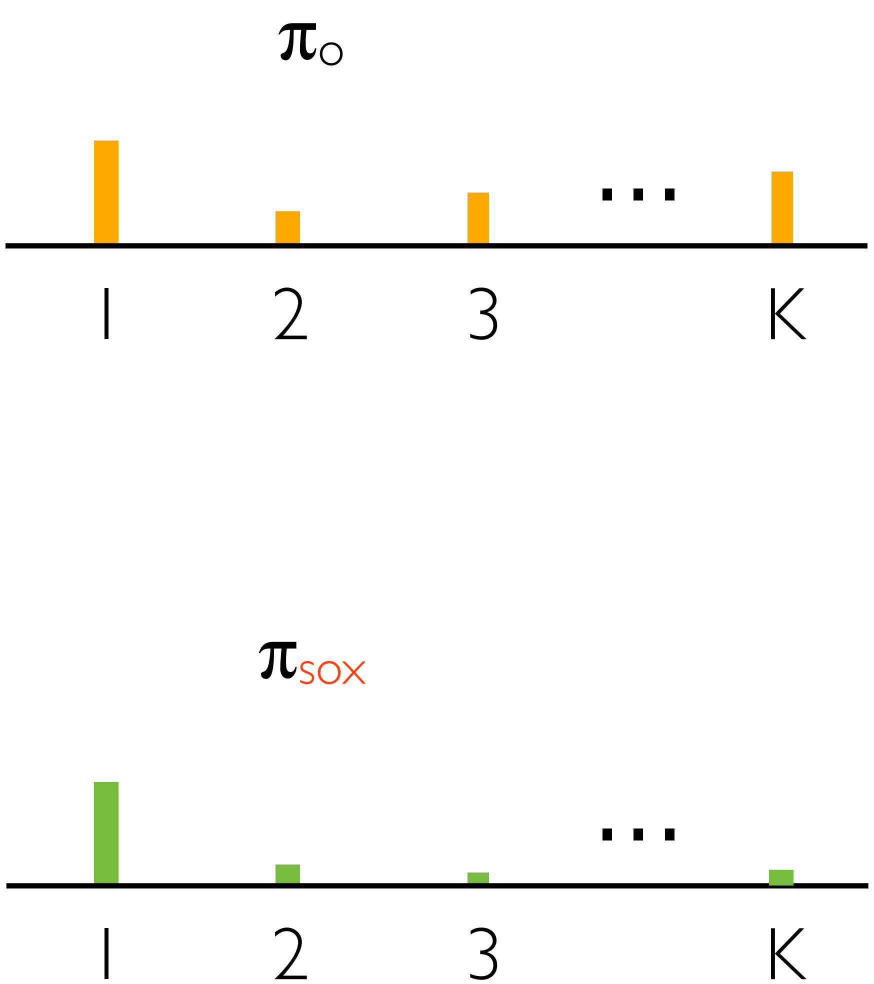
Context-dependent L2S Rules

- Take context into account for learning L2S mapping rules



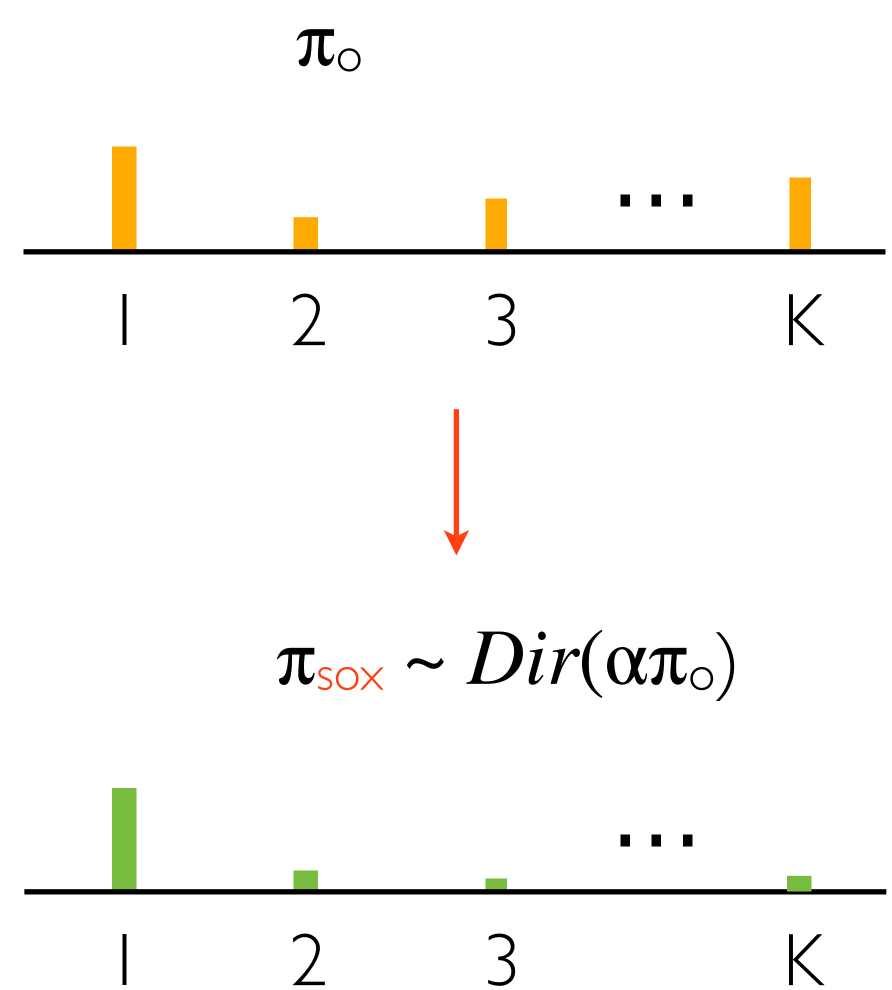
Context-dependent L2S Rules

- Take context into account for learning L2S mapping rules
  - More specific rules



Context-dependent L2S Rules

- Take context into account for learning L2S mapping rules
  - More specific rules
  - Back-off mechanism through hierarchy

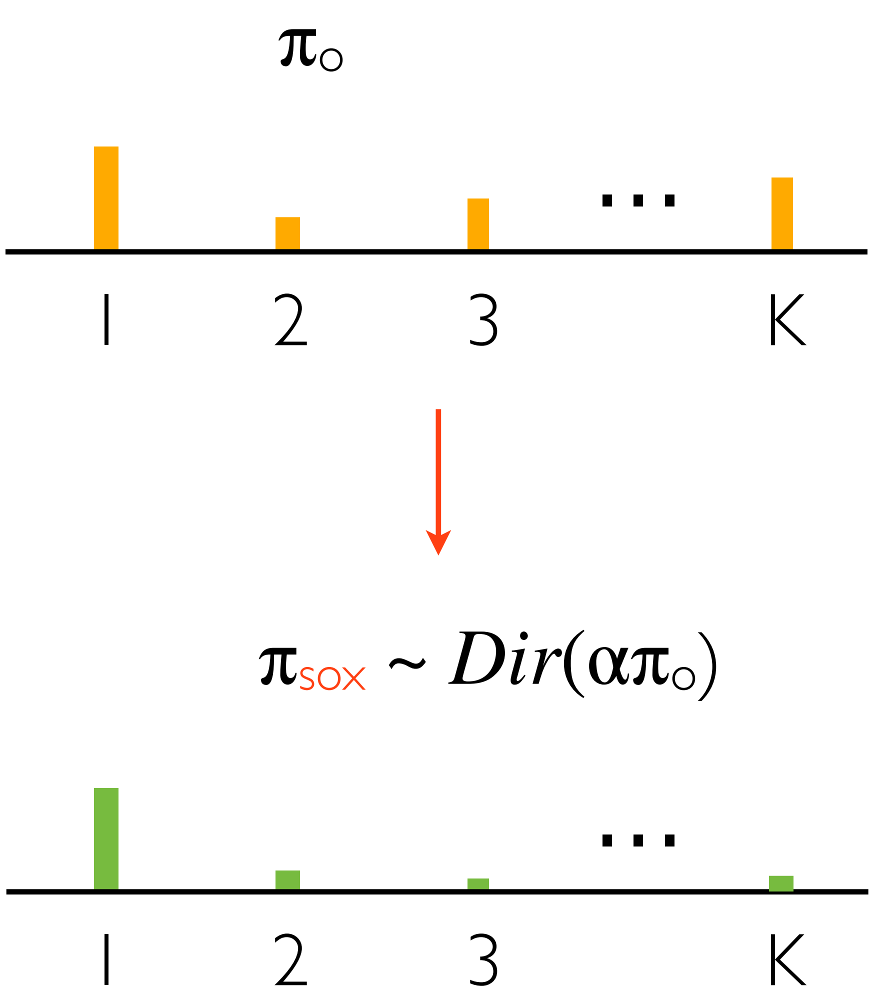


Context-dependent L2S Rules

- Take context into account for learning L2S mapping rules
  - More specific rules
  - Back-off mechanism through hierarchy
- View π_o as the prior of π_{sox}
  - If <u>sox</u> appears frequently

    $\pi_{sox} \longrightarrow$ empirical distribution
  - If <u>sox</u> is rarely observed

    $\pi_{sox} \longrightarrow \pi_o$

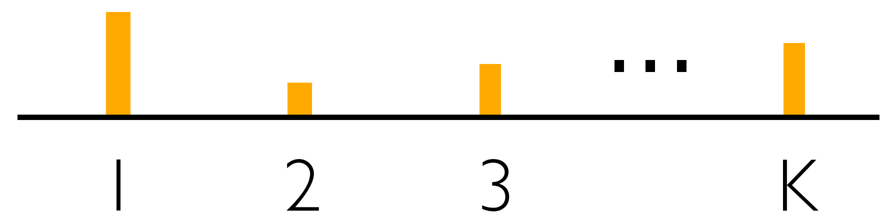


Context-dependent L2S Rules

- Take context into account for learning L2S mapping rules
  - More specific rules
  - Back-off mechanism through hierarchy

$$\beta \sim Dir(\gamma)$$

$$\pi_o \sim Dir(\lambda\beta)$$



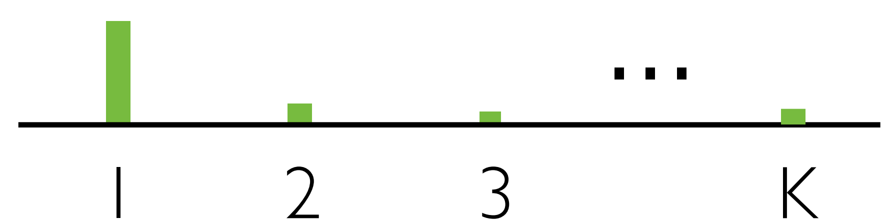
- View π_o as the prior of π_{sox}
  - If sox appears frequently

    $\pi_{sox} \longrightarrow$ empirical distribution

  - If sox is rarely observed

    $\pi_{sox} \longrightarrow \pi_o$

$$\pi_{sox} \sim Dir(\alpha\pi_o)$$



Graphical Model

G : the set of graphemes

$\underline{l}$: sequence of three graphemes

l : observed graphemes

x : observation speech

$\mathbf{d}$: phone duration

c : phone id

n : number of phones a grapheme maps to

L : total number of graphemes

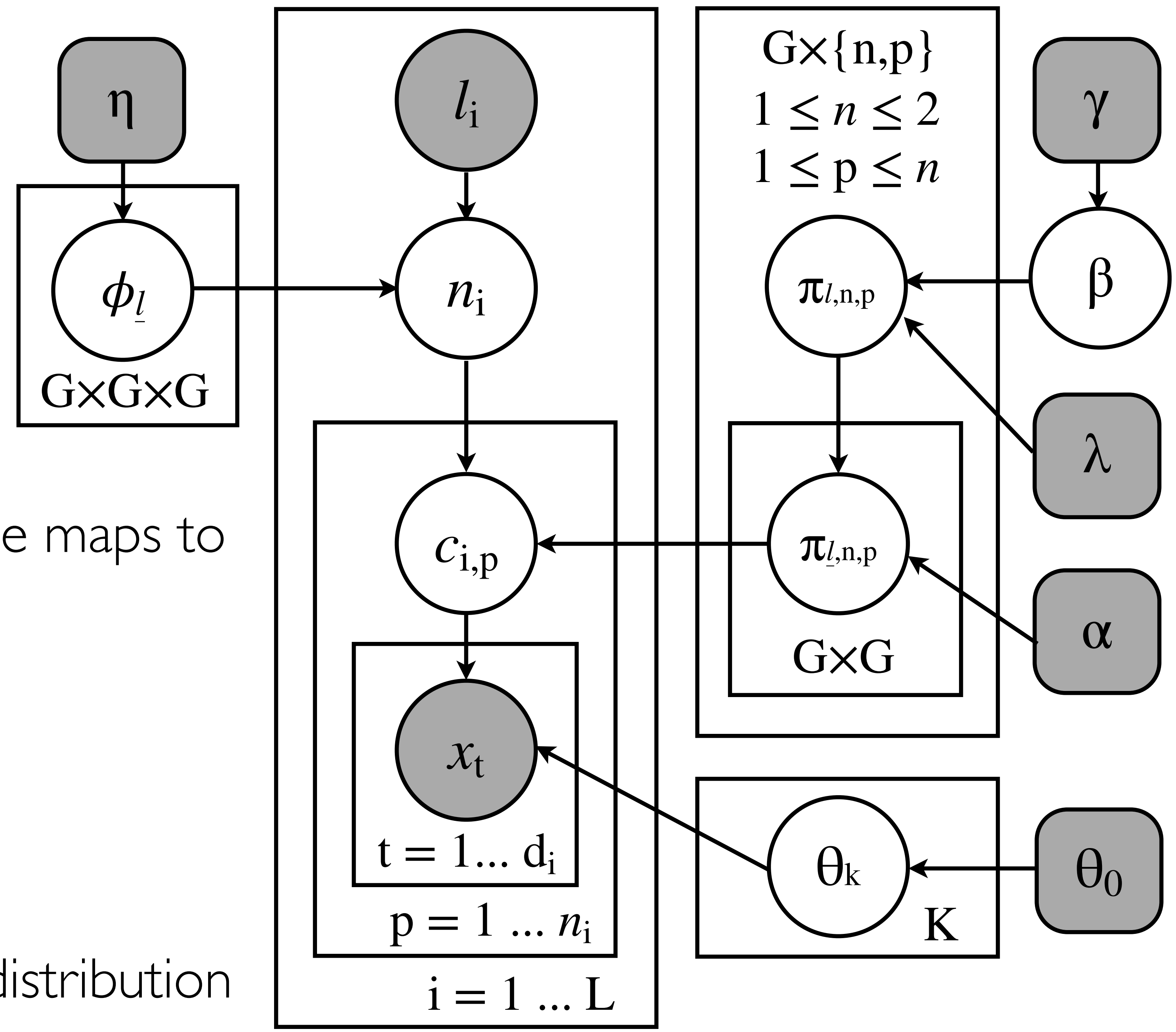
K : total number of HMMs

$\phi_{\underline{l}}$: 3-dim categorical distribution

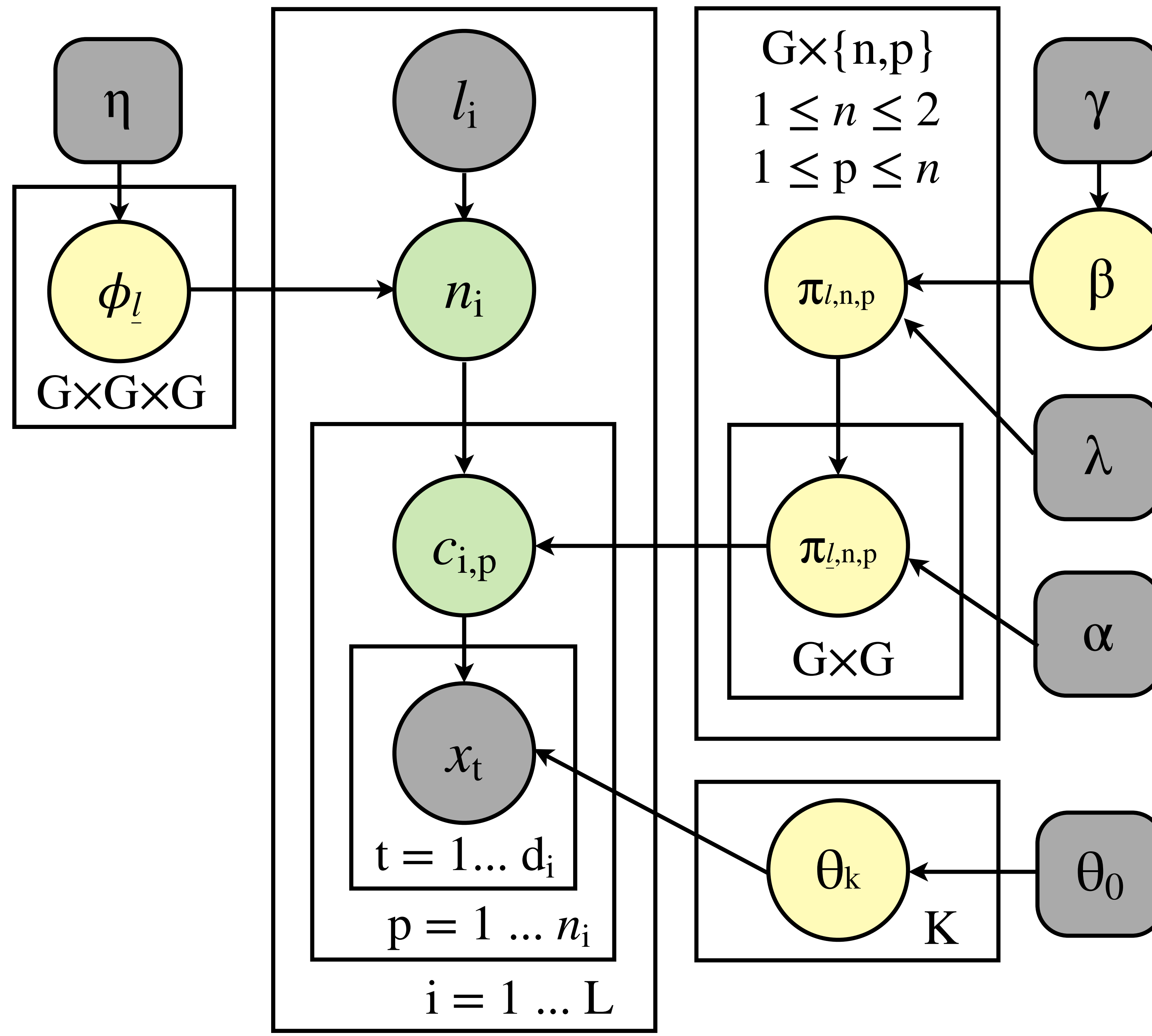
θ_k : a HMM θ_0 : HMM prior

$\pi_{l,n,p}$, $\pi_{\underline{l},n,p}$, β : K-dim categorical distribution

γ, λ, α : concentration parameter

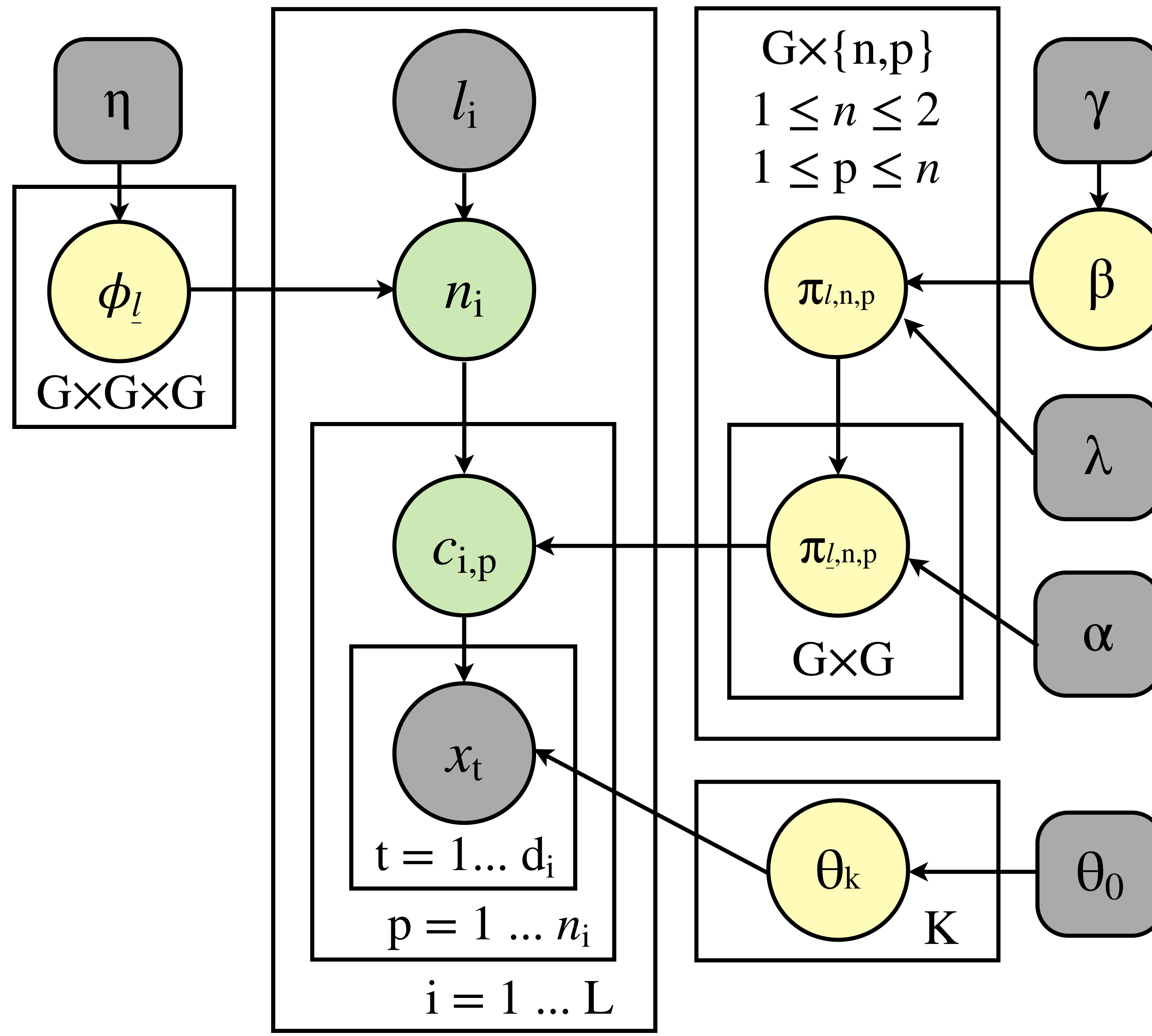


Inference



η
l_i
G×{n,p}
1 ≤ n ≤ 2
1 ≤ p ≤ n
γ
ϕ_l
n_i
$\pi_{l,n,p}$
β
G×G×G
λ
$c_{i,p}$
$\pi_{l,n,p}$
α
G×G
x_t
t = 1... d_i
θ_k
θ_0
K
p = 1 ... n_i
i = 1 ... L

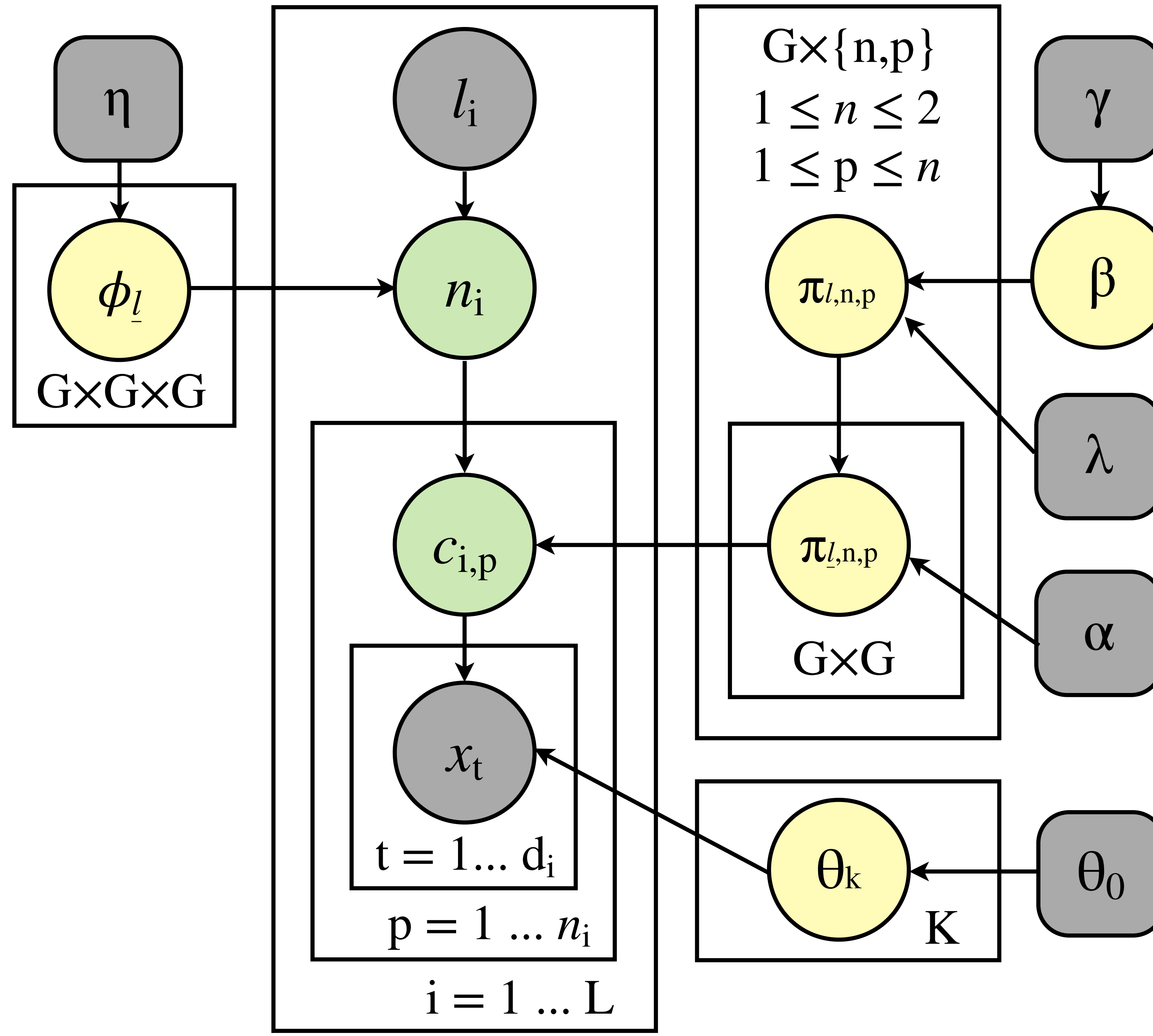
Inference

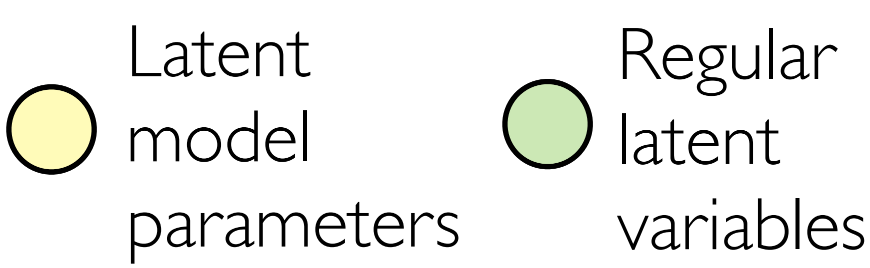

η
l_i
G×{n,p}
$1 \leq n \leq 2$
$1 \leq p \leq n$
γ
$\phi_{\underline{l}}$
n_i
$\pi_{l,n,p}$
β
G×G×G
λ
$c_{i,p}$
$\pi_{\underline{l},n,p}$
α
G×G
x_t
θ_k
θ_0
t = 1… d_i
p = 1 … n_i
K
i = 1 … L


Latent model parameters
Regular latent variables

Inference

- Procedure

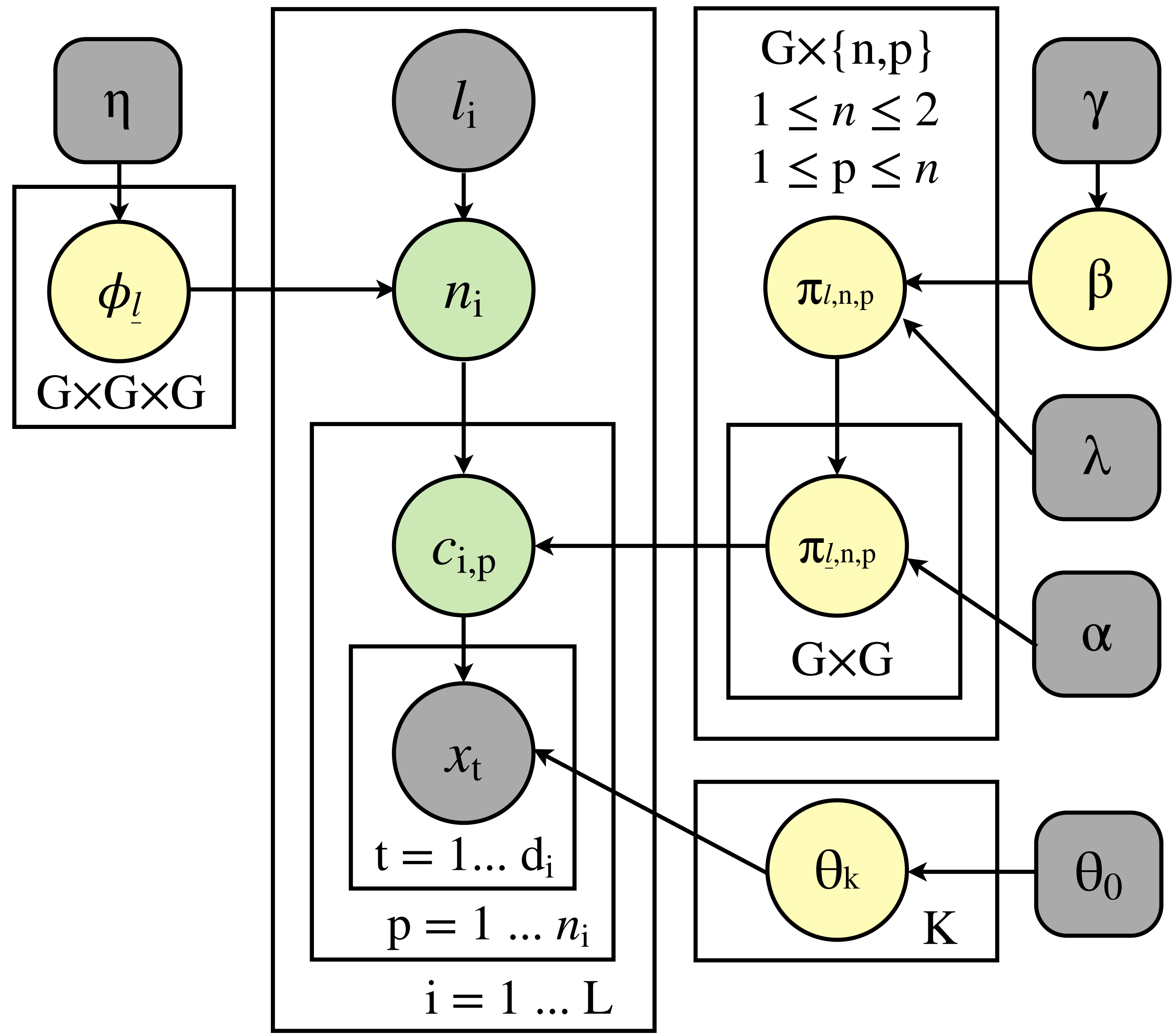




Inference

- **Procedure**

Sample from prior

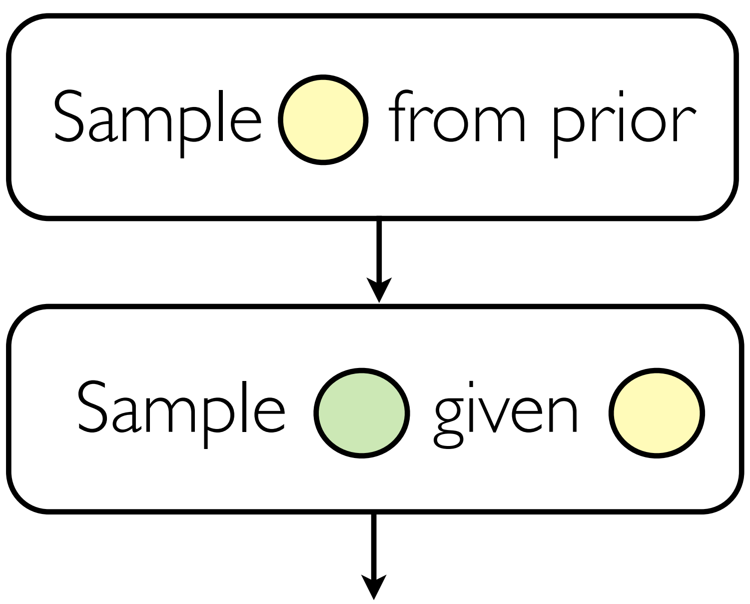


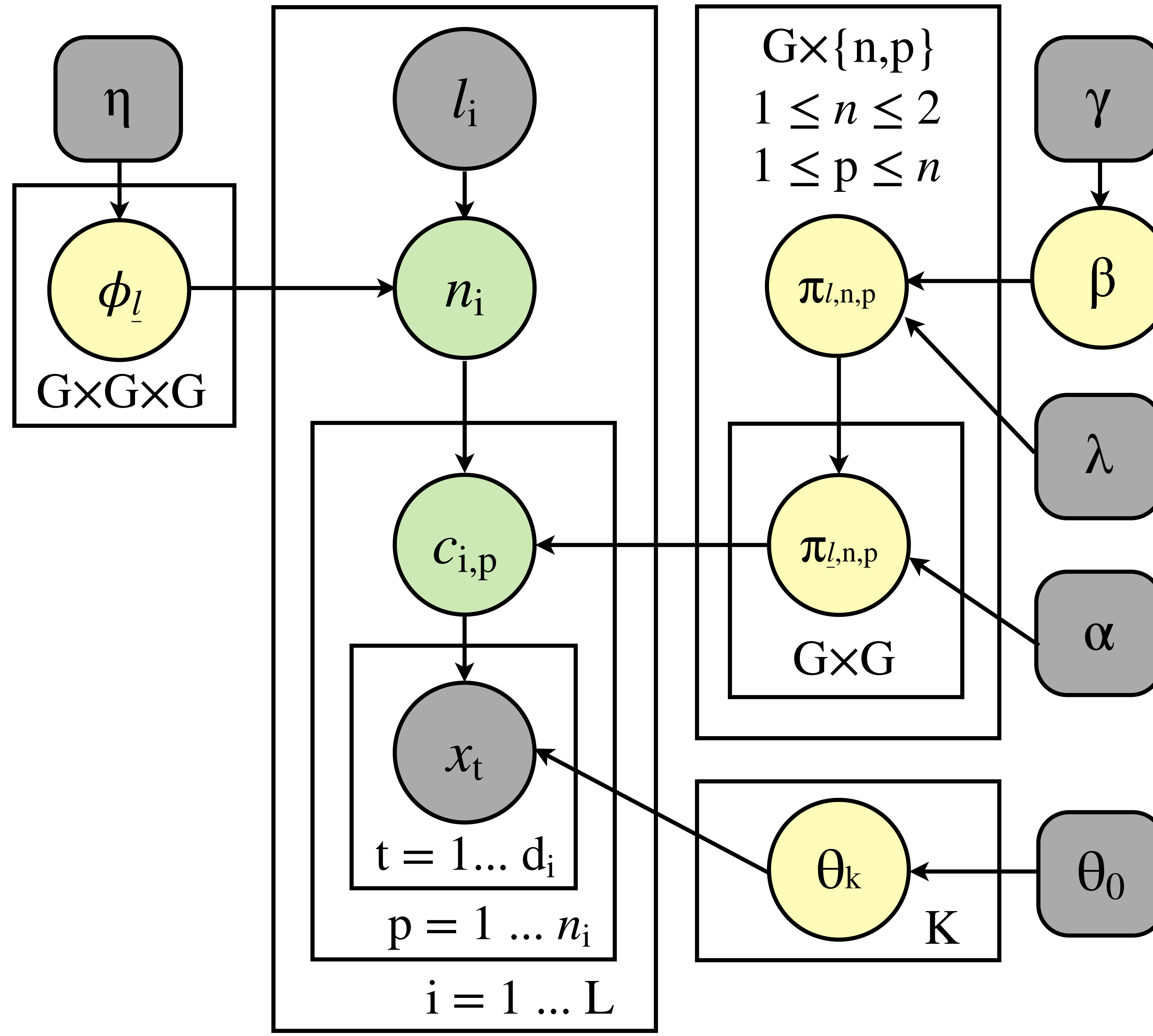
Latent model parameters

Regular latent variables

Inference

- **Procedure**



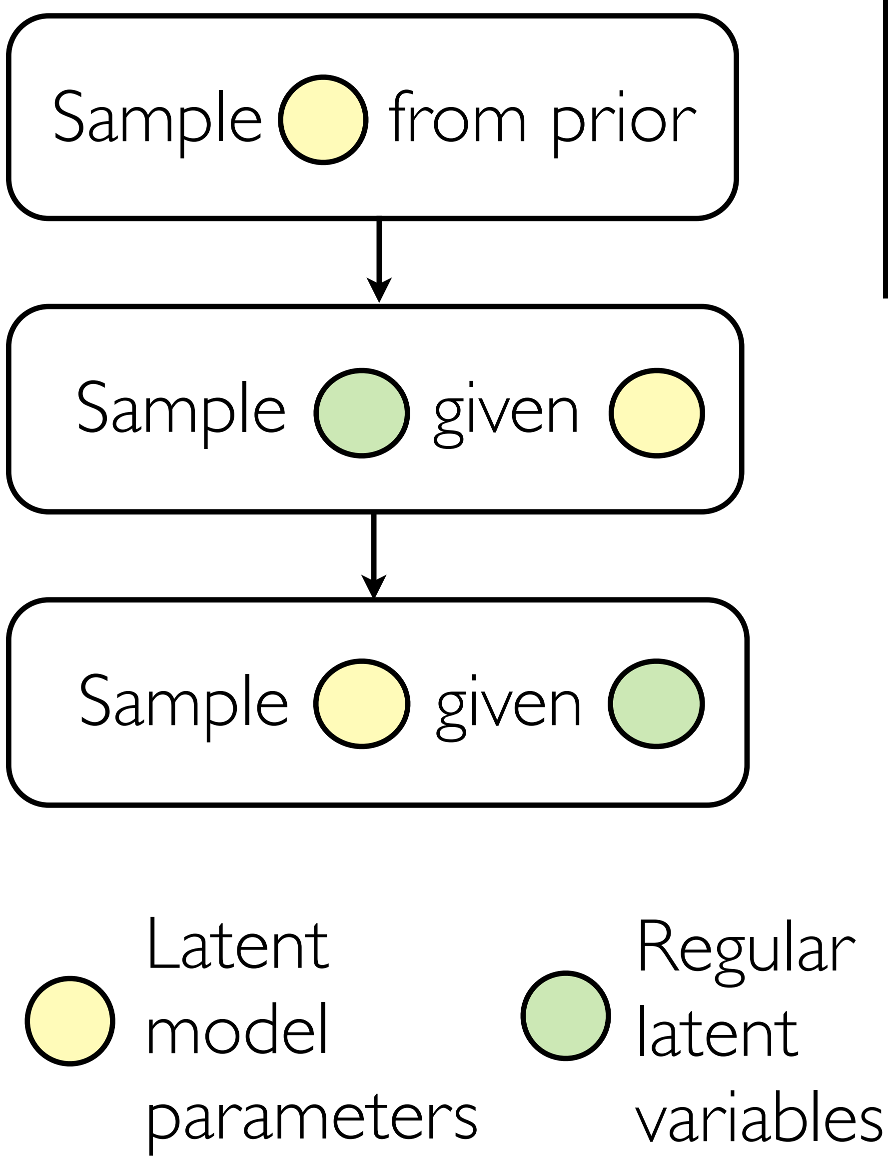


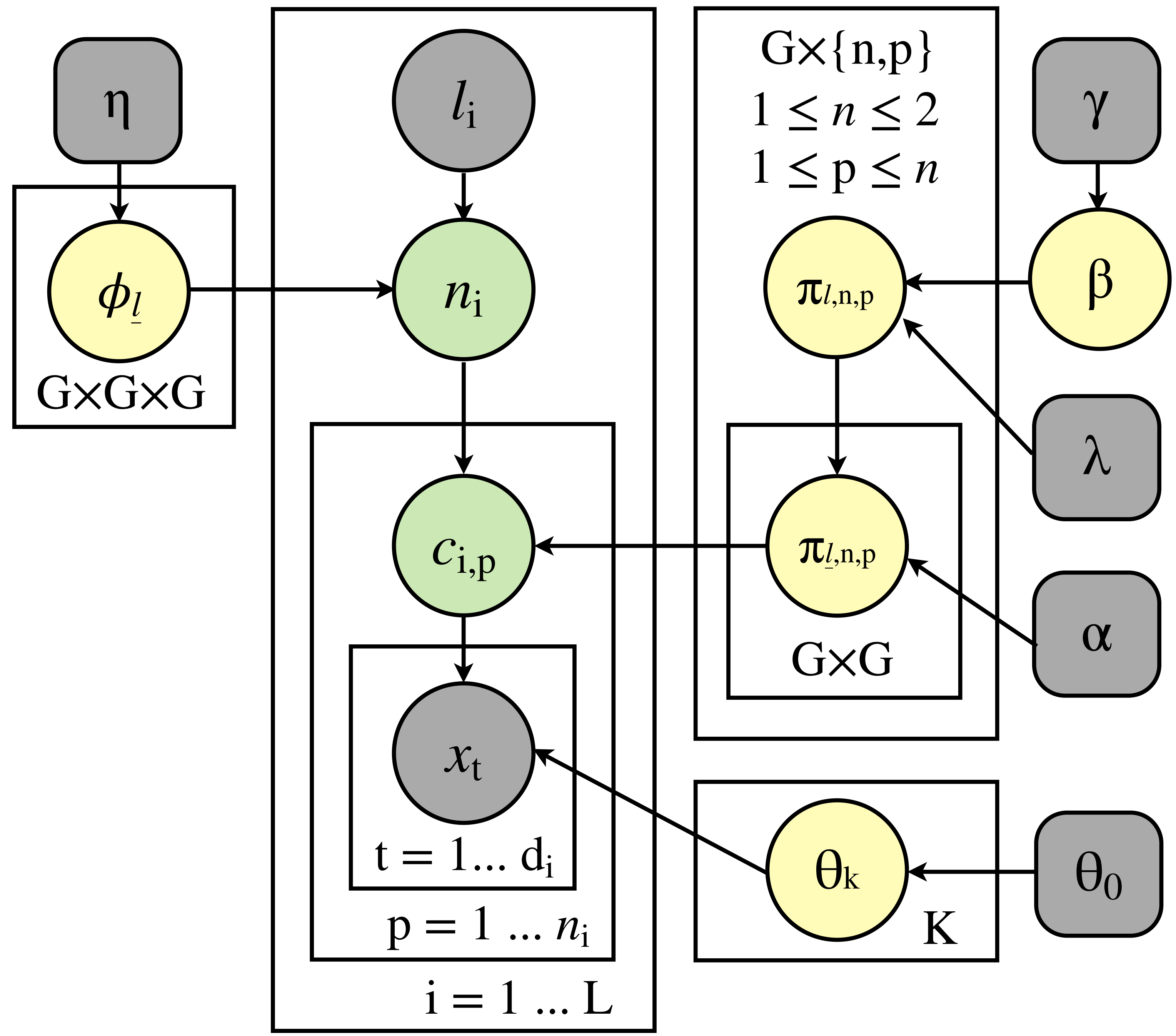
Latent model parameters

Regular latent variables

Inference

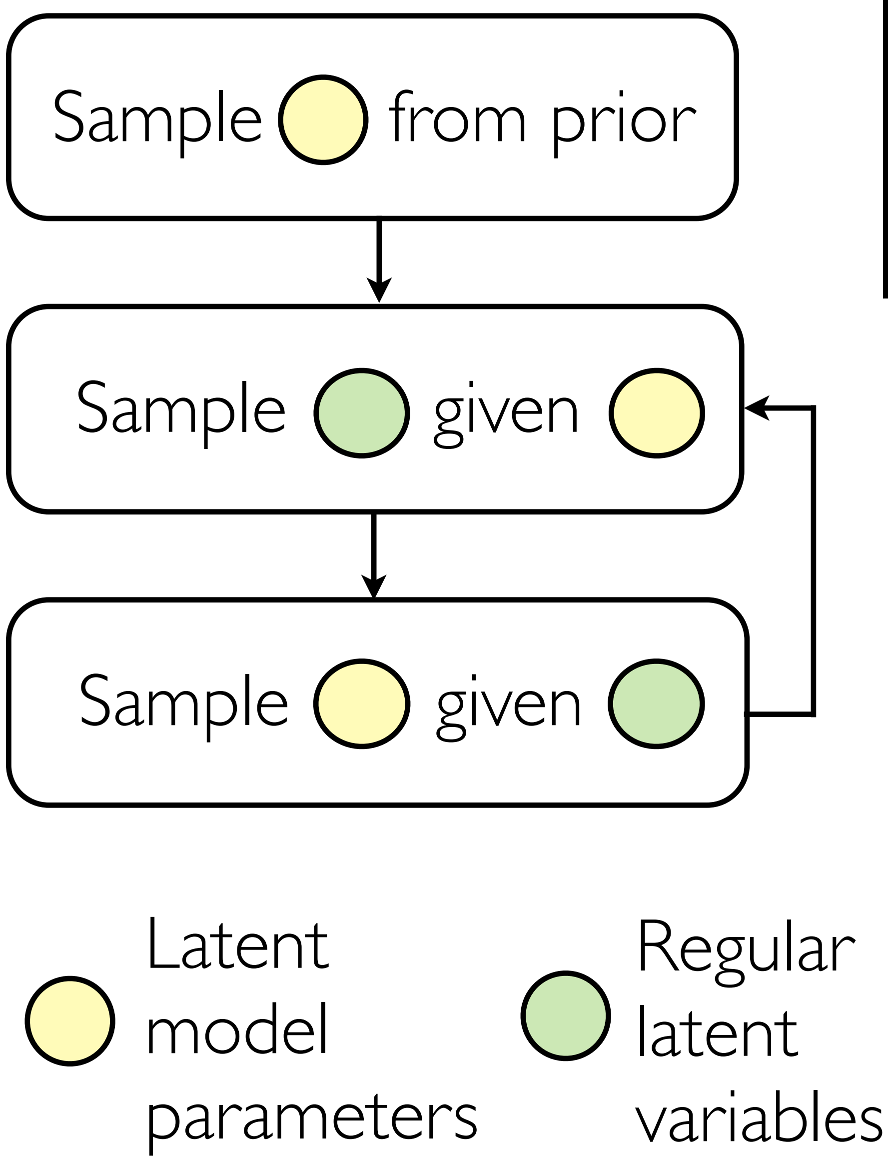
- **Procedure**

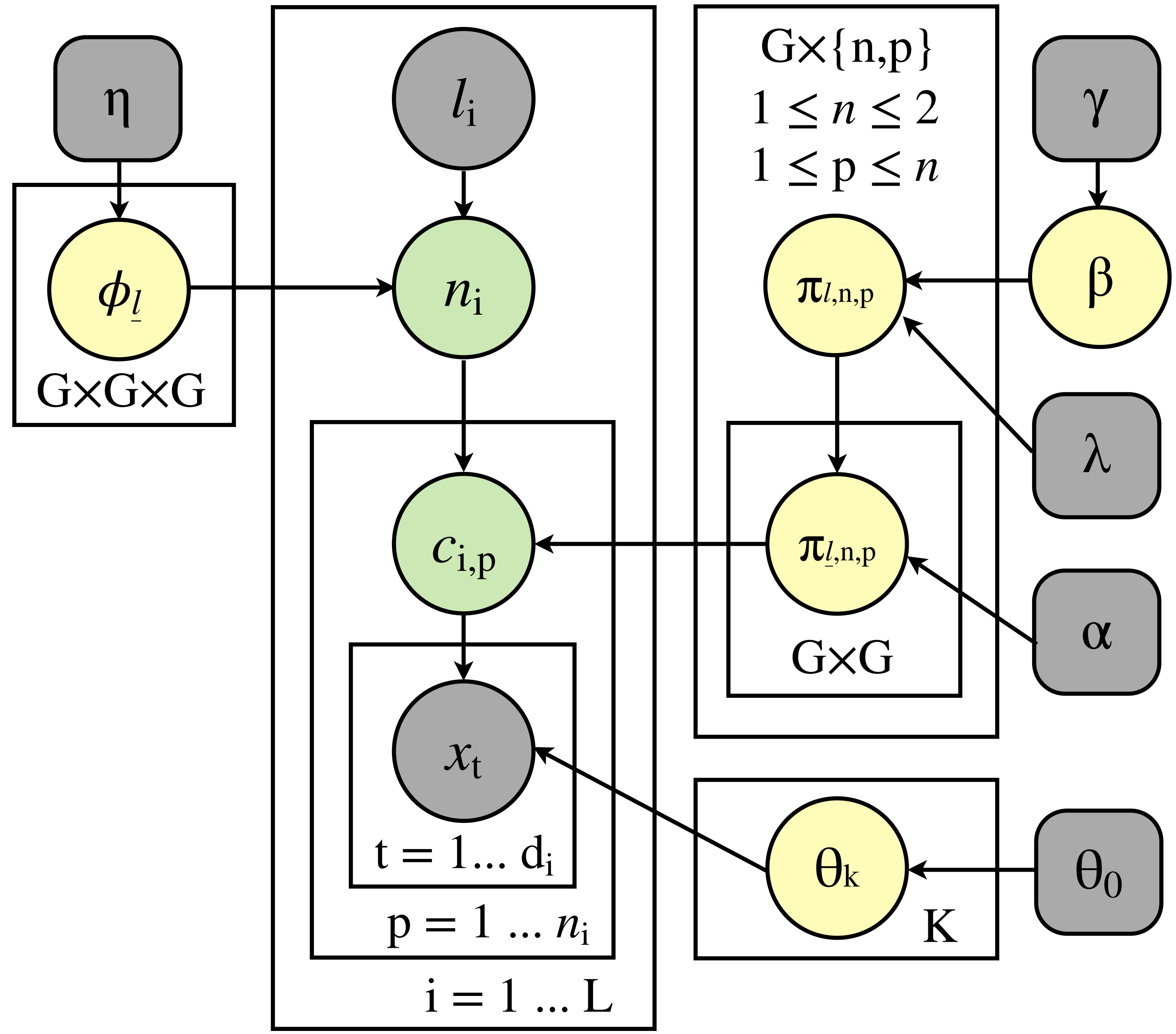




Inference

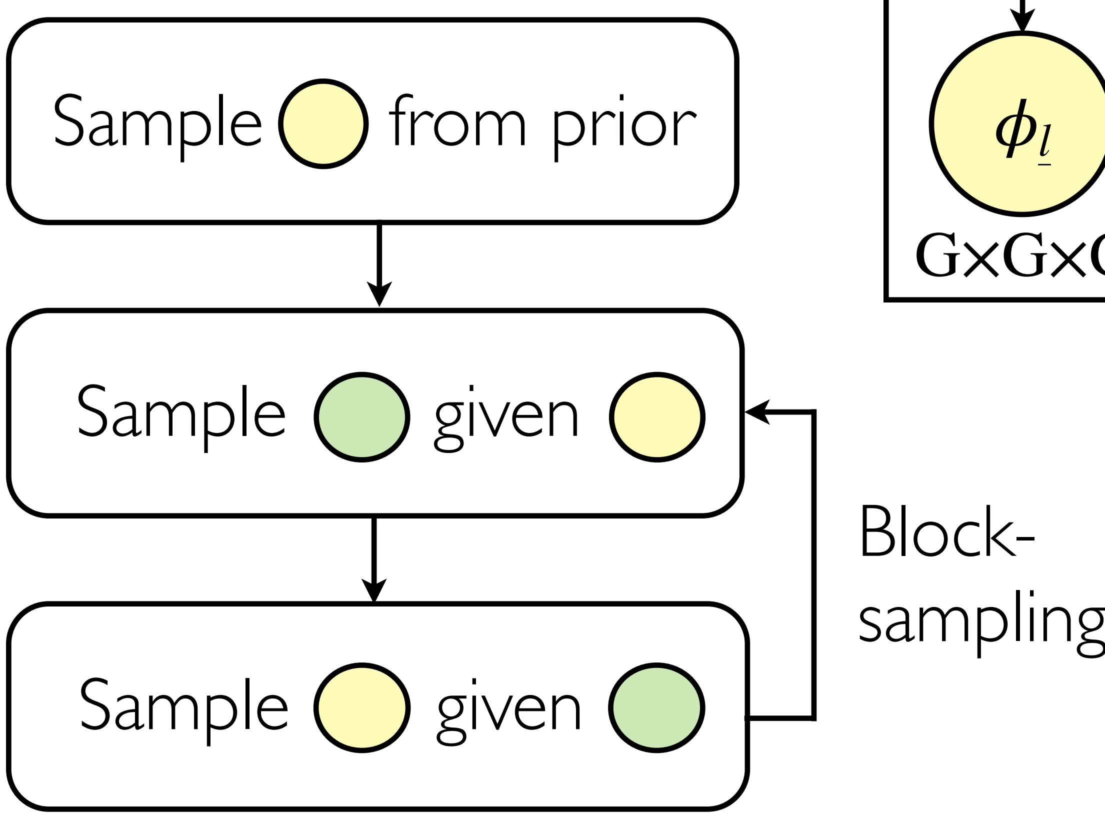
- Procedure

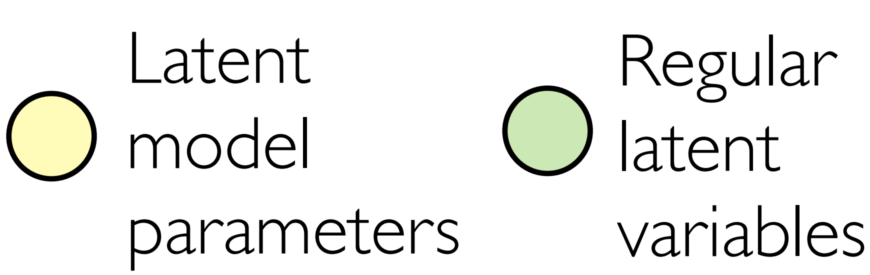


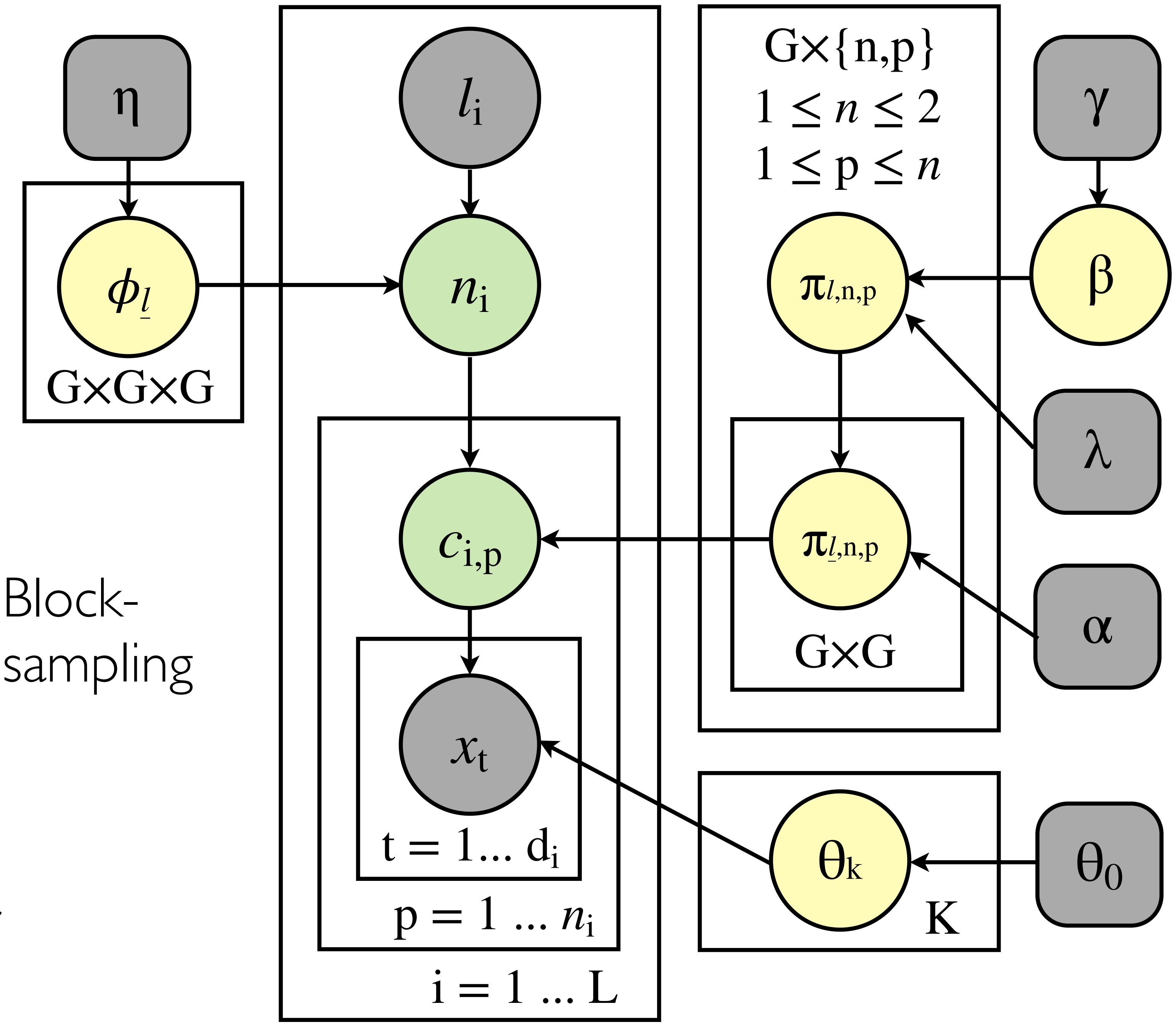


Inference

- **Procedure**
  - 10,000 iterations





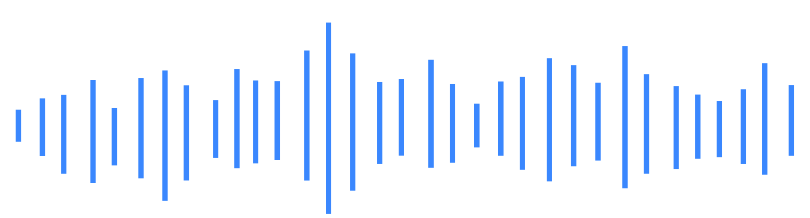


Induce Lexicon and Acoustic Model

- n_i and c_i define word pronunciations and phone transcriptions

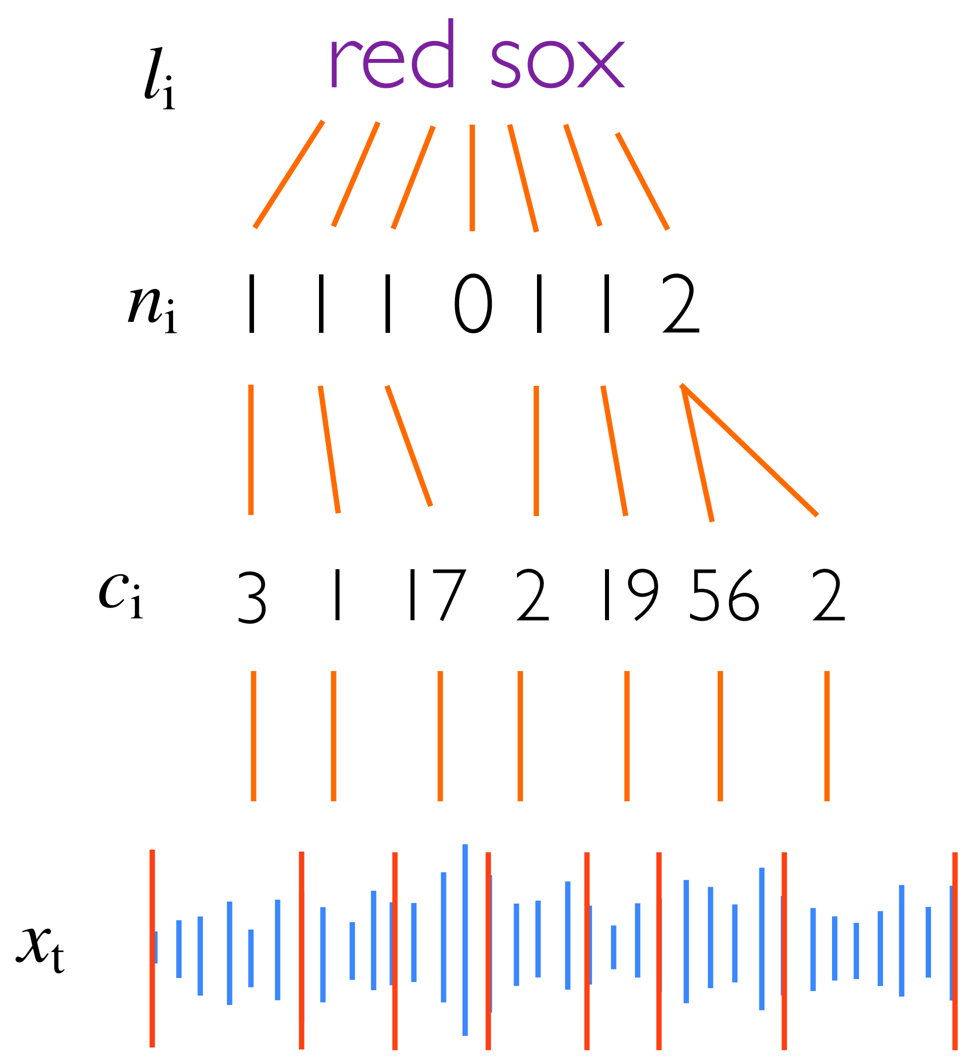
l_i red sox

x_t



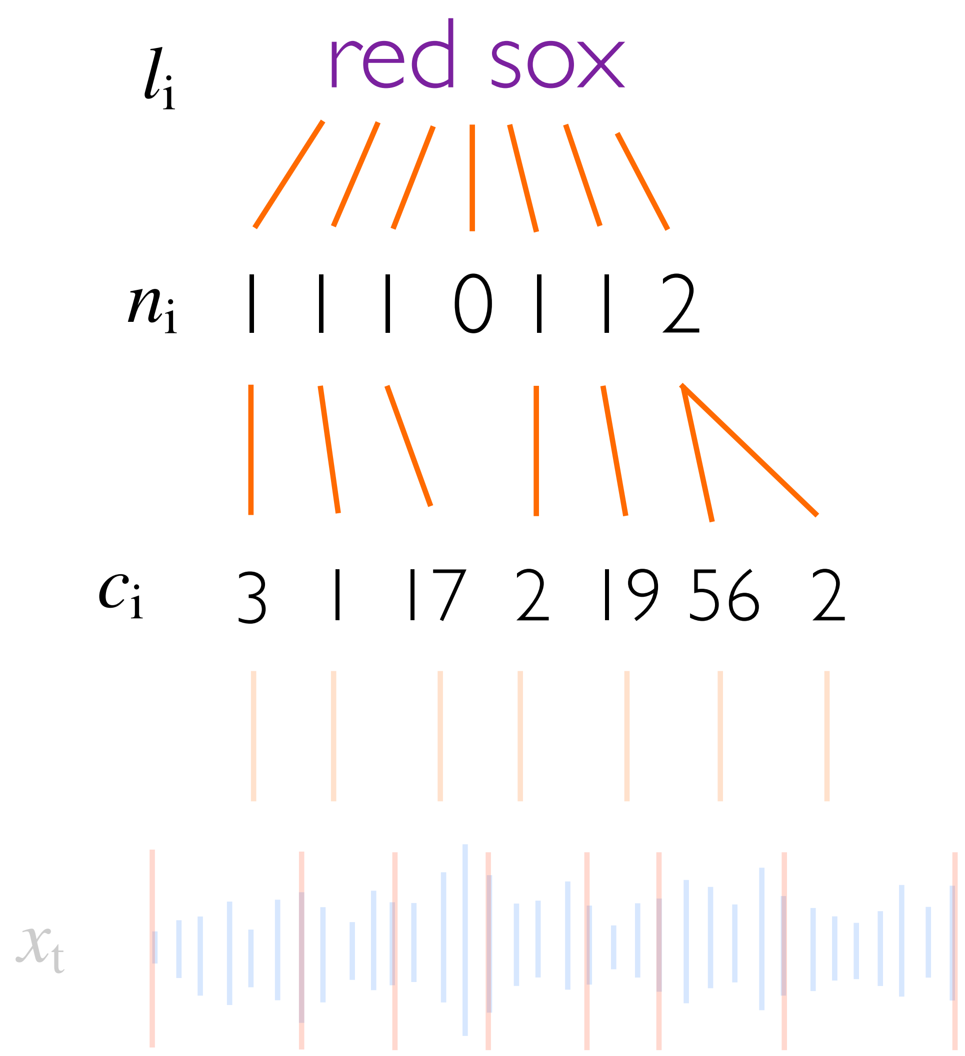
Induce Lexicon and Acoustic Model

- n_i and c_i define word pronunciations and phone transcriptions



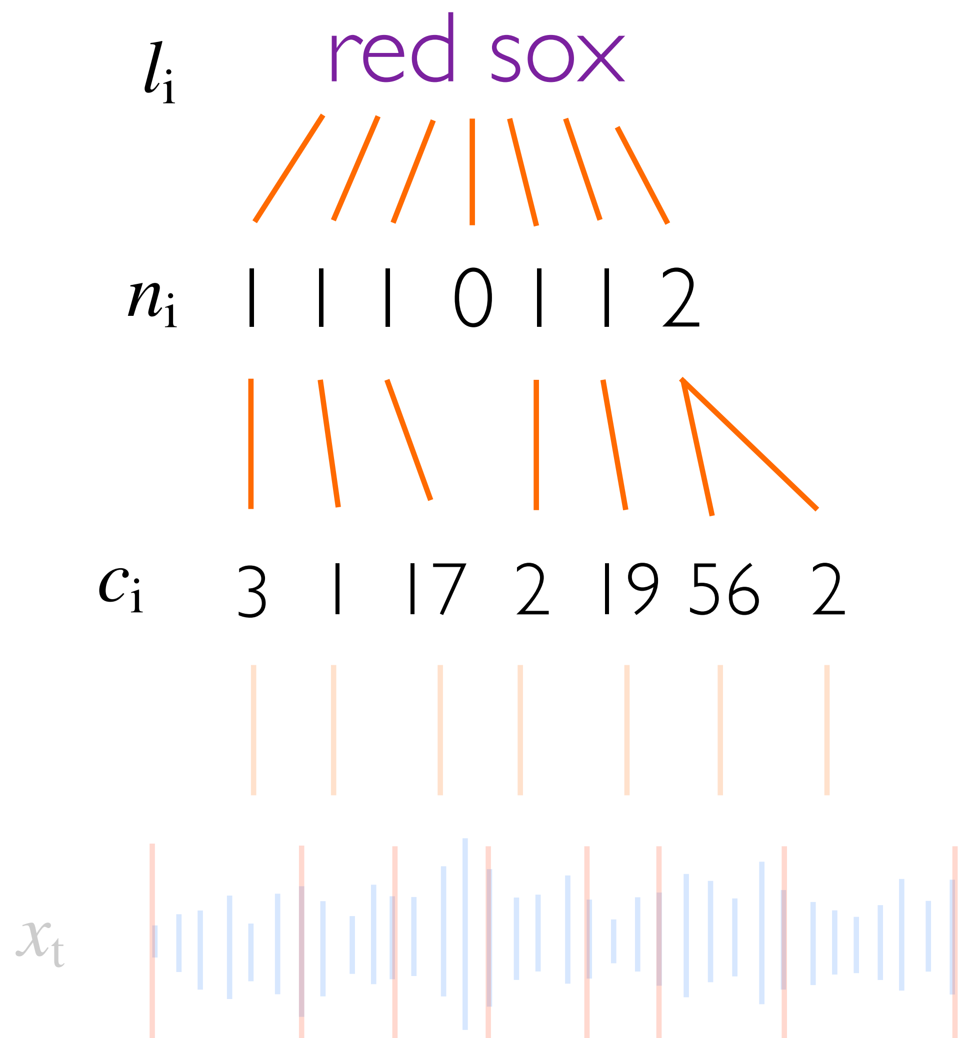
Induce Lexicon and Acoustic Model

- n_i and c_i define word pronunciations and phone transcriptions



Induce Lexicon and Acoustic Model

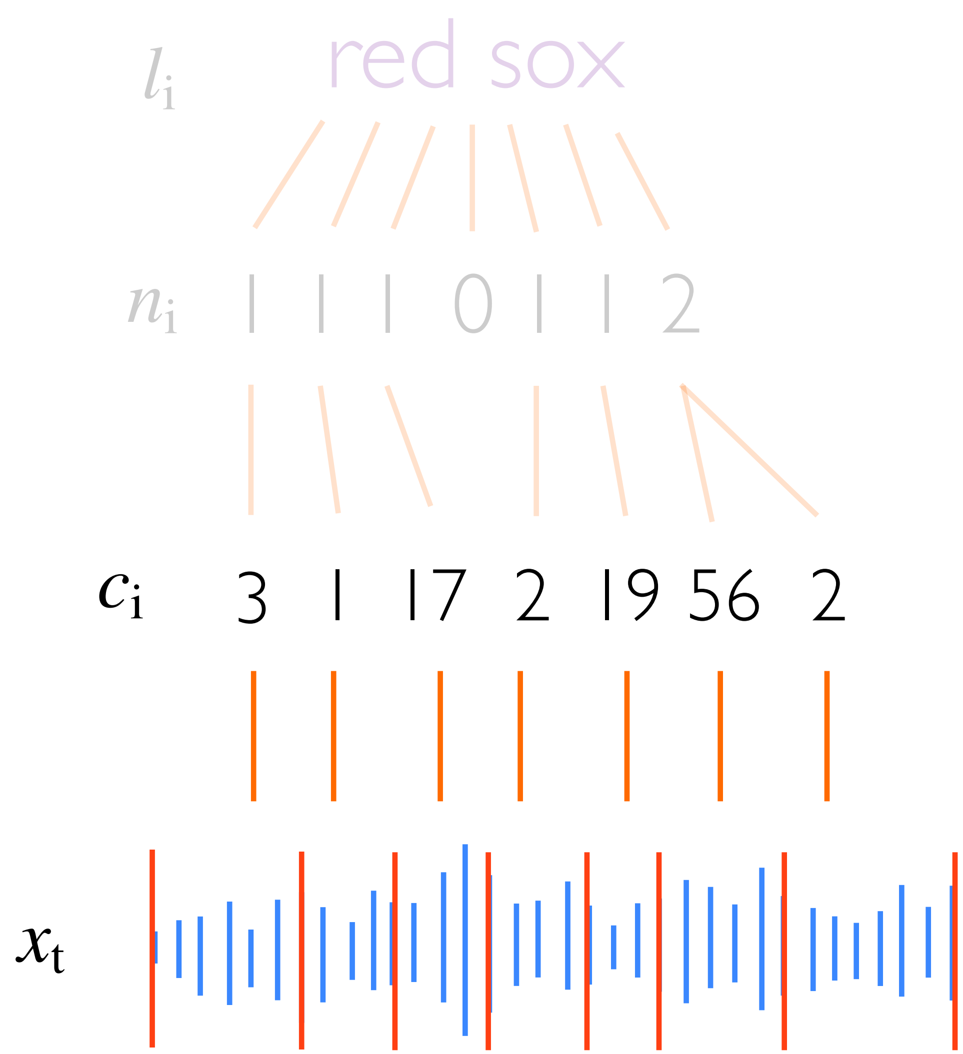
- n_i and c_i define word pronunciations and phone transcriptions



red : 3 1 17
sox : 2 19 56 2

Induce Lexicon and Acoustic Model

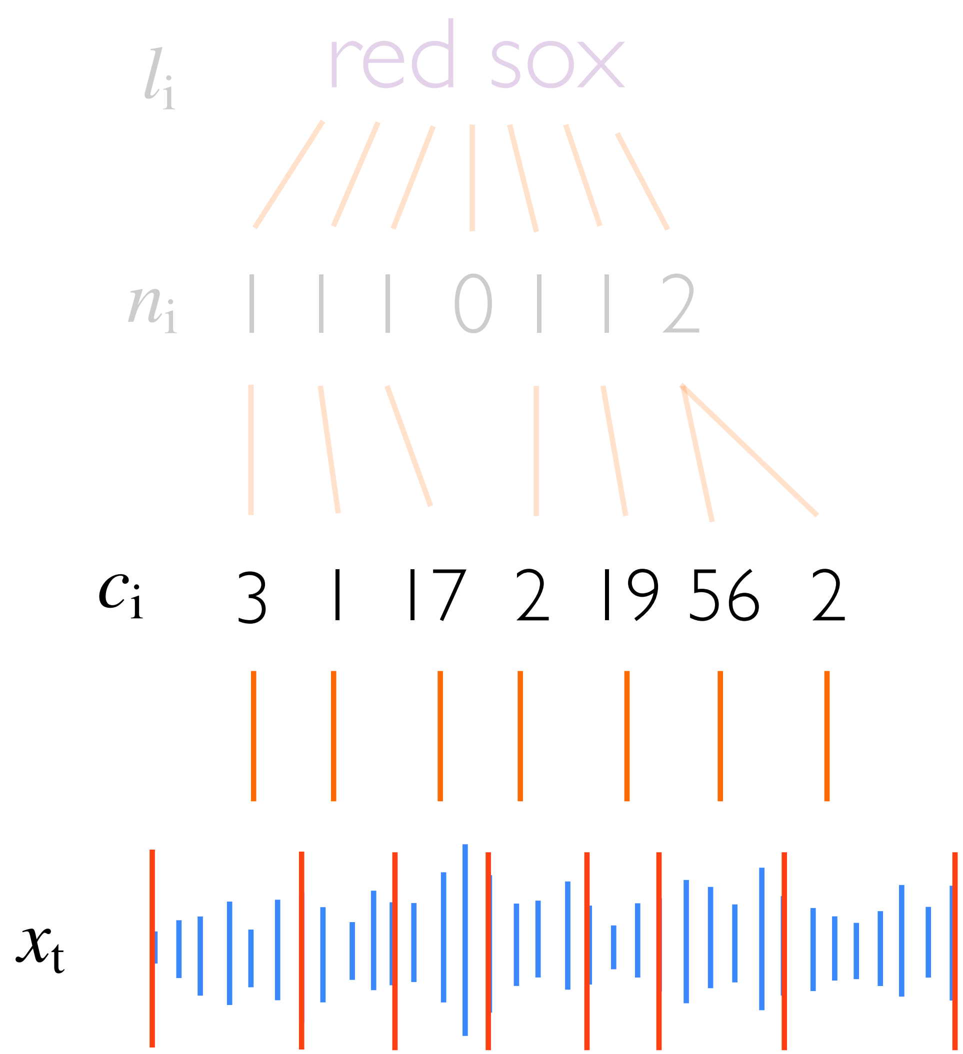
- n_i and c_i define word pronunciations and phone transcriptions



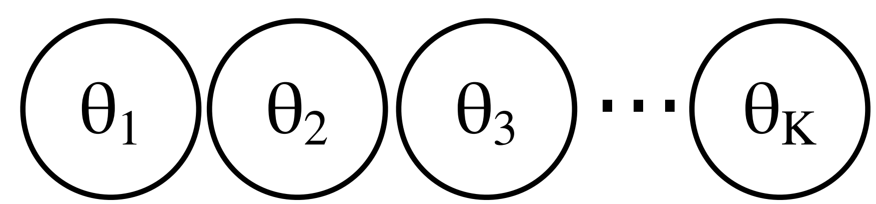
red : 3 1 17
sox : 2 19 56 2

Induce Lexicon and Acoustic Model

- n_i and c_i define word pronunciations and phone transcriptions



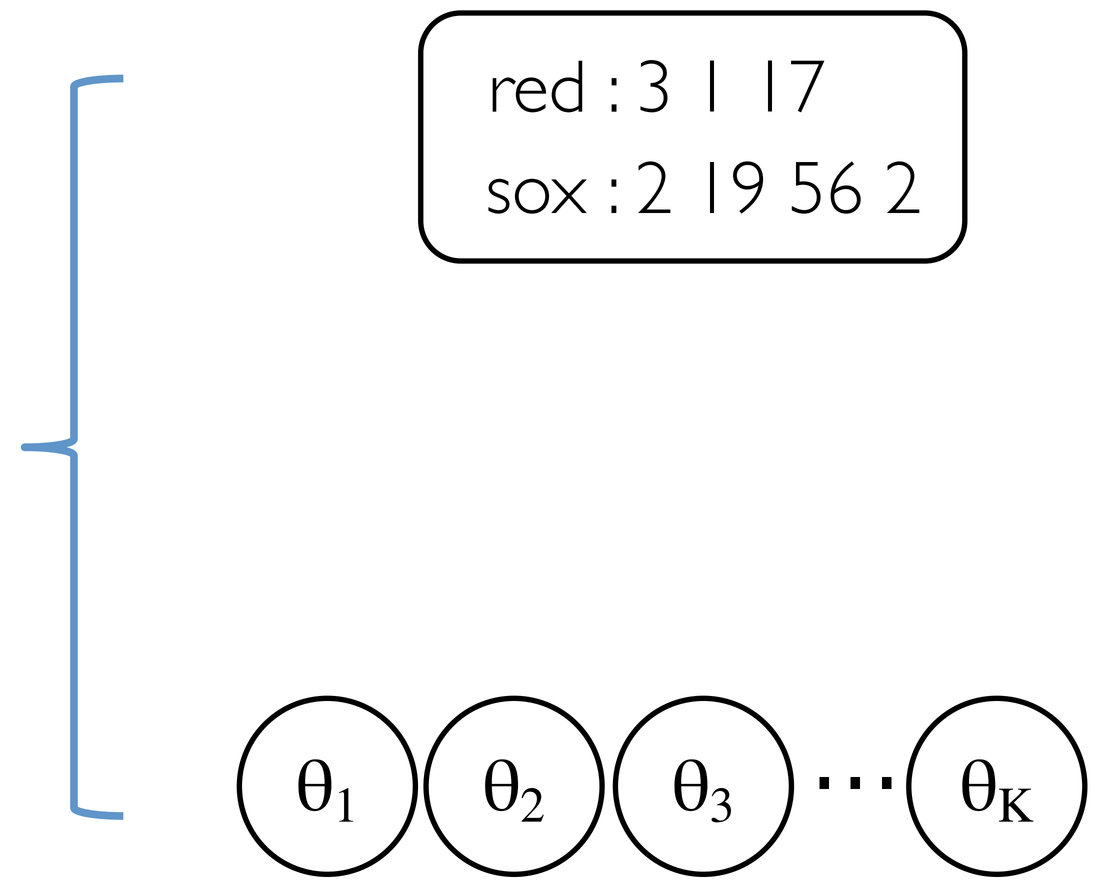
red : 3 1 17
sox : 2 19 56 2



Induce Lexicon and Acoustic Model

- n_i and c_i define word pronunciations and phone transcriptions

Train a speech recognizer



Experimental Setup

- **Dataset**
  - Jupiter [*Zue et al., IEEE Trans. on Speech and Audio Processing, 2000*]
  - Conversational telephone weather information queries
  - 72 hours of training data and 3.2 hours of test data
  - A subset of 8 hours of the training data used for training our model

Experimental Setup

- **Dataset**
  - Jupiter [*Zue et al., IEEE Trans. on Speech and Audio Processing, 2000*]
  - Conversational telephone weather information queries
  - 72 hours of training data and 3.2 hours of test data
  - A subset of 8 hours of the training data used for training our model
- **Benchmark and baseline**
  - A speech recognizer trained with an expert-crafted lexicon (Supervised)
  - A grapheme-based recognizer (Grapheme)

Experimental Setup

- Dataset
  - Jupiter [*Zue et al., IEEE Trans. on Speech and Audio Processing, 2000*]
  - Conversational telephone weather information queries
  - 72 hours of training data and 3.2 hours of test data
  - A subset of 8 hours of the training data used for training our model
- Benchmark and baseline
  - A speech recognizer trained with an expert-crafted lexicon (Supervised)
  - A grapheme-based recognizer (Grapheme)
- A 3-gram language model is used for all experiments

Results - Monophone Acoustic Model

- Word error rate (WER)

	WER (%)
Grapheme	32.7
Our model	17.0
Supervised	13.8

Results - Triphone Acoustic Model

- Word error rate (WER)
  - Singleton questions are used to build the decision trees

Results - Triphone Acoustic Model

- Word error rate (WER)
  - Singleton questions are used to build the decision trees

	WER (%)
Grapheme	15.7
Our model	13.4
Supervised	10.0

Related Work

- Word pronunciation learning
  - A segment model based approach to speech recognition [*Lee et al., ICASSP 1988*]
  - Lexicon-building methods for an acoustic sub-word based speech recognizer [*Paliwal, ICASSP 1990*]
  - Speech recognition based on acoustically derived segment units [*Fukuda et al., ICSLP 1996*]
  - Joint lexicon, acoustic unit inventory and model design [*Bacchiani and Ostendorf, Speech Communication 1999*]

Related Work

- **Word pronunciation learning**
  - A segment model based approach to speech recognition [*Lee et al., ICASSP 1988*]
  - Lexicon-building methods for an acoustic sub-word based speech recognizer [*Paliwal, ICASSP 1990*]
  - Speech recognition based on acoustically derived segment units [*Fukuda et al., ICSLP 1996*]
  - Joint lexicon, acoustic unit inventory and model design [*Bacchiani and Ostendorf, Speech Communication 1999*]
- **Grapheme recognizer**
  - Grapheme based speech recognition [*Killer et al., Eurospeech 2003*]
  - A grapheme based speech recognizer for Russian [*Stuker and Schultz, SPECOM 2004*]

Conclusion

- A joint learning framework for discovering pronunciation lexicon and acoustic model
  - Phonetic units are modeled by a HMM-based mixture model
  - L2S mapping rules are captured by weights over mixtures
  - L2S rules are tied together through a hierarchical structure

Conclusion

- **A joint learning framework for discovering pronunciation lexicon and acoustic model**
  - Phonetic units are modeled by a HMM-based mixture model
  - L2S mapping rules are captured by weights over mixtures
  - L2S rules are tied together through a hierarchical structure
- **Automatic speech recognition experiments**
  - Outperforms a grapheme-based speech recognizer
  - Approaches the performance of a recognizer trained with an expert lexicon

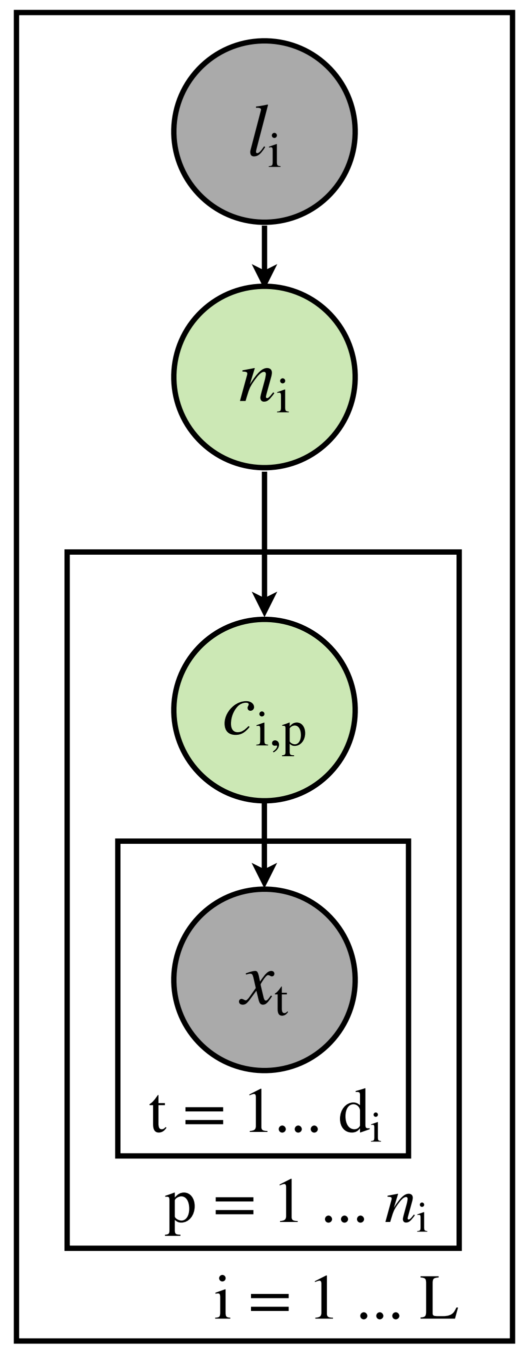
Conclusion

- A joint learning framework for discovering pronunciation lexicon and acoustic model
  - Phonetic units are modeled by a HMM-based mixture model
  - L2S mapping rules are captured by weights over mixtures
  - L2S rules are tied together through a hierarchical structure
- Automatic speech recognition experiments
  - Outperforms a grapheme-based speech recognizer
  - Approaches the performance of a recognizer trained with an expert lexicon
- Apply the lexicon and phone units to existing ASR training methods
  - Use our model as an initialization

Thank you.

Sample n_{i} and $c_{\mathrm{i,p}}$

- n_{i} and $c_{\mathrm{i,p}}$ denote an alignment between text and speech



Sample n_i and $c_{i,p}$

- n_i and $c_{i,p}$ denote an alignment between text and speech
- Sample a new alignment

Sample n_i and $c_{i,p}$

- n_i and $c_{i,p}$ denote an alignment between text and speech
- Sample a new alignment
  - Compute the probabilities of all possible alignments

Sample n_i and $c_{i,p}$

- n_i and $c_{i,p}$ denote an alignment between text and speech
- Sample a new alignment
  - Compute the probabilities of all possible alignments
  - Backward message passing with dynamic programming

Sample n_i and $c_{i,p}$

- n_i and $c_{i,p}$ denote an alignment between text and speech
- Sample a new alignment
  - Compute the probabilities of all possible alignments
  - Backward message passing with dynamic programming
  - Forward block-sample new n_i and $c_{i,p}$

Sample n_i and $c_{i,p}$

- n_i and $c_{i,p}$ **denote an alignment between text and speech**
- **Sample a new alignment**
  - Compute the probabilities of all possible alignments
  - Backward message passing with dynamic programming
  - Forward block-sample new n_i and $c_{i,p}$
  - Similar to inference for hidden semi-Markov models

Refine Induced Lexicon

- **Pronunciations of** *Burma*

pronunciation (b)	p(b)
93 56 87 39 19	0.125
93 56 61 87 73 99	0.125
11 56 61 87 73 99	0.125
93 20 75 87 17 27 52	0.125
55 93 56 61 87 73 84 19	0.125
93 26 61 87 49	0.125
63 83 86 87 73 53 19	0.125
93 26 61 87 61	0.125

$$\sum_{b \in B(w)} p(b) log p(b)$$

$B(w)$: all pronunciations of a word

$p(b)$: pronunciation probability

Refine Induced Lexicon

- **Pronunciations of** *Burma*

pronunciation (b)	p(b)
93 56 87 39 19	0.125
93 56 61 87 73 99	0.125
11 56 61 87 73 99	0.125
93 20 75 87 17 27 52	0.125
55 93 56 61 87 73 84 19	0.125
93 26 61 87 49	0.125
63 83 86 87 73 53 19	0.125
93 26 61 87 61	0.125

$$\sum_{b \in B(w)} p(b) log p(b)$$

$B(w)$: all pronunciations of a word
$p(b)$: pronunciation probability
V : vocabulary of the data

Refine Induced Lexicon

- **Pronunciations of** *Burma*

pronunciation (b)	p(b)
93 56 87 39 19	0.125
93 56 61 87 73 99	0.125
11 56 61 87 73 99	0.125
93 20 75 87 17 27 52	0.125
55 93 56 61 87 73 84 19	0.125
93 26 61 87 49	0.125
63 83 86 87 73 53 19	0.125
93 26 61 87 61	0.125

$$H \equiv \frac{-1}{|V|} \sum_{w \in V} \sum_{b \in B(w)} p(b) log p(b)$$

$B(w)$: all pronunciations of a word

$p(b)$: pronunciation probability

V : vocabulary of the data

Refine Induced Lexicon

- **Pronunciations of** *Burma*

pronunciation (b)	p(b)
93 56 87 39 19	0.125
93 56 61 87 73 99	0.125
11 56 61 87 73 99	0.125
93 20 75 87 17 27 52	0.125
55 93 56 61 87 73 84 19	0.125
93 26 61 87 49	0.125
63 83 86 87 73 53 19	0.125
93 26 61 87 61	0.125
Average entropy (H)	4.58

Refine Induced Lexicon

- **Pronunciations of** *Burma*

pronunciation (b)	p(b)
93 56 87 39 19	0.125
93 56 61 87 73 99	0.125
11 56 61 87 73 99	0.125
93 20 75 87 17 27 52	0.125
55 93 56 61 87 73 84 19	0.125
93 26 61 87 49	0.125
63 83 86 87 73 53 19	0.125
93 26 61 87 61	0.125
Average entropy (H)	4.58
WER (%)	17.0

Refine Induced Lexicon

- **Pronunciations of** *Burma*

pronunciation (b)	pronunciation probabilities		
	Our model	+1 PMM*	+2 PMM*
93 56 87 39 19	0.125		
93 56 61 87 73 99	0.125		
11 56 61 87 73 99	0.125		
93 20 75 87 17 27 52	0.125		
55 93 56 61 87 73 84 19	0.125		
93 26 61 87 49	0.125		
63 83 86 87 73 53 19	0.125		
93 26 61 87 61	0.125		
Average entropy (H)	4.58		
WER (%)	17.0		

*Learning lexicon from speech using a pronunciation mixture model [*McGraw et al., 2013*]

Refine Induced Lexicon

- **Pronunciations of** *Burma*

pronunciation (b)	pronunciation probabilities		
	Our model	+1 PMM*	+2 PMM*
93 56 87 39 19	0.125	-	-
93 56 61 87 73 99	0.125	-	-
11 56 61 87 73 99	0.125	0.400	0.419
93 20 75 87 17 27 52	0.125	0.125	0.124
55 93 56 61 87 73 84 19	0.125	0.220	0.210
93 26 61 87 49	0.125	0.128	0.140
63 83 86 87 73 53 19	0.125	-	-
93 26 61 87 61	0.125	0.127	0.107
Average entropy (H)	4.58		
WER (%)	17.0		

*Learning lexicon from speech using a pronunciation mixture model [*McGraw et al., 2013*]

Refine Induced Lexicon

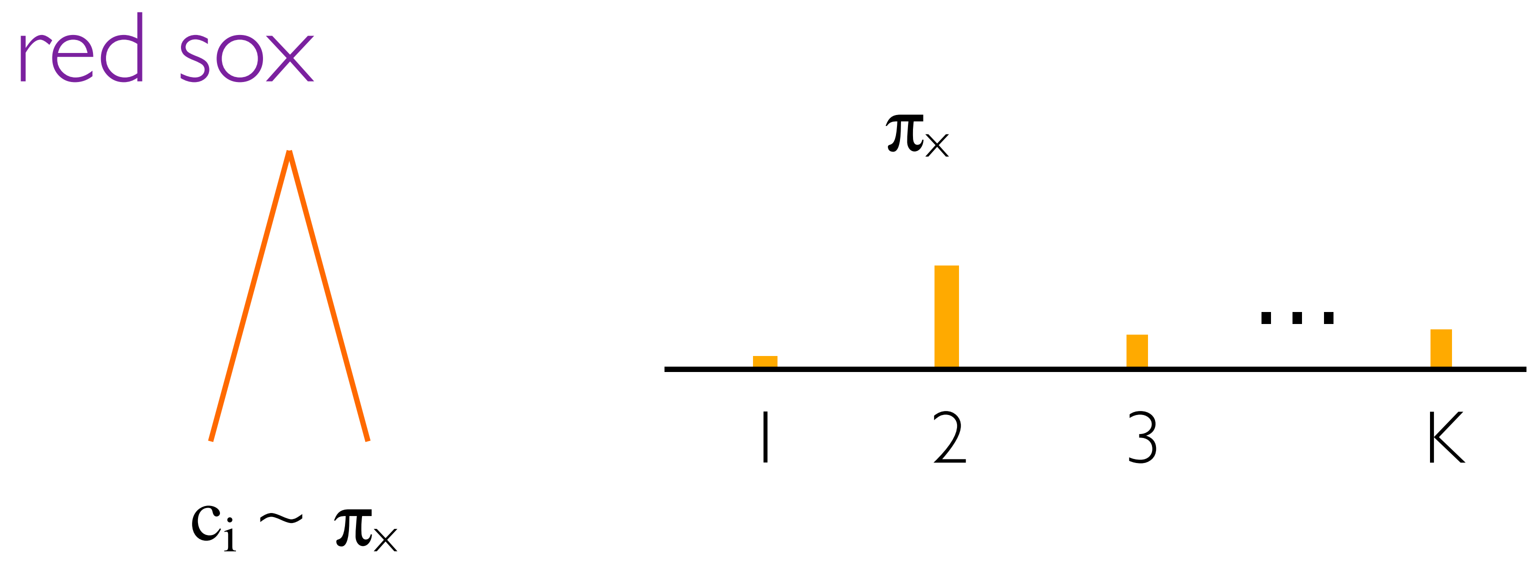
- **Pronunciations of** *Burma*

pronunciation (b)	pronunciation probabilities		
	Our model	+1 PMM*	+2 PMM*
93 56 87 39 19	0.125	-	-
93 56 61 87 73 99	0.125	-	-
11 56 61 87 73 99	0.125	0.400	0.419
93 20 75 87 17 27 52	0.125	0.125	0.124
55 93 56 61 87 73 84 19	0.125	0.220	0.210
93 26 61 87 49	0.125	0.128	0.140
63 83 86 87 73 53 19	0.125	-	-
93 26 61 87 61	0.125	0.127	0.107
Average entropy (H)	4.58	3.47	3.03
WER (%)	17.0	16.6	15.9

*Learning lexicon from speech using a pronunciation mixture model [*McGraw et al., 2013*]

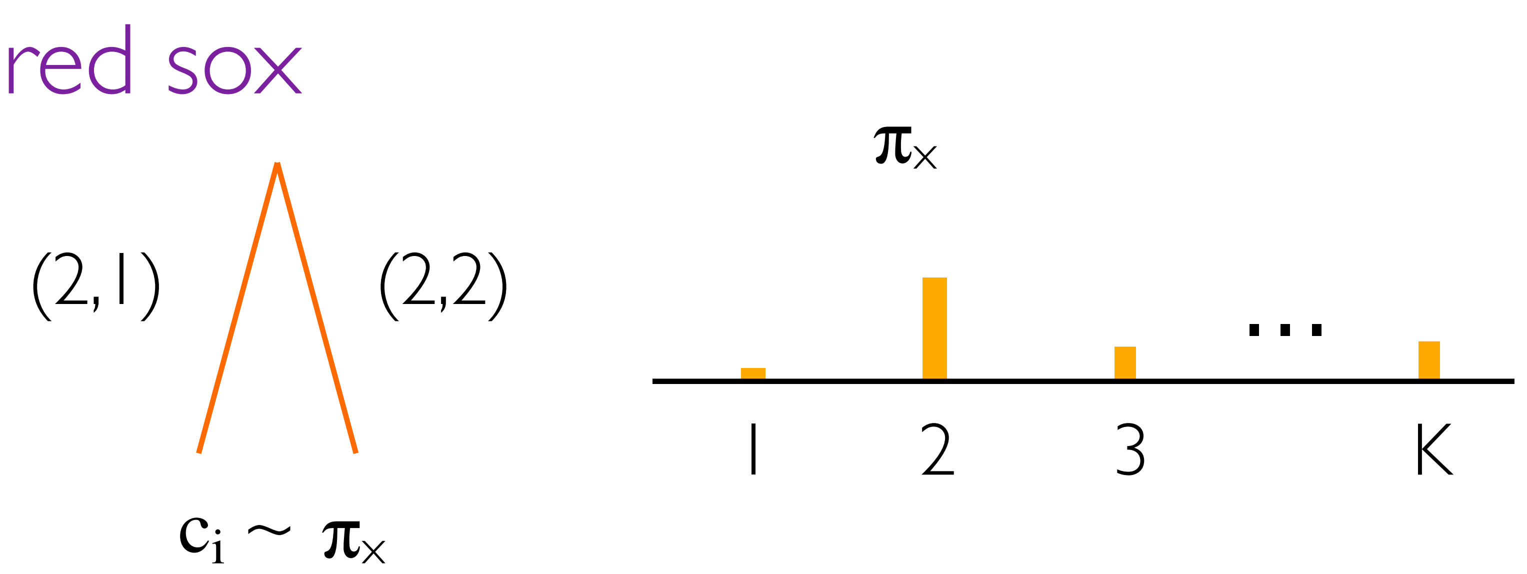
Position-dependent L2S Rules

- Take phone position into account



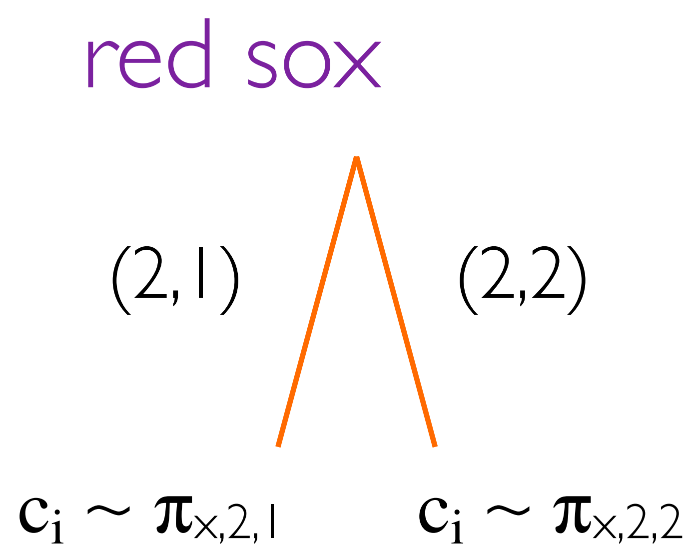
Position-dependent L2S Rules

- Take phone position into account



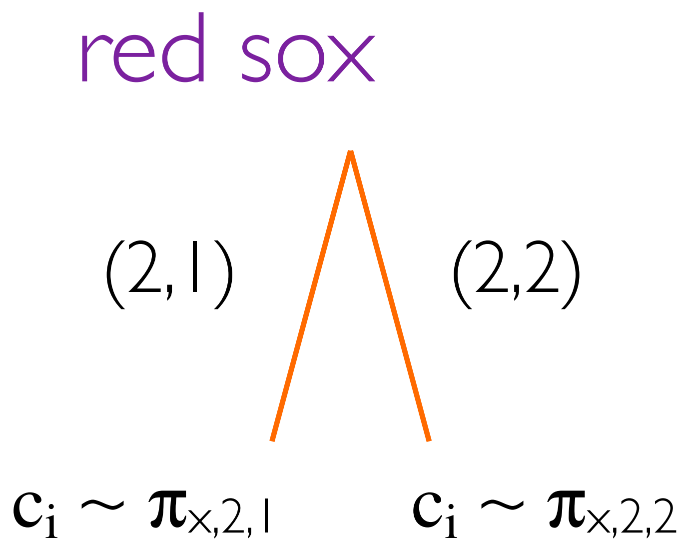
Position-dependent L2S Rules

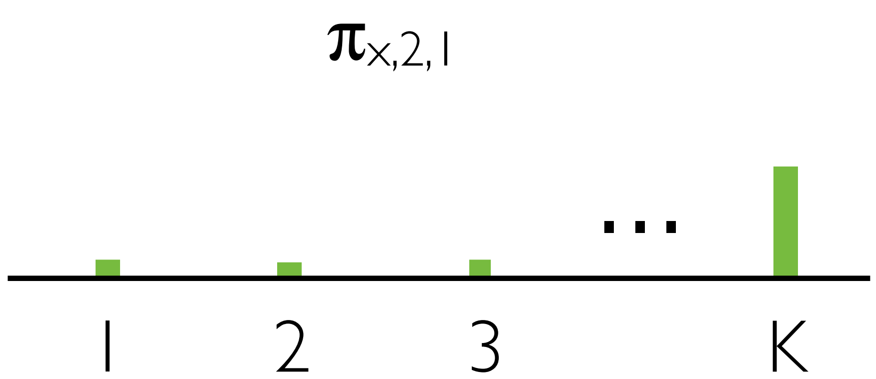
- Take phone position into account



Position-dependent L2S Rules

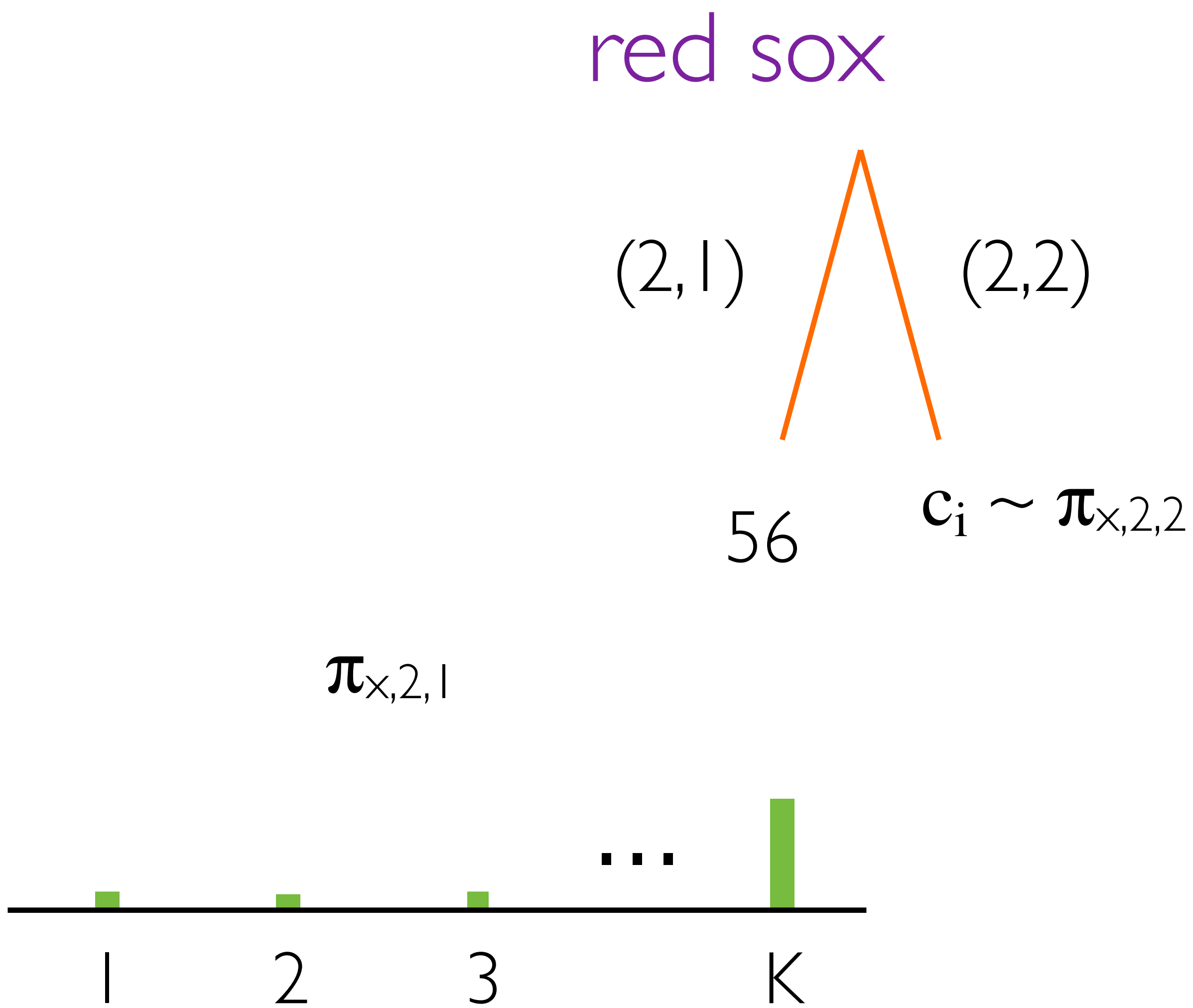
- Take phone position into account





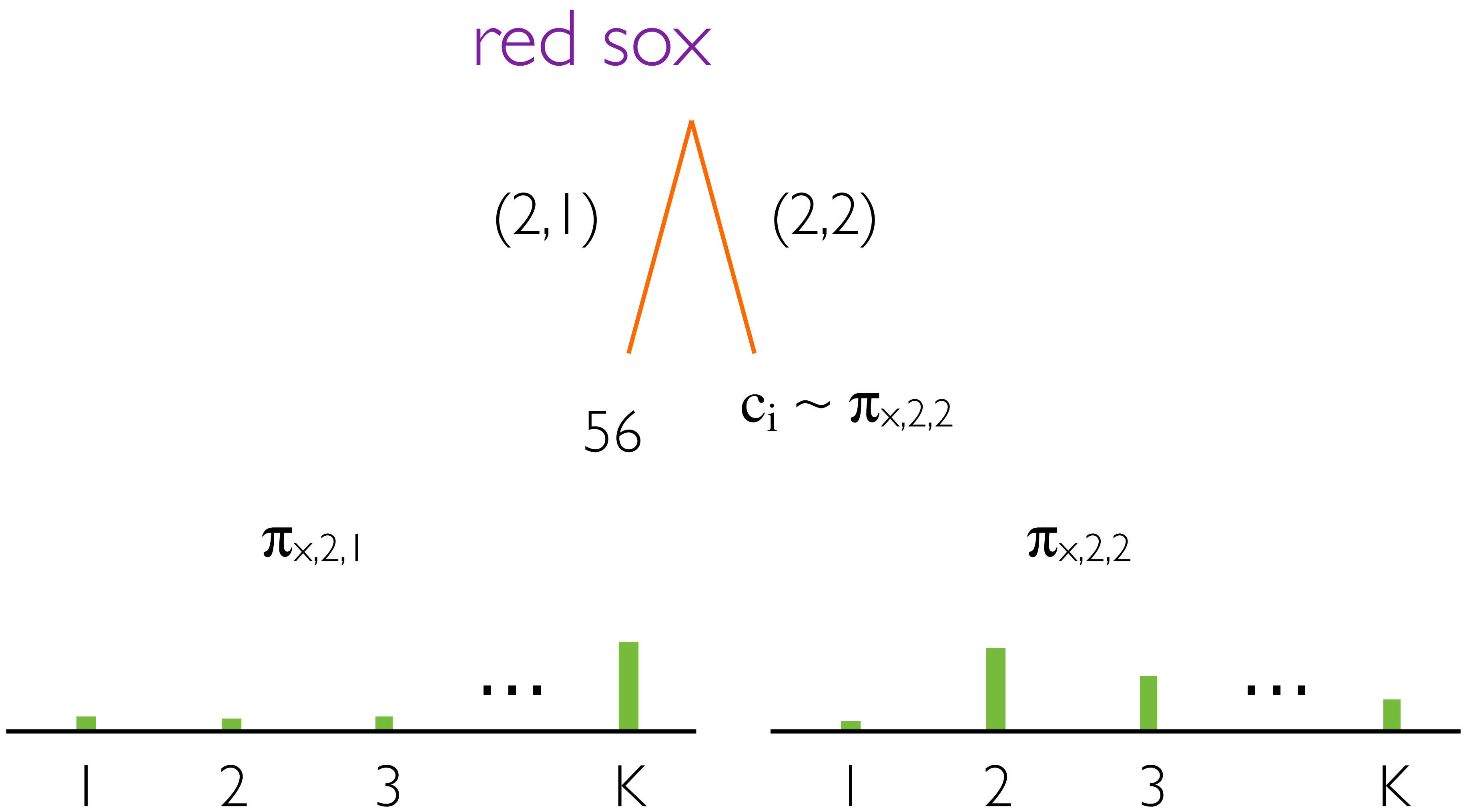
Position-dependent L2S Rules

- Take phone position into account



Position-dependent L2S Rules

- Take phone position into account



Position-dependent L2S Rules

- Take phone position into account