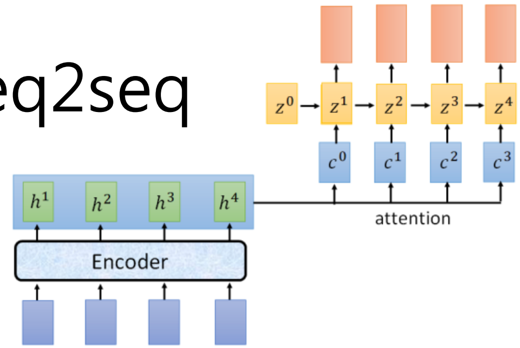
Speech Recognition

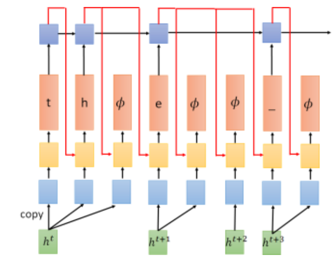
HUNG-YI LEE 李宏毅

Last Time

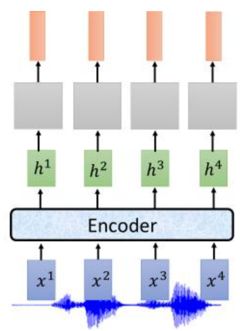
LAS: 就是 seq2seq



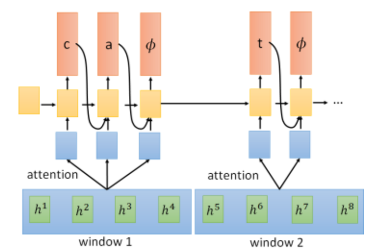
RNN-T: 輸入一個東西可以輸出多個東西的 seq2seq



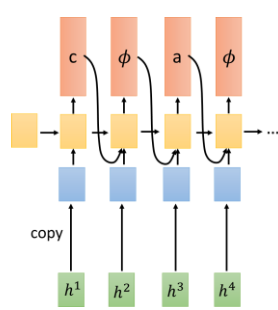
CTC: decoder 是 linear classifier 的 seq2seq



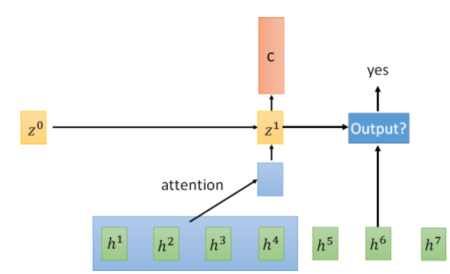
Neural Transducer: 每次輸入一個 window 的 RNN-T



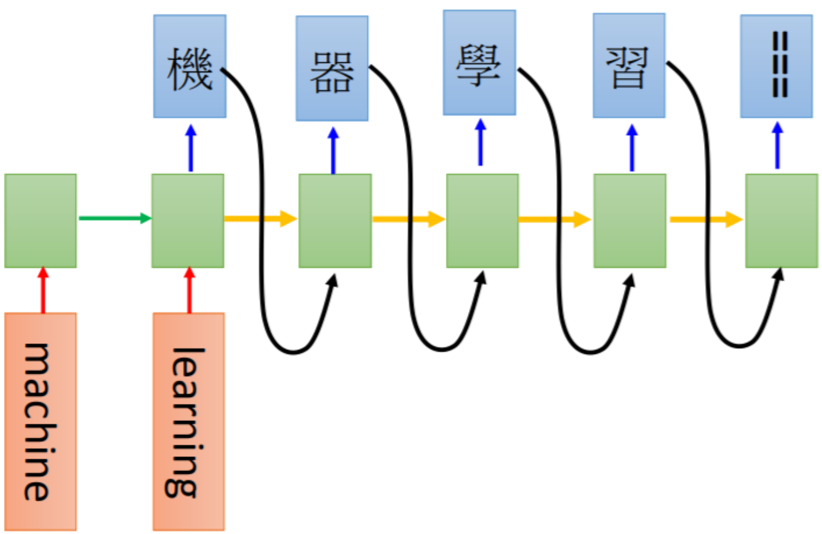
RNA: 輸入一個東西就要輸出一個東西的 seq2seq



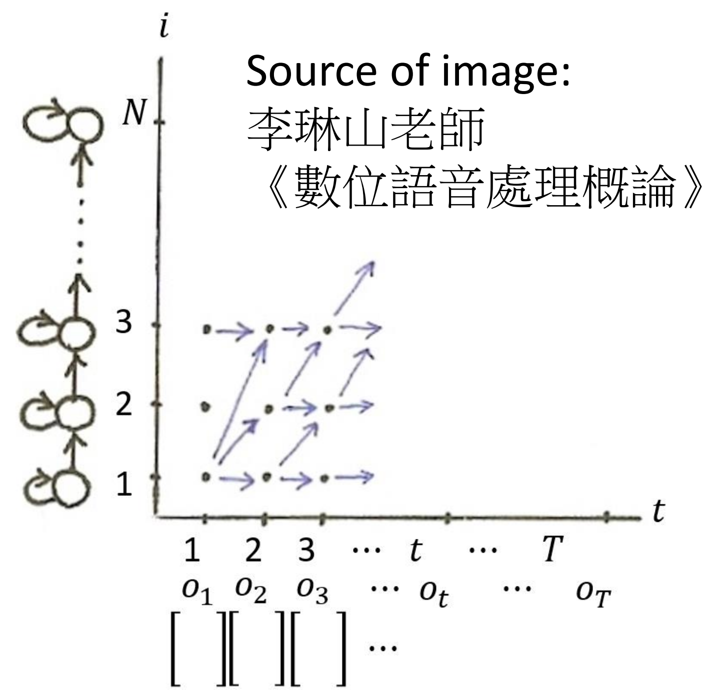
MoCha: window 移動伸縮自如的 Neural Transducer



Two Points of Views

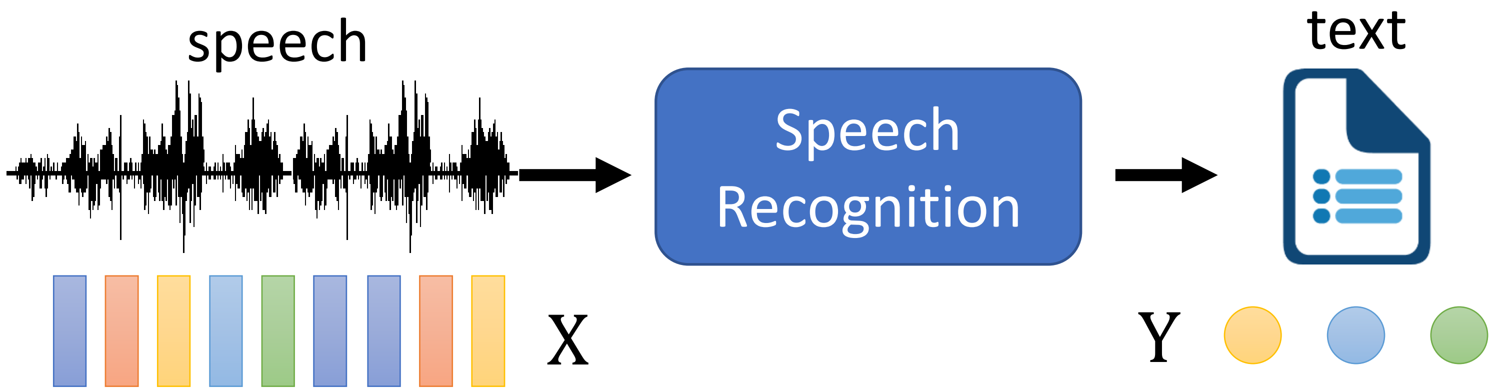


Seq-to-seq



HMM

Hidden Markov Model (HMM)



$$Y^* = arg \max_{Y} P(Y|X)$$

Decode

$$= arg \max_{Y} \frac{P(X|Y)P(Y)}{P(X)}$$

$$= arg \max_{Y} P(X|Y)P(Y)$$

$P(X|Y)$: HMM

Acoustic Model

$P(Y)$:

Language Model

HMM

$$P(X|Y) \longrightarrow P(X|S)$$

A token sequence Y corresponds to a sequence of **states** S

what do you think

Phoneme:

hh w aa t d uw y uw th ih ng k

Tri-phone:

…… t-d+uw d-uw+y uw-y+uw y-uw+th ……

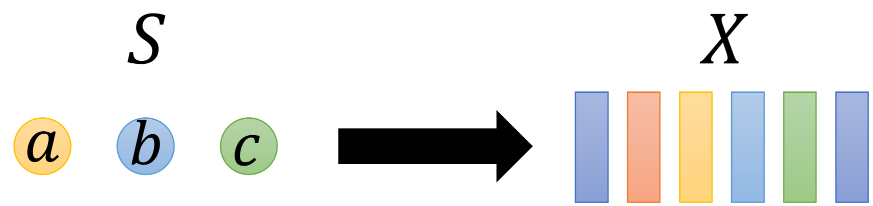
t-d+uw1 t-d+uw2 t-d+uw3 d-uw+y1 d-uw+y2 d-uw+y3

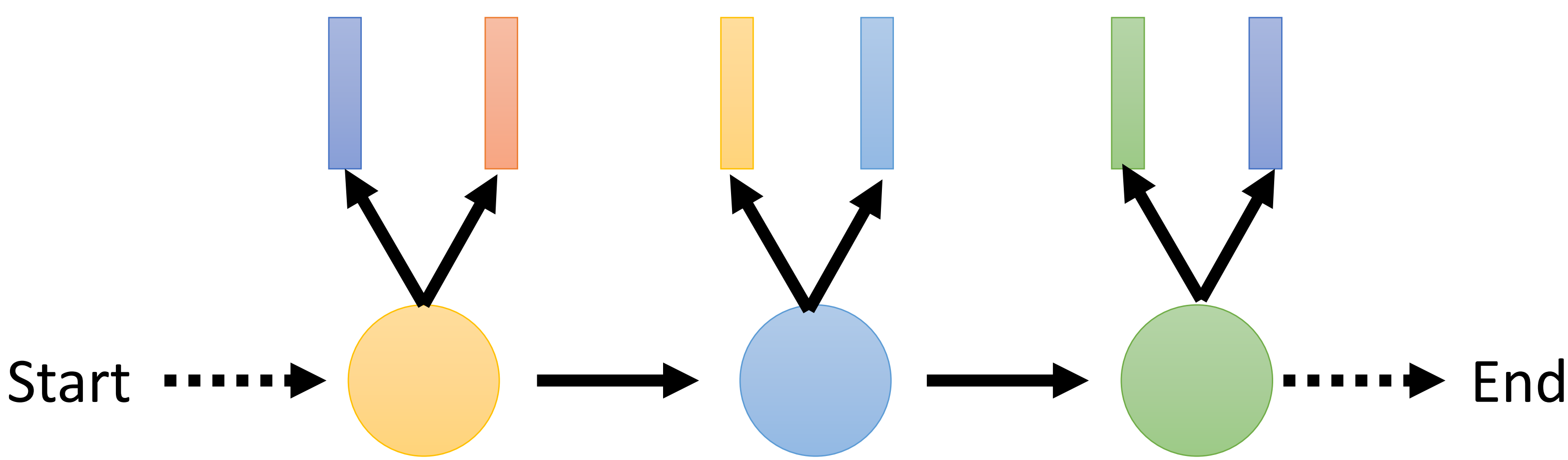
State:

HMM

$$P(X|Y) \longrightarrow P(X|S)$$

A sentence Y corresponds to a sequence of **states** S





HMM

$P(X|Y) \longrightarrow P(X|S)$

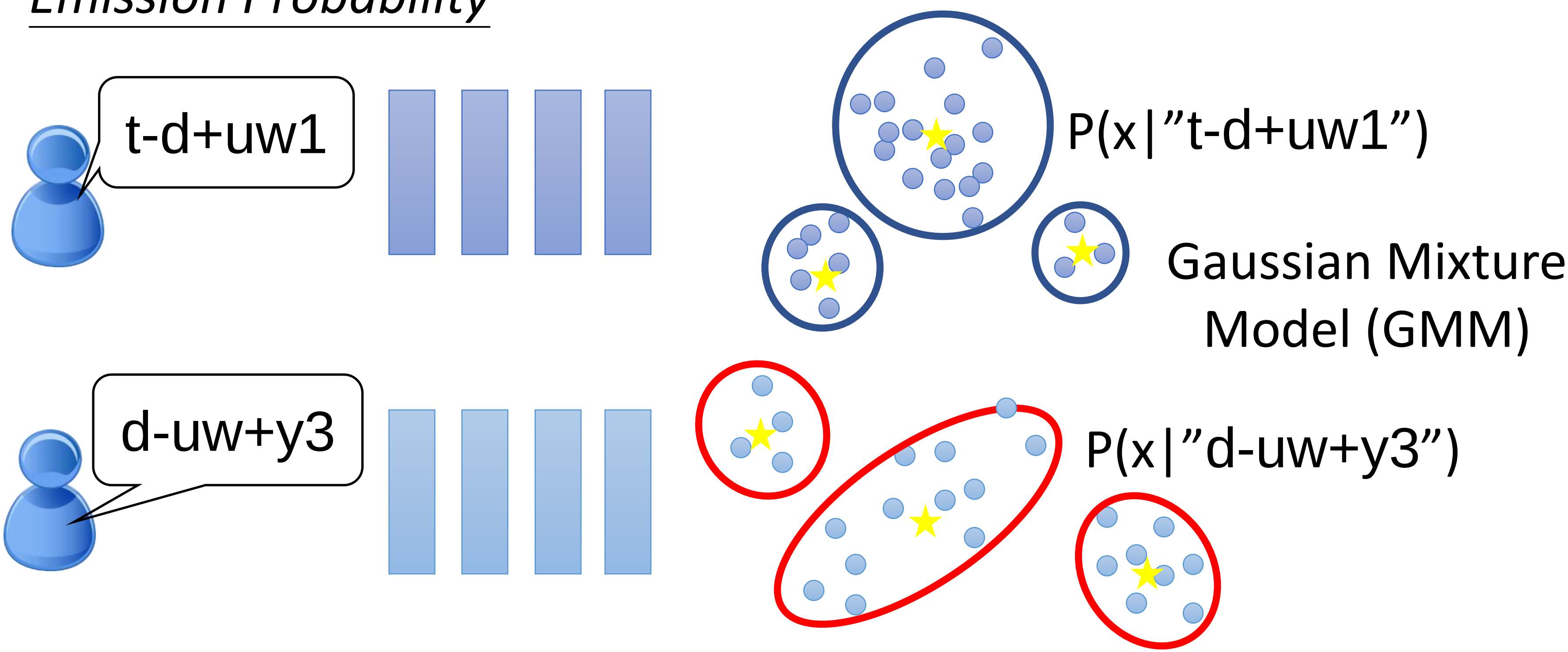
A sentence Y corresponds to a sequence of **states** S

Transition Probability

Probability from one state to another

$a \rightarrow b$ $\quad p(b|a)$

Emission Probability



HMM – Emission Probability

- Too many states ……

Tied-state

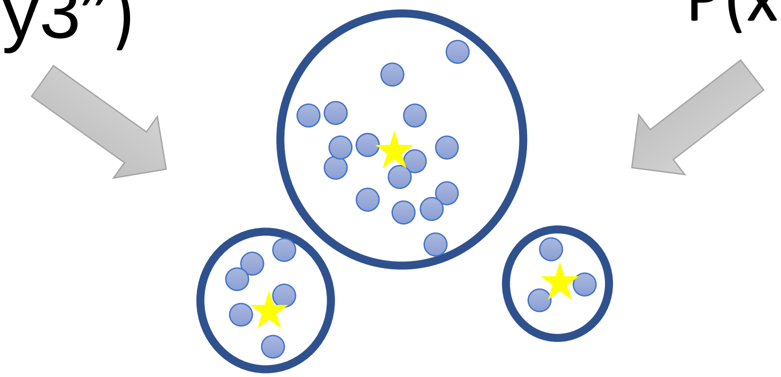
pointer

P(x|”d-uw+y3”)

Same Address

pointer

P(x|”t-d+uw3”)



終極型態: Subspace GMM [Povey, et al., ICASSP’10]

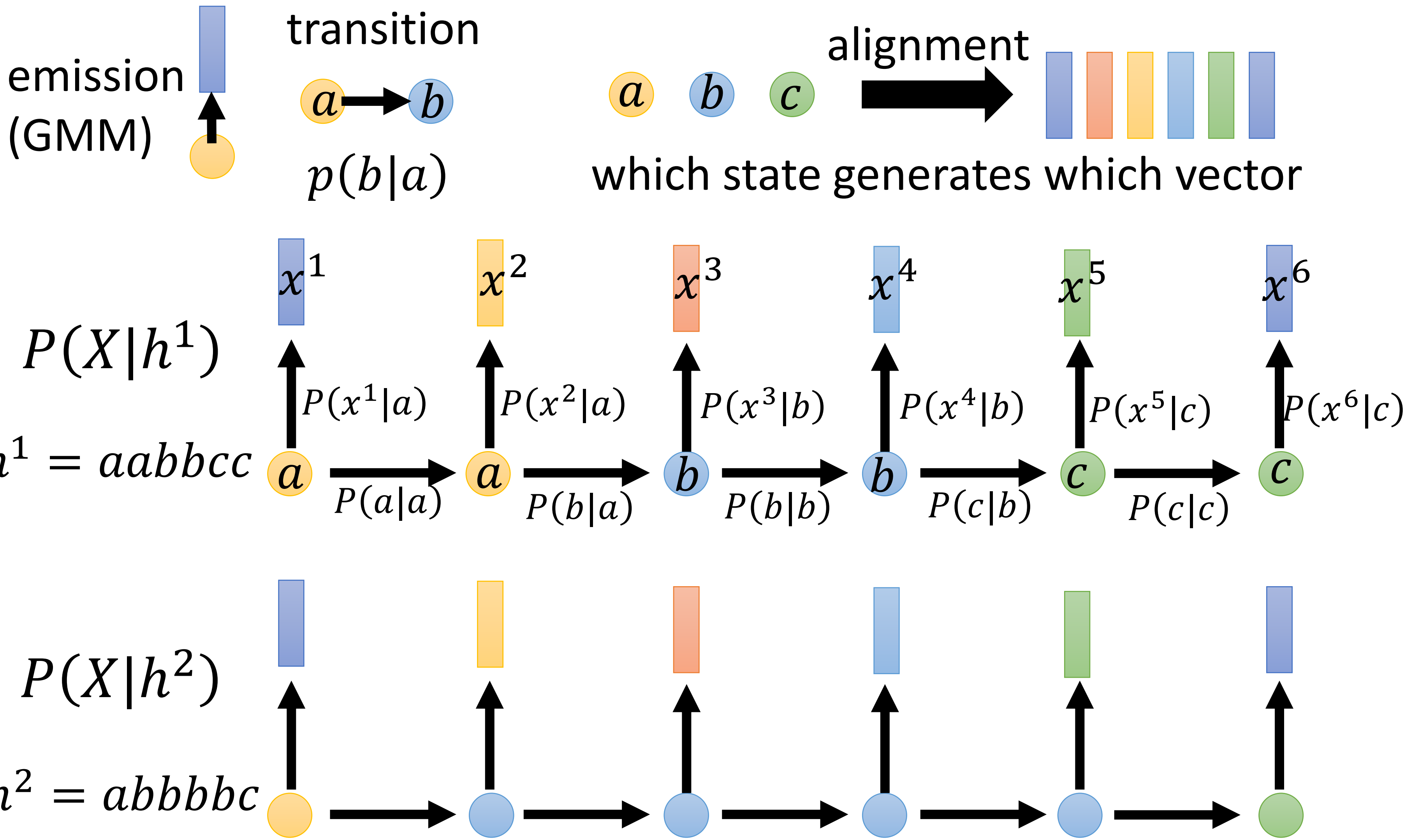
(Geoffrey Hinton also published deep learning for ASR in the same conference)

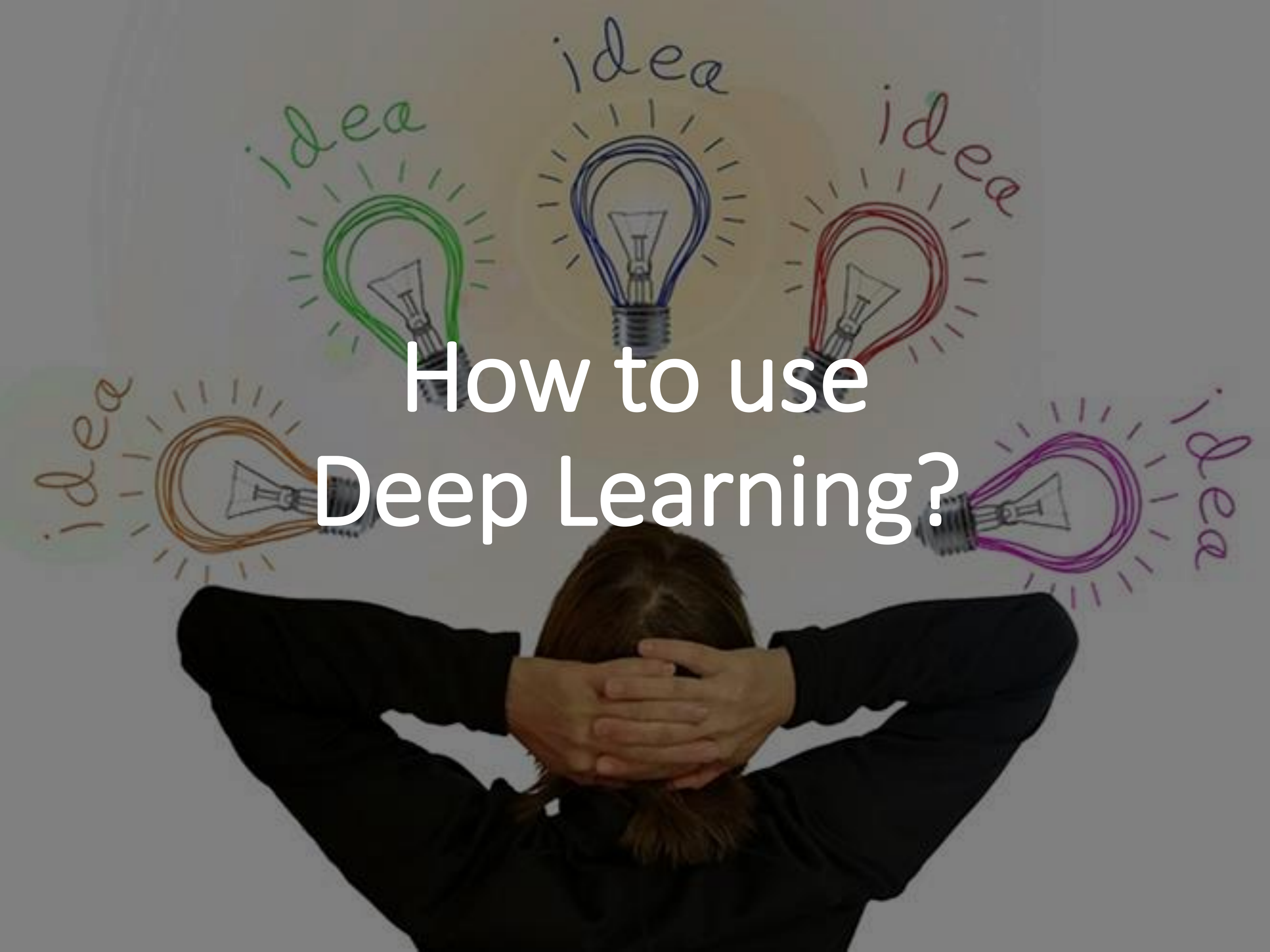
[Mohamed , et al., ICASSP’10]

$$P_{\boxed{\theta}}(X|S) = ? \sum_{\boxed{h \in align(S)}} P(X|h)$$

$h = abccbc$ ✗

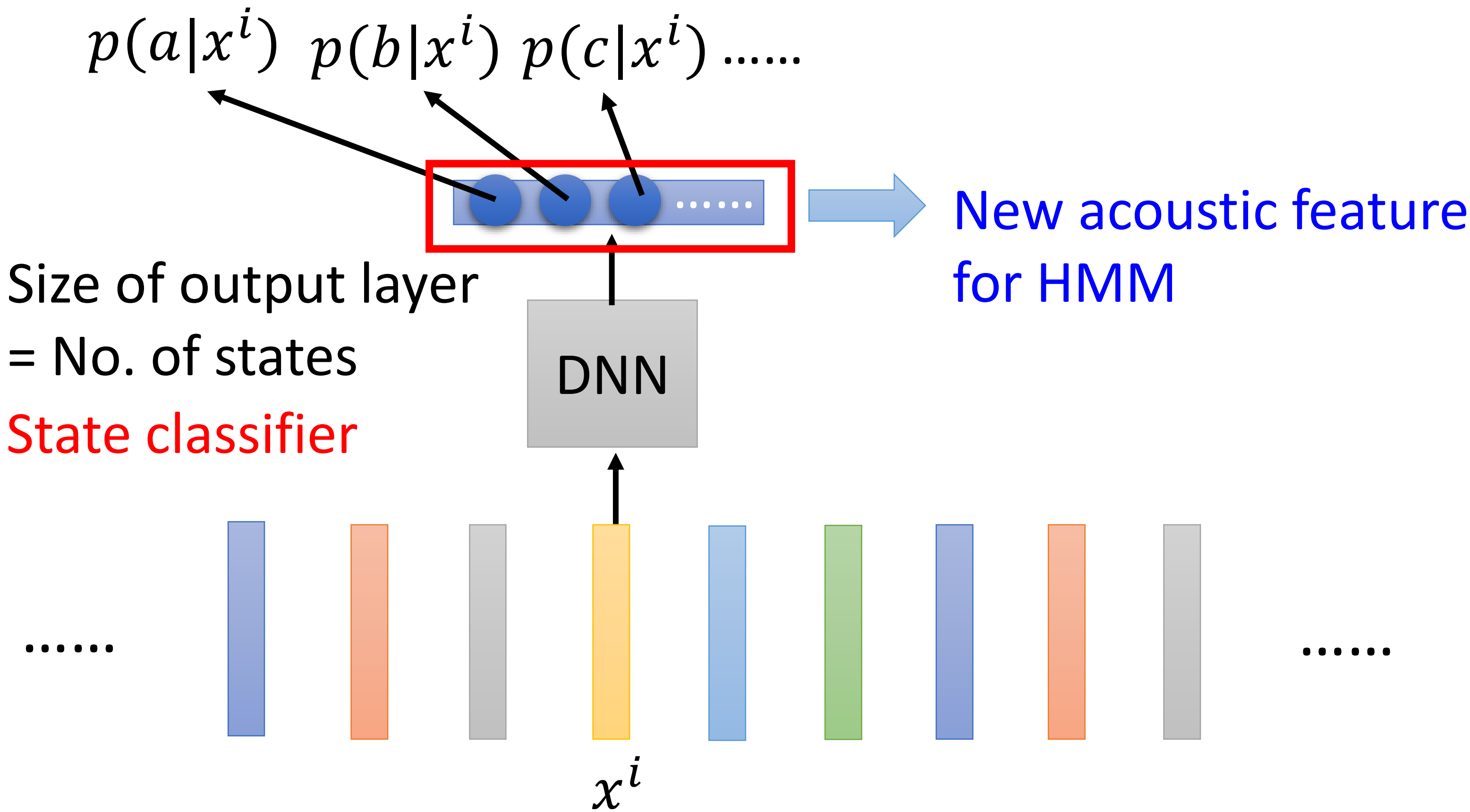
$h = abbbb$ ✗





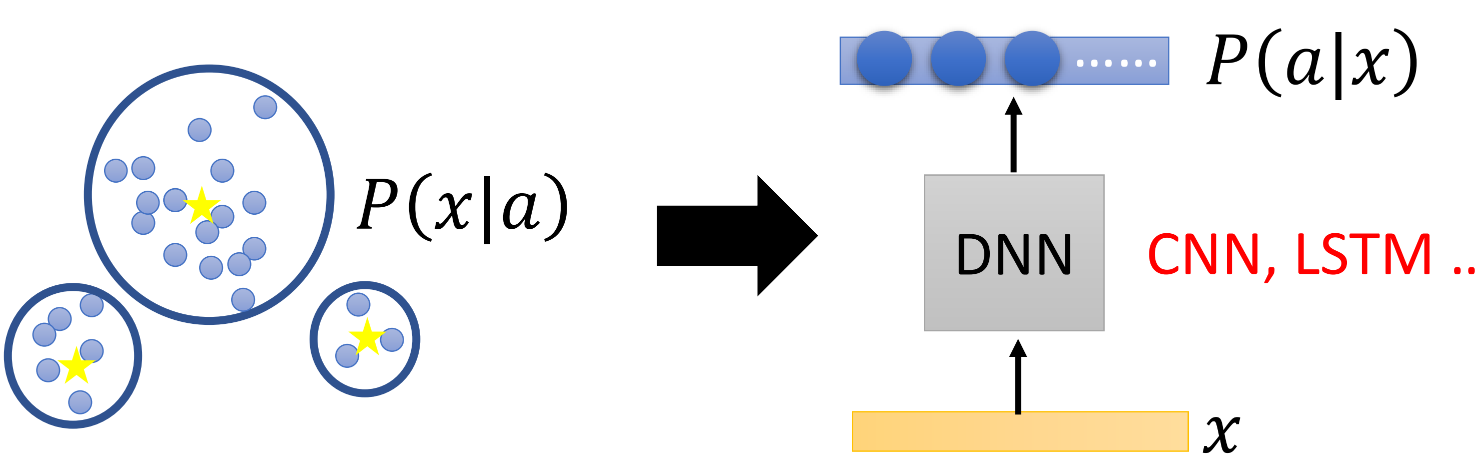
How to use Deep Learning?

Method 1: Tandem



Last hidden layer or bottleneck layer are also possible.

Method 2: DNN-HMM Hybrid

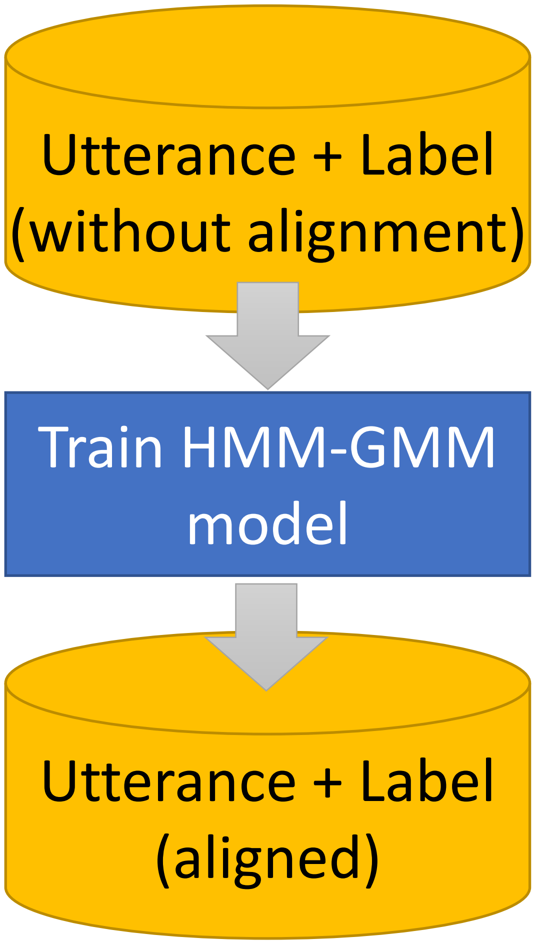


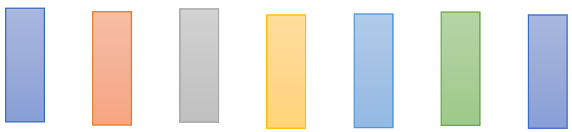
DNN output

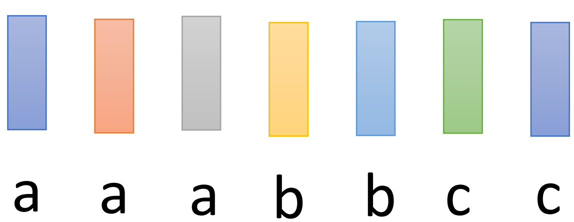
$$P(x|a) = \frac{P(x,a)}{P(a)} = \frac{P(a|x)\cancel{P(x)}}{P(a)}$$

Count from training data

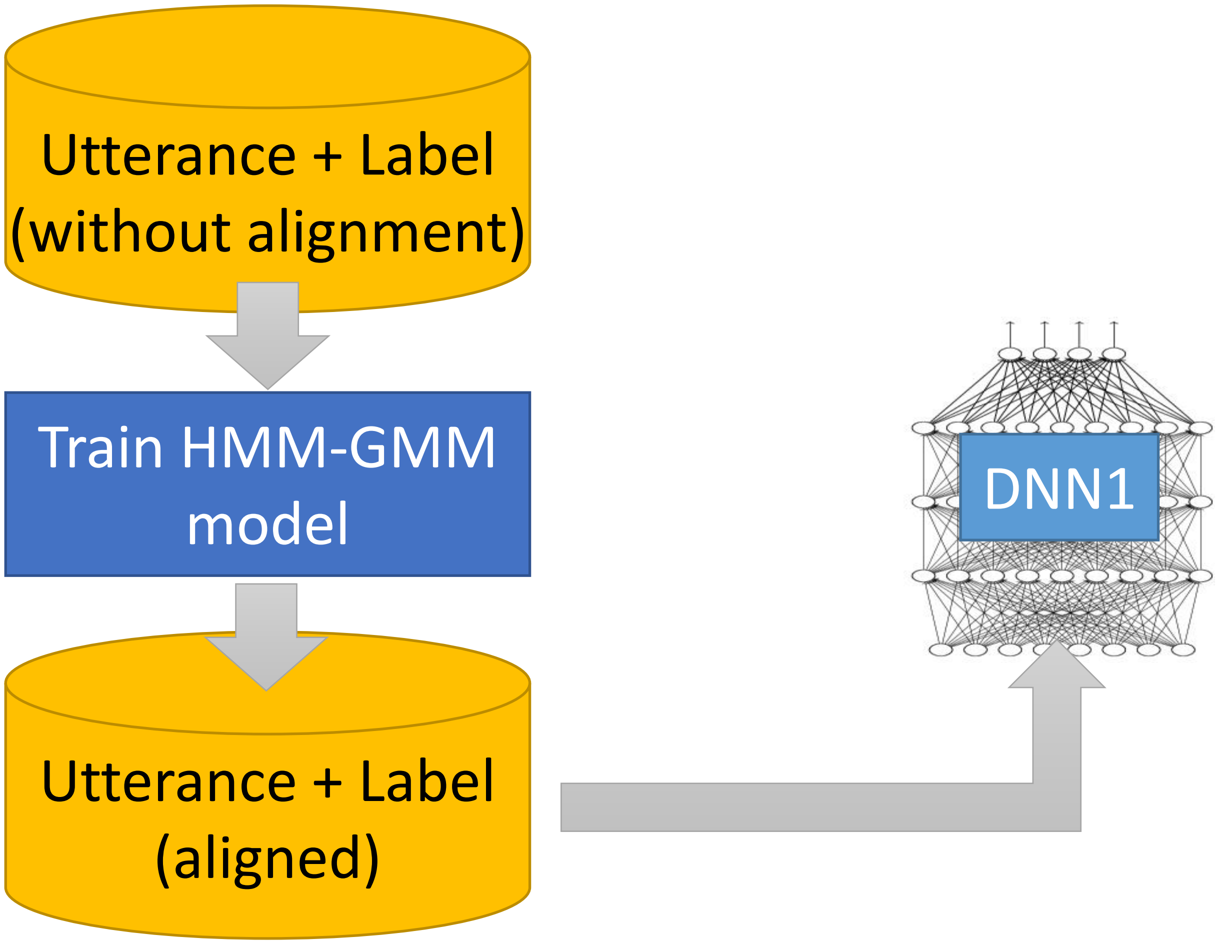
How to train a state classifier?



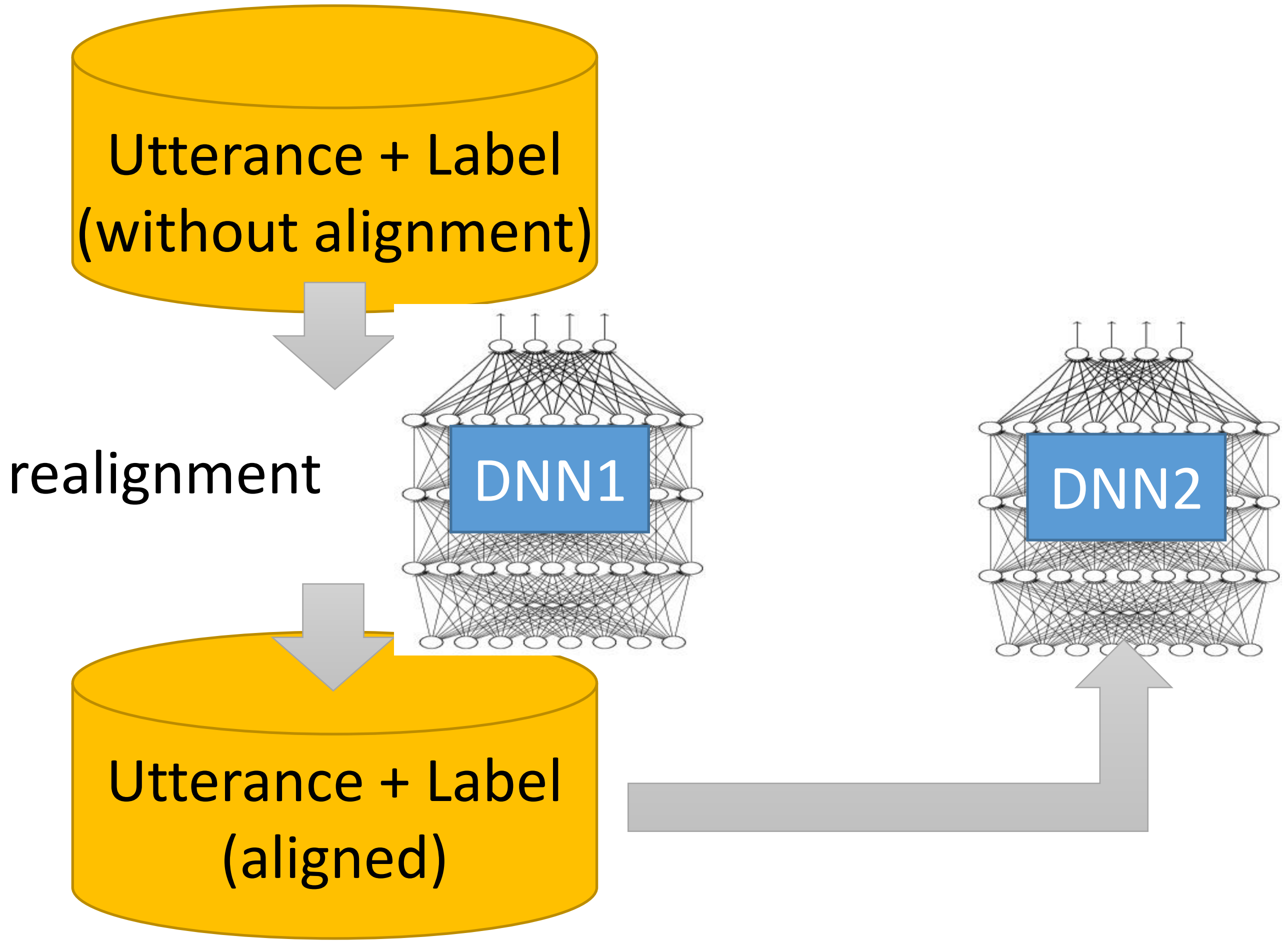




How to train a state classifier?



How to train a state classifier?



Human Parity!

- 微軟語音辨識技術突破重大里程碑：對話辨識能力達人類水準！(2016.10)
  - https://www.bnext.com.tw/article/41414/bn-2016-10-19-020437-216

Machine 5.9% v.s. Human 5.9%

[Yu, et al., INTERSPEECH'16]

- IBM vs Microsoft: 'Human parity' speech recognition record changes hands again (2017.03)
  - http://www.zdnet.com/article/ibm-vs-microsoft-human-parity-speech-recognition-record-changes-hands-again/

Machine 5.5% v.s. Human 5.1%

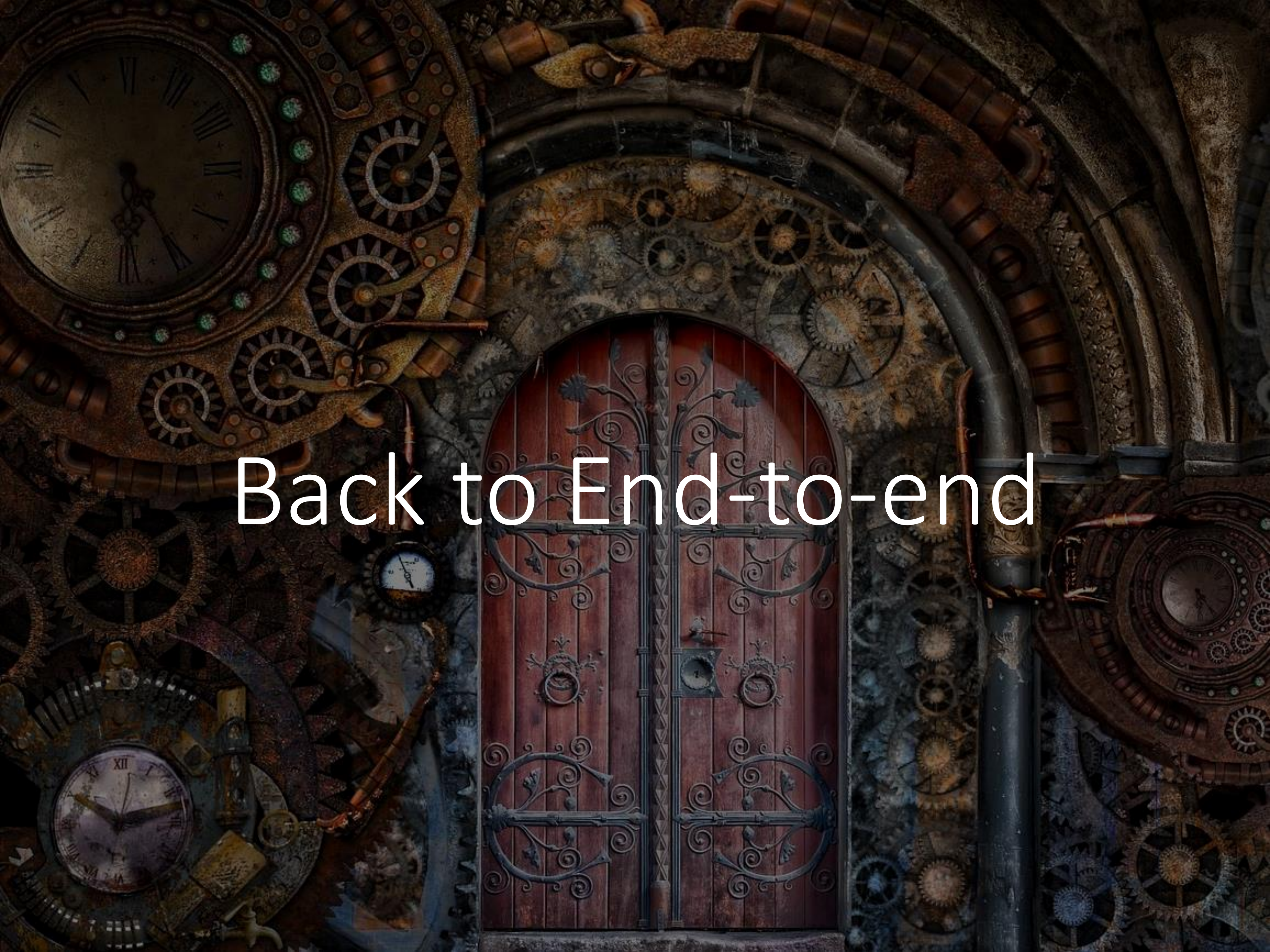
[Saon, et al., INTERSPEECH'17]

Very Deep

VGG Net (85M Parameters)	Residual-Net (38M Parameters)	LACE (65M Parameters)
14 weight layers	49 weight layers	22 weight layers
40x41 input	40x41 input	40x61 input
3 – conv 3x3, 96	3 – [conv 1x1, 64 conv 3x3, 64 conv 1x1, 256]	5 – conv 3x3, 128
Max pool	4 – [conv 1x1, 128 conv 3x3, 128 conv 1x1, 512]	5 – conv 3x3, 256
4 – conv 3x3, 192	6 – [conv 1x1, 256 conv 3x3, 256 conv 1x1, 1024]	5 – conv 3x3, 512
Max pool	3 – [conv 1x1, 512 conv 3x3, 512 conv 1x1, 2048]	5 – conv 3x3, 1024
4 – conv 3x3, 384	Average pool	1 – conv 3x4, 1
Max pool	Softmax (9000)	Softmax (9000)
2 – FC – 4096		
Softmax (9000)		

[Yu, et al., INTERSPEECH'16]

Back to End-to-end

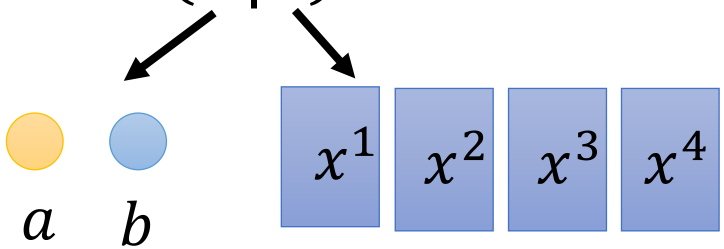


LAS

Decoding: $Y^* = arg\max_{Y} logP(Y|X)$ Beam Search

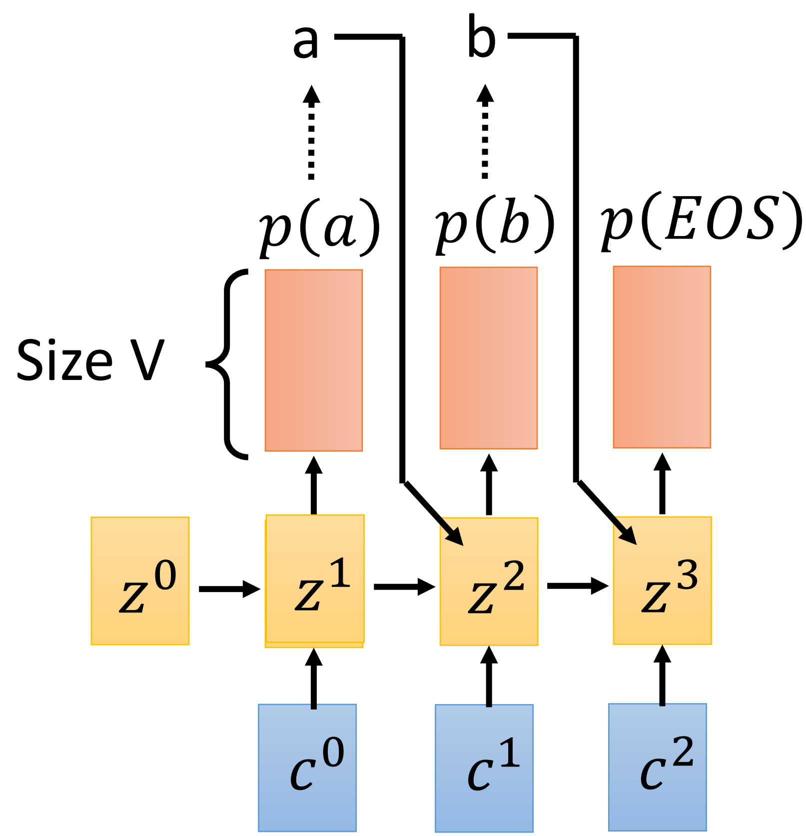
Training: $\theta^* = arg\max_{\theta} logP_\theta(\hat{Y}|X)$

$P(Y|X)$ =?



- LAS directly computes $P(Y|X)$

$$P(Y|X) = p(a|X)p(b|a,X)\ldots$$

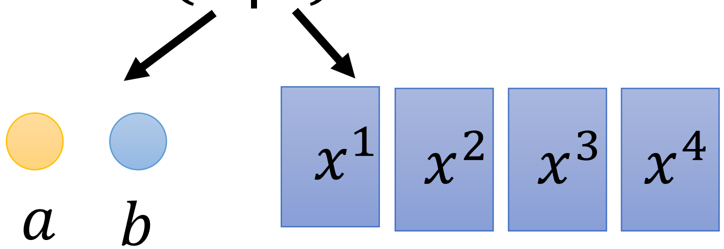


CTC, RNN-T

Decoding: $Y^* = arg\max_{Y} logP(Y|X)$ Beam Search

Training: $\theta^* = arg\max_{\theta} logP_{\theta}(\hat{Y}|X)$

$P(Y|X) = ?$



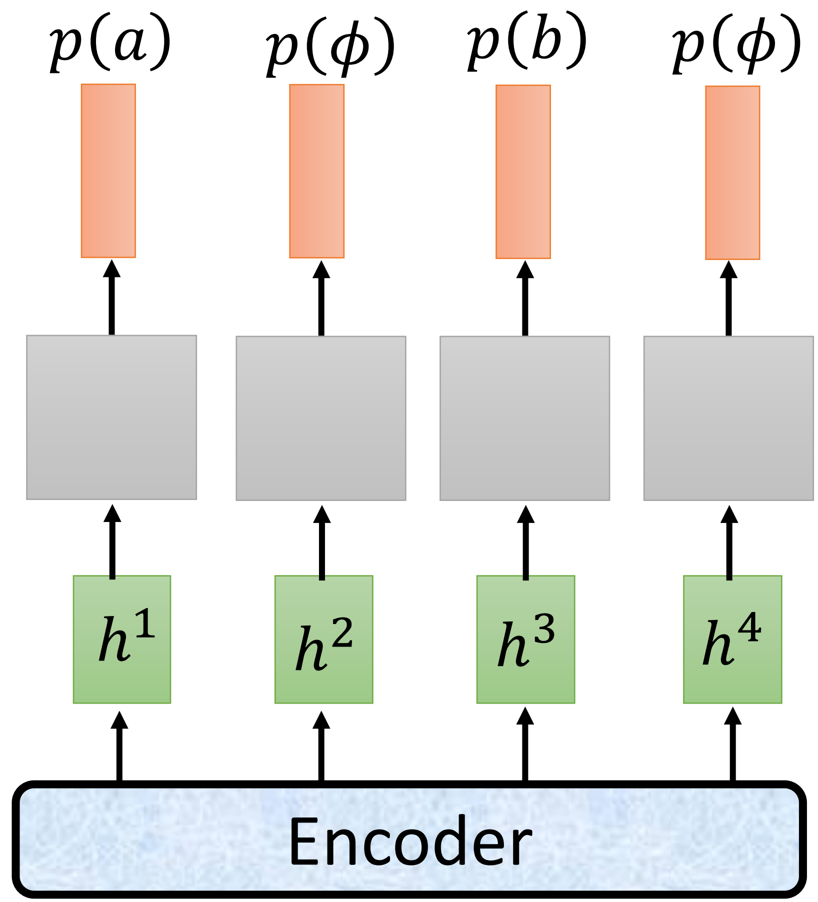
- LAS directly computes $P(Y|X)$

  $P(Y|X) = p(a|X)p(b|a,X)\ldots$

- CTC and RNN-T need ***alignment***

  $$\mathrm{P}(Y|X) = \sum_{h \in align(Y)} P(h|X)$$

$P(h|X)$ $h = a\ \phi\ b\ \phi \rightarrow a\ b$



HMM, CTC, RNN-T

HMM

$$P_\theta(X|S) = \boxed{\sum_{\underline{h \in align(S)}} P(X|h)}$$

CTC, RNN-T

$$P_\theta(Y|X) = \boxed{\sum_{\underline{h \in align(Y)}} P(h|X)}$$

1. Enumerate all the possible alignments

2. How to sum over all the alignments

3. Training: $\boxed{\theta^* = arg \max_\theta log P_\theta(\hat{Y}|X)}$ $\quad \dfrac{\partial P_\theta(\hat{Y}|X)}{\partial \theta} = ?$

4. Testing (Inference, decoding):

$$Y^* = arg \max_Y log P(Y|X)$$

HMM, CTC, RNN-T

HMM

$$P(X|S) = \sum_{h \in align(S)} P(X|h)$$

CTC, RNN-T

$$P(\mathrm{Y}|X) = \sum_{h \in align(Y)} P(h|X)$$

$\boxed{\text{1. Enumerate all the possible alignments}}$

2. How to sum over all the alignments

3. Training:

$$\theta^* = arg\max_{\theta} log\mathrm{P}_{\theta}(\hat{Y}|X) \qquad \frac{\partial \mathrm{P}_{\theta}(\hat{Y}|X)}{\partial \theta} = ?$$

4. Testing (Inference, decoding):

$$\mathrm{Y}^* = arg\max_{\mathrm{Y}} logP(\mathrm{Y}|X)$$

All the alignments

LAS

你們在忙什麼 ☺

	Y		$h \in align(Y)$
HMM	c a t	duplicate to length T	c c a a a t c a a a a t …
CTC	c a t	{duplicate, add ϕ} to length T	c ϕ a a t t ϕ c a ϕ t ϕ …
RNN-T	c a t	add ϕ x T	c ϕ ϕ ϕ a ϕ ϕ t ϕ c ϕ ϕ a ϕ ϕ t ϕ ϕ ……

HMM c a t → c c a a a t c a a a a t …

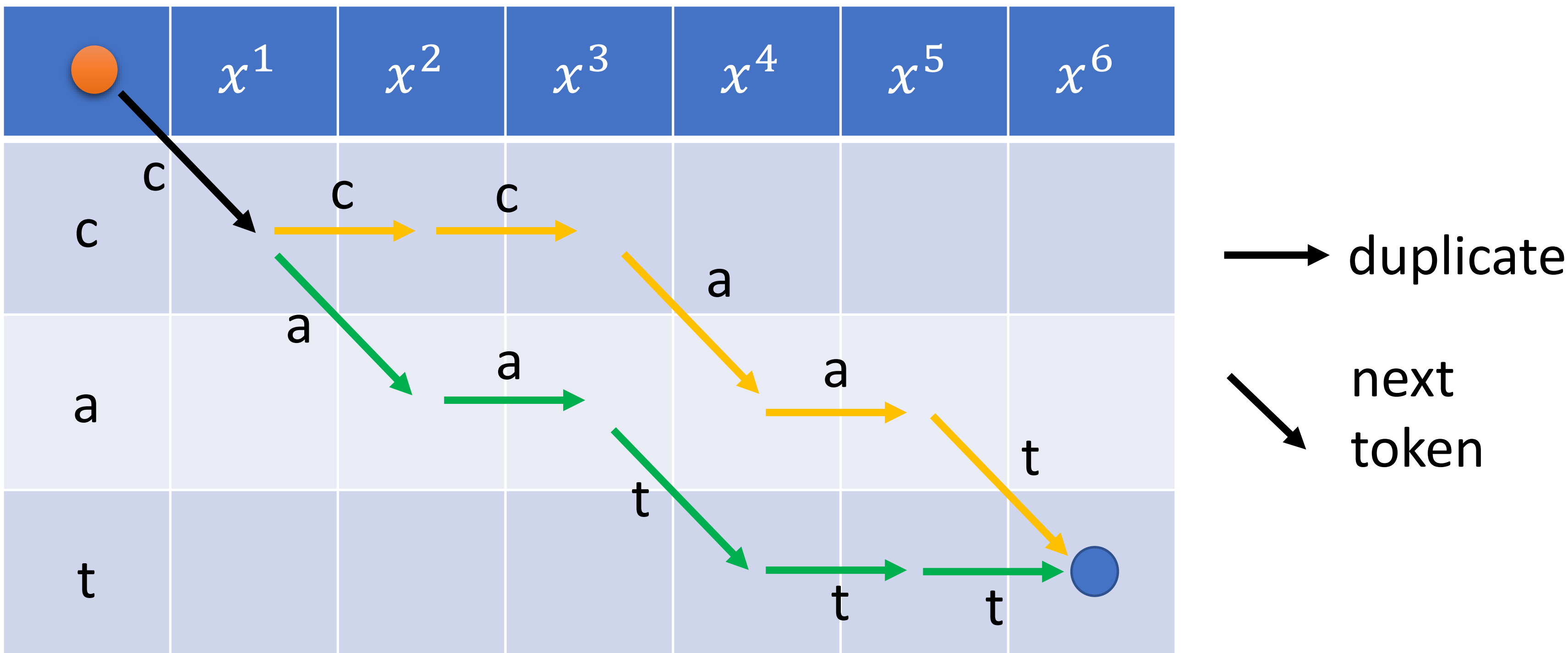
duplicate to length T

For n = 1 to N

output the n-th token t_n times

constraint: $t_1 + t_2 + \cdots t_N = T, t_n > 0$

Trellis Graph



HMM c a t → c c a a a t c a a a a t ...

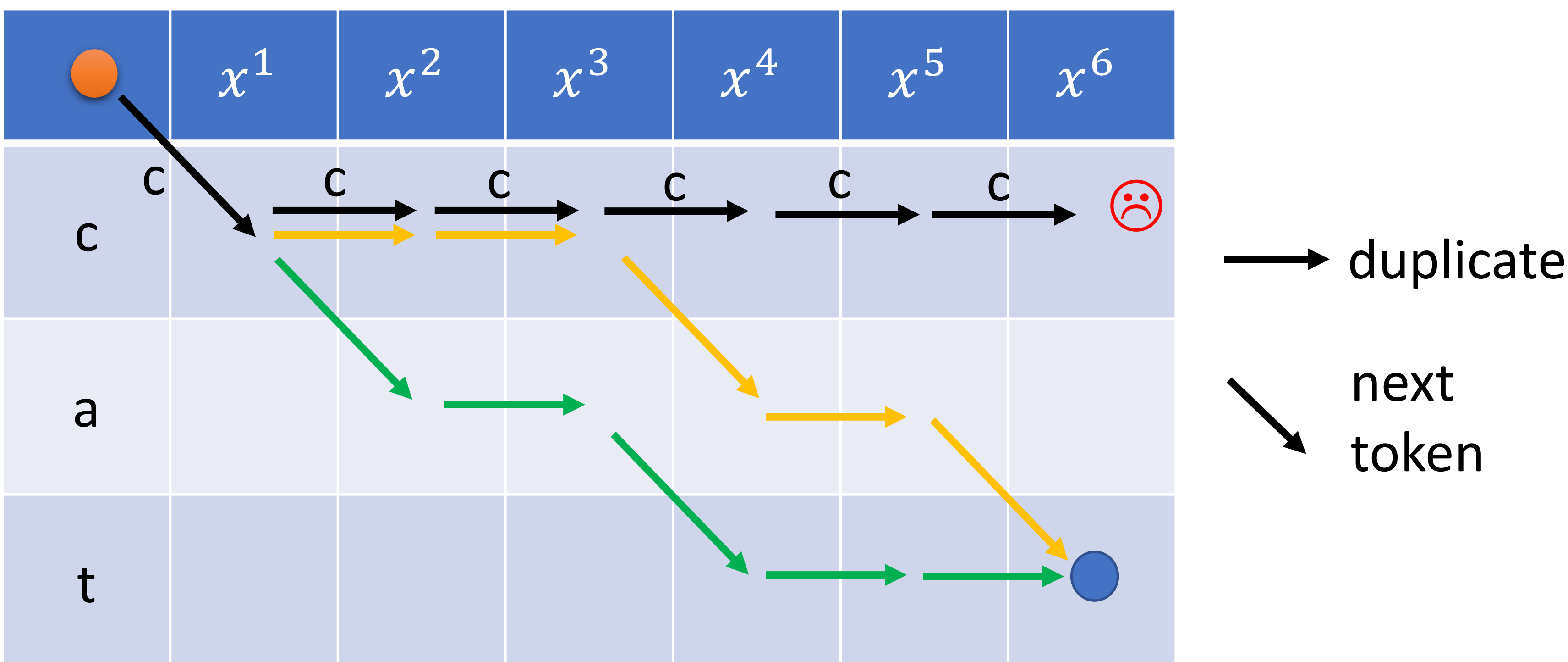
duplicate to length T

For n = 1 to N

output the n-th token t_n times

constraint: $t_1 + t_2 + \cdots t_N = T, t_n > 0$

Trellis Graph



CTC c a t $\xrightarrow{\qquad\qquad}$ c ϕ a a t t $\quad$ ϕ c a ϕ t ϕ …

$\left\{\begin{array}{l}\text{duplicate}\\ \text{add } \phi\end{array}\right.$ to length T

output "ϕ" c_0 times

For n = 1 to N

output the n-th token t_n times

output "ϕ" c_n times

constraint: $t_1 + t_2 + \cdots t_N +$

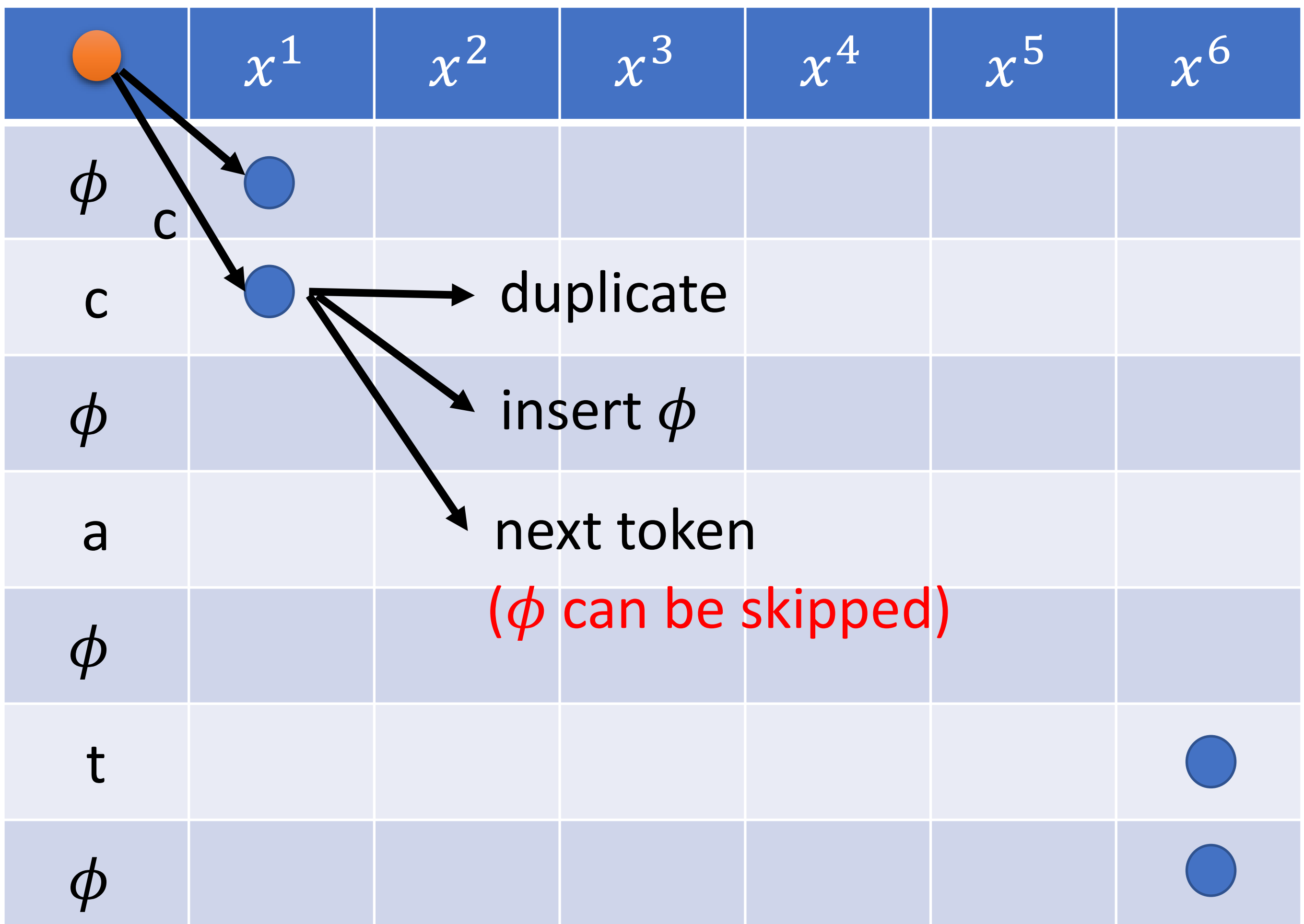
$c_0 + c_1 + \cdots c_N = T$

$t_n > 0 \quad c_n \geq 0$

CTC

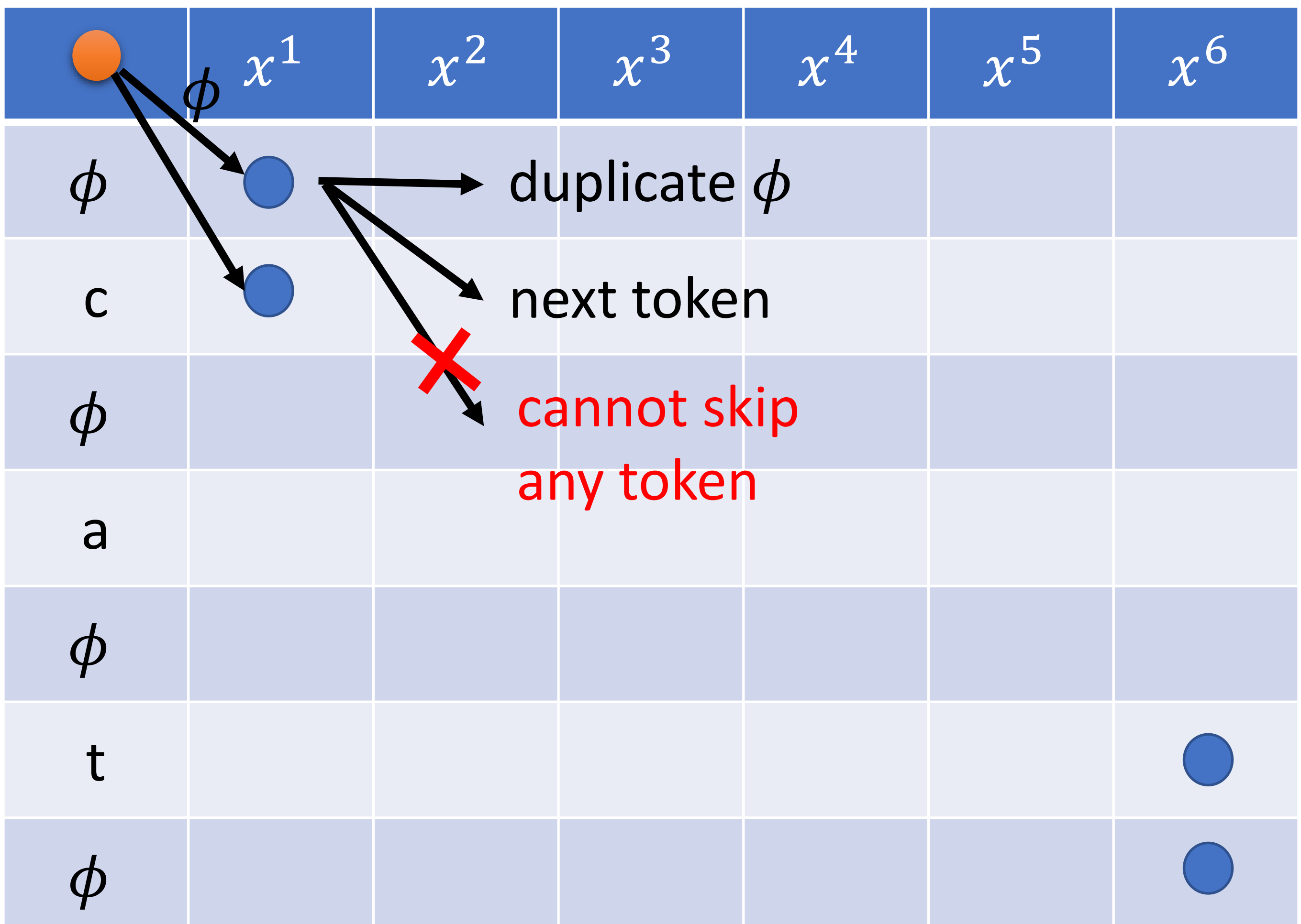
c a t $\longrightarrow$ c ϕ a a t t ϕ c a ϕ t ϕ …

$\left\{\begin{array}{l}\text{duplicate}\\ \text{add } \phi\end{array}\right.$ to length T



CTC c a t $\longrightarrow$ c ϕ a a t t ϕ c a ϕ t ϕ …

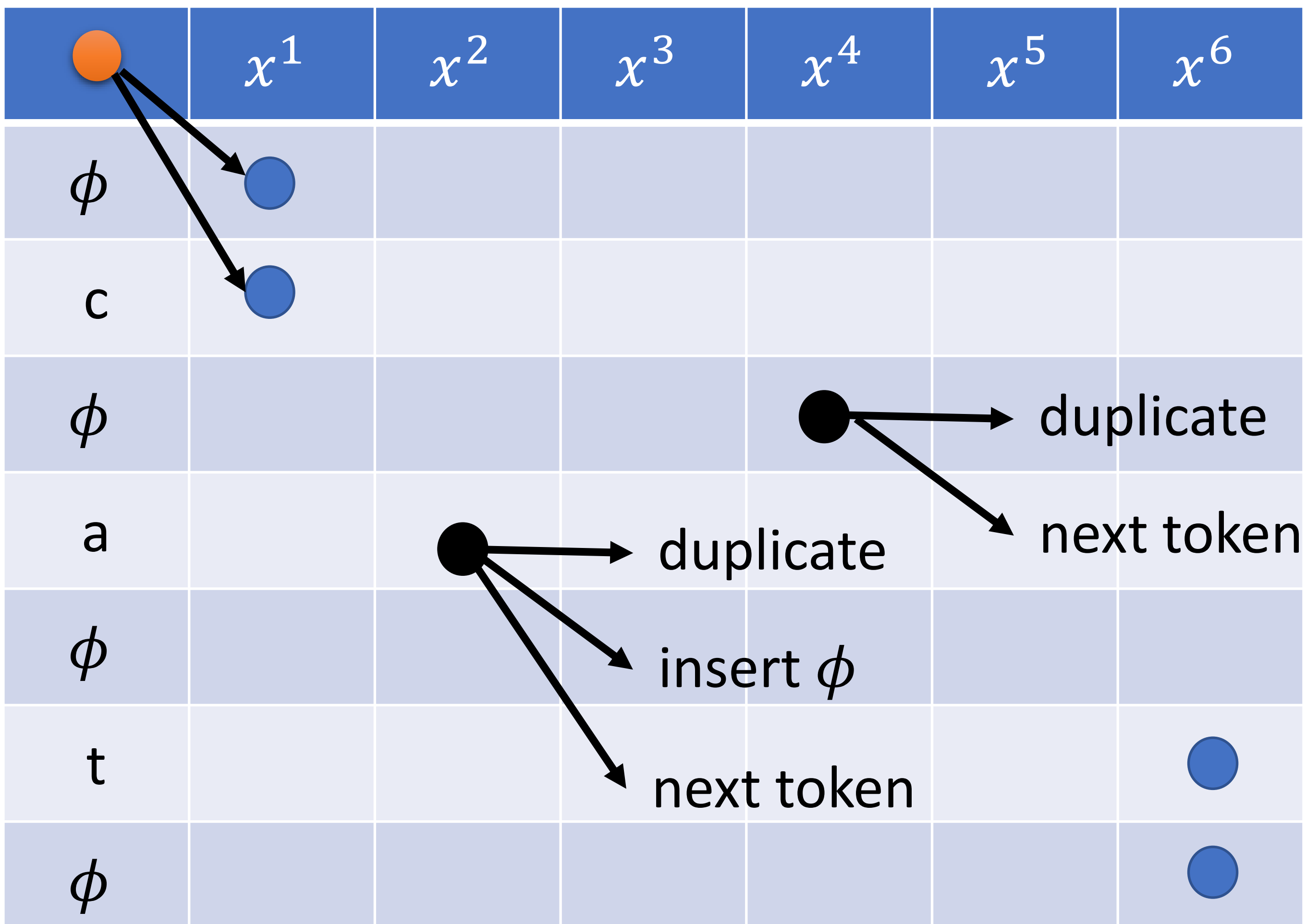
$\left\{\begin{array}{l}\text{duplicate}\\ \text{add } \phi\end{array}\right.$ to length T



CTC

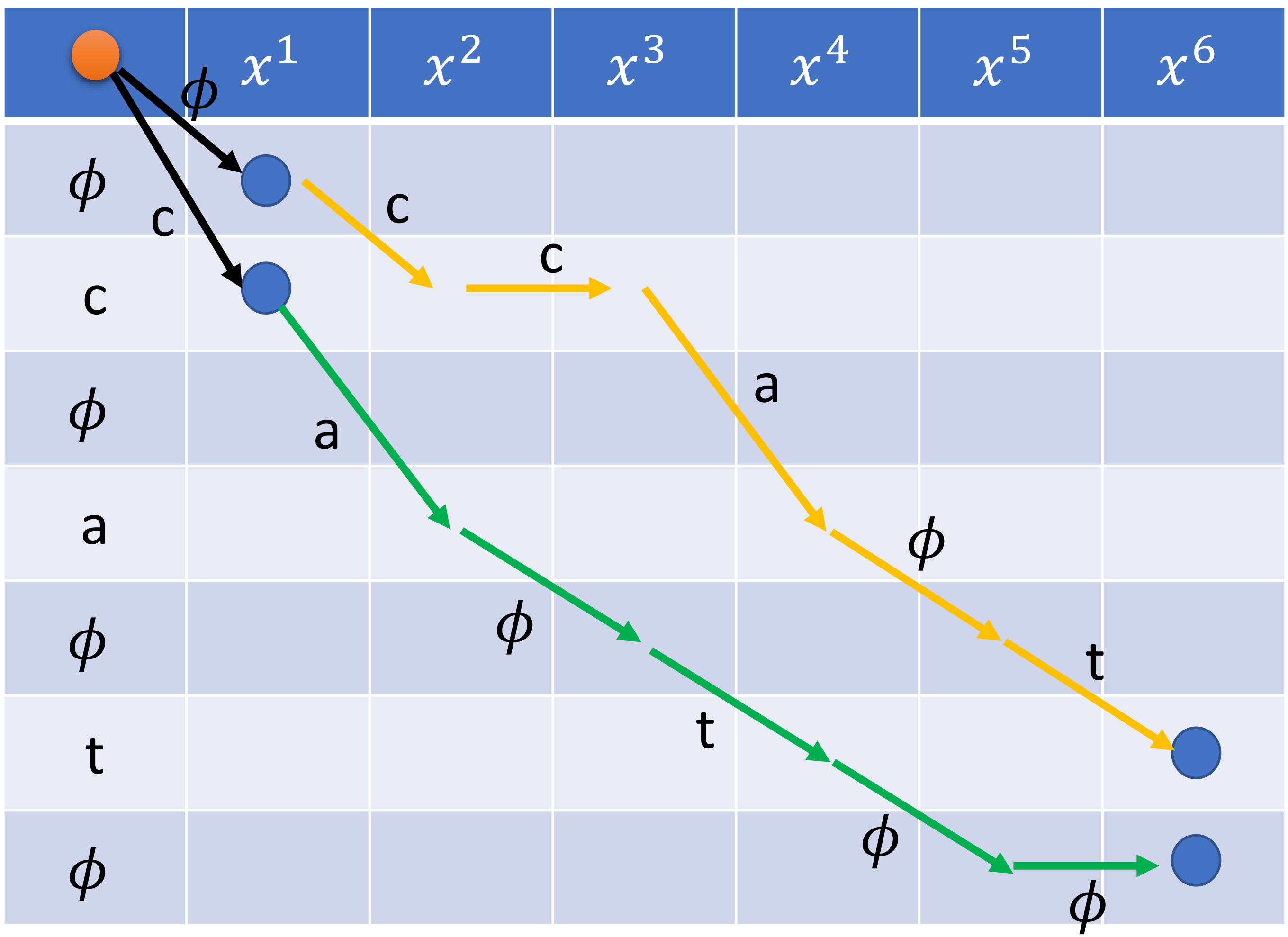
c a t $\longrightarrow$ c ϕ a a t t ϕ c a ϕ t ϕ …

$\left\{\begin{array}{l}\text{duplicate}\\ \text{add } \phi\end{array}\right.$ to length T



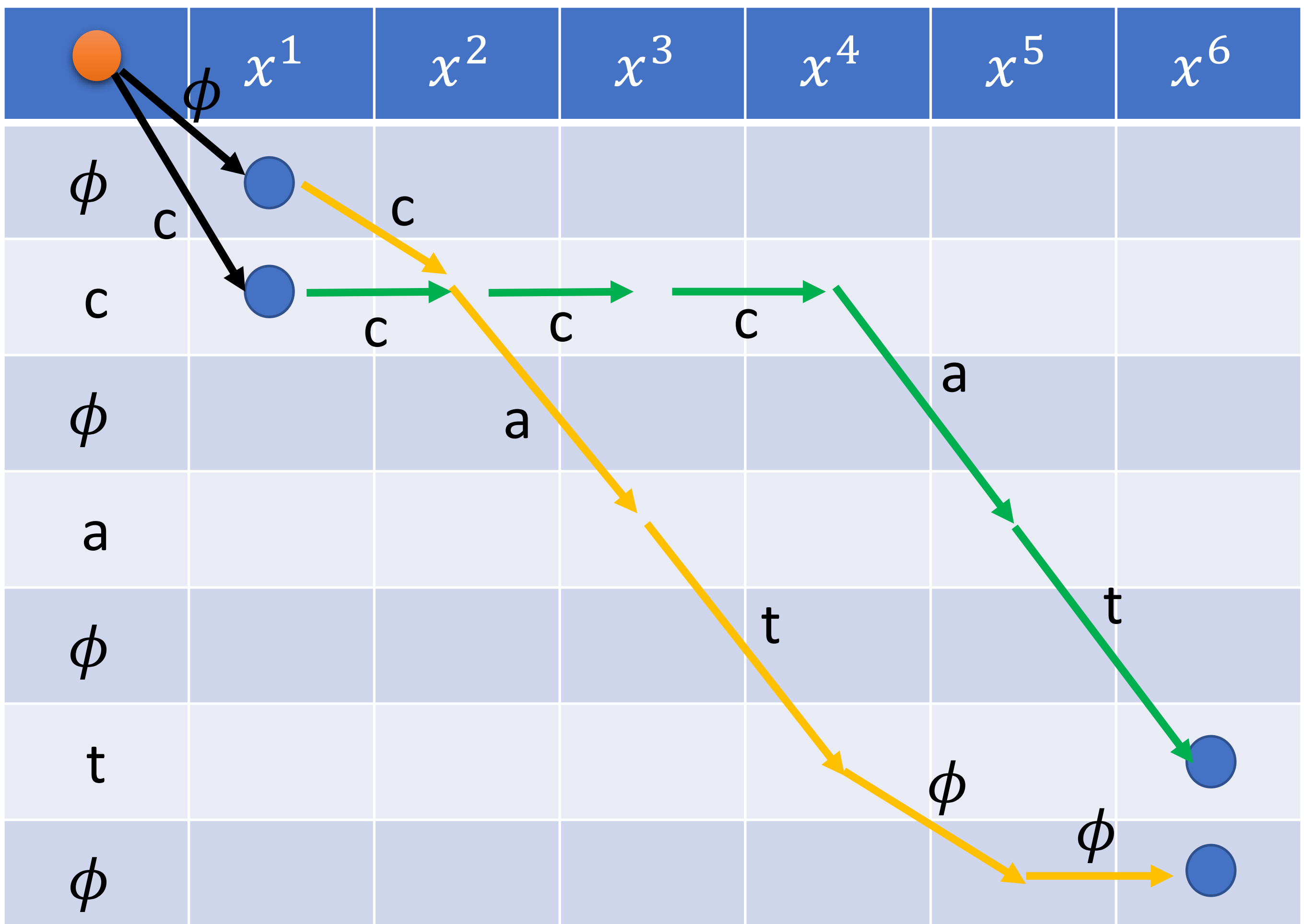
CTC   c a t $\longrightarrow$ c ϕ a a t t ϕ c a ϕ t ϕ …

$\left\{\begin{array}{l}\text{duplicate}\\ \text{add } \phi\end{array}\right.$ to length T



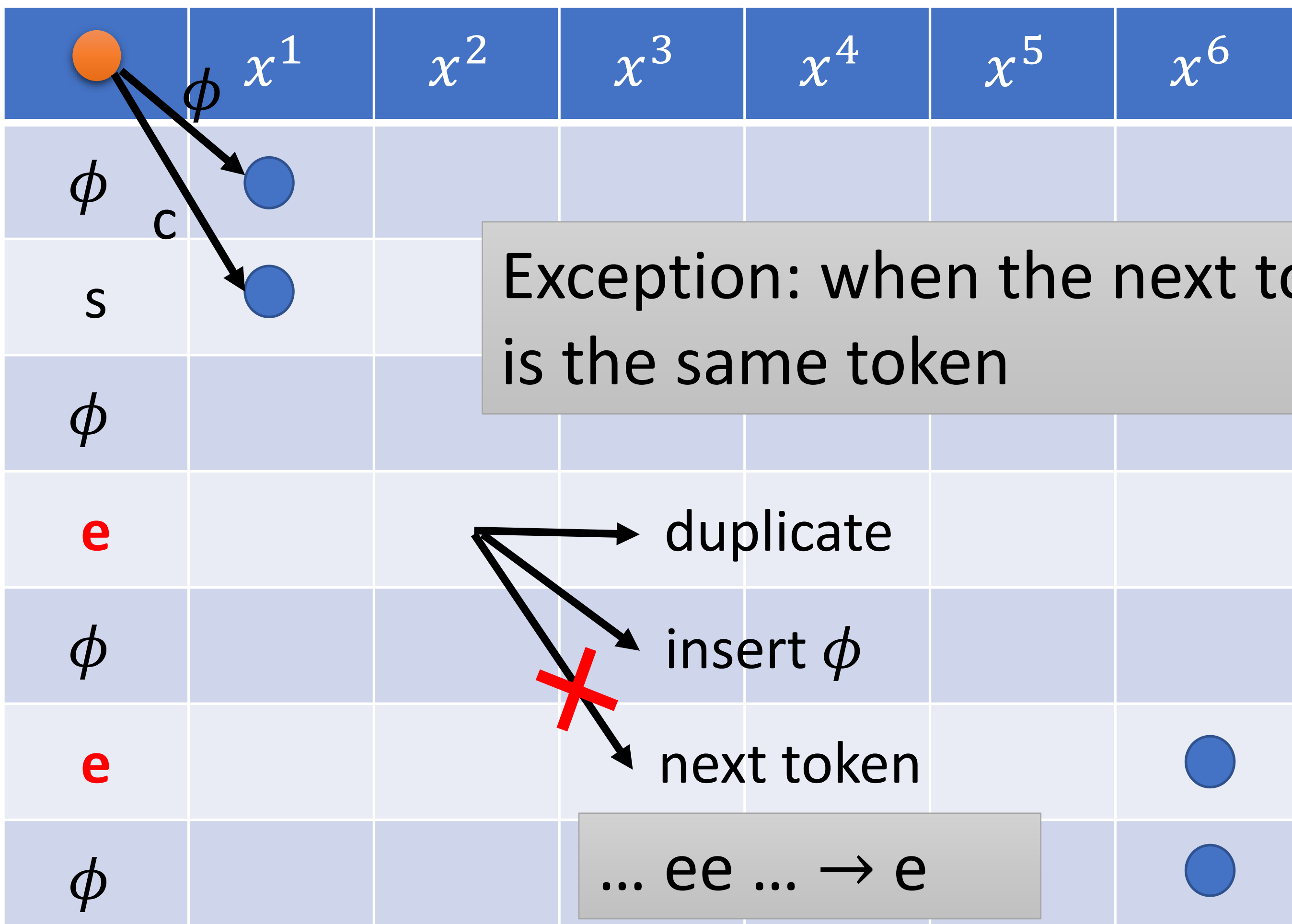
CTC c a t $\longrightarrow$ c ϕ a a t t ϕ c a ϕ t ϕ …

$\left\{ \begin{array}{l} \text{duplicate} \\ \text{add } \phi \end{array} \right.$ to length T



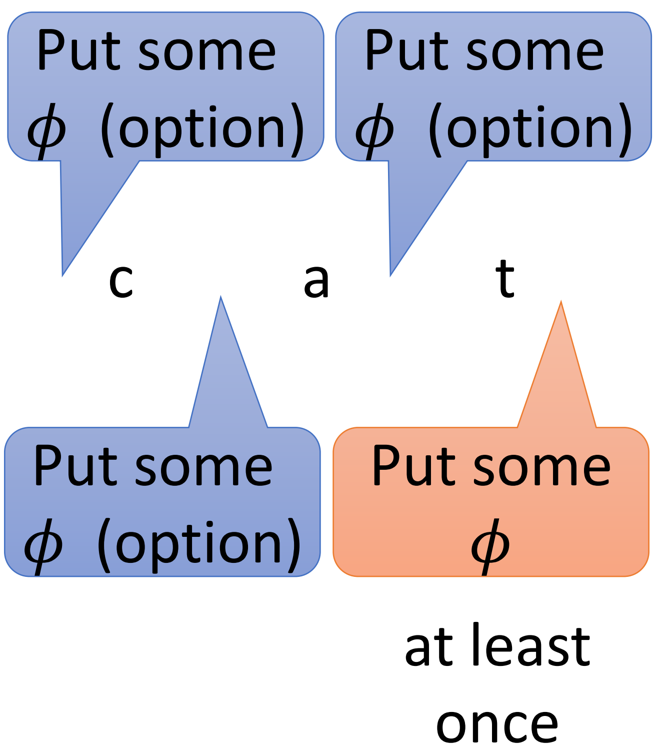
CTC c a t $\xrightarrow{}$ c ϕ a a t t ϕ c a ϕ t ϕ …

$\left\{\begin{array}{l}\text{duplicate} \\ \text{add } \phi\end{array}\right.$ to length T



Exception: when the next token is the same token

RNN-T c a t $\xrightarrow{\text{add } \phi \text{ x } T}$ c ϕ ϕ ϕ a ϕ ϕ t ϕ ……
c ϕ ϕ a ϕ ϕ t ϕ ϕ



output "ϕ" c_0 times

For n = 1 to N

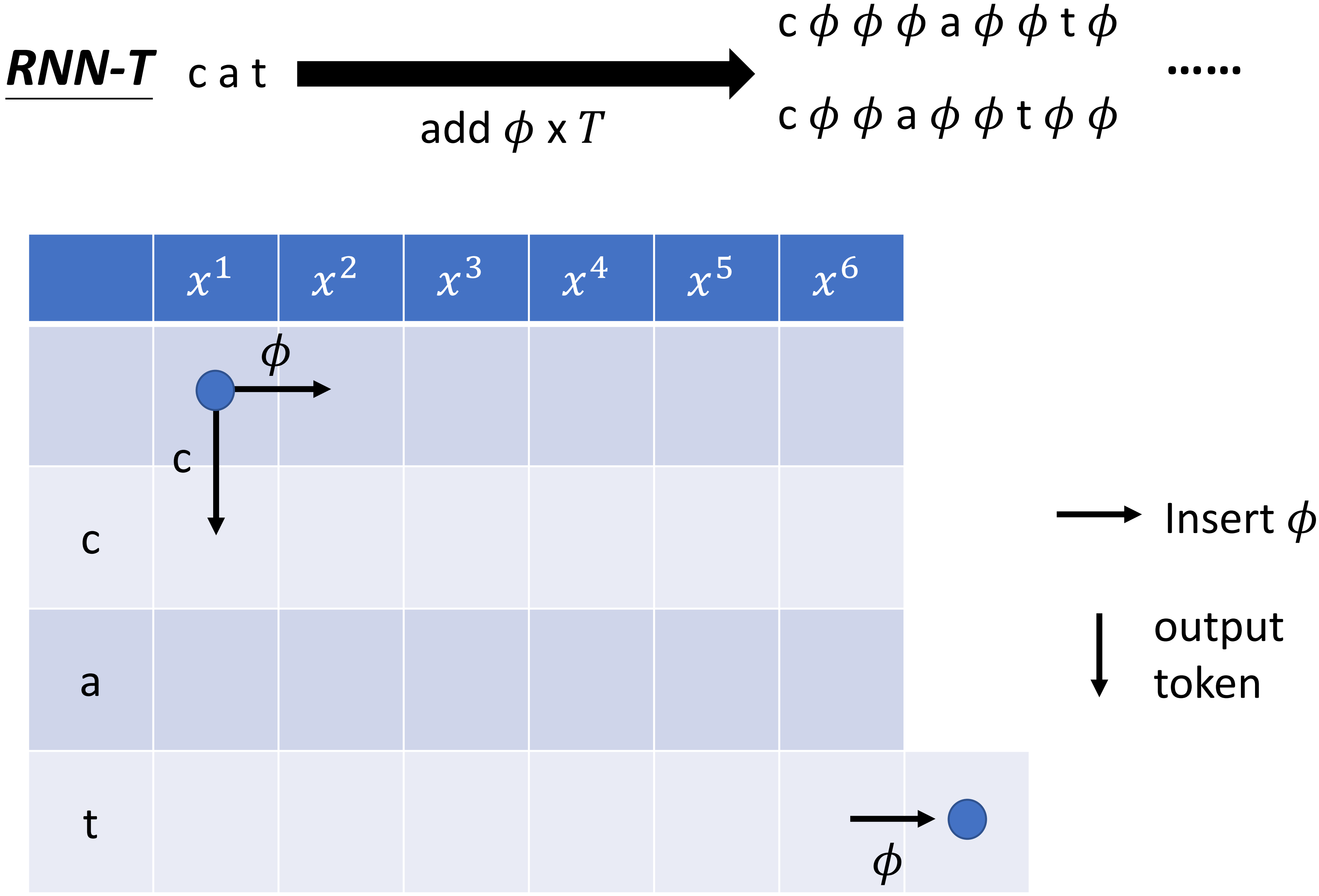
output the n-th token 1 times

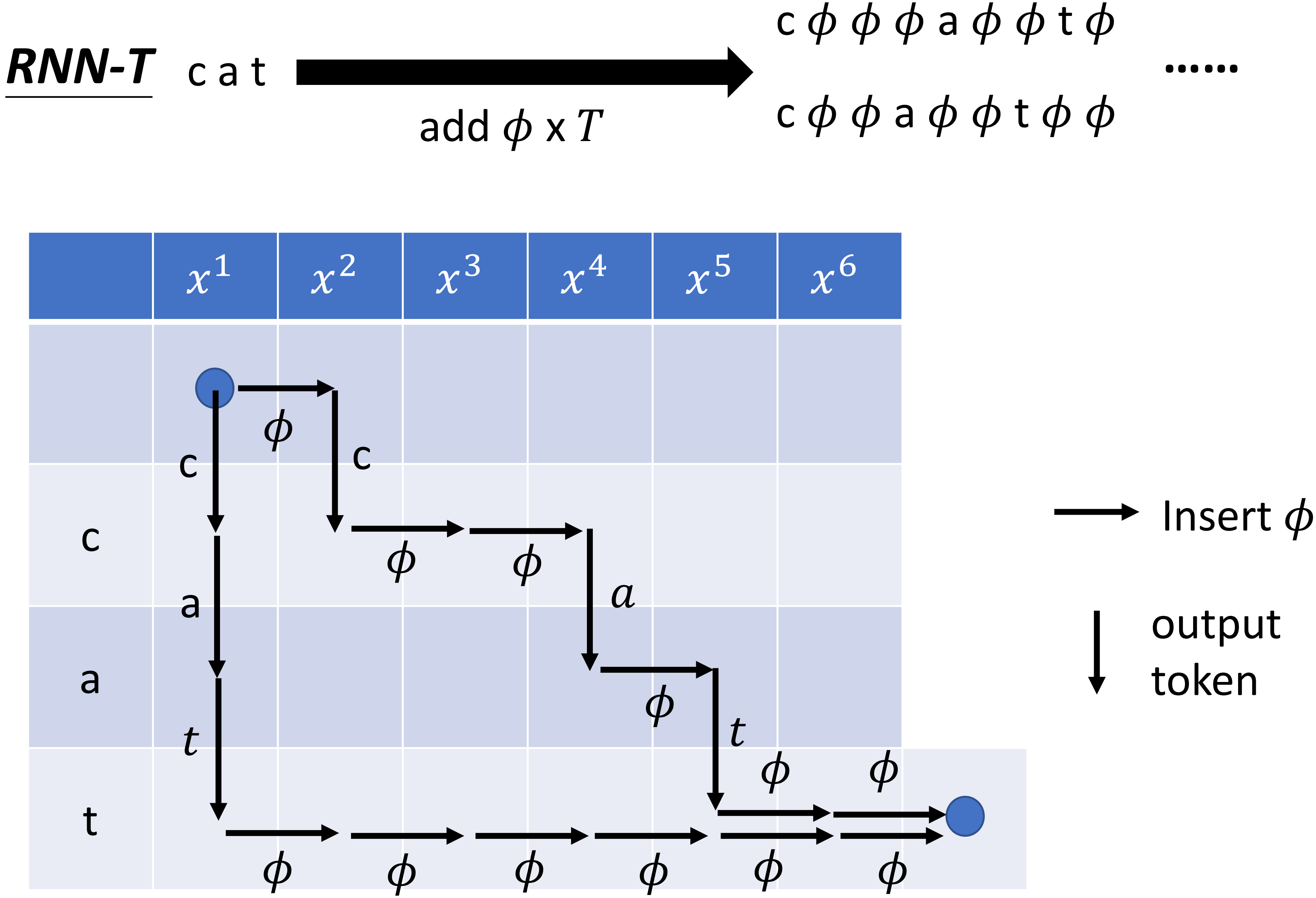
output "ϕ" c_n times

constraint: $c_0 + c_1 + \cdots c_N = T$

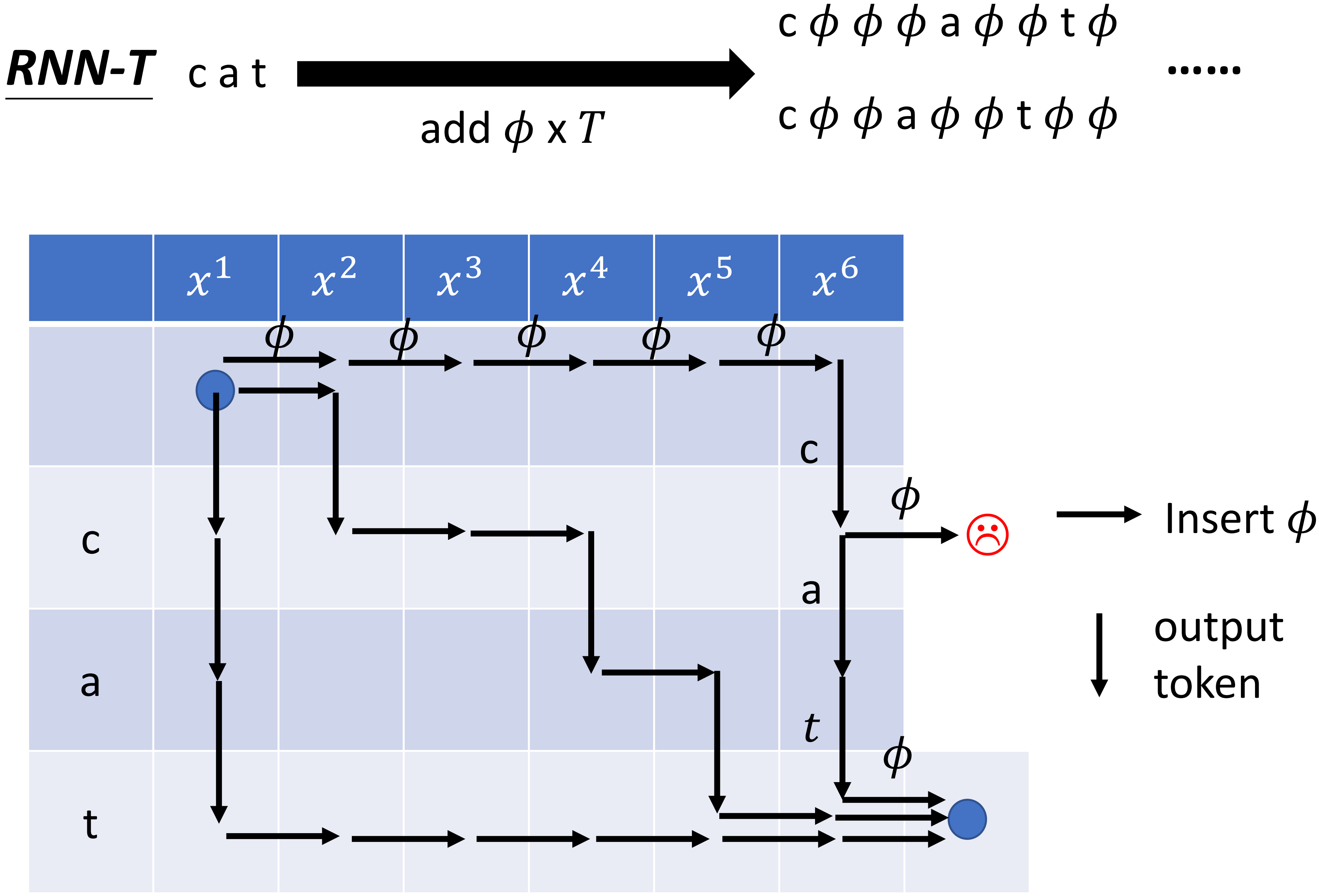
$c_N > 0$

$c_n \geq 0$ for n = 1 to $N - 1$

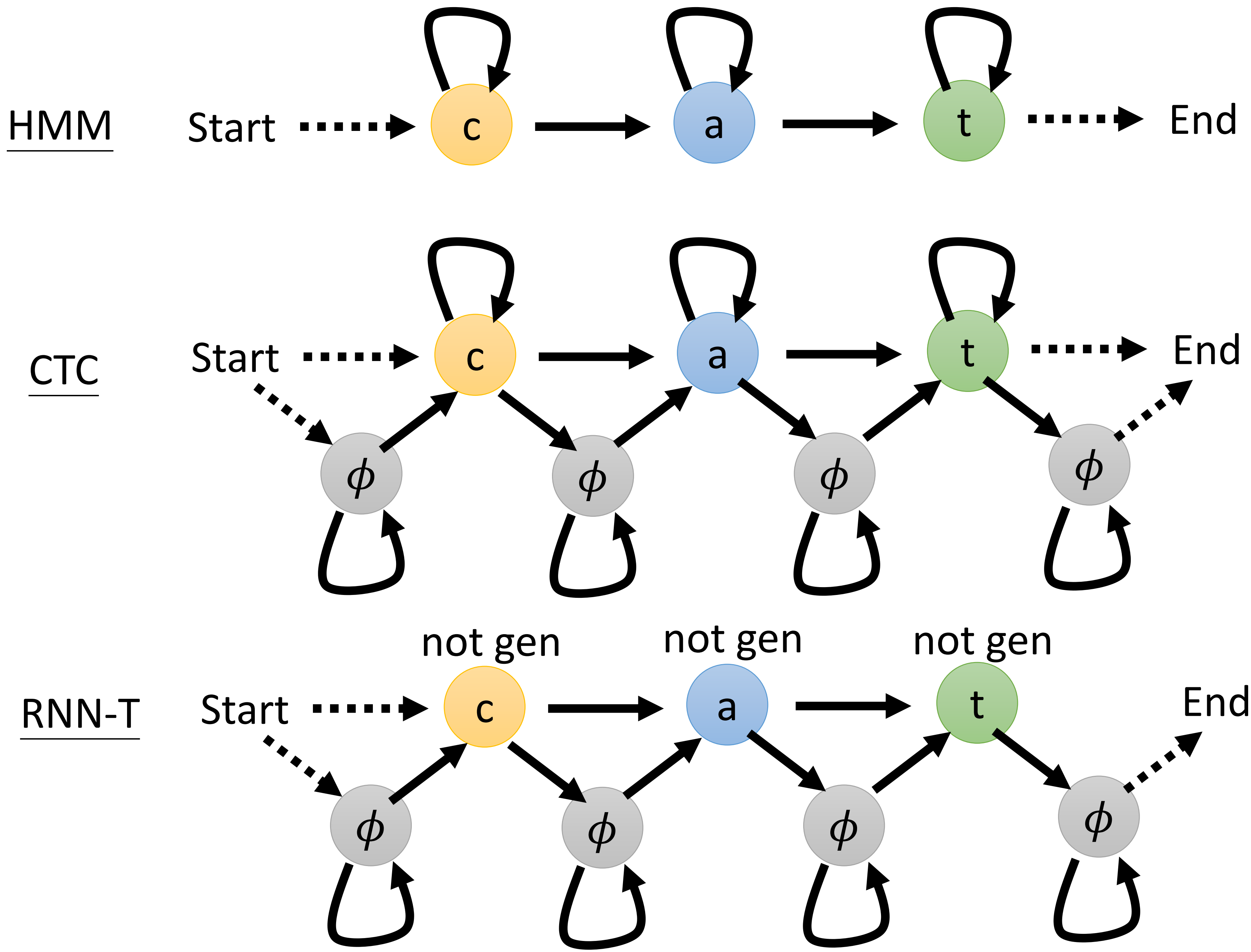

RNN-T
c a t
add ϕ x T
c ϕ ϕ ϕ a ϕ ϕ t ϕ
c ϕ ϕ a ϕ ϕ t ϕ ϕ
……
x^1 x^2 x^3 x^4 x^5 x^6
c
a
t
ϕ
c
ϕ
Insert ϕ
output token



RNN-T
c a t
add ϕ x T
c ϕ ϕ ϕ a ϕ ϕ t ϕ
c ϕ ϕ a ϕ ϕ t ϕ ϕ
……
x^1 x^2 x^3 x^4 x^5 x^6
c
a
t
Insert ϕ
output token



RNN-T
c a t
add ϕ x T
c ϕ ϕ ϕ a ϕ ϕ t ϕ
c ϕ ϕ a ϕ ϕ t ϕ ϕ
......
x^1 x^2 x^3 x^4 x^5 x^6
c
a
t
ϕ
Insert ϕ
output token


HMM
Start
c
a
t
End
CTC
Start
c
a
t
End
ϕ
ϕ
ϕ
ϕ
RNN-T
not gen
not gen
not gen
Start
c
a
t
End
ϕ
ϕ
ϕ
ϕ

HMM, CTC, RNN-T

<u>*HMM*</u>

$$P(X|\mathrm{Y}) = \sum_{h \in align(Y)} P(X|h)$$

<u>*CTC, RNN-T*</u>

$$P(\mathrm{Y}|X) = \sum_{h \in align(Y)} P(h|X)$$

1. Enumerate all the possible alignments

$\boxed{\text{2. How to sum over all the alignments}}$

3. Training:

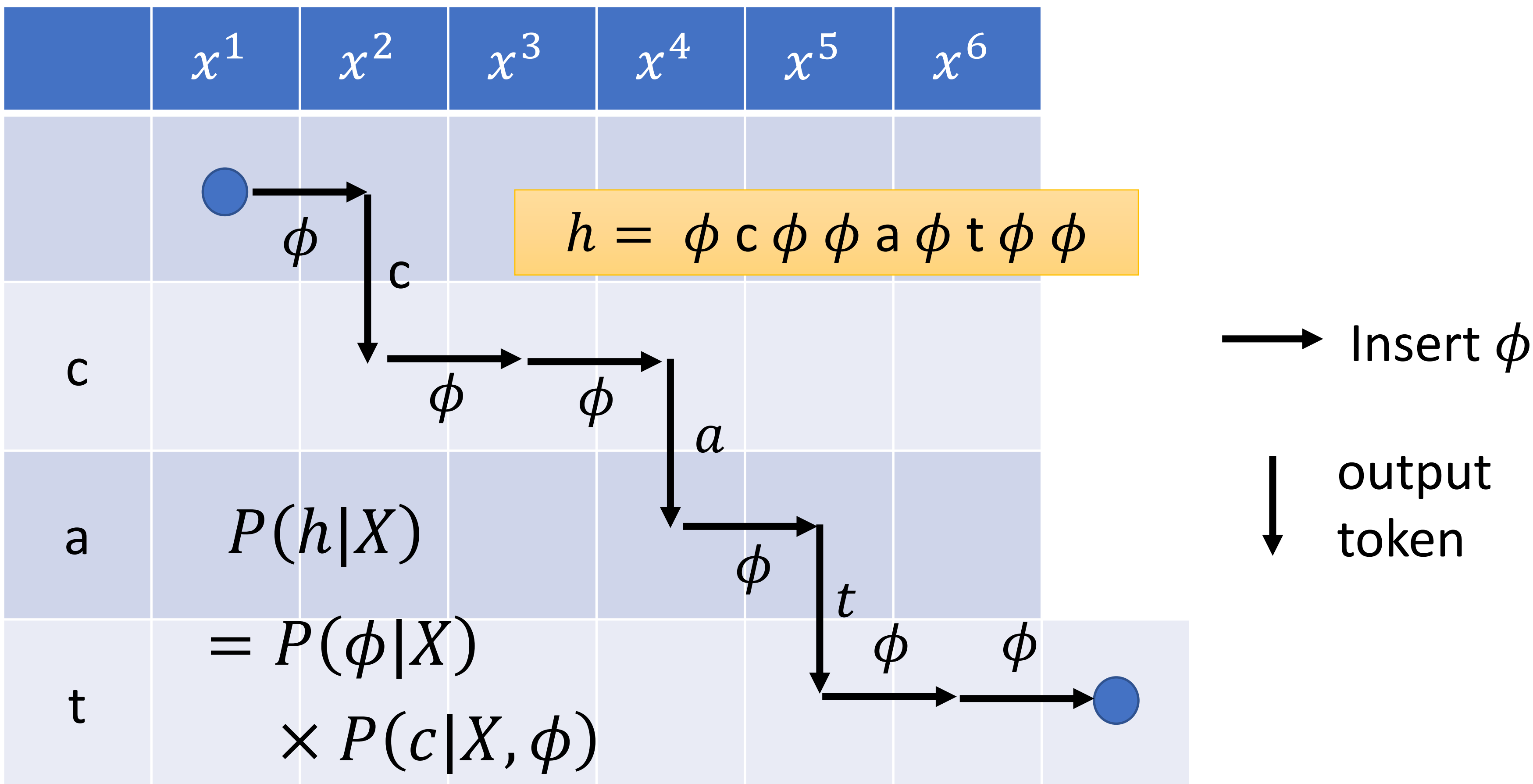
$$\theta^* = arg \max_{\theta} log \mathrm{P}_\theta(\hat{Y}|X) \qquad \frac{\partial \mathrm{P}_\theta(\hat{Y}|X)}{\partial \theta} = ?$$

4. Testing (Inference, decoding):

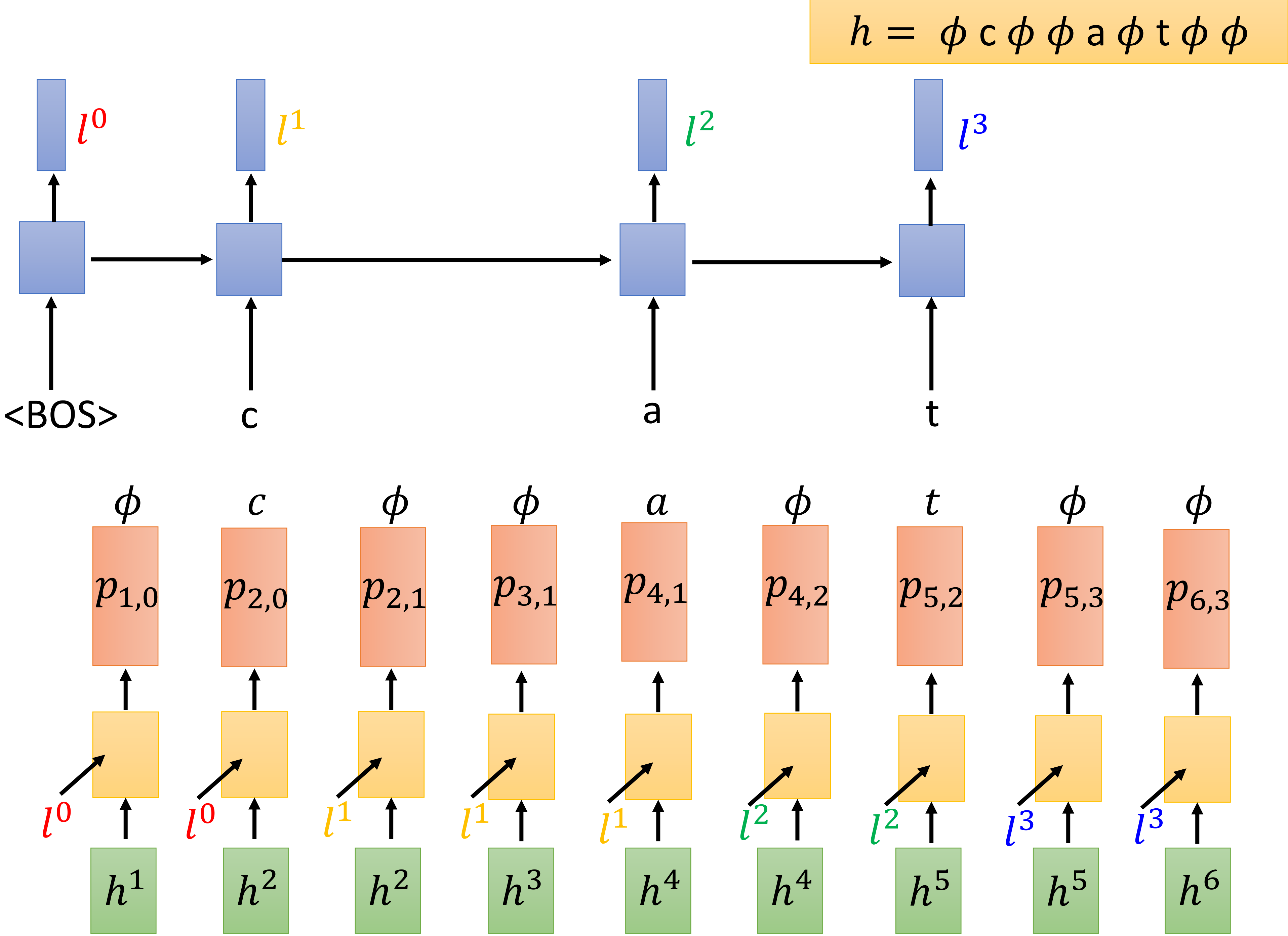
$$\mathrm{Y}^* = arg \max_{\mathrm{Y}} log P(\mathrm{Y}|X)$$

This part is challenging.

Score Computation

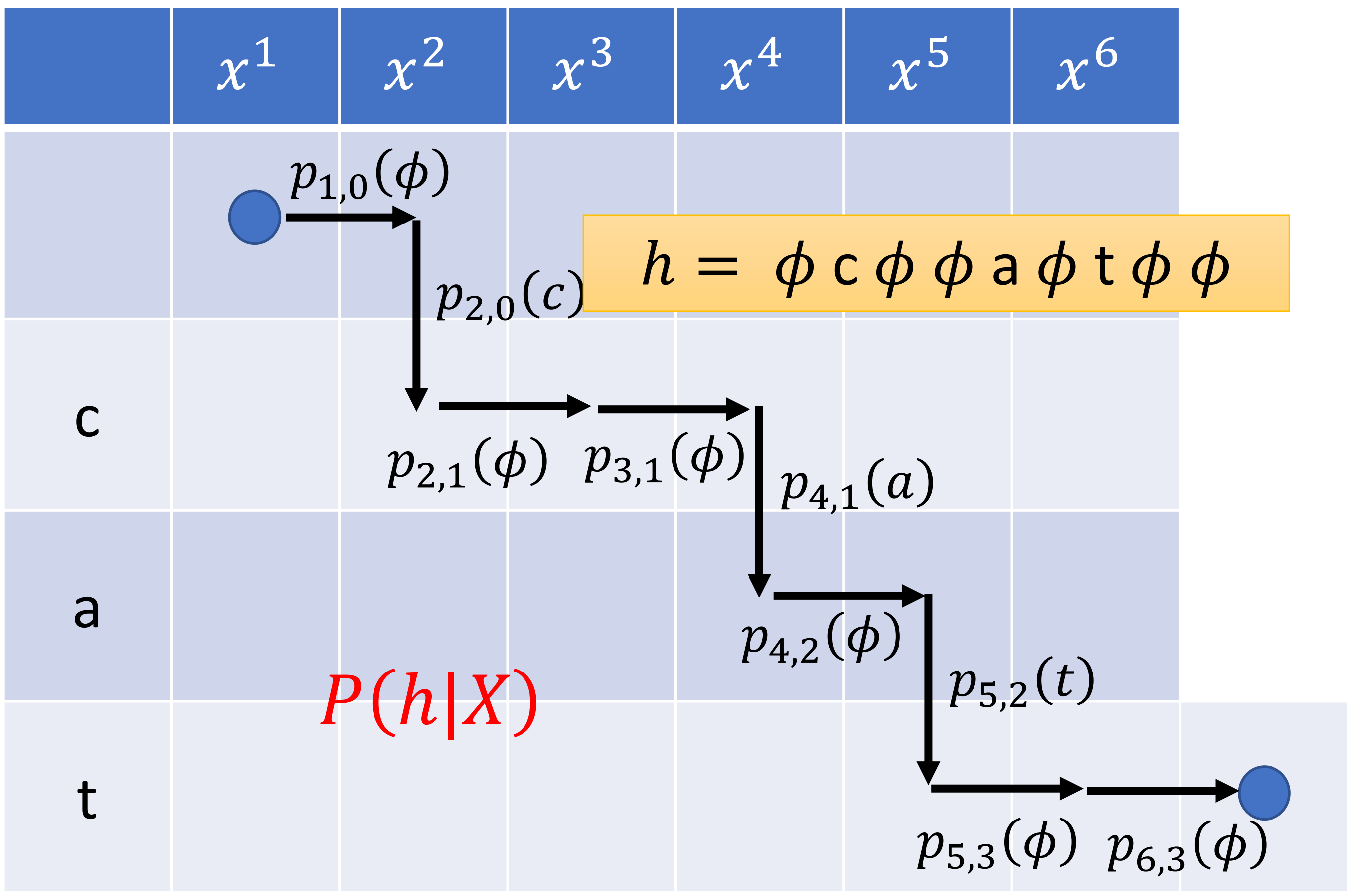


$$
\begin{aligned}
P(h|X) &= P(\phi|X) \\
&\times P(c|X,\phi) \\
&\times P(\phi|X,\phi c) \cdots\cdots
\end{aligned}
$$

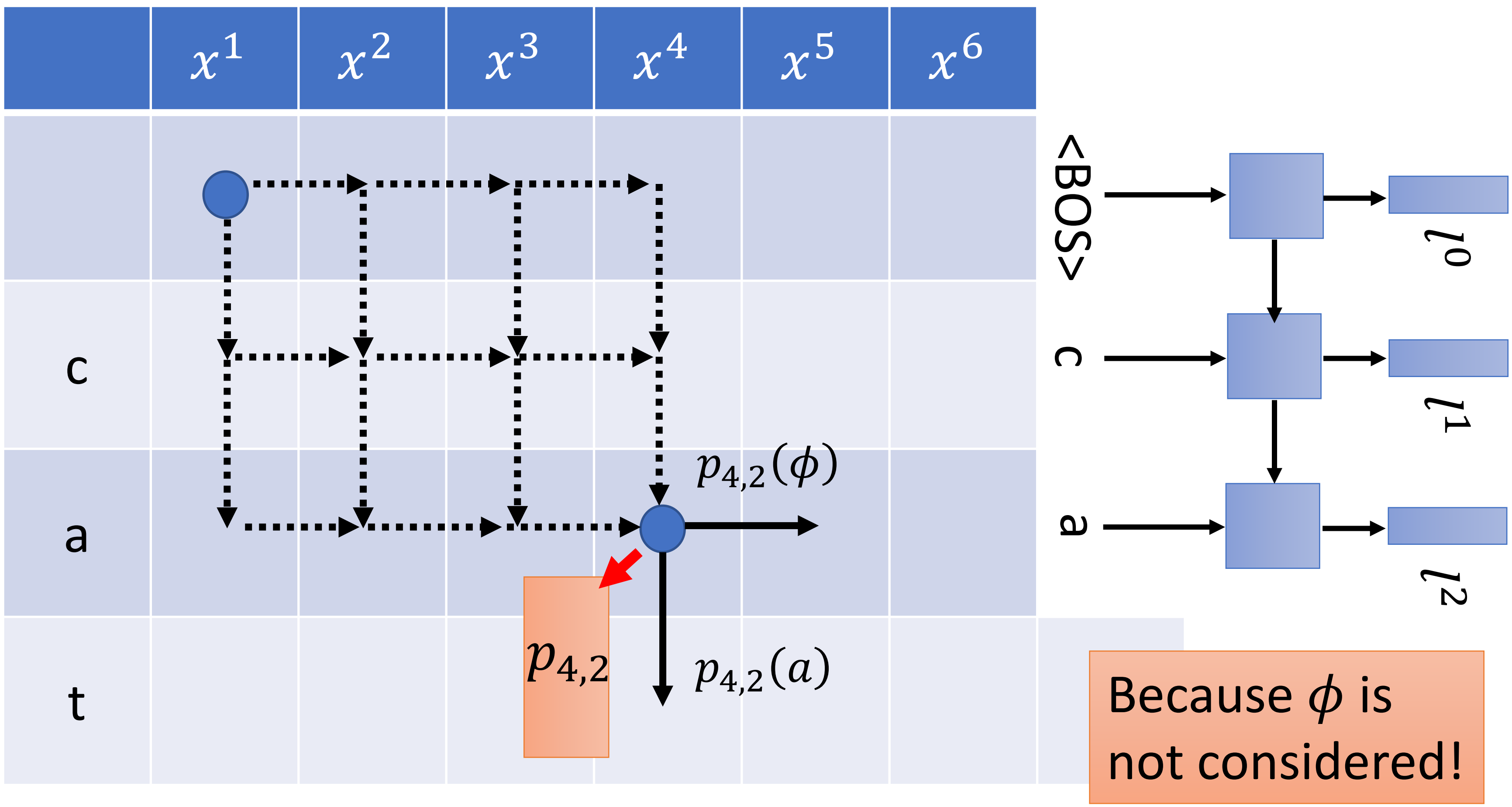


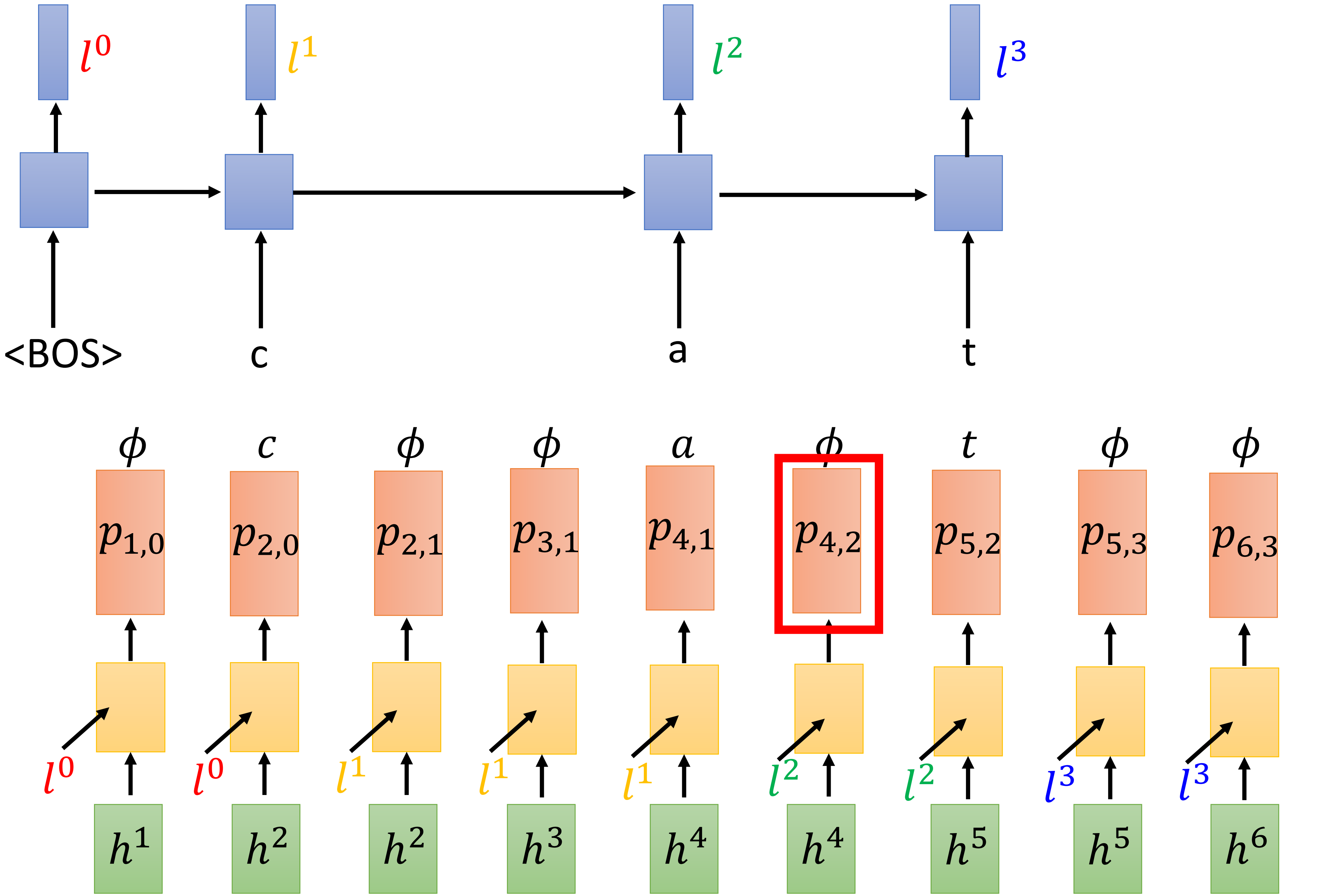
$h = \phi$ c ϕ ϕ a ϕ t ϕ ϕ
l^0
l^1
l^2
l^3
<BOS>
c
a
t
ϕ c ϕ ϕ a ϕ t ϕ ϕ
$p_{1,0}$ $p_{2,0}$ $p_{2,1}$ $p_{3,1}$ $p_{4,1}$ $p_{4,2}$ $p_{5,2}$ $p_{5,3}$ $p_{6,3}$
l^0 l^0 l^1 l^1 l^1 l^2 l^2 l^3 l^3
h^1 h^2 h^2 h^3 h^4 h^4 h^5 h^5 h^6

Score Computation

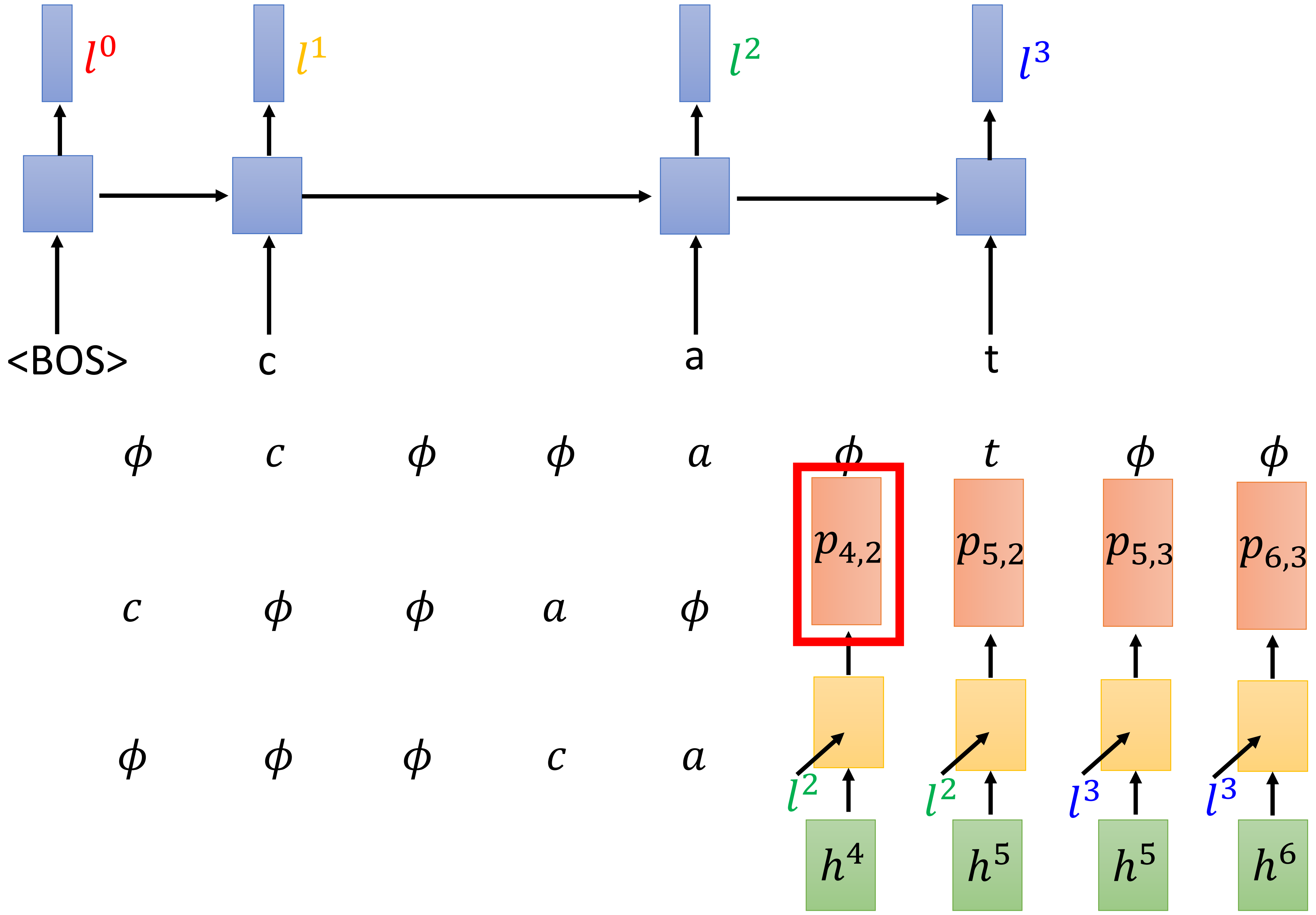


Score Computation





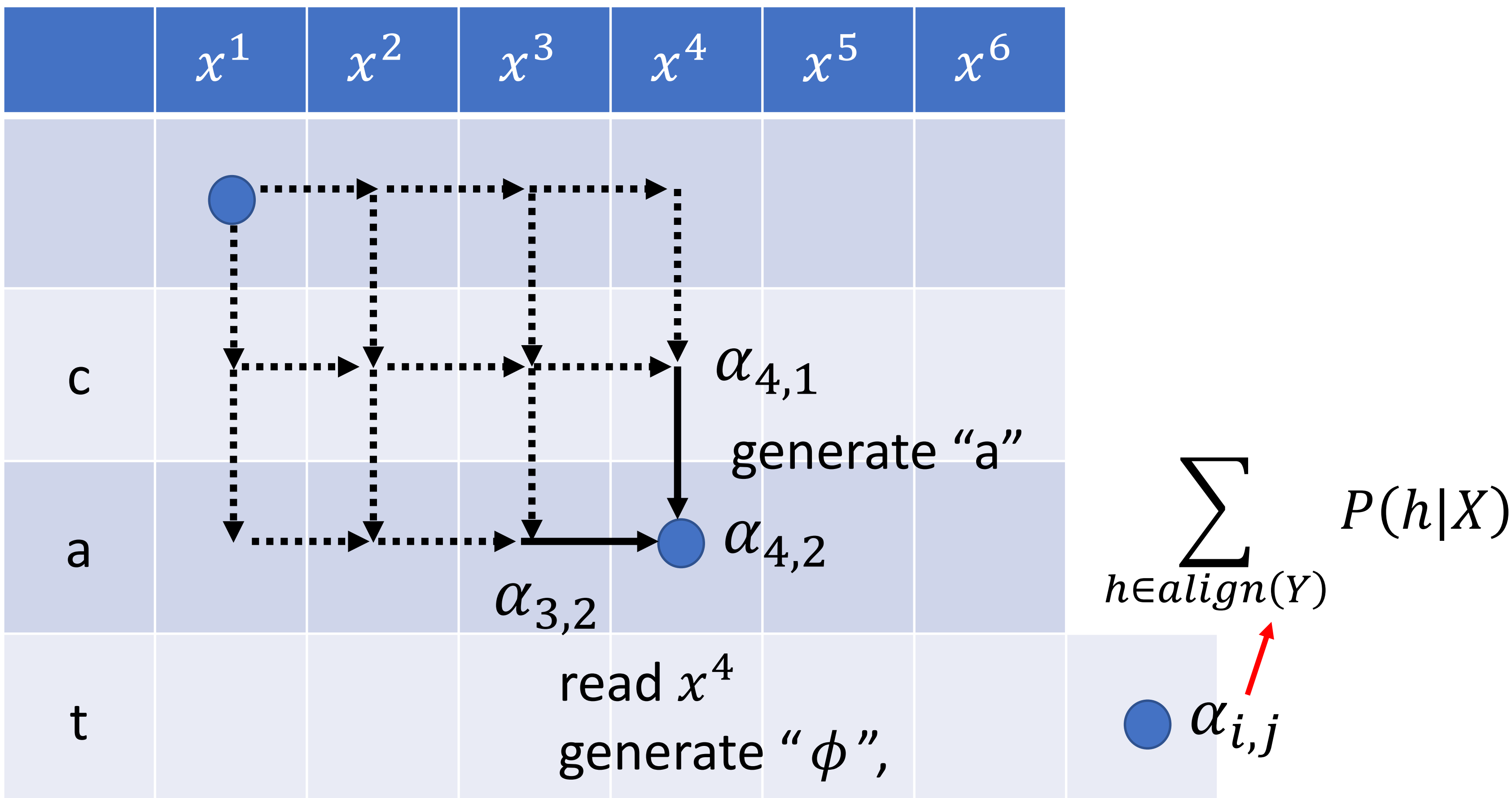
l^0
l^1
l^2
l^3
<BOS>
c
a
t
ϕ c ϕ ϕ a ϕ t ϕ ϕ
$p_{1,0}$ $p_{2,0}$ $p_{2,1}$ $p_{3,1}$ $p_{4,1}$ $p_{4,2}$ $p_{5,2}$ $p_{5,3}$ $p_{6,3}$
l^0 l^0 l^1 l^1 l^1 l^2 l^2 l^3 l^3
h^1 h^2 h^2 h^3 h^4 h^4 h^5 h^5 h^6



l^0
l^1
l^2
l^3
<BOS>
c
a
t
ϕ c ϕ ϕ a ϕ t ϕ ϕ
c ϕ ϕ a ϕ
ϕ ϕ ϕ c a
$p_{4,2}$
$p_{5,2}$
$p_{5,3}$
$p_{6,3}$
l^2
l^2
l^3
l^3
h^4
h^5
h^5
h^6

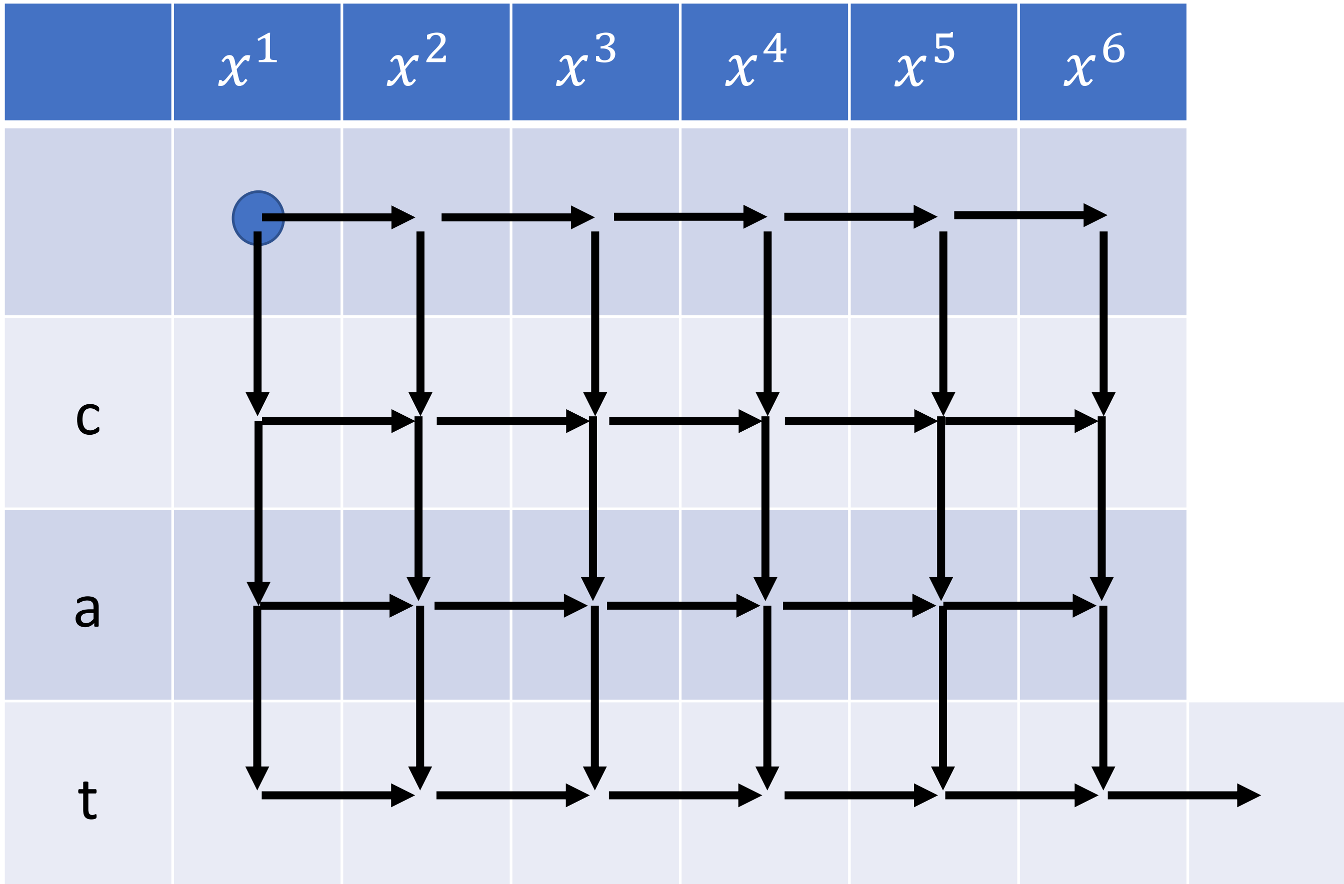
$\alpha_{i,j}$: the summation of the scores of all the alignments that read i-th acoustic features and output j-th tokens

$$\alpha_{4,2} = \alpha_{4,1} p_{4,1}(a) + \alpha_{3,2} p_{3,2}(\phi)$$



$\alpha_{i,j}$: the summation of the scores of all the alignments that read i-th acoustic features and output j-th tokens

$$\alpha_{4,2} = \alpha_{4,1} p_{4,1}(a) + \alpha_{3,2} p_{3,2}(\phi)$$



You can compute summation of the scores of all the alignments.

HMM, CTC, RNN-T

HMM

$$P_\theta(X|Y) = \sum_{h \in align(Y)} P(X|h)$$

CTC, RNN-T

$$P_\theta(Y|X) = \sum_{h \in align(Y)} P(h|X)$$

1. Enumerate all the possible alignments

2. How to sum over all the alignments

3. Training: $\boxed{\theta^* = arg \max_{\theta} log P_\theta(\hat{Y}|X)}$

$$\frac{\partial P_\theta(\hat{Y}|X)}{\partial \theta} = ?$$

4. Testing (Inference, decoding):

$$Y^* = arg \max_{Y} log P(Y|X)$$

Training

$$\theta^* = arg\max_{\theta} logP(\hat{Y}|X)$$

$$P(\hat{Y}|X) = \sum_h P(h|X)$$

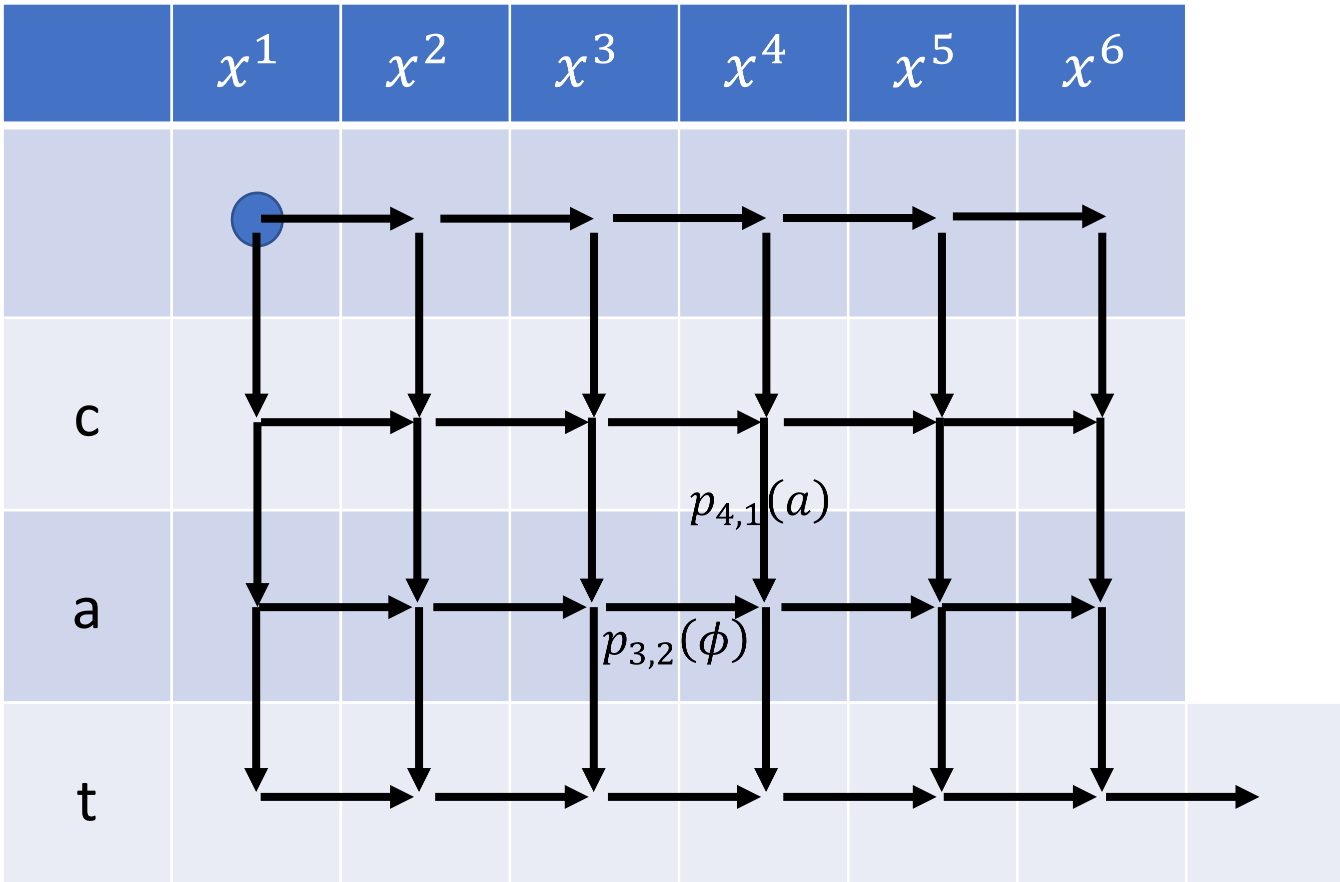
ϕ c ϕ ϕ a ϕ t ϕ ϕ

$$p_{1,0}(\phi) \quad p_{2,0}(c) \quad p_{2,1}(\phi) \quad p_{3,1}(\phi) \quad p_{4,1}(a) \quad p_{4,2}(\phi) \quad p_{5,2}(t) \quad p_{5,3}(\phi) \quad p_{6,3}(\phi)$$

$$\frac{\partial P(\hat{Y}|X)}{\partial \theta} = ?$$

$$P(\hat{Y}|X) = \underbrace{\sum_{h} P(h|X)}$$

$p_{1,0}(\phi)$ $p_{2,0}(c)$ $p_{2,1}(\phi)$ $p_{3,1}(\phi)$ $p_{4,1}(a)$ $p_{4,2}(\phi)$ $p_{5,2}(t)$ $p_{5,3}(\phi)$ $p_{6,3}(\phi)$



Each arrow is a component in $P(\hat{Y}|X) = \sum_{h} P(h|X)$

Training

$$\begin{tikzcd}
 & p_{4,1}(a) \arrow[dr] & \\
\theta \arrow[ur] \arrow[r] \arrow[dr] & p_{3,2}(\phi) \arrow[r] & P(\hat{Y}|X) \\
 & \cdots\cdots \arrow[ur] &
\end{tikzcd}$$

$$\theta^* = arg \max_{\theta} logP(\hat{Y}|X)$$

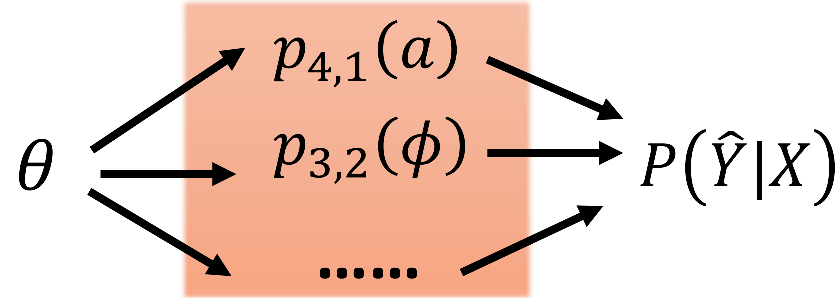
$$P(\hat{Y}|X) = \sum_{h} P(h|X)$$

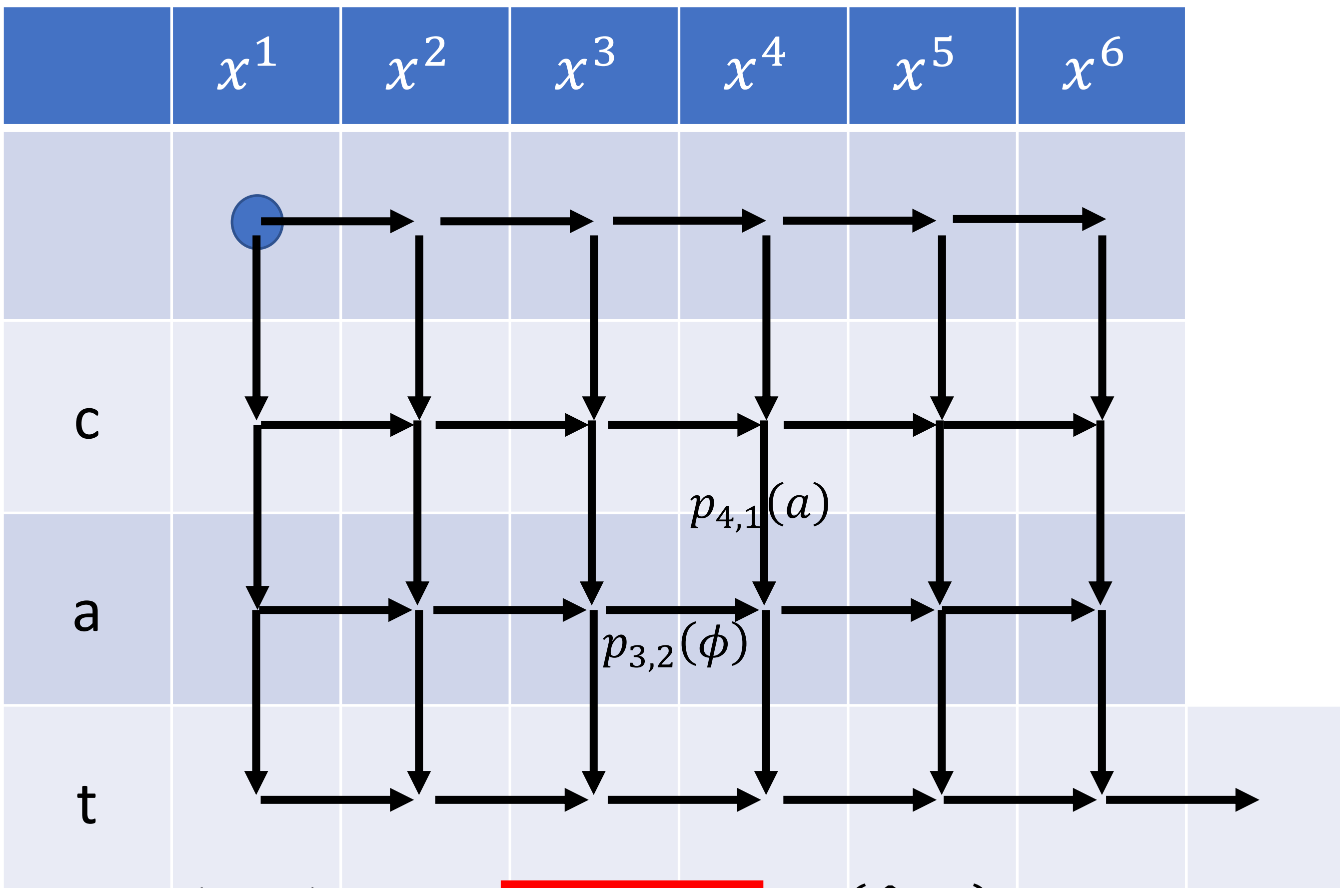
ϕ c ϕ ϕ a ϕ t ϕ ϕ

$$\overbrace{p_{1,0}(\phi) \quad p_{2,0}(c) \quad p_{2,1}(\phi) \quad p_{3,1}(\phi) \quad p_{4,1}(a) \quad p_{4,2}(\phi) \quad p_{5,2}(t) \quad p_{5,3}(\phi) \quad p_{6,3}(\phi)}$$

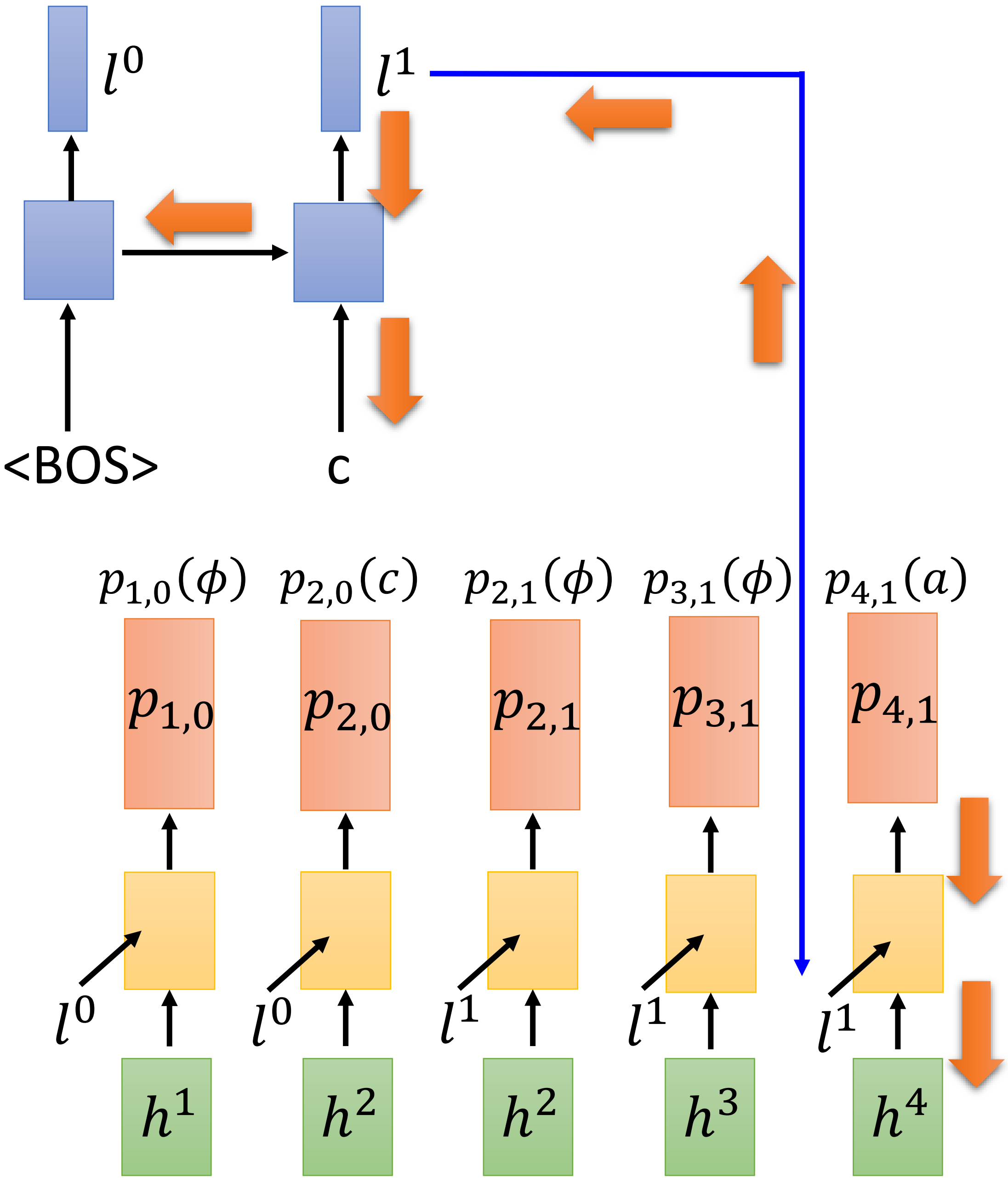
$$\frac{\partial P(\hat{Y}|X)}{\partial \theta} =? \qquad \frac{\partial p_{4,1}(a)}{\partial \theta} \frac{\partial P(\hat{Y}|X)}{\partial p_{4,1}(a)} + \frac{\partial p_{3,2}(\phi)}{\partial \theta} \frac{\partial P(\hat{Y}|X)}{\partial p_{3,2}(\phi)} + \cdots$$

Each arrow is a component





$$\frac{\partial P(\hat{Y}|X)}{\partial \theta} =? \quad \boxed{\frac{\partial p_{4,1}(a)}{\partial \theta}} \frac{\partial P(\hat{Y}|X)}{\partial p_{4,1}(a)} + \frac{\partial p_{3,2}(\phi)}{\partial \theta} \frac{\partial P(\hat{Y}|X)}{\partial p_{3,2}(\phi)} + \cdots$$



To encoder

$$\frac{\partial p_{4,1}(a)}{\partial \theta} = ?$$

Backpropagation
(through time)

$$\frac{\partial P(\hat{Y}|X)}{\partial \theta} = ? \quad \frac{\partial p_{4,1}(a)}{\partial \theta} \boxed{\frac{\partial P(\hat{Y}|X)}{\partial p_{4,1}(a)}} + \frac{\partial p_{3,2}(\phi)}{\partial \theta} \frac{\partial P(\hat{Y}|X)}{\partial p_{3,2}(\phi)} + \cdots$$

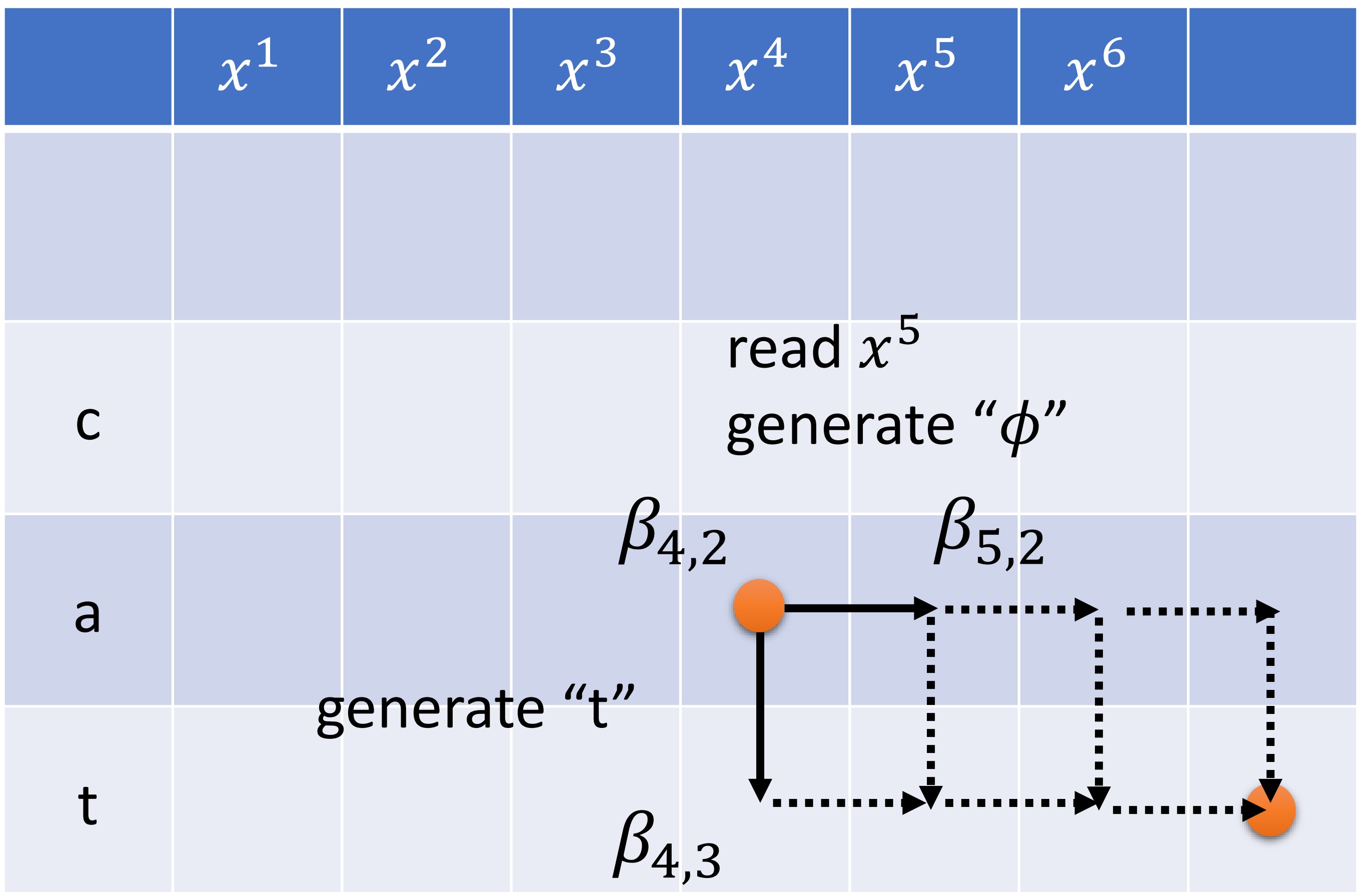
$$P(\hat{Y}|X) = \sum_{h \text{ with } p_{4,1}(a)} \underbrace{\boxed{P(h|X)}}_{p_{4,1}(a) \times other} + \sum_{h \text{ without } p_{4,1}(a)} P(h|X)$$

$$\frac{\partial P(\hat{Y}|X)}{\partial p_{4,1}(a)} = \sum_{h \text{ with } p_{4,1}(a)} other = \sum_{h \text{ with } p_{4,1}(a)} \frac{P(h|X)}{p_{4,1}(a)}$$

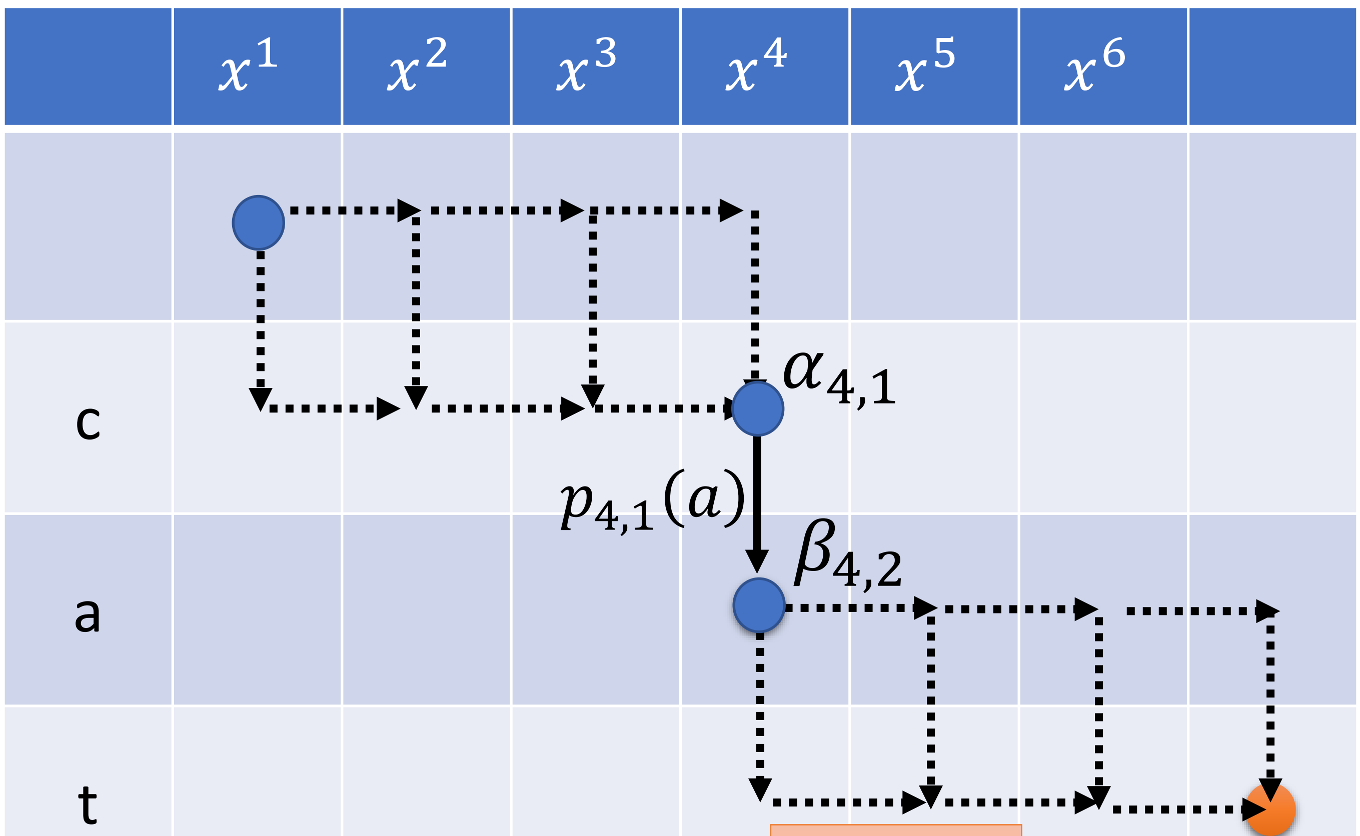
$$= \frac{1}{p_{4,1}(a)} \sum_{h \text{ with } p_{4,1}(a)} P(h|X)$$

$\beta_{i,j}$: the summation of the score of all the alignments staring from i-th acoustic features and j-th tokens

$$\beta_{4,2} = \beta_{4,3} p_{4,2}(t) + \beta_{5,2} p_{4,2}(\phi)$$



$$\frac{\partial P(\hat{Y}|X)}{\partial p_{4,1}(a)} = \frac{1}{p_{4,1}(a)} \boxed{\sum_{a \text{ with } p_{4,1}(a)} P(h|X)} \; \alpha_{4,1}\, p_{4,1}(a) \beta_{4,2}$$



$$\frac{\partial P(\hat{Y}|X)}{\partial \theta} =? \quad \frac{\partial p_{4,1}(a)}{\partial \theta} \boxed{\alpha_{4,1}\beta_{4,2}} + \frac{\partial p_{3,2}(\phi)}{\partial \theta} \frac{\partial P(\hat{Y}|X)}{\partial p_{3,2}(\phi)} + \cdots$$

HMM, CTC, RNN-T

HMM

$$\mathrm{P}_\theta(X|\mathrm{Y}) = \sum_{h \in align(Y)} P(X|h)$$

CTC, RNN-T

$$\mathrm{P}_\theta(\mathrm{Y}|X) = \sum_{h \in align(Y)} P(h|X)$$

1. Enumerate all the possible alignments

2. How to sum over all the alignments

3. Training: $\theta^* = arg \max_\theta log \mathrm{P}_\theta(\hat{Y}|X)$ $\quad \dfrac{\partial \mathrm{P}_\theta(\hat{Y}|X)}{\partial \theta} = ?$

4. Testing (Inference, decoding):

$$\mathrm{Y}^* = arg \max_{\mathrm{Y}} log P(\mathrm{Y}|X)$$

Decoding

$$\mathrm{Y}^* = arg \max_{\mathrm{Y}} log\mathrm{P}(\mathrm{Y}|X)$$

理想 $$= arg \max_{\mathrm{Y}} log \boxed{\sum_{h \in align(Y)} P(h|X)} \quad \max_{h \in align(Y)} P(h|X)$$

現實 $$\approx arg \max_{\mathrm{Y}} \max_{h \in align(Y)} logP(h|X)$$

$$\mathrm{Y}^* = align^{-1}(h^*)$$

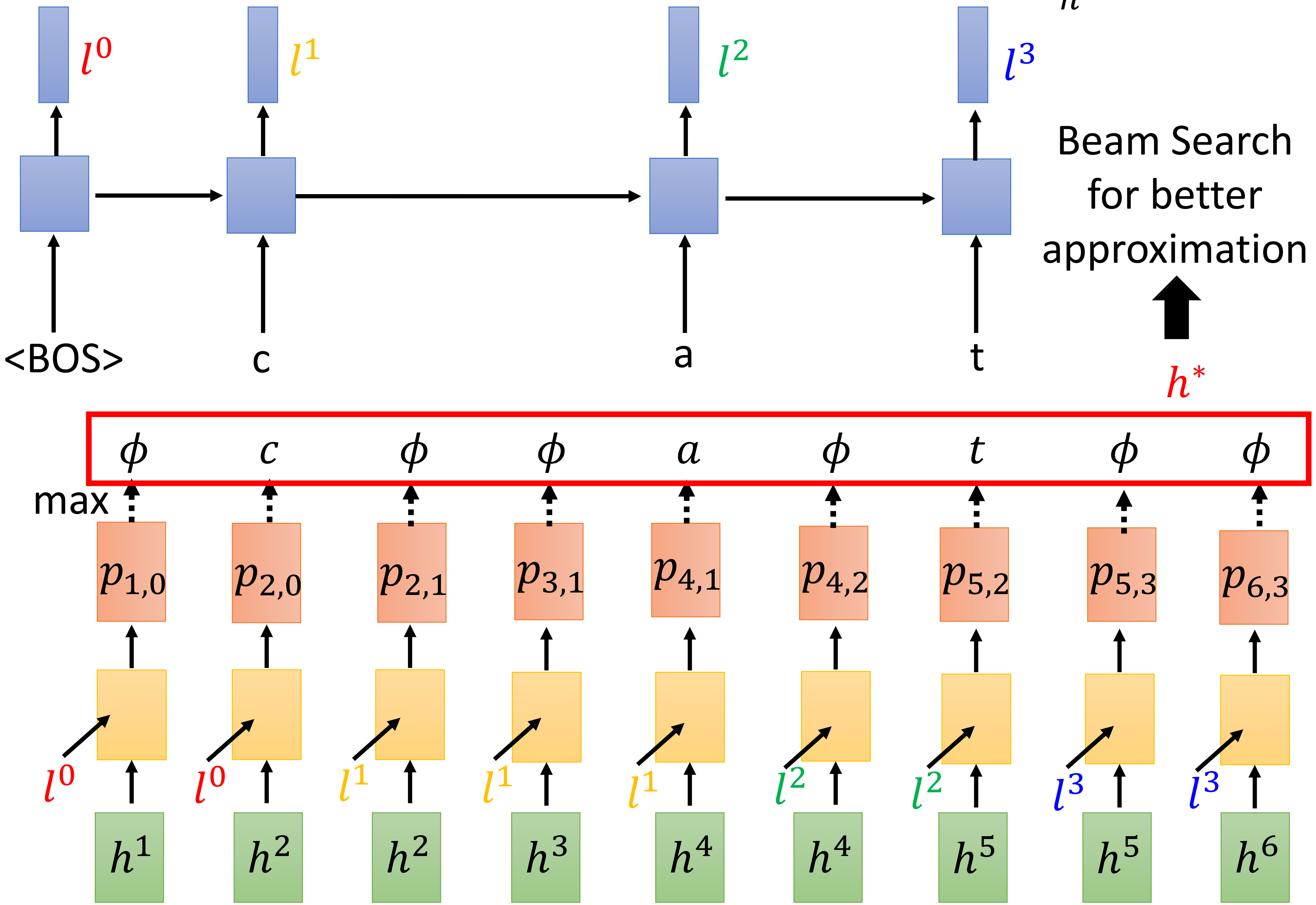
$$h^* = arg \max_{h} logP(h|X)$$

$h_1 \; h_2 \; h_3$

$h = \phi\ \mathrm{c}\ \phi\ \phi\ \mathrm{a}\ \phi\ \mathrm{t}\ \phi\ \phi\ \ldots$

$$P(h|X) = P(h_1|X)P(h_2|X, h_1)P(h_3|X, h_1, h_2)\ \ldots$$

$$h^* = arg \max_h logP(h|X)$$



Summary

	LAS	CTC	RNN-T
Decoder	dependent	independent	dependent
Alignment	not explicit (soft alignment)	Yes	Yes
Training	just train it	sum over alignment	sum over alignment
On-line	No	Yes	Yes

Reference

- [Yu, et al., INTERSPEECH’16] Dong Yu, Wayne Xiong, Jasha Droppo, Andreas Stolcke , Guoli Ye, Jinyu Li , Geoffrey Zweig, Deep Convolutional Neural Networks with Layer-wise Context Expansion and Attention, INTERSPEECH, 2016
- [Saon, et al., INTERSPEECH’17] George Saon, Gakuto Kurata, Tom Sercu, Kartik Audhkhasi, Samuel Thomas, Dimitrios Dimitriadis, Xiaodong Cui, Bhuvana Ramabhadran, Michael Picheny, Lynn-Li Lim, Bergul Roomi, Phil Hall, English Conversational Telephone Speech Recognition by Humans and Machines, INTERSPEECH, 2017
- [Povey, et al., ICASSP’10] Daniel Povey, Lukas Burget, Mohit Agarwal, Pinar Akyazi, Kai Feng, Arnab Ghoshal, Ondrej Glembek, Nagendra Kumar Goel, Martin Karafiat, Ariya Rastrow, Richard C. Rose, Petr Schwarz, Samuel Thomas, Subspace Gaussian Mixture Models for speech recognition, ICASSP, 2010
- [Mohamed , et al., ICASSP’10] Abdel-rahman Mohamed and Geoffrey Hinton, Phone recognition using Restricted Boltzmann Machines, ICASSP, 2010