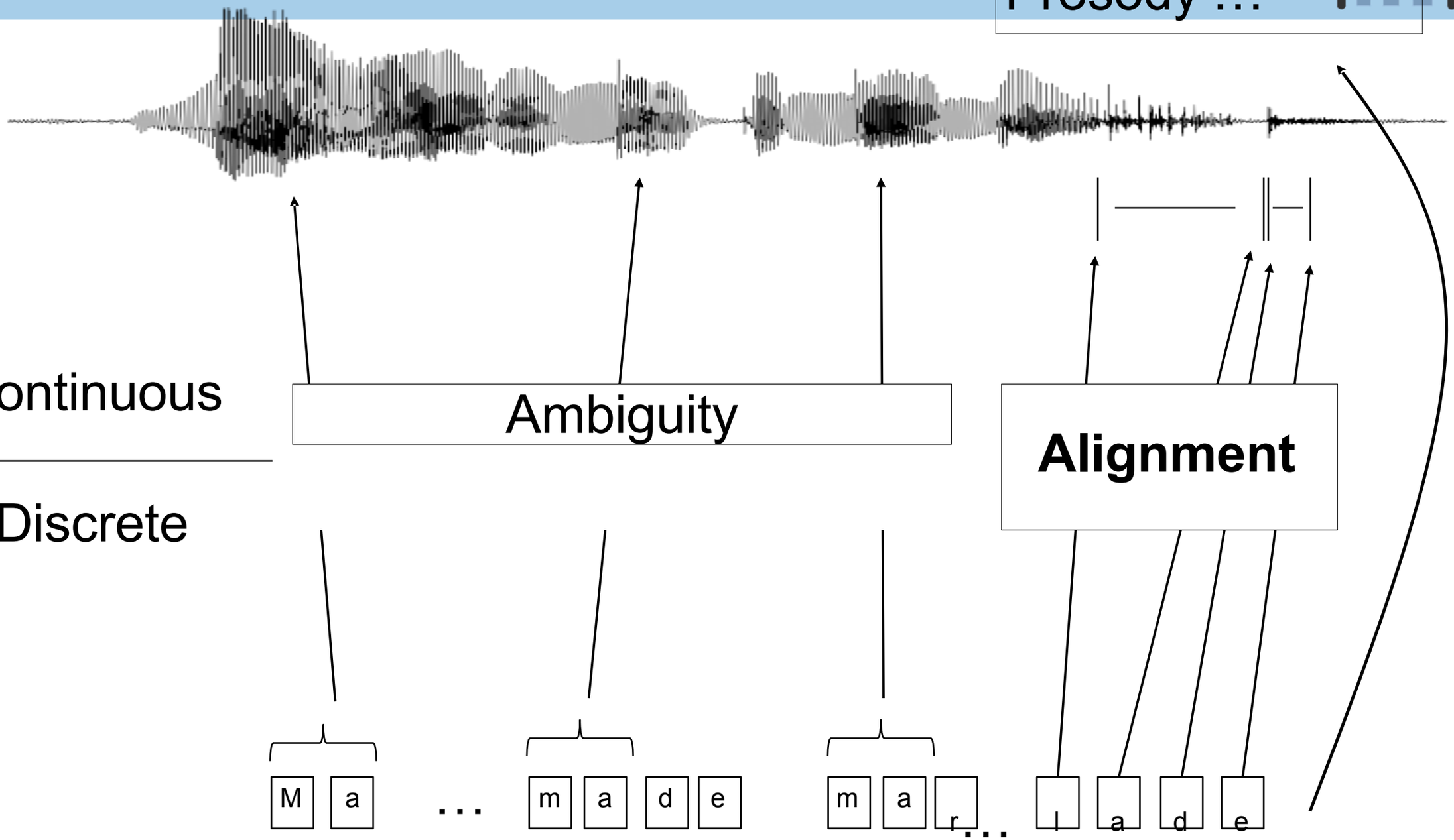
Neural TTS + Spoken LM

EECS 183/283a: Natural Language Processing

Some slides from Xing Wang/ Yamagishi/ Watanabe/ Black/ Tan

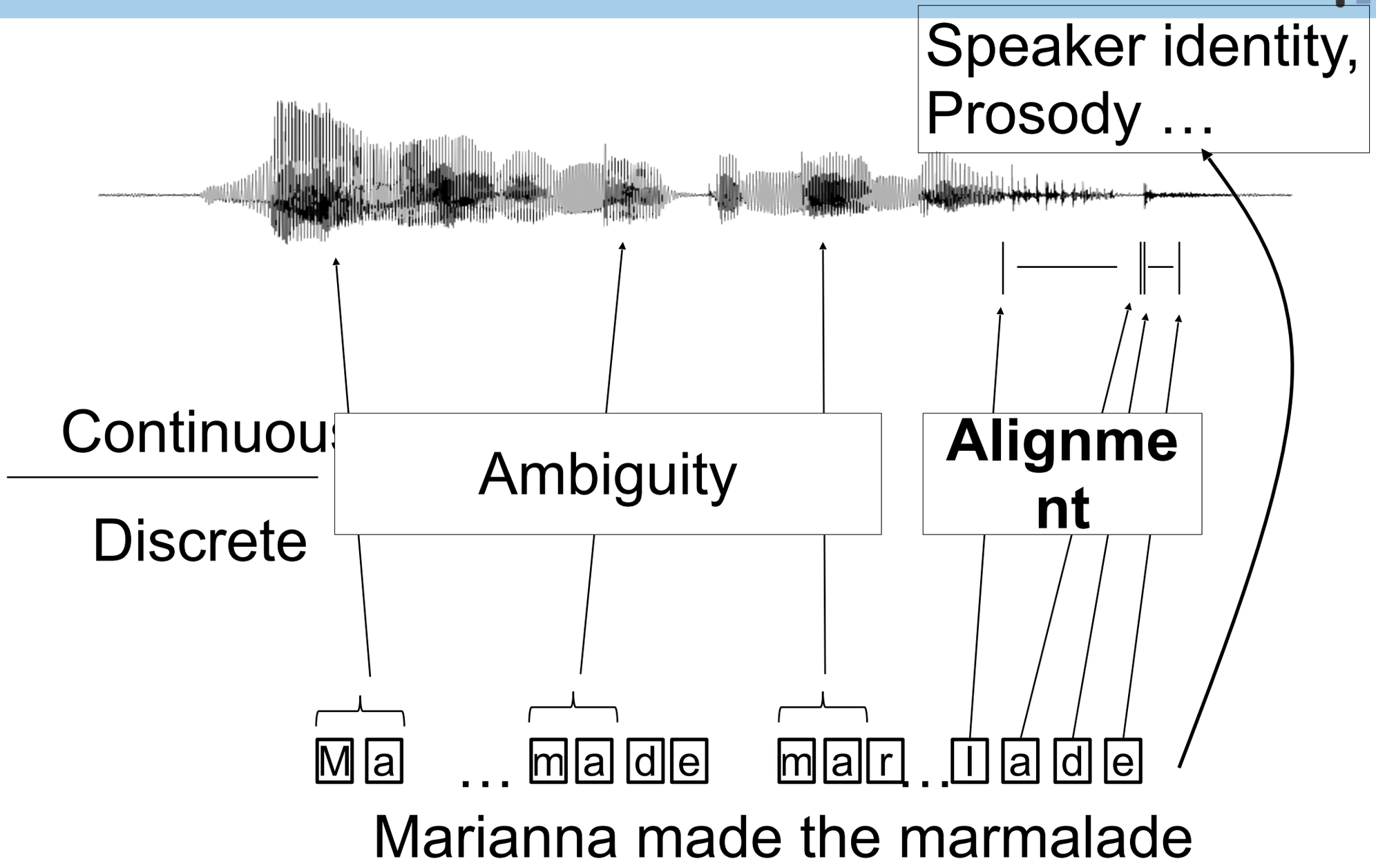
RECAP – THE TTS PROBLEM

Speaker identity,
Prosody …

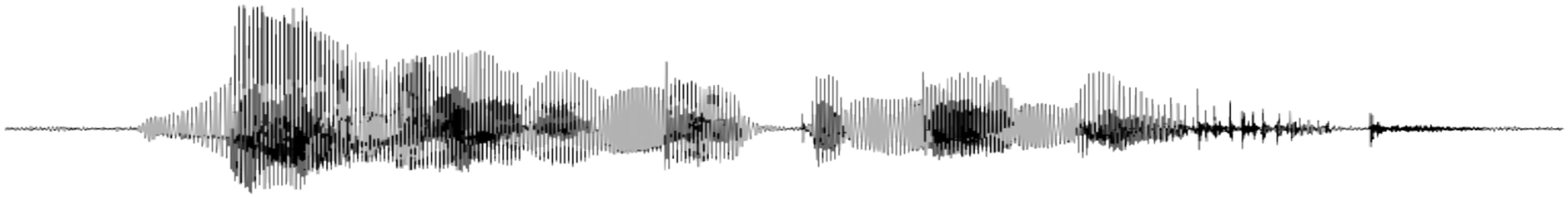


Sentence from: Beckman, M. E. & Ayers, G. Guidelines for ToBI labelling. OSU Res. Found. 3, (1997)

Text-to-speech synthesis



Text-to-speech synthesis



Waveform generation

Acoustic realization

+Prosody tags

To phone

Normali-zation

H* H* L-L%

M AA R IY AA N AH M EY D DH AH M AA R M AH L EY D

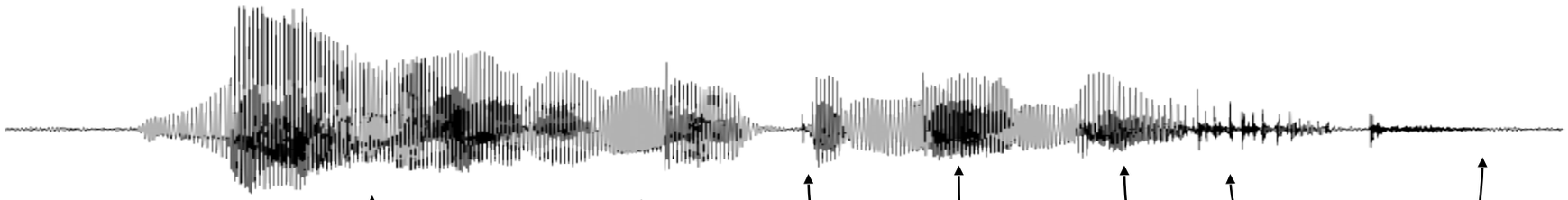
M AA R IY AA N AH M EY D DH AH M AA R M AH L EY D

M a … m a d e m a r … l a d e

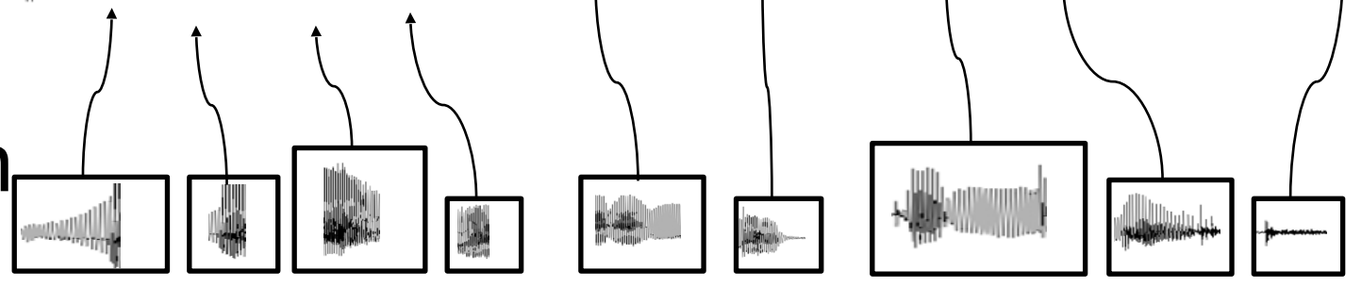
Marianna made the marmalade

Sentence from: Beckman, M. E. & Ayers, G. Guidelines for ToBI labelling. OSU Res. Found. 3, (1997)
LOGIOS Lexicon tool: http://www.speech.cs.cmu.edu/tools/lextool.html
H*, L-L%: ToBI labels Beckman, M. E. & Ayers, G. Guidelines for ToBI labelling. OSU Res. Found. 3, (1997)

Text-to-speech synthesis – Unit selection



Waveform concatenation



Acoustic realization

H* H* L-L%

+Prosody tags

M AA R IY AA N AH M EY D DH AH M AA

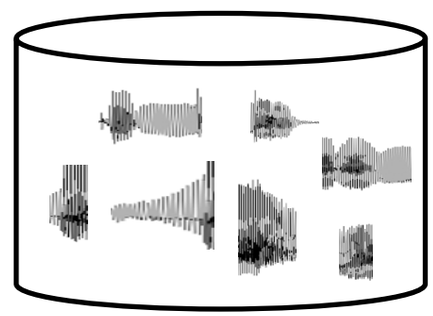
To phone

M AA R IY AA N AH M EY D DH AH M AA R M AH L EY D

Normali-zation

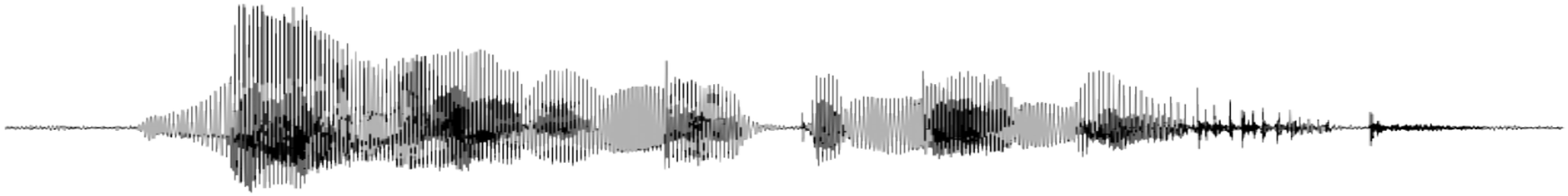
M a … m a d e m a r … l a d e

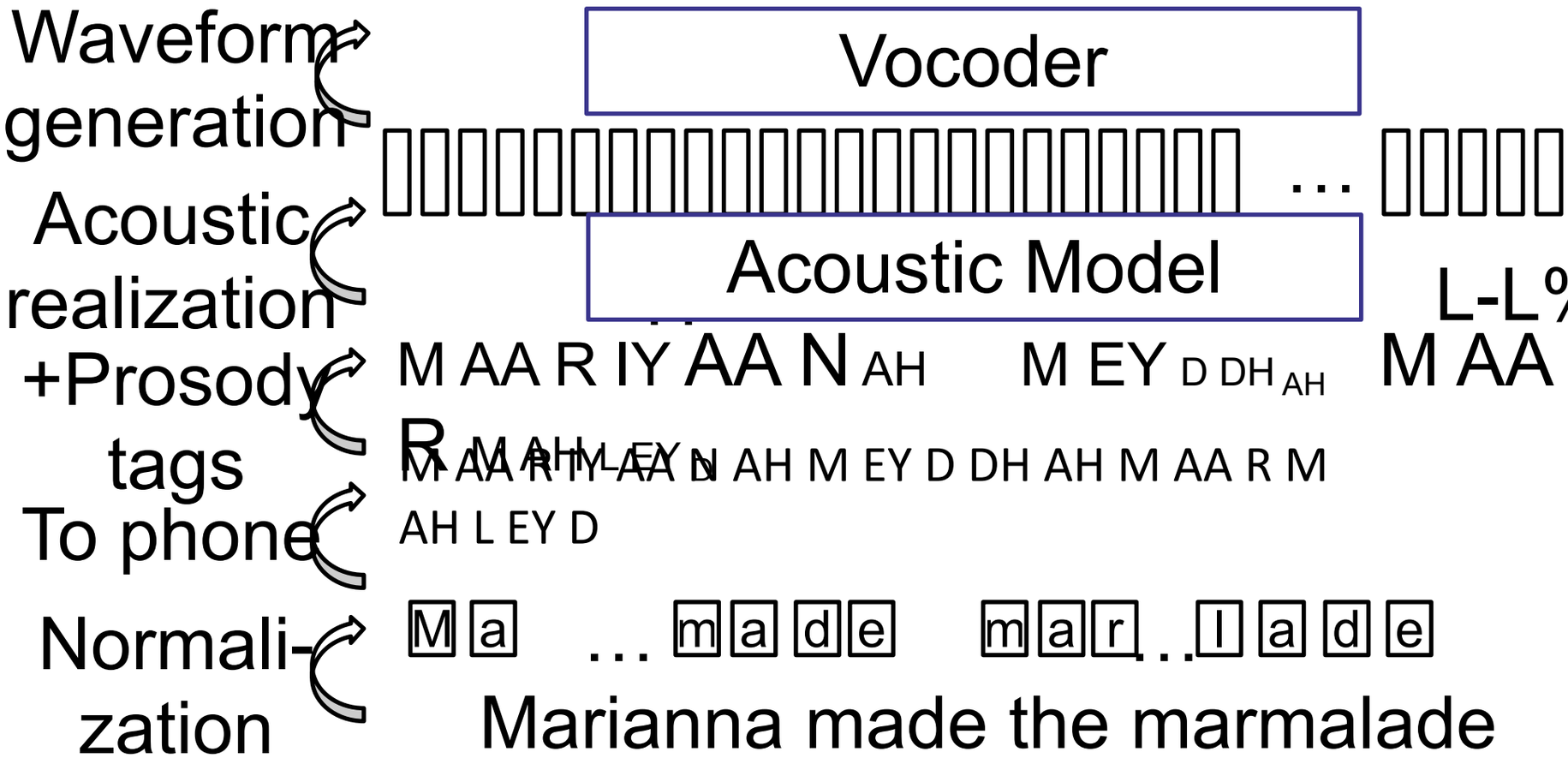
Marianna made the marmalade



Hunt, A. J. & Black, A. W. Unit selection in a concatenative speech synthesis system using a large speech database. in Proc. ICASSP 373–376 (1996).
Black, A. W. & Taylor, P. A. Automatically clustering similar units for unit selection in speech synthesis. (1997)

Text-to-speech synthesis – Statistical



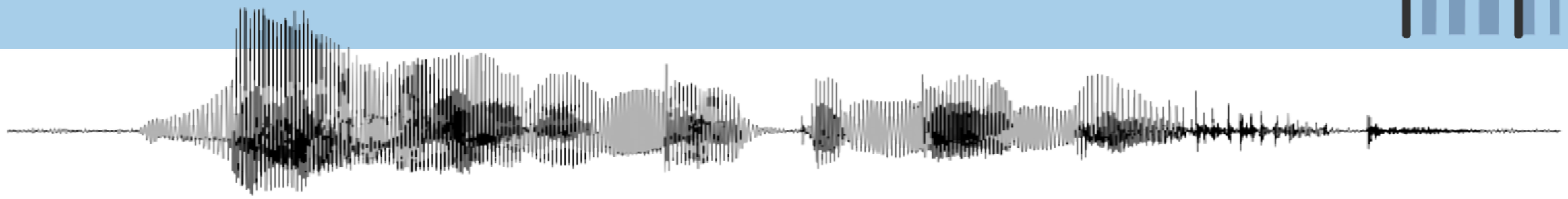


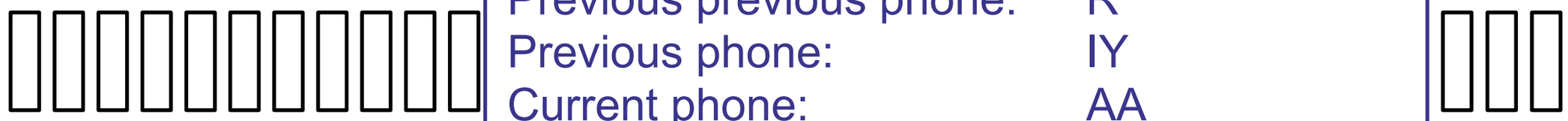
Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T. & Kitamura, T. Speech parameter generation algorithms for HMM-based speech synthesis. in Proc. ICASSP 936–939 (2000).

Keiichi Tokuda, Heiga Zen, and Alan W Black. An HMM-Based Speech Synthesis System Applied to English. In *Proc. SSW*, 227–230. 2002.

HMM-Based Speech Synthesis Toolkit (HTS), home page: http://hts.sp.nitech.ac.jp/?Welcome

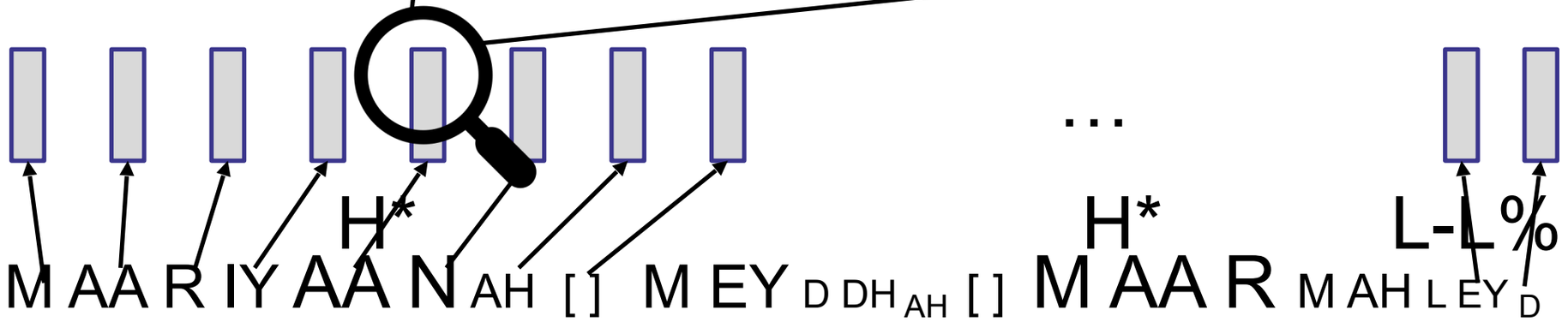
Overview – Statistical speech synthesis system





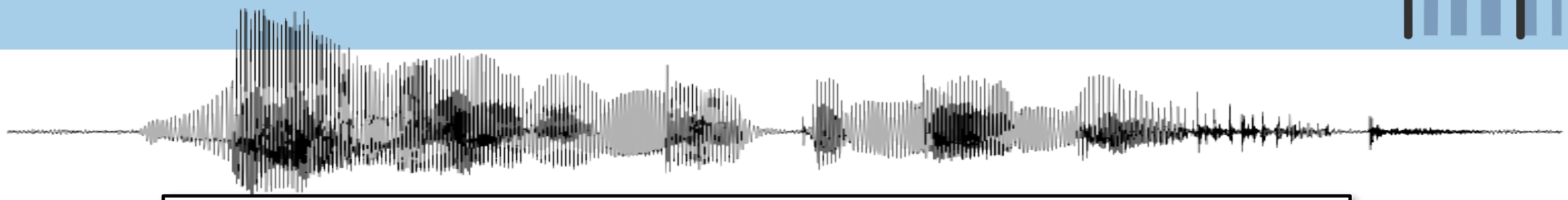
Context-dependent labels (or linguistic feature vector)

Previous previous phone: R
Previous phone: IY
Current phone: AA
Next phone: N
Next Next phone: AH
Stressed syllable: True
Word position: 1
…



Marianna made the marmalade

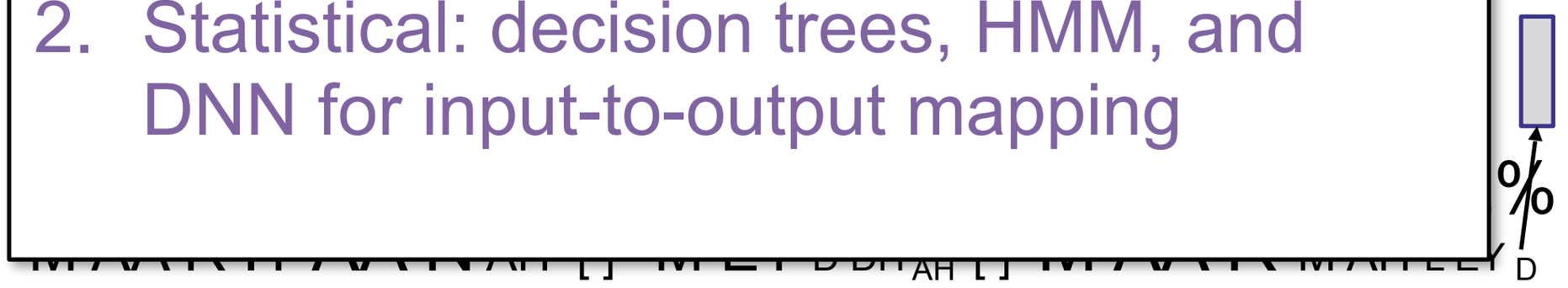
OVERVIEW – STATISTICAL SPEECH SYNTHESIS SYSTEM



Statistical parametric speech synthesis (SPSS)

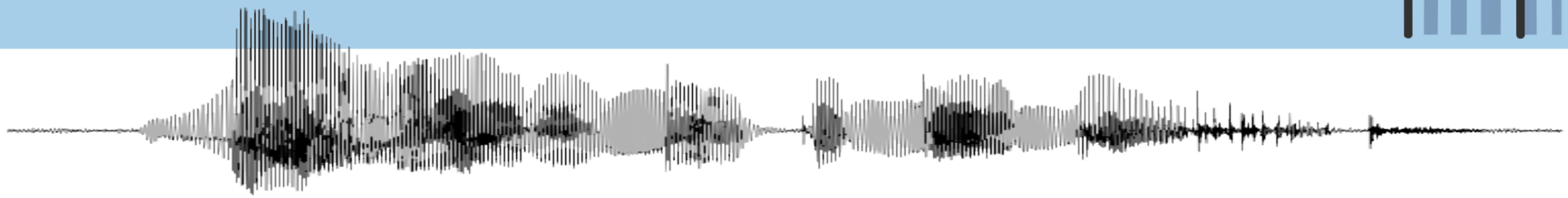
1. Parametric: speech parameterized as acoustic features vectors

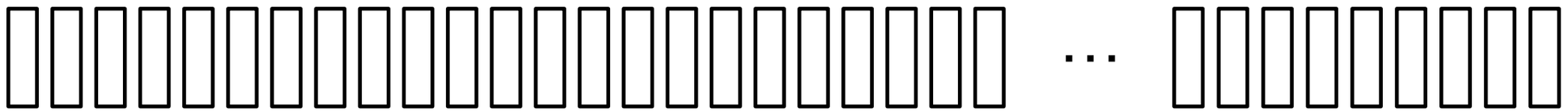
2. Statistical: decision trees, HMM, and DNN for input-to-output mapping



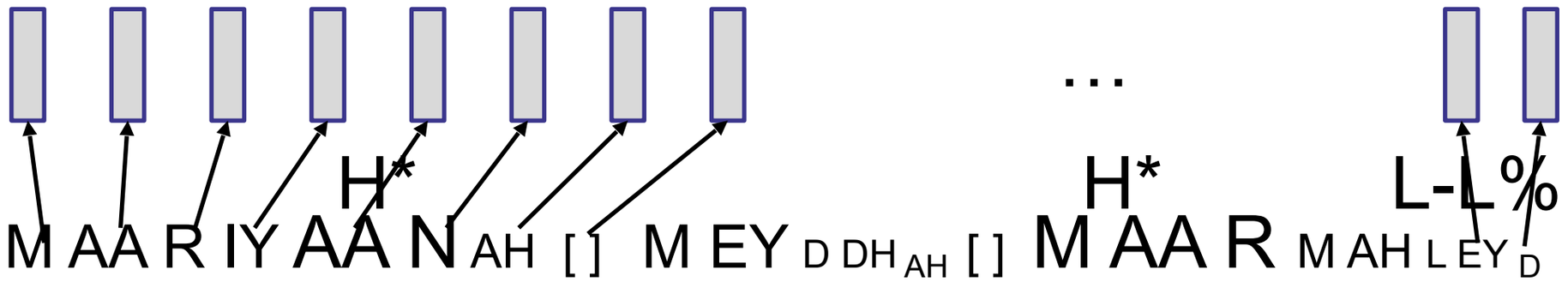
Marianna made the marmalade

OVERVIEW – STATISTICAL SPEECH SYNTHESIS SYSTEM



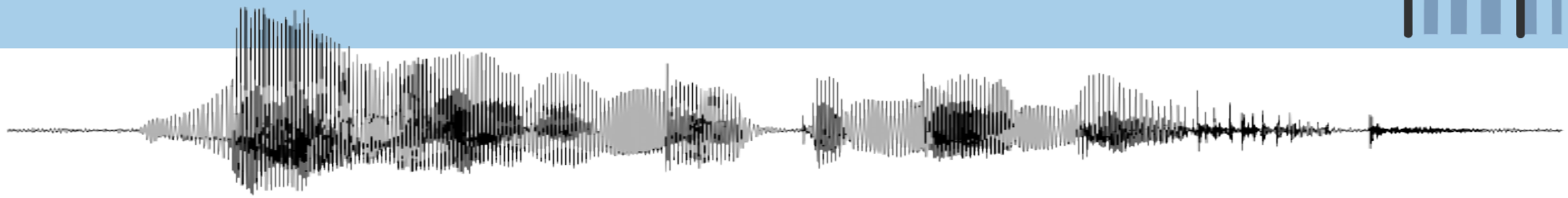


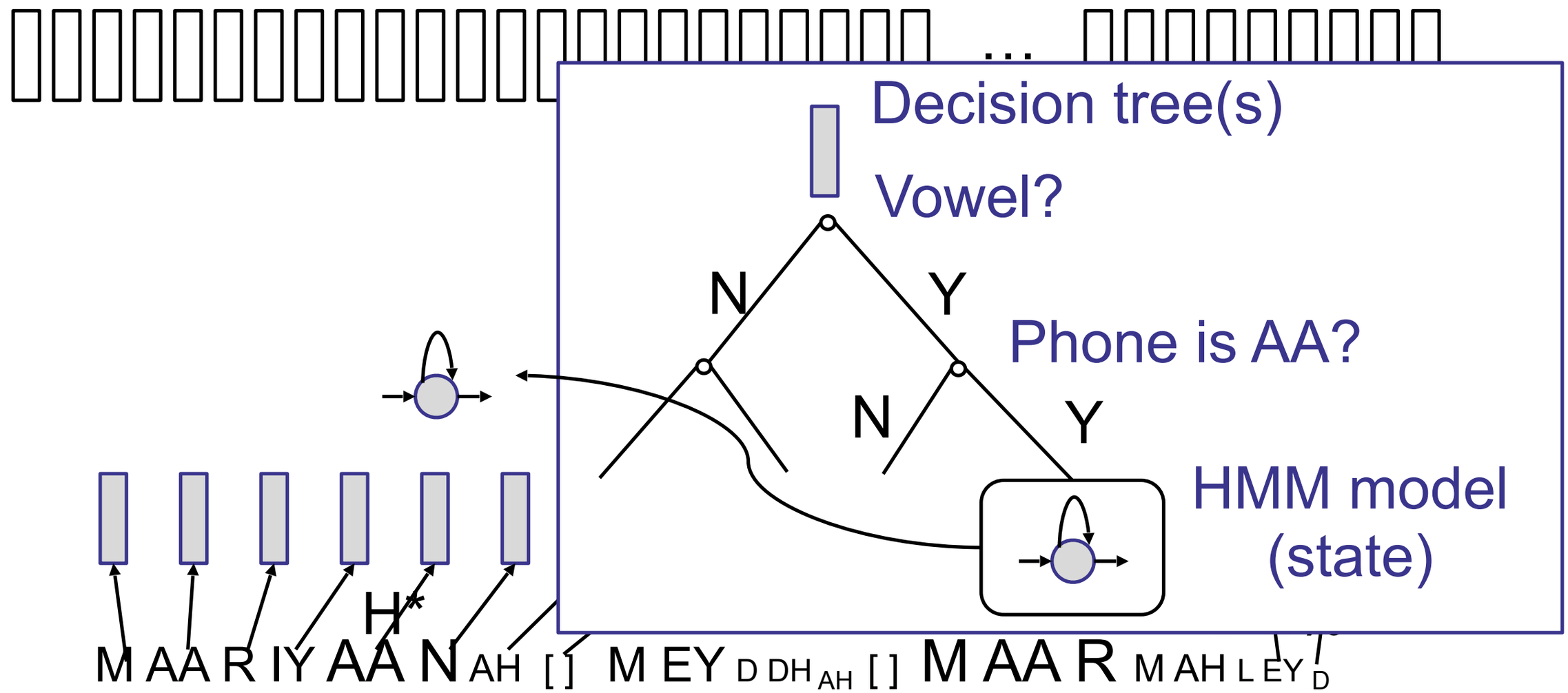
Input sequence is short,
Output sequence is long!?



Marianna made the marmalade

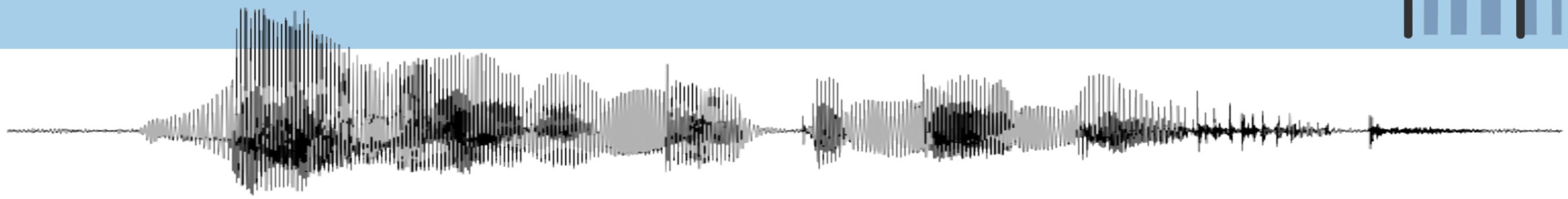
OVERVIEW – STATISTICAL SPEECH SYNTHESIS SYSTEM

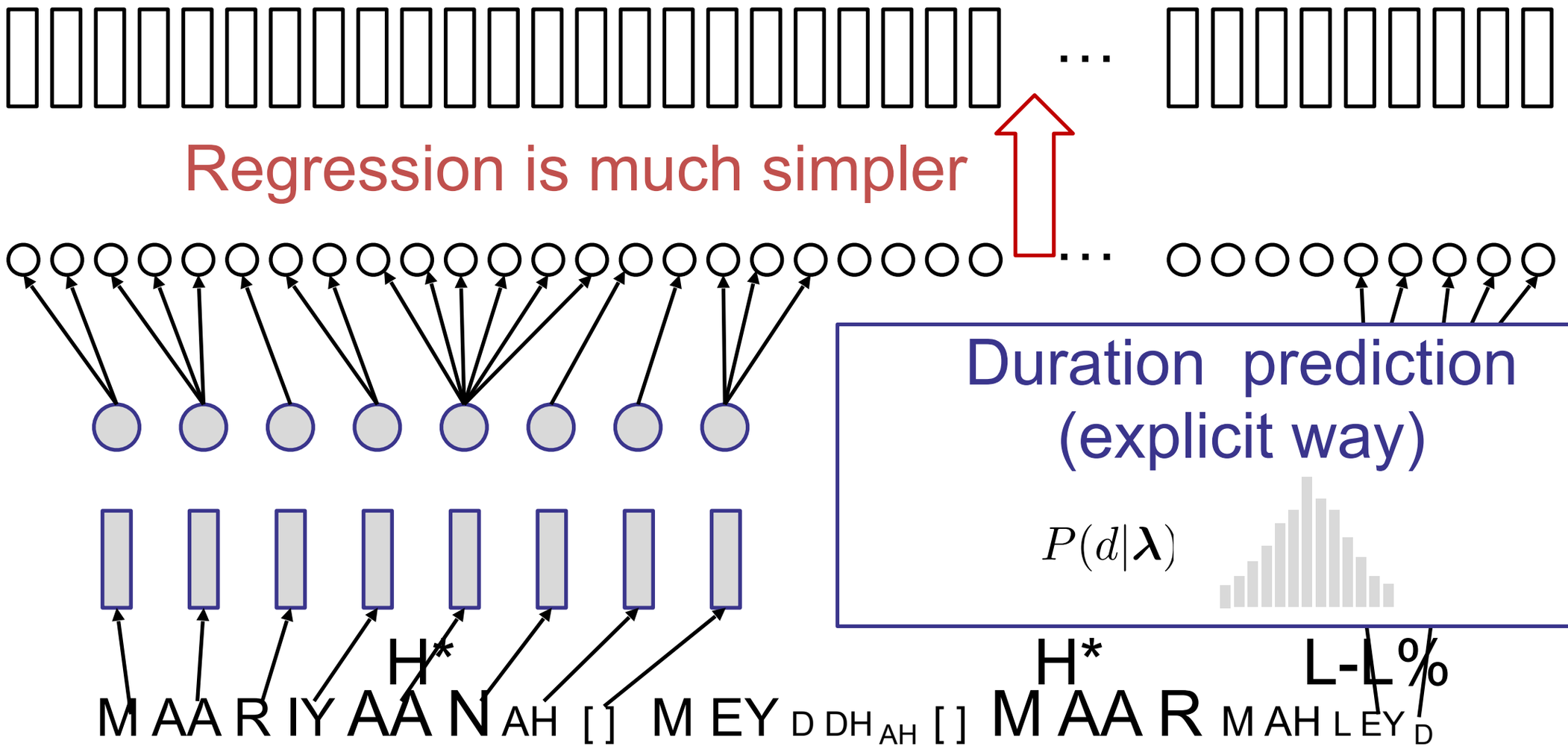




Marianna made the marmalade

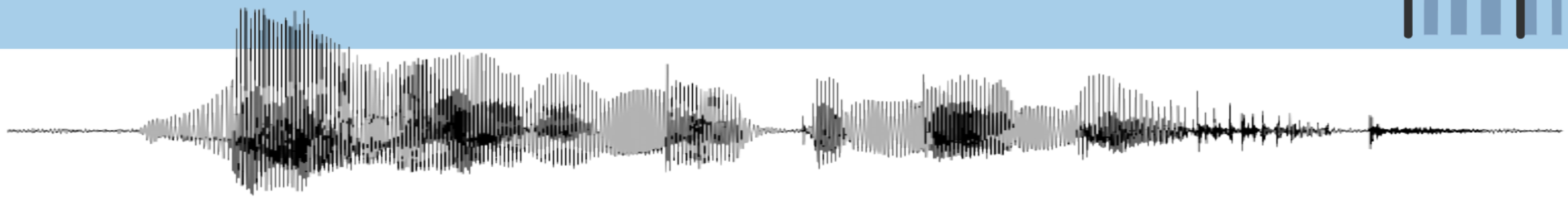
OVERVIEW – STATISTICAL SPEECH SYNTHESIS SYSTEM

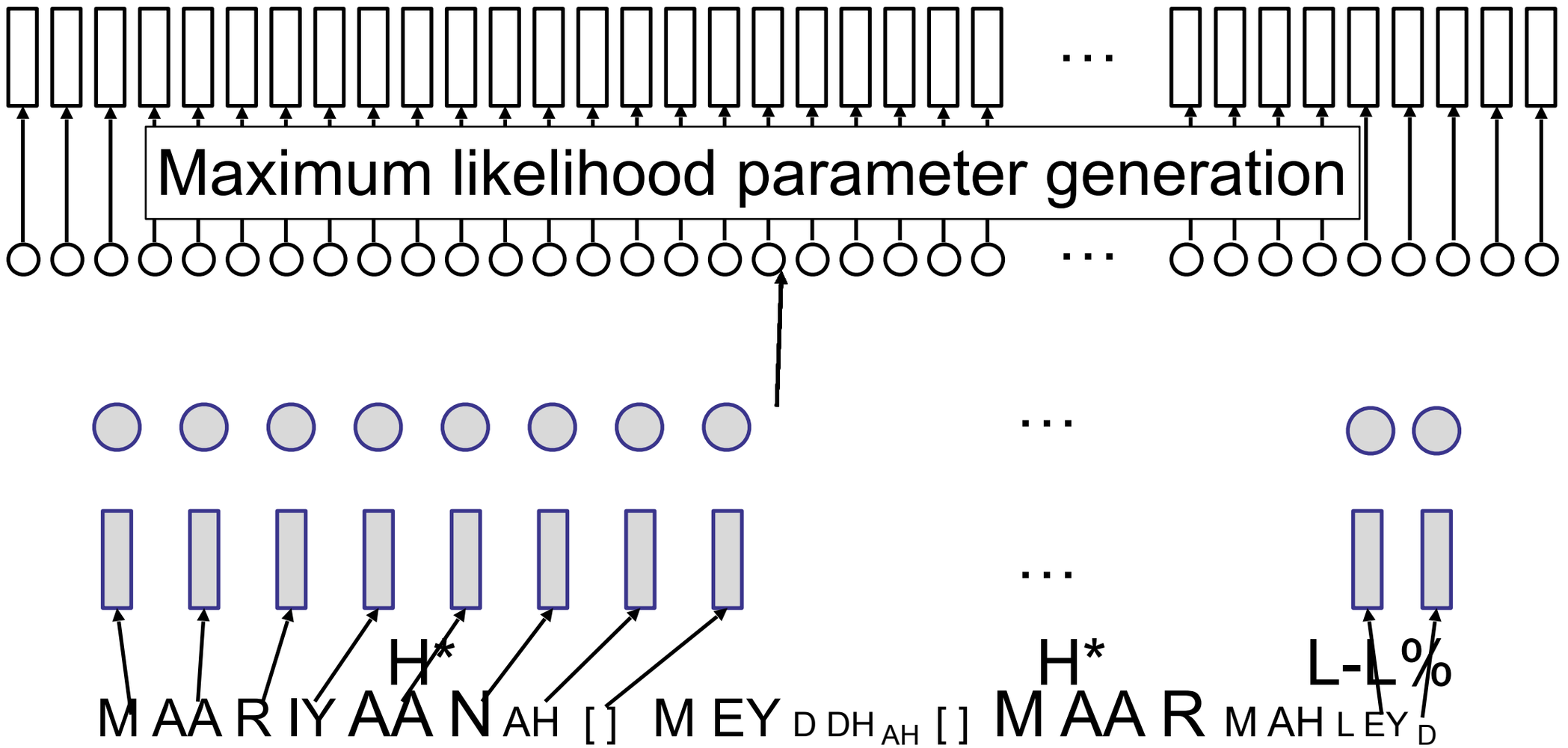




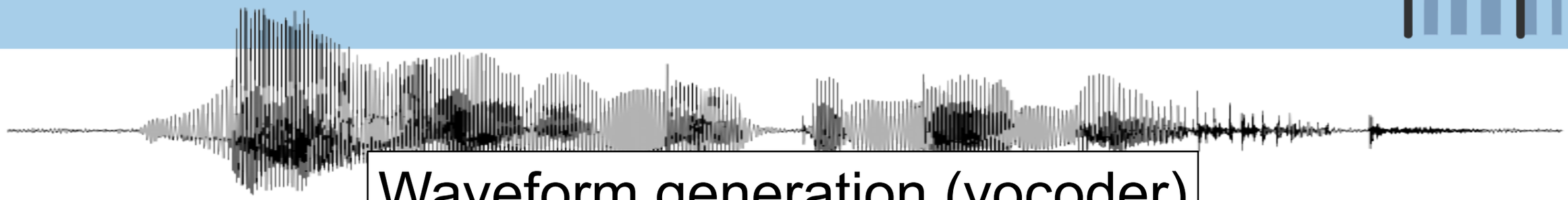
Marianna made the marmalade

OVERVIEW – STATISTICAL SPEECH SYNTHESIS SYSTEM





OVERVIEW – STATISTICAL & DNN



Waveform generation (vocoder)

Aligned!

Deep neural networks (Feedforward, RNN …)

Explicit duration prediction (alignment for generation)

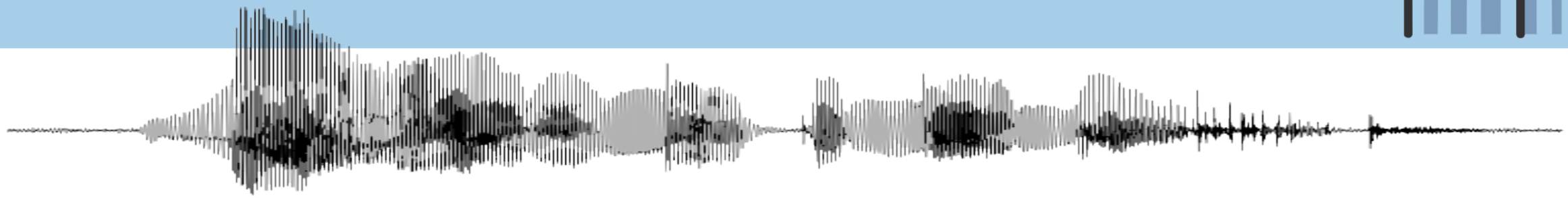
Linguistic feature extraction

M AA R IY A … R M AH L EY D

L-L%

Marianna made the marmalade

OVERVIEW – STATISTICAL & DNN



Aligned!

RNNs are usually restricted to problems where the alignment between the input and output sequence is known in advance (Graves 2012)

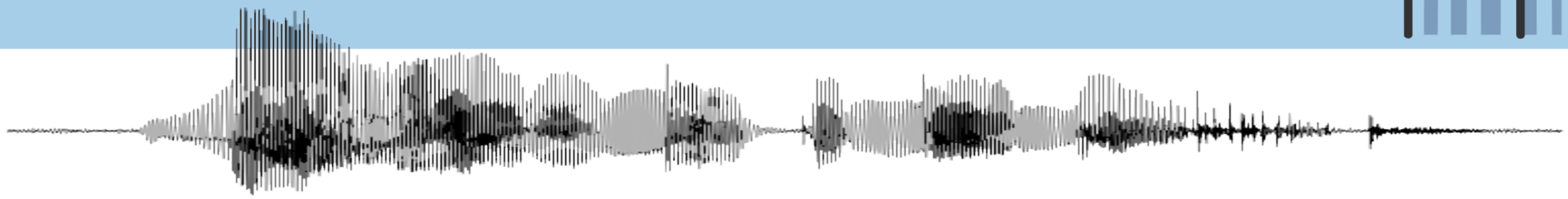
HMM is needed for alignment learning

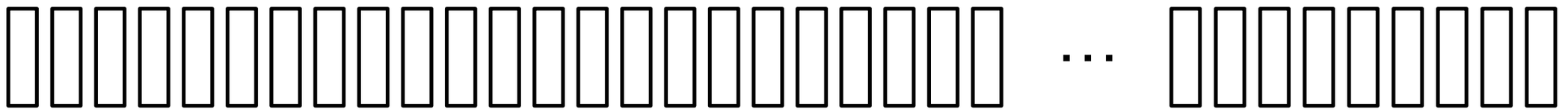
H* H* L-L%

M AA R IY AA N AH [ ] M EY D DH AH [ ] M AA R M AH L EY D

Marianna made the marmalade

OVERVIEW – SEQ-TO-SEQ MODELS

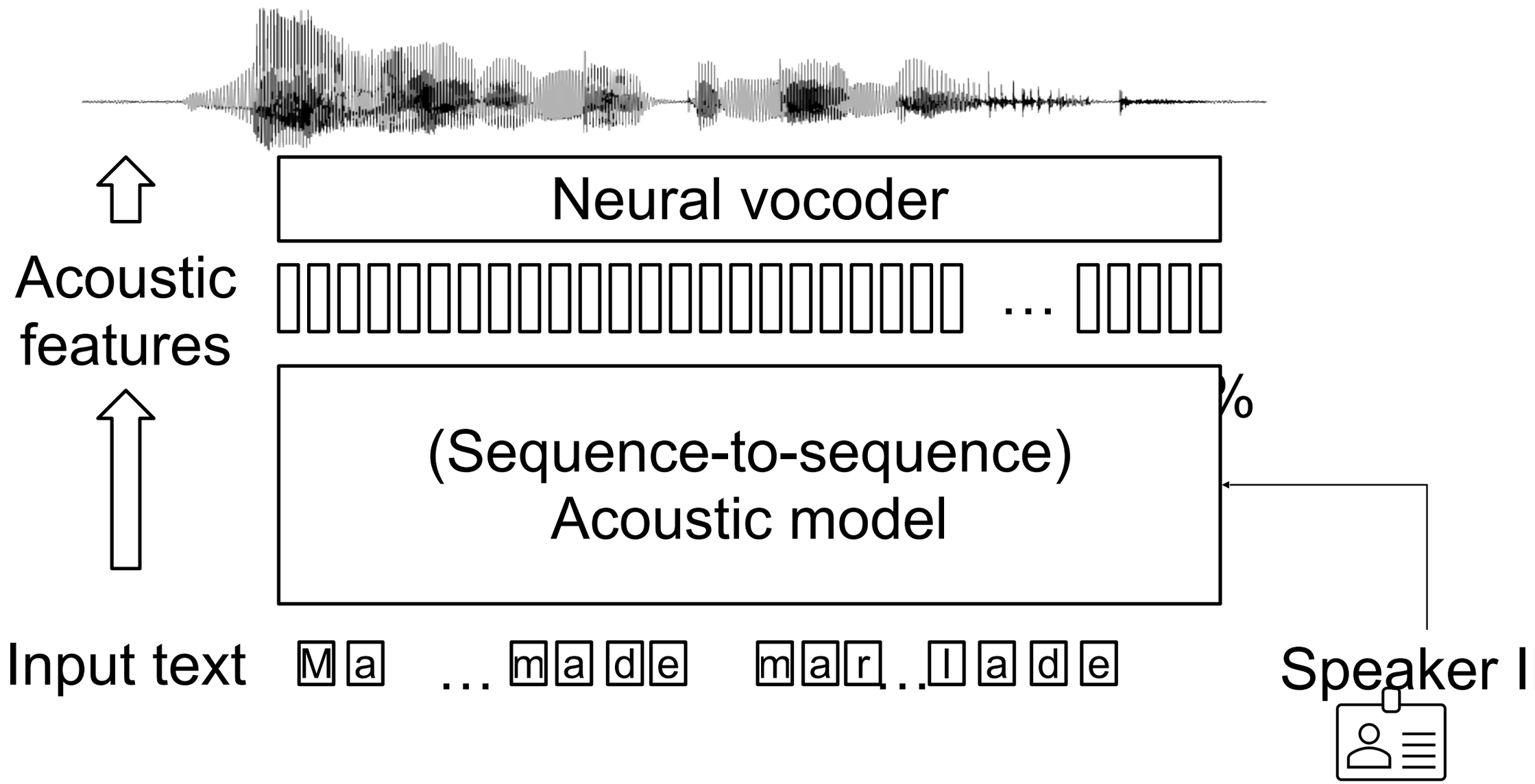




Single neural network for
Joint alignment & regression & linguistic analysis

Marianna made the marmalade

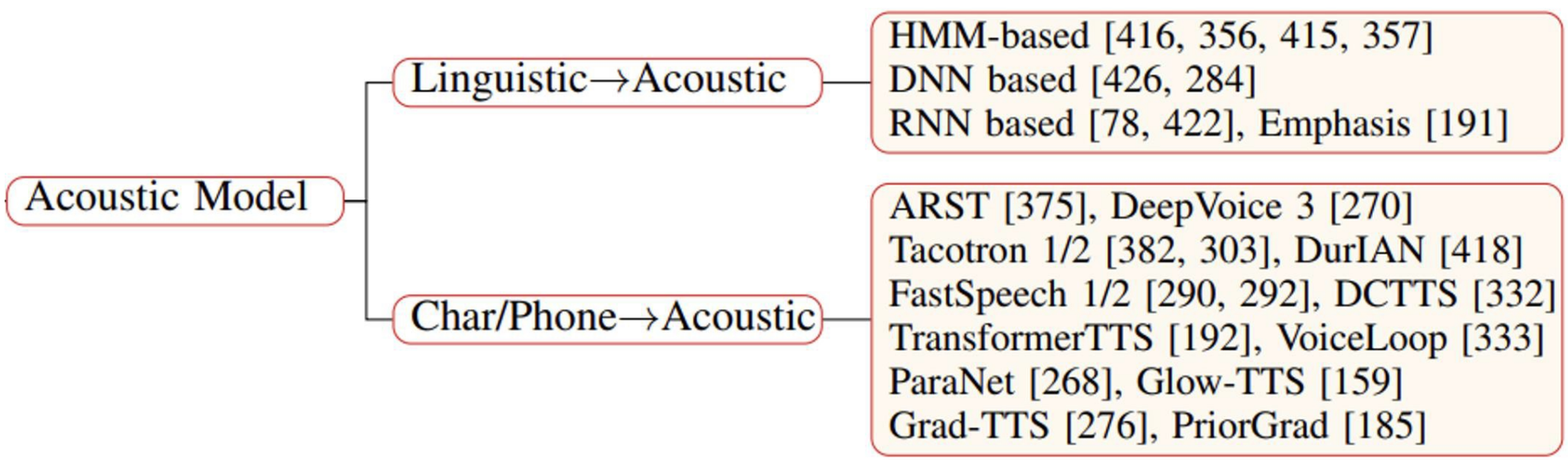
Text-to-speech synthesis – Recent methods



Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. A Survey on Neural Speech Synthesis. *ArXiv Preprint ArXiv:2106.15561*. 2021.
Xin Wang, Neural statistical parametric speech synthesis, ISCA Odyssey 2020, tutorial: https://tonywangx.github.io/slide.html#dec-2020

Acoustic model

- Predict acoustic features from linguistic features

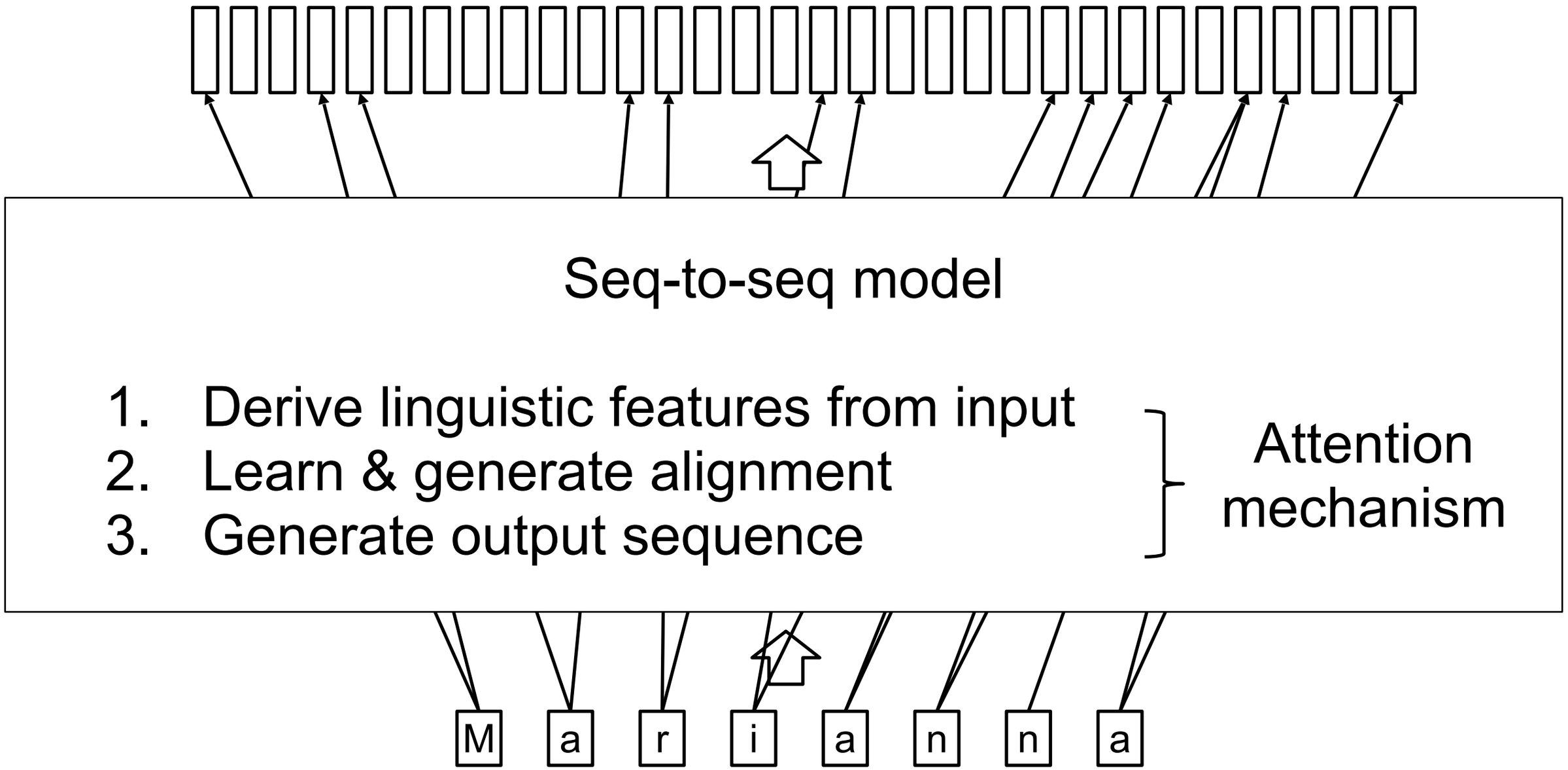


Recent Seq-to-seq TTS

How do they work?

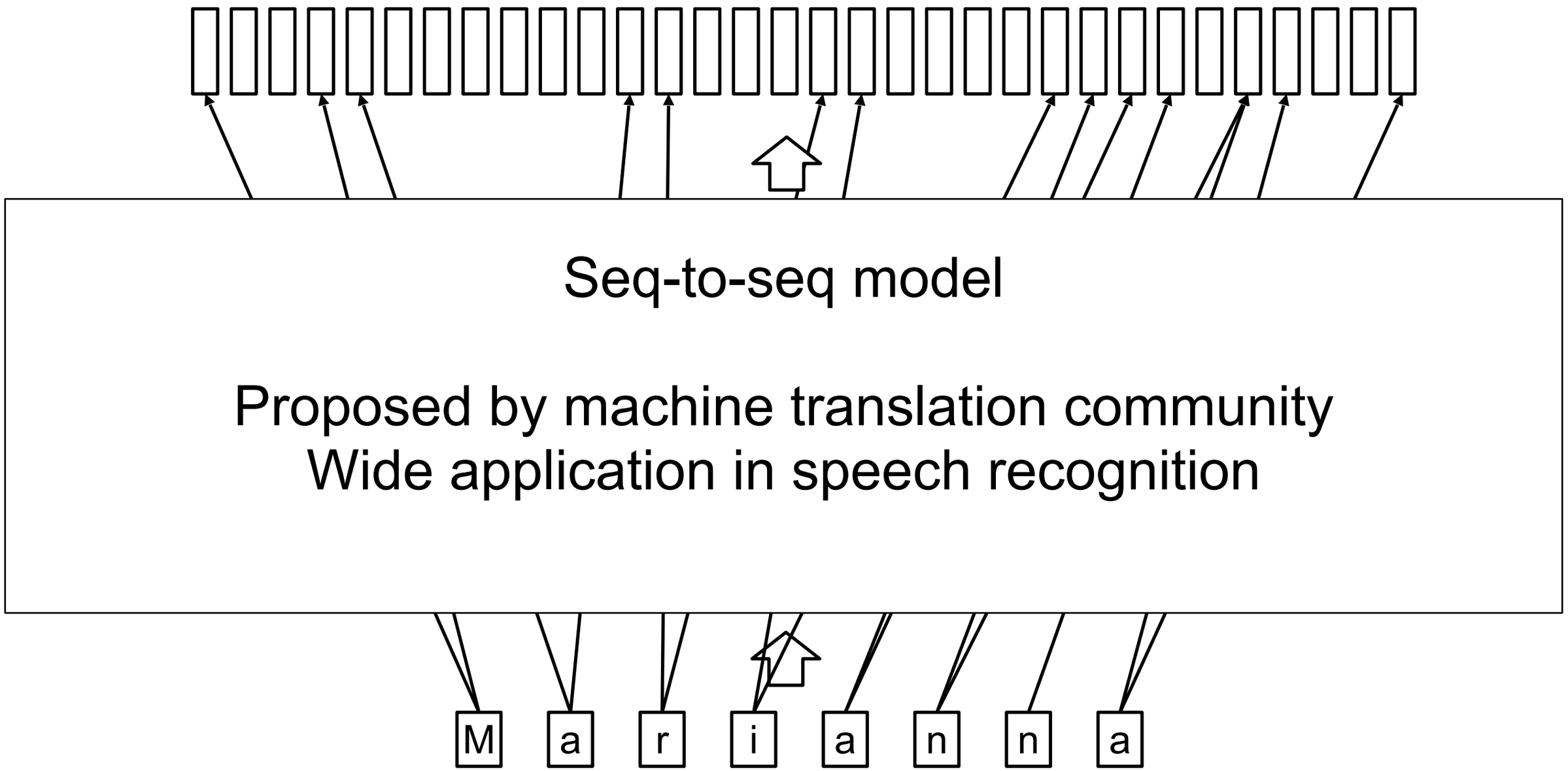
SEQ-TO-SEQ MODEL

- **Task**



SEQ-TO-SEQ MODEL

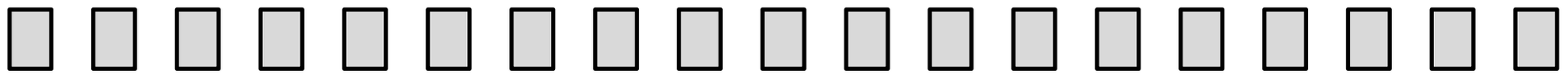
- Task



SEQ-TO-SEQ MODEL

- **Task**

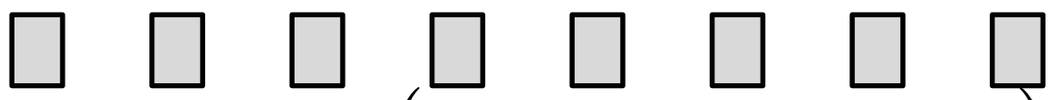
$$\boldsymbol{y}_1\, \boldsymbol{y}_2 \qquad\qquad \boldsymbol{y}_N$$



$$\boldsymbol{y}_{1:N} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N)$$

$$p(\boldsymbol{y}_{1:N} | \boldsymbol{x}_{1:M})$$

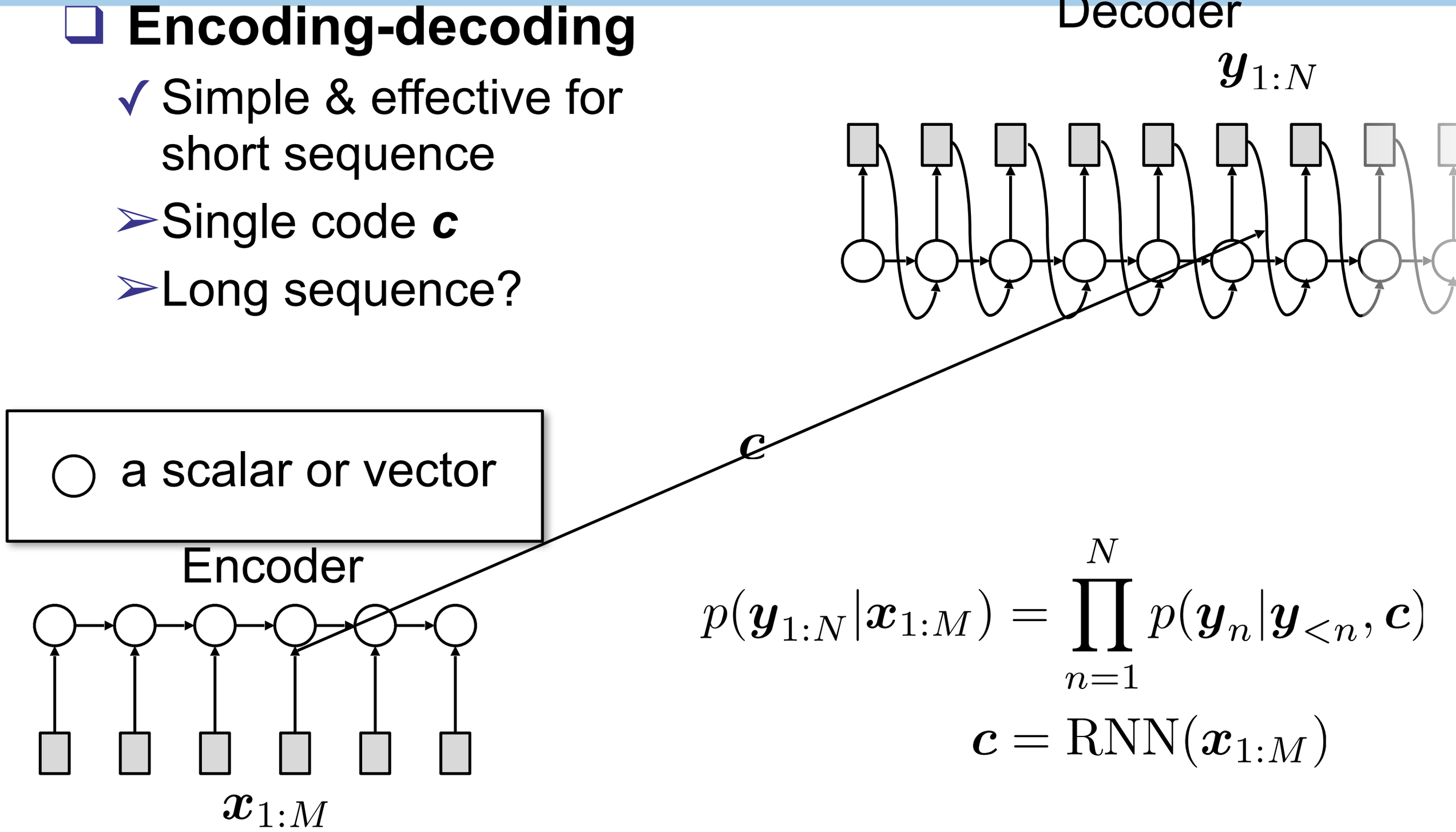
$$\boldsymbol{x}_1 \qquad\qquad \boldsymbol{x}_M$$



$$\boldsymbol{x}_{1:M} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_M)$$

SEQ-TO-SEQ MODEL

- **Encoding-decoding**
  - ✓ Simple & effective for short sequence
  - ➢ Single code ***c***
  - ➢ Long sequence?



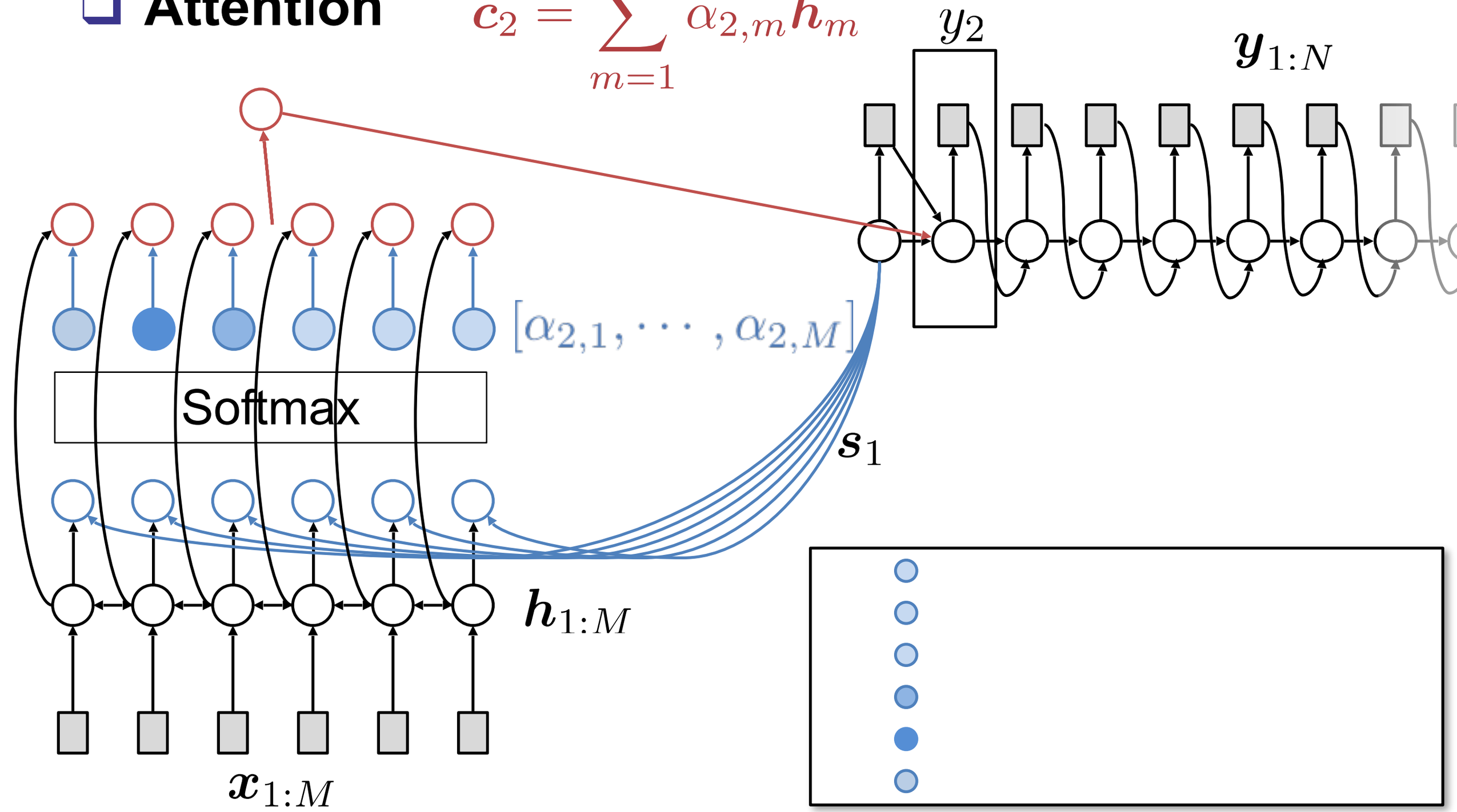
$$p(\boldsymbol{y}_{1:N}|\boldsymbol{x}_{1:M}) = \prod_{n=1}^{N} p(\boldsymbol{y}_n|\boldsymbol{y}_{<n}, \boldsymbol{c})$$

$$\boldsymbol{c} = \mathrm{RNN}(\boldsymbol{x}_{1:M})$$

SEQ-TO-SEQ MODEL

- **Attention**

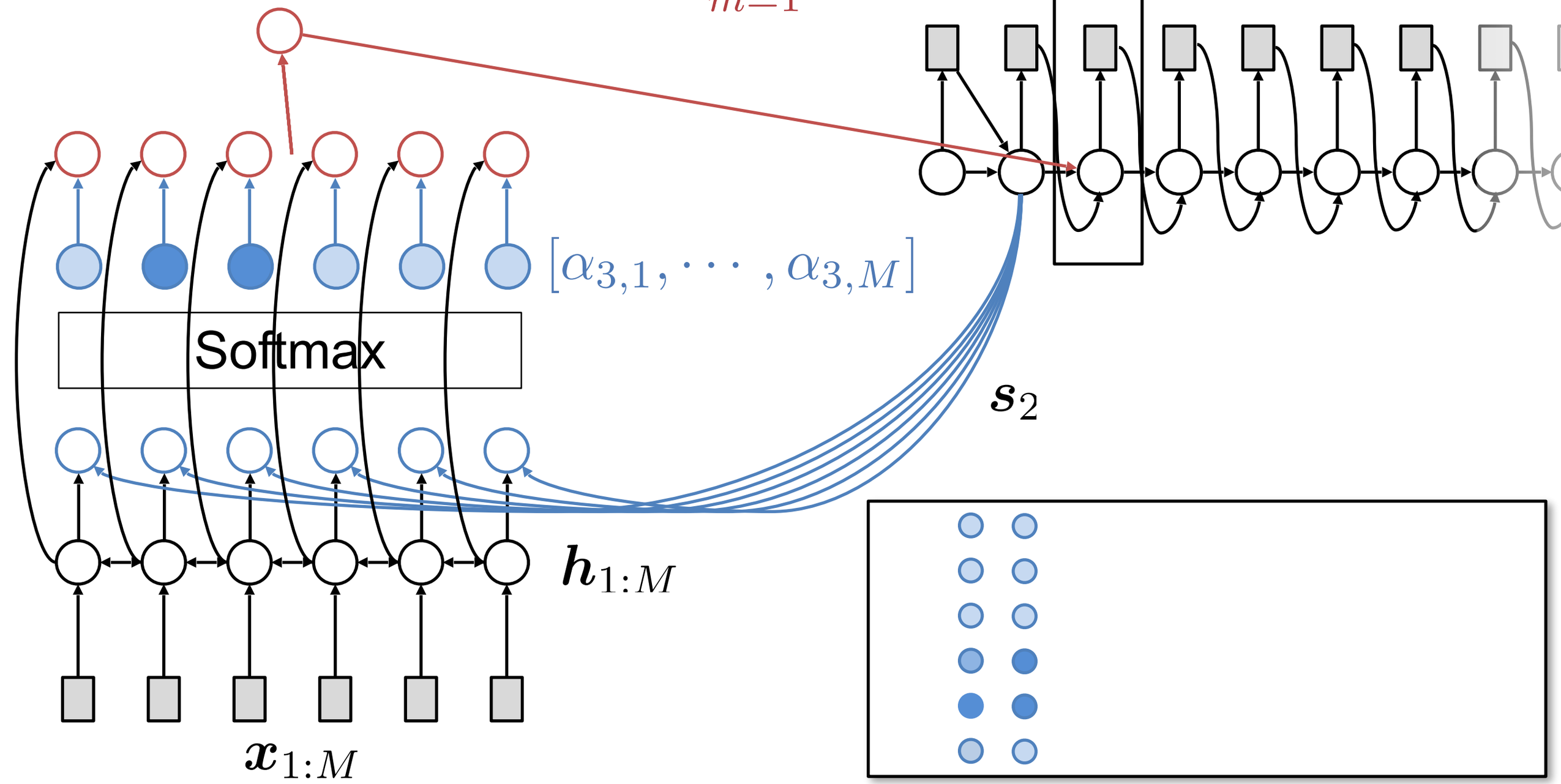
$$\boldsymbol{c}_2 = \sum_{m=1}^{M} \alpha_{2,m} \boldsymbol{h}_m$$



SEQ-TO-SEQ MODEL

- **Attention**

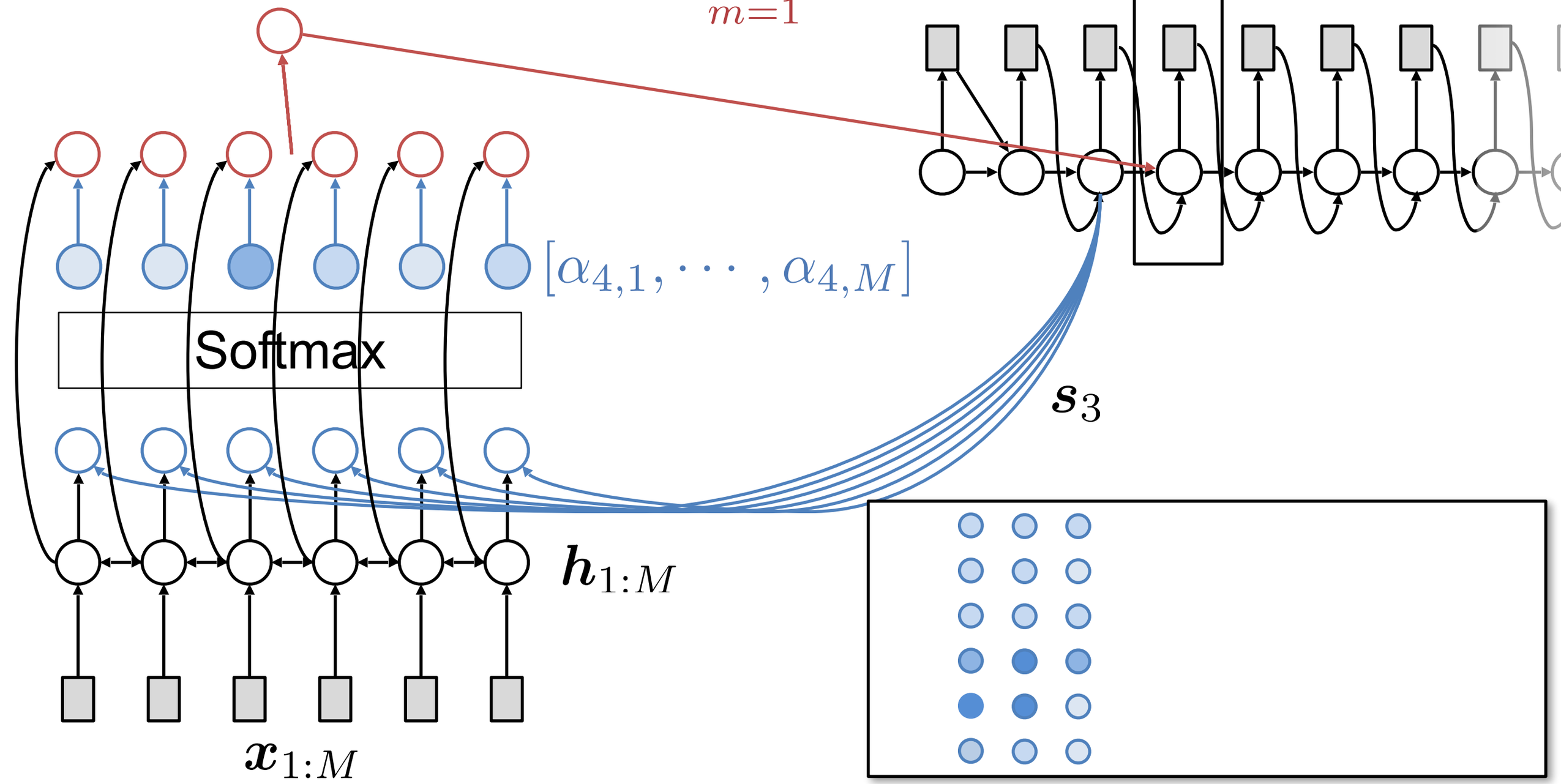
$$\boldsymbol{c}_3 = \sum_{m=1}^{M} \alpha_{3,m} \boldsymbol{h}_m$$



SEQ-TO-SEQ MODEL

- **Attention**

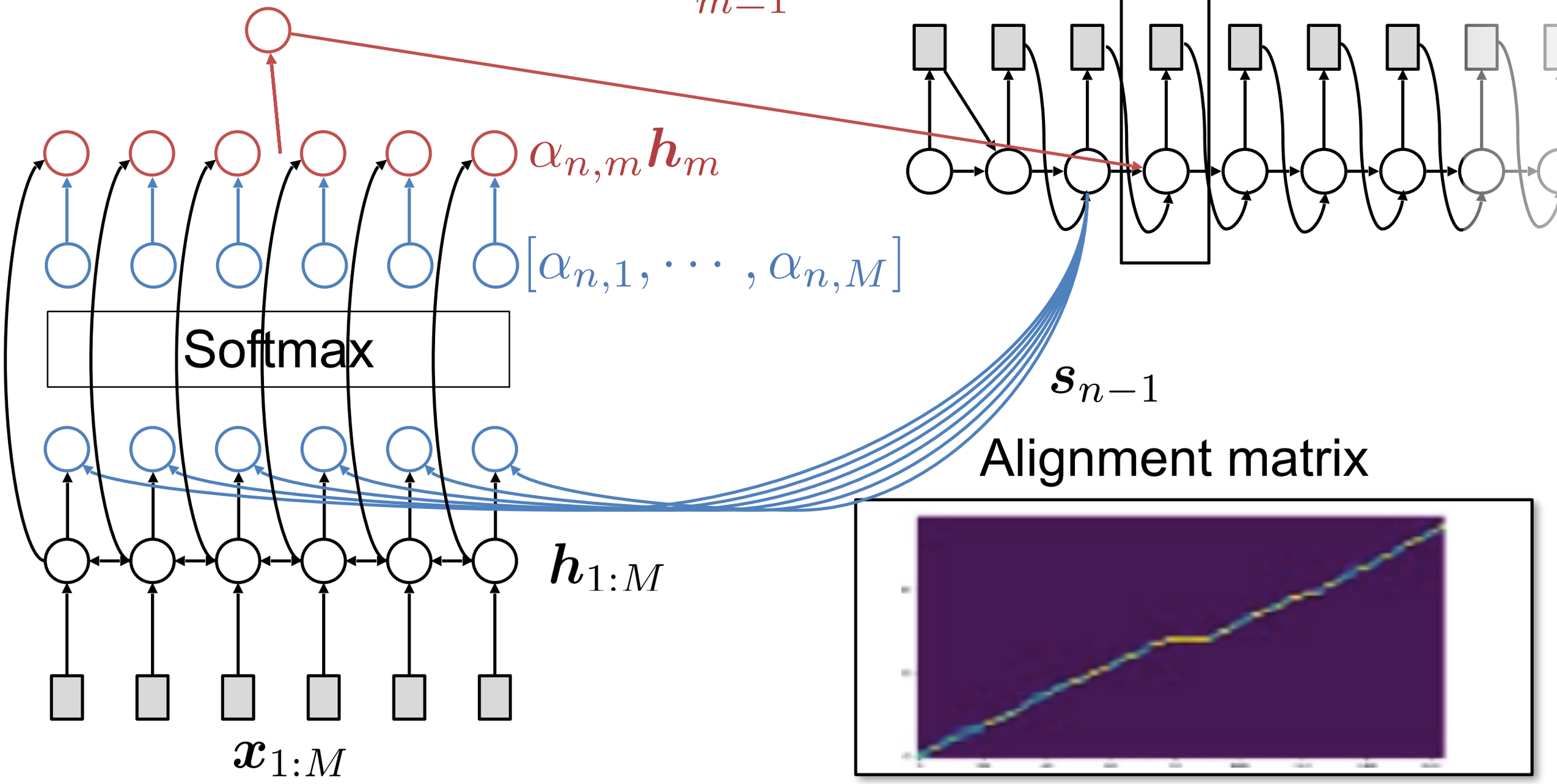
$$\boldsymbol{c}_4 = \sum_{m=1}^{M} \alpha_{4,m} \boldsymbol{h}_m$$



SEQ-TO-SEQ MODEL

- **Attention**

$$\boldsymbol{c}_n = \sum_{m=1}^{M} \alpha_{n,m} \boldsymbol{h}_m$$



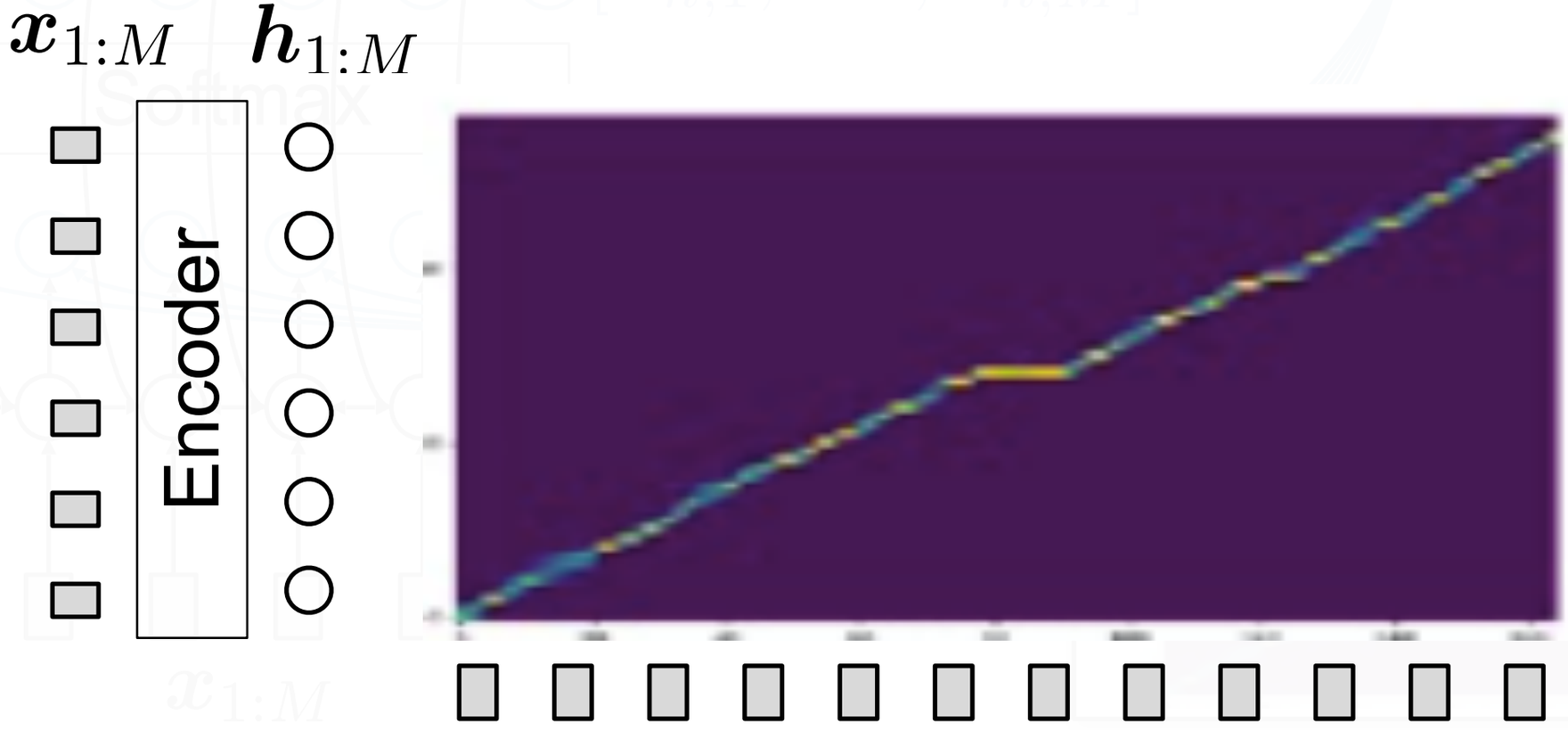
Alignment matrix

SEQ-TO-SEQ MODEL

- **Attention**

$$p(\boldsymbol{y}_{1:N}|\boldsymbol{x}_{1:M}) = \prod_{n=1}^{N} p(\boldsymbol{y}_n|\boldsymbol{y}_{<n}, \boldsymbol{c}_n)$$

$$\boldsymbol{c}_n = \sum_{m=1}^{M} \alpha_{n,m}\boldsymbol{h}_m = \sum_{m=1}^{M} \alpha_{n,m}\text{Encoder}(\boldsymbol{x}_{1:M})_m$$

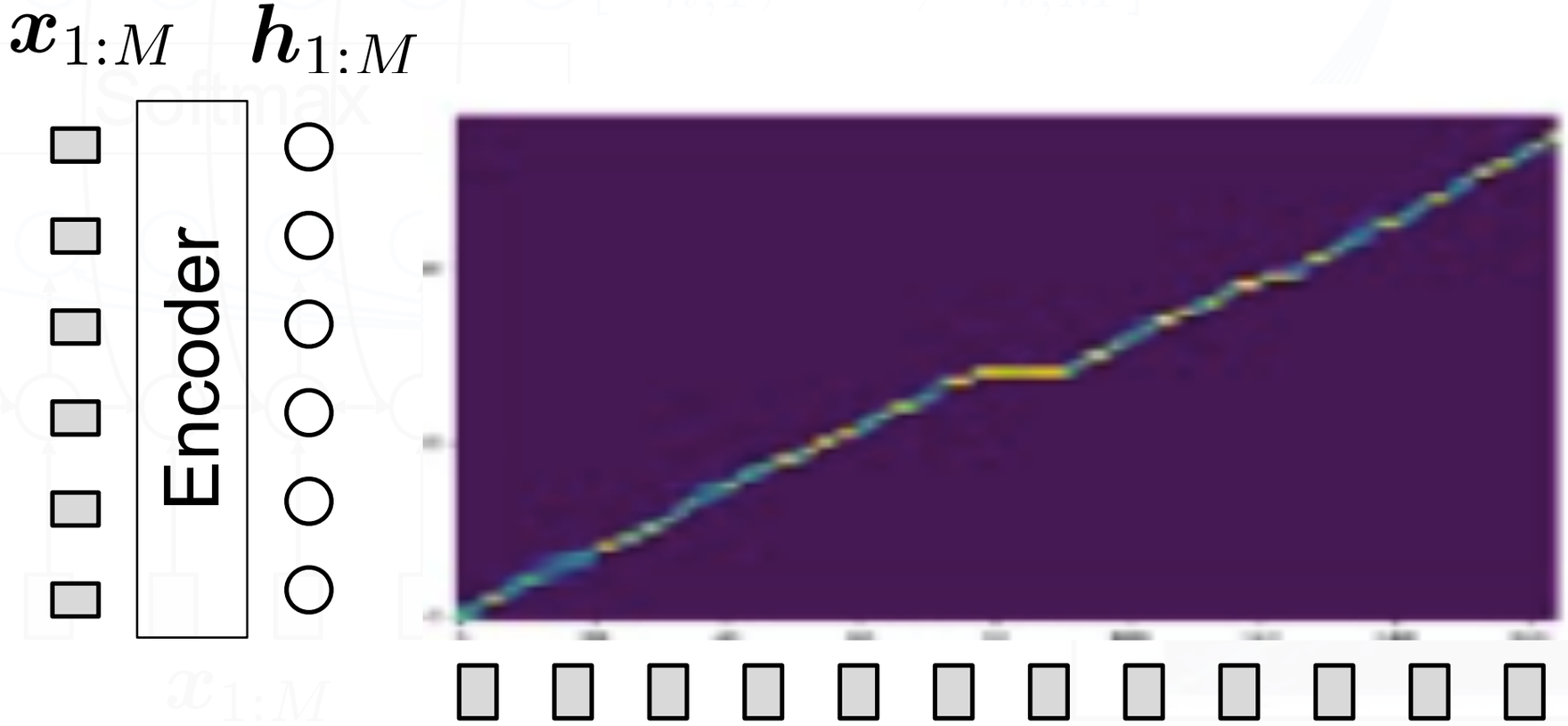


Seq-to-seq model Failure Cases

- **Attention**

$$p(\boldsymbol{y}_{1:N}|\boldsymbol{x}_{1:M}) = \prod_{n=1}^{N} p(\boldsymbol{y}_n|\boldsymbol{y}_{<n}, \boldsymbol{c}_n)$$

$$\boldsymbol{c}_n = \sum_{m=1}^{M} \alpha_{n,m}\boldsymbol{h}_m = \sum_{m=1}^{M} \alpha_{n,m}\text{Encoder}(\boldsymbol{x}_{1:M})_m$$



TACOTRON

- **TTS systems**

$y_{1:N}$

Acoustic feature vectors

Tacotron

Griffin-Lim reconstruction

Linear-scale spectrogram

CBHG

Seq2seq target with r=3

Decoder RNN

Attention RNN

Pre-net

<GO> frame

Attention

Attention is applied to all decoder steps

CBHG

Pre-net

Character embeddings

Decoder (autogressive)

Attention

Encoder

$x_{1:M}$

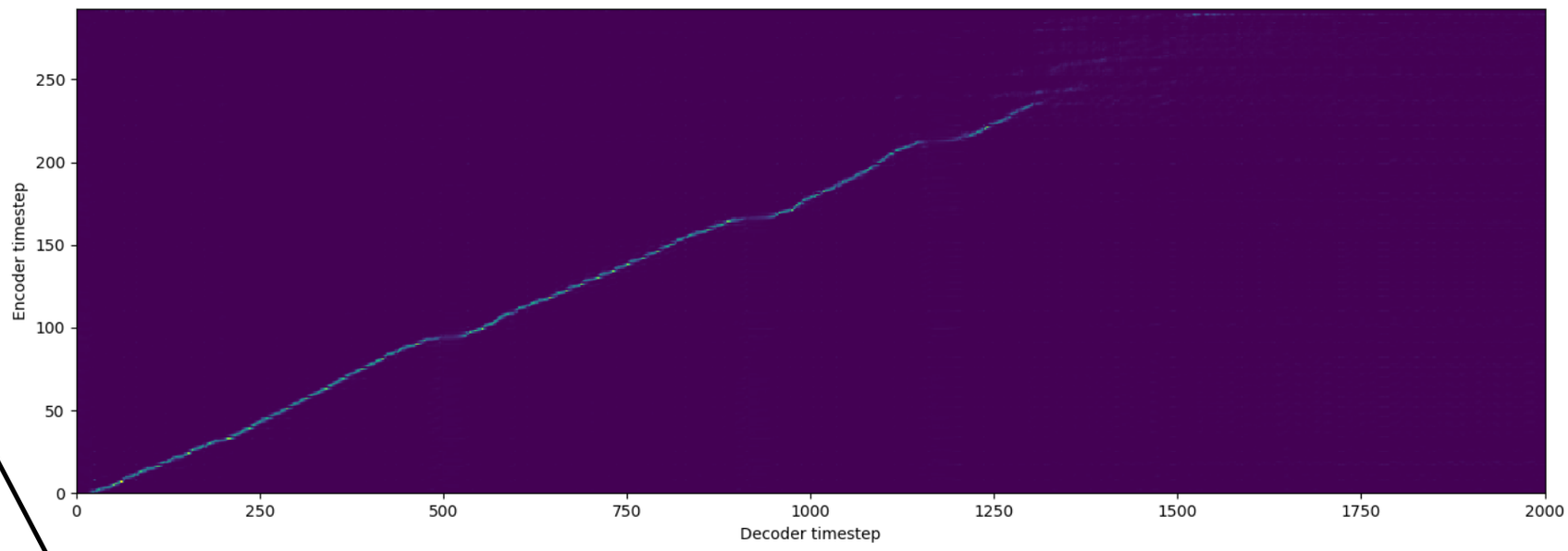
Phoneme / characters

Wang, Y. *et al.* Tacotron: Towards End-to-End Speech Synthesis. in *Proc. Interspeech* 4006–4010 (2017).

A Failure Case of Soft Alignment

Samples & audios from (He 2019)

$\boldsymbol{x}_{1:M}$



$\mathbb{R}^{M \times N}$

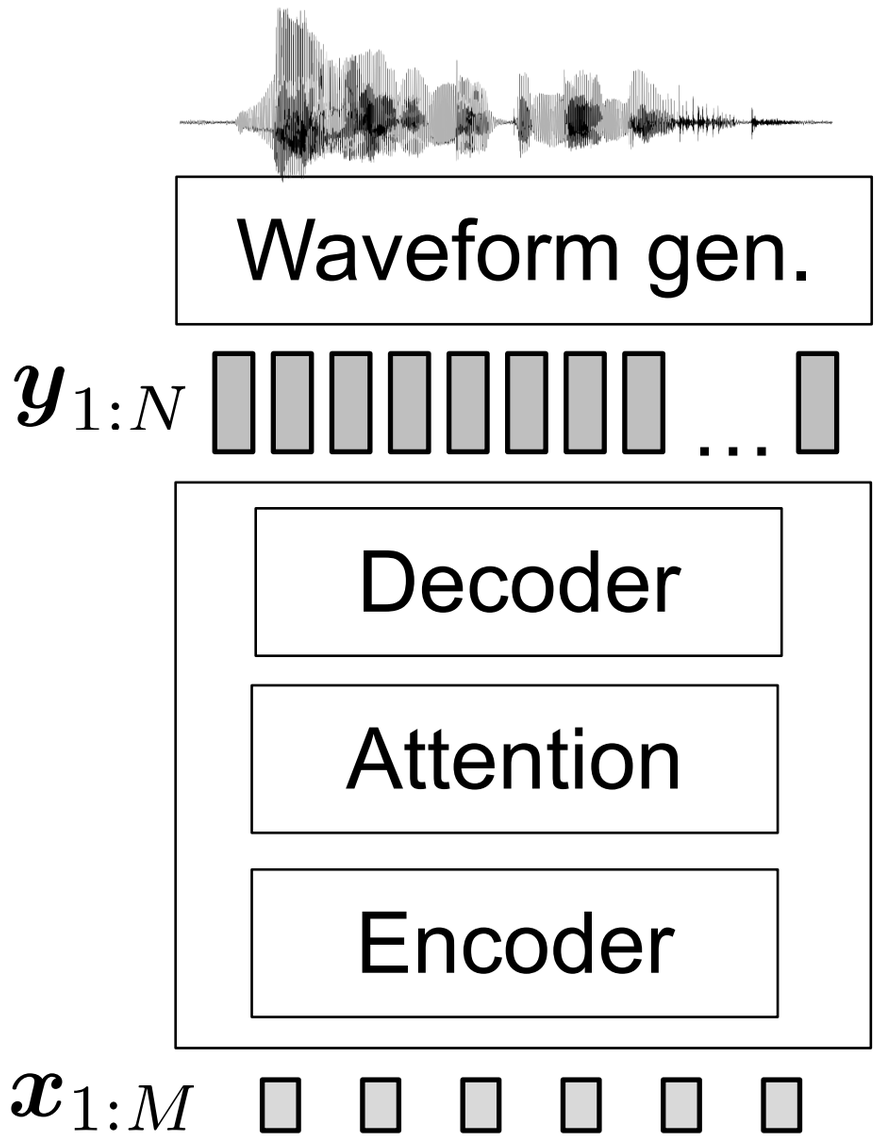
$\boldsymbol{y}_{1:N}$

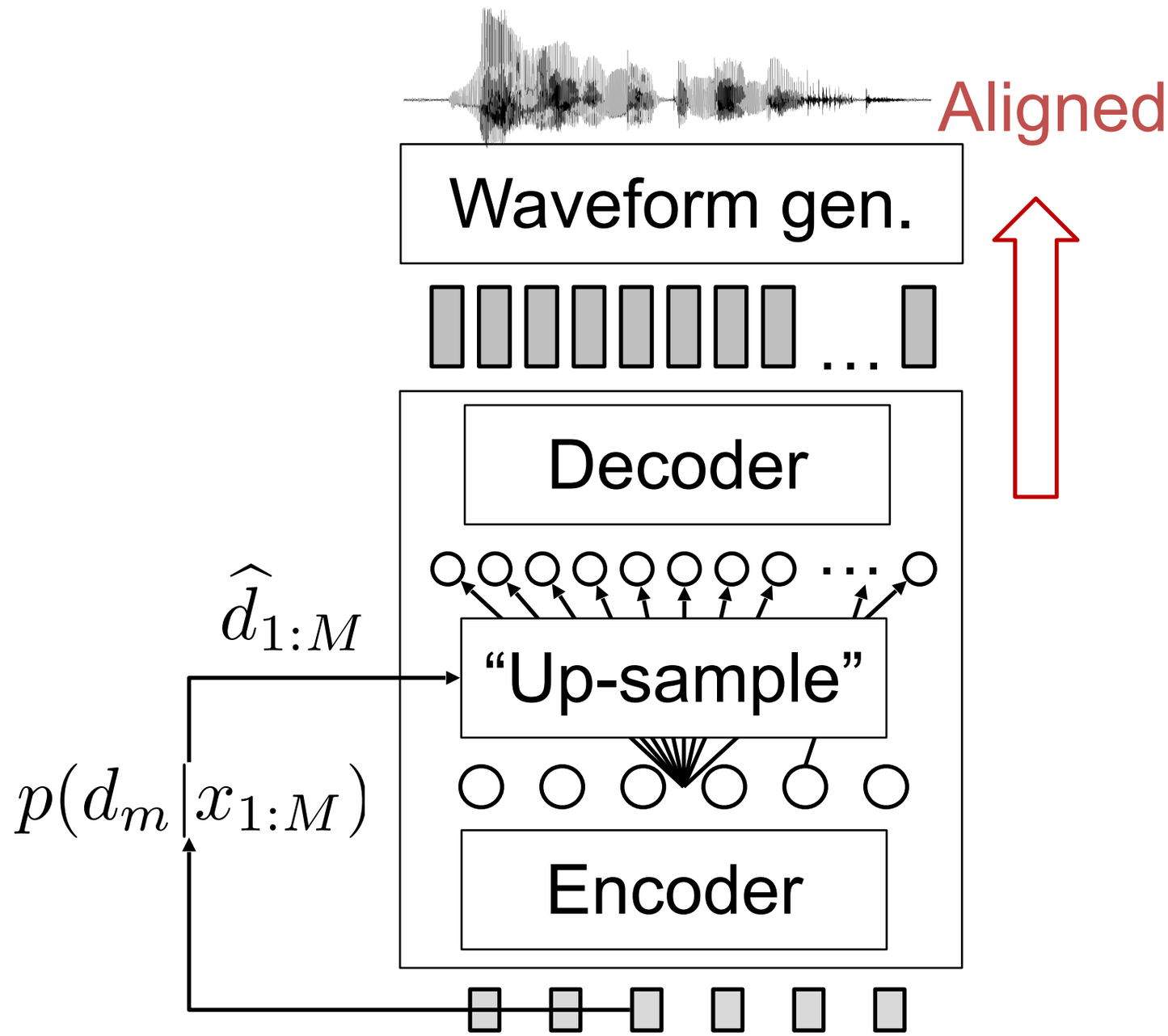
Crashes \ ReadAVs \ 00000000 \ foo , doc WriteAVs \ 11111111 \ foo 2 , doc:
That makes post-processing a little painful since the fuzzer reports crashes in the hierarchical structure mentioned above.

He, M., Deng, Y. & He, L. Robust Sequence-to-Sequence Acoustic Modeling with Stepwise Monotonic Attention for Neural TTS. in *Proc. Interspeech* 1293–1297 (2019). doi:10.21437/Interspeech.2019-1972
https://mutiann.github.io/papers/interspeech2019/

HYBRID APPROACHES (HARD ALIGNMENT)

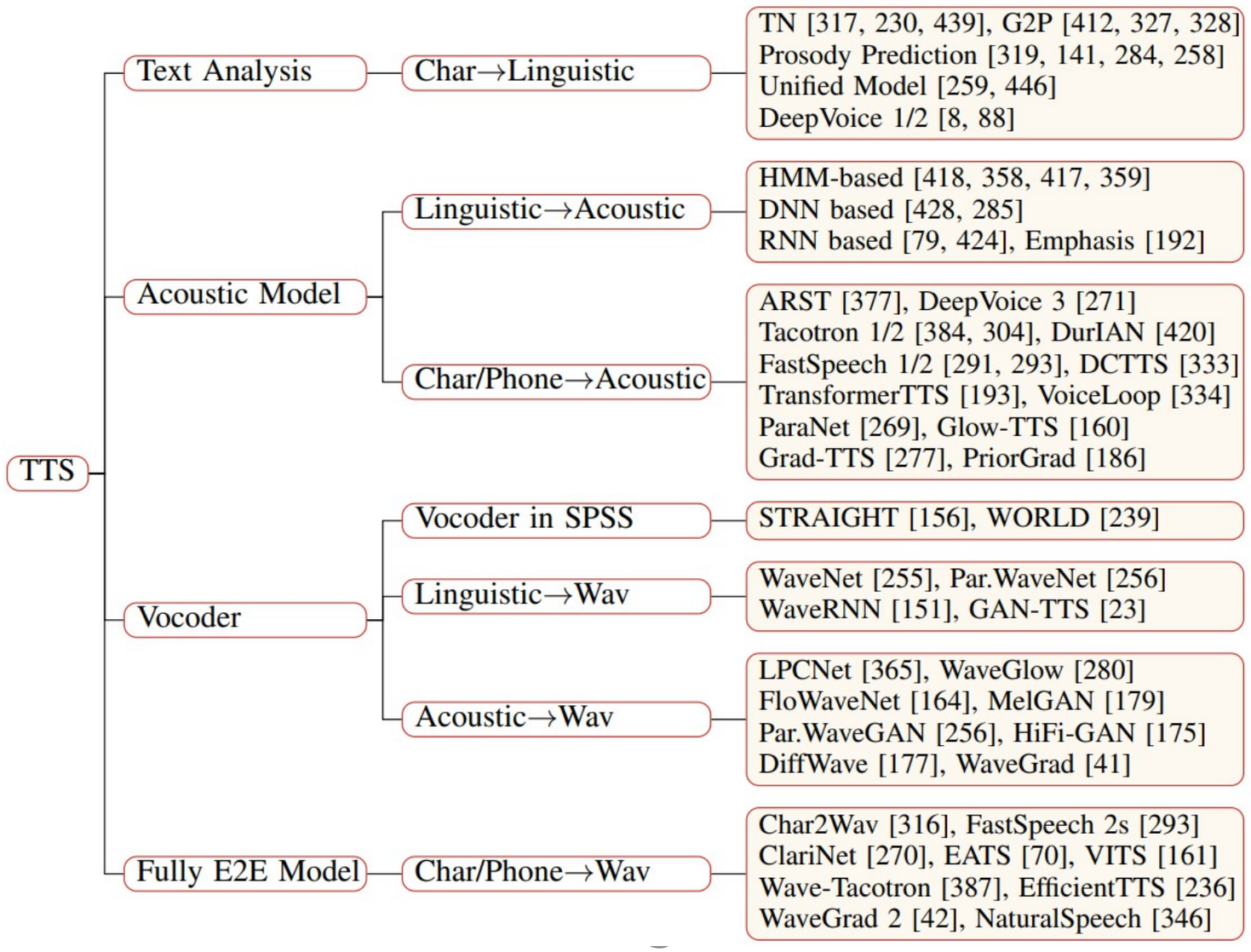
- **Generation: similar to DNN-HMM approach**





Ren, Y. *et al.* Fastspeech: Fast, robust and controllable text to speech. in *Proc. NIPS* 3171–3180 (2019).
Yu, C. *et al.* Durian: Duration informed attention network for multimodal synthesis. *arXiv Prepr* (2019)
Zeng, Z., Wang, J., Cheng, N., Xia, T. & Xiao, J. AlignTTS: Efficient feed-forward text-to-speech system without explicit alignment. in *Proc. ICASSP* 6714–6718 (2020).

Key components in TTS



Acoustic model

- Acoustic model in SPSS
- Acoustic models in end-to-end TTS
  - RNN-based (e.g., Tacotron series)
  - CNN-based (e.g., DeepVoice series)
  - Transformer-based (e.g., FastSpeech series)
  - Other (e.g., Flow, GAN, VAE, Diffusion)

SPSS

RNN

CNN

Transformer

Flow

VAE

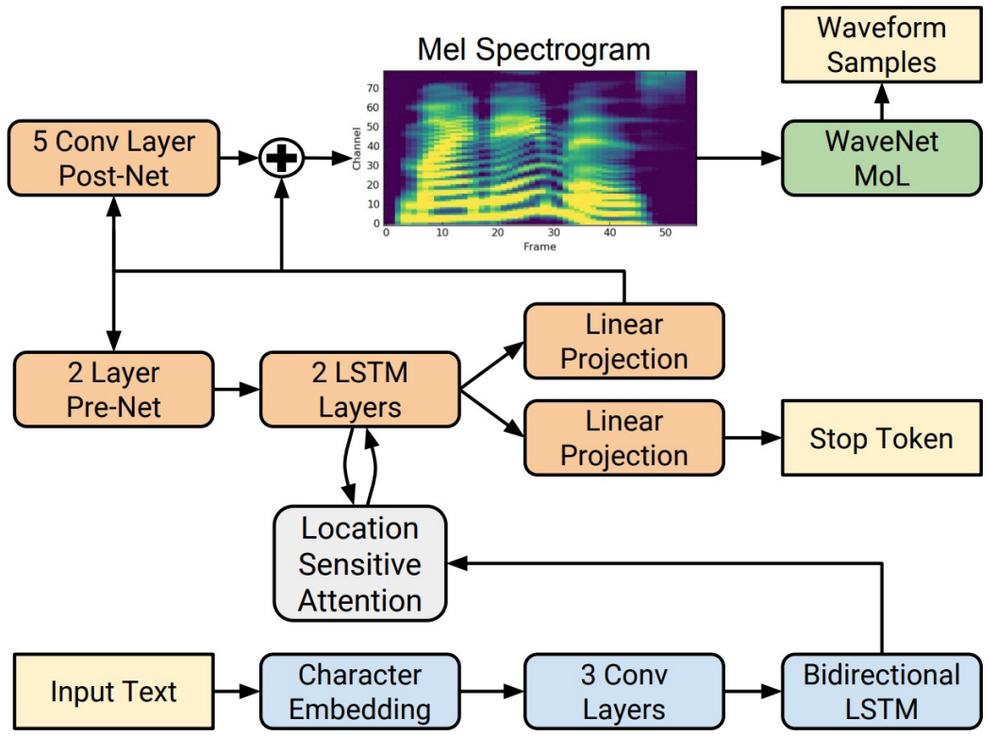
GAN

Diffusion

Acoustic Model	Input→Output	AR/NAR	Modeling	Structure
HMM-based [424, 363]	Ling→MCC+F0	/	/	HMM
DNN-based [434]	Ling→MCC+BAP+F0	NAR	/	DNN
LSTM-based [79]	Ling→LSP+F0	AR	/	RNN
EMPHASIS [195]	Ling→LinS+CAP+F0	AR	/	Hybrid
ARST [382]	Ph→LSP+BAP+F0	AR	Seq2Seq	RNN
VoiceLoop [339]	Ph→MGC+BAP+F0	AR	/	hybrid
Tacotron [389]	Ch→LinS	AR	Seq2Seq	Hybrid/RNN
Tacotron 2 [309]	Ch→MelS	AR	Seq2Seq	RNN
DurIAN [426]	Ph→MelS	AR	Seq2Seq	RNN
Non-Att Tacotron [310]	Ph→MelS	AR	/	Hybrid/CNN/RNN
Para. Tacotron 1/2 [75, 76]	Ph→MelS	NAR	/	Hybrid/Self-Att/CNN
MelNet [374]	Ch→MelS	AR	/	RNN
DeepVoice [8]	Ch/Ph→MelS	AR	/	CNN
DeepVoice 2 [88]	Ch/Ph→MelS	AR	/	CNN
DeepVoice 3 [276]	Ch/Ph→MelS	AR	Seq2Seq	CNN
ParaNet [274]	Ph→MelS	NAR	Seq2Seq	CNN
DCTTS [338]	Ch→MelS	AR	Seq2Seq	CNN
SpeedySpeech [368]	Ph→MelS	NAR	/	CNN
TalkNet 1/2 [19, 18]	Ch→MelS	NAR	/	CNN
TransformerTTS [196]	Ph→MelS	AR	Seq2Seq	Self-Att
MultiSpeech [39]	Ph→MelS	AR	Seq2Seq	Self-Att
FastSpeech 1/2 [296, 298]	Ph→MelS	NAR	Seq2Seq	Self-Att
AlignTTS [437]	Ch/Ph→MelS	NAR	Seq2Seq	Self-Att
JDI-T [201]	Ph→MelS	NAR	Seq2Seq	Self-Att
FastPitch [185]	Ph→MelS	NAR	Seq2Seq	Self-Att
AdaSpeech 1/2/3 [40, 411, 412]	Ph→MelS	NAR	Seq2Seq	Self-Att
AdaSpeech 4 [399]	Ph→MelS	NAR	Seq2Seq	Self-Att
DenoiSpeech [442]	Ph→MelS	NAR	Seq2Seq	Self-Att
DeviceTTS [127]	Ph→MelS	NAR	/	Hybrid/DNN/RNN
LightSpeech [226]	Ph→MelS	NAR	/	Hybrid/Self-Att/CNN
DelightfulTTS [216]	Ph→MelS	NAR	Seq2Seq	Self-Att
Flow-TTS [240]	Ch/Ph→MelS	NAR*	Flow	Hybrid/CNN/RNN
Glow-TTS [162]	Ph→MelS	NAR	Flow	Hybrid/Self-Att/CNN
Flowtron [373]	Ph→MelS	AR	Flow	Hybrid/RNN
EfficientTTS [241]	Ch→MelS	NAR	Flow	Hybrid/CNN
GMVAE-Tacotron [120]	Ph→MelS	AR	VAE	Hybrid/RNN
VAE-TTS [451]	Ph→MelS	AR	VAE	Hybrid/RNN
BVAE-TTS [191]	Ph→MelS	NAR	VAE	CNN
VARA-TTS [208]	Ph→MelS	NAR	VAE	CNN
GAN exposure [100]	Ph→MelS	AR	GAN	Hybrid/RNN
TTS-Stylization [230]	Ch→MelS	AR	GAN	Hybrid/RNN
Multi-SpectroGAN [190]	Ph→MelS	NAR	GAN	Hybrid/Self-Att/CNN
Diff-TTS [142]	Ph→MelS	NAR*	Diffusion	Hybrid/CNN
Grad-TTS [282]	Ph→MelS	NAR	Diffusion	Hybrid/Self-Att/CNN
PriorGrad [189]	Ph→MelS	NAR	Diffusion	Hybrid/Self-Att/CNN
Guided-TTS [161]	Ph→MelS	NAR	Diffusion	Hybrid/Self-Att/CNN
DiffGAN-TTS [215]	Ph→MelS	NAR	Diffusion	Hybrid/Self-Att/CNN

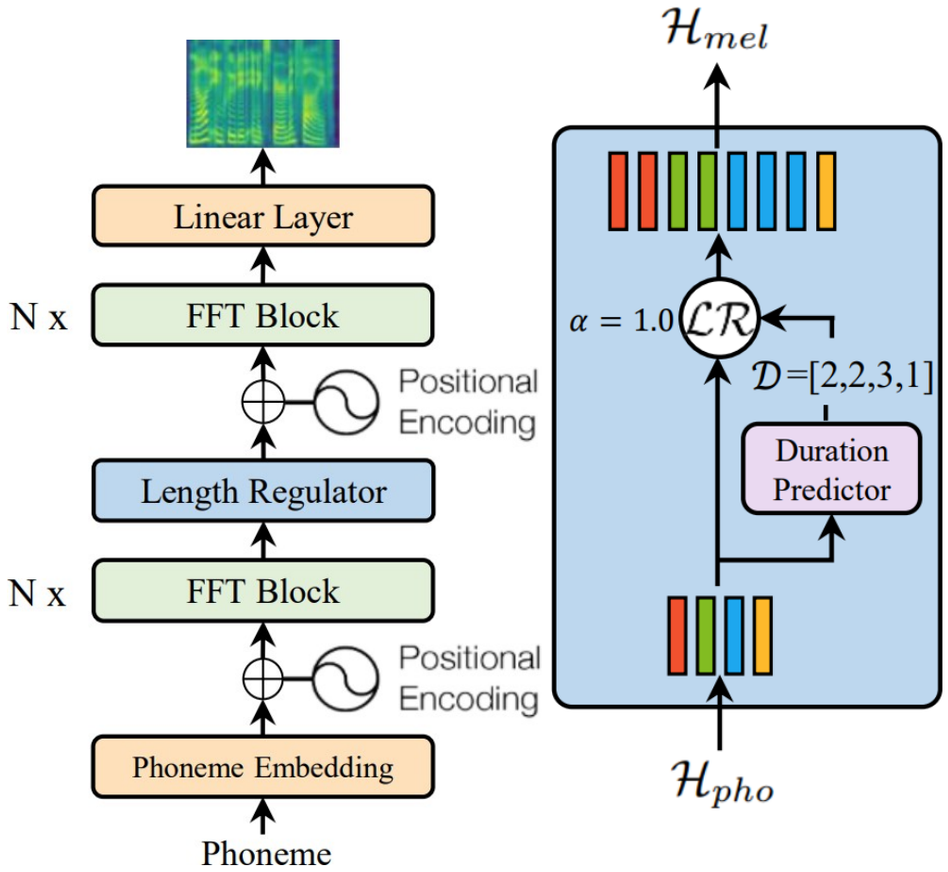
Acoustic model——RNN based

- Tacotron 2
  - Evolved from Tacotron
  - Text to mel-spectrogram generation
  - LSTM based encoder and decoder
  - Location sensitive attention
  - WaveNet as the vocoder



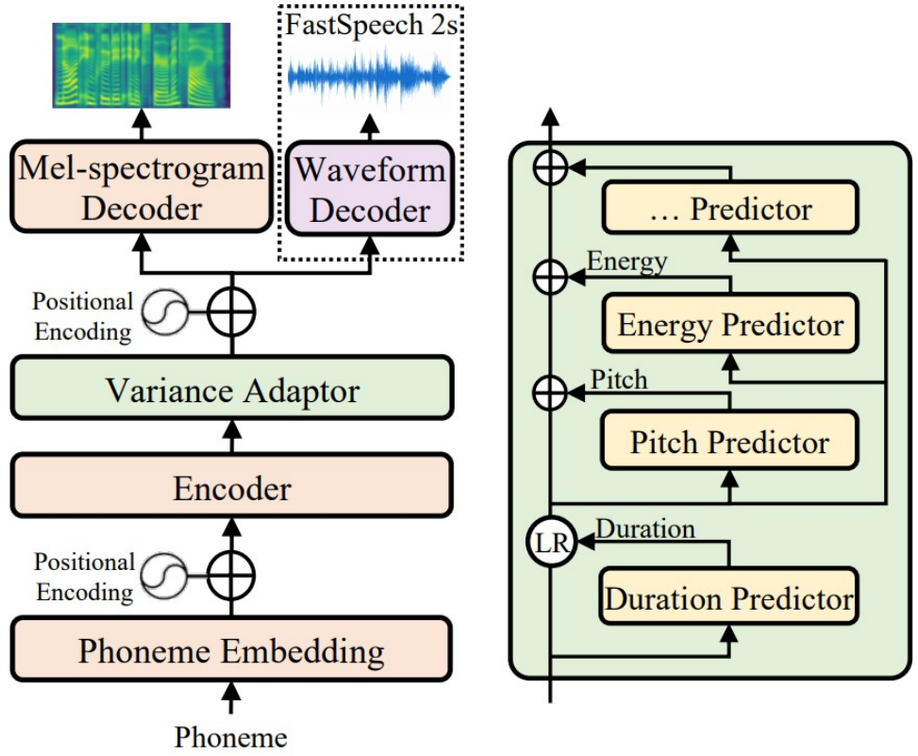
Acoustic model——Transformer based

- FastSpeech [290]
  - Generate mel-spectrogram in parallel (for speedup)
  - Remove the text-speech attention mechanism (for robustness)
- Feed-forward transformer with length regulator (for controllability)



Acoustic model——Transformer based

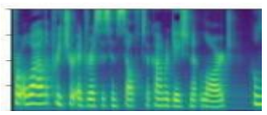
- FastSpeech 2 [292]
  - Improve FastSpeech
  - Use variance adaptor to predict duration, pitch, energy, etc
  - Simplify training pipeline of FastSpeech (KD)
- FastSpeech 2s: a fully end-to-end parallel text to wave model



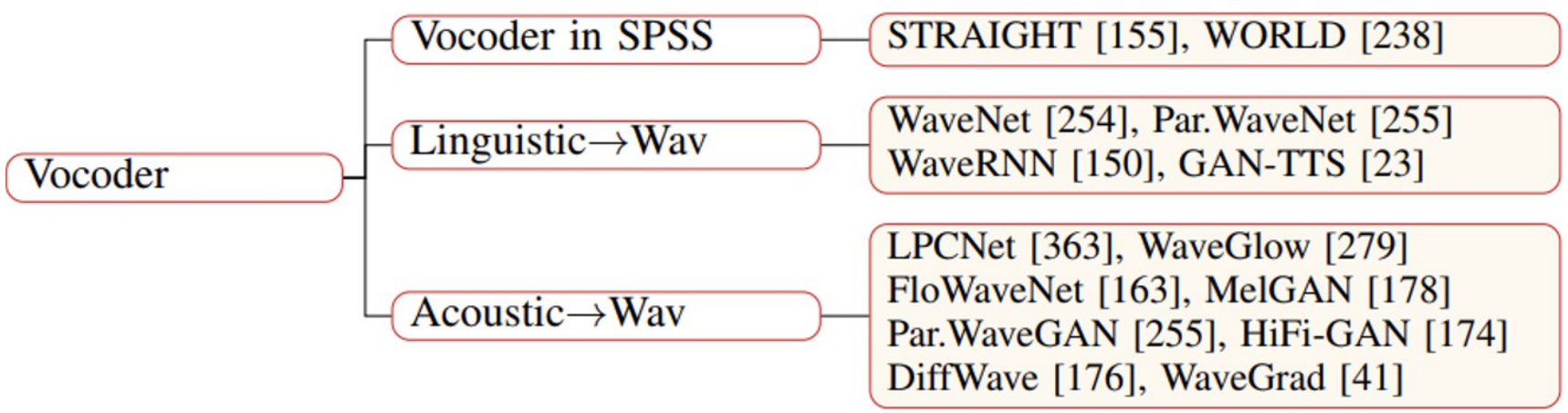
(a) FastSpeech 2 (b) Variance adaptor

- Other works
  - FastPitch [181]
  - JDI-T [197], AlignTTS [429]

Vocoder



Acoustic Features → Vocoder →  Speech



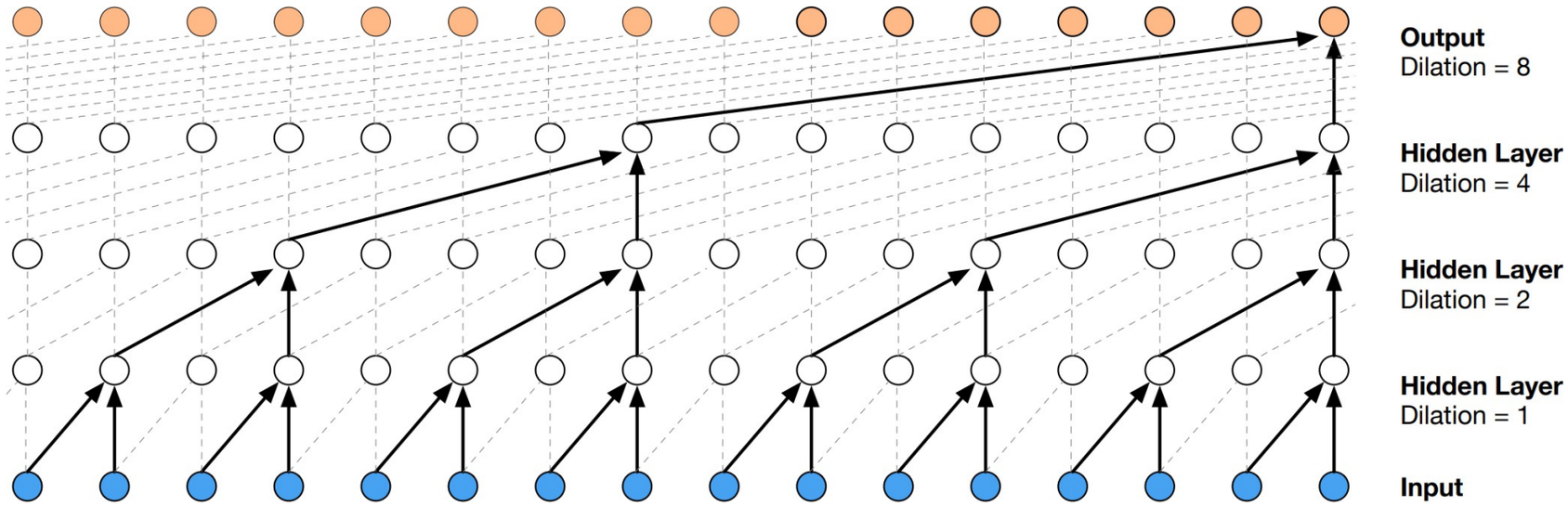
Vocoder

- Autoregressive vocoder
- Flow-based vocoder
- GAN-based vocoder
- VAE-based vocoder
- Diffusion-based vocoder

	Vocoder	Input	AR/NAR	Modeling	Architecture
AR	WaveNet [260]	Linguistic Feature	AR	/	CNN
	SampleRNN [239]	/	AR	/	RNN
	WaveRNN [151]	Linguistic Feature	AR	/	RNN
	LPCNet [370]	BFCC	AR	/	RNN
	Univ. WaveRNN [221]	Mel-Spectrogram	AR	/	RNN
	SC-WaveRNN [271]	Mel-Spectrogram	AR	/	RNN
	MB WaveRNN [426]	Mel-Spectrogram	AR	/	RNN
	FFTNet [146]	Cepstrum	AR	/	CNN
	iSTFTNet [153]	Mel-Spectrogram	NAR	/	CNN
Flow	Par. WaveNet [261]	Linguistic Feature	NAR	Flow	CNN
	WaveGlow [285]	Mel-Spectrogram	NAR	Flow	Hybrid/CNN
	FloWaveNet [166]	Mel-Spectrogram	NAR	Flow	Hybrid/CNN
	WaveFlow [277]	Mel-Spectrogram	AR	Flow	Hybrid/CNN
	SqueezeWave [441]	Mel-Spectrogram	NAR	Flow	CNN
GAN	WaveGAN [69]	/	NAR	GAN	CNN
	GELP [150]	Mel-Spectrogram	NAR	GAN	CNN
	GAN-TTS [23]	Linguistic Feature	NAR	GAN	CNN
	MelGAN [182]	Mel-Spectrogram	NAR	GAN	CNN
	Par. WaveGAN [410]	Mel-Spectrogram	NAR	GAN	CNN
	HiFi-GAN [178]	Mel-Spectrogram	NAR	GAN	Hybrid/CNN
	VocGAN [416]	Mel-Spectrogram	NAR	GAN	CNN
	GED [97]	Linguistic Feature	NAR	GAN	CNN
	Fre-GAN [164]	Mel-Spectrogram	NAR	GAN	CNN
VAE	Wave-VAE [274]	Mel-Spectrogram	NAR	VAE	CNN
Diffusion	WaveGrad [41]	Mel-Spectrogram	NAR	Diffusion	Hybrid/CNN
	DiffWave [180]	Mel-Spectrogram	NAR	Diffusion	Hybrid/CNN
	PriorGrad [189]	Mel-Spectrogram	NAR	Diffusion	Hybrid/CNN
	SpecGrad [176]	Mel-Spectrogram	NAR	Diffusion	Hybrid/CNN

Vocoder—AR

- WaveNet: autoregressive model with dilated causal convolution [van den oord et al 2016]



- Other works
  - WaveRNN
  - LPCNet

Vocoder— HIFiGan

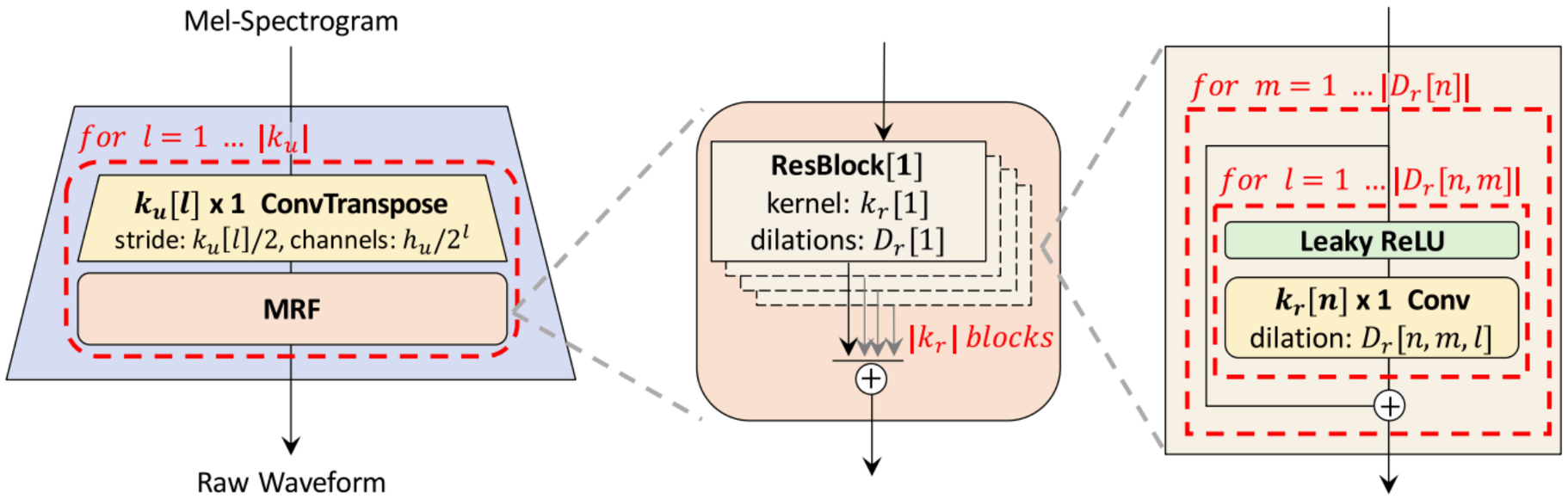


Figure 1: The generator upsamples mel-spectrograms up to $|k_u|$ times to match the temporal resolution of raw waveforms. A MRF module adds features from $|k_r|$ residual blocks of different kernel sizes and dilation rates. Lastly, the n-th residual block with kernel size $k_r[n]$ and dilation rates $D_r[n]$ in a MRF module is depicted.

- GANs: Simultaneously train generative + adversarial nets
- Generator: produce audio from the spectrogram.
  Discriminator: distinguish generator outputs from real audio
- Faster and high quality than Autoregressive

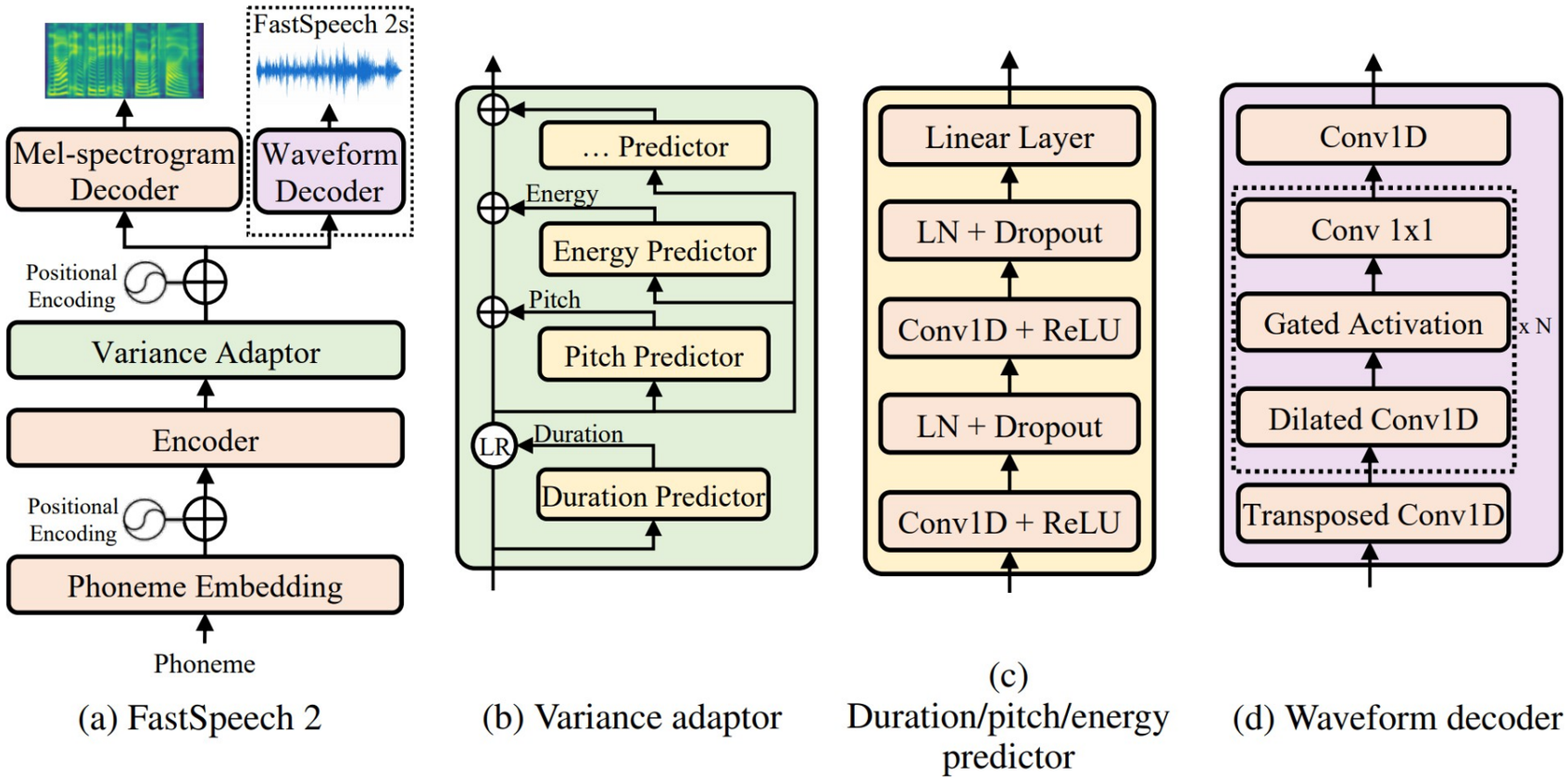
Generative models for acoustic model/vocoder

- Text to speech mapping p(x|y) is multimodal, since one text can correspond to multiple speech variations
  - Acoustic model, phoneme-spectrogram mapping: duration/pitch/energy/ formant
  - Vocoder, spectrogram-waveform mapping: phase

- How to model a multimodal conditional distribution p(x|y)?
  - Autoregressive, GAN, VAE, Flow, Diffusion Model, etc
  - Since L1/L2 can be applied to mel-spectrogram, while cannot be directly applied to waveform
  - Advanced generative models are developed faster in vocoder than in acoustic model, but finally acoustic models catch up ☺

Fully End-to-End TTS

- FastSpeech 2s: fully parallel text to wave model



(a) FastSpeech 2 (b) Variance adaptor (c) Duration/pitch/energy predictor (d) Waveform decoder

Spoken Language Modelling on Speech Tokens

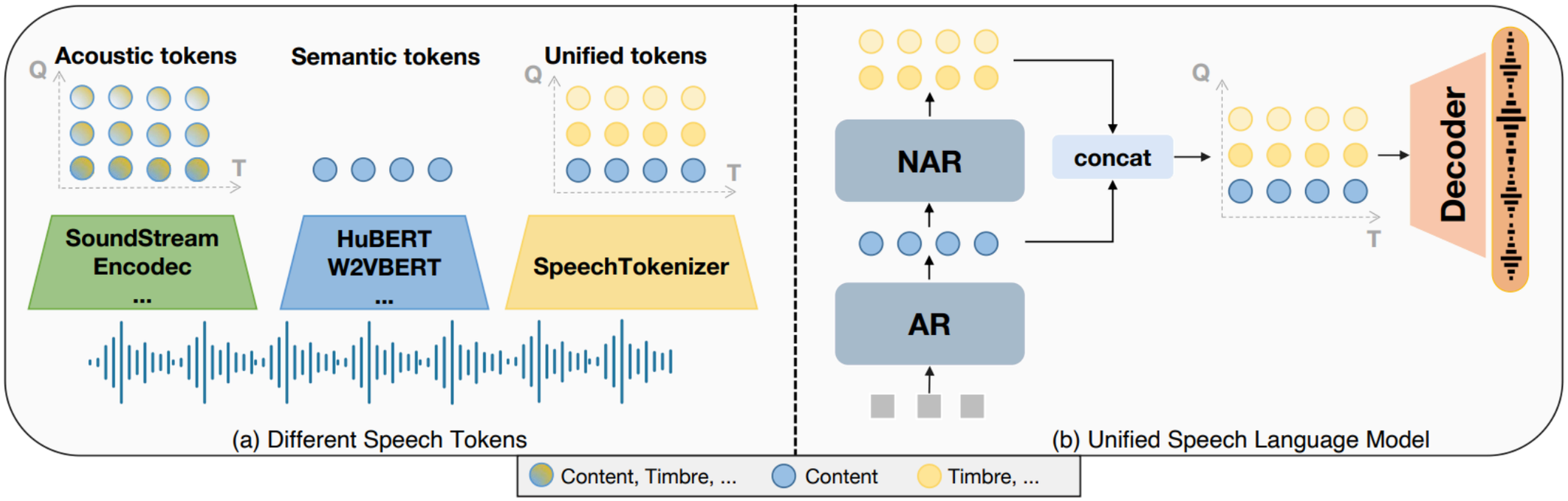


Figure 1: **Left**: Illustration of information composition of different discrete speech representations. **Right**: Illustration of unified speech language models. AR refers to autoregressive and NAR refers to non-autoregressive. Speech tokens are represented as colored circles and different colors represent different information.

SpeechTokenizer, Zhang, et al, 2024

Audio/Speech Tokenization

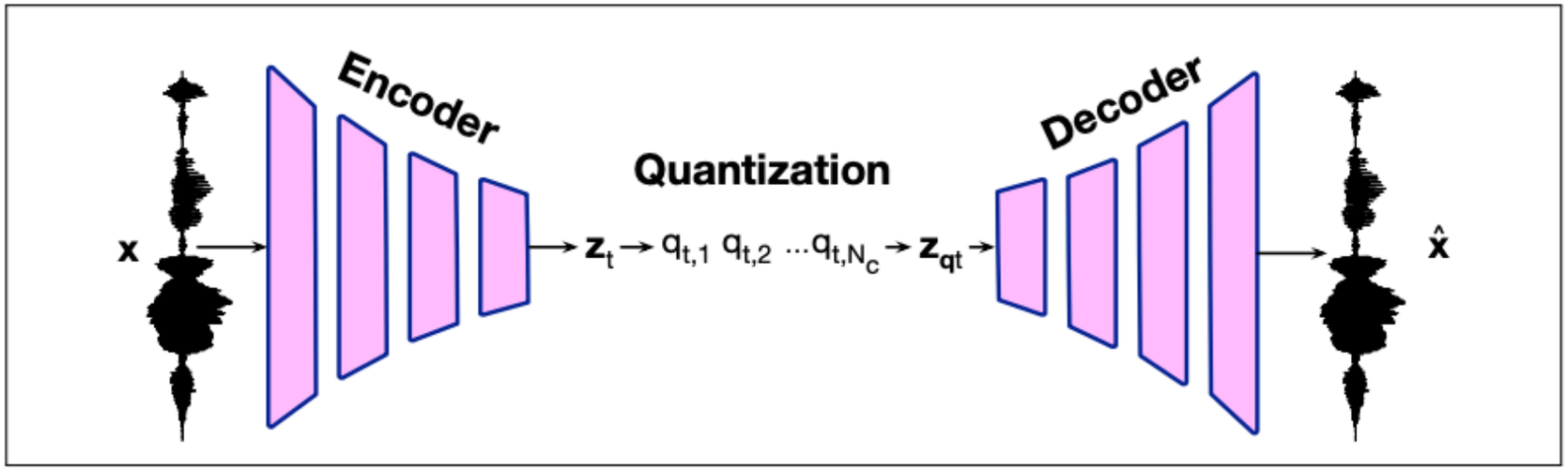


Figure 16.2 Standard architecture of an audio tokenizer performing inference, figure adapted from Mousavi et al. (2025). An input waveform $\mathbf{x}$ is encoded (generally using a series of downsampling convolution networks) into a series of embeddings $\mathbf{z_t}$. Each embedding is then passed through a quantizer to produce a series of quantized tokens q_t. To regenerate the speech signal, the quantized tokens are re-mapped back to a vector $\mathbf{z_{q_t}}$ and then encoded (usually using a series of upsampling convolution networks) back to a waveform. We'll discuss how the architecture is trained in Section 16.2.4.

Spoken Language Modelling on Speech Tokens

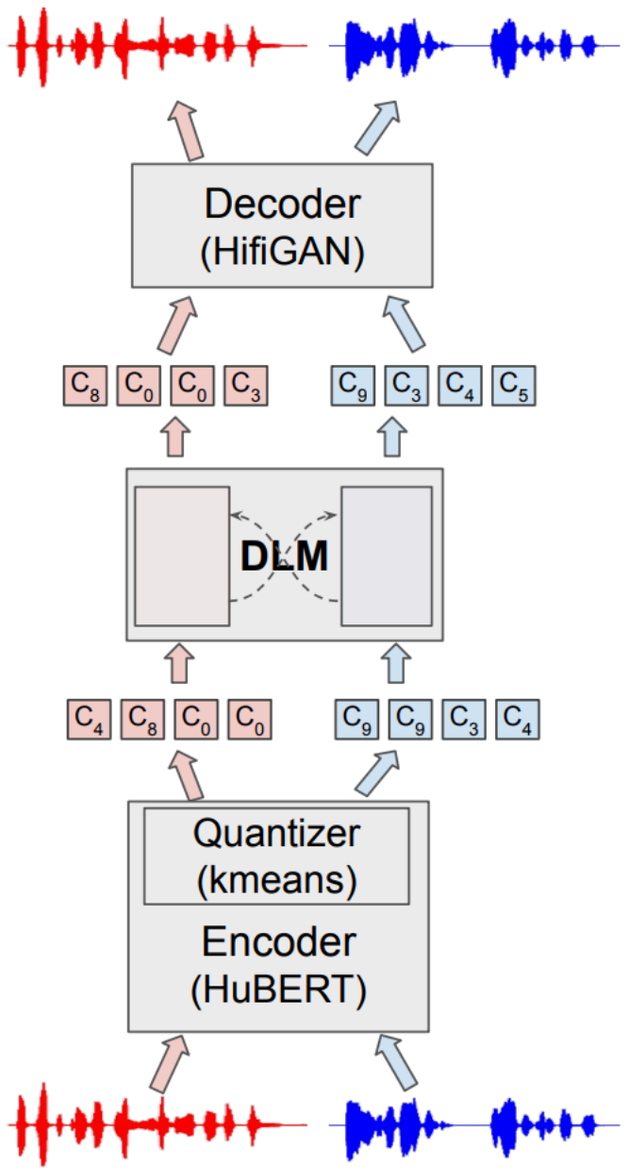


Figure 1: General Schema for dGSLM: A discrete encoder (HuBERT+kmeans) turns each channel of a dialogue into a string of discrete units $(c_1, ..c_N)$. A Dialogue Language Model (DLM) is trained to autoregressively produce units that are turned into waveforms us-

Dialog GSLM (2022)

TTS as a large language model (VALL-E)

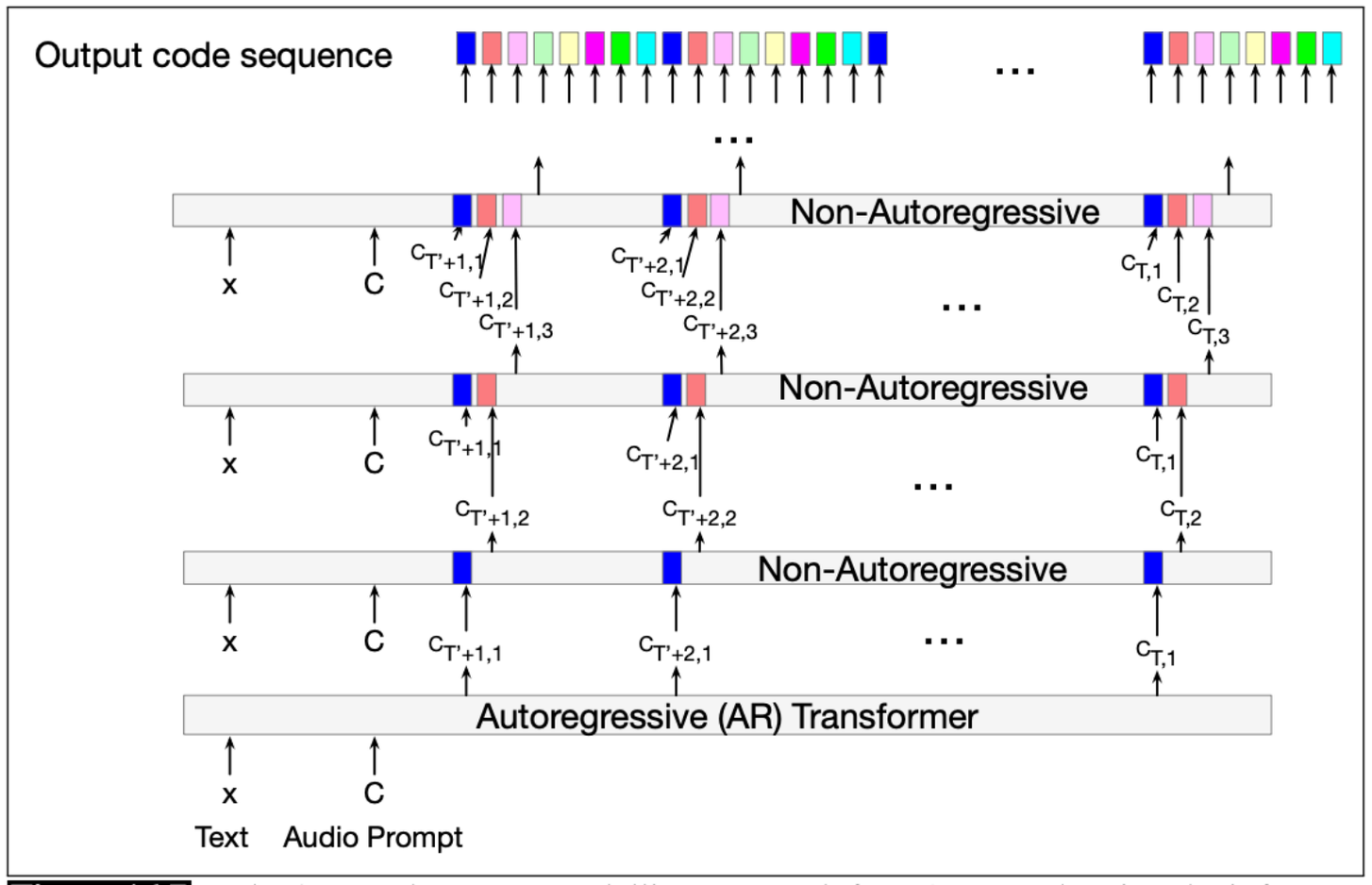


Figure 16.7 The 2-stage language modelling approach for VALL-E, showing the inference stage for the autoregressive transformer and the first 3 of the 7 non-autoregressive transformers. The output sequence of discrete audio codes is generated in two stages. First the autoregressive LM generates all the codes for the first quantizer from left to right. Then the non-autoregressive model is called 7 times to generate the remaining codes conditioned on all the codes from the preceding quantizer, including conditioning on the codes to the right.

TTS as a large language model (VALL-E)

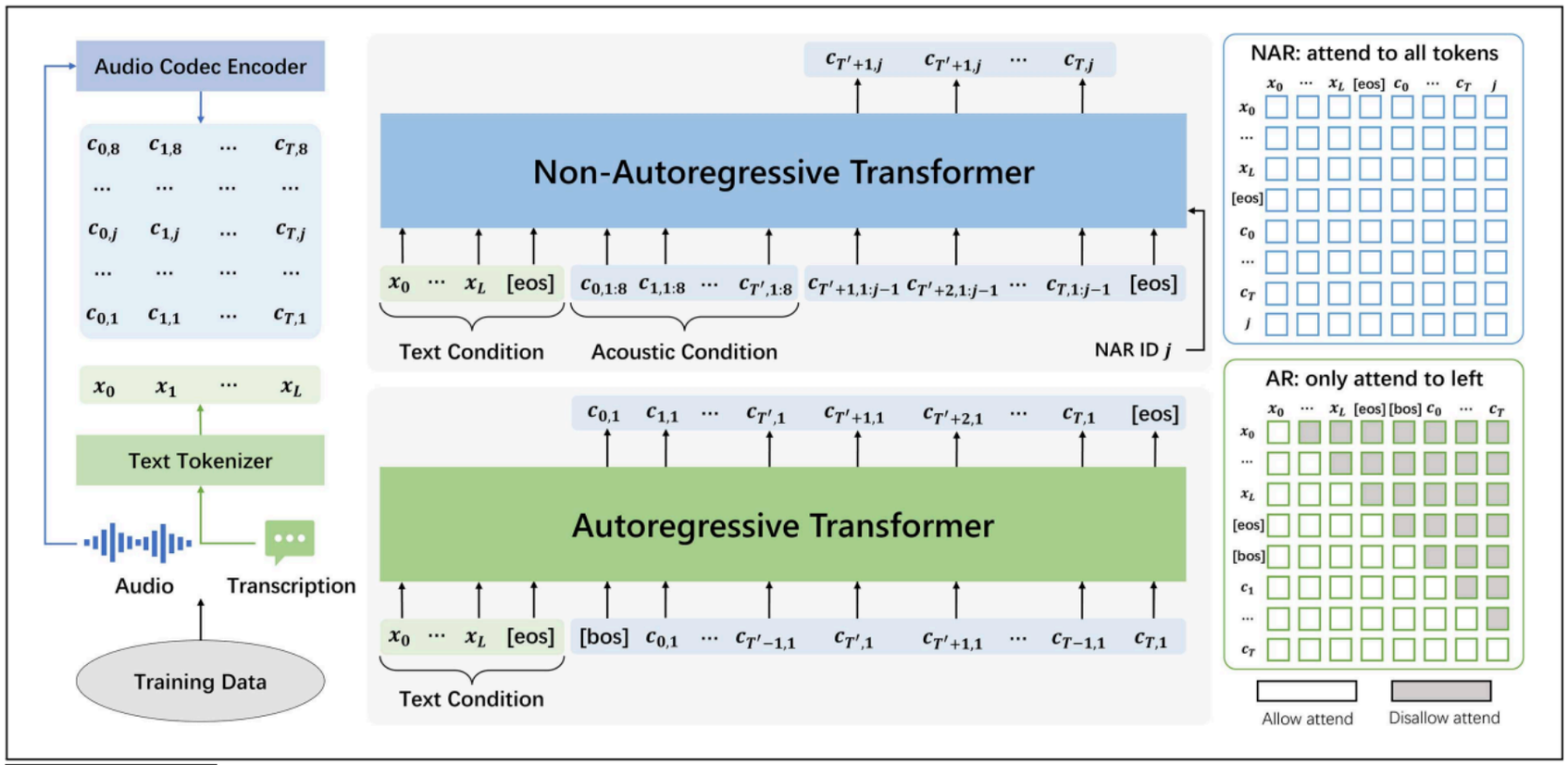


Figure 16.8 Training procedure for VALL-E. Given the text prompt, the autoregressive transformer is first trained to generate each code of the first-quantizer code sequence, autoregressively The the non-autoregressive transformer generates the rest of the codes. Figure from Chen et al. (2025).

TTS as a large language model (VALL-E)

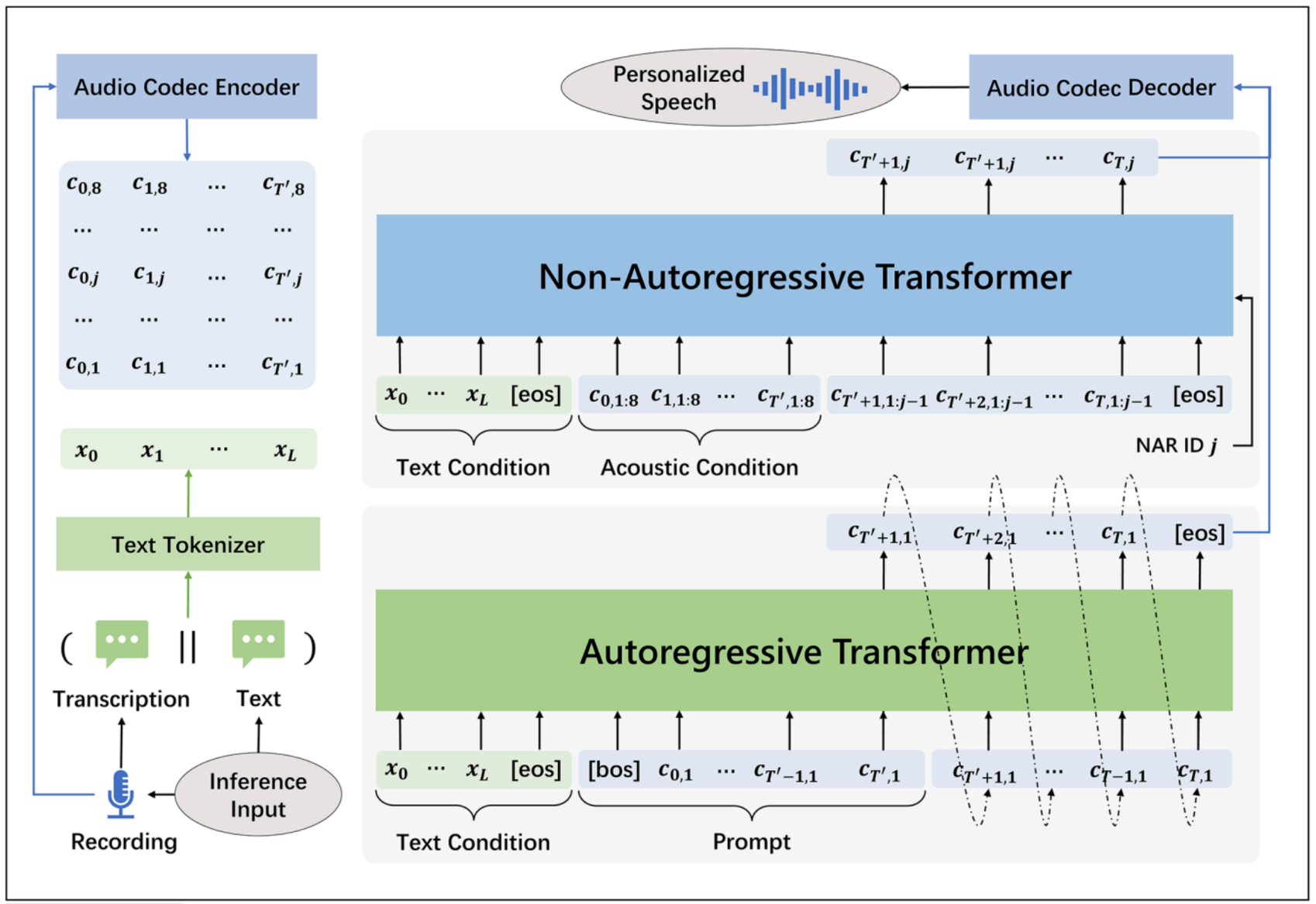
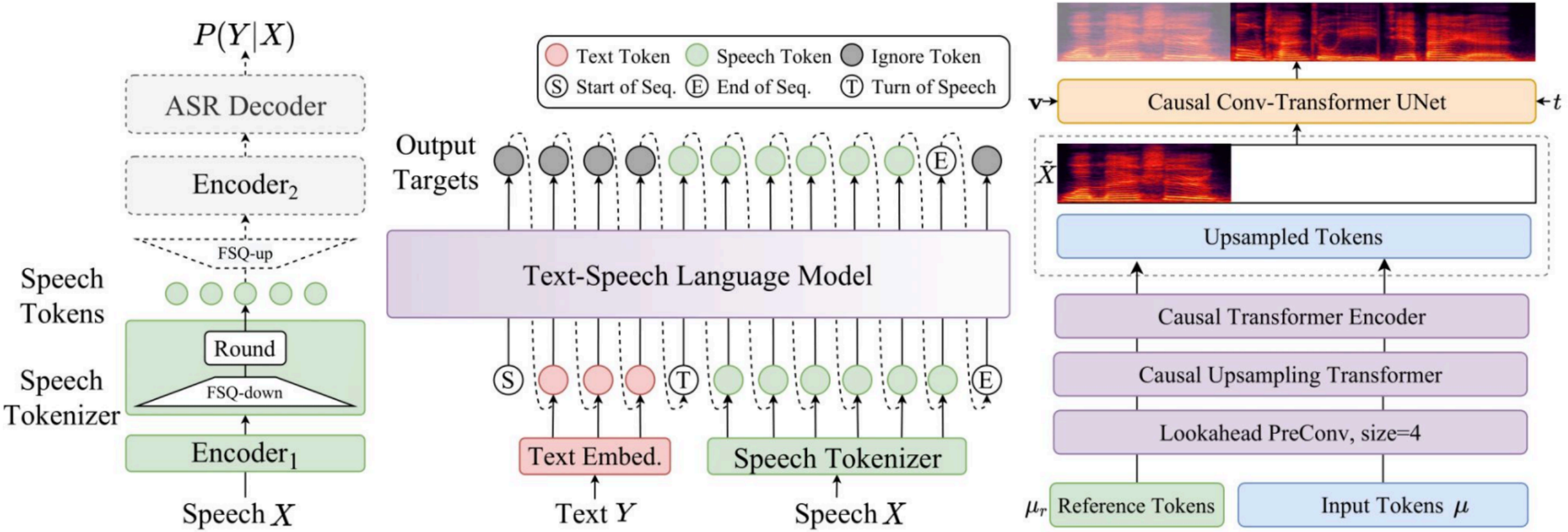


Figure 16.9 Inference procedure for VALL-E. Figure from Chen et al. (2025). The transcript for the 3 seconds of enrolled speech is first prepended to the text to be generated, and both the speech and text are tokenized. Next the autoregressive transformer starts generating the first codes $\mathbf{c}_{t'+1,1}$ conditioned on the transcript and acoustic prompt.

Interleaved modeling of Speech and Text Tokens

Cozyvoice2 (Du et al., 2024)



(a) Supervised speech tokenizer (b) Text-Speech Language Model (c) Chunk-aware Flow Matching

Current developments in TTS

- Fully Controllable TTS
  - Identity
  - Dialect
  - Contextual Appropriateness

- Explainable, light weight models
  - Grounded in Human mechanisms
  - Edge Deployable
  - Seamless interfacing with other Modalities