

Neural Speech Synthesis



Xu Tan
Microsoft Research Asia



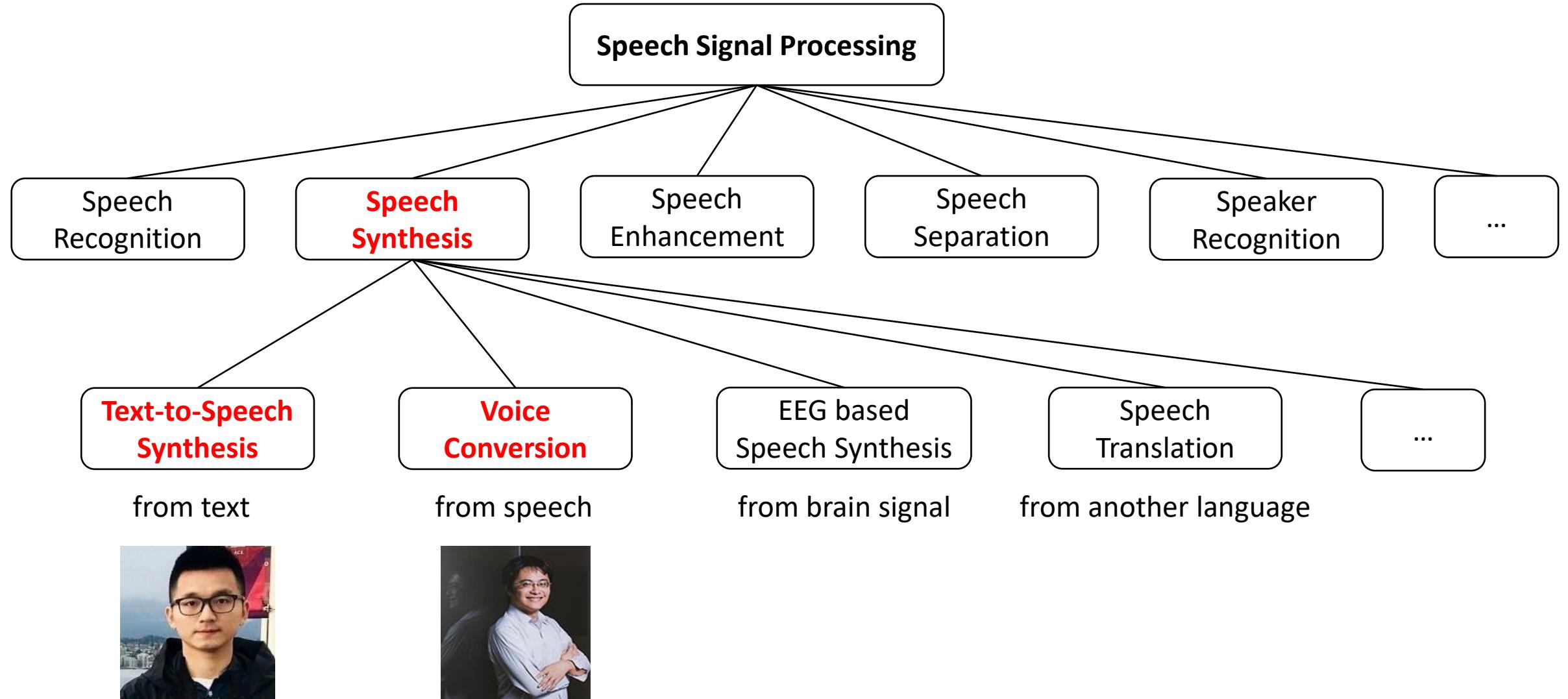
Hung-yi Lee
National Taiwan University

Speaker information

- Xu Tan (谭旭)
- Principal Researcher and Research Manager @ Microsoft Research Asia
- Research interests
 - Speech/Sound: TTS, AI music
 - NLP: machine translation, text generation, language pre-training
 - Digital human generation
- Some links
 - Homepage: <https://tan-xu.github.io/>, <https://www.microsoft.com/en-us/research/people/xuta/>
 - Google Scholar: <https://scholar.google.com/citations?user=tob-U1oAAAAJ>

Speaker information

- Hung-yi Lee (李宏毅)
- Associate Professor @ National Taiwan University
- Research interests
 - Speech processing: voice conversion, speech recognition, etc.
 - Natural language processing: abstractive summarization, question answering, etc.
- Some links
 - Homepage: <https://speech.ee.ntu.edu.tw/~hylee/index.php>
 - Google Scholar: <https://scholar.google.com/citations?user=DxLO11IAAAAJ>



Part 1: Text-to-Speech Synthesis

- Background
- Key Components of TTS
- Advanced Topics in TTS
- Summary and Future Directions

Previous TTS tutorials: ISCSLP 2021, IJCAI 2021, ICASSP 2022

<https://github.com/tts-tutorial>

TTS survey paper: *A Survey on Neural Speech Synthesis*

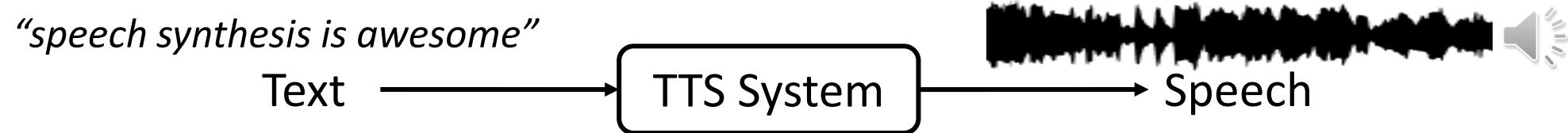
<https://arxiv.org/pdf/2106.15561.pdf>

Part 1: Text-to-Speech Synthesis

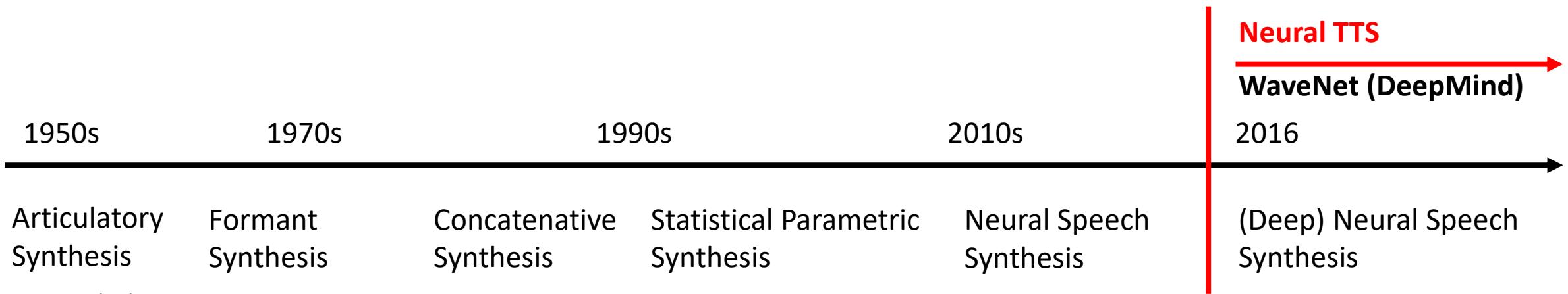
Part 1.1: Background

Text-to-Speech Synthesis

- Text-to-speech (TTS): generate intelligible and natural speech from text



- Enabling machine to speak is an important part of AI
 - **TTS (speaking)** is as important as **ASR (listening)**, **NLU (reading)**, **NLG (writing)**
 - Human beings tried to build TTS systems dating back to the **12th century**



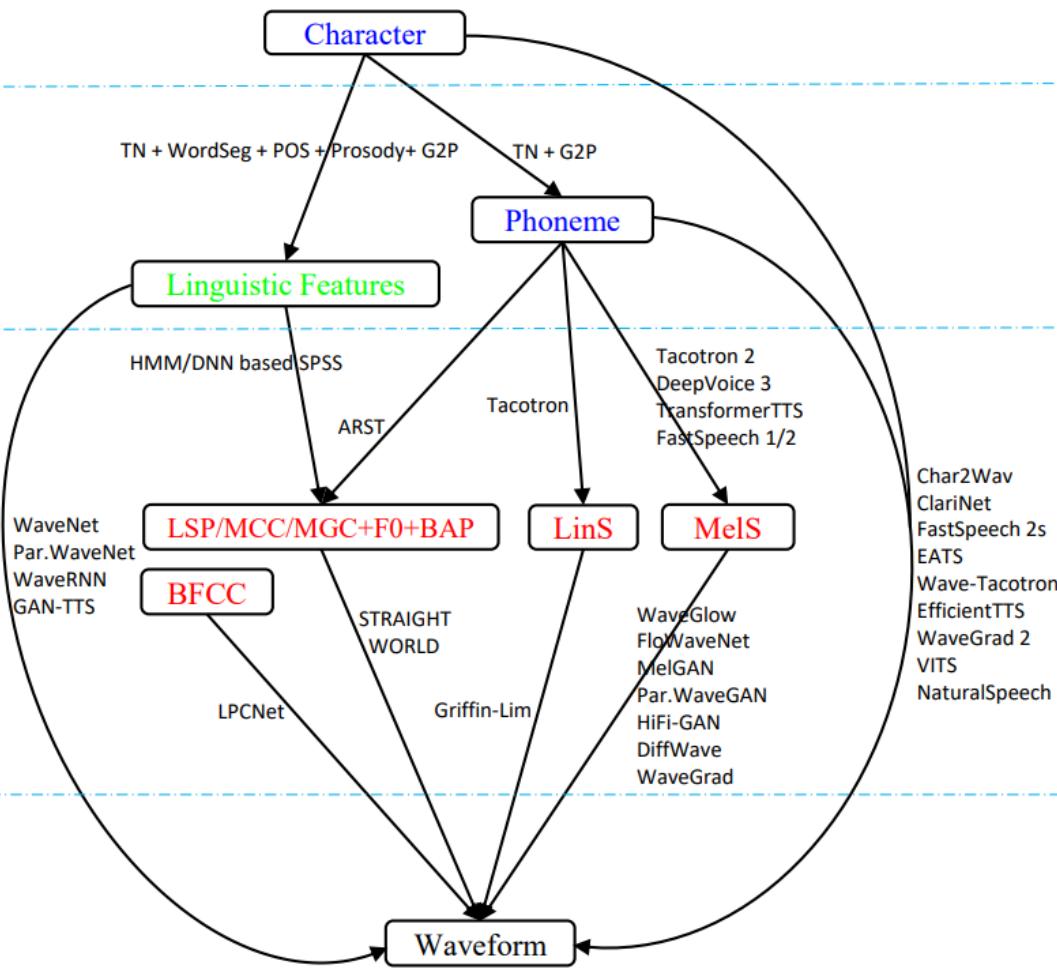
Data conversion pipeline

Text

Linguistic Features

Acoustic Features

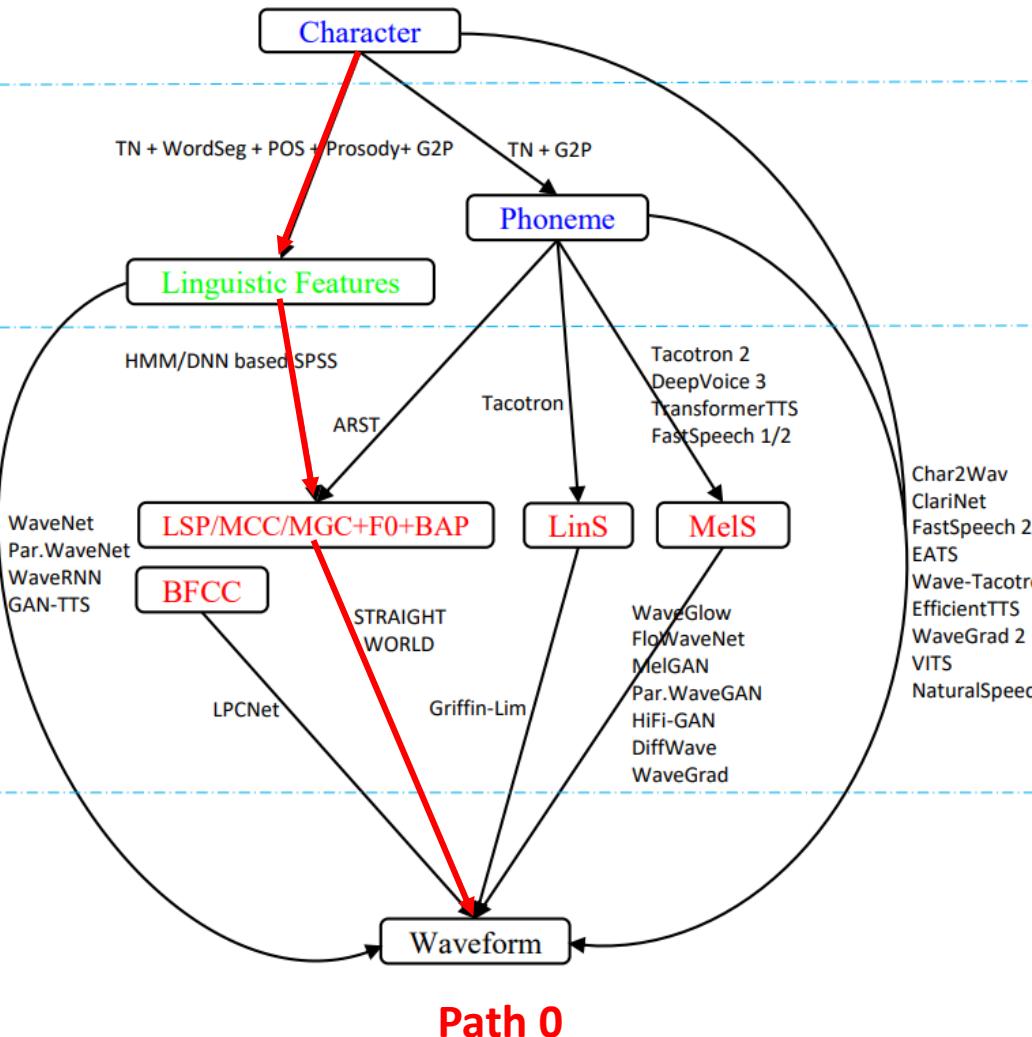
Waveform



Char2Wav
ClariNet
FastSpeech 2s
EATS
Wave-Tacotron
EfficientTTS
WaveGrad 2
VITS
NaturalSpeech

Data conversion pipeline

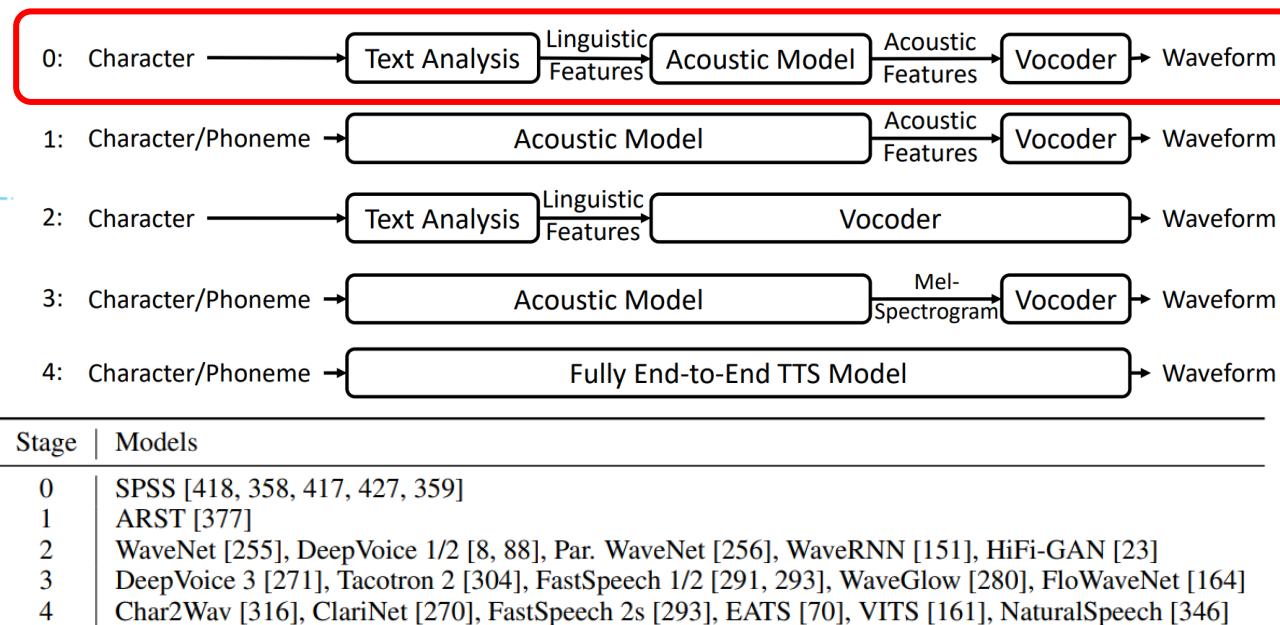
Text



Linguistic Features

Acoustic Features

Waveform



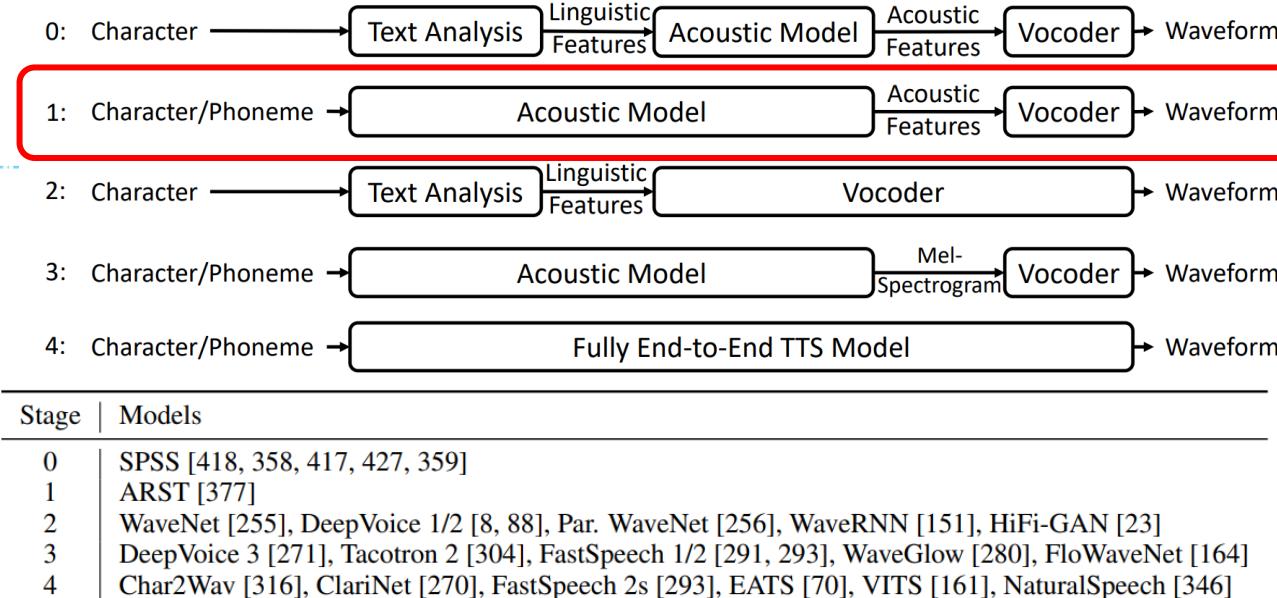
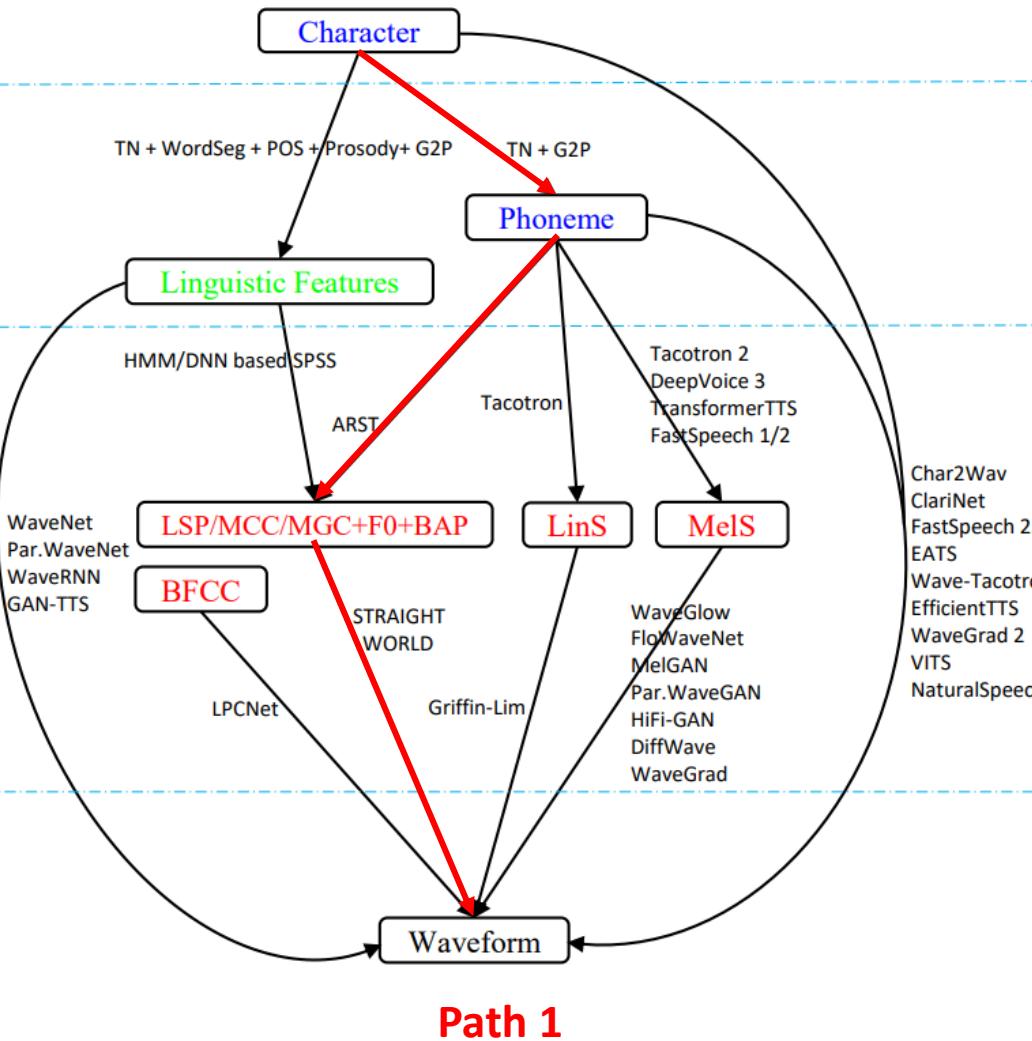
Data conversion pipeline

Text

Linguistic Features

Acoustic Features

Waveform



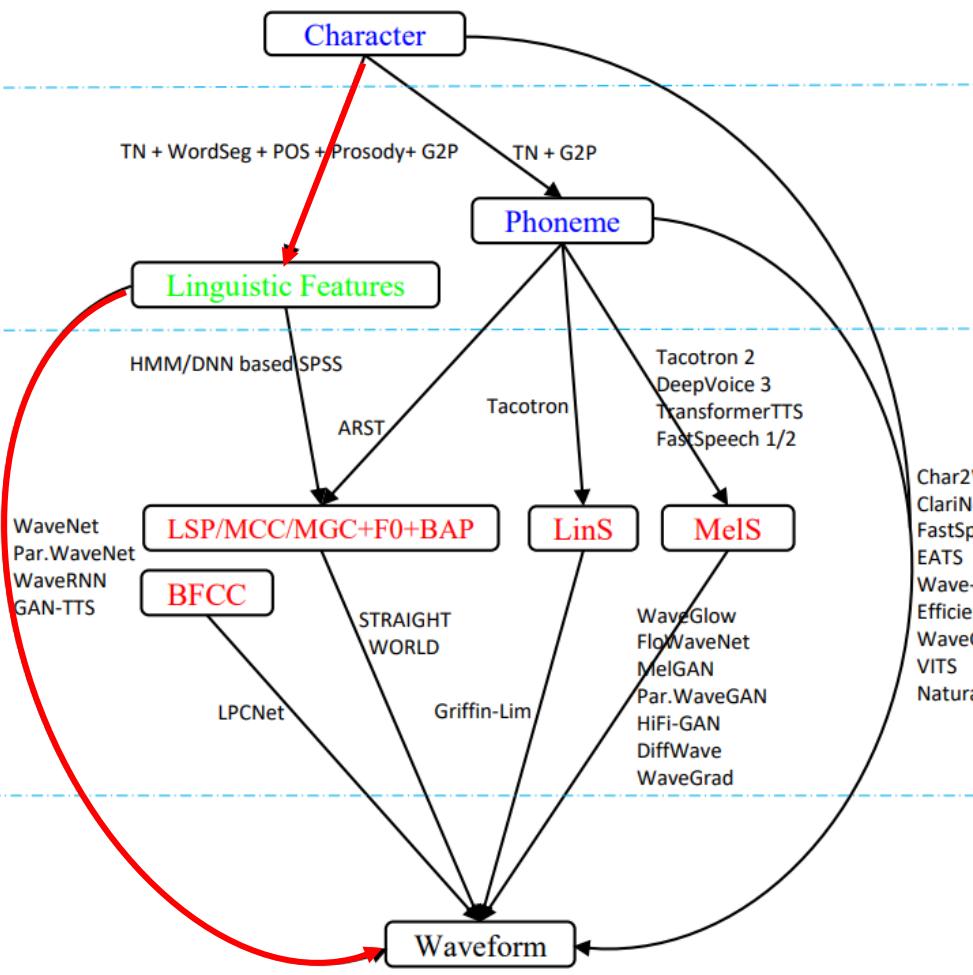
Data conversion pipeline

Text

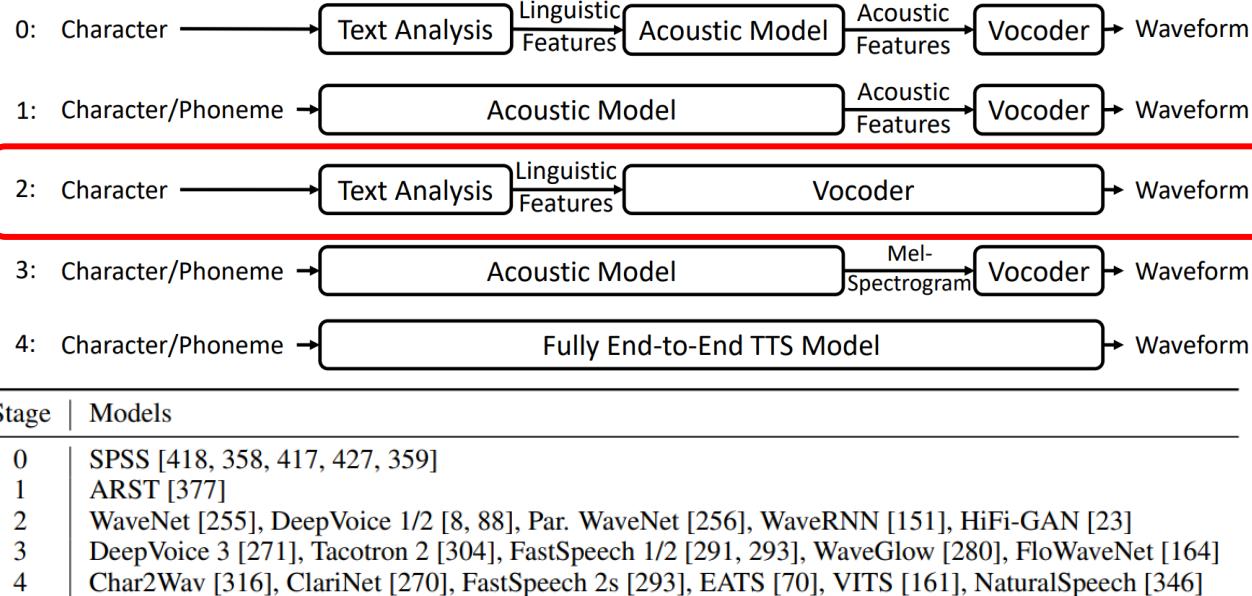
Linguistic Features

Acoustic Features

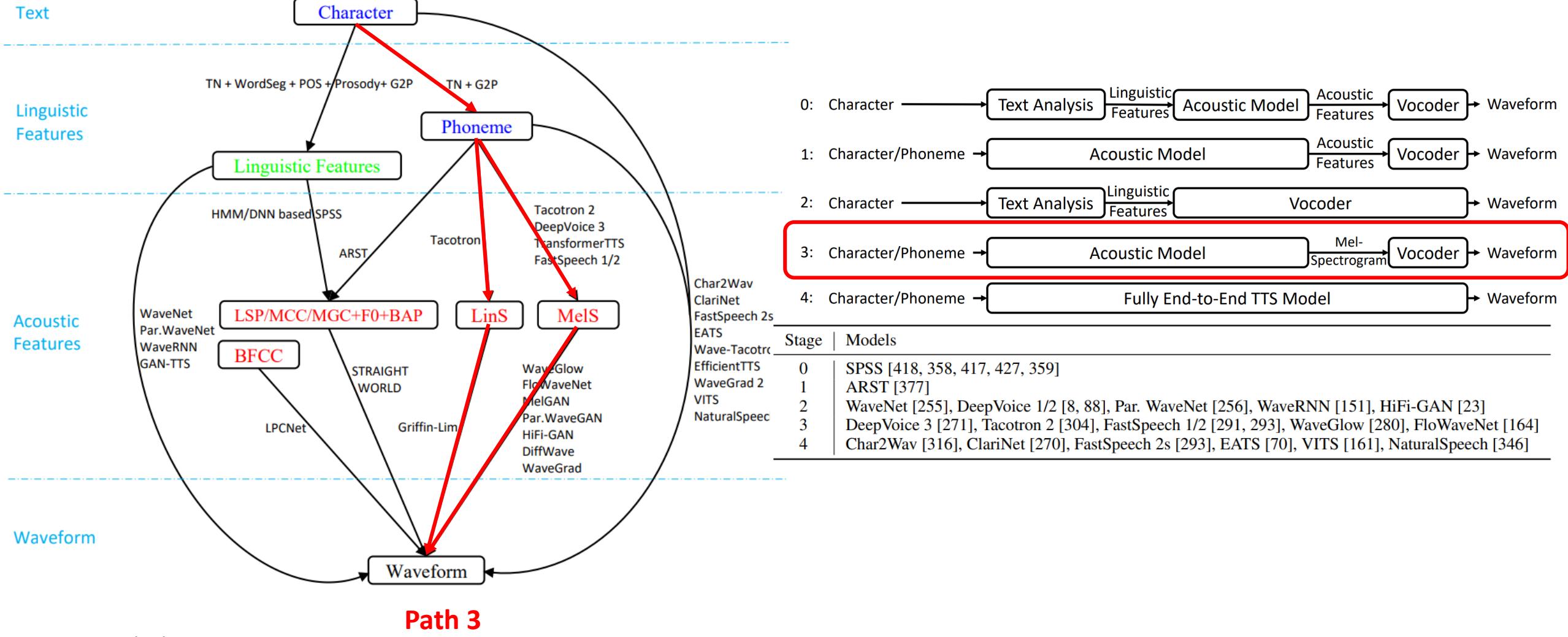
Waveform



Path 2



Data conversion pipeline



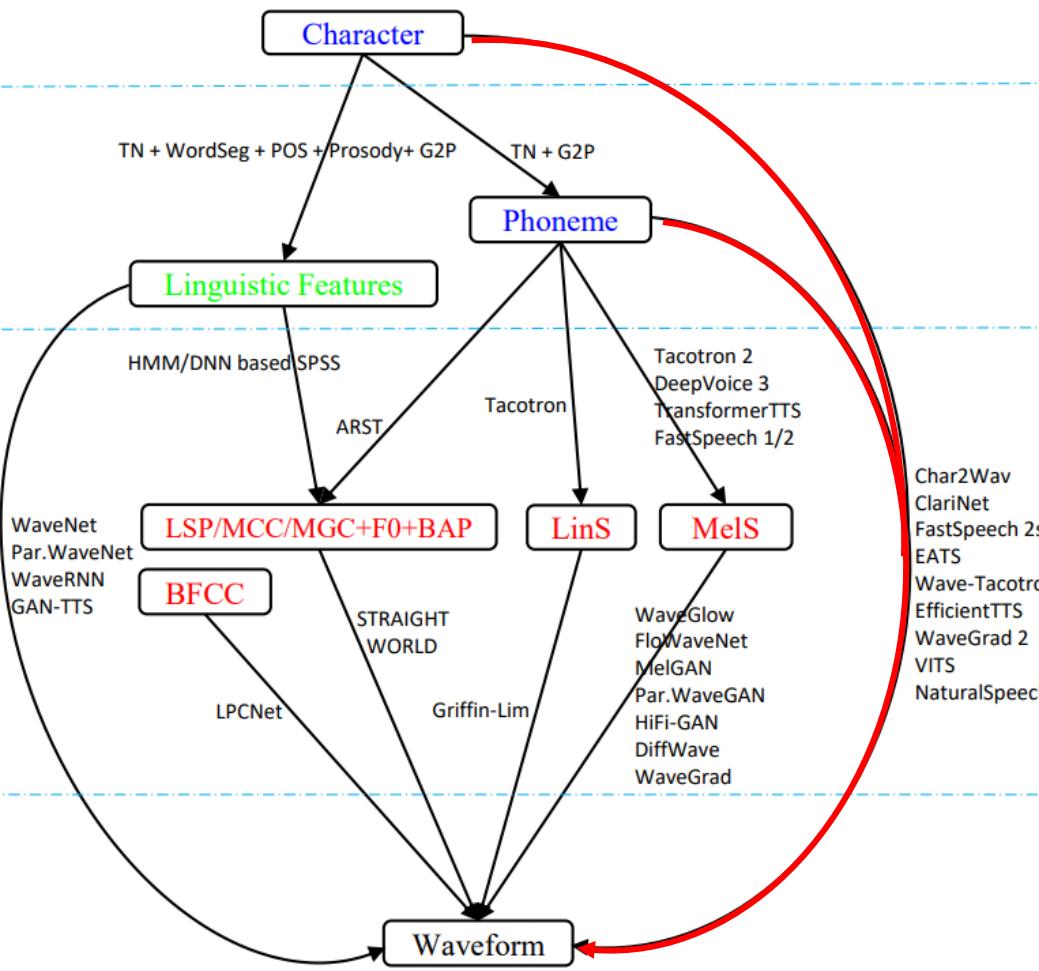
Data conversion pipeline

Text

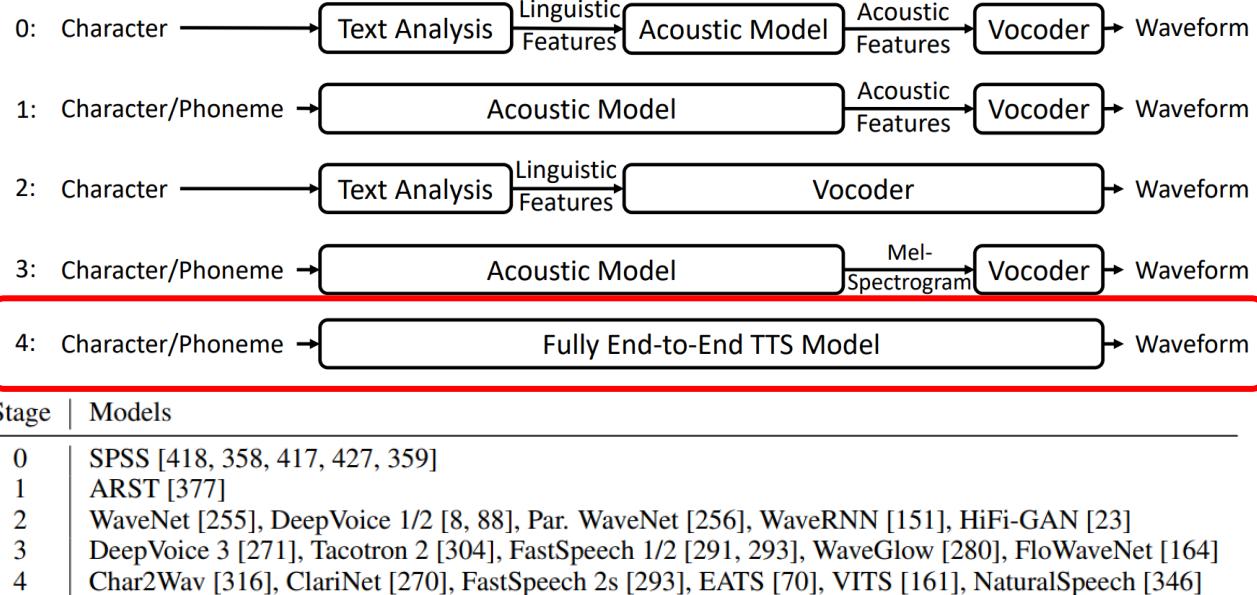
Linguistic Features

Acoustic Features

Waveform



Path 4

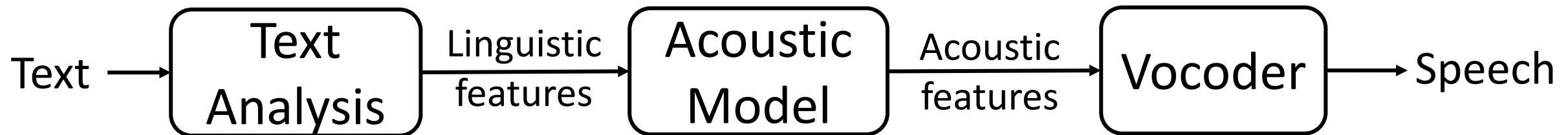


Part 1: Text-to-Speech Synthesis

Part 1.2: Key Components of TTS

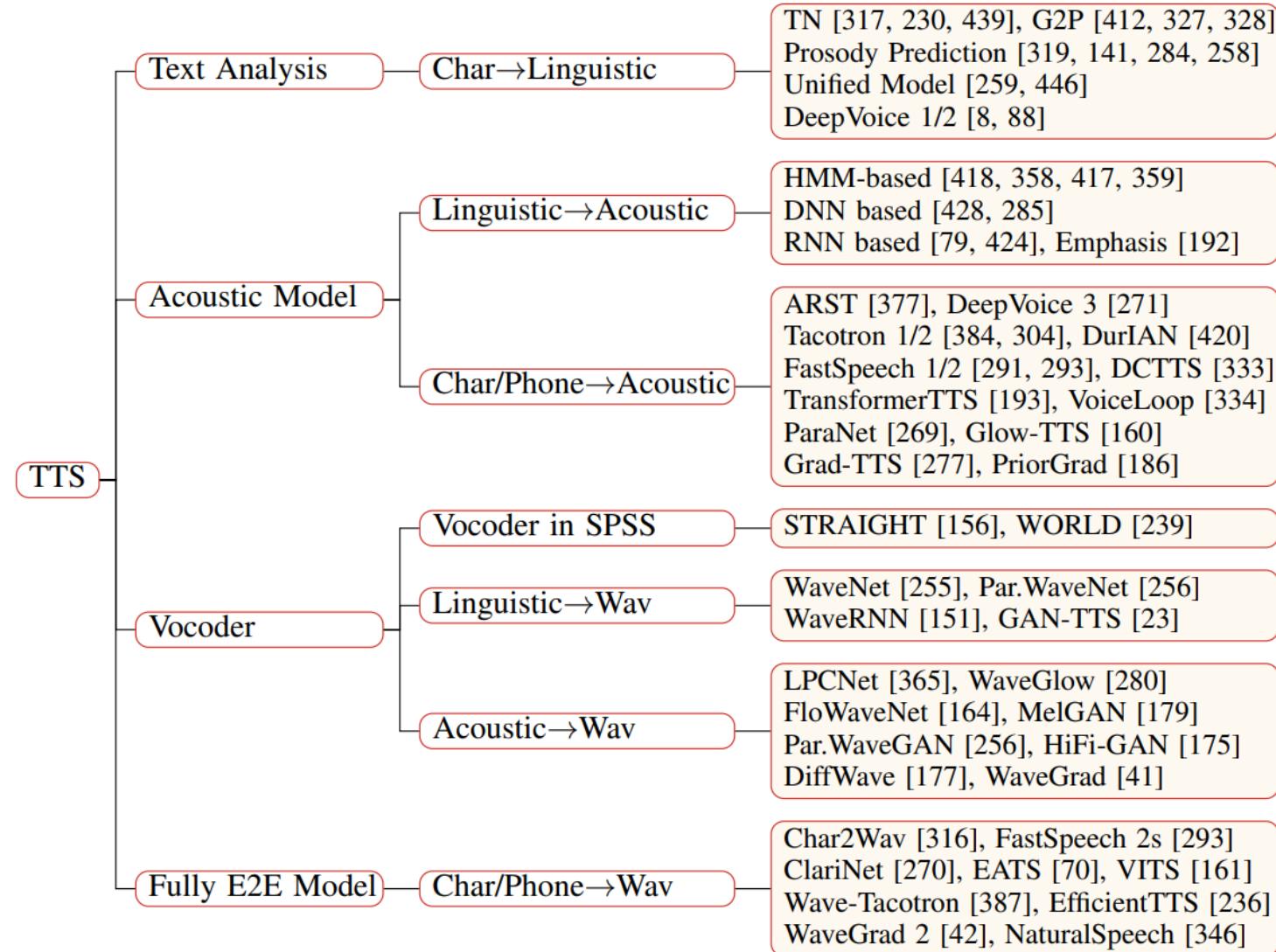
Key components of neural TTS systems

- Text analysis, acoustic model, and vocoder



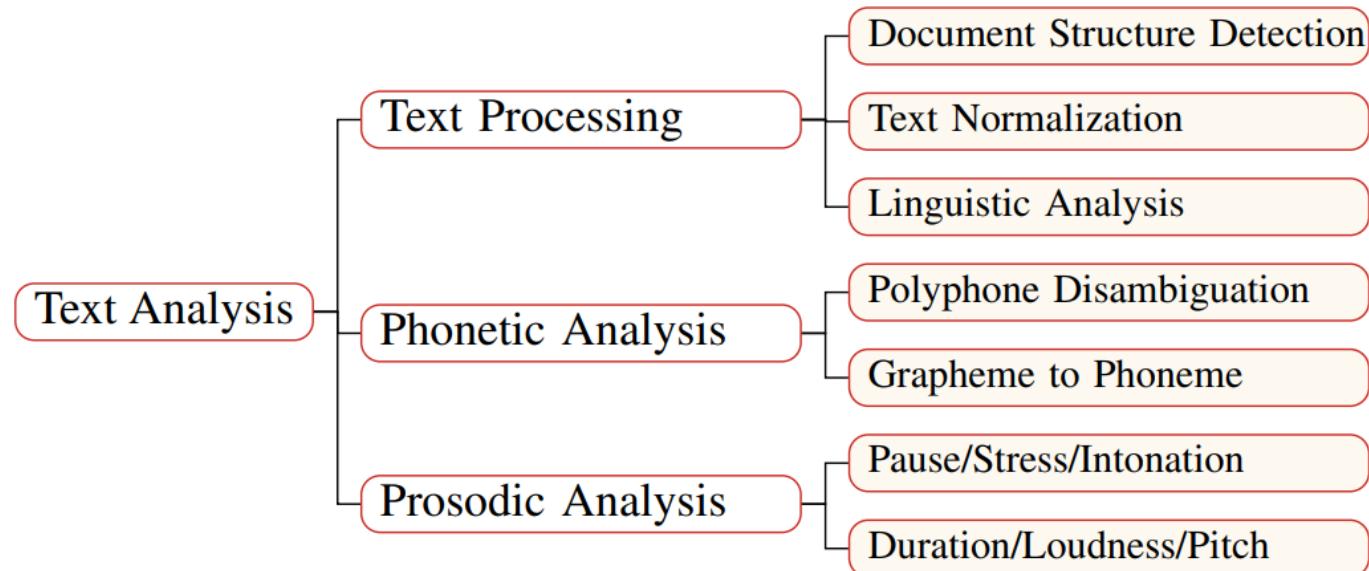
- Text analysis: text → linguistic features
- Acoustic model: linguistic features → acoustic features
- Vocoder: acoustic features → speech

Key components in TTS



Text analysis

- Transform input text into linguistic features that contain rich information about pronunciation and prosody to ease the speech synthesis.



Text analysis——Text processing

- Document Structure Detection
 - Sentence breaking: a knowledge of the sentence unit is important for correct pronunciation and prosodic breaking
- Text Normalization
 - Convert text from nonorthographic form (written form) into orthographic form (speakable form)
 - 2:18 pm, 05/23/2022, \$32
- Linguistic Analysis
 - Sentence Type Detection: . ! ?
 - Word/Phrase Segmentation: Chinese word segmentation
 - Part-of-Speech Tagging: noun, verb, preposition

Text analysis——Phonetic analysis

- Polyphone Disambiguation
 - Polyphone refers to word that can be pronounced in two or more different ways, where each way represents a different word sense
 - Polyphone disambiguation is to decide the appropriate pronunciation based on the context of this word/character
 - E.g., resume: /ri' zju:m' / or /' rezjumei/, “奇” in /ji-/ or /qi'/
- Grapheme-to-Phoneme Conversion
 - Transform character (grapheme) into pronunciation (phoneme)
 - Alphabetic languages (e.g., Spanish): handcrafted rules
 - Alphabetic languages (e.g., English): use G2P model and lexicon
 - Non-alphabetic languages (e.g., Chinese): use lexicon

Text analysis——Prosody analysis

- Prosody explicitly perceived by human
 - Intonation, stress pattern, loudness variations, pausing, and rhythm
- Latent factors: Pitch, Duration, and Energy

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

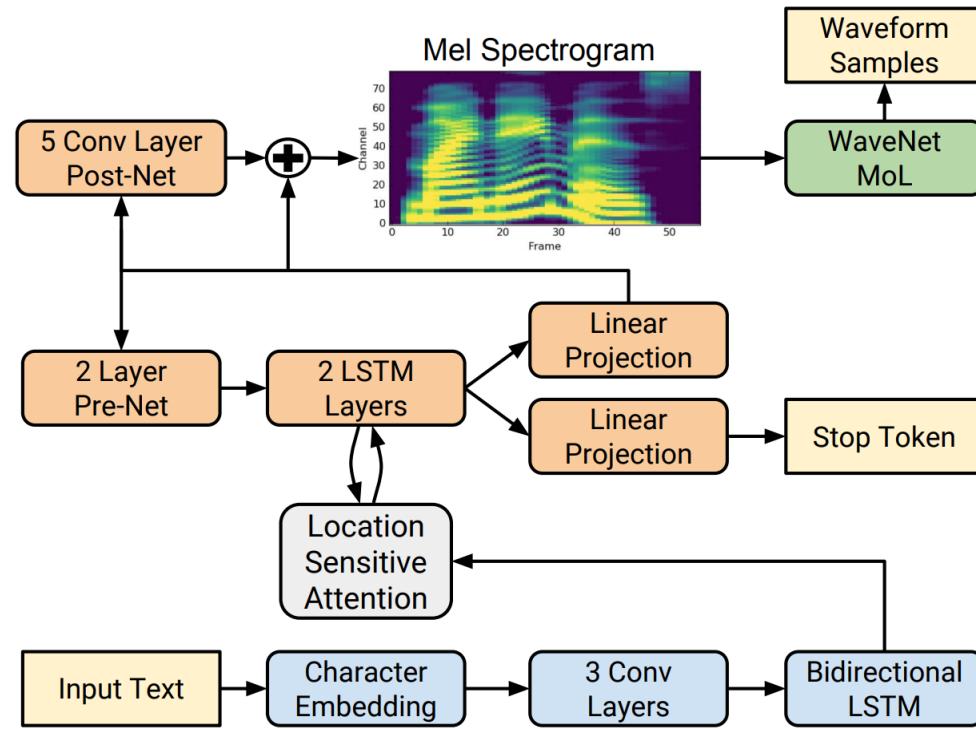
GAN

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
JDIT-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

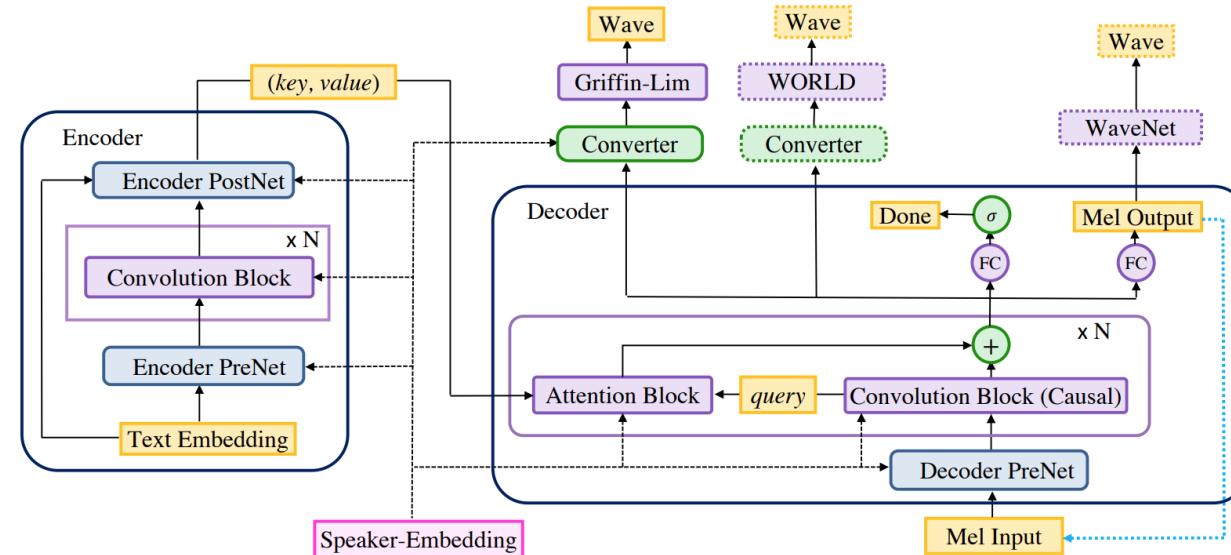
Acoustic model——RNN based

- Tacotron 2 [303]
 - Evolved from Tacotron [382]
 - Text to mel-spectrogram generation
 - LSTM based encoder and decoder
 - Location sensitive attention
 - WaveNet as the vocoder
- Other works
 - GST-Tacotron [383], Ref-Tacotron [309]
 - DurlAN [418]
 - Non-Attentative Tacotron [304]
 - WaveTacotron [385]



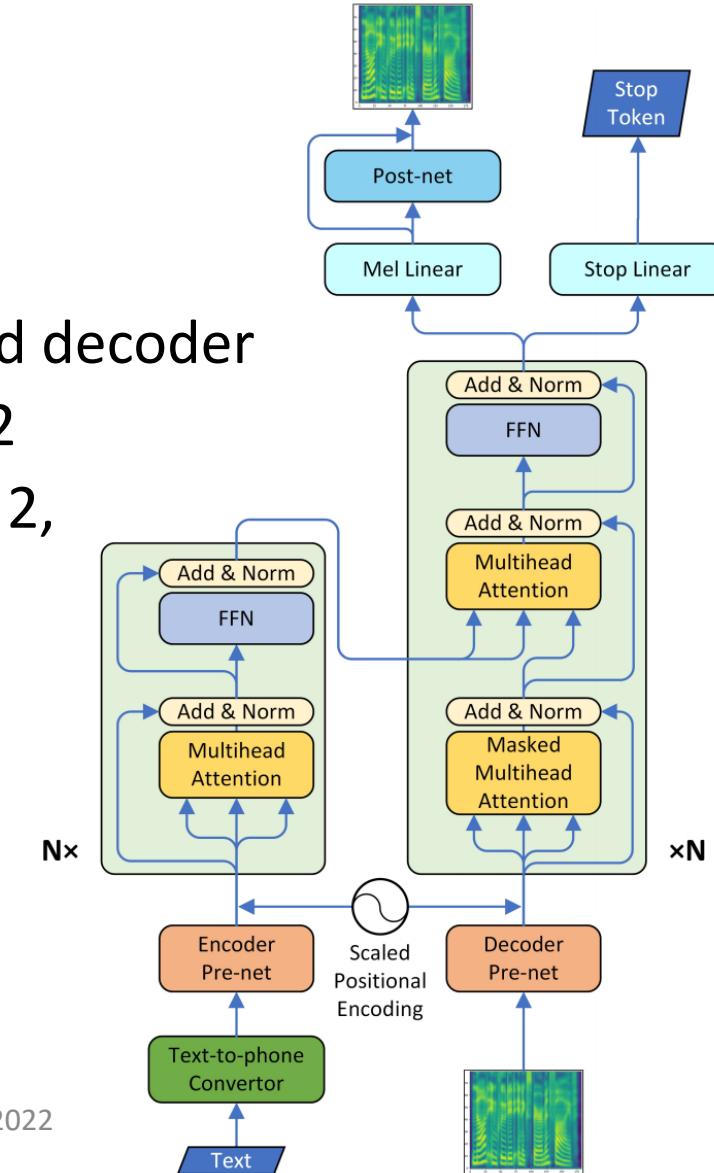
Acoustic model——CNN based

- DeepVoice 3 [270]
 - Evolved from DeepVoice 1/2 [8, 87]
 - Enhanced with purely CNN based structure
 - Support different acoustic features as output
 - Support multi-speakers
- Other works
 - DCTTS [332] (**Contemporary**)
 - ClariNet [269]
 - ParaNet [268]



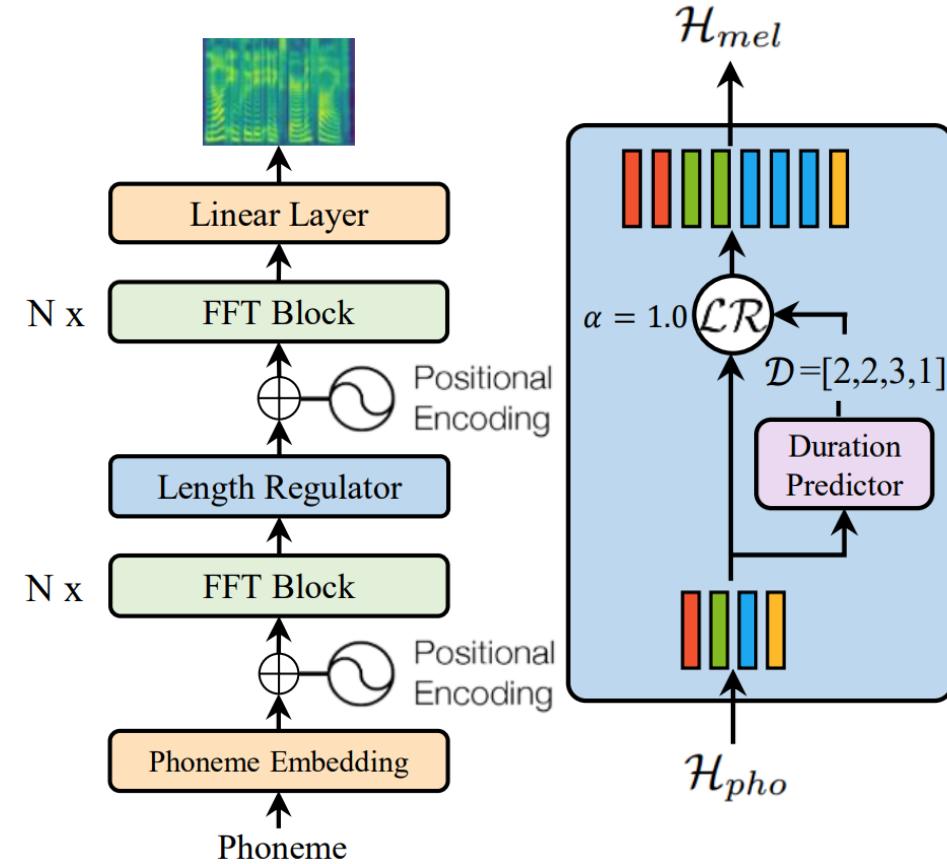
Acoustic model——Transformer based

- TransformerTTS [192]
 - Framework is like Tacotron 2
 - Replace LSTM with Transformer in encoder and decoder
 - Parallel training, quality on par with Tacotron 2
 - Attention with more challenges than Tacotron 2, due to parallel computing
- Other works
 - MultiSpeech [39]
 - Robutrans [194]



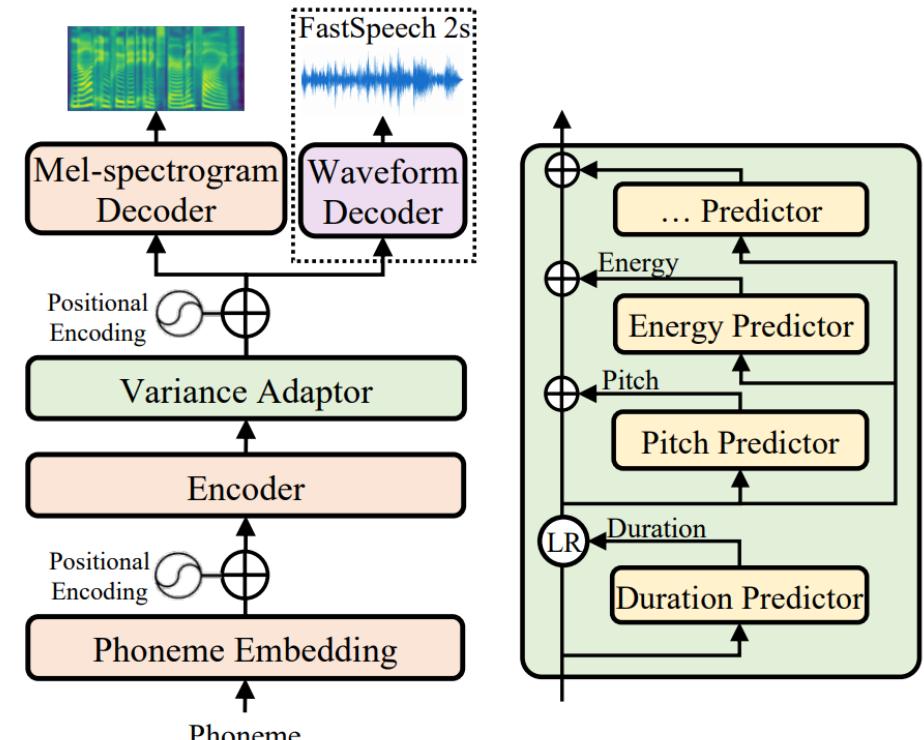
Acoustic model—Transformer based

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



(a) FastSpeech 2

(b) Variance adaptor

Vocoder

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

AR

Flow

GAN

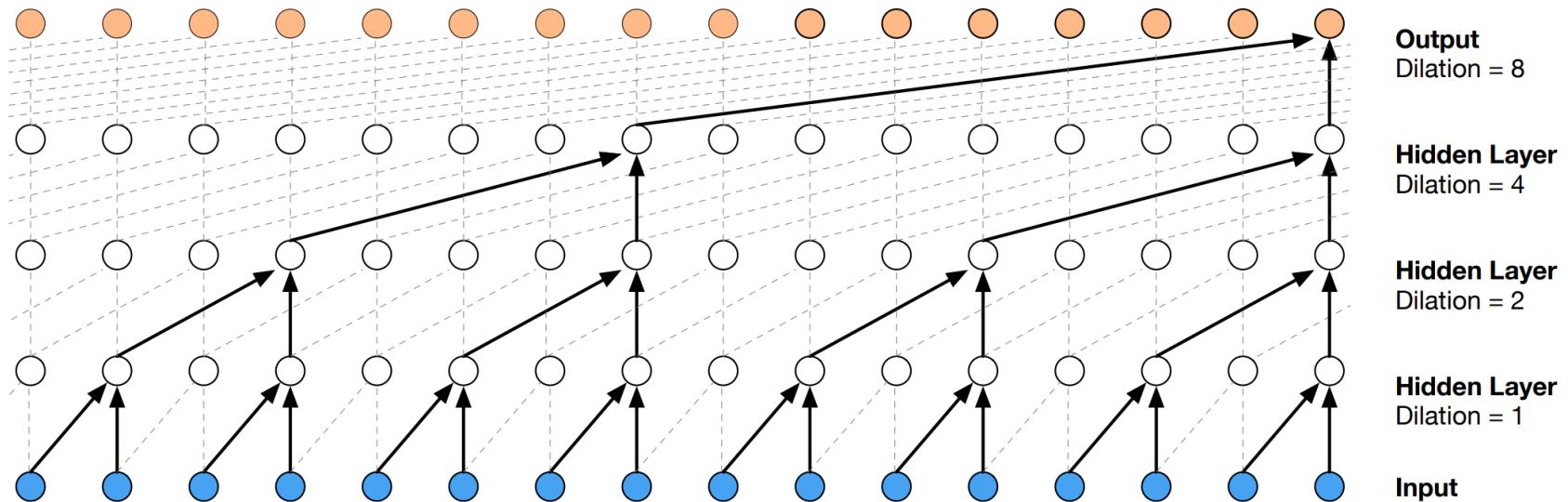
VAE

Diffusion

Vocoder	Input	AR/NAR	Modeling	Architecture
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
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
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
Wave-VAE [274]	Mel-Spectrogram	NAR	VAE	CNN
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

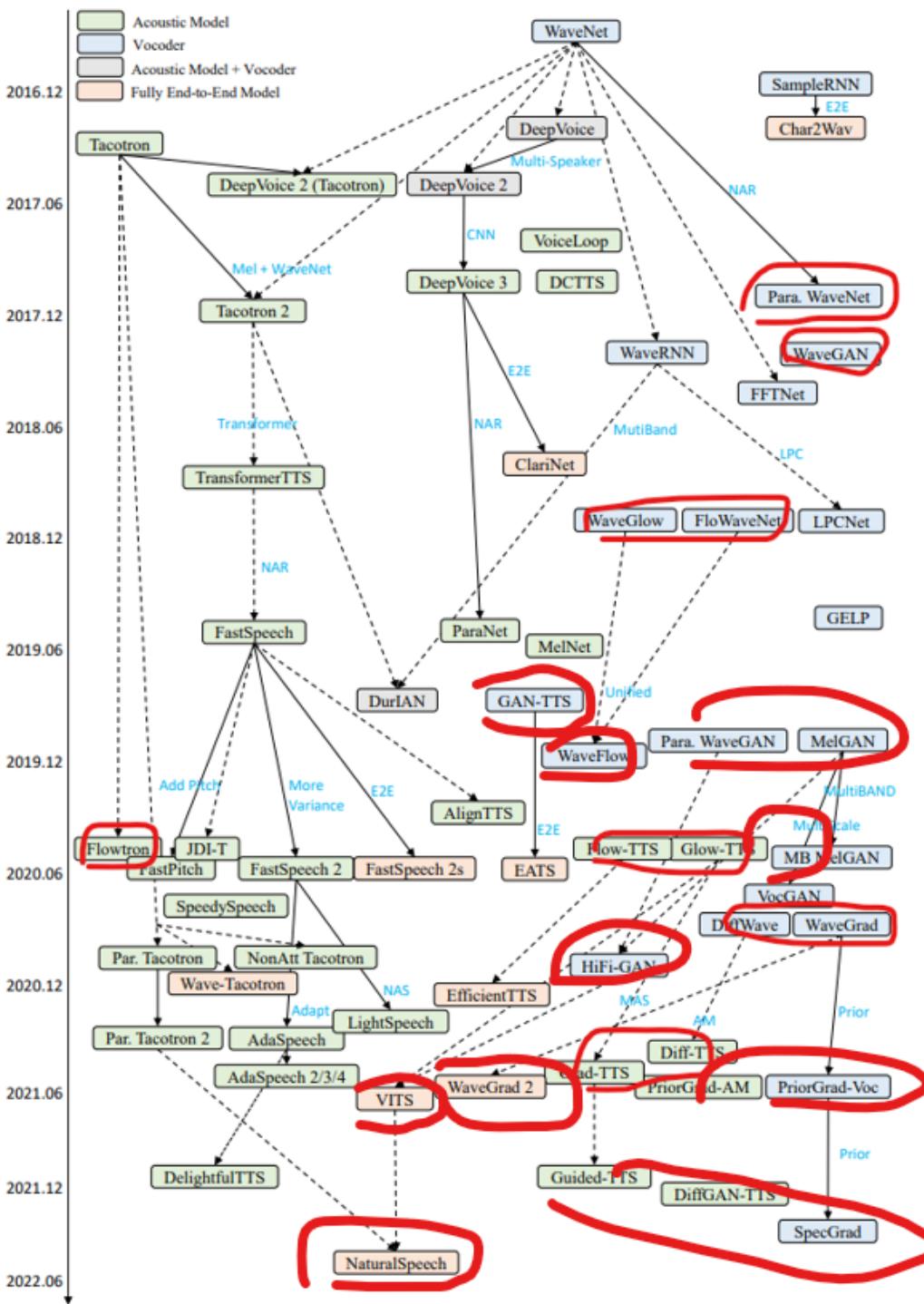
- WaveNet: autoregressive model with dilated causal convolution [254]



- Other works
 - WaveRNN [150]
 - LPCNet [363]

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 ☺



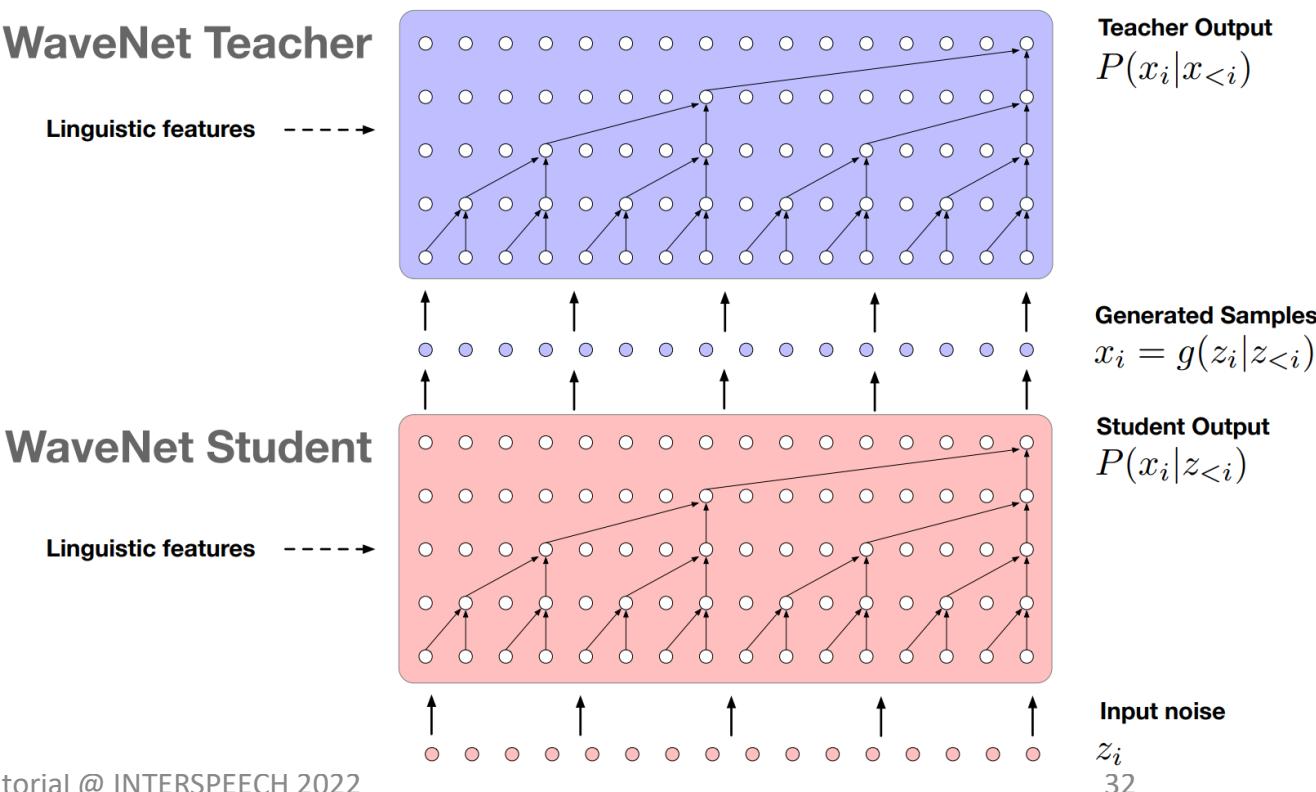
Generative models—Flow

- Map between data distribution $p(x)$ and standard (normalizing) prior distribution $p(z)$ Evaluation $z = f^{-1}(x)$ Synthesis $x = f(z)$
- Category of normalizing flow
 - AR (autoregressive): AF (autoregressive flow) and IAF (inverse autoregressive flow)
 - Bipartite: RealNVP and Glow

Flow	Evaluation $z = f^{-1}(x)$	Synthesis $x = f(z)$
AR	AF [261] $z_t = x_t \cdot \sigma_t(x_{<t}; \theta) + \mu_t(x_{<t}; \theta)$	$x_t = \frac{z_t - \mu_t(x_{<t}; \theta)}{\sigma_t(x_{<t}; \theta)}$
	IAF [169] $z_t = \frac{x_t - \mu_t(z_{<t}; \theta)}{\sigma_t(z_{<t}; \theta)}$	$x_t = z_t \cdot \sigma_t(z_{<t}; \theta) + \mu_t(z_{<t}; \theta)$
Bipartite	RealNVP [66] $z_a = x_a,$	$x_a = z_a,$
	Glow [167] $z_b = x_b \cdot \sigma_b(x_a; \theta) + \mu_b(x_a; \theta)$	$x_b = \frac{z_b - \mu_b(x_a; \theta)}{\sigma_b(x_a; \theta)}$

Generative models—Flow

- Parallel WaveNet [255] (AR)
 - Knowledge distillation: Student (IAF). Teacher (AF)
 - Combine the best of both worlds **WaveNet Teacher**
 - Parallel inference of IAF student
 - Parallel training of AF teacher
- Other works
 - ClariNet [269]



Generative models—Flow

- WaveGlow [279] (Bipartite)

- Flow based transformation

$$z = f_k^{-1} \circ f_{k-1}^{-1} \circ \dots f_0^{-1}(x) \quad x = f_0 \circ f_1 \circ \dots f_k(z)$$

- Affine Coupling Layer

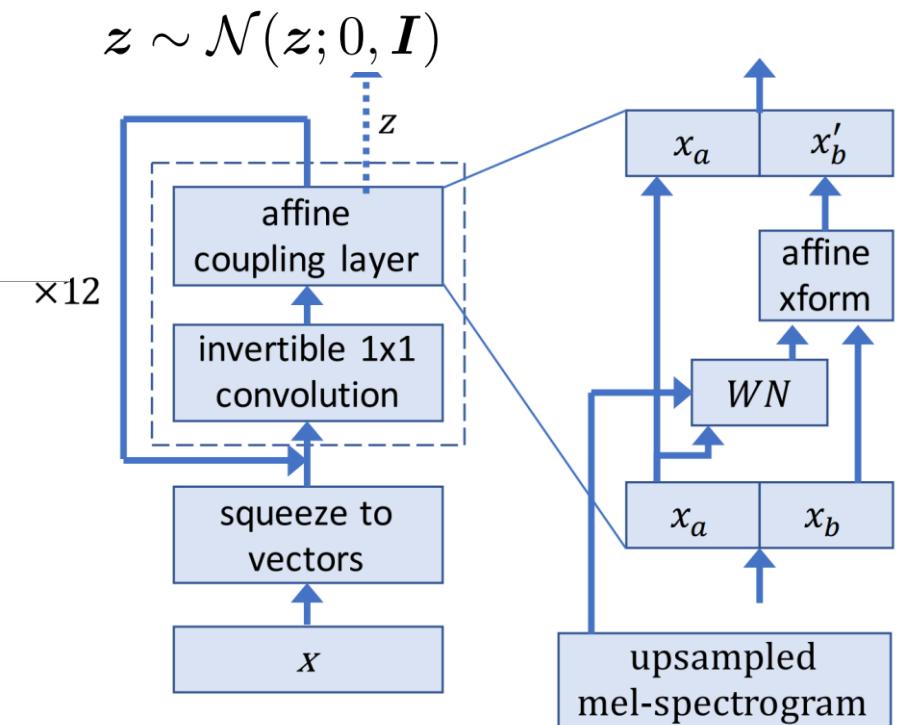
$$\begin{aligned} x_a, x_b &= \text{split}(x) \\ (\log s, t) &= WN(x_a, \text{mel-spectrogram}) \end{aligned}$$

$$x_{b'} = s \odot x_b + t$$

$$f_{coupling}^{-1}(x) = \text{concat}(x_a, x_{b'})$$

- Other works

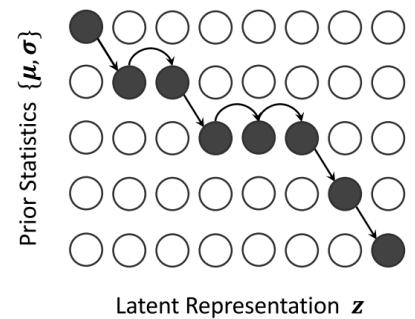
- FloWaveNet [163]
 - WaveFlow [271]



Generative models—Flow

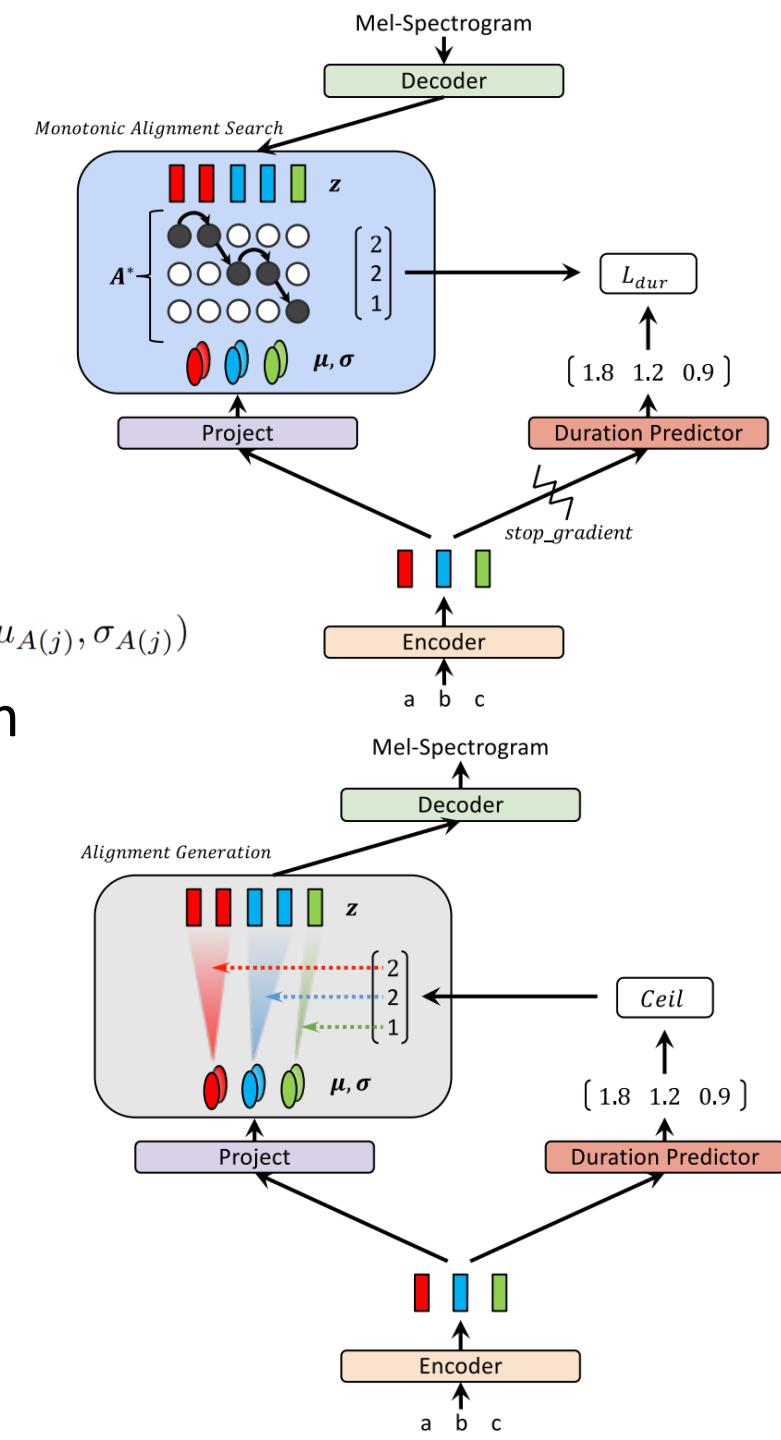
- Glow-TTS [159]

- Log likelihood $\log P_X(x|c) = \log P_Z(z|c) + \log \left| \det \frac{\partial f_{dec}^{-1}(x)}{\partial x} \right|$
- Prior is learnt from phoneme text $\log P_Z(z|c; \theta, A) = \sum_{j=1}^{T_{mel}} \log \mathcal{N}(z_j; \mu_{A(j)}, \sigma_{A(j)})$
- Alignment A is obtained by monotonic alignment search



- Other works

- FlowTTS, Flowtron, EfficientTTS



Generative models——GAN

- Adversarial loss

$$\mathcal{L}_{Adv}(D; G) = \mathbb{E}_{(x,s)} \left[(D(x) - 1)^2 + (D(G(s)))^2 \right]$$

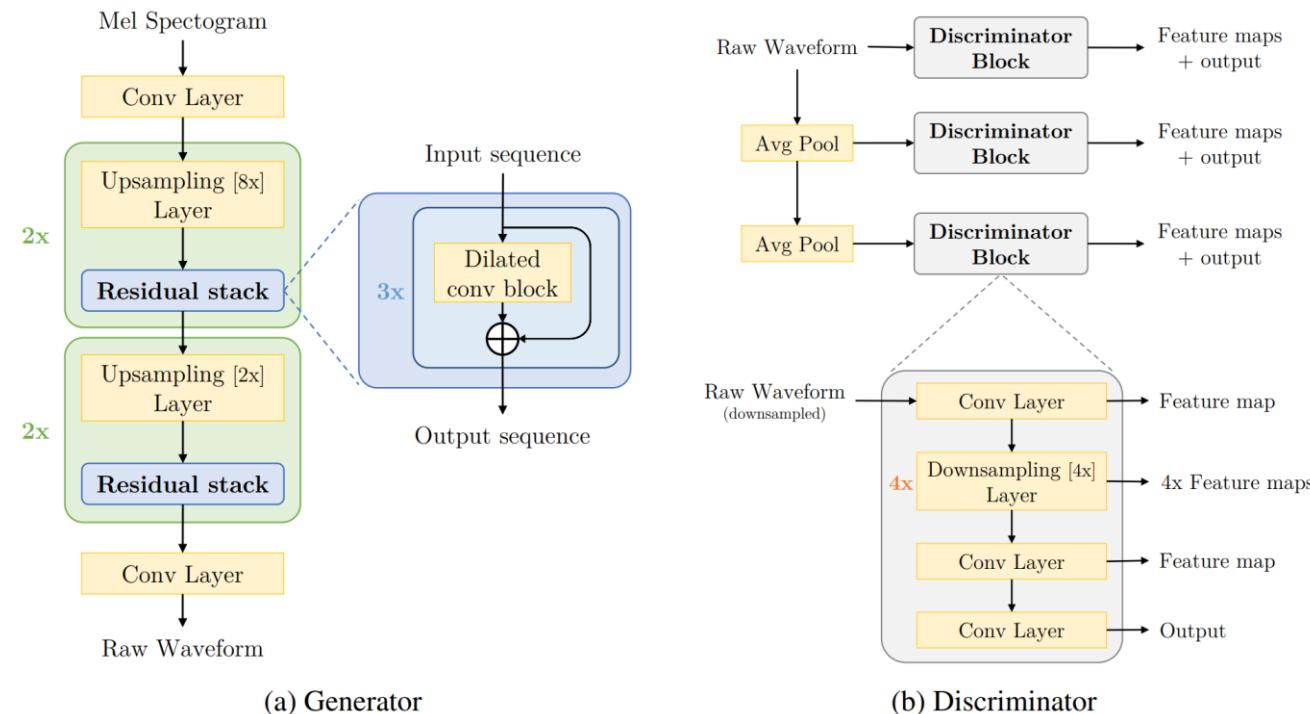
$$\mathcal{L}_{Adv}(G; D) = \mathbb{E}_s \left[(D(G(s)) - 1)^2 \right]$$

- Category of GAN based vocoders

GAN	Generator	Discriminator	Loss
WaveGAN [68]	DCGAN [287]	/	WGAN-GP [97]
GAN-TTS [23]	/	Random Window D	Hinge-Loss GAN [198]
MelGAN [178]	/	Multi-Scale D	LS-GAN [231] Feature Matching Loss [182]
Par.WaveGAN [402]	WaveNet [254]	/	LS-GAN, Multi-STFT Loss
HiFi-GAN [174]	Multi-Receptive Field Fusion	Multi-Period D, Multi-Scale D	LS-GAN, STFT Loss, Feature Matching Loss
VocGAN [408]	Multi-Scale G	Hierarchical D	LS-GAN, Multi-STFT Loss, Feature Matching Loss
GED [96]	/	Random Window D	Hinge-Loss GAN, Repulsive loss

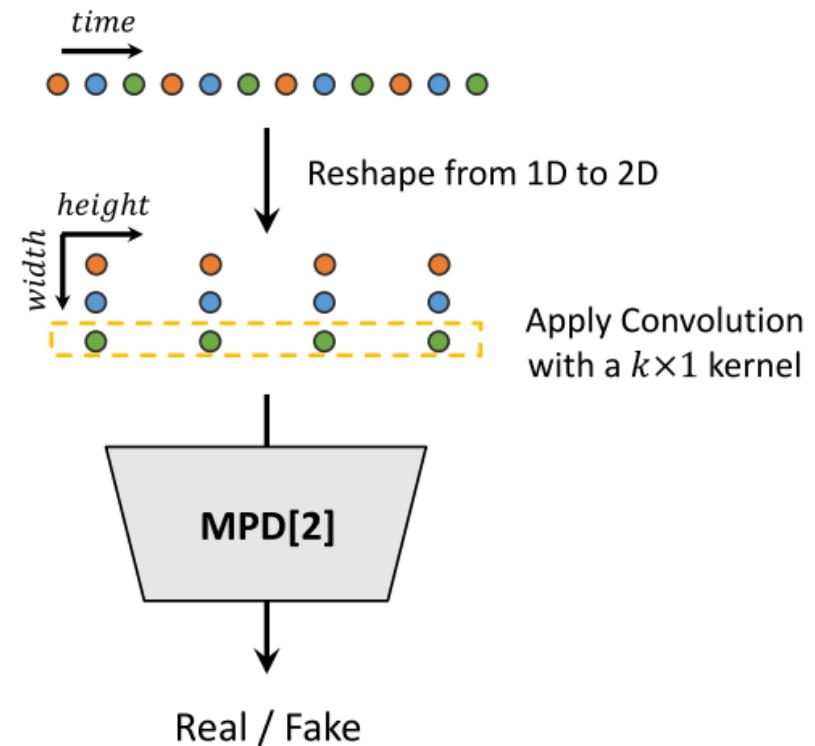
Generative models——GAN

- MelGAN [68]
 - Generator: Transposed conv for upsampling, dilated conv to increase receptive field
 - Discriminator: Multi-scale discrimination



Generative models——GAN

- HiFiGAN [68]
 - Multi-Scale Discriminator (MSD)
 - Multi-Period Discriminator (MPD)

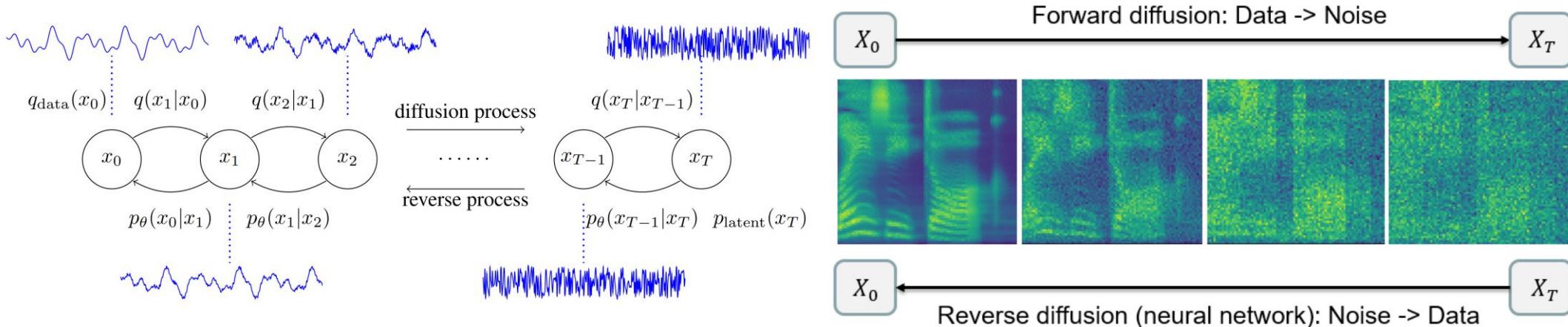


Generative models—Diffusion

- Diffusion probabilistic model

- Forward (diffusion) process: $q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$

- Reverse (denoising) process $p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \quad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$



Generative models—Diffusion

- Loss derived from ELBO: $L_{\text{simple}}(\theta) := \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2 \right]$
- Training and inference process

Algorithm 1 Training

```
for  $i = 1, 2, \dots, N_{\text{iter}}$  do
    Sample  $x_0 \sim q_{\text{data}}$ ,  $\epsilon \sim \mathcal{N}(0, I)$ , and
     $t \sim \text{Uniform}(\{1, \dots, T\})$ 
    Take gradient step on
     $\nabla_\theta \|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t)\|_2^2$ 
    according to Eq. (7)
end for
```

Algorithm 2 Sampling

```
Sample  $x_T \sim p_{\text{latent}} = \mathcal{N}(0, I)$ 
for  $t = T, T - 1, \dots, 1$  do
    Compute  $\mu_\theta(x_t, t)$  and  $\sigma_\theta(x_t, t)$  using Eq. (5)
    Sample  $x_{t-1} \sim p_\theta(x_{t-1}|x_t) =$ 
         $\mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_\theta(x_t, t)^2 I)$ 
end for
return  $x_0$ 
```

Generative models——Diffusion

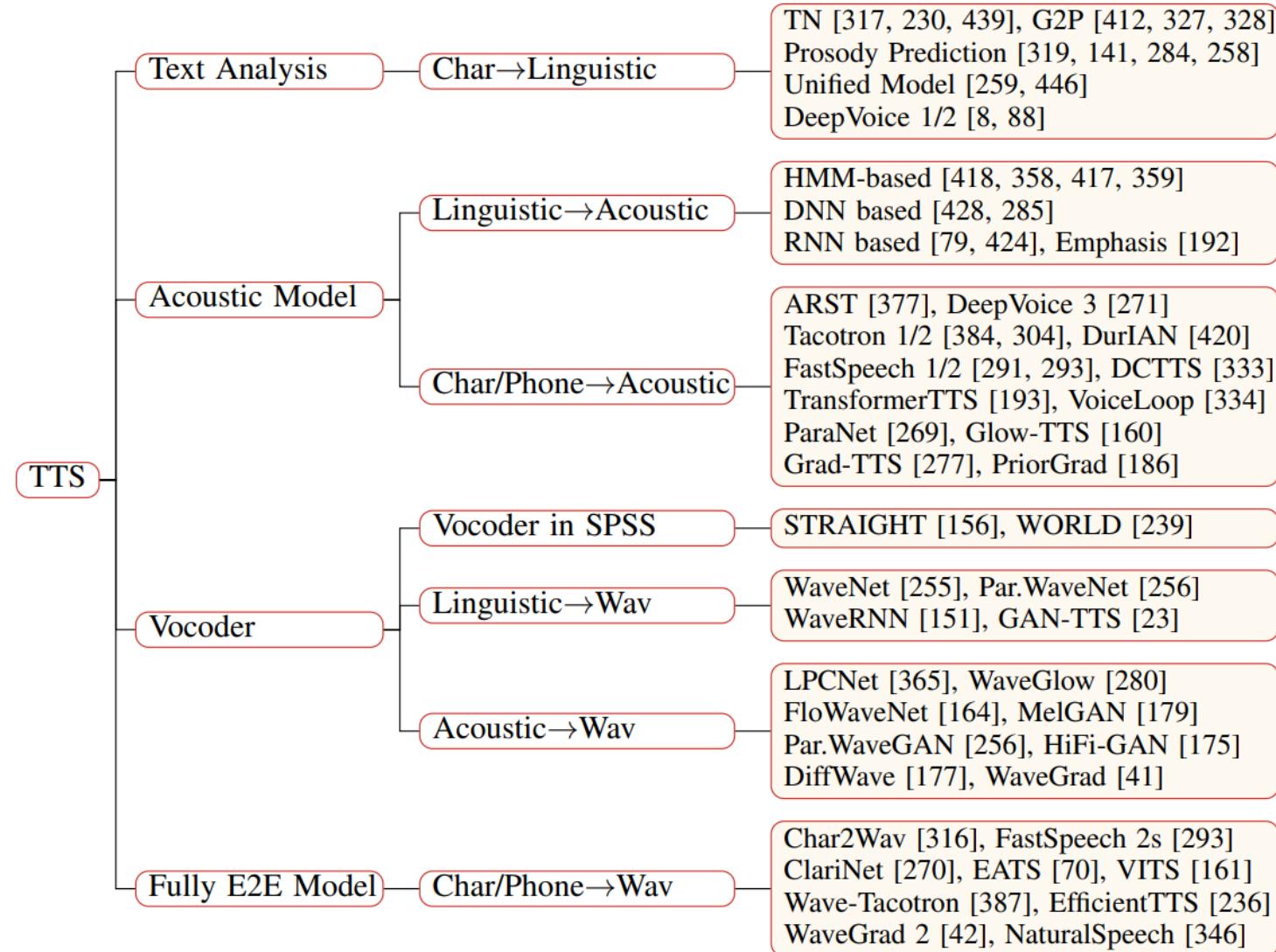
- Diffusion model for vocoder: DiffWave [176], WaveGrad [41]
- Diffusion model for acoustic model: Diff-TTS, Grad-TTS
- Improving diffusion model for TTS
 - PriorGrad, SpecGrad, DiffGAN-TTS, WaveGrad 2, etc
- With sufficient diffusion steps, the quality is good enough, but latency is high
- How to reduce inference cost while maintaining the quality is challenging, and has a long way to go

Generative models—Comparison

- A comparison among different generative models

Generative Model	AR	VAE	Flow/AR	Flow/Bipartite	Diffusion	GAN
Parallel	N	Y	Y	Y	Y	Y
Latent Manipulate	N	Y	Y	Y	Y	Y*
Latent Inference	N	Y	Y	Y	Y	N
Distribution Loss	N	N	Y	Y	Y	Y
Likelihood Estimate	Y	Y	Y	Y	Y	N

Key components in TTS

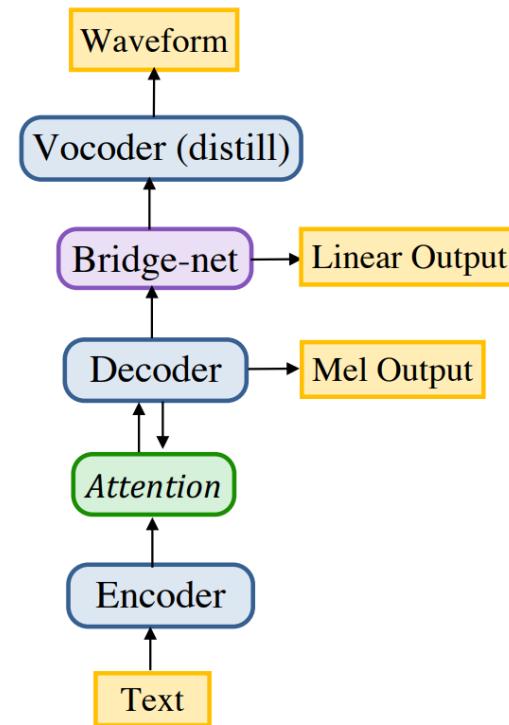


Fully End-to-End TTS

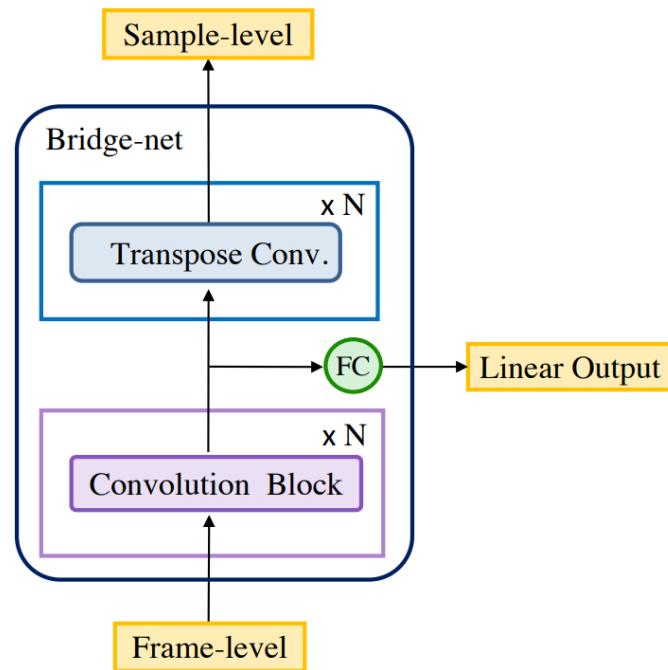
- Direct text/phoneme to waveform generation
- Advantages:
 - Fully differentiable optimization (towards the end goal)
 - Reduce cascaded errors (training/inference mismatch)
 - No mel-spectrogram bias (mel-spectrogram is not an optimal representation)

Fully End-to-End TTS

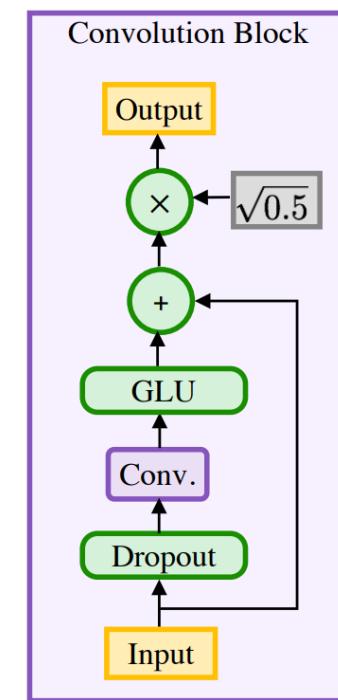
- ClariNet: AR acoustic model and NAR vocoder [269]



(a) Text-to-wave architecture



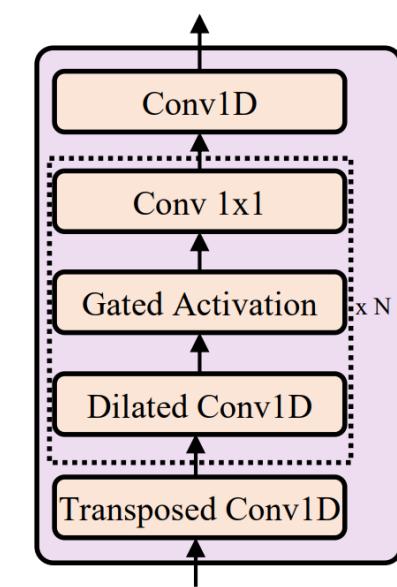
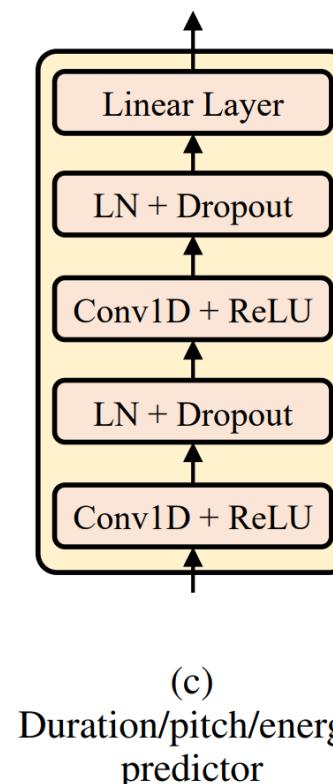
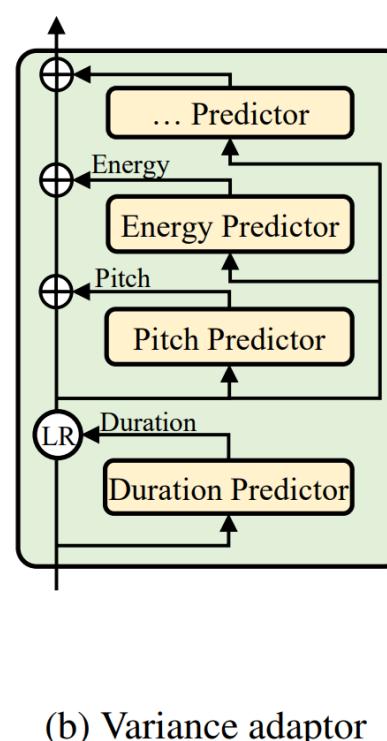
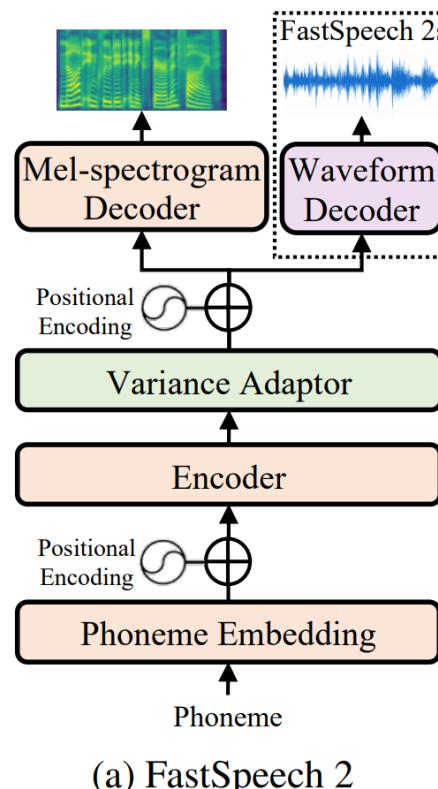
(b) Bridge-net



(c) Convolution block

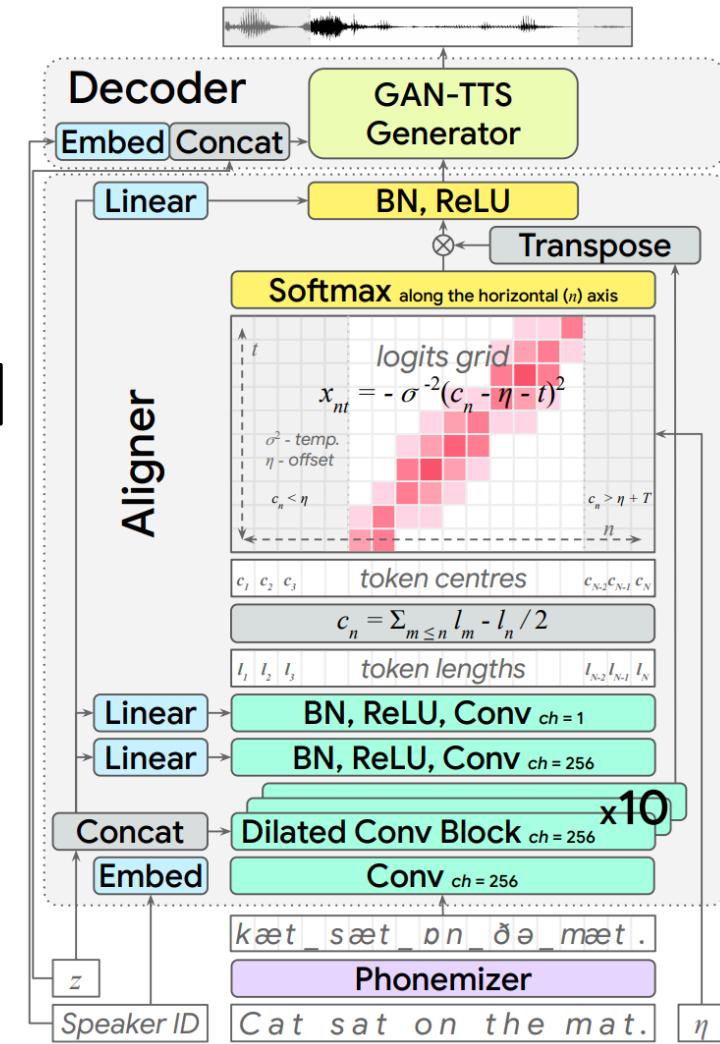
Fully End-to-End TTS

- FastSpeech 2s: fully parallel text to wave model [292]



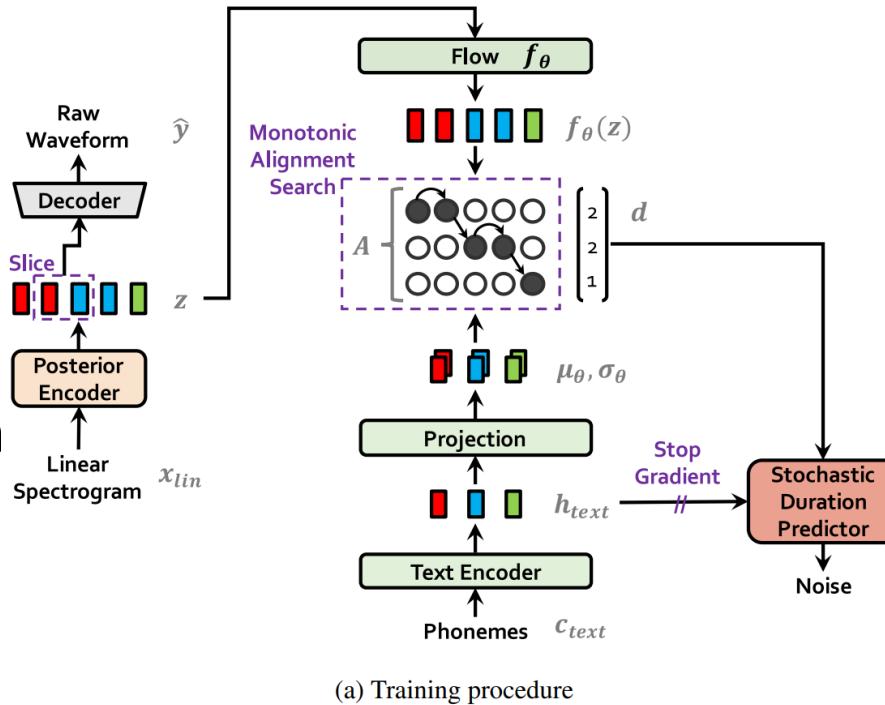
Fully End-to-End TTS

- EATS: fully parallel text to wave model [69]
 - Duration prediction
 - Monotonic interpolation for upsampling
 - Soft dynamic time warping loss
 - Adversarial training

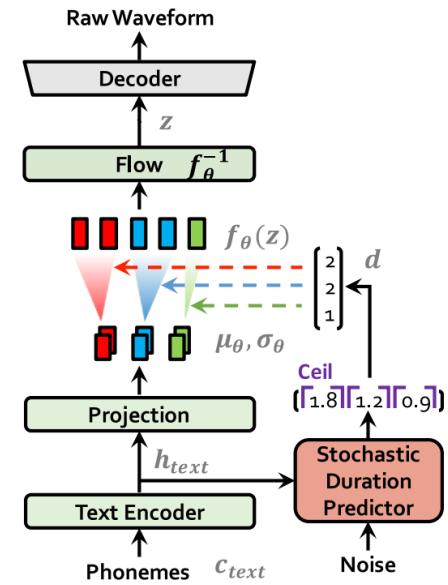


Fully End-to-End TTS

- VITS [160]
 - VAE, Flow, GAN
 - VAE: mel \rightarrow waveform
 - Flow for VAE prior
 - GAN for waveform generation
 - Monotonic alignment search



(a) Training procedure



(b) Inference procedure

Fully End-to-End TTS

- NaturalSpeech: achieving human-level quality on LJSpeech dataset (CMOS)
- Questions
 - 1) how to define human-level quality in TTS?
 - 2) how to judge whether a TTS system has achieved human-level quality or not?
 - 3) how to build a TTS system to achieve human-level quality?
- Define human-level quality
 - *If there is no statistically significant difference between the quality scores of the speech generated by a TTS system and the quality scores of the corresponding human recordings on a test set, then this TTS system achieves human-level quality on this test set.*

Fully End-to-End TTS

- NaturalSpeech: achieving human-level quality on LJSpeech dataset (CMOS)
- Questions
 - 1) how to define human-level quality in TTS?
 - 2) how to judge whether a TTS system has achieved human-level quality or not?
 - 3) how to build a TTS system to achieve human-level quality?
- Judge human-level quality
 - At least 50 utterances, and each judged by 20 judges (native speakers)
 - CMOS → 0, and Wilcoxon signed rank test $p > 0.05$

Fully End-to-End TTS

- NaturalSpeech: achieving human-level quality on LJSpeech dataset (CMOS)
- Questions
 - 1) how to define human-level quality in TTS?
 - 2) how to judge whether a TTS system has achieved human-level quality or not?
 - 3) how to build a TTS system to achieve human-level quality?
- Judge human-level quality

System	MOS	Wilcoxon p-value	CMOS	Wilcoxon p-value
Human Recordings	4.52 ± 0.11	-	0	-
FastSpeech 2 [18] + HiFiGAN [17]	4.32 ± 0.10	1.0e-05	-0.30	5.1e-20
Glow-TTS [13] + HiFiGAN [17]	4.33 ± 0.10	1.3e-06	-0.23	8.7e-17
Grad-TTS [14] + HiFiGAN [17]	4.37 ± 0.10	0.0127	-0.23	1.2e-11
VITS [15]	4.49 ± 0.10	0.2429	-0.19	2.9e-04

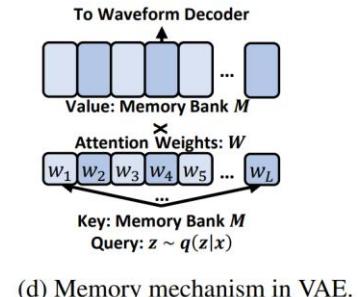
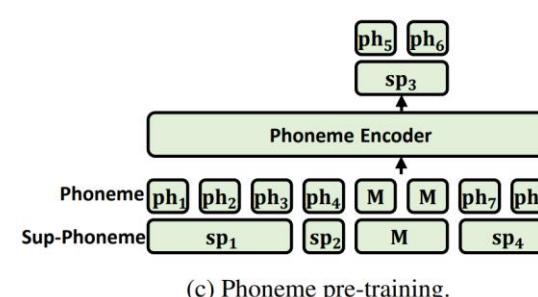
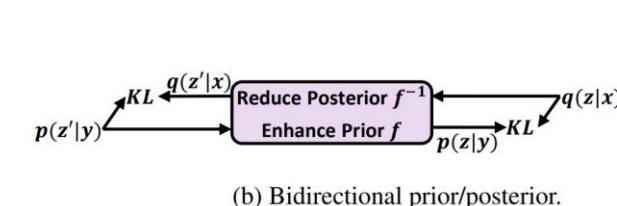
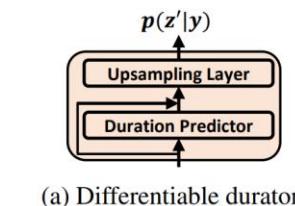
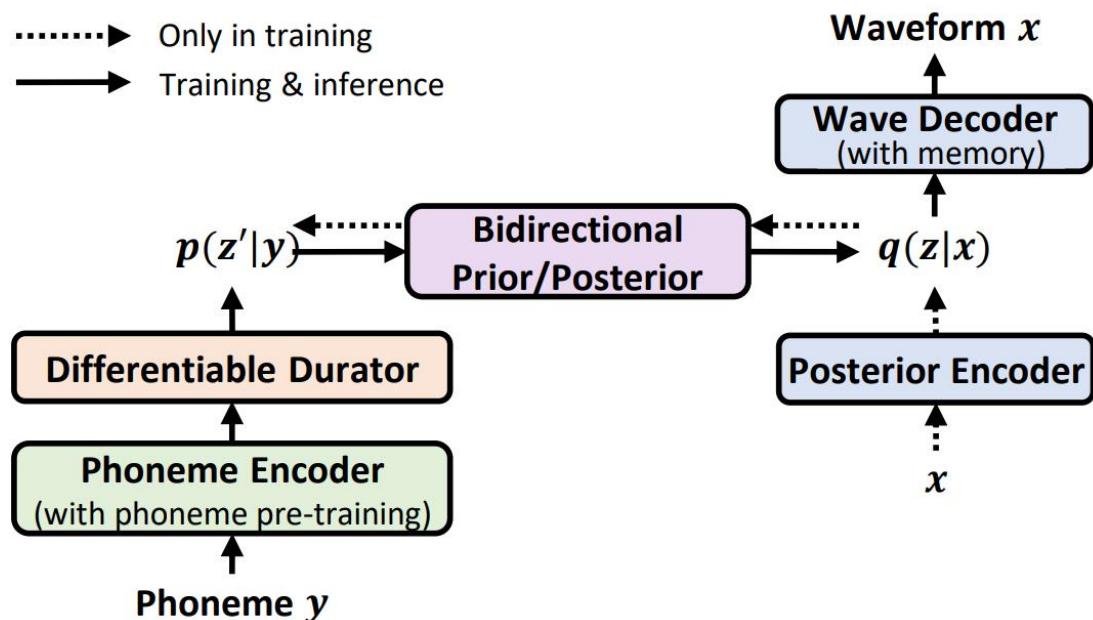
Fully End-to-End TTS

- NaturalSpeech: achieving human-level quality on LJSpeech dataset (CMOS)
- Leverage VAE to compress high-dimensional waveform x into frame-level representations $z \sim q(z|x)$, and is used to reconstruct waveform $x \sim p(x|z)$
- To enable text to waveform synthesis, z is predicted from y , $z \sim p(z|y)$
- However, the posterior $z \sim q(z|x)$ is more complicated than the prior $z \sim p(z|y)$.

Fully End-to-End TTS

- Solutions
 - Phoneme encoder with large-scale phoneme pre-training
 - Differentiable durator
 - Bidirectional prior/posterior
 - Memory based VAE

..... → Only in training
 → Training & inference

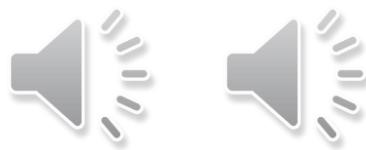


Fully End-to-End TTS

- Evaluations
 - MOS and CMOS on par with recordings, p-value >>0.05

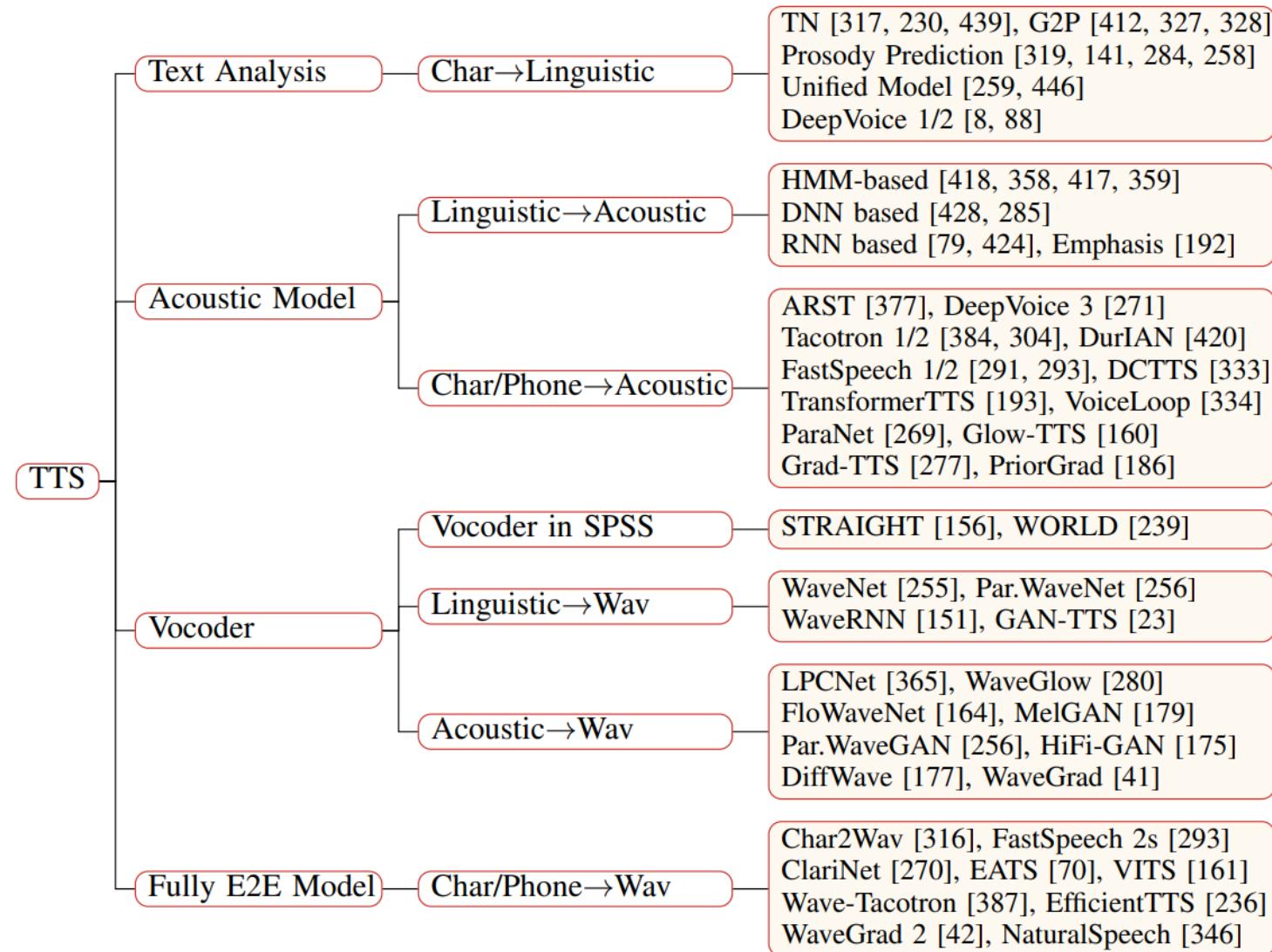
Human Recordings	NaturalSpeech	Wilcoxon p-value
4.58 ± 0.13	4.56 ± 0.13	0.7145

Human Recordings	NaturalSpeech	Wilcoxon p-value
0	-0.01	0.6902



Achieving human-level quality on LJSpeech dataset for the first time!

Key components in TTS

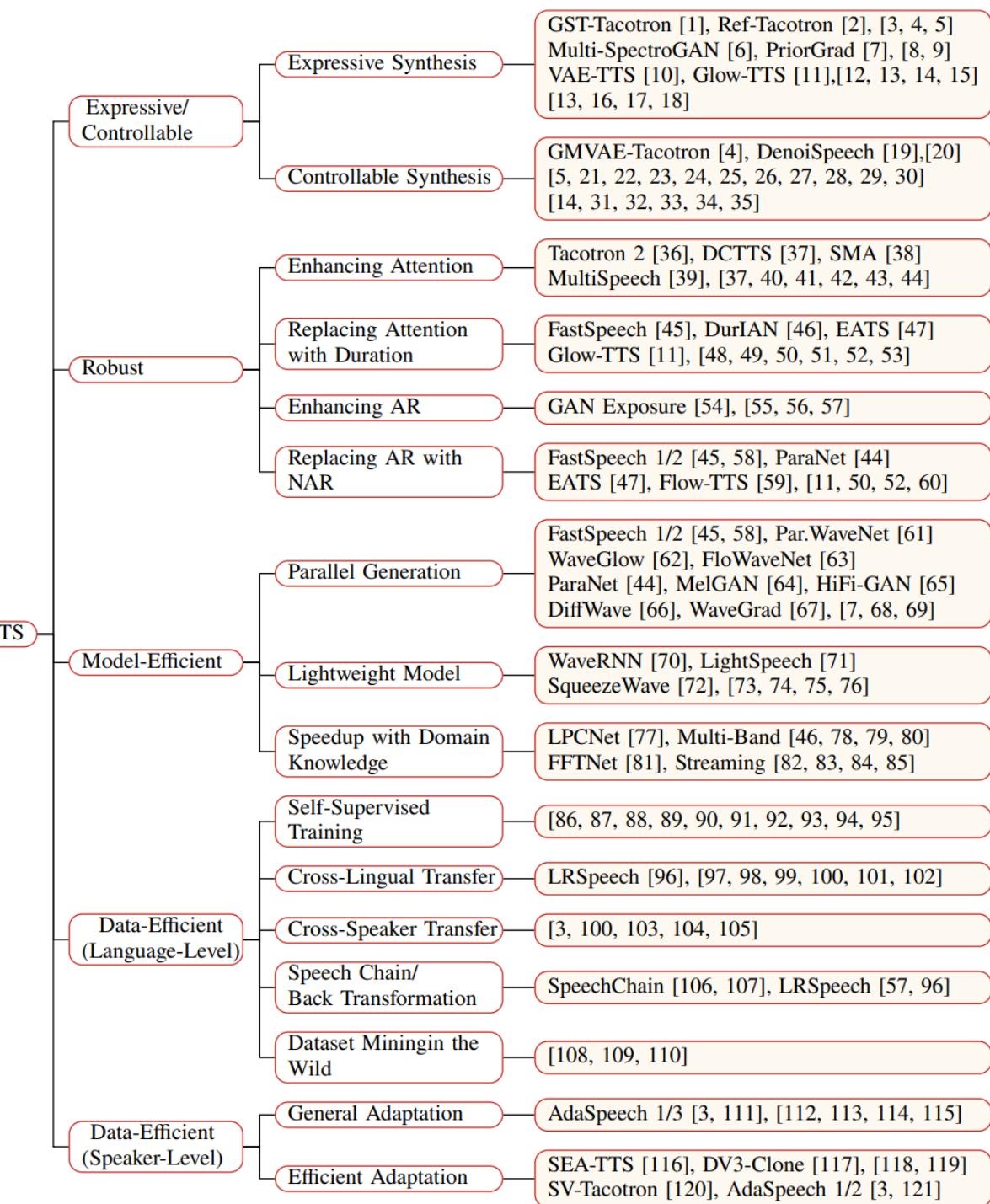


Part 1: Text-to-Speech Synthesis

Part 1.3: Advanced Topics in TTS

Advanced topics in TTS

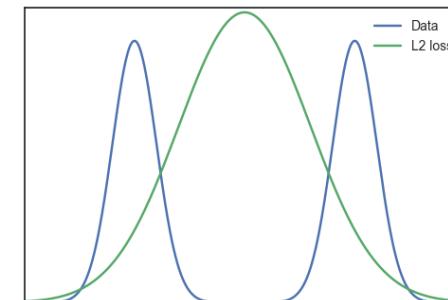
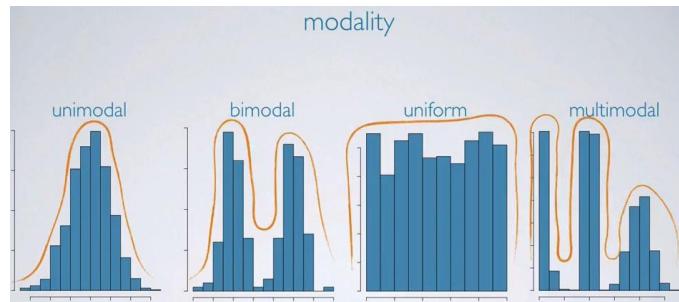
- Expressive/Controllable TTS
- Robust TTS
- Model-Efficient TTS
- Data-Efficient TTS



Expressive TTS

- Expressiveness
 - Characterized by content (what to say), speaker/timbre (who to say), prosody/emotion/style (how to say), noisy environment (where to say), etc
- Over-smoothing prediction
 - One to many mapping in text to speech: $p(y|x)$ multimodal distribution

Text
↓
multiple speech variations
(duration, pitch, sound volume, speaker, style, emotion, etc)



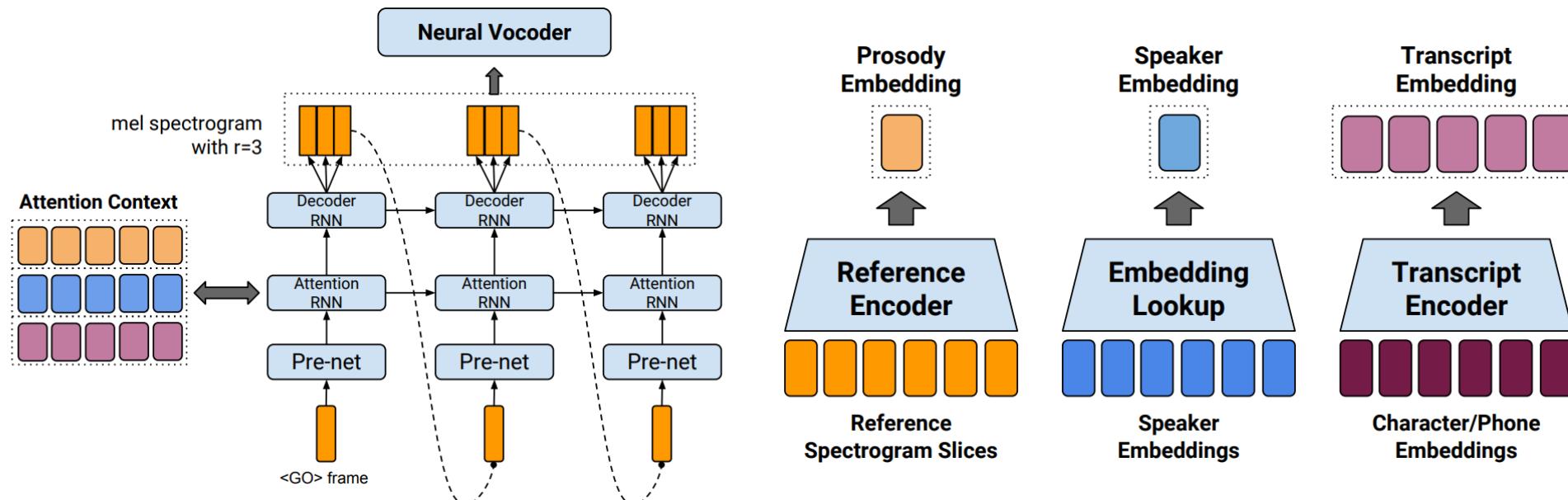
Expressive TTS

- Modeling variation information

Perspective	Category	Description	Work
Information Type	Explicit	Language/Style/Speaker ID	[445, 247, 195, 162, 39]
		Pitch/Duration/Energy	[290, 292, 181, 158, 239, 365]
	Implicit	Reference encoder	[309, 383, 224, 142, 9, 49, 37, 40]
		VAE	[119, 4, 443, 120, 324, 325, 74]
	Text pre-training	GAN/Flow/Diffusion	[224, 186, 366, 234, 159, 141]
		Text pre-training	[81, 104, 393, 143]
Information Granularity	Language/Speaker Level	Multi-lingual/speaker TTS	[445, 247, 39]
	Paragraph Level	Long-form reading	[11, 395, 376]
	Utterance Level	Timbre/Prosody/Noise	[309, 383, 142, 321, 207, 40]
	Word/Syllable Level		[325, 116, 45, 335]
	Character/Phoneme Level	Fine-grained information	[188, 324, 430, 325, 45, 40, 189]
	Frame Level		[188, 158, 49, 434]

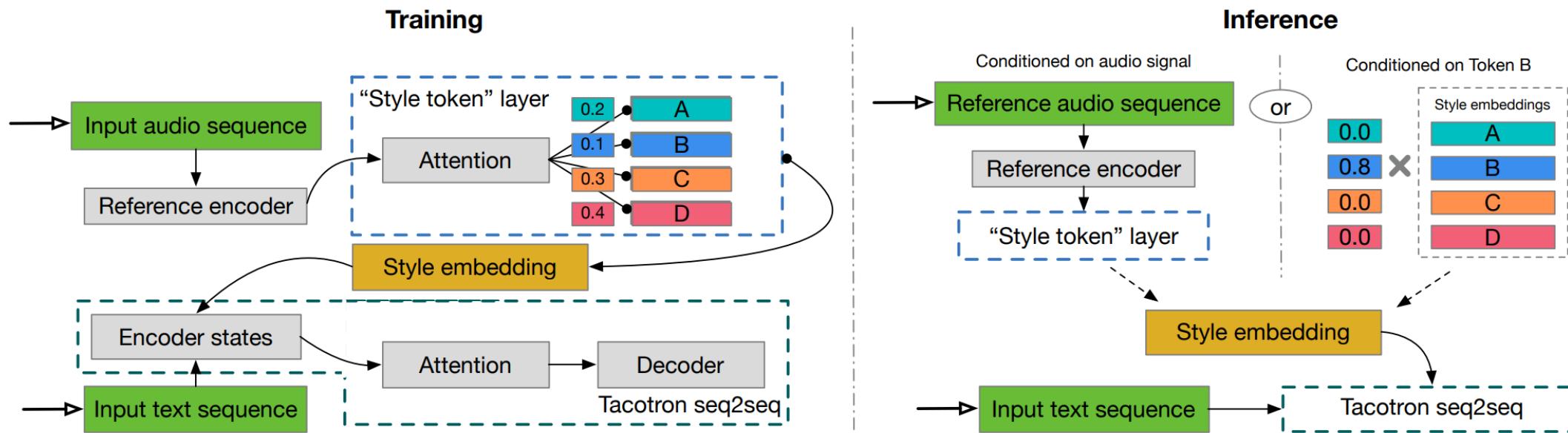
Expressive TTS—Reference encoder

- Prosody embedding from reference audio [309]



Expressive TTS—Reference encoder

- Style tokens [383]
 - Training: attend to style tokens
 - Inference: attend to style tokens or simply pick style tokens



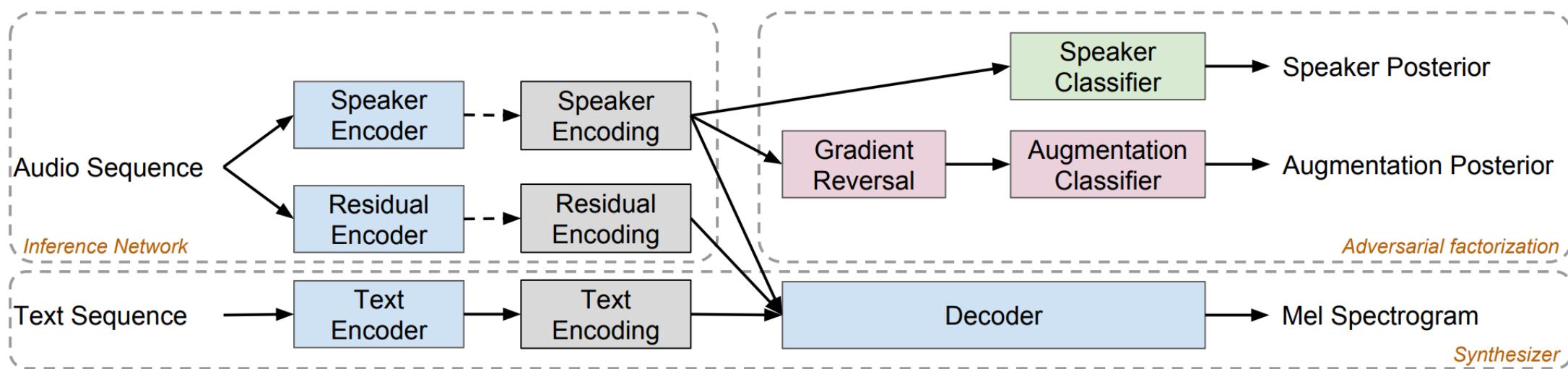
Controllable TTS—Disentangling, Controlling and Transferring

- Disentangling for control
 - Content/speaker/style/noise, e.g., adversarial training, semi-supervised learning
- Improving Controllability
 - Cycle consistency/feedback loss
- Transferring with control
 - Changing variance information for transfer

Technique	Description	Work
Disentangling for Control	Adversarial Training	[5, 19, 20, 21]
	Semi-Supervised Learning	[4, 5, 14, 19, 163]
Improving Controllability	Cycle Consistency/Feedback Loss	[31, 32, 33, 34, 35]
Transferring with Control	Changing Variance Information in Inference	[1, 2, 3, 10, 120]

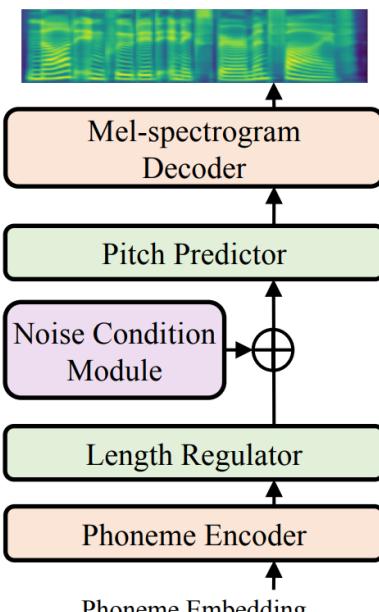
Controllable TTS—Disentangling, Controlling and Transferring

- Disentangling correlated speaker and noise [120]
 - Synthesize clean speech for noisy speakers

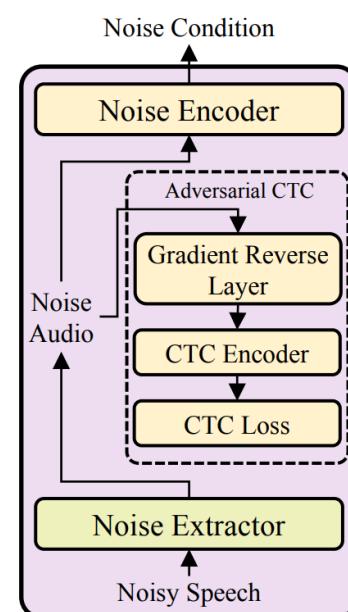


Controllable TTS—Disentangling, Controlling and Transferring

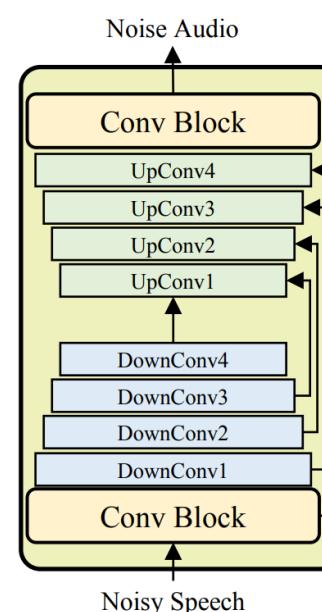
- Disentangling correlated speaker and noise with frame-level modeling [434]
 - Synthesize clean speech for noisy speakers



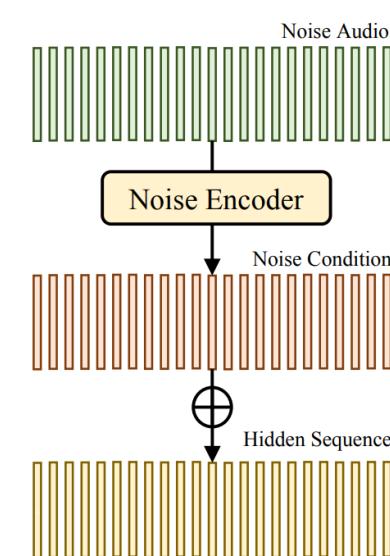
(a) DenoSpeech



(b) Noise Condition Module



(c) Noise Extractor



(d) Noise Encoder

Robust TTS

- Robustness issues
 - Word skipping, repeating, attention collapse

You can call me directly at 4257037344 or my cell 4254447474 or send me a meeting request with all the appropriate information.



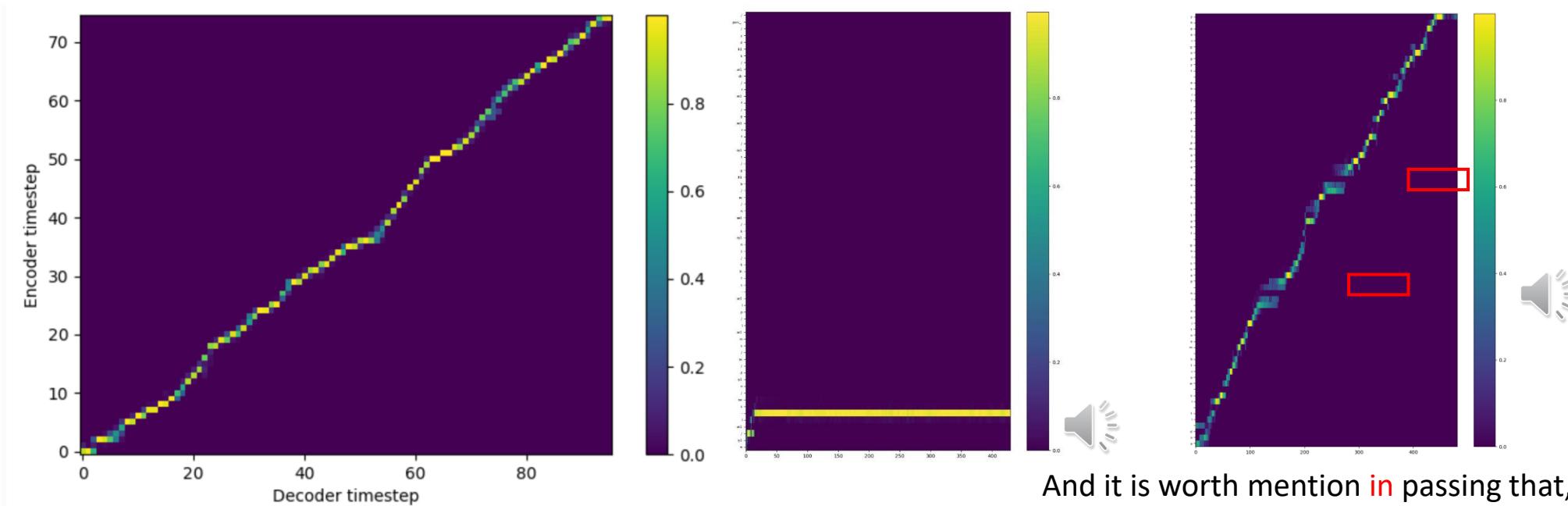
- The cause of robustness issues
 - The difficulty of alignment learning between text and mel-spectrograms
 - Exposure bias and error propagation in AR generation
- The solutions
 - Enhance attention
 - Replace attention with duration prediction
 - Enhance AR
 - Replace AR with NAR

Robust TTS

Category	Technique	Work
Enhancing Attention	Content-based attention	[382, 192]
	Location-based attention	[315, 333, 367, 17]
	Content/Location hybrid attention	[303]
	Monotonic attention	[438, 107, 411]
	Windowing or off-diagonal penalty	[332, 438, 270, 39]
	Enhancing enc-dec connection	[382, 303, 270, 203, 39]
	Positional attention	[268, 234, 204]
Replacing Attention with Duration Prediction	Label from encoder-decoder attention	[290, 361, 197, 181]
	Label from CTC alignment	[19]
	Label from HMM alignment	[292, 418, 194, 252, 74, 304]
	Dynamic programming	[429, 193, 235]
	Monotonic alignment search	[159]
	Monotonic interpolation with soft DTW	[69, 75]
Enhancing AR	Professor forcing	[99, 205]
	Reducing training/inference gap	[361]
	Knowledge distillation	[209]
	Bidirectional regularization	[291, 452]
Replacing AR with NAR	Parallel generation	[290, 292, 268, 69]

Robust TTS——Attention improvement

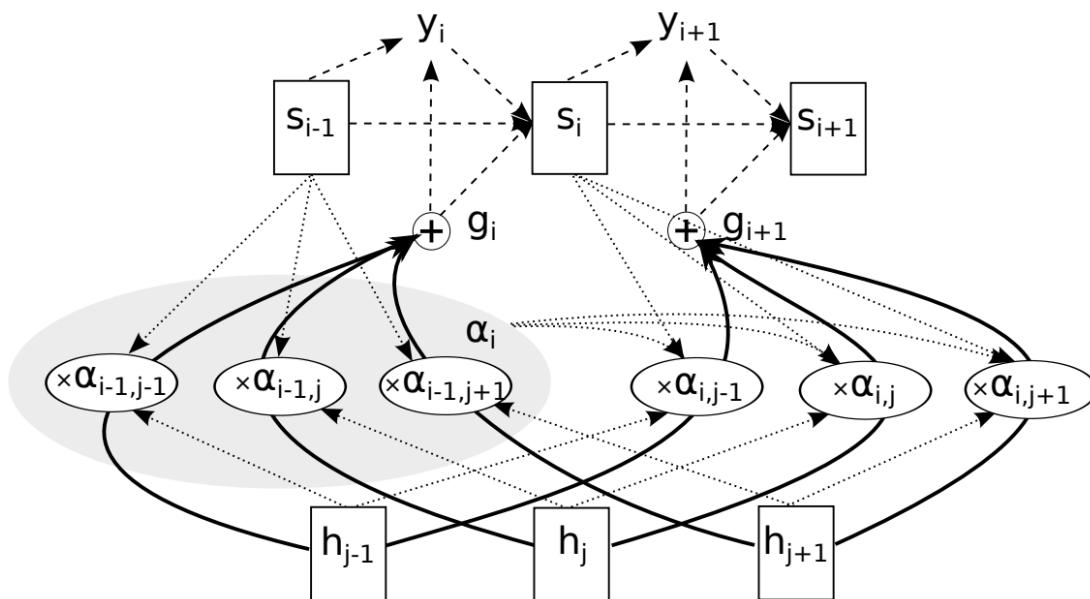
- Encoder-decoder attention: alignment between text and mel
 - Local, monotonic, and complete



And it is worth mention **in** passing that,
as an example of fine typography

Robust TTS——Attention improvement

- Location sensitive attention [50, 303]
 - Use previous alignment to compute the next attention alignment



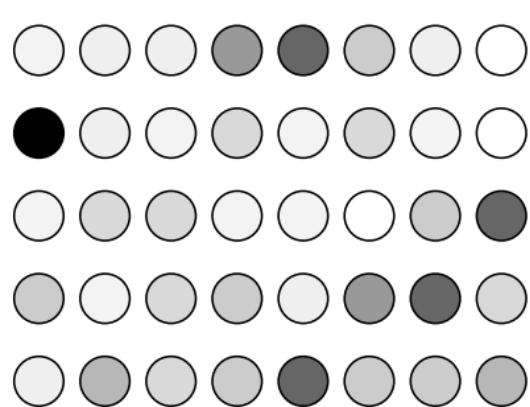
$$\alpha_i = \text{Attend}(s_{i-1}, \alpha_{i-1}, h)$$

$$g_i = \sum_{j=1}^L \alpha_{i,j} h_j$$

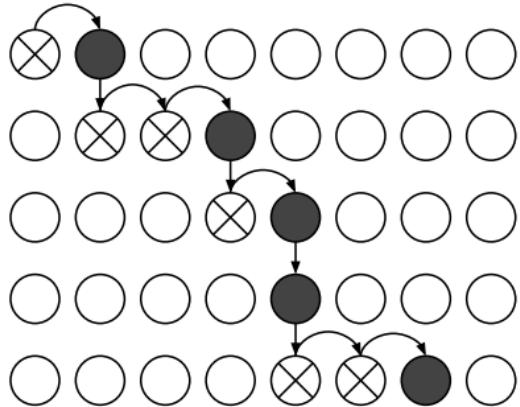
$$y_i \sim \text{Generate}(s_{i-1}, g_i),$$

Robust TTS——Attention improvement

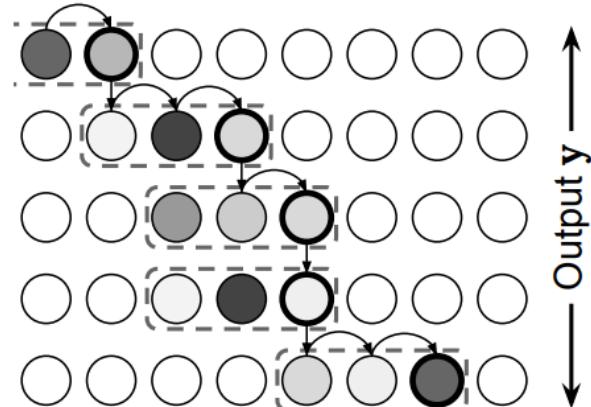
- Monotonic attention [288, 47]
 - The attention position is monotonically increasing



(a) Soft attention.



(b) Hard monotonic attention.



(c) Monotonic chunkwise attention.

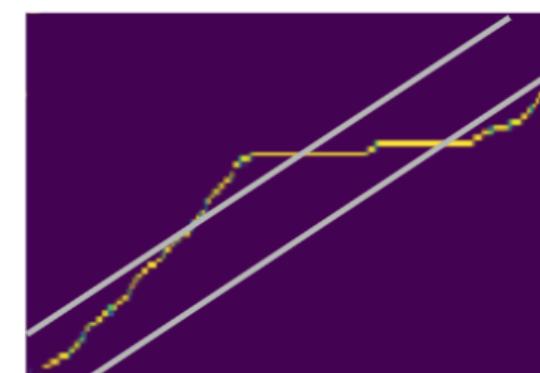
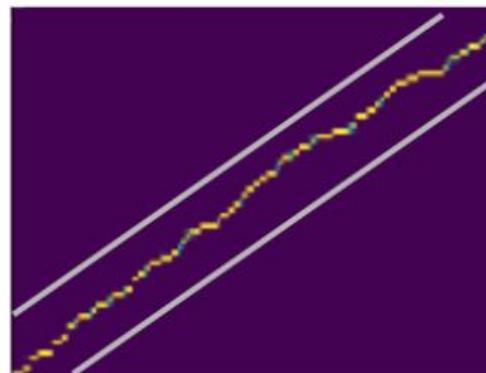
$$e_{i,j} = \text{MonotonicEnergy}(s_{i-1}, h_j)$$

$$p_{i,j} = \sigma(e_{i,j})$$

$$z_{i,j} \sim \text{Bernoulli}(p_{i,j})$$

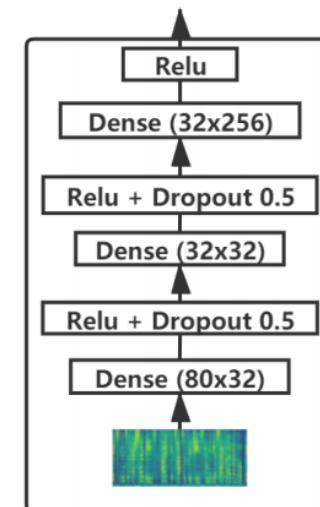
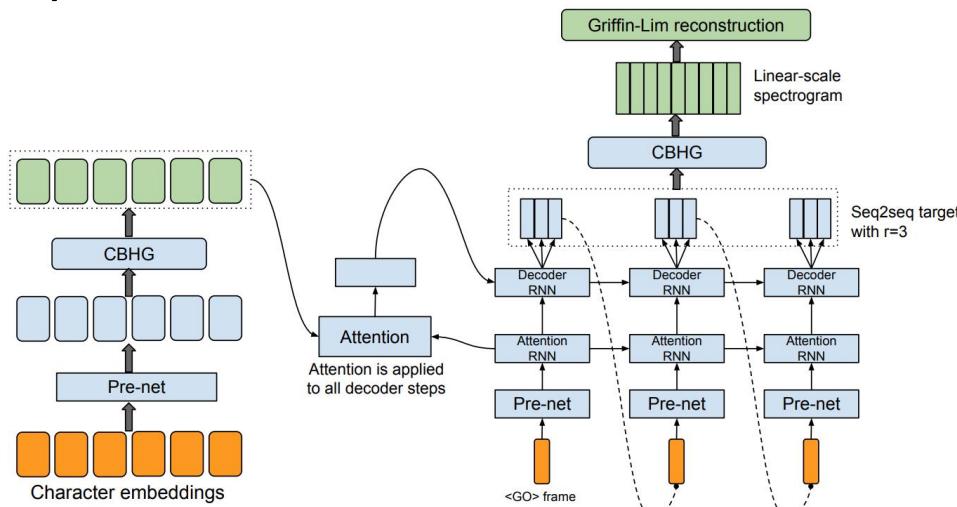
Robust TTS—Attention improvement

- Windowing [332, 438]
 - Only a subset of the encoding results $\hat{\mathbf{x}} = [\mathbf{x}_{p-w}, \dots, \mathbf{x}_{p+w}]$ are considered at each decoder timestep when using the windowing technique
- Penalty loss for off-diagonal attention distribution [39]
 - Guided attention loss with diagonal band mask



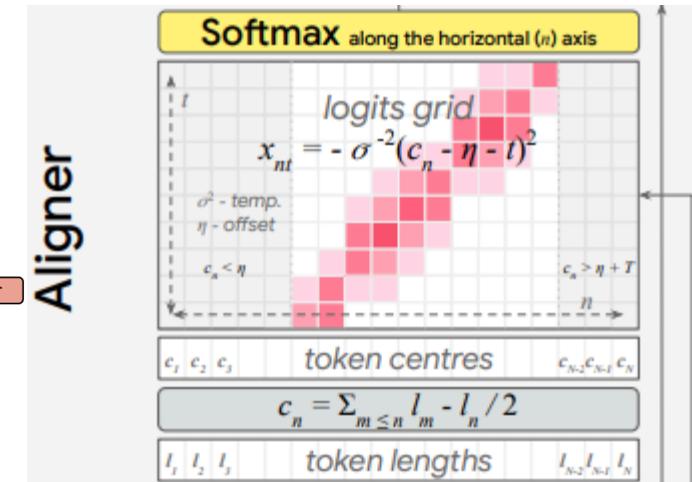
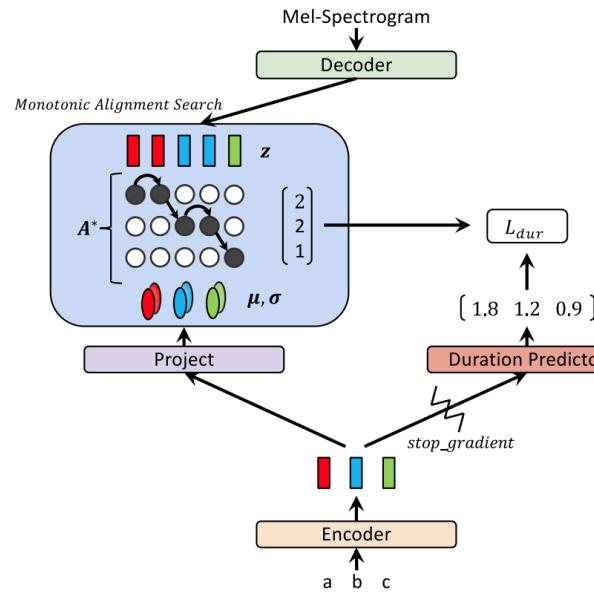
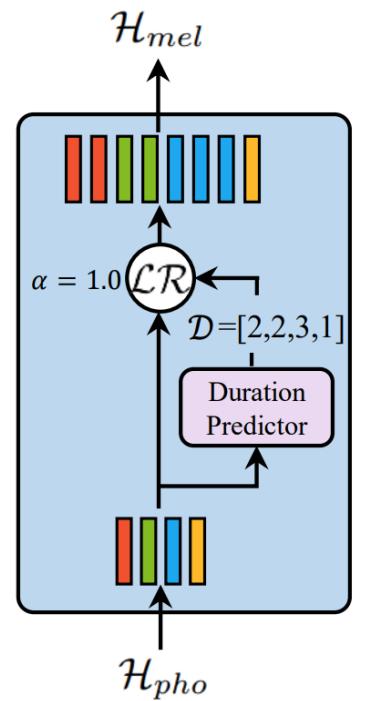
Robust TTS—Attention improvement

- Multi-frame prediction [382]
 - Predicting multiple, non-overlapping output frames at each decoder step
 - Increase convergence speed, with a much faster (and more stable) alignment learned from attention
- Decoder prenet dropout/bottleneck [382,39]
 - 0.5 dropout, small hidden size as bottleneck



Robust TTS——Durator

- Duration prediction and expansion
 - SPSS → Seq2Seq model with attention → Non-autoregressive model
 - Duration → attention, no duration → duration prediction (technique renaissance)



Robust TTS——Durator

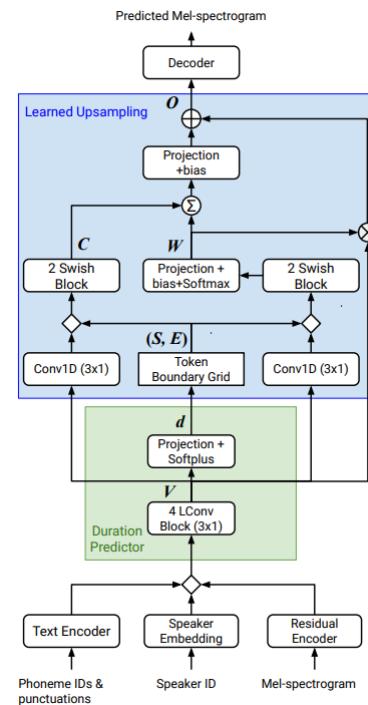
- Differentiable duration modeling

$$S_{i,j} = i - \sum_{k=1}^{j-1} d_k, \quad E_{i,j} = \sum_{k=1}^j d_k - i, \quad S_{m \times n} \quad E_{m \times n}$$

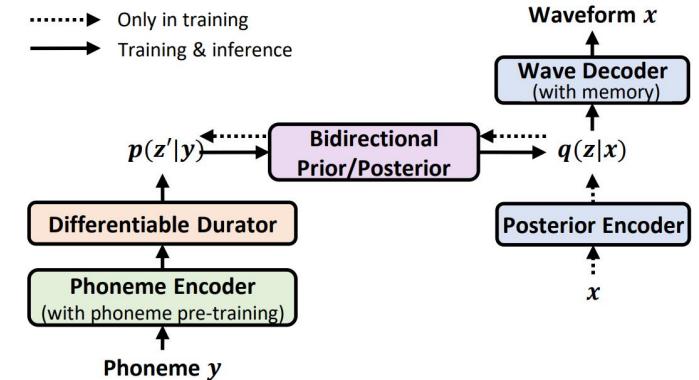
$$\mathbf{W} = \text{Softmax}_{10 \rightarrow q}(\text{MLP}([\mathbf{S}, \mathbf{E}, \text{Expand}(\text{Conv1D}(\text{Proj}(\mathbf{H})))])),$$

$$\mathbf{C} = \text{MLP}_{10 \rightarrow p}([\mathbf{S}, \mathbf{E}, \text{Expand}(\text{Conv1D}(\text{Proj}(\mathbf{H})))]),$$

$$\mathbf{O} = \text{Proj}_{qh \rightarrow h}(\mathbf{W}\mathbf{H}) + \text{Proj}_{qp \rightarrow h}(\text{Einsum}(\mathbf{W}, \mathbf{C}))$$



Parallel Tacotron 2



NaturalSpeech

Robust TTS

- A new taxonomy of TTS

AR?		AR	Non-AR
Attention?	Attention	Tacotron 2 [303], DeepVoice 3 [270]	ParaNet [268], Flow-TTS [234]
	Non-Attention	DurIAN [418], Non-Att Tacotron [304]	FastSpeech [290, 292], EATS [69]

Model-Efficient TTS

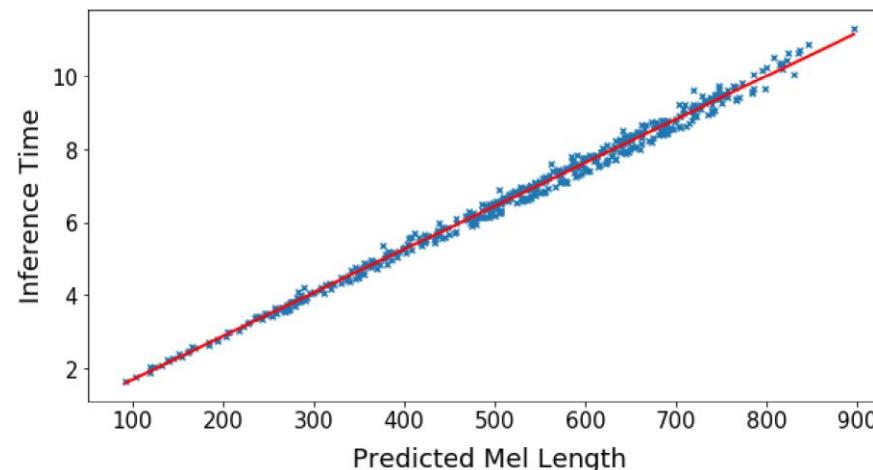
- Fast synthesis speed, small memory storage, and low computation cost
- Parallel generation
 - Increase the parallelism of computation and improve inference/training speed

Modeling Paradigm	TTS Model	Training	Inference
AR (RNN)	Tacotron 1/2, SampleRNN, LPCNet	$\mathcal{O}(N)$	$\mathcal{O}(N)$
AR (CNN/Self-Att)	DeepVoice 3, TransformerTTS, WaveNet	$\mathcal{O}(1)$	$\mathcal{O}(N)$
NAR (CNN/Self-Att)	FastSpeech 1/2, ParaNet	$\mathcal{O}(1)$	$\mathcal{O}(1)$
NAR (GAN/VAE)	MelGAN, HiFi-GAN, FastSpeech 2s, EATS	$\mathcal{O}(1)$	$\mathcal{O}(1)$
Flow (AR)	Par. WaveNet, ClariNet, Flowtron	$\mathcal{O}(1)$	$\mathcal{O}(1)$
Flow (Bipartite)	WaveGlow, FloWaveNet, Glow-TTS	$\mathcal{O}(T)$	$\mathcal{O}(T)$
Diffusion	DiffWave, WaveGrad, Grad-TTS, PriorGrad	$\mathcal{O}(T)$	$\mathcal{O}(T)$

- Lightweight modeling
 - Small model size, low computation, and fast inference speed
 - Pruning, quantization, knowledge distillation, and neural architecture search
- Efficient modeling with domain knowledge
 - linear prediction, multiband modeling, subscale prediction, multi-frame prediction, streaming synthesis

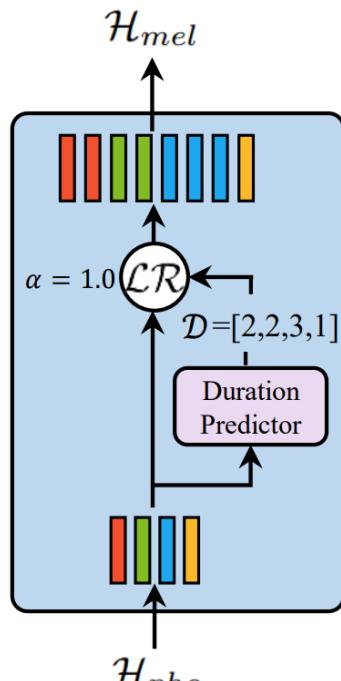
Model-Efficient TTS — Parallel Generation

- The model usually adopts autoregressive mel and waveform generation
 - Sequence is very long, e.g., 1s speech, 100 mel, 24000 waveform points
 - Slow inference speed

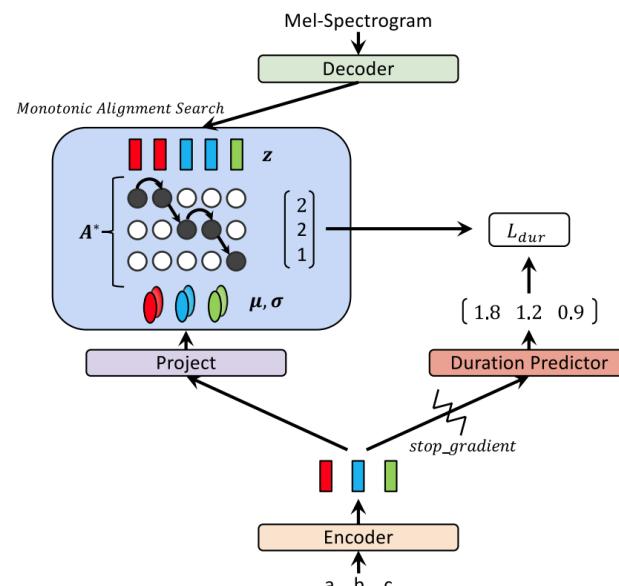


Model-Efficient TTS — Parallel Generation

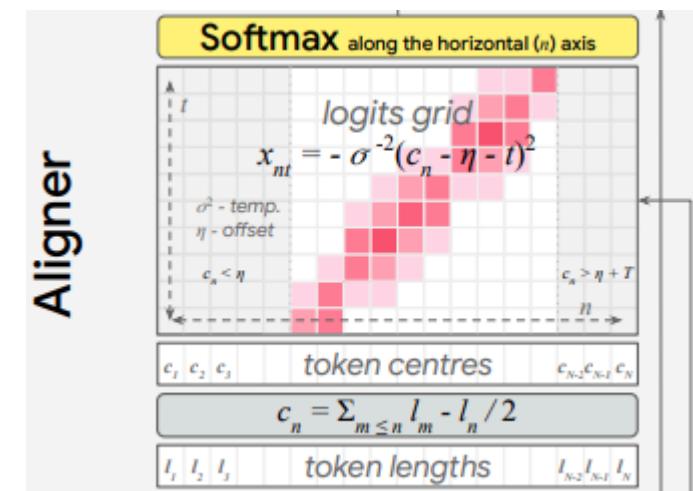
- The key is to bridge the length mismatch between text and speech



FastSpeech 1/2



Glow-TTS



EATS

Model-Efficient TTS — Parallel Generation

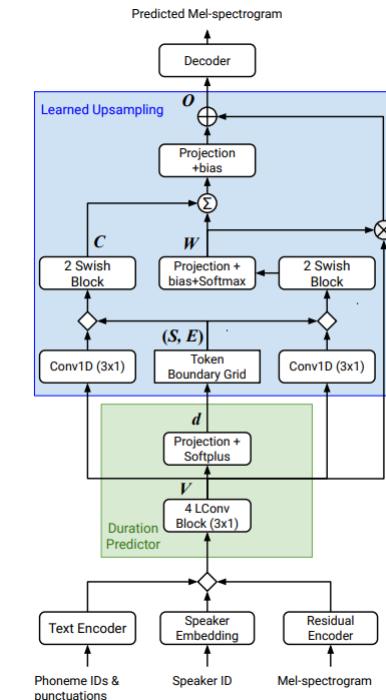
- The key is to bridge the length mismatch between text and speech

$$S_{i,j} = i - \sum_{k=1}^{j-1} d_k, \quad E_{i,j} = \sum_{k=1}^j d_k - i, \quad S_{m \times n} \quad E_{m \times n}$$

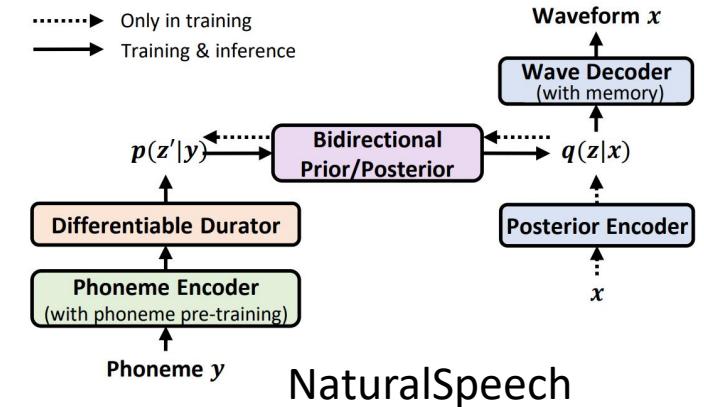
$$\mathbf{W} = \text{Softmax}(\underset{10 \rightarrow q}{\text{MLP}}([\mathbf{S}, \mathbf{E}, \text{Expand}(\text{Conv1D}(\text{Proj}(\mathbf{H})))])),$$

$$\mathbf{C} = \underset{10 \rightarrow p}{\text{MLP}}([\mathbf{S}, \mathbf{E}, \text{Expand}(\text{Conv1D}(\text{Proj}(\mathbf{H})))]),$$

$$\mathbf{O} = \underset{qh \rightarrow h}{\text{Proj}}(\mathbf{W}\mathbf{H}) + \underset{qp \rightarrow h}{\text{Proj}}(\text{Einsum}(\mathbf{W}, \mathbf{C}))$$



Parallel Tacotron 2



Data-Efficient TTS

- Language level: TTS for every language
 - There are **7,000+** languages in the world, but popular commercialized speech services only support **dozens or hundreds** of languages
 - However, lack of data in low-resource languages and the data collection cost is high.
- Speaker level: TTS for everyone
 - 1) Pre-training on multi-speaker TTS model; 2) Fine-tuning on speech data from target speaker; 3) Inference speech for target speaker



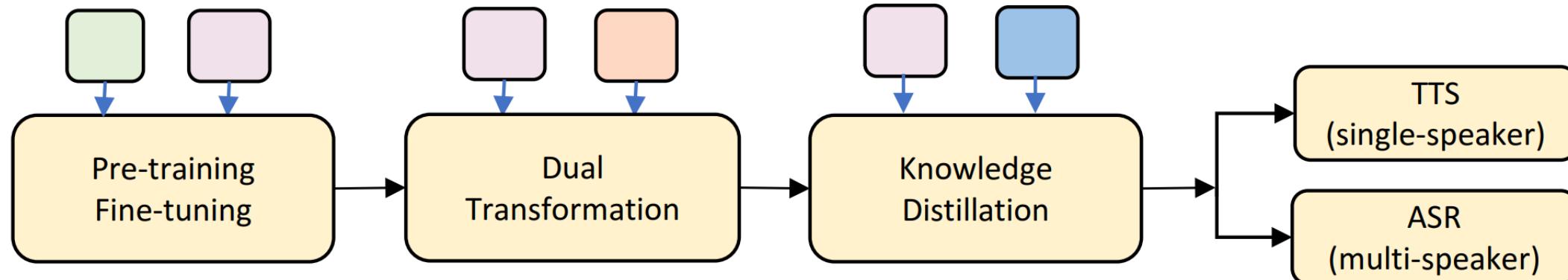
Language-Level Data-Efficient TTS

Techniques	Data	Work
Self-Supervised Training	Unpaired text or speech	[1, 2, 3, 4, 5, 6, 7, 8, 9]
Cross-Lingual Transfer	Paired text and speech	[10, 11, 12, 13, 14, 15, 16]
Semi-Supervised Training	Unpaired text or speech	[11, 17, 18, 19]
Dataset Mining in the Wild	Paired text and speech	[20, 21, 22]
Purely Unsupervised Learning	Unpaired text and speech	[23, 24, 25]

- Self-supervised training
 - Text pre-training, speech pre-training, discrete token quantization
- Cross-lingual transfer
 - Languages share similarity, phoneme mapping/re-initialization/IPA/byte
- Semi-supervised training
 - Speech chain/back transformation (TTS \leftrightarrow ASR)
- Dataset mining in the wild
 - Speech enhancement, denoising, disentangling
- Purely unsupervised learning

Language-Level Data-Efficient TTS—LRSpeech [396]

- Paired data from rich-resource language
- Paired data (single-speaker high, multi-speaker low)
- Unpaired data (multi-speaker low)
- Synthesized data



- **Step 1:** Language transfer
 - Human languages share similar pronunciations; Rich-resource language data is “free”
- **Step 2:** TTS and ASR help with each other
 - Leverage the task duality with unpaired speech and text data
- **Step 3:** Customization for product deployment with knowledge distillation
 - Better accuracy by data knowledge distillation
 - Customize multi-speaker TTS to a target-speaker TTS, and to small model

Speaker-Level Data-Efficient TTS

- Voice adaptation, voice cloning, custom voice
- Challenges
 - To support diverse customers, the source model needs to be generalizable enough, the target speech may be diverse (different acoustics/styles/languages)
 - To support many customers, the adaptation needs to be data and parameter efficient

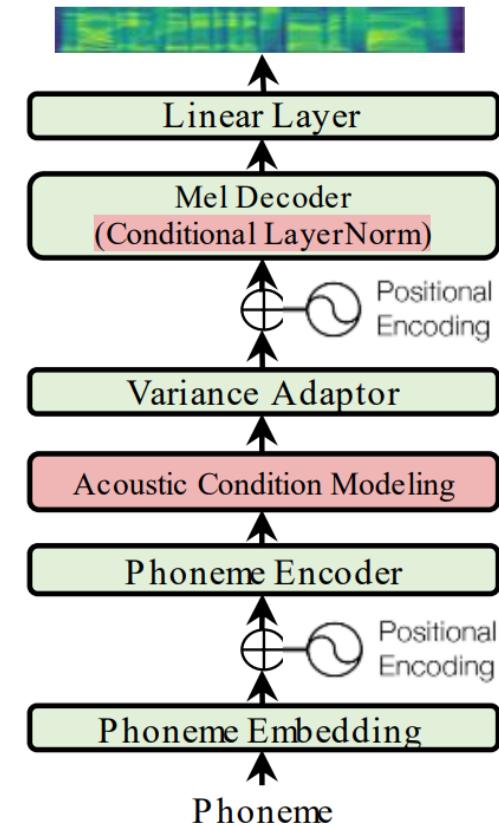
Speaker-Level Data-Efficient TTS

- A taxonomy on adaptive TTS

Category	Topic	Work
Improving Generalization	Modeling Variation Information	[43]
	Increasing Data Coverage	[13, 55]
Cross-Domain Adaption	Cross-Acoustic Adaptation	[43, 56]
	Cross-Style Adaptation	[57, 58, 59]
	Cross-Lingual Adaptation	[60, 61, 62]
Few-Data Adaption	Transcribed Data Adaptation	[41, 42, 43, 63, 64, 65, 66, 67]
	Untranscribed Data Adaptation	[68, 69, 70]
Few-Parameter Adaptation	-	[41, 42, 43]
Zero-Shot Adaptation	-	[41, 42, 71, 72]

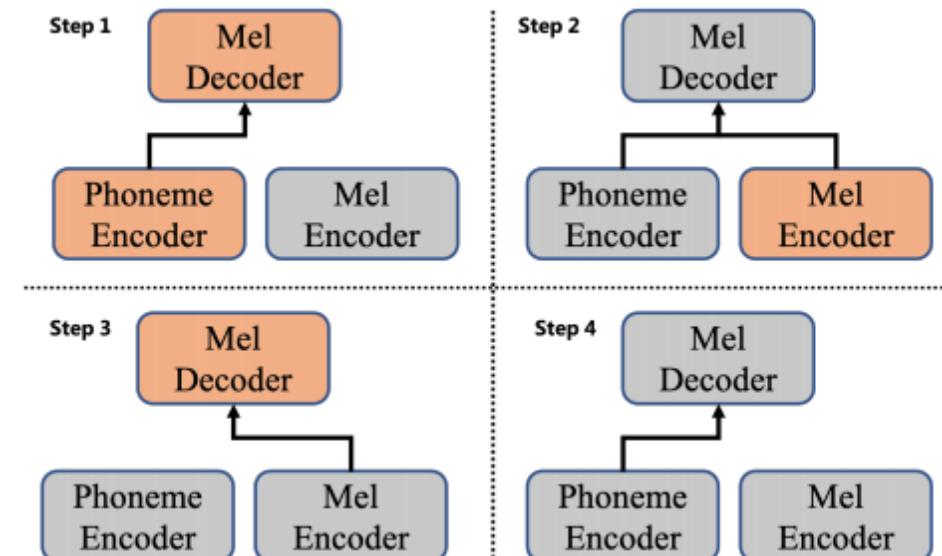
Speaker-Level Data-Efficient TTS—AdaSpeech [40]

- AdaSpeech
 - Acoustic condition modeling
 - Model diverse acoustic conditions at speaker/utterance /phoneme level
 - Support diverse conditions in target speaker
 - Conditional layer normalization
 - To fine-tune as small parameters as possible while ensuring the adaptation quality



Speaker-Level Data-Efficient TTS—AdaSpeech 2 [403]

- Only untranscribed data, how to adapt?
 - In online meeting, only speech can be collected, without corresponding transcripts
- AdaSpeech 2, speech reconstruction with latent alignment
 - Step 1: source TTS model training
 - Step 2: speech reconstruction
 - Step 3: speaker adaptation
 - Step 4: inference

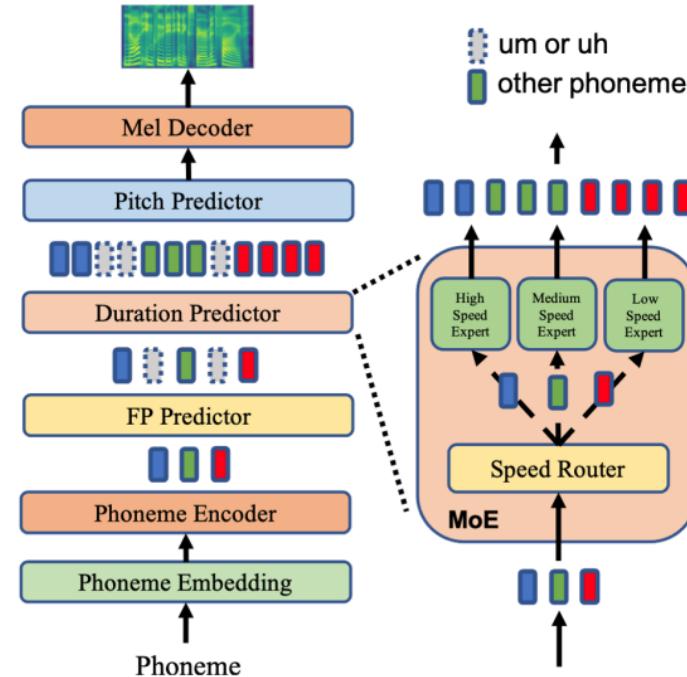


Speaker-Level Data-Efficient TTS—AdaSpeech 3 [404]

- Spontaneous style
 - Current TTS voices mostly focus on reading style.
 - Spontaneous-style voice is useful for scenarios like podcast, conversation, etc.
- AdaSpeech 3
 - Construct spontaneous dataset
 - Modeling filled pauses (FP, um and uh) and diverse rhythms

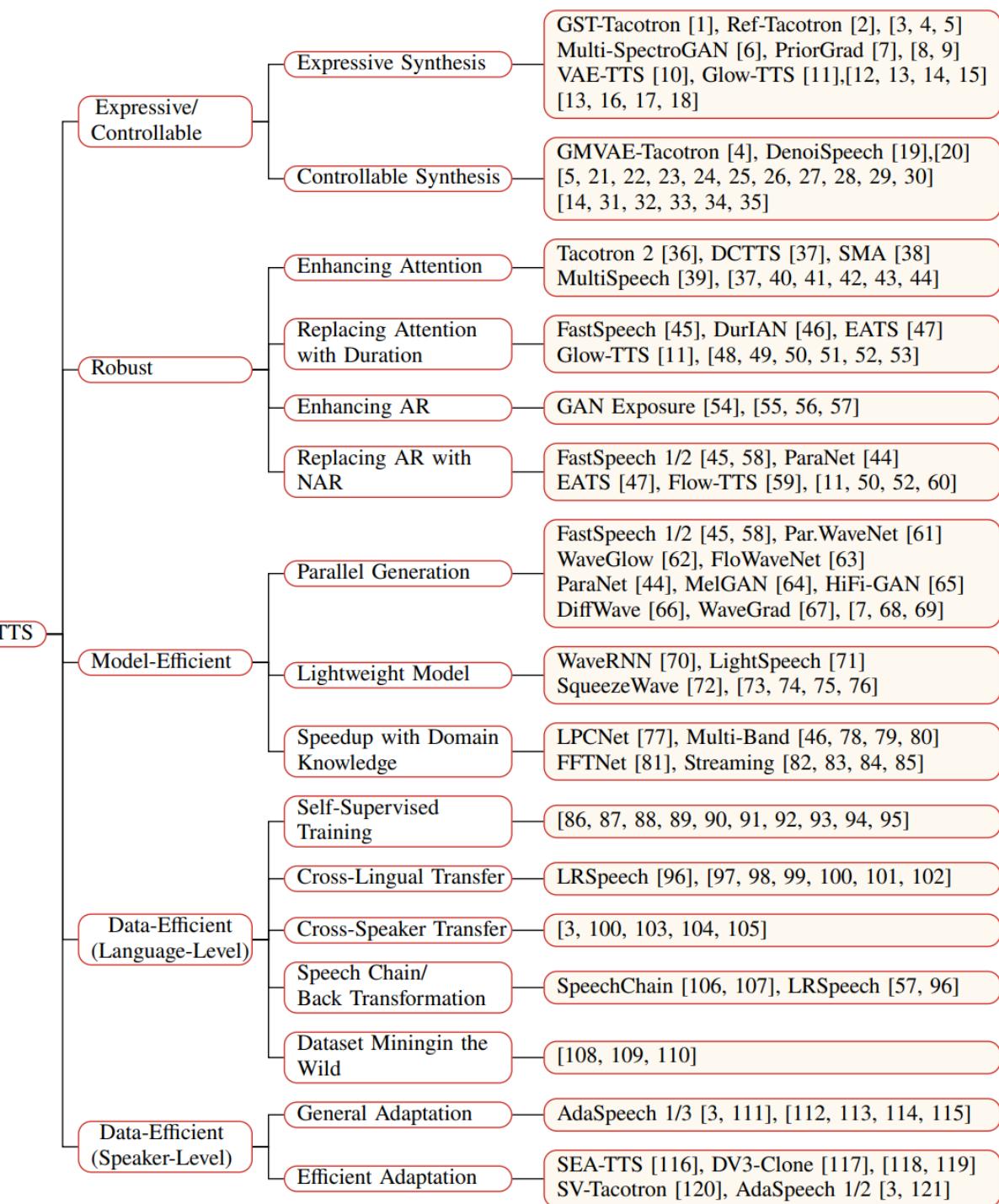


*Cecily package in all of that **um yeah** so ...*



Advanced topics in TTS

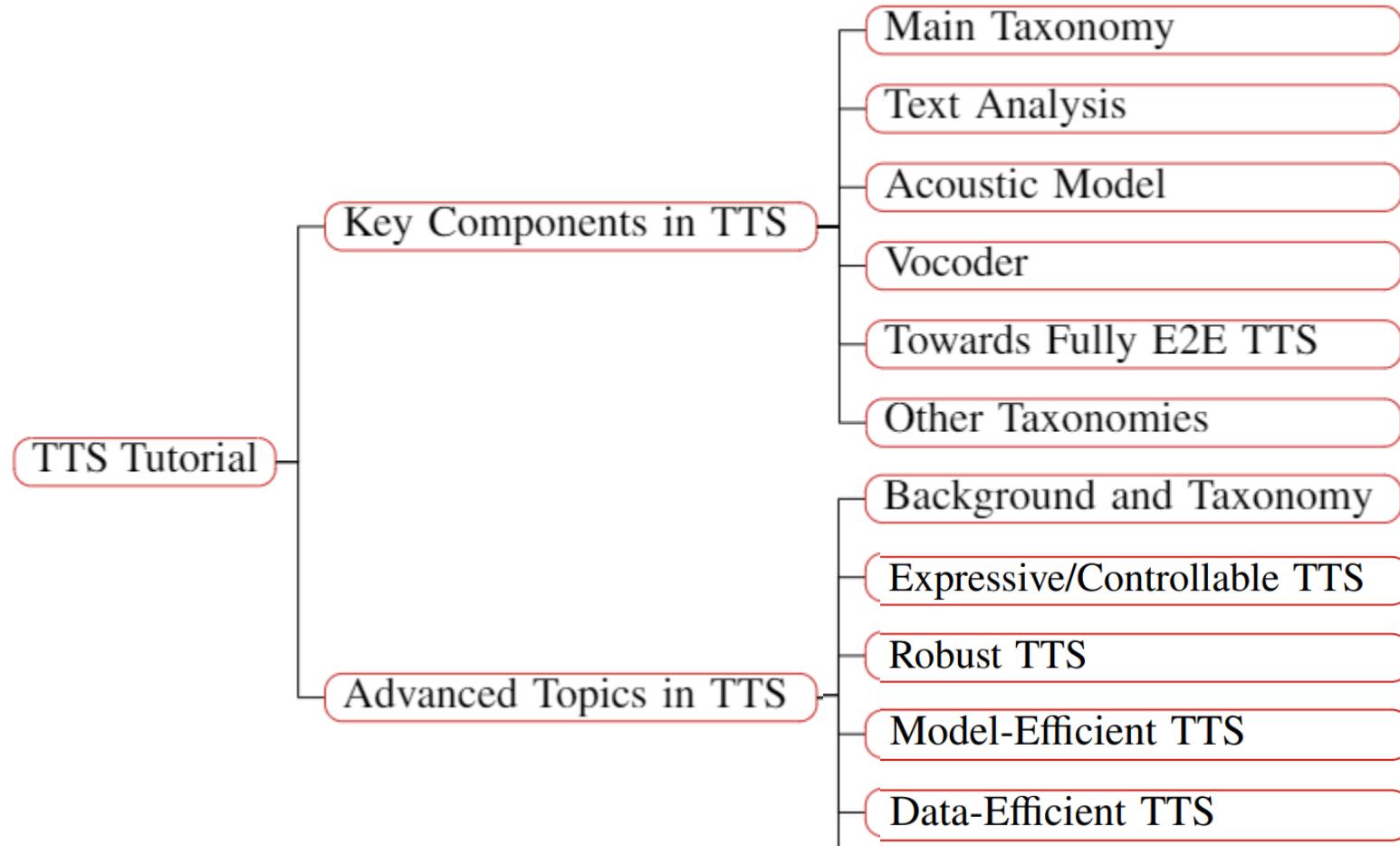
- Expressive/Controllable TTS
- Robust TTS
- Model-Efficient TTS
- Data-Efficient TTS



Part 1: Text-to-Speech Synthesis

Part 1.4: Summary and Future Directions

Summary



Outlook: Higher-quality synthesis

- Powerful generative models
- Better representation learning
- Expressive/controllable/transferrable speech synthesis
- More human-like speech synthesis
 - NaturalSpeech has achieved human-level quality in LJSpeech audiobook at sentence level, but expressive voices, longform audiobook voices are still challenging!
 - **Expressive/emotional voice**
 - Variation information modeling and control
 - Generative models for expressive synthesis
 - **Long-form reading** (article, paragraph, novel)
 - Expressive and consistent prosody
 - **Spontaneous speech** 



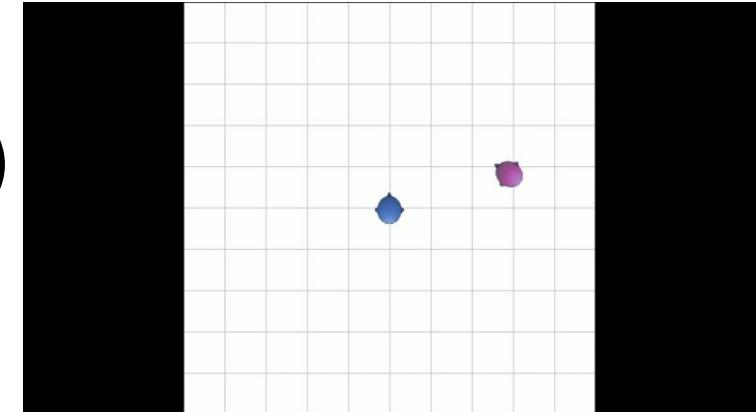
"A Dream of Red Mansions"

Outlook: More efficient synthesis

- Data-efficient TTS
 - Language expansion (**TTS for every language**)
 - Speaker expansion (**TTS for everyone**)
- Model-efficient TTS
 - Computation/memory/time-efficient: Cloud-Edge-End (**TTS for everywhere**)

Outlook: Beyond speech synthesis

- Binaural audio synthesis (spatial sound/metaverse)



- Audio event/effect synthesis



- Singing/music synthesis



- Visualization of speech: talking face synthesis



Reference

See the references in:

A Survey on Neural Speech Synthesis

<https://arxiv.org/pdf/2106.15561.pdf>

A Survey on Neural Speech Synthesis

Xu Tan*, Tao Qin, Frank Soong, Tie-Yan Liu
{xuta, taoqin, frankkps, tyliu}@microsoft.com
Microsoft Research Asia

<https://speechresearch.github.io/>

<https://speechresearch.github.io>

Speech Research

This page lists some speech related research at Microsoft Research Asia, conducted by the team led by [Xu Tan](#). The research topics cover text to speech, singing voice synthesis, music generation, automatic speech recognition, etc. Some research are open-sourced via [NeuralSpeech](#) and [Muzic](#).

We are hiring researchers on speech, NLP, and deep learning at Microsoft Research Asia. Please contact xuta@microsoft.com if you have interests.

[Machine Translation with Speech-Aware Length Control for Video Dubbing](#)

August 30, 2022

[BinauralGrad: A Two-Stage Conditional Diffusion Probabilistic Model for Binaural Audio Synthesis](#)

May 29, 2022

[NaturalSpeech: End-to-End Text to Speech Synthesis with Human-Level Quality](#)

May 03, 2022

[Mixed-Phoneme BERT: Improving BERT with Mixed Phoneme and Sup-Phoneme Representations for Text to Speech](#)

April 02, 2022

[AdaSpeech 4: Adaptive Text to Speech in Zero-Shot Scenarios](#)

March 06, 2022

[Speech-T: Transducer for Text to Speech and Beyond](#)

October 06, 2021

[TeleMelody: Lyric-to-Melody Generation with a Template-Based Two-Stage Method](#)

A book on TTS

A book on “*Neural Text-to-Speech Synthesis*”, by Xu Tan

will be published soon!

Watch this repo for update: <https://github.com/tts-tutorial/book>

We are hiring

- Research FTE (social/campus hire)
 - Speech (TTS/ASR)
 - NLP (Machine Translation, Text Summarization, Pre-training, etc)
 - Generative Models (AR, GAN, Flow, VAE, Diffusion Model)
 - Machine Learning, Deep Learning
- Research Intern
 - Speech, Music, Machine Translation, Machine Learning

Machine Learning Group, Microsoft Research Asia

Xu Tan xuta@microsoft.com

Thank You!

Xu Tan/谭旭

Principal Researcher and Research Manager @ Microsoft Research Asia

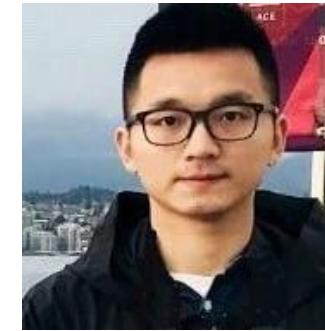
xuta@microsoft.com

tan-xu.github.io

<https://www.microsoft.com/en-us/research/people/xuta/>

<https://speechresearch.github.io/>

Neural Speech Synthesis



Xu Tan
Microsoft Research Asia

Hung-yi Lee
National Taiwan University



Slides can be found in <https://github.com/tts-tutorial/interspeech2022>

INTER SPEECH 2022
2022-09-18