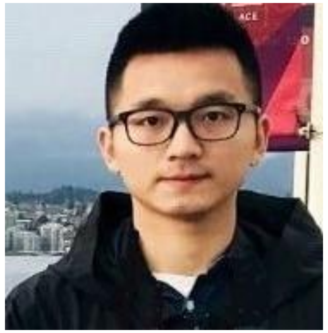
Microsoft

IEEE ICASSP 2022

Recent Advances in Neural Speech Synthesis



Xu Tan and Tao Qin
Microsoft Research Asia

Tutorial slides: https://github.com/tts-tutorial/icassp2022
Survey paper: https://arxiv.org/pdf/2106.15561

Outline

1. Evolution and taxonomy of TTS, Tao Qin
2. Key components in TTS, Xu Tan
3. Advanced topics in TTS, Xu Tan
4. Summary and future directions, Xu Tan
5. QA

Part 1: Evolution and Taxonomy

-- Evolution, basic modules, taxonomies

alexa

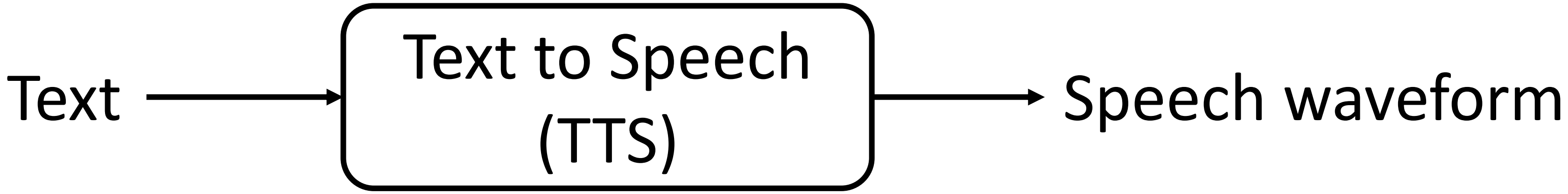
Hi, I'm Cortana.

Google

Siri

Text to speech synthesis

- The artificial production of human speech from text



- Disciplines: acoustics, linguistics, digital signal processing, statistics and deep learning
- The quality of the synthesized speech is measured by
  - Intelligibility and naturalness

Formant TTS

How does it work?

- produce speech segments by generating artificial signals based on a set of specified rules mimicking the formant structure and other spectral properties of natural speech
- using additive synthesis and an acoustic model (with parameters like voicing, fundamental frequency, noise levels)

Advantages:

- highly intelligible, even at high speeds
- well-suited for embedded systems, with limited memory and computation power

Limitations:

- not natural, produces artificial, robotic-sounding speech, far from human speech
- difficult to design rules that specify model parameters

Concatenative TTS

How does it work?

- a very large database of short and high-quality speech fragments are recorded from a single speaker
- speech fragments are recombined to form complete utterances

Advantages: intelligible

Limitations:

- require huge databases and hard-coding the combination
- emotionless, not natural
- difficult to modify the voice (e.g., switching to a different speaker, or altering the emphasis or emotion) without recording a whole new database

Parametric TTS

How does it work?

- using learning based parametric models, e.g., HMM
- all the information required to generate speech is stored in the parameters of the model
- also called statistical parametric synthesis (SPSS)

Advantages: lower data cost and more flexible

Limitations: less intelligible than concatenative TTS

Neural TTS

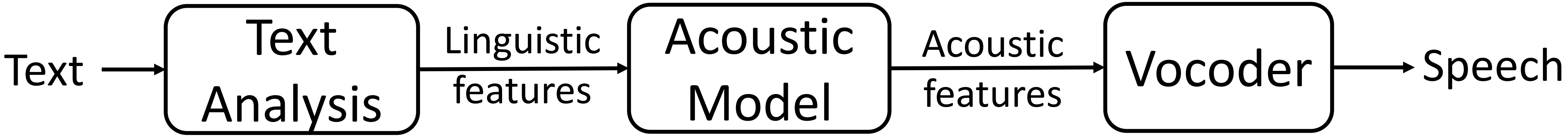
How does it work?

- a special kind of parametric models
- text to waveform mapping is modeled by (deep) neural networks

- Advantages:
  - huge quality improvement, in terms of both intelligibility and naturalness
  - less human preprocessing and feature engineering
- Disadvantages:
  - Data hungry
  - Training/inference costly

Basic components of parametric/neural TTS systems

- Text analysis, acoustic model, and vocoder



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

Text analysis

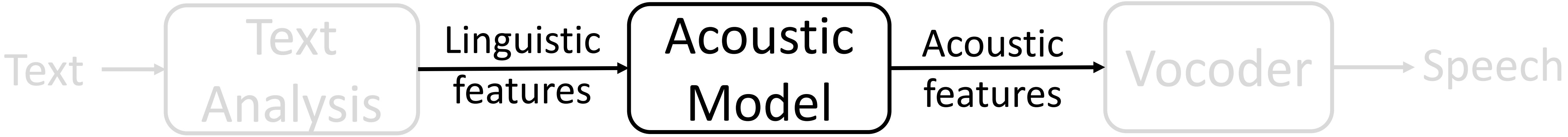
- Transforms input text into linguistic features:
  - Text normalization
    - 1989 → nineteen eighty-nine, *Jan. 24th* → *January twenty-fourth*
  - Homograph disambiguation
    - Do you **live (/l ih v/)** near a zoo with **live (/l ay v/)** animals?
  - Phrase/word/syllable segmentation
    - synthesis → syn-the-sis
  - Part of speech (POS) tagging
    - Mary went to the store → noun, verb, prep, noun,
  - ToBI (Tones and Break Indices)
    - Mary went to the store ? → Mary' store' H%
  - Grapheme-to-phoneme conversion
    - *Speech* → *s p iy ch*

Text analysis: linguistic features

- phoneme:
  - current phoneme
  - preceding and succeeding two phonemes
  - position of current phoneme within current syllable
- syllable:
  - numbers of phonemes within preceding, current, and succeeding syllables
  - stress[3] and accent[4] of preceding, current, and succeeding syllables
  - positions of current syllable within current word and phrase
  - numbers of preceding and succeeding stressed syllables within current phrase
  - numbers of preceding and succeeding accented syllables within current phrase
  - number of syllables from previous stressed syllable
  - number of syllables to next stressed syllable
  - number of syllables from previous accented syllable
  - number of syllables to next accented syllable
  - vowel identity within current syllable
- word:
  - guess at part of speech of preceding, current, and succeeding words
  - numbers of syllables within preceding, current, and succeeding words
  - position of current word within current phrase
  - numbers of preceding and succeeding content words within current phrase
  - number of words from previous content word
  - number of words to next content word
- phrase:
  - numbers of syllables within preceding, current, and succeeding phrases
  - position of current phrase in major phrases
  - ToBI endtone of current phrase
- utterance:
  - numbers of syllables, words, and phrases in utterance

Acoustic model

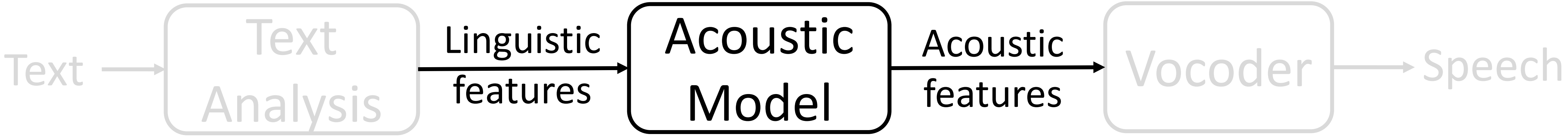
- Generate acoustic features from linguistic features

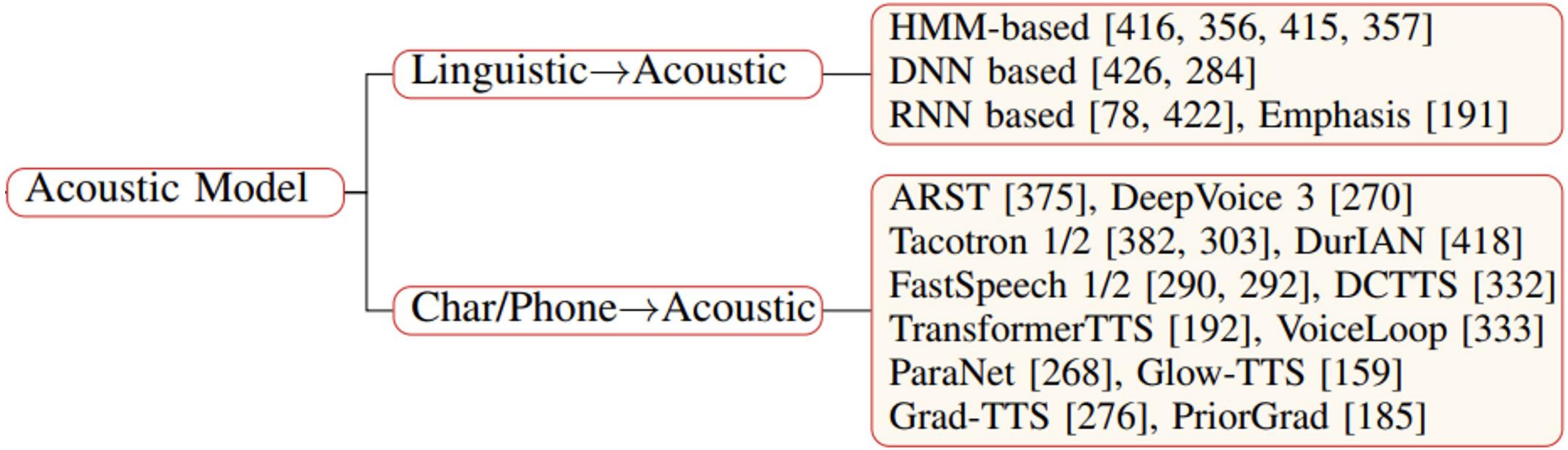


  - F0, V/UV, energy
  - Mel-scale Frequency Cepstral Coefficients (MFCC), Bark-Frequency Cepstral Coefficients (BFCC)
  - Mel-generalized coefficients (MGC), band aperiodicity (BAP),
  - Linear prediction coefficients (LPC),
  - Mel-spectrograms
    - Pre-emphasis, Framing, Windowing, Short-Time Fourier Transform (STFT), Mel filter

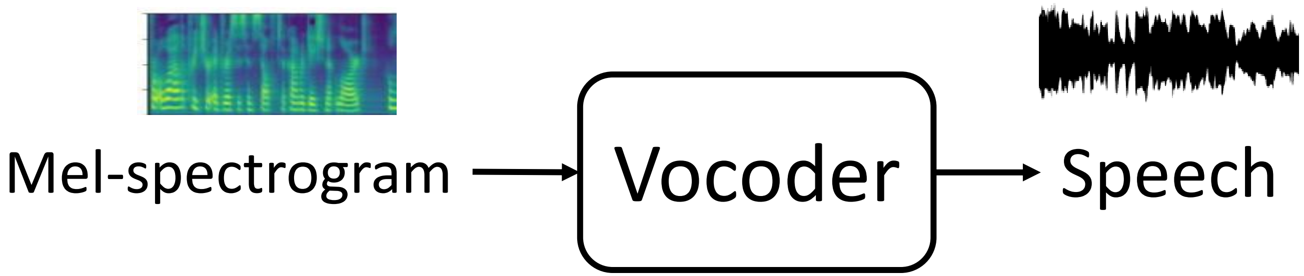
Acoustic model

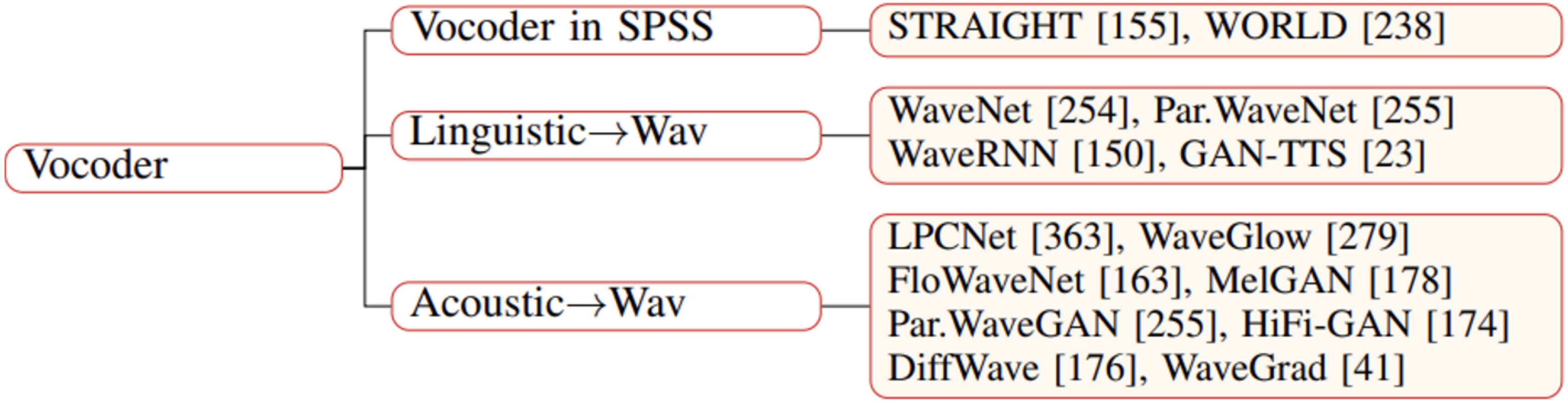
- Predict acoustic features from linguistic features



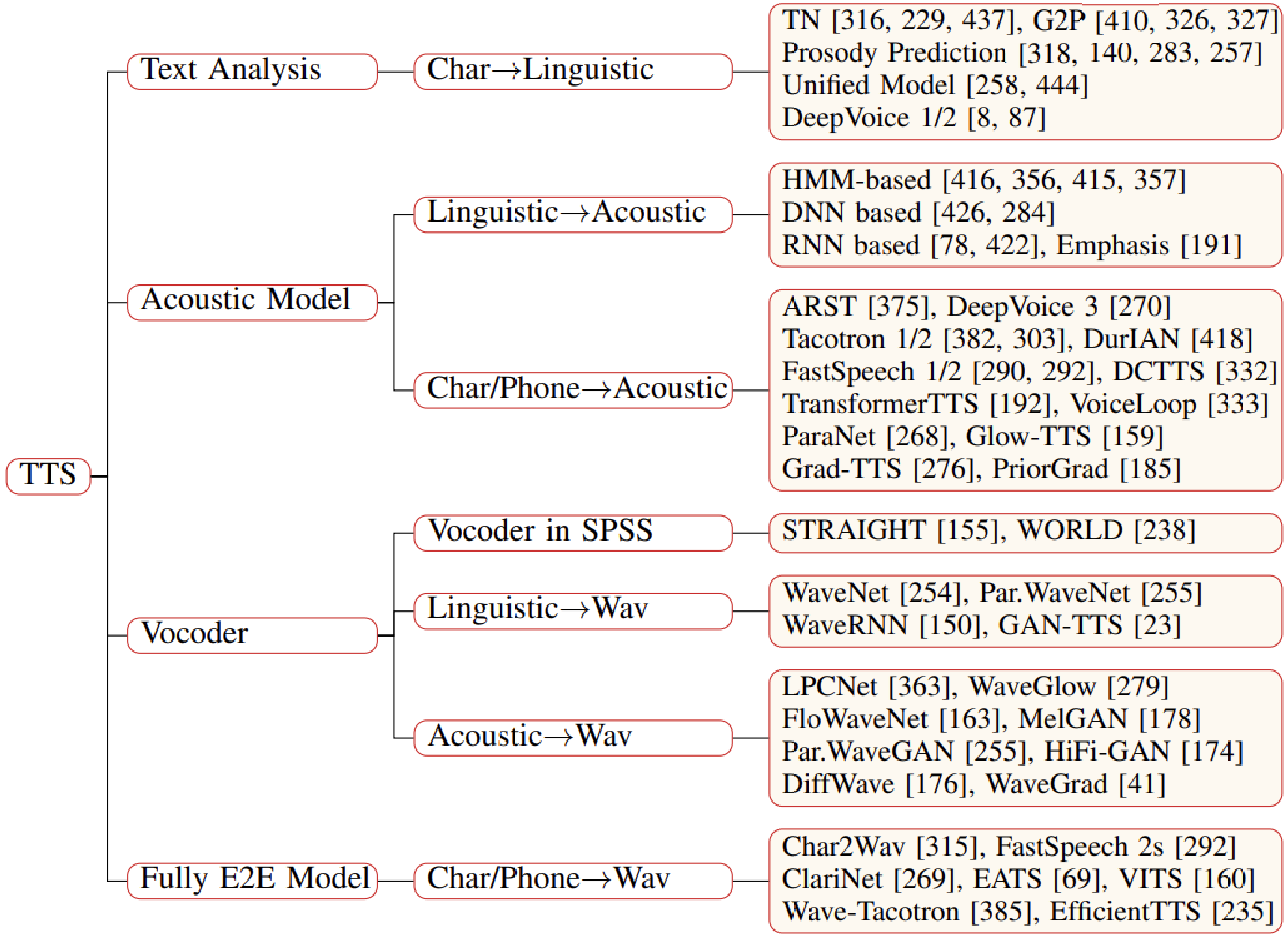


Vocoder

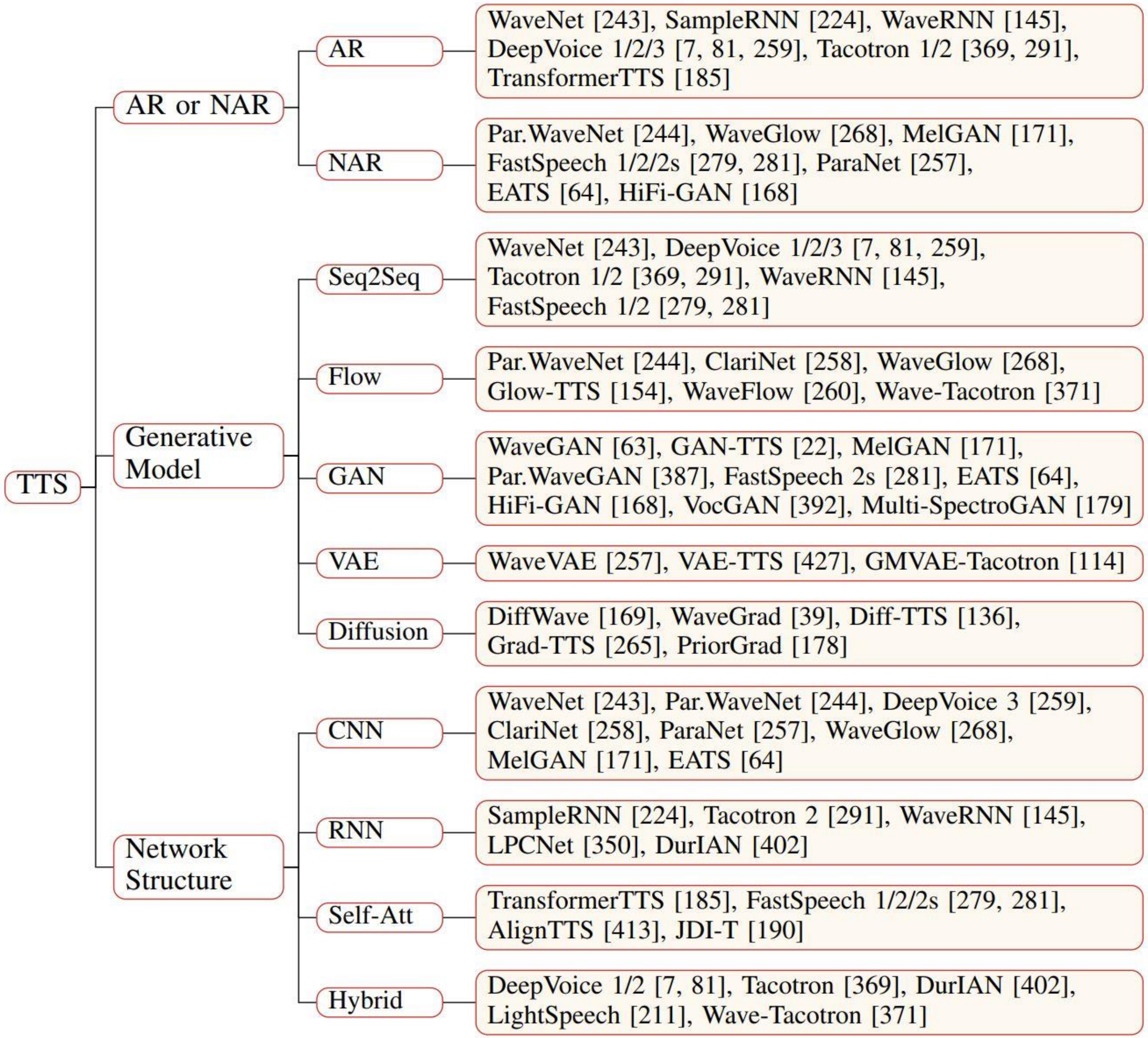

Mel-spectrogram
Vocoder
Speech


Vocoder
Vocoder in SPSS
STRAIGHT [155], WORLD [238]
Linguistic→Wav
WaveNet [254], Par.WaveNet [255]
WaveRNN [150], GAN-TTS [23]
Acoustic→Wav
LPCNet [363], WaveGlow [279]
FloWaveNet [163], MelGAN [178]
Par.WaveGAN [255], HiFi-GAN [174]
DiffWave [176], WaveGrad [41]

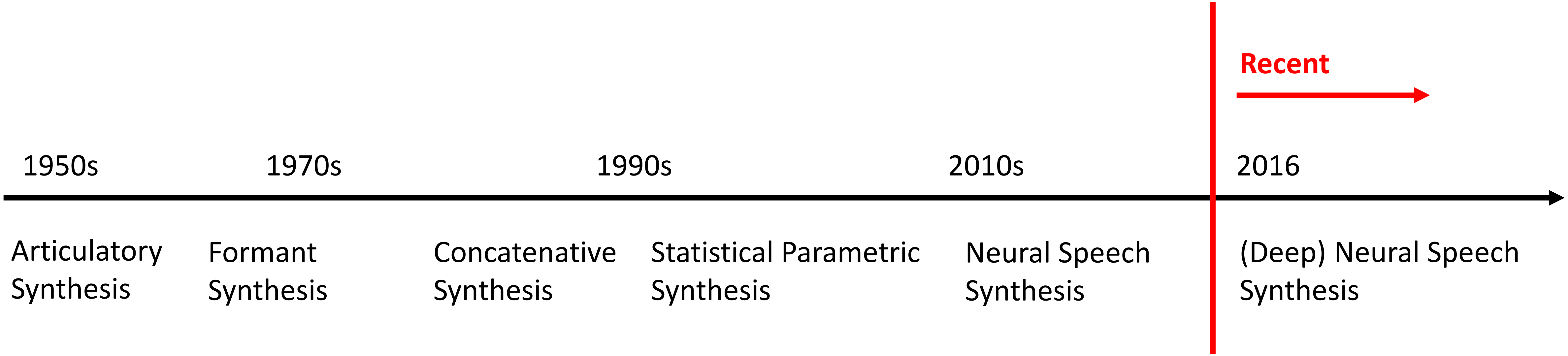
Taxonomy from the perspective of components



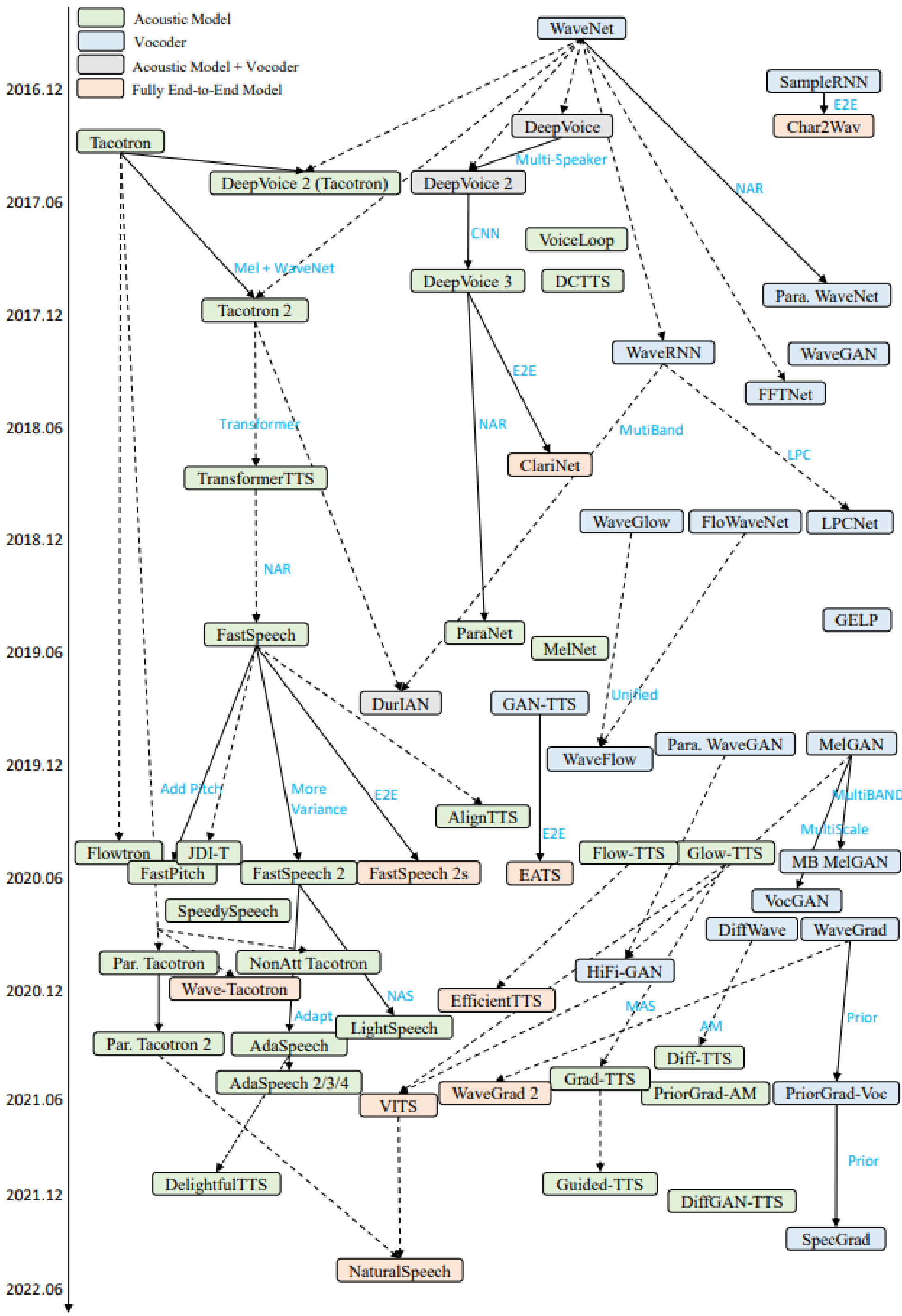
Taxonomy from other perspectives



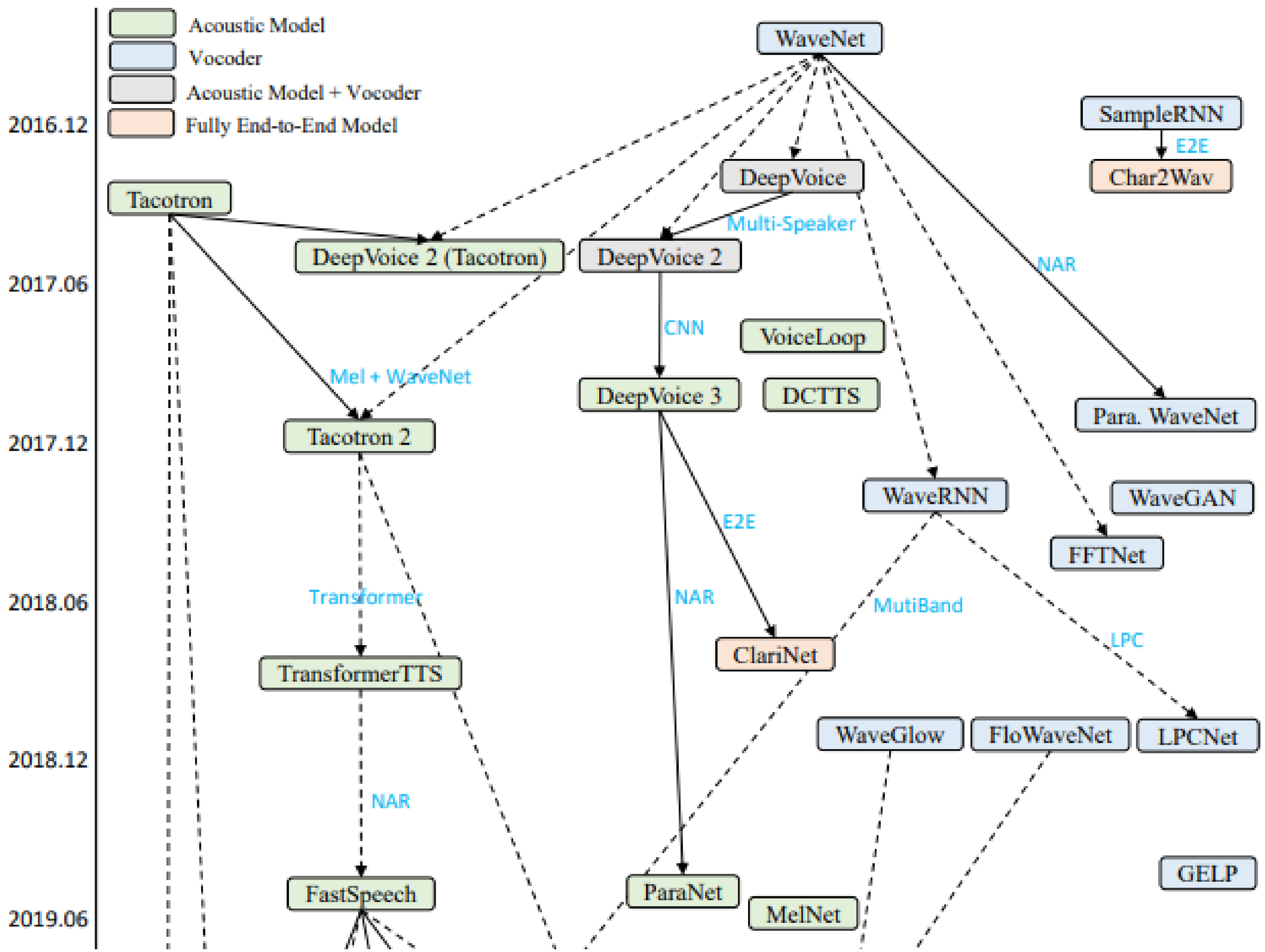
How “recent” this tutorial covers?

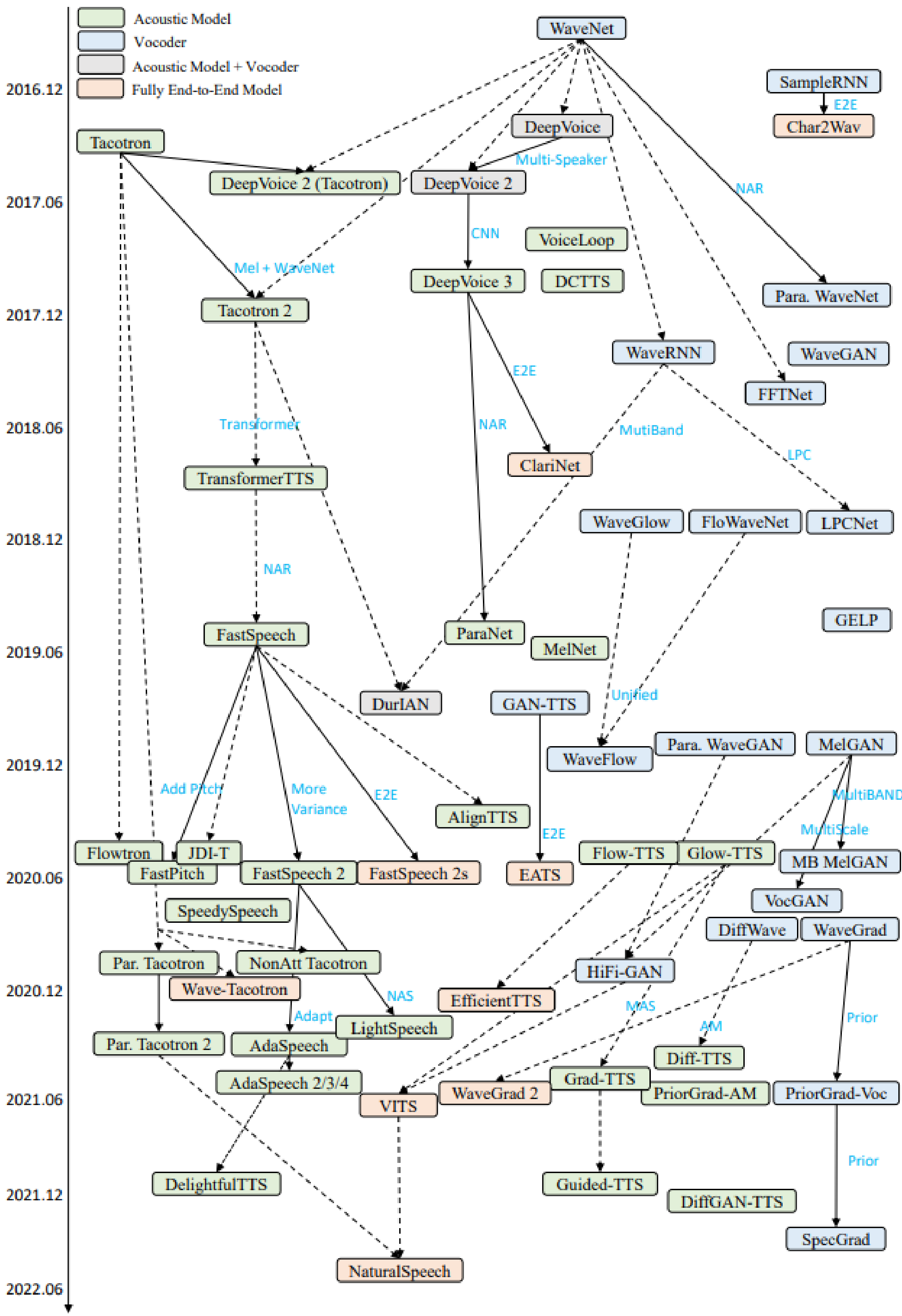


Recent advances

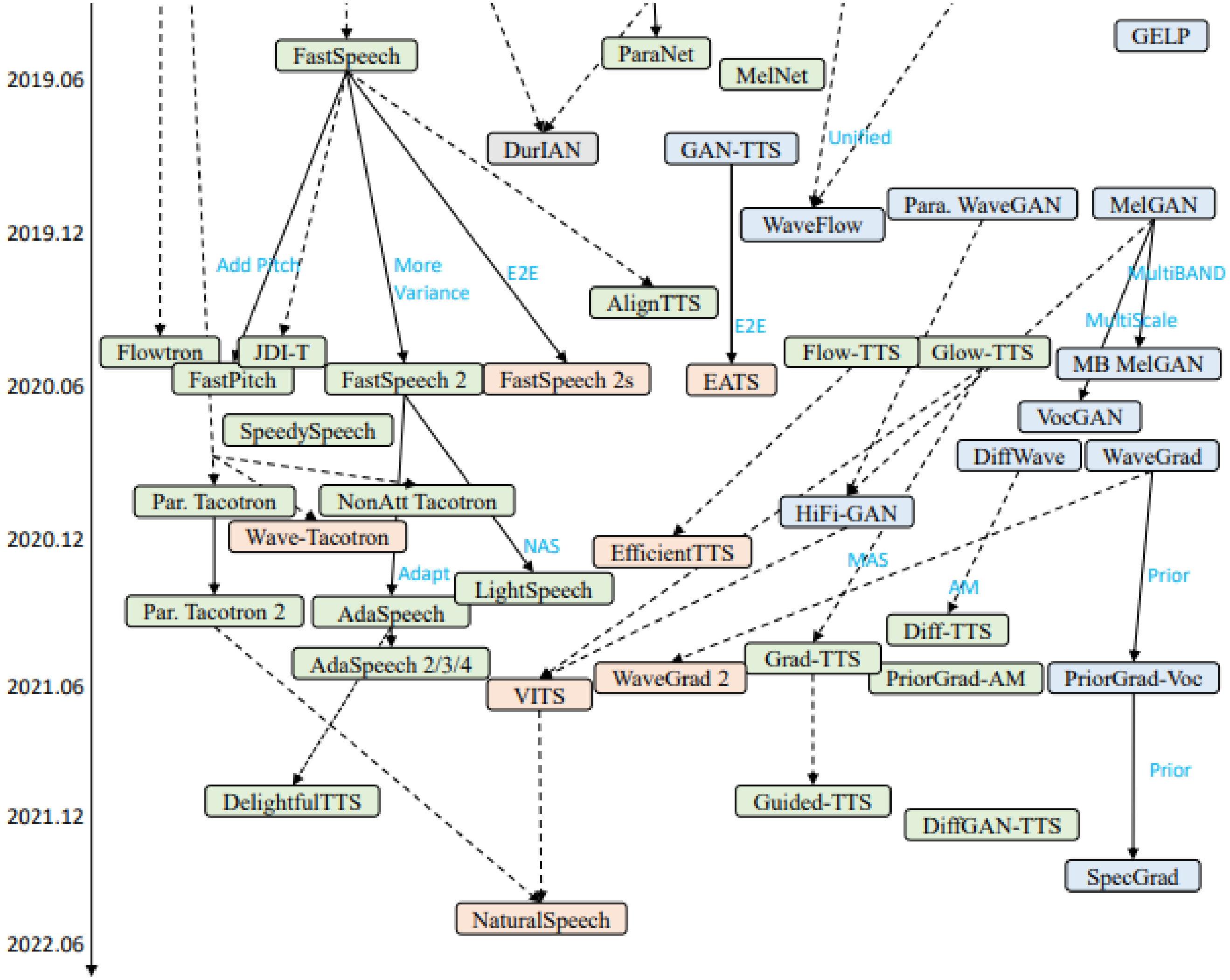


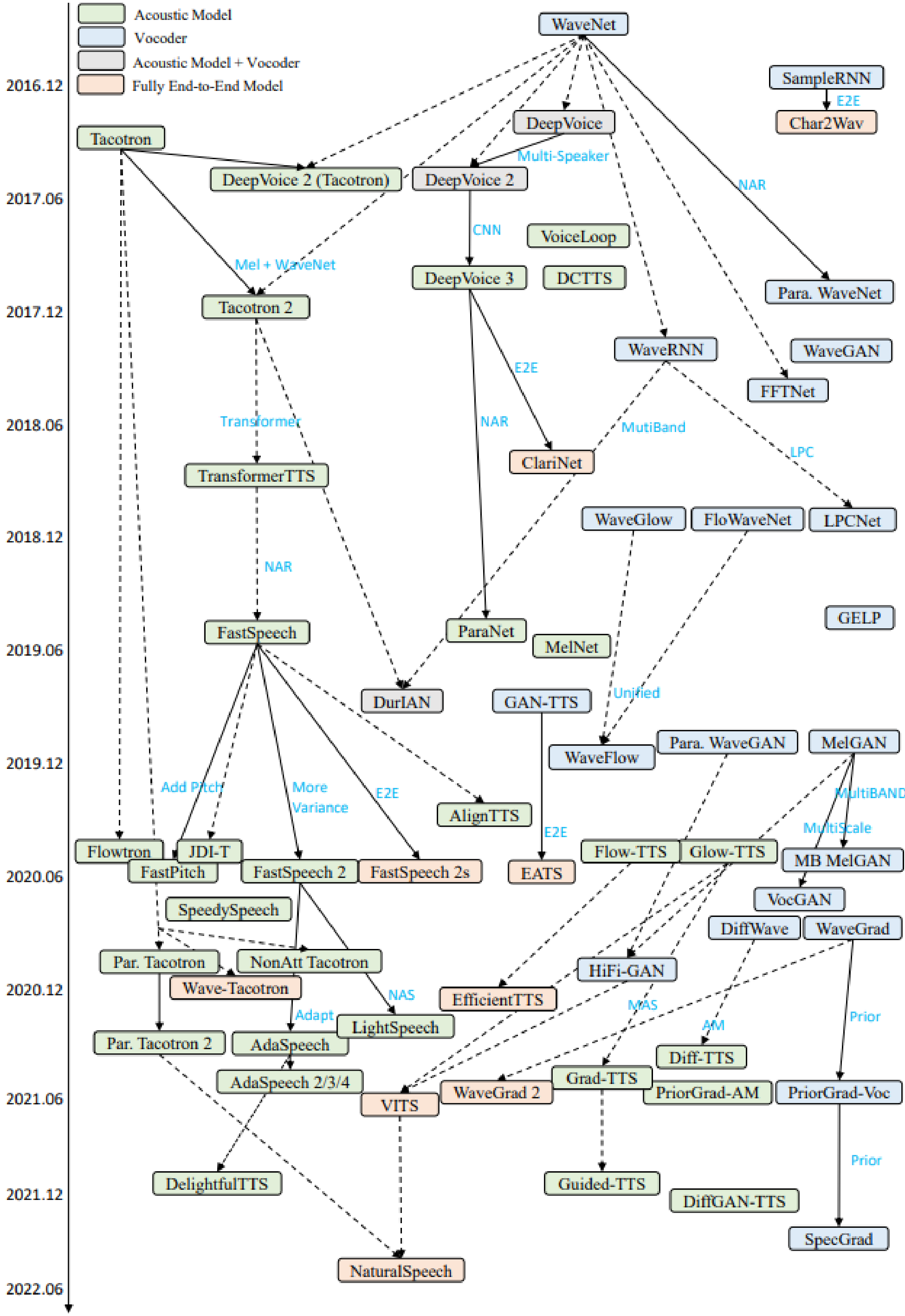
Recent advances





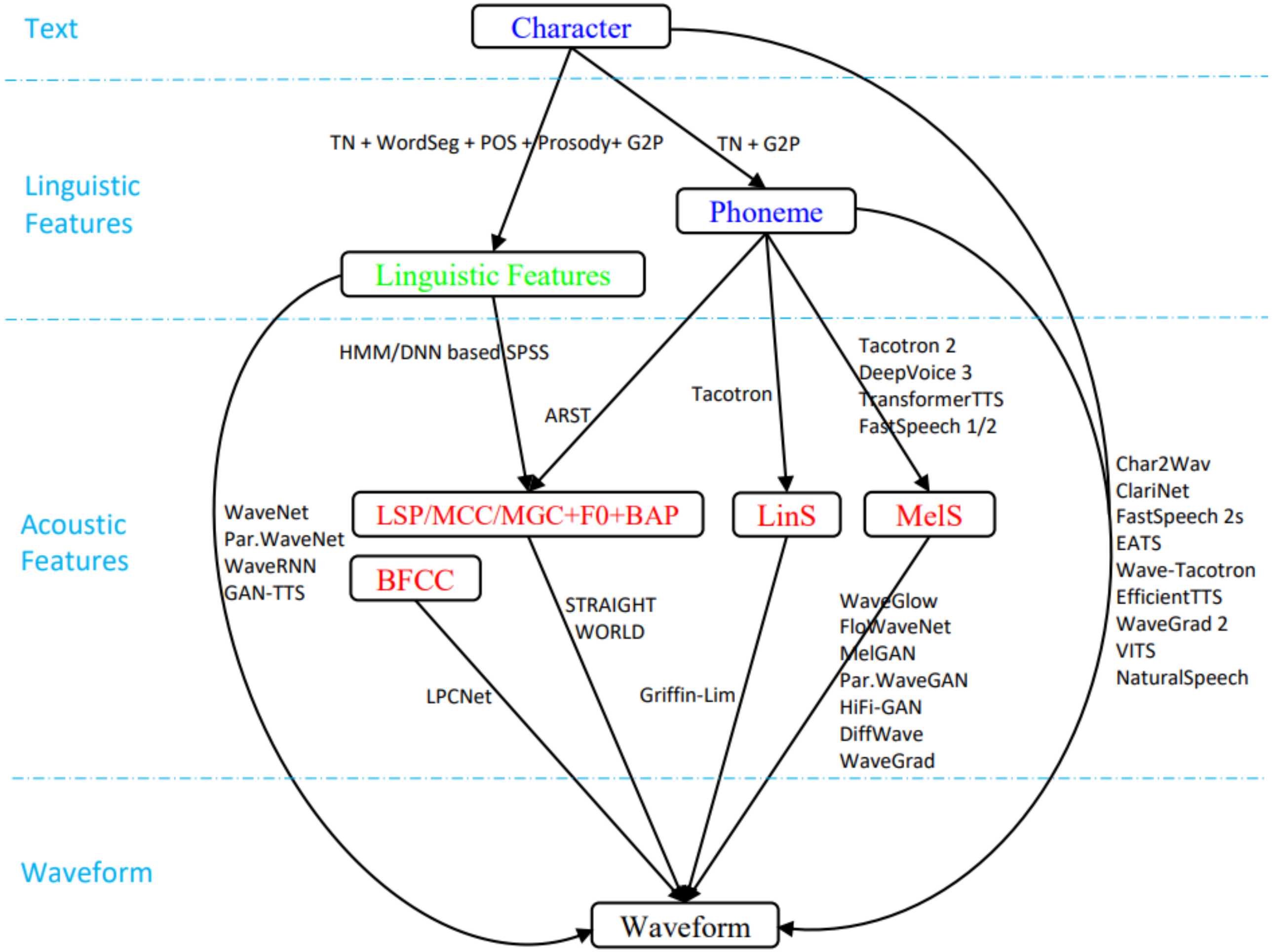
Recent advances



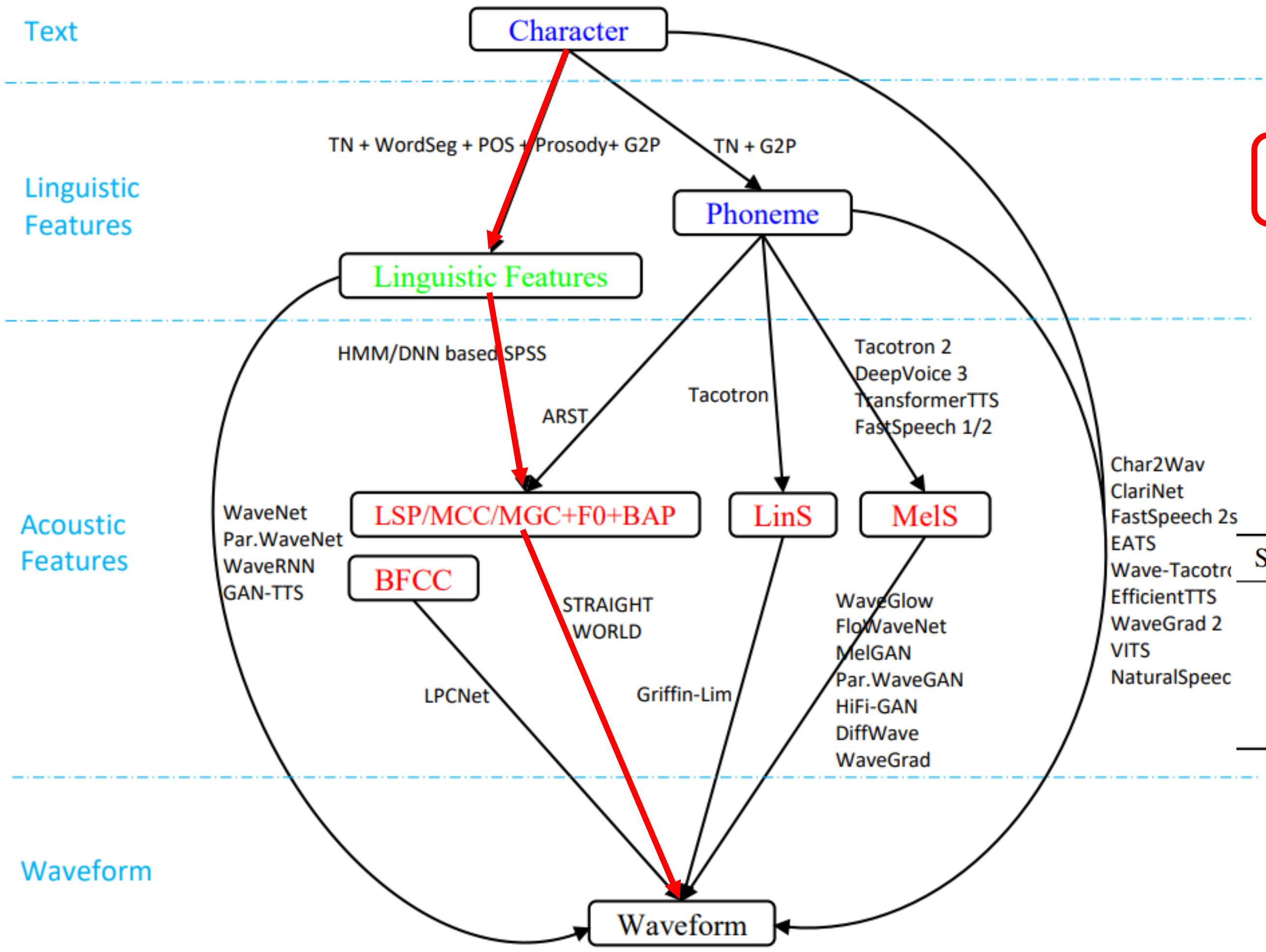


Part 2: Key Components in TTS

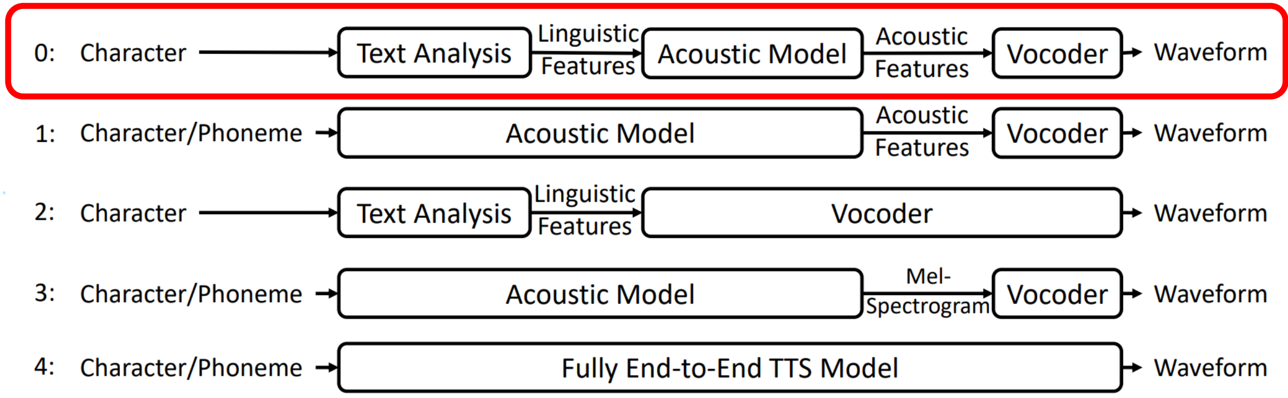
Data conversion pipeline



Data conversion pipeline

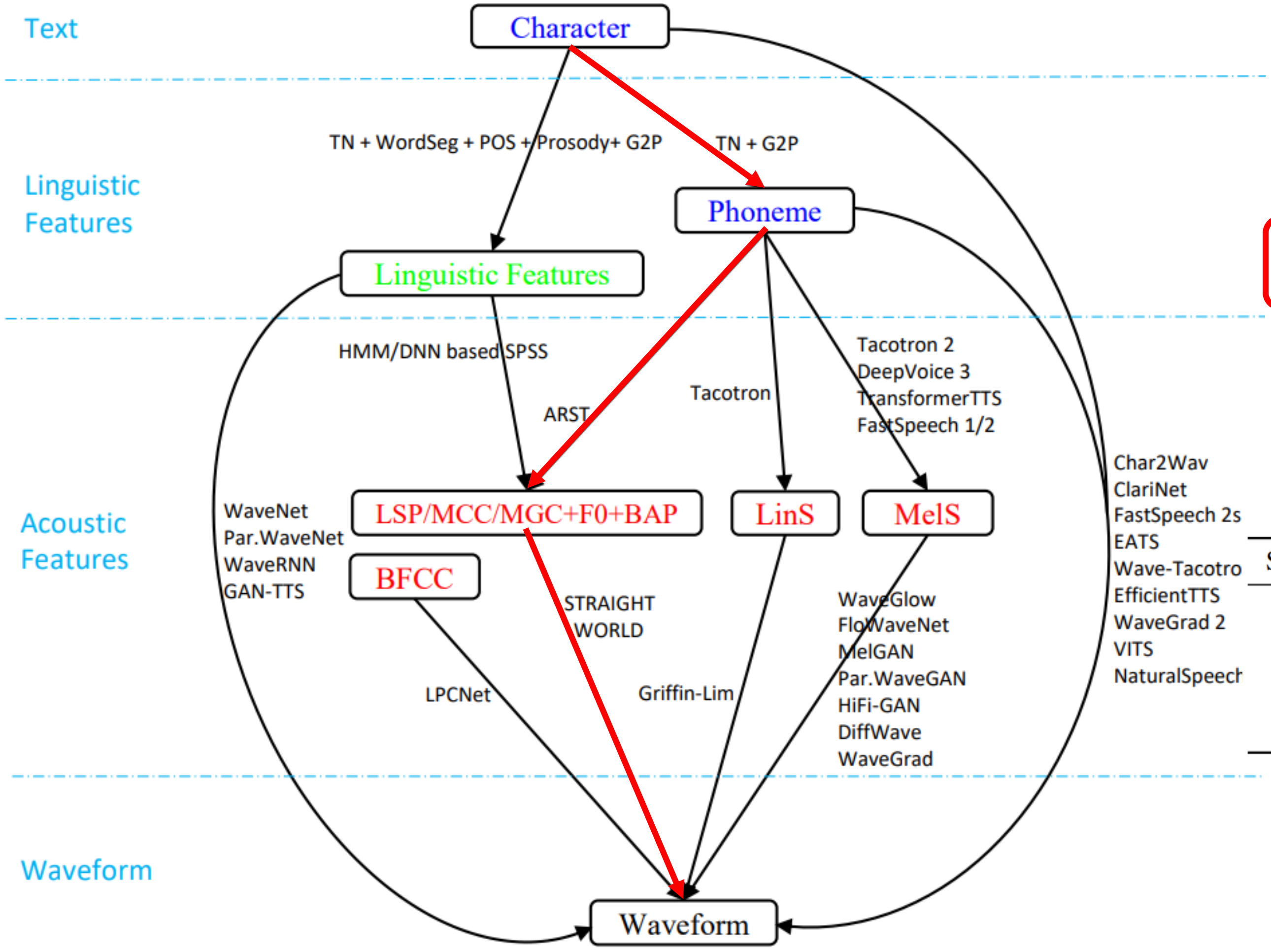


Path 0

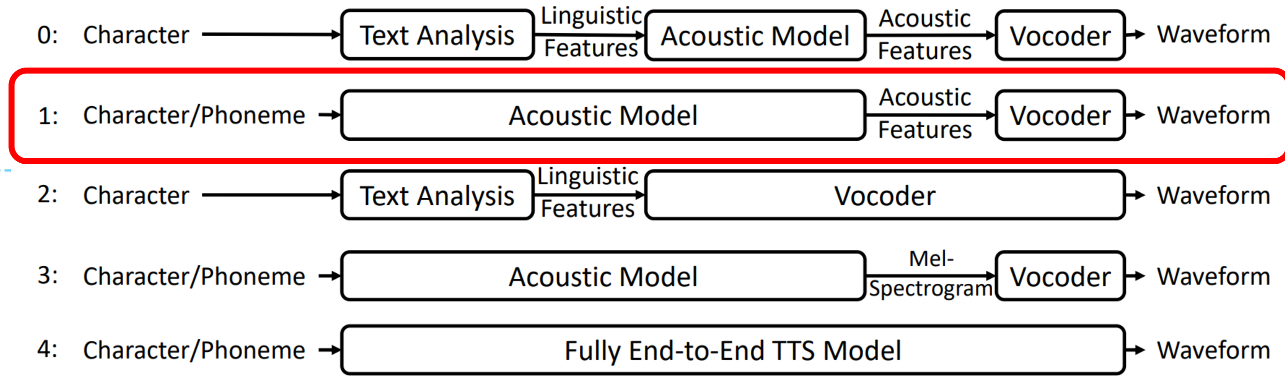


Stage	Models
0	SPSS [418, 358, 417, 427, 359]
1	ARST [377]
2	WaveNet [255], DeepVoice 1/2 [8, 88], Par. WaveNet [256], WaveRNN [151], HiFi-GAN [23]
3	DeepVoice 3 [271], Tacotron 2 [304], FastSpeech 1/2 [291, 293], WaveGlow [280], FloWaveNet [164]
4	Char2Wav [316], ClariNet [270], FastSpeech 2s [293], EATS [70], VITS [161], NaturalSpeech [346]

Data conversion pipeline

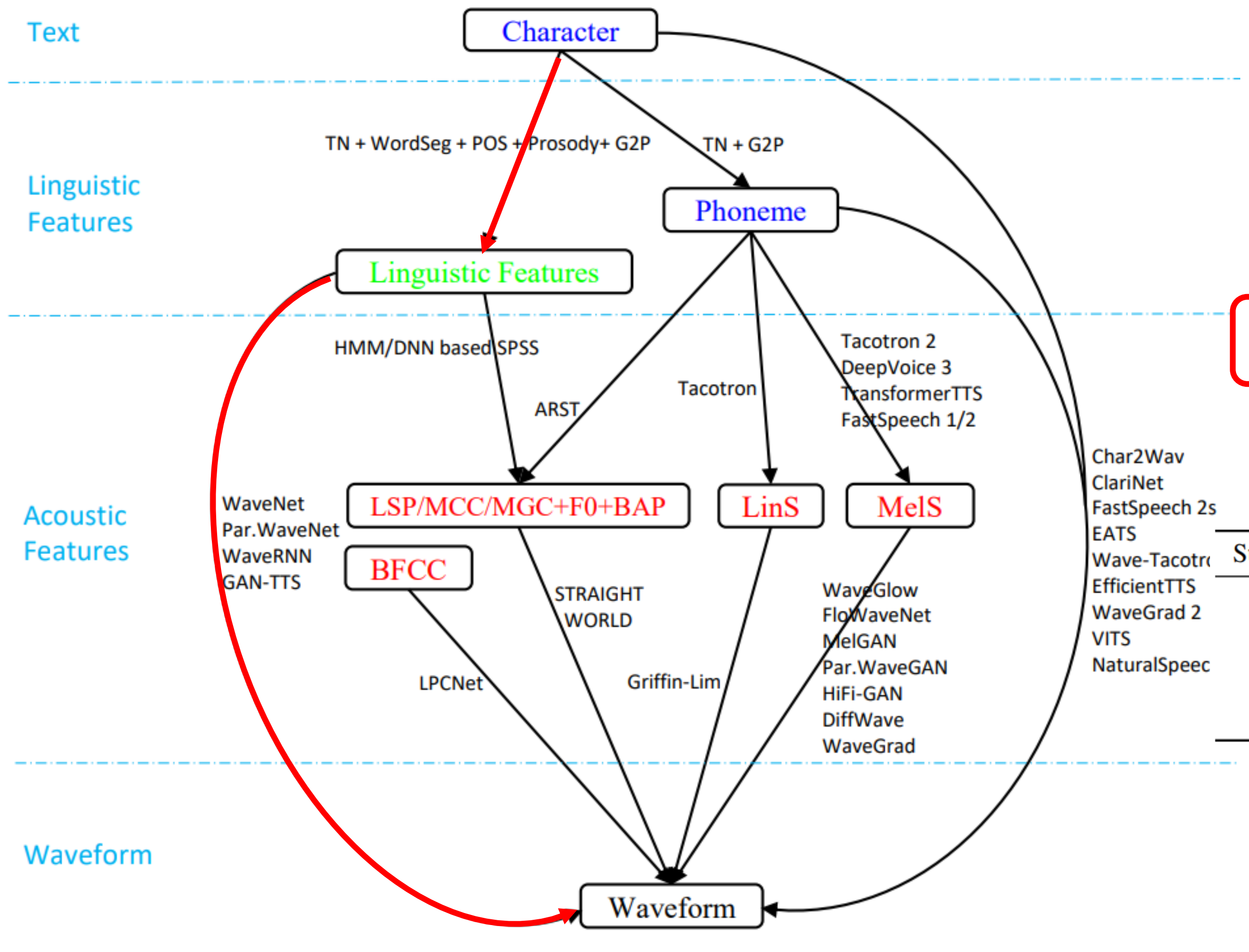


Path 1

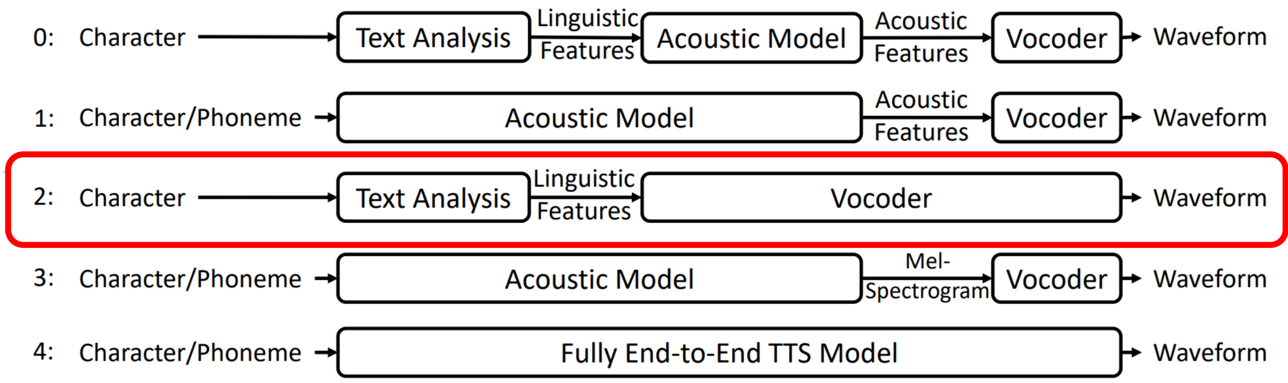


Stage	Models
0	SPSS [418, 358, 417, 427, 359]
1	ARST [377]
2	WaveNet [255], DeepVoice 1/2 [8, 88], Par. WaveNet [256], WaveRNN [151], HiFi-GAN [23]
3	DeepVoice 3 [271], Tacotron 2 [304], FastSpeech 1/2 [291, 293], WaveGlow [280], FloWaveNet [164]
4	Char2Wav [316], ClariNet [270], FastSpeech 2s [293], EATS [70], VITS [161], NaturalSpeech [346]

Data conversion pipeline

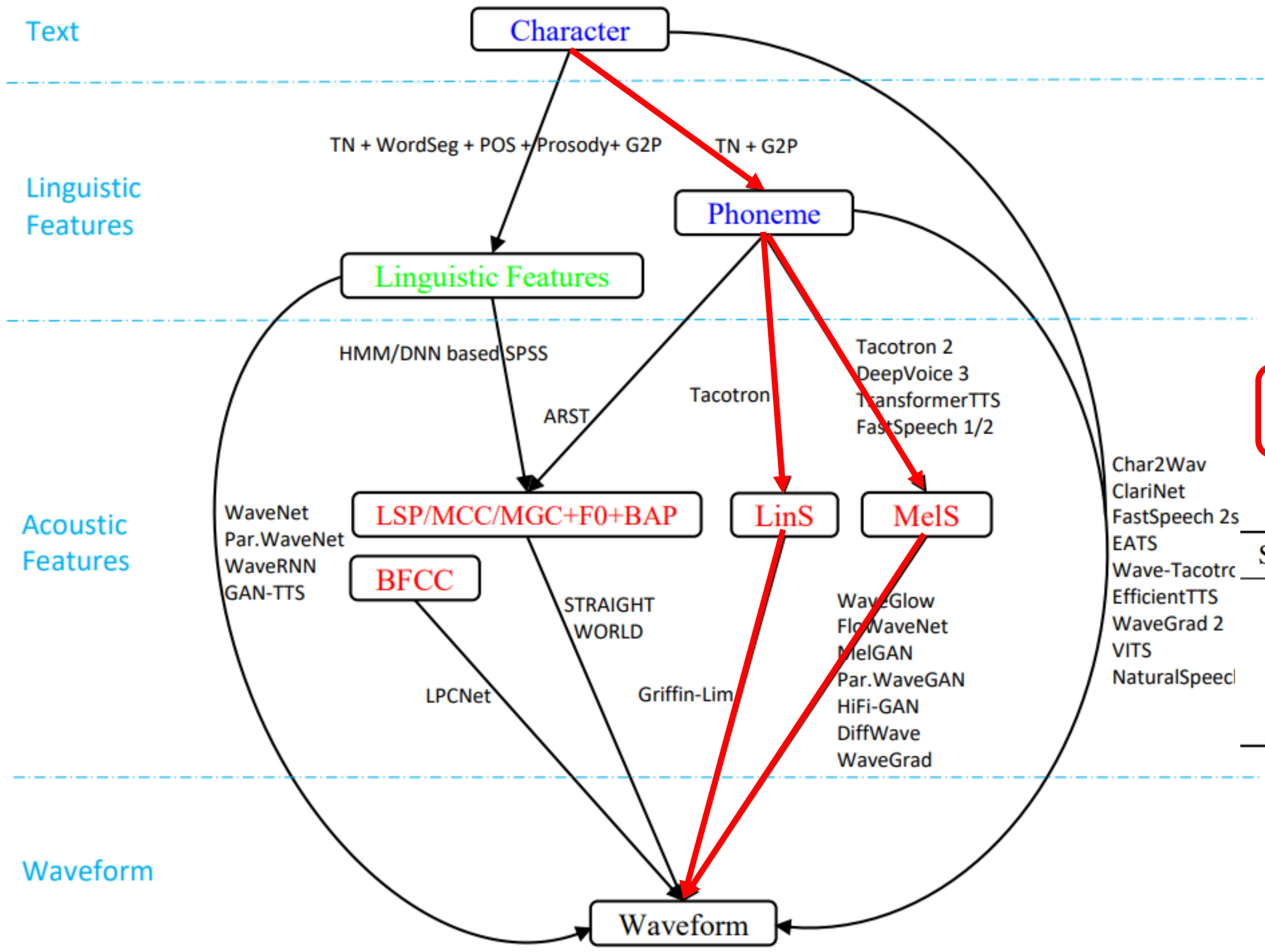


Path 2

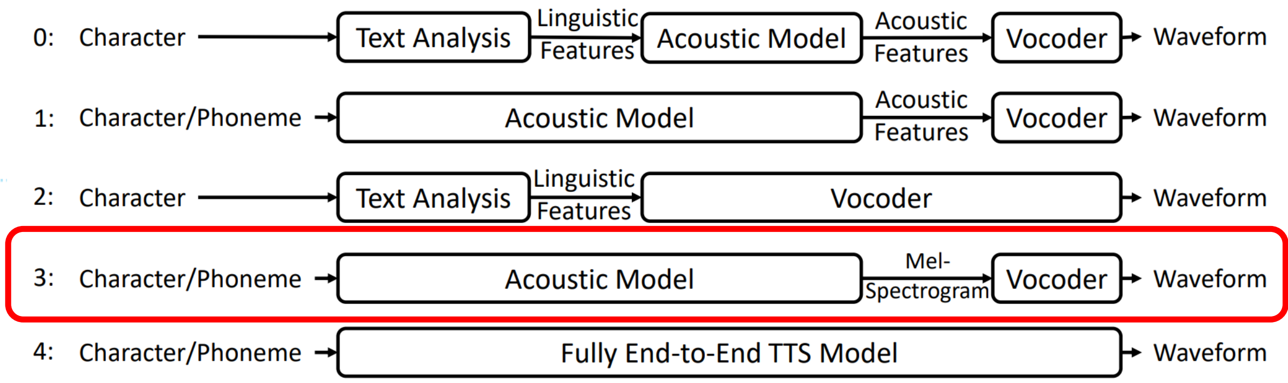


Stage	Models
0	SPSS [418, 358, 417, 427, 359]
1	ARST [377]
2	WaveNet [255], DeepVoice 1/2 [8, 88], Par. WaveNet [256], WaveRNN [151], HiFi-GAN [23]
3	DeepVoice 3 [271], Tacotron 2 [304], FastSpeech 1/2 [291, 293], WaveGlow [280], FloWaveNet [164]
4	Char2Wav [316], ClariNet [270], FastSpeech 2s [293], EATS [70], VITS [161], NaturalSpeech [346]

Data conversion pipeline

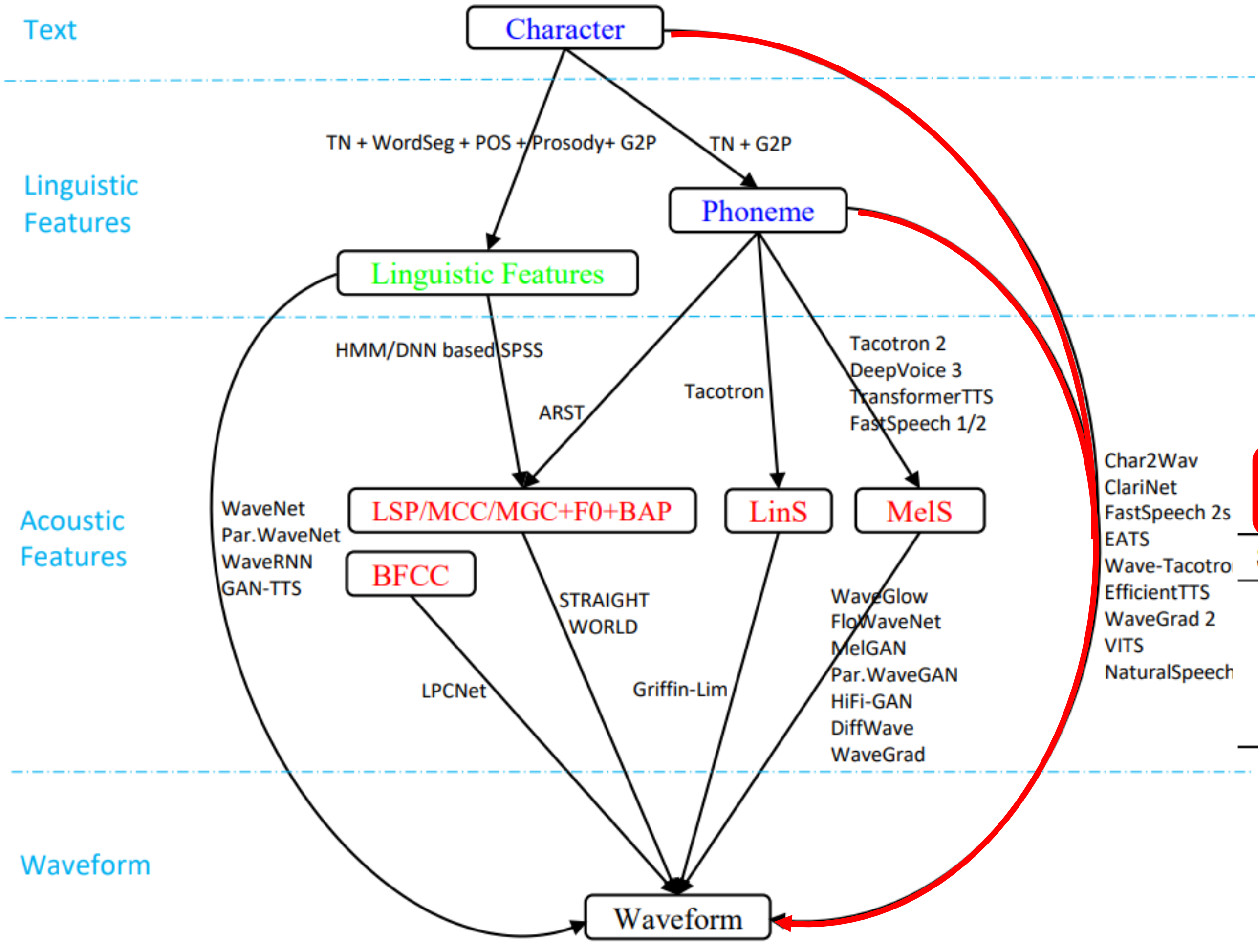


Path 3

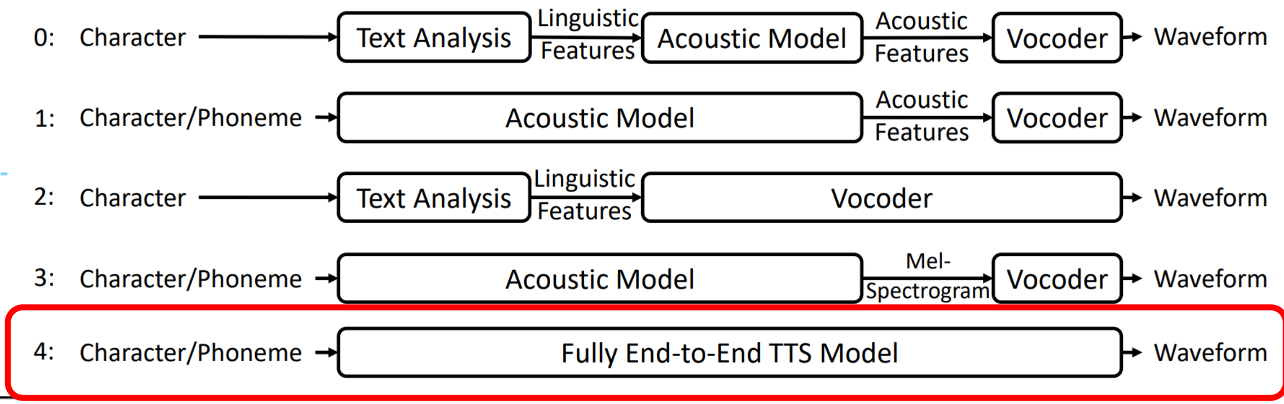


Stage	Models
0	SPSS [418, 358, 417, 427, 359]
1	ARST [377]
2	WaveNet [255], DeepVoice 1/2 [8, 88], Par. WaveNet [256], WaveRNN [151], HiFi-GAN [23]
3	DeepVoice 3 [271], Tacotron 2 [304], FastSpeech 1/2 [291, 293], WaveGlow [280], FloWaveNet [164]
4	Char2Wav [316], ClariNet [270], FastSpeech 2s [293], EATS [70], VITS [161], NaturalSpeech [346]

Data conversion pipeline

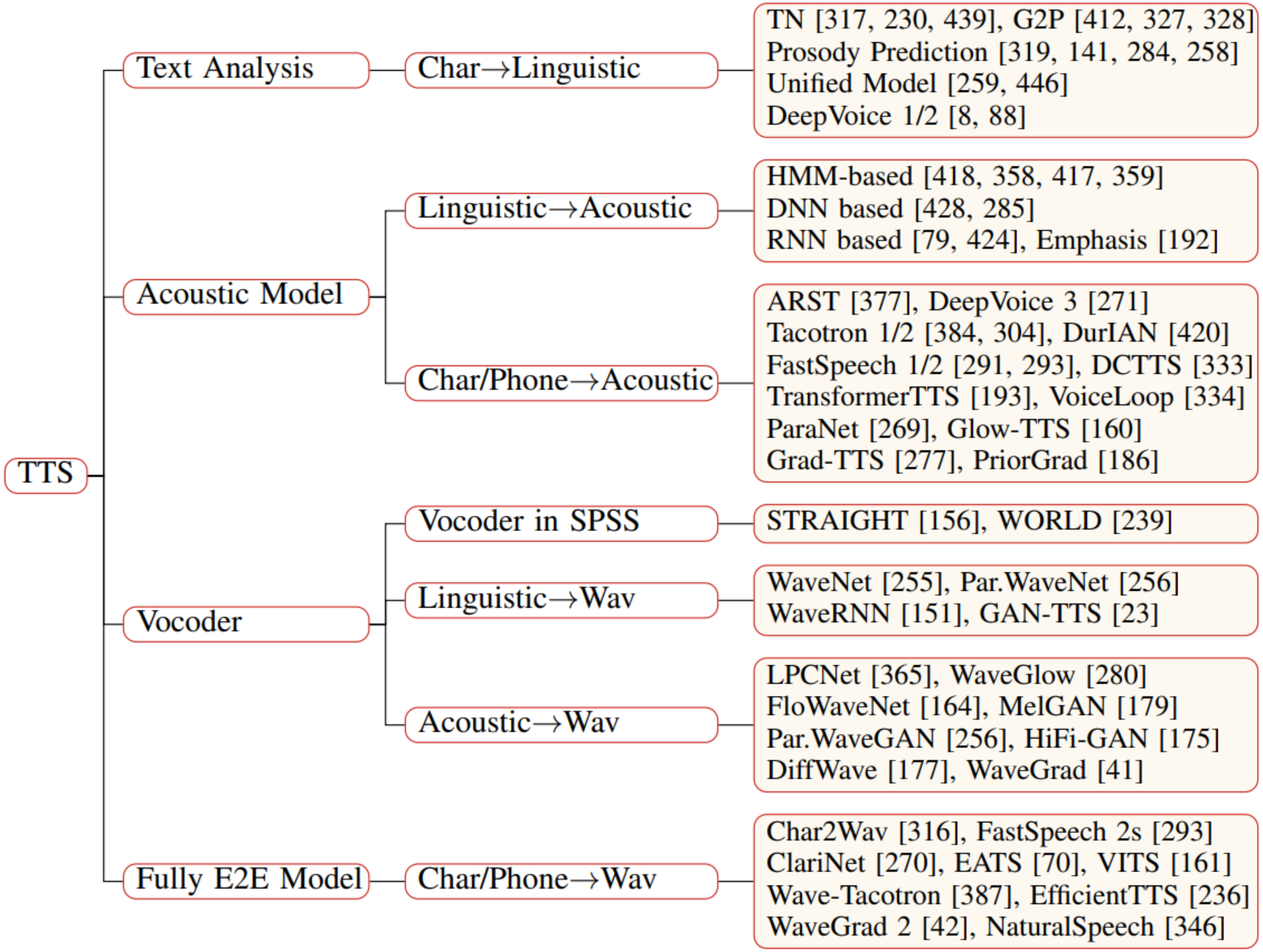


Path 4



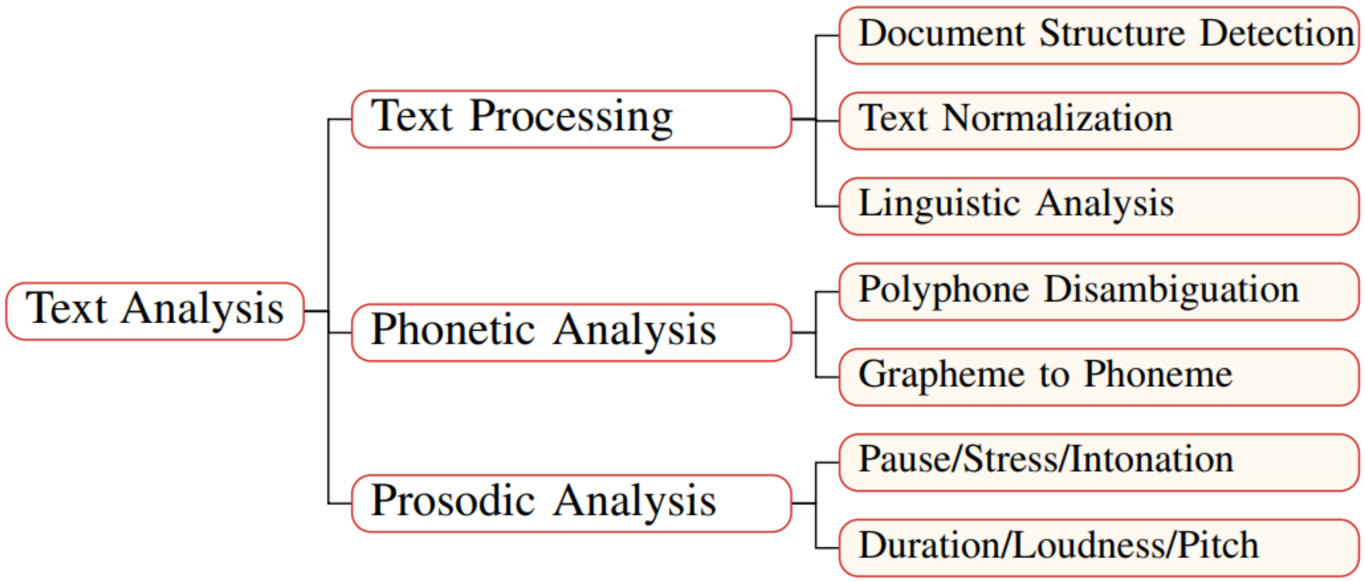
Stage	Models
0	SPSS [418, 358, 417, 427, 359]
1	ARST [377]
2	WaveNet [255], DeepVoice 1/2 [8, 88], Par. WaveNet [256], WaveRNN [151], HiFi-GAN [23]
3	DeepVoice 3 [271], Tacotron 2 [304], FastSpeech 1/2 [291, 293], WaveGlow [280], FloWaveNet [164]
4	Char2Wav [316], ClariNet [270], FastSpeech 2s [293], EATS [70], VITS [161], NaturalSpeech [346]

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

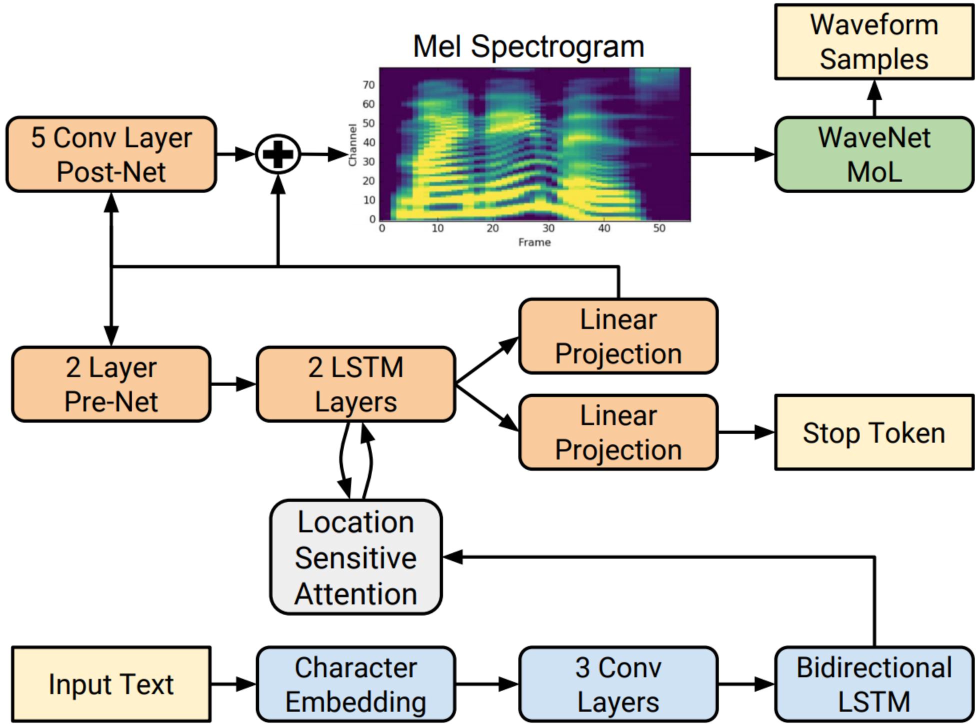
Acoustic model

- 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)

	Acoustic Model	Input→Output	AR/NAR	Modeling	Structure
SPSS	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
RNN	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
CNN	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
Transformer	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	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
VAE	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	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
Diffusion	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

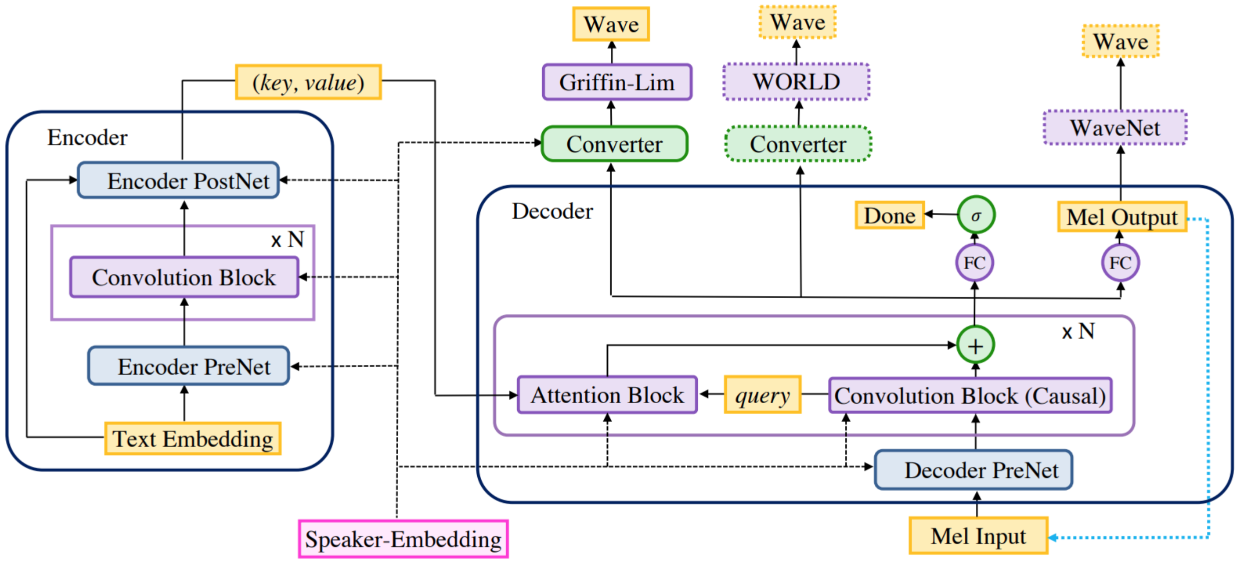
- Tacotron 2 [303]
  - Evolved from Tacotron [382]
  - Text to mel-spectrogram generation
  - LSTM based encoder and decoder
  - Location sensitive attention
  - WaveNet as the vocoder



- Other works
  - GST-Tacotron [383], Ref-Tacotron [309]
  - DurIAN [418]
  - Non-Attentative Tacotron [304]
  - Patallel Tacotron 1/2 [74, 75]
  - WaveTacotron [385]

Acoustic model——CNN based

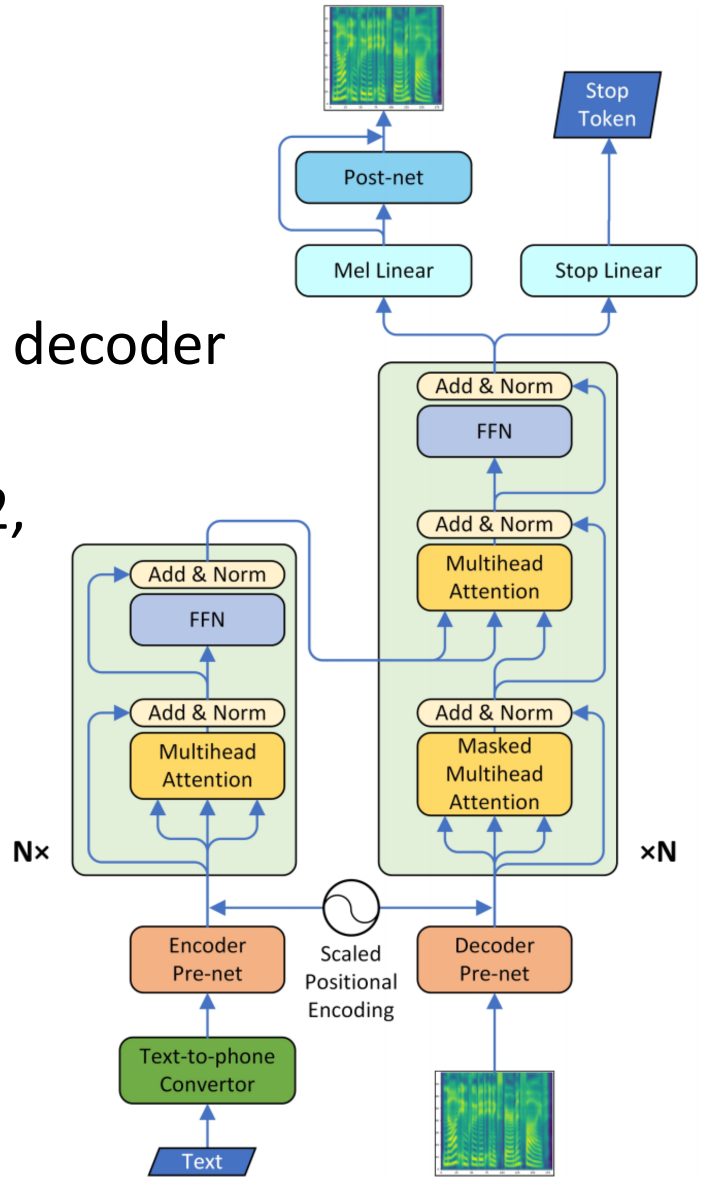
- DeepVoice 3 [270]
  - Evolved from DeepVoice 1/2 [8, 87]
  - Enhanced with purely CNN based structure
  - Support different acoustic features as output
  - Support multi-speakers



- Other works
  - DCTTS [332] (**Contemporary**)
  - ClariNet [269]
  - ParaNet [268]

Acoustic model——Transformer based

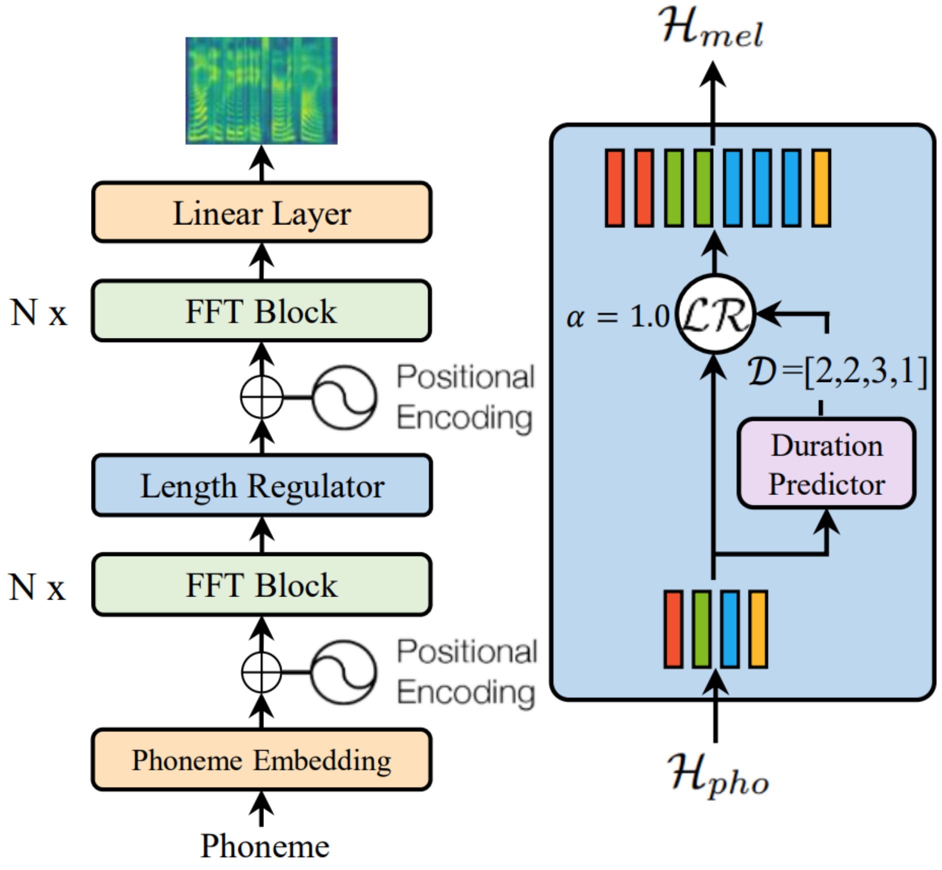
- TransformerTTS [192]
  - Framework is like Tacotron 2
  - Replace LSTM with Transformer in encoder and decoder
  - Parallel training, quality on par with Tacotron 2
  - Attention with more challenges than Tacotron 2, due to parallel computing



- Other works
  - MultiSpeech [39]
  - Robutrans [194]

Acoustic model——Transformer based

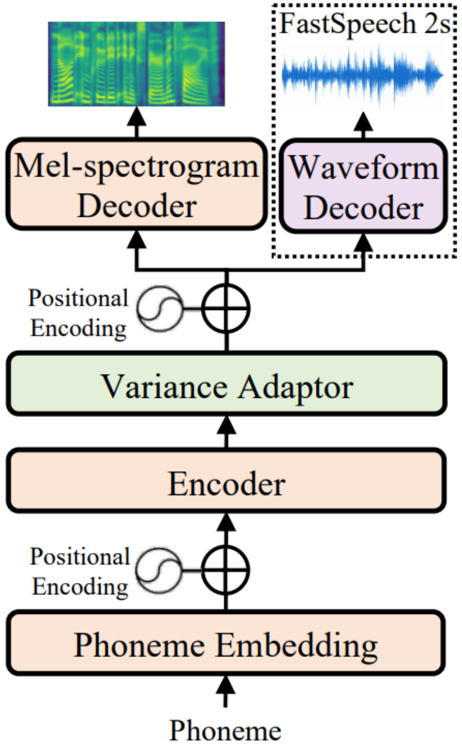
- FastSpeech [290]
  - Generate mel-spectrogram in parallel (for speedup)
  - Remove the text-speech attention mechanism (for robustness)
  - Feed-forward transformer with length regulator (for controllability)



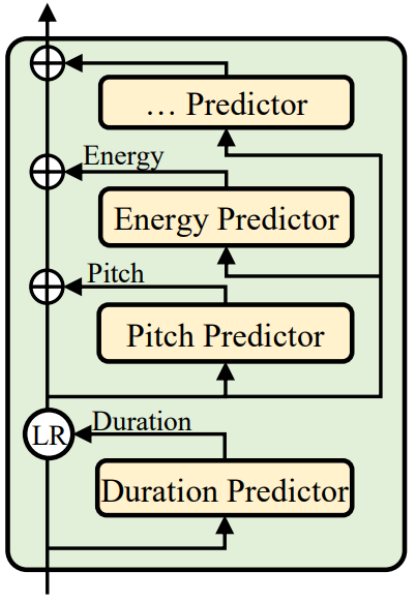
Acoustic model——Transformer based

- 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

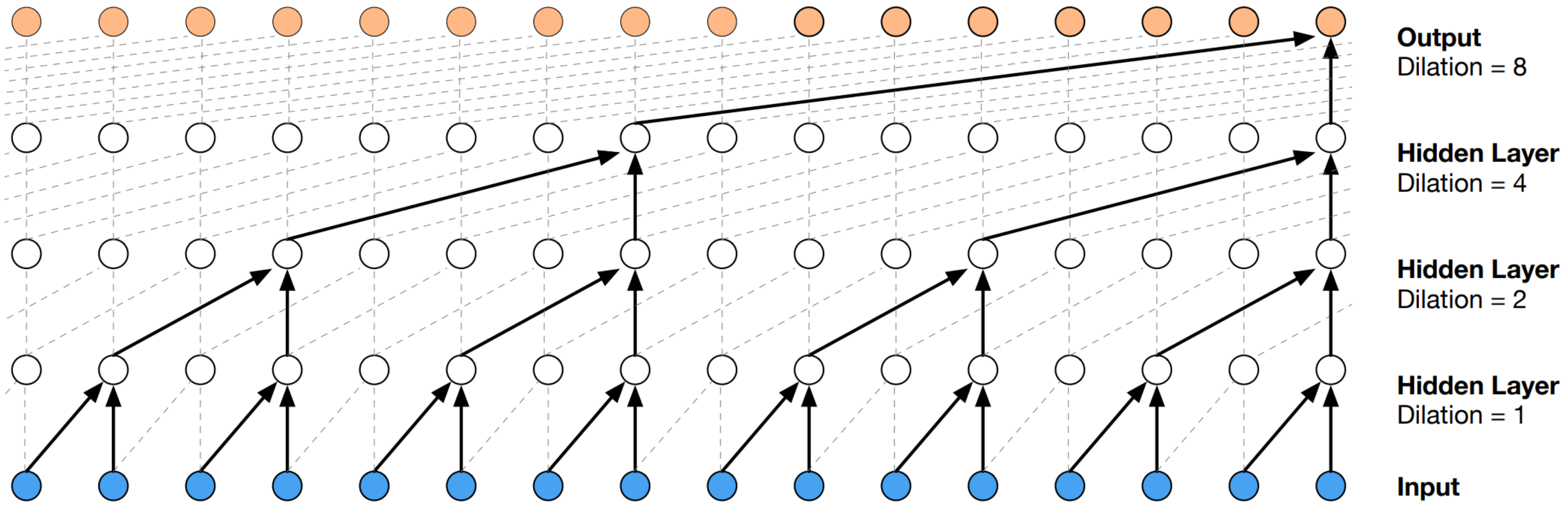
Vocoder

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

	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

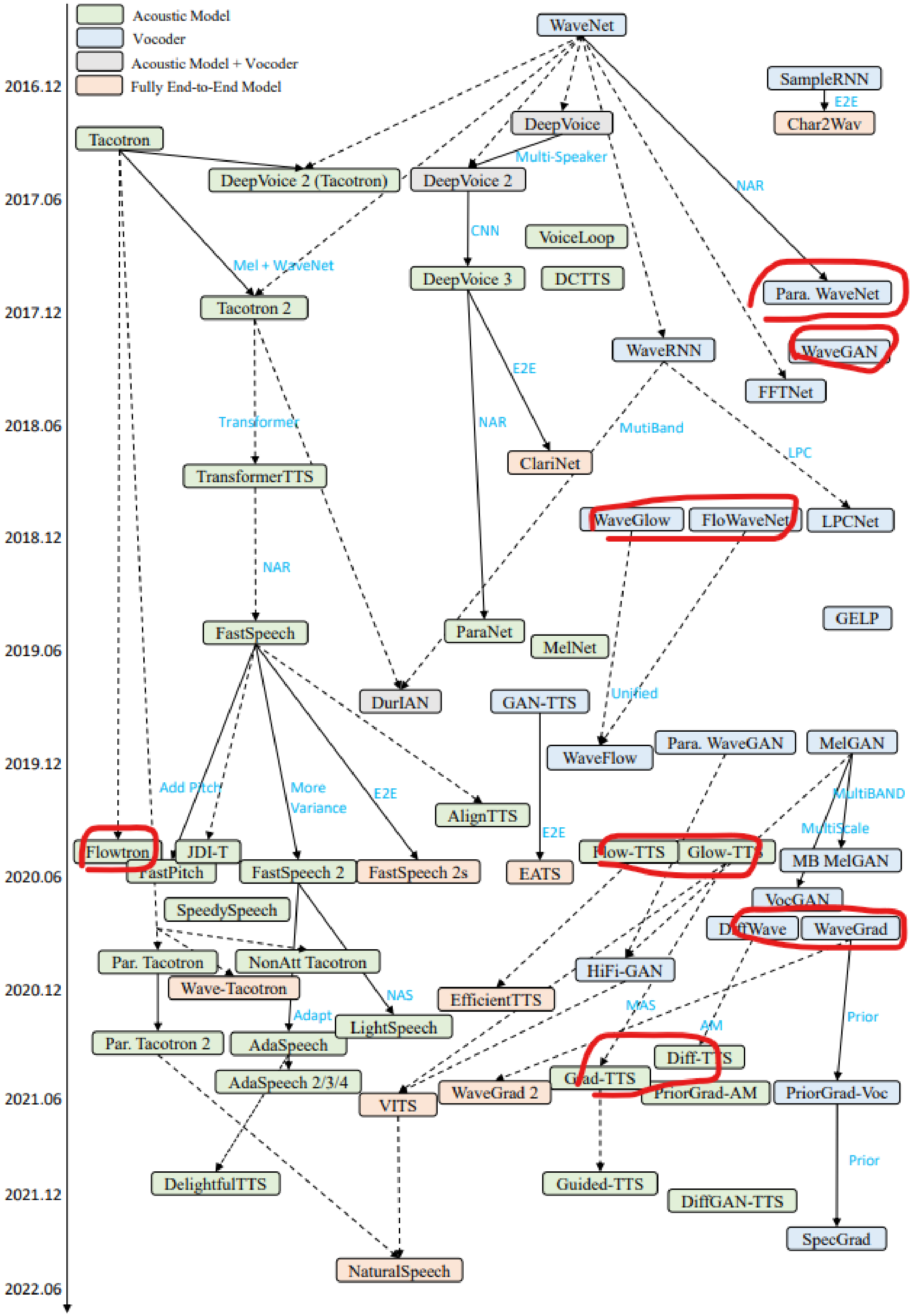
- WaveNet: autoregressive model with dilated causal convolution [254]



- 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 ☺


Acoustic Model
Vocoder
Acoustic Model + Vocoder
Fully End-to-End Model
2016.12
2017.06
2017.12
2018.06
2018.12
2019.06
2019.12
2020.06
2020.12
2021.06
2021.12
2022.06
WaveNet
SampleRNN
E2E
Char2Wav
DeepVoice
Tacotron
Multi-Speaker
DeepVoice 2 (Tacotron)
DeepVoice 2
NAR
CNN
VoiceLoop
Mel + WaveNet
DeepVoice 3
DCTTS
Para. WaveNet
Tacotron 2
WaveGAN
WaveRNN
E2E
FFTNet
Transformer
NAR
MutiBand
LPC
ClariNet
TransformerTTS
WaveGlow
FloWaveNet
LPCNet
NAR
GELP
FastSpeech
ParaNet
MelNet
DurIAN
GAN-TTS
Unified
Para. WaveGAN
MelGAN
WaveFlow
Add Pitch
More Variance
E2E
AlignTTS
MultiBAND
E2E
MultiScale
Flowtron
JDI-T
FastPitch
FastSpeech 2
FastSpeech 2s
EATS
Flow-TTS
Glow-TTS
MB MelGAN
VocGAN
SpeedySpeech
DiffWave
WaveGrad
Par. Tacotron
NonAtt Tacotron
HiFi-GAN
Wave-Tacotron
NAS
EfficientTTS
MAS
Adapt
LightSpeech
Prior
AM
Par. Tacotron 2
AdaSpeech
Diff-TTS
AdaSpeech 2/3/4
Grad-TTS
VITS
WaveGrad 2
PriorGrad-AM
PriorGrad-Voc
Prior
DelightfulTTS
Guided-TTS
DiffGAN-TTS
SpecGrad
NaturalSpeech

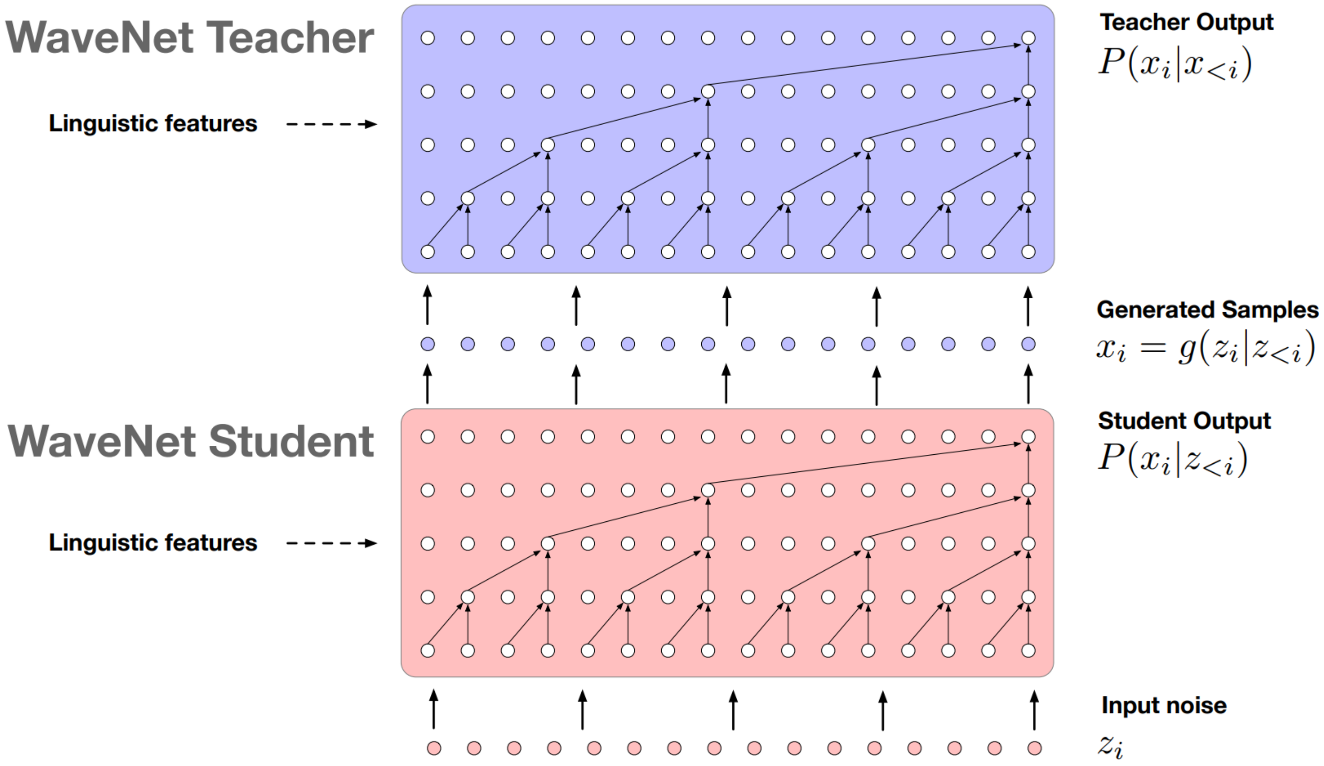
Generative models——Flow

- 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]	$z_t = x_t \cdot \sigma_t(x_{<t};\theta) + \mu_t(x_{<t};\theta)$	$x_t = \frac{z_t - u_t(x_{<t};\theta)}{\sigma_t(x_{<t};\theta)}$
	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
    - Parallel inference of IAF student
    - Parallel training of AF teacher



- Other works
  - ClariNet [269]

Generative models——Flow

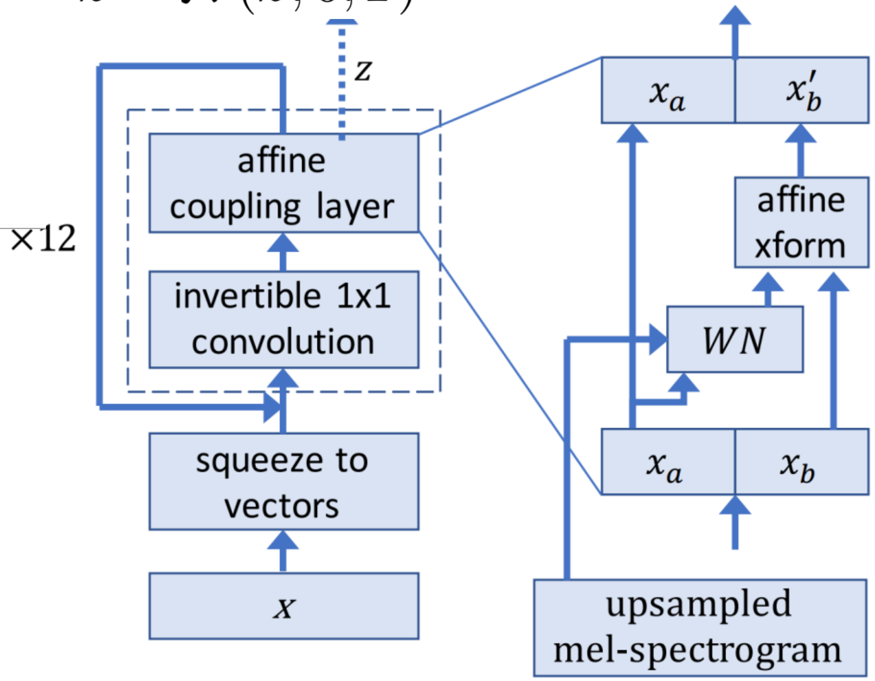
- WaveGlow [279] (Bipartite)
  - Flow based transformation

$$\boldsymbol{z} = \boldsymbol{f}_k^{-1} \circ \boldsymbol{f}_{k-1}^{-1} \circ \ldots \boldsymbol{f}_0^{-1}(\boldsymbol{x}) \qquad \boldsymbol{x} = \boldsymbol{f}_0 \circ \boldsymbol{f}_1 \circ \ldots \boldsymbol{f}_k(\boldsymbol{z}) \qquad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{z}; 0, \boldsymbol{I})$$

  - Affine Coupling Layer

$$\boldsymbol{x}_a, \boldsymbol{x}_b = split(\boldsymbol{x}) \qquad \boldsymbol{x}_b\prime = \boldsymbol{s} \odot \boldsymbol{x}_b + \boldsymbol{t}$$

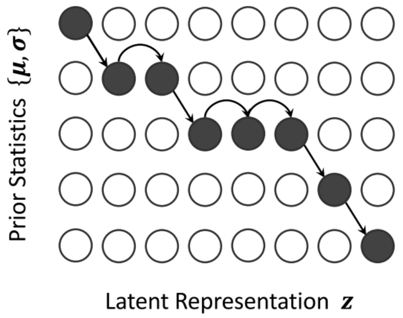
$$(\log \boldsymbol{s}, \boldsymbol{t}) = WN(\boldsymbol{x}_a, mel\text{-}spectrogram) \qquad \boldsymbol{f}_{coupling}^{-1}(\boldsymbol{x}) = concat(\boldsymbol{x}_a, \boldsymbol{x}_b\prime)$$



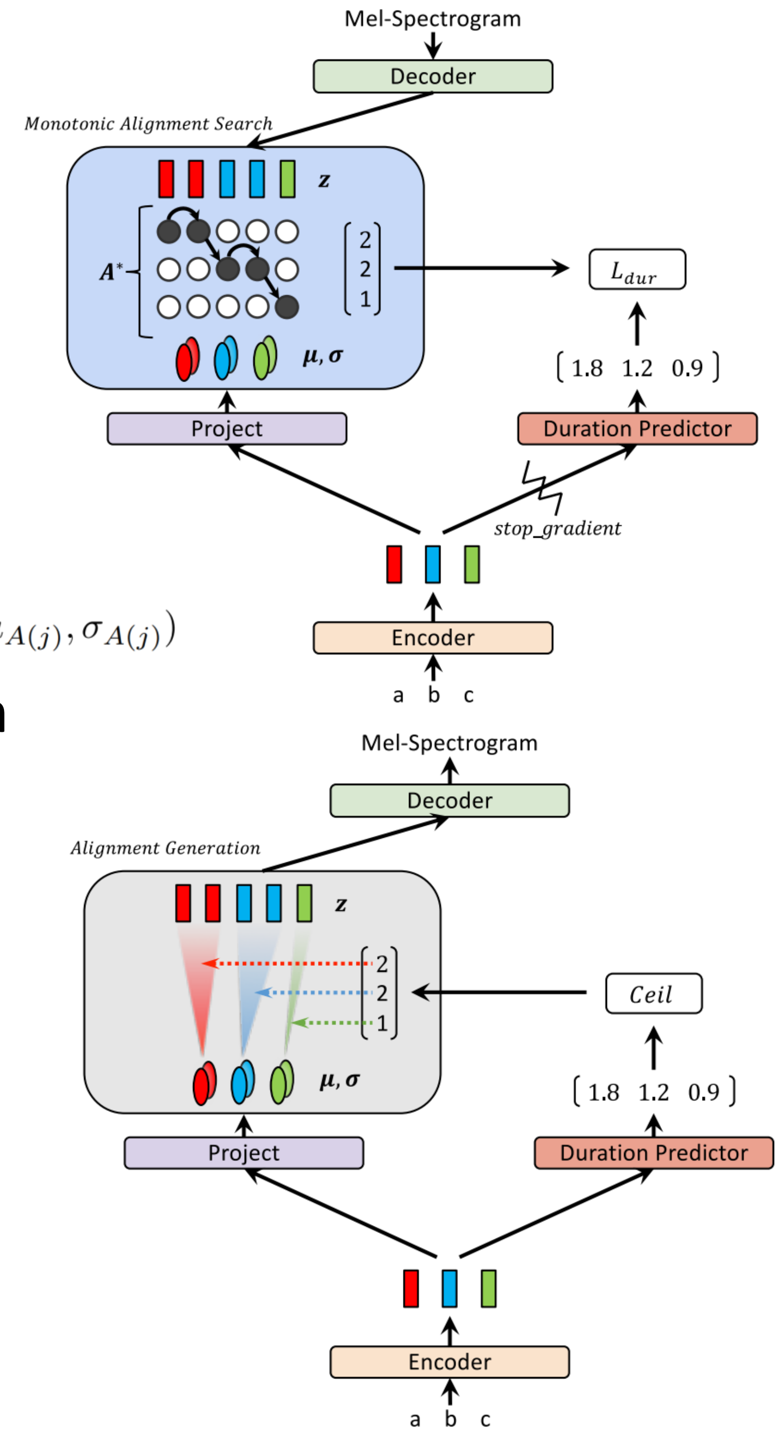
- Other works
  - FloWaveNet [163]
  - WaveFlow [271]

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

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

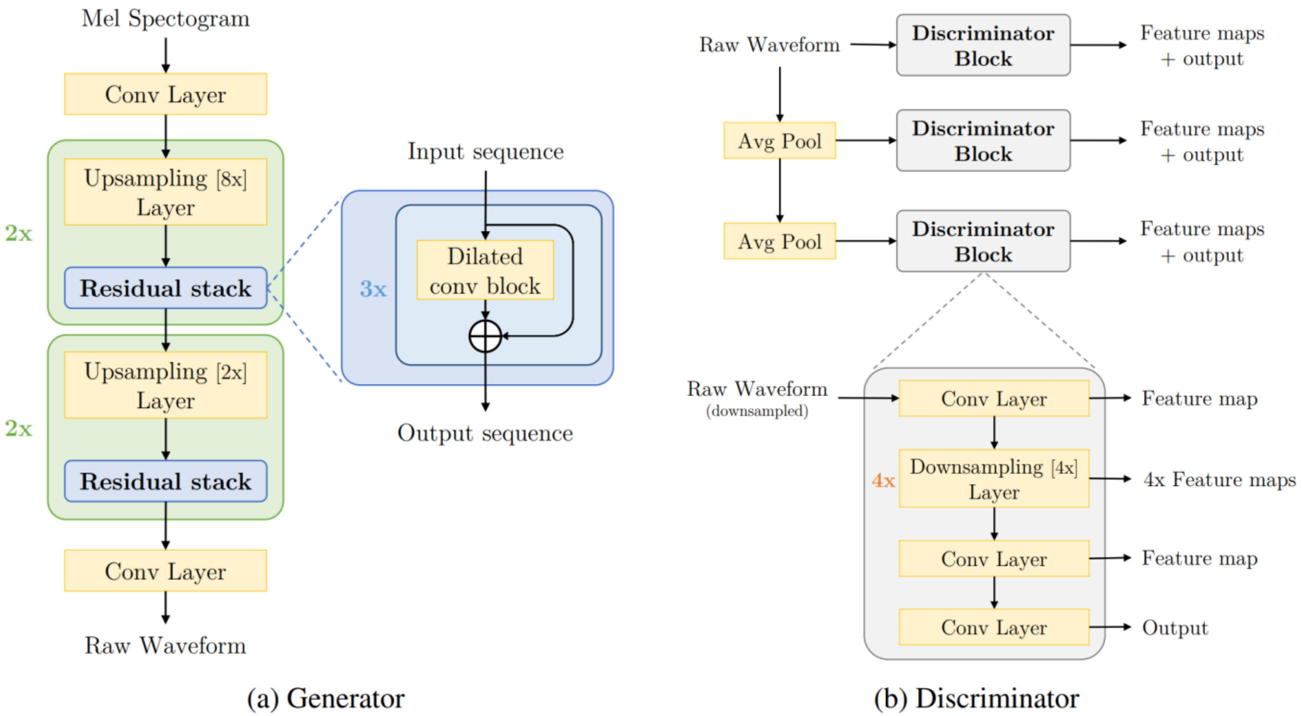
$$\mathcal{L}_{Adv}(G;D) = \mathbb{E}_{s}\left[(D(G(s))-1)^2\right]$$

- 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

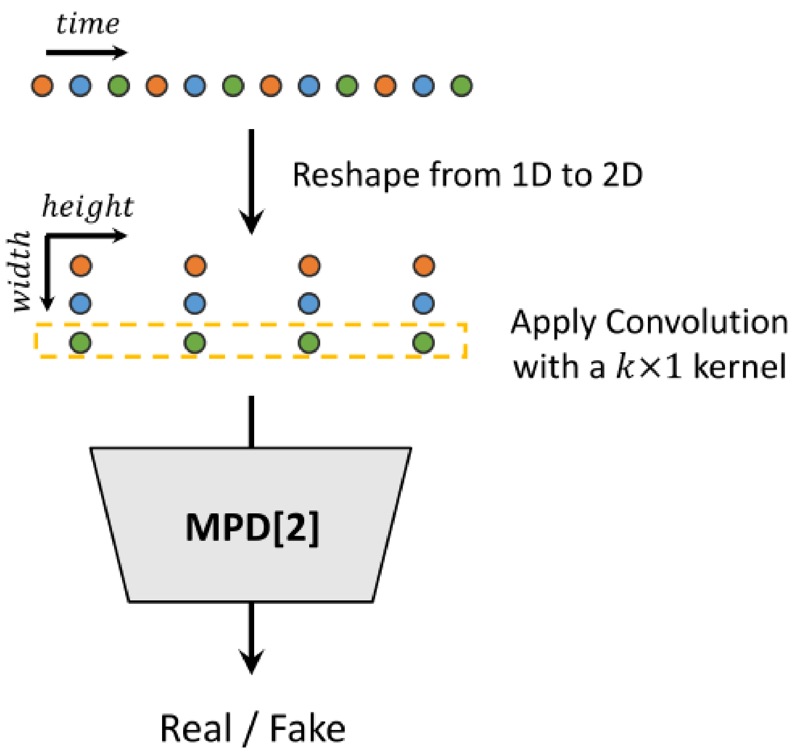
Generative models——GAN

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



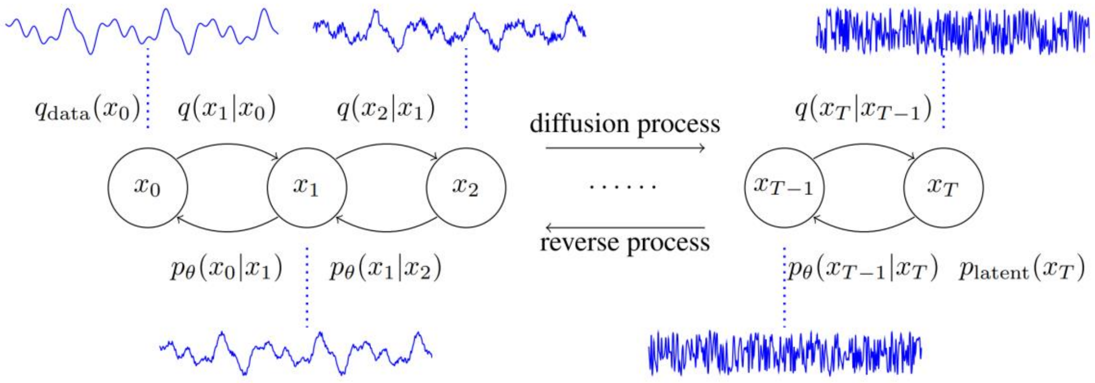
Generative models——GAN

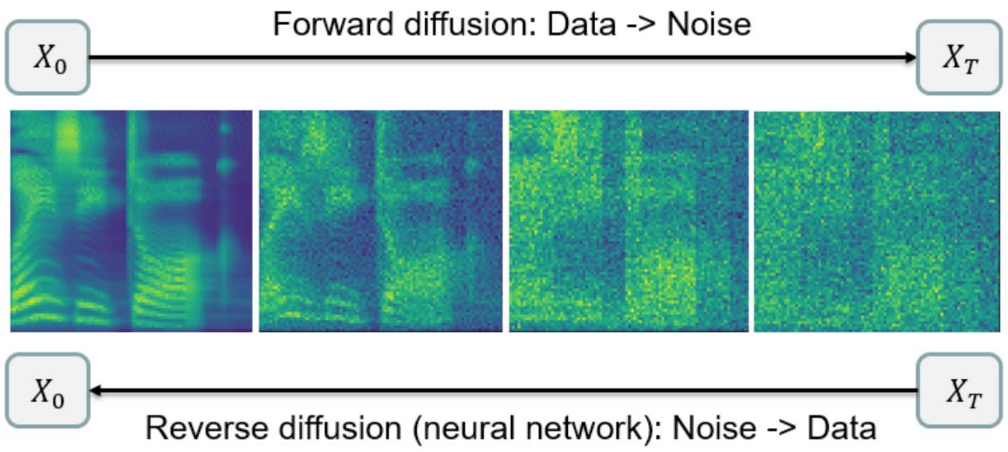
- HiFiGAN [68]
  - Multi-Scale Discriminator (MSD)
  - Multi-Period Discriminator (MPD)



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

Algorithm 1 Training

> **for** $i = 1, 2, \cdots, N_{\text{iter}}$ **do**
> Sample $x_0 \sim q_{\text{data}}$, $\epsilon \sim \mathcal{N}(0, I)$, and $t \sim \text{Uniform}(\{1, \cdots, T\})$
> 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** $t = T, T-1, \cdots, 1$ **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

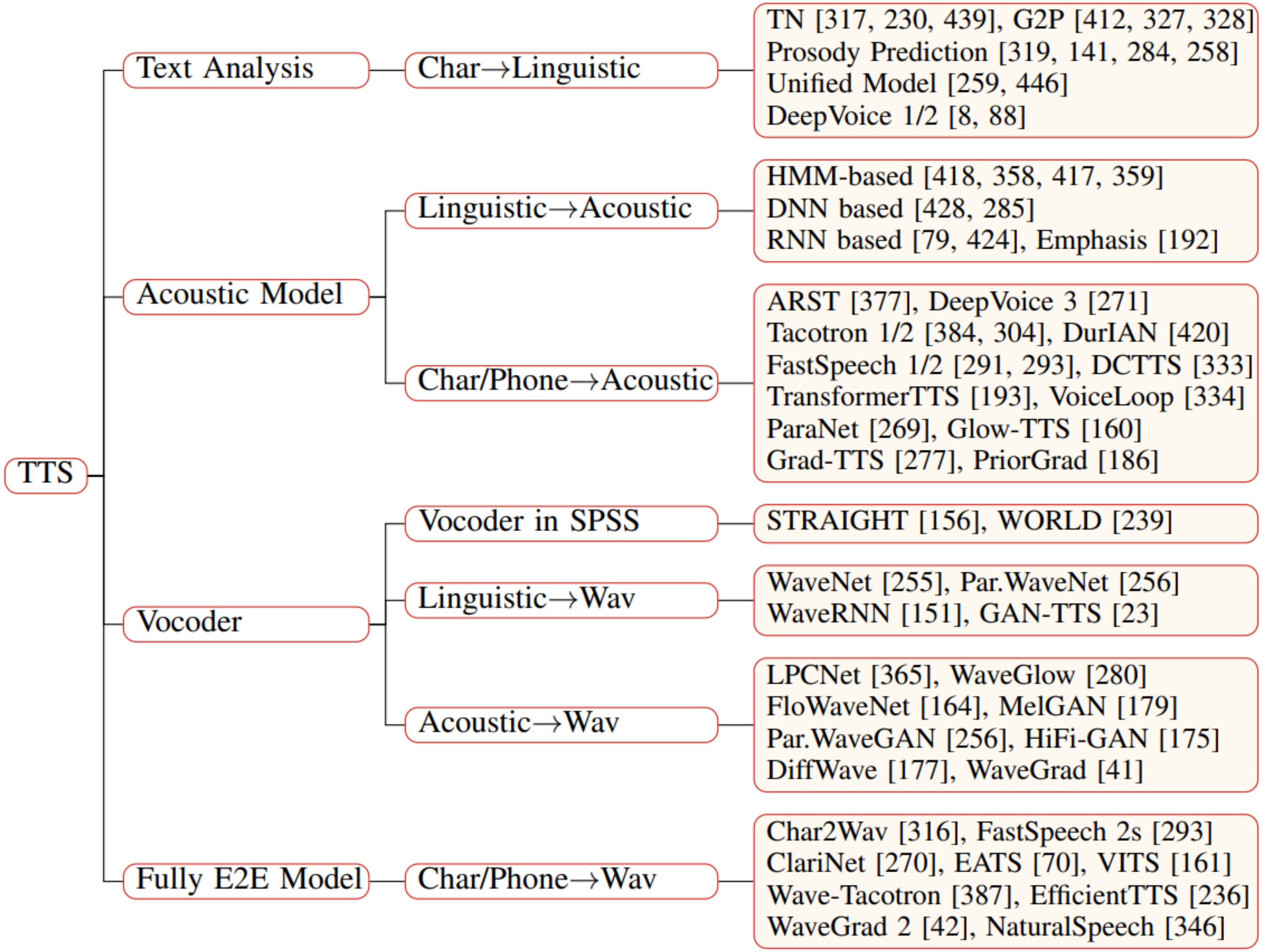
Generative models——Comparison

- A comparison among different generative models
  - Simplicity in math formulation and optimization
  - Support parallel generation
  - Support latent manipulation
  - Support likelihood estimation

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

GAN is weak in latent manipulation, since the condition in TTS is so strong, P(y|x) is not that much multi-modal compared to image synthesis

Key components in TTS

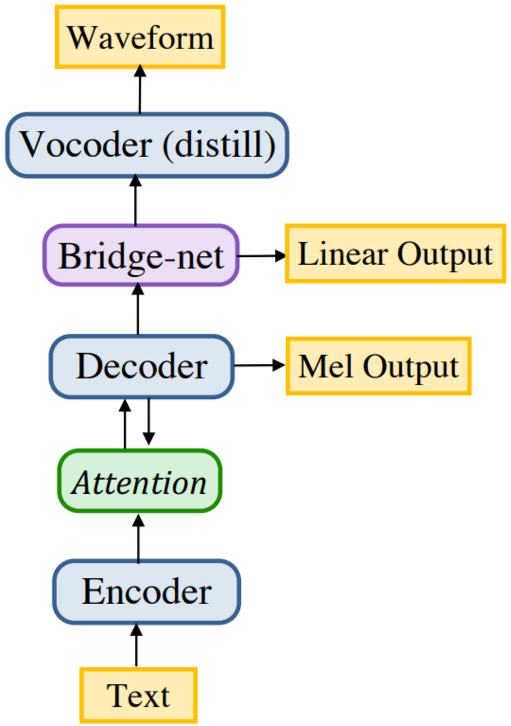


Fully End-to-End TTS

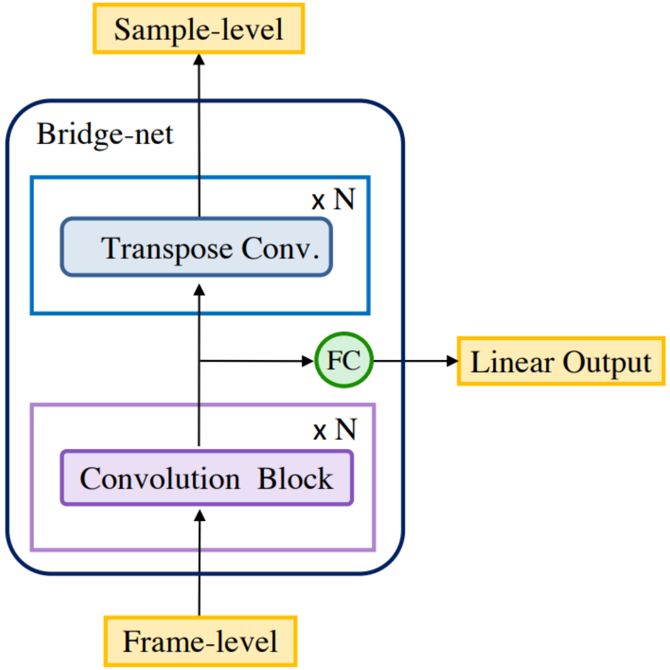
- Direct text/phoneme to waveform generation
- Advantages:
  - Fully differentiable optimization (towards the end goal)
  - Reduce cascaded errors (training/inference mismatch)
  - No mel-spectrogram bias (mel-spectrogram is not an optimal representation)

Fully End-to-End TTS

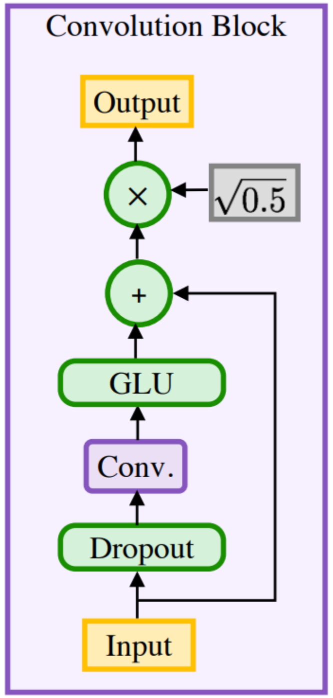
- ClariNet: AR acoustic model and NAR vocoder [269]



(a) Text-to-wave architecture



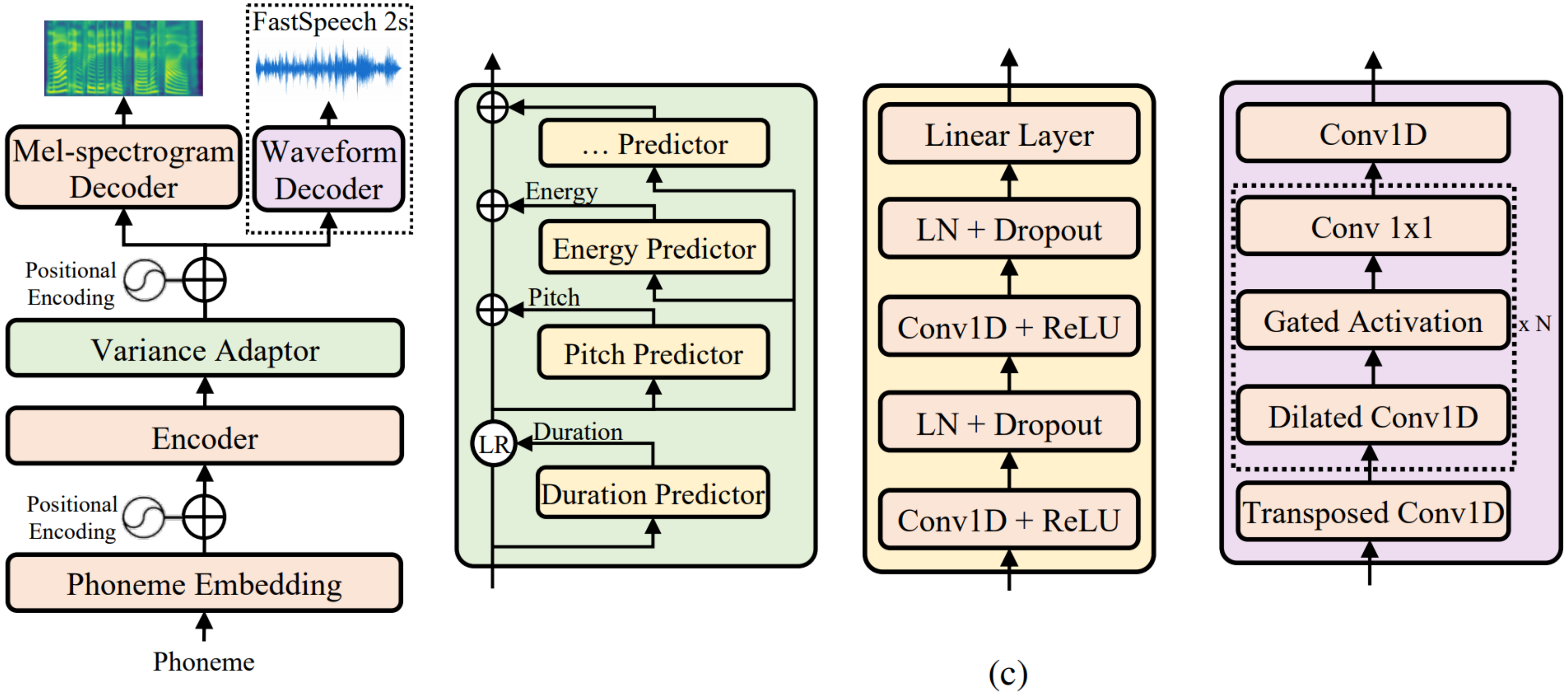
(b) Bridge-net



(c) Convolution block

Fully End-to-End TTS

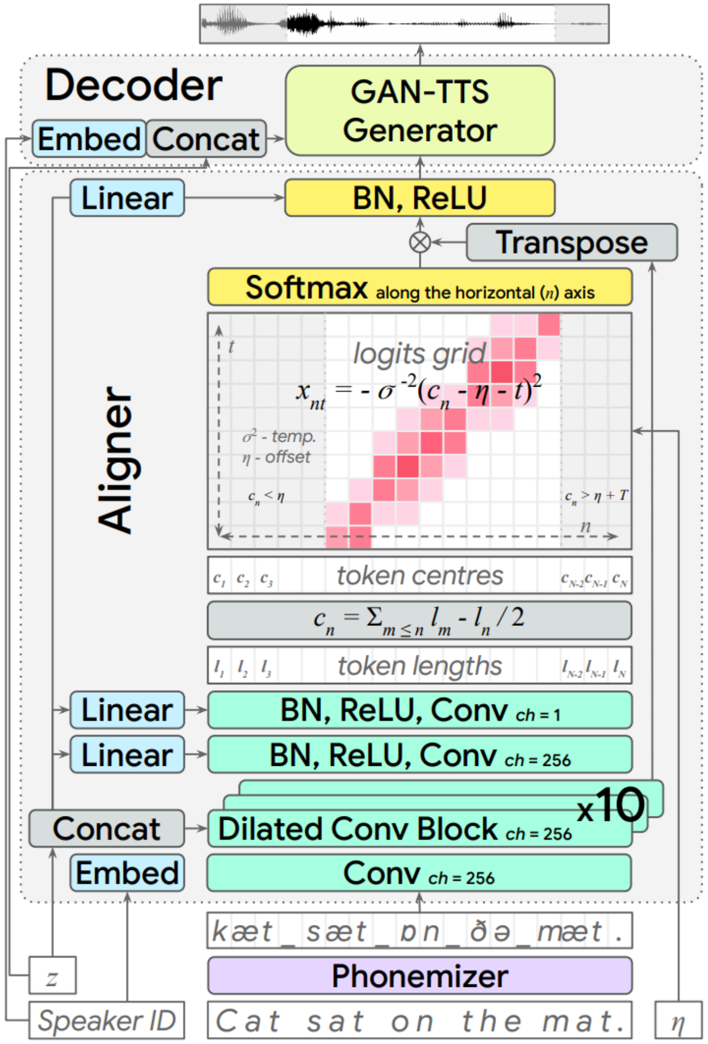
- FastSpeech 2s: fully parallel text to wave model [292]



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

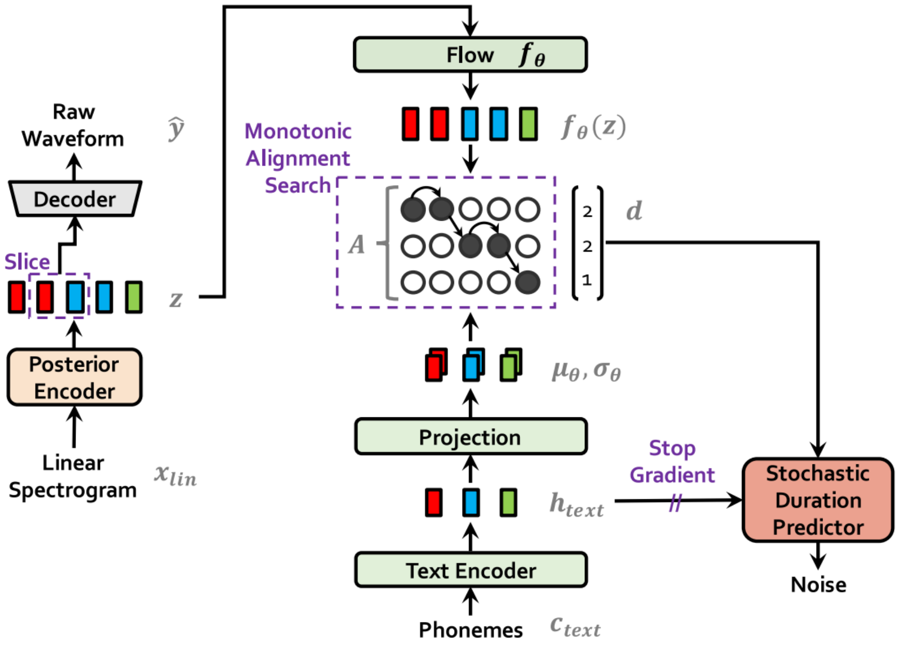
Fully End-to-End TTS

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

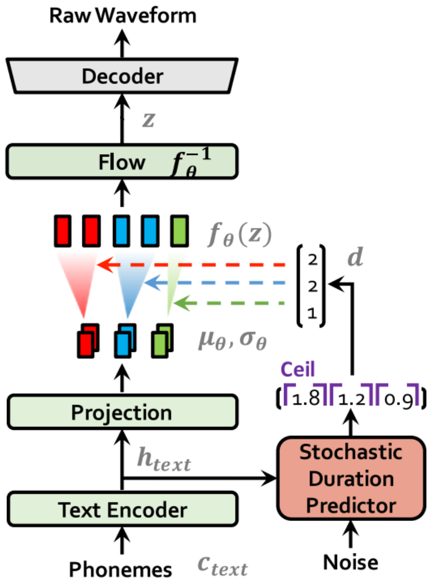


Fully End-to-End TTS

- VITS [160]
  - VAE, Flow, GAN
  - VAE: mel→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

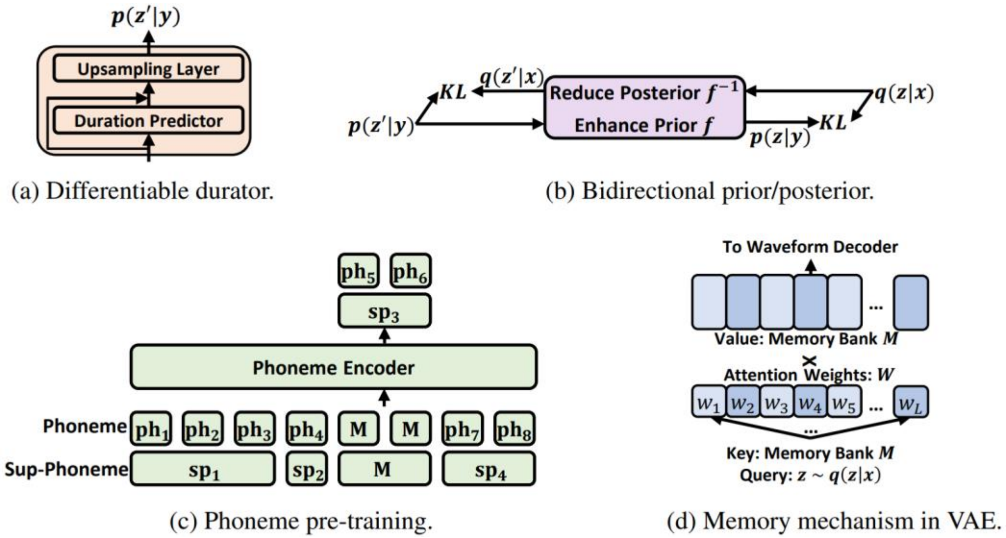
Fully End-to-End TTS

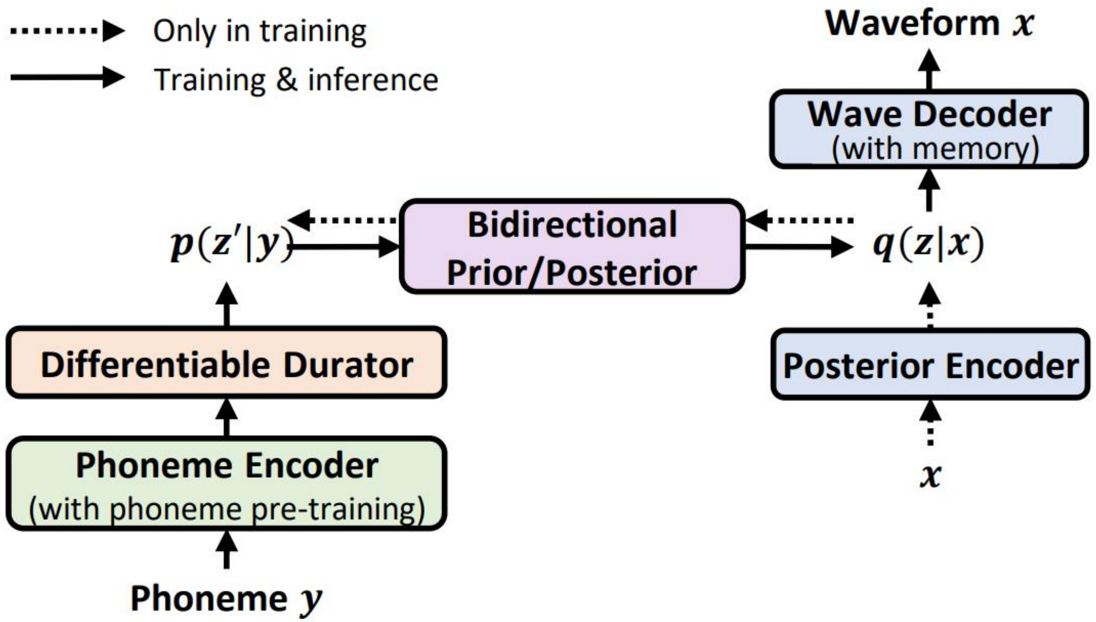
- NaturalSpeech: achieving human-level quality on LJSpeech dataset (CMOS)
- Leverage VAE to compress high-dimensional waveform x into frame-level representations $z \sim q(z|x)$, and is used to reconstruct waveform $x \sim p(x|z)$
- To enable text to waveform synthesis, z is predicted from y, $z \sim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

- 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





Fully End-to-End TTS

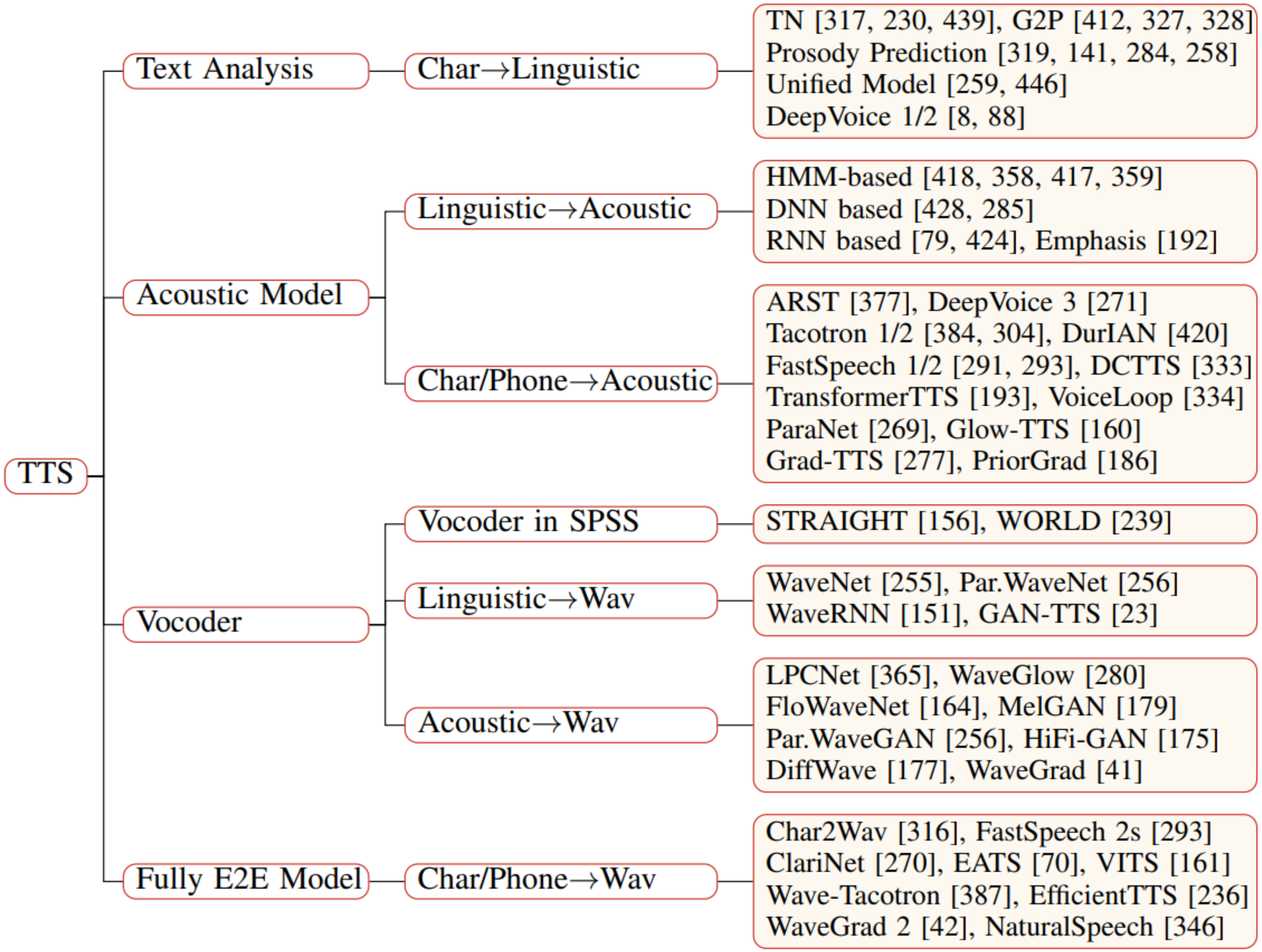
- Evaluations
  - MOS and CMOS on par with recordings, p-value >>0.05

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!

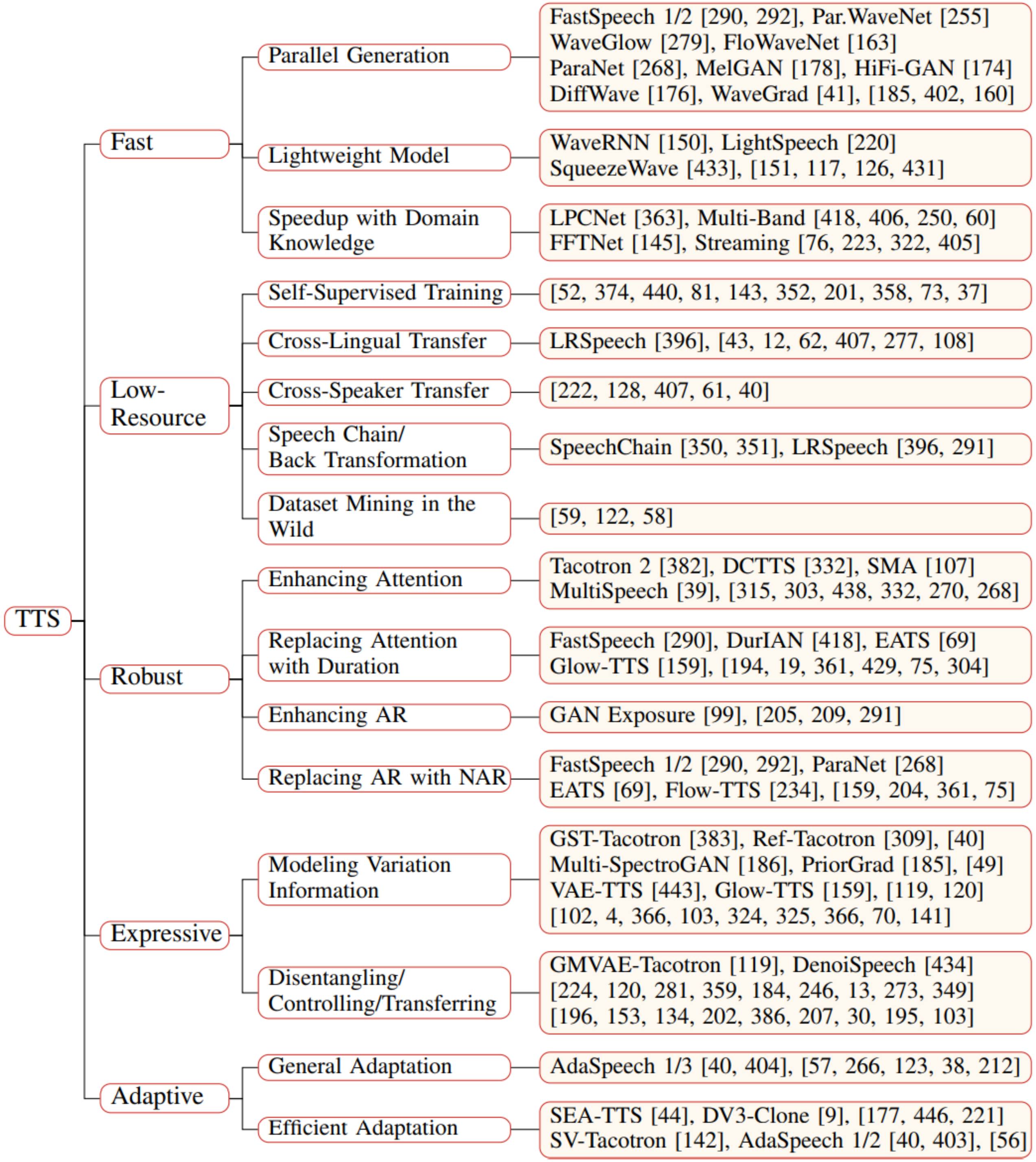
Key components in TTS



Part 3: Advanced Topics in TTS

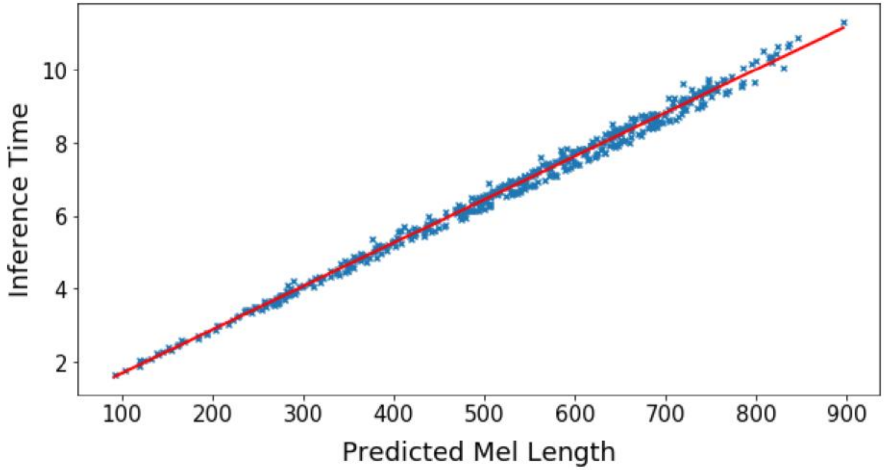
Advanced topics in TTS

- Fast TTS
- Low-resource TTS
- Robust TTS
- Expressive TTS
- Adaptive TTS



Fast TTS

- 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



- The model size is usually large
  - Slow in low-end GPU and edge device

Fast TTS

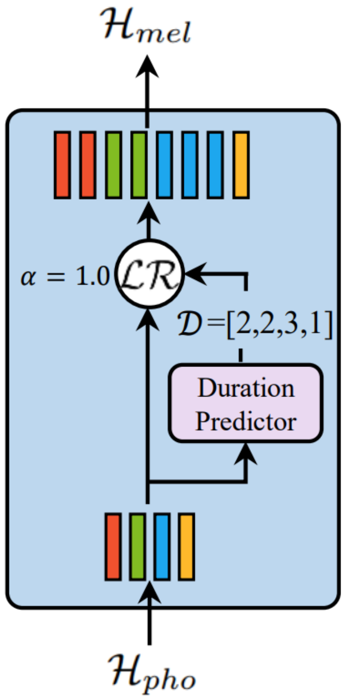
- Parallel generation

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	$\mathcal{O}(T)$	$\mathcal{O}(T)$

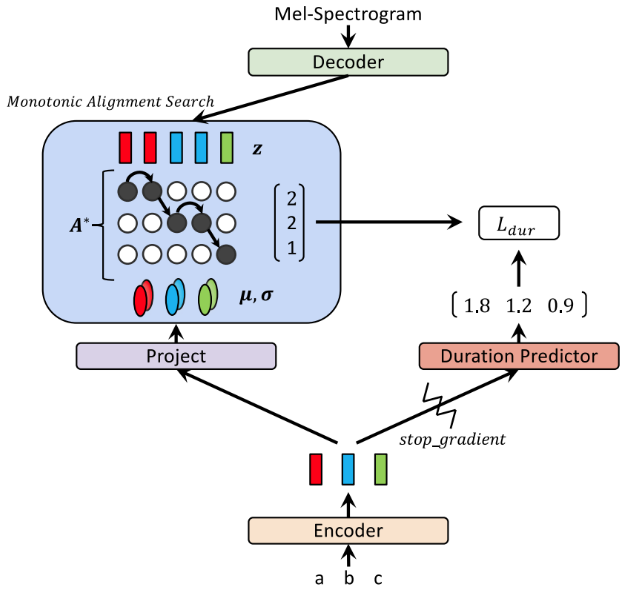
- Lightweight model
  - pruning, quantization, knowledge distillation, and neural architecture search
- Speedup with domain knowledge
  - linear prediction, multiband modeling, subscale prediction, multi-frame prediction, streaming synthesis

Fast TTS——Parallel generation

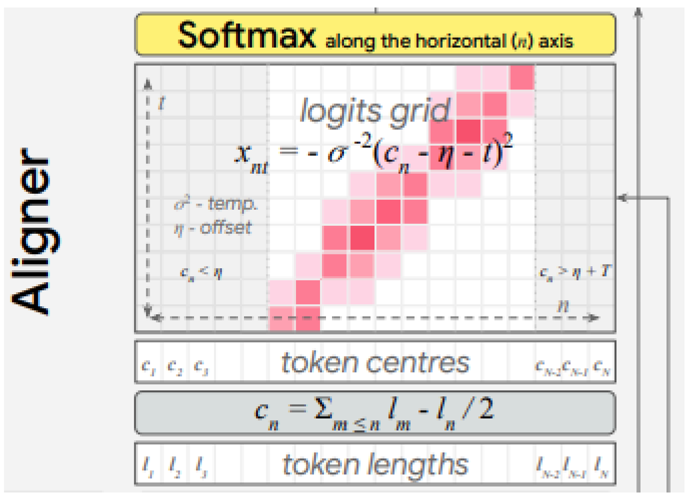
- The key is to bridge the length mismatch between text and speech



FastSpeech 1/2



Glow-TTS



EATS

Fast TTS——Parallel generation

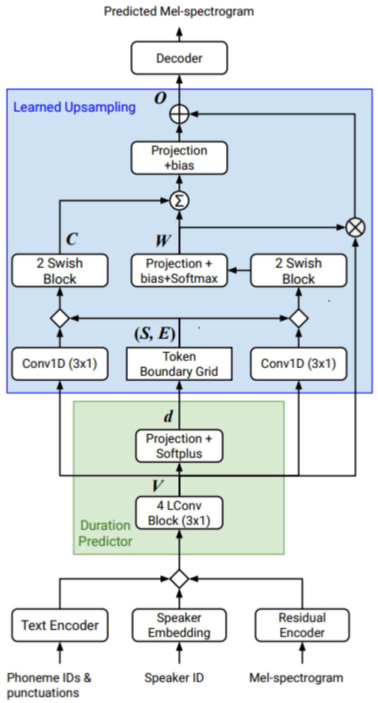
- The key is to bridge the length mismatch between text and speech

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

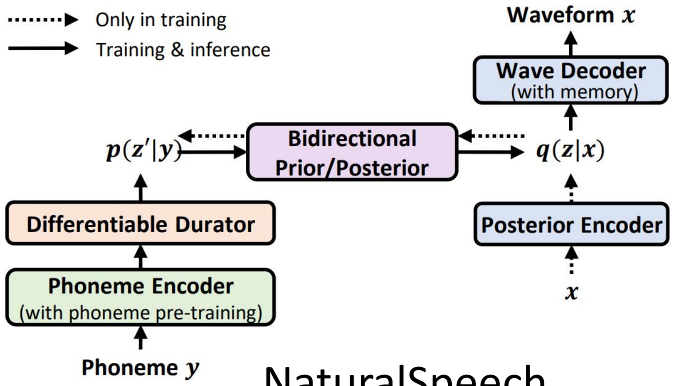
$$W = \text{Softmax}(\underset{10\to q}{\text{MLP}}([S, E, \text{Expand}(\text{Conv1D}(\text{Proj}(H)))])),$$

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

$$O = \underset{qh\to h}{\text{Proj}}(WH) + \underset{qp\to h}{\text{Proj}}(\text{Einsum}(W, C))$$



Parallel Tacotron 2



NaturalSpeech

Low-resource TTS

- There are **7,000+** languages in the world, but popular commercialized speech services only support **dozens or hundreds of** languages
  - There is strong business demand to support more languages in TTS.

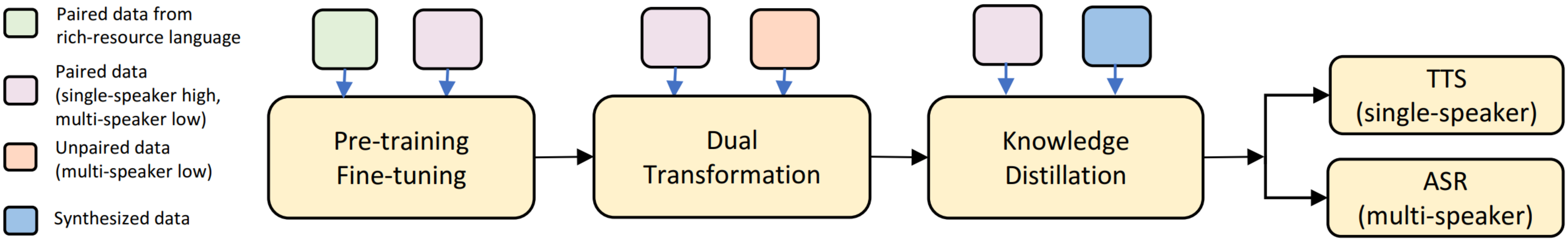
- However, lack of data in low-resource languages and the data collection cost is high.

Low-resource TTS

Techniques	Data	Work
Self-supervised Training	Unpaired text or speech	[52, 374, 440, 81, 143, 352, 201, 358, 73]
Cross-lingual Transfer	Paired text and speech	[43, 396, 12, 407, 62, 277, 108]
Cross-speaker Transfer	Paired text and speech	[222, 128, 61, 407, 40]
Speech chain/Back transformation	Unpaired text or speech	[291, 396, 350, 351]
Dataset mining in the wild	Paired text and speech	[59, 122, 58]

- Self-supervised training
  - Text pre-training, speech pre-training, discrete token quantization
- Cross-lingual transfer
  - Languages share similarity, phoneme mapping/re-initialization/IPA/byte
- Cross-speaker transfer
  - Voice conversion, voice adaptation
- Speech chain/back transformation
  - TTS ←→ASR
- Dataset mining in the wild
  - Speech enhancement, denoising, disentangling

Low-resource TTS——LRSpeech [396]



- **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

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.

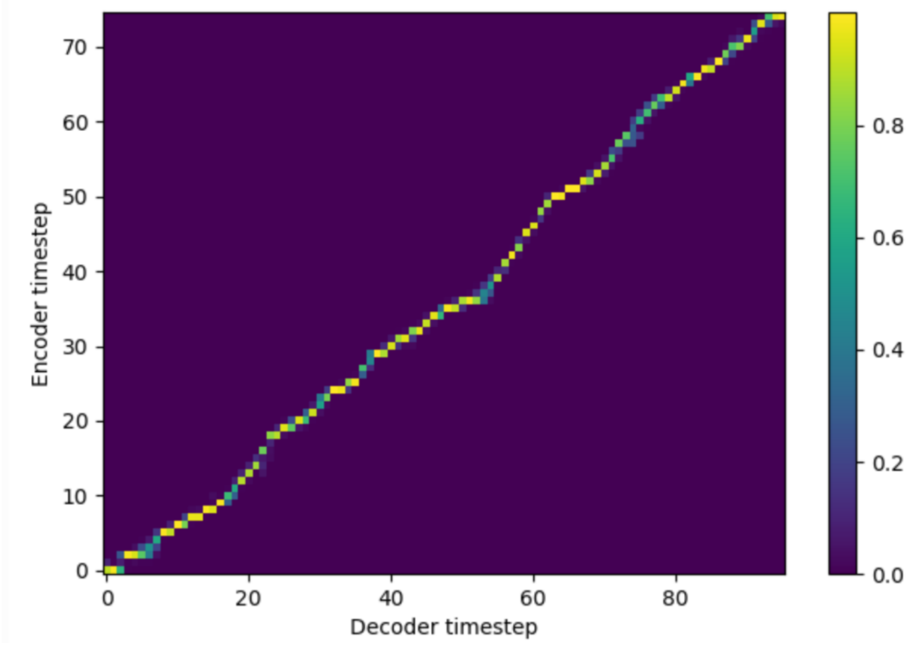
- The cause of robustness issues
  - The difficulty of alignment learning between text and mel-spectrograms
  - Exposure bias and error propagation in AR generation
- The solutions
  - Enhance attention
  - Replace attention with duration prediction
  - Enhance AR
  - Replace AR with NAR

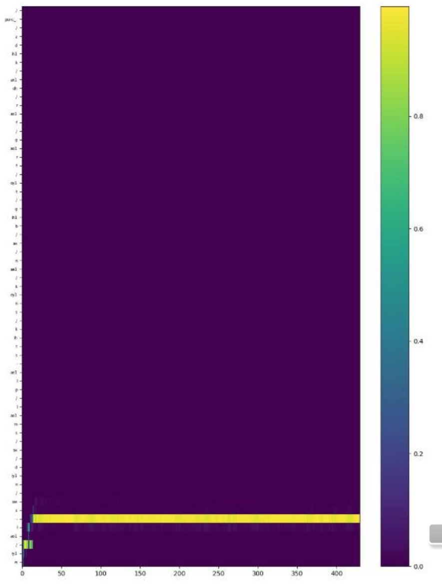
Robust TTS

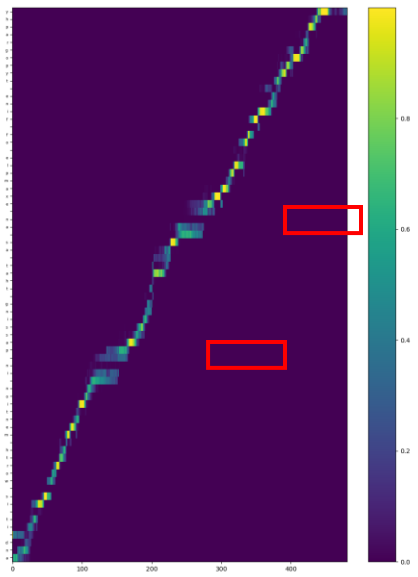
Category	Technique	Work
Enhancing Attention	Content-based attention	[382, 192]
	Location-based attention	[315, 333, 367, 17]
	Content/Location hybrid attention	[303]
	Monotonic attention	[438, 107, 411]
	Windowing or off-diagonal penalty	[332, 438, 270, 39]
	Enhancing enc-dec connection	[382, 303, 270, 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



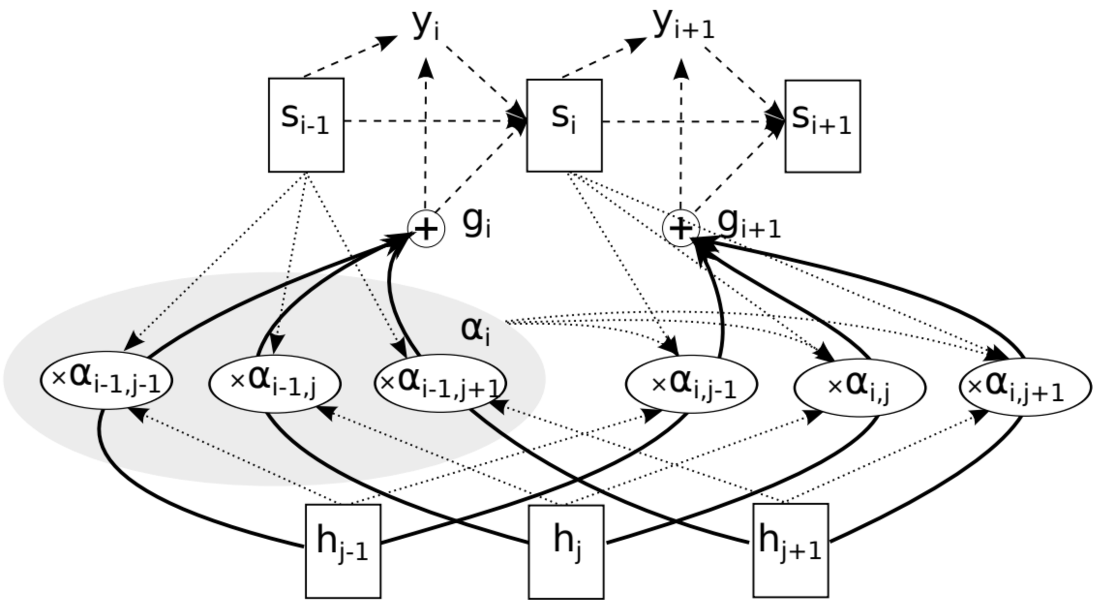




And it is worth mention in passing that, as an example of fine typography

Robust TTS——Attention improvement

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



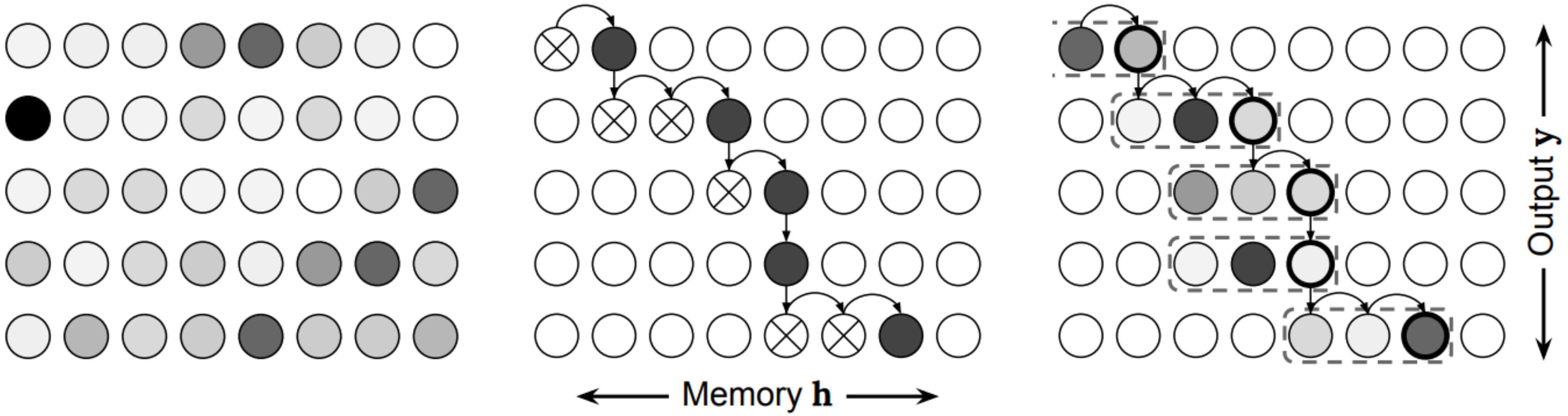
$$\alpha_i = Attend(s_{i-1}, \alpha_{i-1}, h)$$

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

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

Robust TTS——Attention improvement

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



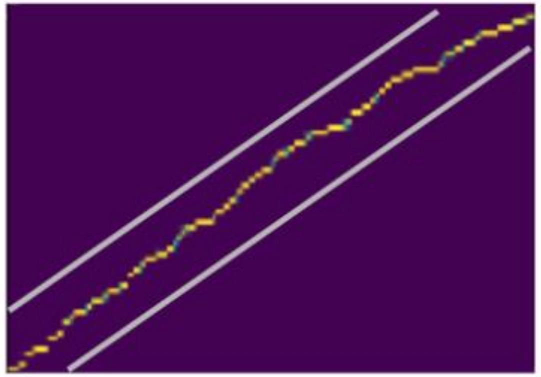
(a) Soft attention. (b) Hard monotonic attention. (c) Monotonic chunkwise attention.

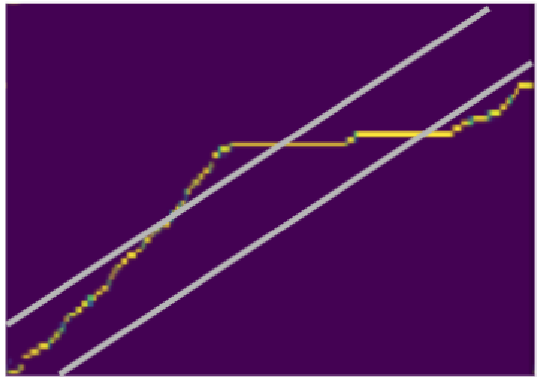
$$e_{i,j} = \mathrm{MonotonicEnergy}(s_{i-1}, h_j)$$
$$p_{i,j} = \sigma(e_{i,j})$$
$$z_{i,j} \sim \mathrm{Bernoulli}(p_{i,j})$$

Robust TTS——Attention improvement

- Windowing [332, 438]
  - Only a subset of the encoding results $\hat{\boldsymbol{x}} = [\boldsymbol{x}_{p-w}, ..., \boldsymbol{x}_{p+w}]$ are considered at each decoder timestep when using the windowing technique

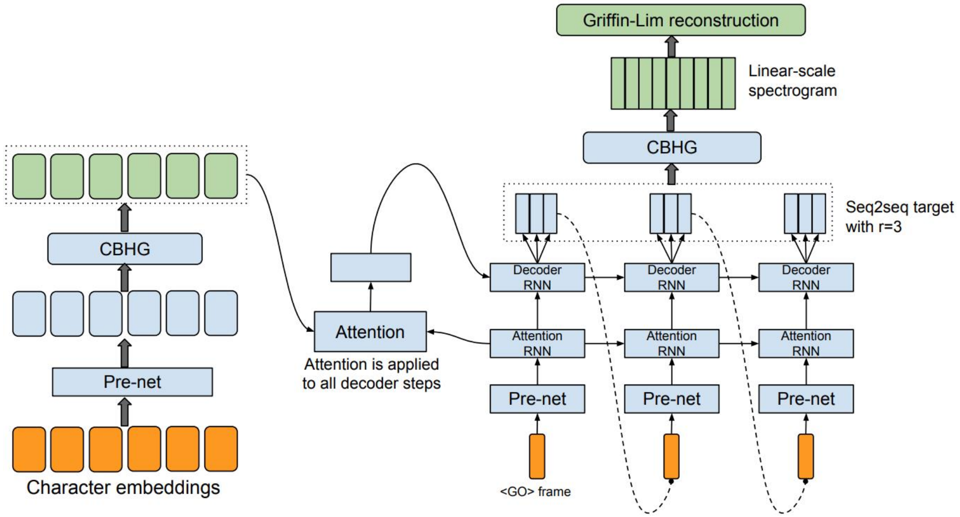
- Penalty loss for off-diagonal attention distribution [39]
  - Guided attention loss with diagonal band mask

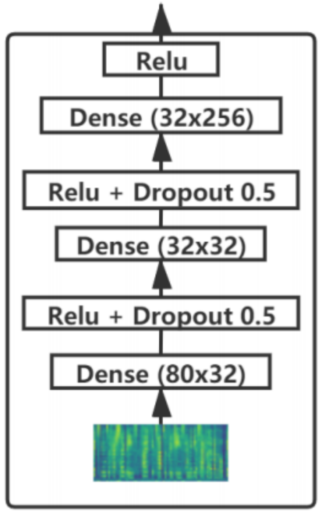




Robust TTS——Attention improvement

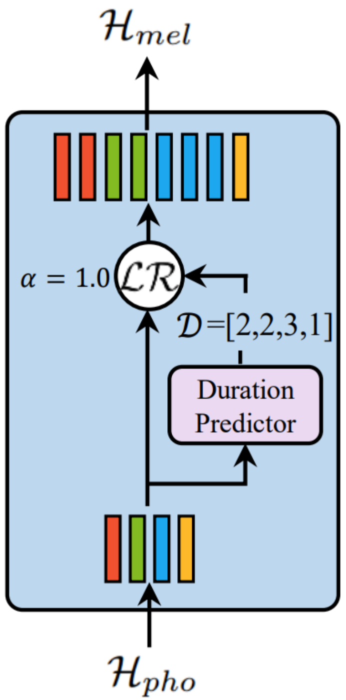
- Multi-frame prediction [382]
  - Predicting multiple, non-overlapping output frames at each decoder step
  - Increase convergence speed, with a much faster (and more stable) alignment learned from attention
- Decoder prenet dropout/bottleneck [382,39]
  - 0.5 dropout, small hidden size as bottleneck



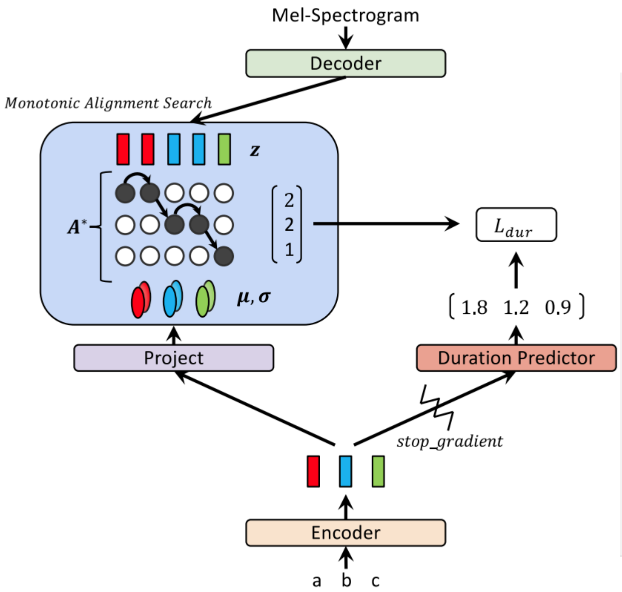


Robust TTS——Durator

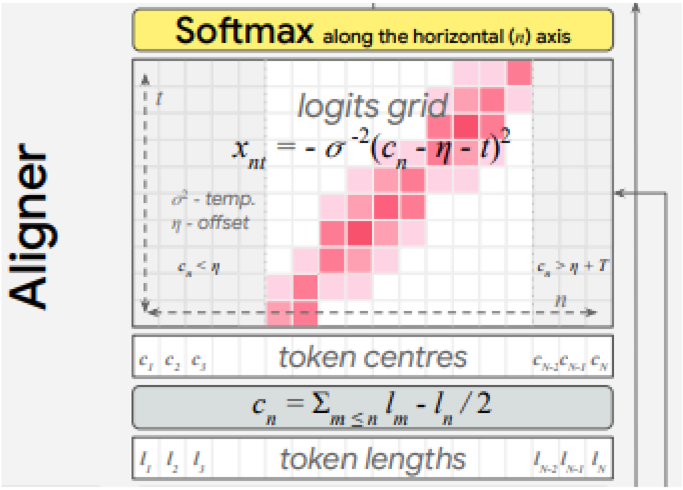
- Duration prediction and expansion
  - SPSS → Seq2Seq model with attention → Non-autoregressive model
  - Duration → attention, no duration → duration prediction (technique renaissance)



FastSpeech 1/2



Glow-TTS



EATS

Robust TTS——Durator

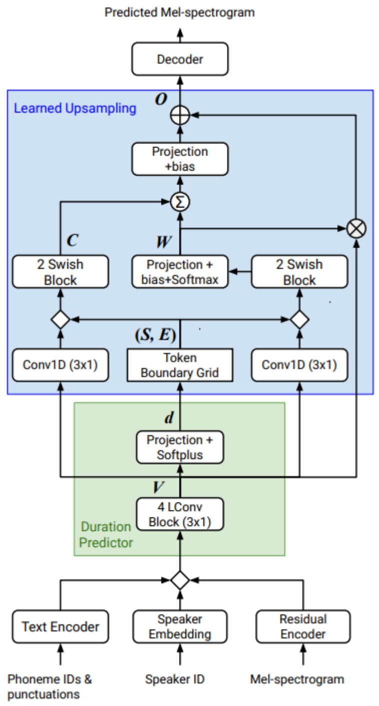
- Differentiable duration modeling

$$\boldsymbol{S}_{i,j} = i - \sum_{k=1}^{j-1} d_k, \quad \boldsymbol{E}_{i,j} = \sum_{k=1}^{j} d_k - i, \qquad \boldsymbol{S}_{m\times n} \qquad \boldsymbol{E}_{m\times n}$$

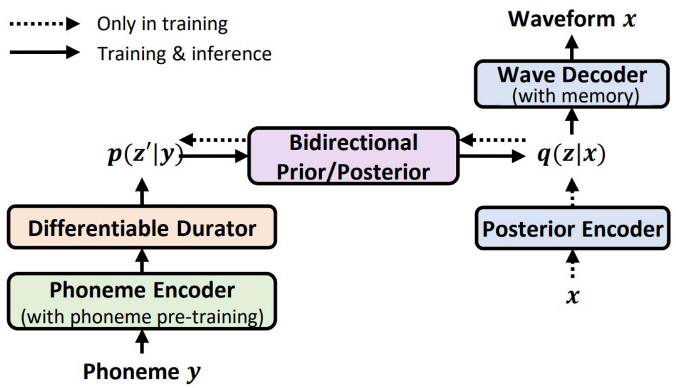
$$\boldsymbol{W} = \text{Softmax}(\underset{10\to q}{\text{MLP}}([\boldsymbol{S}, \boldsymbol{E}, \text{Expand}(\text{Conv1D}(\text{Proj}(\boldsymbol{H})))])),$$

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

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



Parallel Tacotron 2



NaturalSpeech

Robust TTS

- A new taxonomy of TTS

Attention? \ AR?	AR	Non-AR
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]

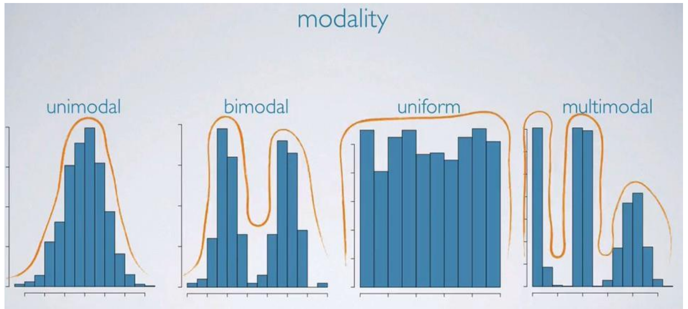
Expressive TTS

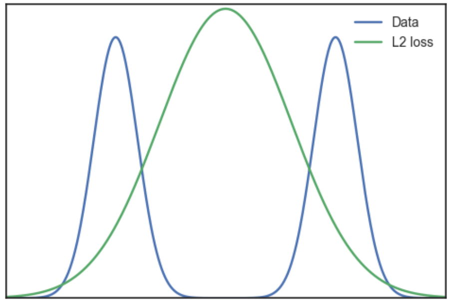
- Expressiveness
  - Characterized by content (what to say), speaker/timbre (who to say), prosody/emotion/style (how to say), noisy environment (where to say), etc
- Over-smoothing prediction
  - One to many mapping in text to speech: p(y|x) multimodal distribution

Text

multiple speech variations

(duration, pitch, sound volume, speaker, style, emotion, etc)





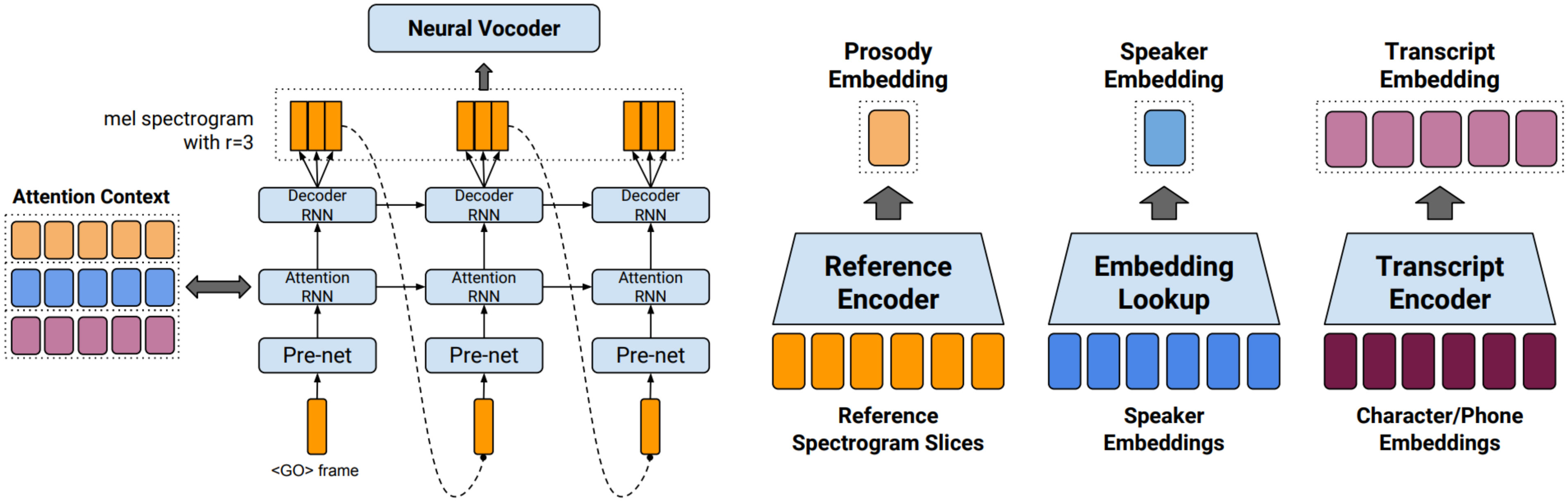
Expressive TTS

- 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]
		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	Fine-grained information	[325, 116, 45, 335]
	Character/Phoneme Level		[188, 324, 430, 325, 45, 40, 189]
	Frame Level		[188, 158, 49, 434]

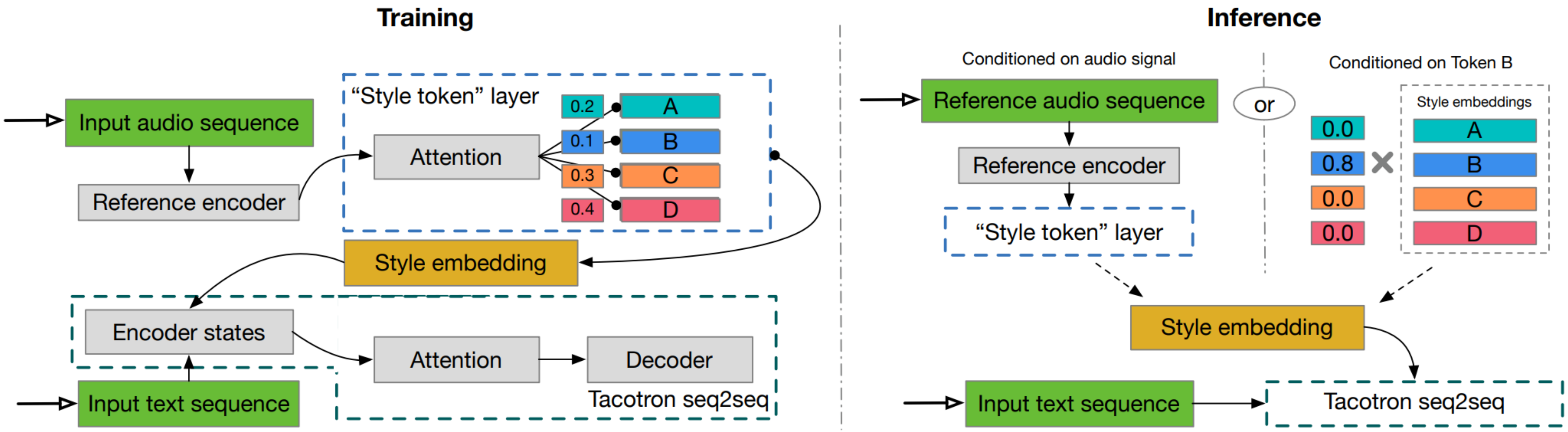
Expressive TTS——Reference encoder

- 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



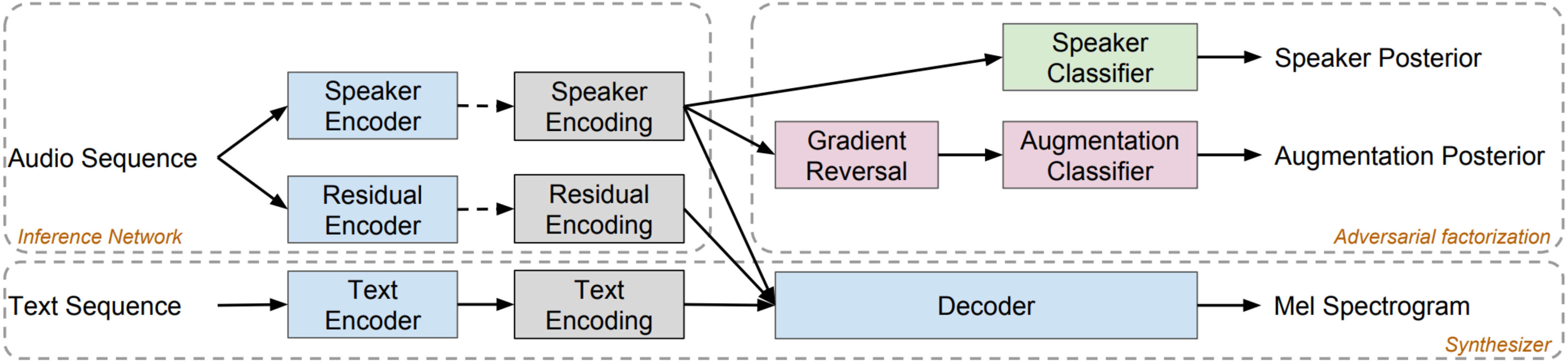
Expressive TTS——Disentangling, Controlling and Transferring

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

Technique	Description	Work
Disentangling with Adversarial Training	Disentanglement for control	[224, 120, 281, 434]
Cycle Consistency/Feedback for Control	Enhance style/timbre generation	[202, 386, 207, 30, 195]
Semi-Supervised Learning for Control	Use VAE and adversarial training	[103, 119, 120, 434, 302]
Changing Variance Information for Transfer	Different information in inference	[309, 383, 142, 443, 40]

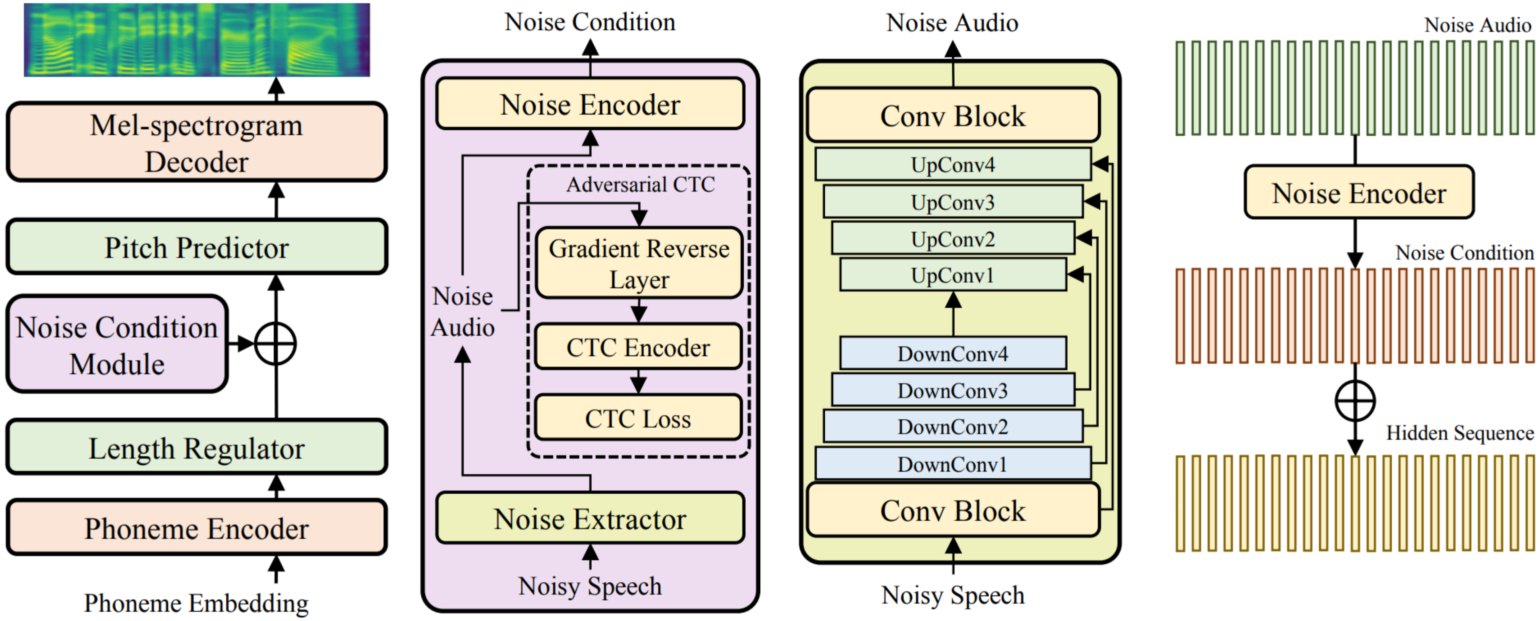
Expressive TTS——Disentangling, Controlling and Transferring

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



Expressive TTS——Disentangling, Controlling and Transferring

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



(a) DenoiSpeech (b) Noise Condition Module (c) Noise Extractor (d) Noise Encoder

Adaptive TTS

- Voice adaptation, voice cloning, custom voice
- Empower TTS for everyone
  - Pre-training on multi-speaker TTS model
  - Fine-tuning on speech data from target speaker
  - Inference speech for target speaker
- 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

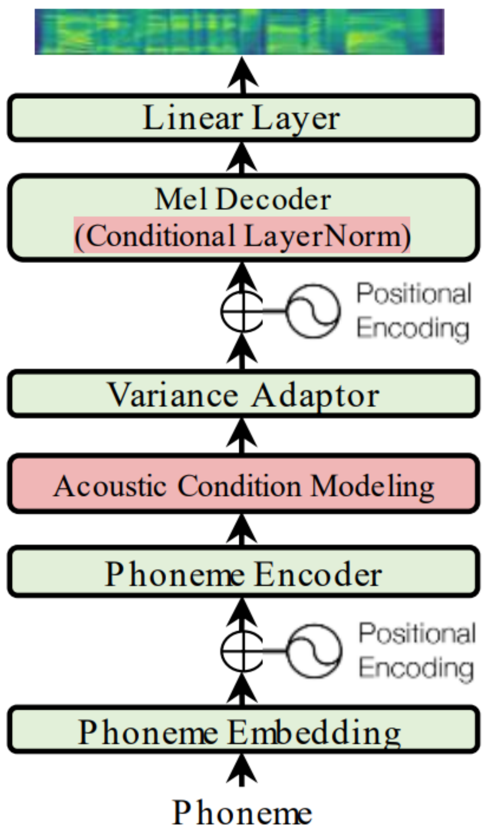
Adaptive TTS

- A taxonomy on adaptive TTS

Category	Topic	Work
General Adaptation	Modeling Variation Information	[40]
	Increasing Data Coverage	[57, 407]
	Cross-Acoustic Adaptation	[40, 54]
	Cross-Style Adaptation	[404, 266, 123]
	Cross-Lingual Adaptation	[445, 38, 212]
Efficient Adaptation	Few-Data Adaptation	[44, 9, 177, 240, 446, 49, 40, 236]
	Untranscribed Data Adaptation	[403, 133, 221]
	Few-Parameter Adaptation	[9, 44, 40]
	Zero-Shot Adaptation	[9, 44, 142, 56]

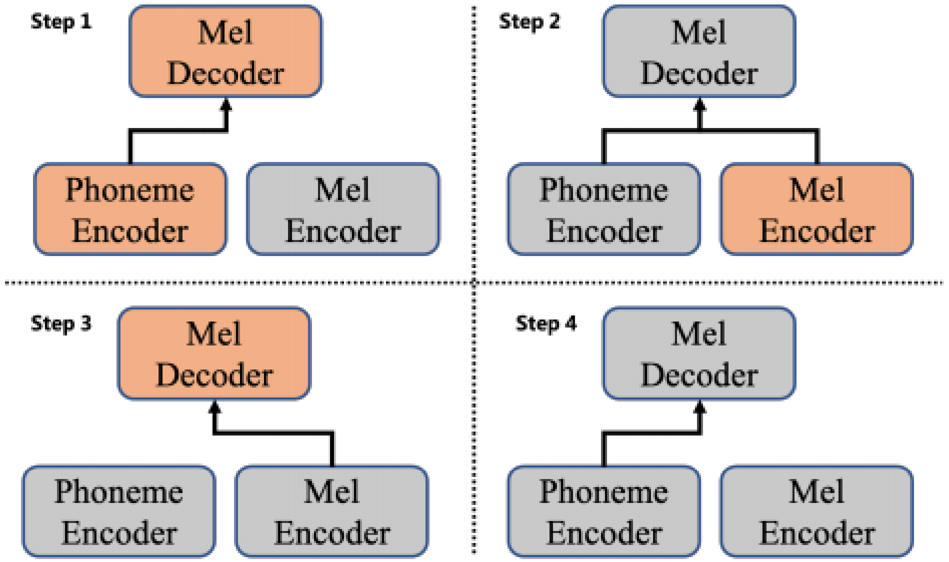
Adaptive TTS——AdaSpeech [40]

- 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



Adaptive TTS——AdaSpeech 2 [403]

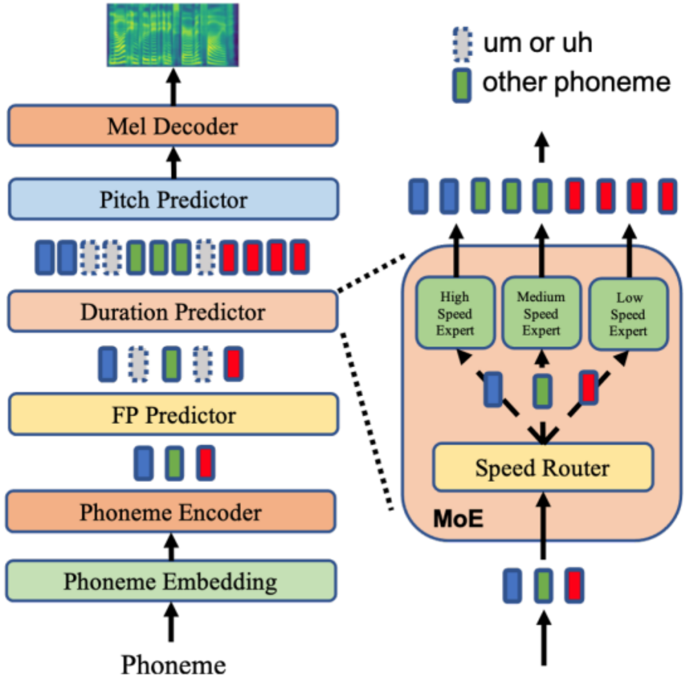
- Only untranscribed data, how to adapt?
  - In online meeting, only speech can be collected, without corresponding transcripts
- AdaSpeech 2, speech reconstruction with latent alignment
  - Step 1: source TTS model training
  - Step 2: speech reconstruction
  - Step 3: speaker adapatation
  - Step 4: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



Adaptive TTS——AdaSpeech 3 [404]

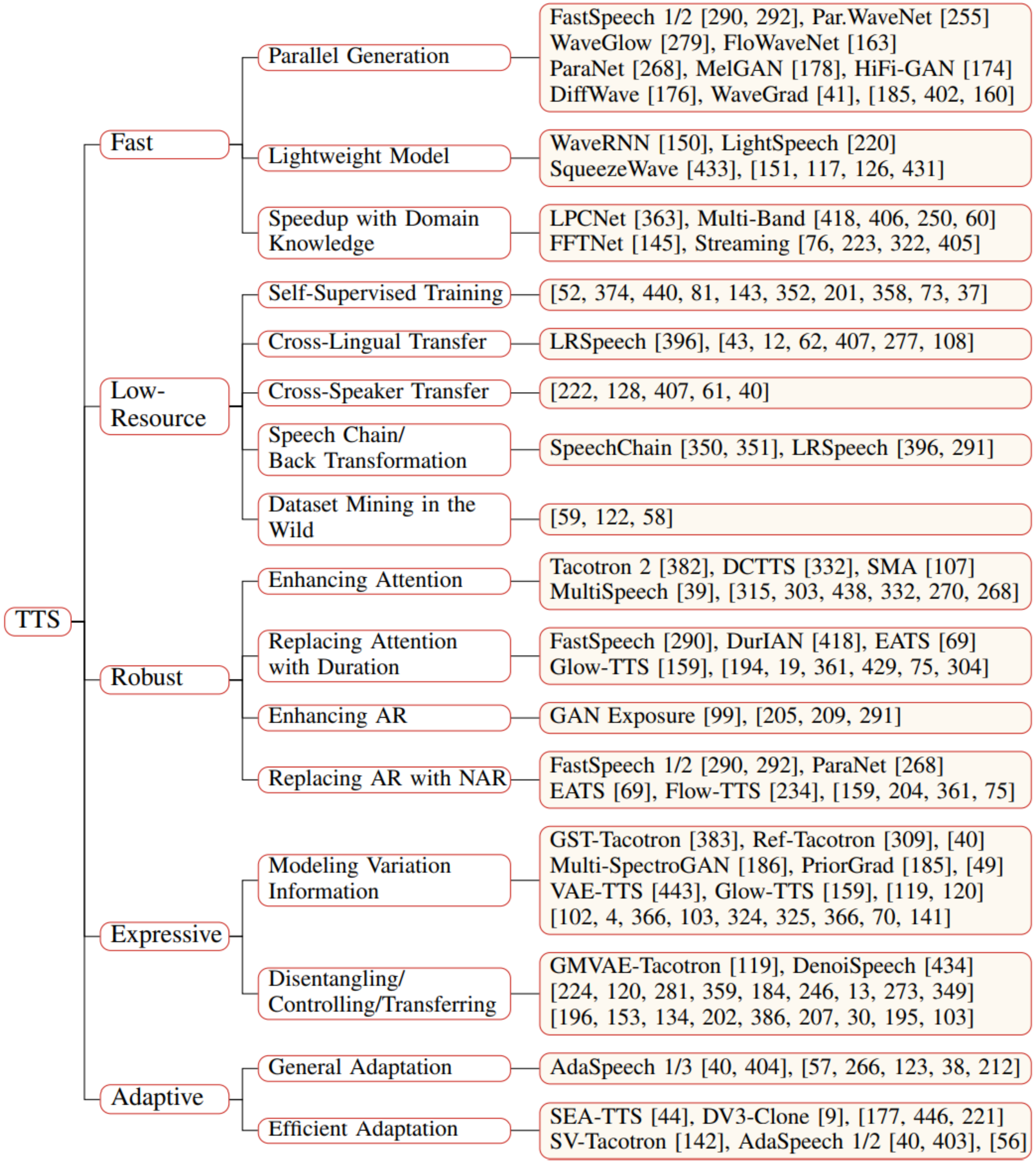
- Spontaneous style
  - Current TTS voices mostly focus on reading style.
  - Spontaneous-style voice is useful for scenarios like podcast, conversation, etc.
- AdaSpeech 3
  - Construct spontaneous dataset
  - Modeling filled pauses (FP, um and uh) and diverse rhythms

Cecily package in all of that um yeah so …



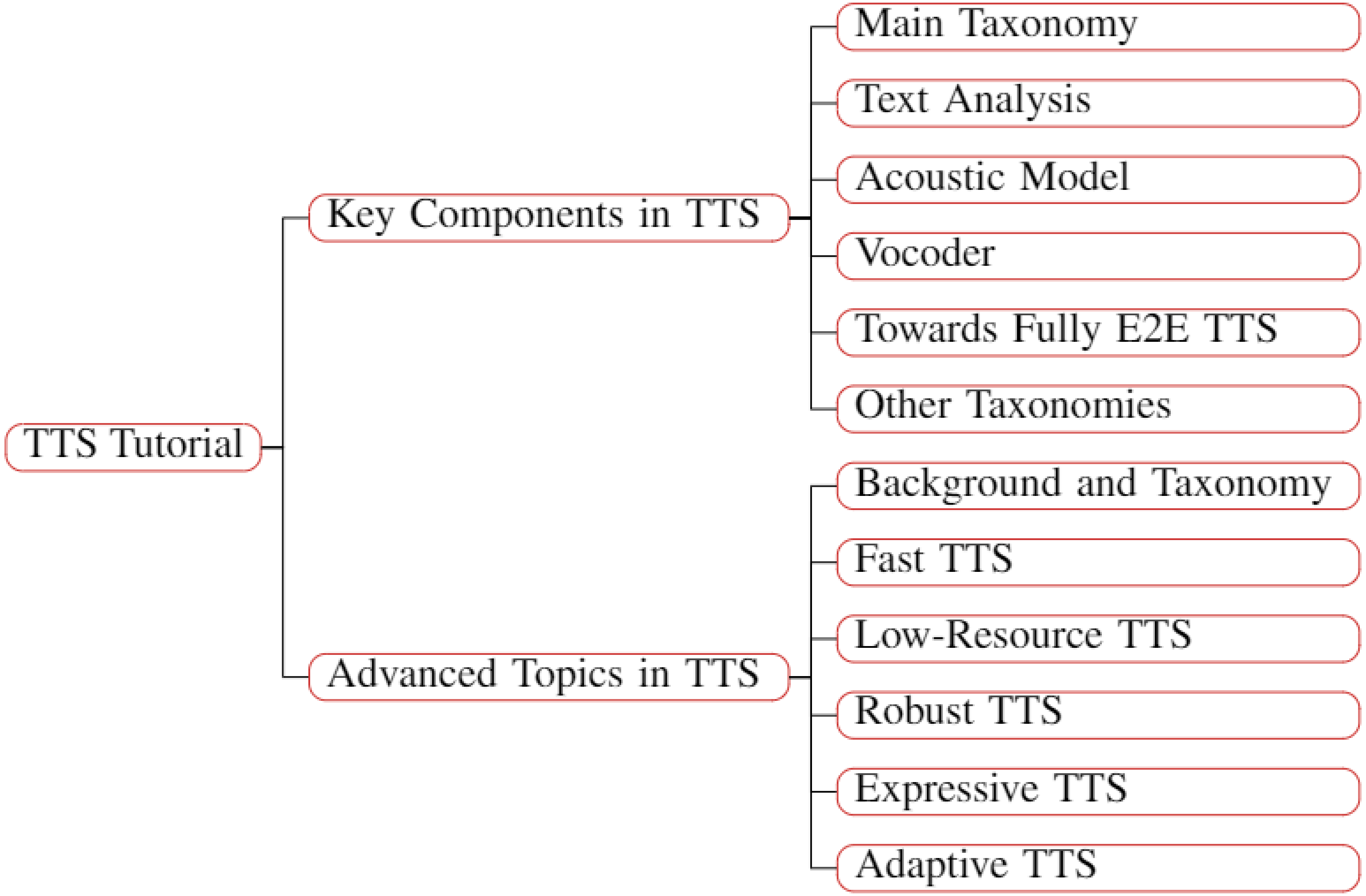
Advanced topics in TTS

- Fast TTS
- Low-resource TTS
- Robust TTS
- Expressive TTS
- Adaptive TTS



Part 4: Summary and Future Directions

Summary


TTS Tutorial
Key Components in TTS
Advanced Topics in TTS
Main Taxonomy
Text Analysis
Acoustic Model
Vocoder
Towards Fully E2E TTS
Other Taxonomies
Background and Taxonomy
Fast TTS
Low-Resource TTS
Robust TTS
Expressive TTS
Adaptive TTS

Outlook: higher-quality synthesis

- Powerful generative models
- Better representation learning
- Robust speech synthesis
- 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!

Outlook: more efficient synthesis

- Data-efficient TTS
- Parameter-efficient TTS
- Energy-efficient TTS

Reference

See the reference in:

A Survey on Neural Speech Synthesis

https://arxiv.org/pdf/2106.15561v3.pdf

https://speechresearch.github.io/

We are hiring

- Research FTE (social/campus hire)
  - Speech (TTS/ASR)
  - NLP (NMT, Summarization, Conversation, Pre-training, etc)
  - Machine Learning, Deep Learning
  - Generative Models
- Research Intern
  - Speech, Music, NLP, ML

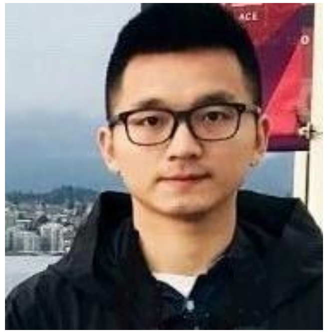
Machine Learning Group, Microsoft Research Asia
Xu Tan xuta@microsoft.com

Thank You!

Xu Tan/谭旭
Senior Researcher @ Microsoft Research Asia
xuta@microsoft.com

https://www.microsoft.com/en-us/research/people/xuta/
https://speechresearch.github.io/

Recent Advances in Neural Speech Synthesis



Xu Tan and Tao Qin
Microsoft Research Asia

Tutorial slides: https://github.com/tts-tutorial/icassp2022
Survey paper: https://arxiv.org/pdf/2106.15561