A Survey on Neural Speech Synthesis

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

Abstract

Text to speech (TTS), or speech synthesis, which aims to synthesize intelligible and natural speech given text, is a hot research topic in speech, language, and machine learning communities and has broad applications in the industry. As the development of deep learning and artificial intelligence, neural network-based TTS has significantly improved the quality of synthesized speech in recent years. In this paper, we conduct a comprehensive survey on neural TTS, aiming to provide a good understanding of current research and future trends. We focus on the key components in neural TTS, including text analysis, acoustic models, and vocoders, and several advanced topics, including fast TTS, low-resource TTS, robust TTS, expressive TTS, and adaptive TTS, etc. We further summarize resources related to TTS (e.g., datasets, opensource implementations) and discuss future research directions. This survey can serve both academic researchers and industry practitioners working on TTS.

1 Introduction

Text to speech (TTS), also known as speech synthesis, which aims to synthesize intelligible and natural speech from text [346], has broad applications in human communication [1] and has long been a research topic in artificial intelligence, natural language and speech processing [296, 228, 147]. Developing a TTS system requires knowledge about languages and human speech production, and involves multiple disciplines including linguistics [63], acoustics [170], digital signal processing [320], and machine learning [25, 146].

As the development of deep learning [183, 89], neural network-based TTS has thrived, and a large amount of research work comes out focusing on different aspects of neural TTS [426, 254, 382, 303, 150, 270, 192, 290]. Consequently, the quality of synthesized speech has been largely improved in recent years. Understanding the current research status and figuring out unsolved research problems are very helpful for people working on TTS. While there are multiple survey papers on statistical parametric speech synthesis [27, 425, 357, 422] and neural TTS [331, 226, 306, 248, 118, 260, 242], a comprehensive survey on the basics and the recent developments of neural TTS is still necessary since the topics in this area are diverse and evolve quickly. In this paper, we conduct a deep and comprehensive survey on neural TTS[2] [3].

In the following subsections, we first briefly review the history of TTS technologies, then introduce some basic knowledge of neural TTS, and finally outline this survey.

[*]Corresponding author: Xu Tan, xuta@microsoft.com

[2]This survey paper is originated from our TTS tutorials, including TTS tutorial at ISCSLP 2021 (https://tts-tutorial.github.io/iscslp2021/) and TTS tutorial at IJCAI 2021 (https://tts-tutorial.github.io/ijcai2021/).

[3]Readers can use this Github page (https://github.com/tts-tutorial/survey) to check updates and initiate discussions on this survey paper.

Preprint. Under review.

1.1 History of TTS Technology

People have tried to build machines to synthesize human speech dating back to the 12th century [388]. In the 2nd half of the 18th century, the Hungarian scientist, Wolfgang von Kempelen, had constructed a speaking machine with a series of bellows, springs, bagpipes and resonance boxes to produce some simple words and short sentences [72]. The first speech synthesis system that built upon computer came out in the latter half of the 20th century [388]. The early computer-based speech synthesis methods include articulatory synthesis [53, 300], formant synthesis [299, 5, 171, 172], and concatenative synthesis [253, 241, 297, 127, 26]. Later, as the development of statistics machine learning, statistical parametric speech synthesis (SPSS) is proposed [416, 356, 425, 357], which predicts parameters such as spectrum, fundamental frequency and duration for speech synthesis. From 2010s, neural network-based speech synthesis [426, 284, 78, 424, 375, 191, 254, 382] has gradually become the dominant methods and achieved much better voice quality.

Articulatory Synthesis Articulatory synthesis [53, 300] produces speech by simulating the behavior of human articulator such as lips, tongue, glottis and moving vocal tract. Ideally, articulatory synthesis can be the most effective method for speech synthesis since it is the way how human generates speech. However, it is very difficult to model these articulator behaviors in practice. For example, it is hard to collect the data for articulator simulation. Therefore, the speech quality by articulatory synthesis is usually worse than that by later formant synthesis and concatenative synthesis.

Formant Synthesis Formant synthesis [299, 5, 171] produces speech based on a set of rules that control a simplified source-filter model. These rules are usually developed by linguists to mimic the formant structure and other spectral properties of speech as closely as possible. The speech is synthesized by an additive synthesis module and an acoustic model with varying parameters like fundamental frequency, voicing, and noise levels. The formant synthesis can produce highly intelligible speech with moderate computation resources that are well-suited for embedded systems, and does not rely on large-scale human speech corpus as in concatenative synthesis. However, the synthesized speech sounds less natural and has artifacts. Moreover, it is difficult to specify rules for synthesis.

Concatenative Synthesis Concatenative synthesis [253, 241, 297, 127, 26] relies on the concatenation of pieces of speech that are stored in a database. Usually, the database consists of speech units ranging from whole sentence to syllables that are recorded by voice actors. In inference, the concatenative TTS system searches speech units to match the given input text, and produces speech waveform by concatenating these units together. Generally speaking, concatenative TTS can generate audio with high intelligibility and authentic timbre close to the original voice actor. However, concatenative TTS requires huge recording database in order to cover all possible combinations of speech units for spoken words. Another drawback is that the generated voice is less natural and emotional, since concatenation can result in less smoothness in stress, emotion, prosody, etc.

Statistical Parametric Synthesis To address the drawbacks of concatenative TTS, statistical parametric speech synthesis (SPSS) is proposed [416, 356, 415, 425, 357]. The basic idea is that instead of direct generating waveform through concatenation, we can first generate the acoustic parameters [82, 355, 156] that are necessary to produce speech and then recover speech from the generated acoustic parameters using some algorithms [132, 131, 155, 238]. SPSS usually consists of three components: a text analysis module, a parameter prediction module (acoustic model), and a vocoder analysis/synthesis module (vocoder). The text analysis module first processes the text, including text normalization [317], grapheme-to-phoneme conversion [24], word segmentation, etc, and then extracts the linguistic features, such as phonemes, duration and POS tags from different granularities. The acoustic models (e.g., hidden Markov model (HMM) based) are trained with the paired linguistic features and parameters (acoustic features), where the acoustic features include fundamental frequency, spectrum or cepstrum [82, 355], etc, and are extracted from the speech through vocoder analysis [132, 155, 238]. The vocoders synthesize speech from the predicted acoustic features. SPSS has several advantages over previous TTS systems: 1) naturalness, the audio is more natural; 2) flexibility, it is convenient to modify the parameters to control the generate speech; 3) low data cost, it requires less recordings than concatenative synthesis. However, SPSS also has its drawbacks: 1) the generated speech has lower intelligibility due to artifacts such as muffled, buzzing or noisy audio; 2) the generated voice is still robotic and can be easily differentiated from human recording speech.

In near 2010s, as neural network and deep learning have achieved rapid progress, some works first introduce deep neural network into SPSS, such as deep neural network (DNN) based [426, 284] and recurrent neural network (RNN) based [78, 422, 424]. However, these models replace HMM with neural networks and still predict the acoustic features from linguistic features, which follow the paradigm of SPSS. Later, Wang et al. [375] propose to directly generate acoustic features from phoneme sequence instead of linguistic features, which can be regarded as the first exploration for end-to-end[4] speech synthesis. In this survey, we focus on neural based speech synthesis, and mostly on end-to-end models. Since later SPSS also uses neural networks as the acoustic models, we briefly describe these models but do not dive deep into the details.

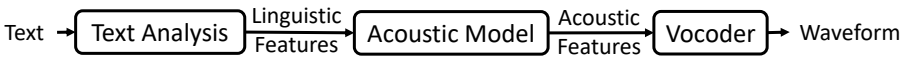


Figure 1: The three key components in neural TTS.

Neural Speech Synthesis As the development of deep learning, neural network-based TTS (neural TTS for short) is proposed, which adopts (deep) neural networks as the model backbone for speech synthesis. Some early neural models are adopted in SPSS to replace HMM for acoustic modeling. Later, WaveNet [254] is proposed to directly generate waveform from linguistic features, which can be regarded as the first modern neural TTS model. Other models like DeepVoice 1/2 [8, 87] still follow the three components in statistical parametric synthesis, but upgrade them with the corresponding neural network based models. Furthermore, some end-to-end models (e.g., Tacotron 1/2 [382, 303], Deep Voice 3 [270], and FastSpeech 1/2 [290, 292]) are proposed to simplify text analysis modules and directly take character/phoneme sequences as input, and simplify acoustic features with mel-spectrograms. Later, fully end-to-end TTS systems are developed to directly generate waveform from text, such as ClariNet [269], FastSpeech 2s [292] and EATS [69]. Compared to previous TTS systems based on concatenative synthesis and statistical parametric synthesis, the advantages of neural network based speech synthesis include high voice quality in terms of both intelligibility and naturalness, and less requirement on human preprocessing and feature development.

1.2 Organization of This Survey

In this paper, we mainly review research works on neural TTS, which consists of two parts, as shown in Figure 2.

Key Components in TTS A modern TTS system consists of three basic components[5]: a text analysis module, an acoustic model, and a vocoder. As shown in Figure 1, the text analysis module converts a text sequence into linguistic features, the acoustic models generate acoustic features from linguistic features, and then the vocoders synthesize waveform from acoustic features. We review the research on the three components of neural TTS in Section 2. Specifically, we first introduce the main taxonomy for the basic components of neural TTS in Section 2.1, and then introduce the works on text analysis, acoustic models, and vocoders in Section 2.2, Section 2.3, and Section 2.4 respectively. We further introduce the research towards fully end-to-end TTS in Section 2.5. Although we mainly review the research works according to the taxonomy of key components in neural TTS, we also describe several other taxonomies, including the way of sequence generation (autoregressive or non-autoregressive), different generative models, and different network structures in Section 2.6. Besides, we also illustrate the time evolution of some representative TTS works in Section 2.6.

[4]The term "end-to-end" in TTS has a vague meaning. In early studies, "end-to-end" refers to that the text-to-spectrogram model is end-to-end, but still uses a separate waveform synthesizer (vocoder). It can also broadly refer to the neural based TTS models which do not use complicated linguistic or acoustic features. For example, WaveNet [254] does not use acoustic features but directly generate waveform from linguistic features, and Tacotron [382] does not use linguistic features but directly generate spectrogram from character or phoneme. However, the strict end-to-end model refers to directly generating waveform from text. Therefore, in this paper we use "end-to-end", "more end-to-end" and "fully end-to-end" to differentiate the degree of end-to-end for TTS models.

[5]Although some end-to-end models do not explicitly use text analysis (e.g., Tacotron 2 [303]), acoustic models (e.g., WaveNet [254]), or vocoders (e.g., Tacotron [382], and some systems only use a single end-to-end model (e.g., FastSpeech 2s [292]), using these components are still popular in current TTS research and product.

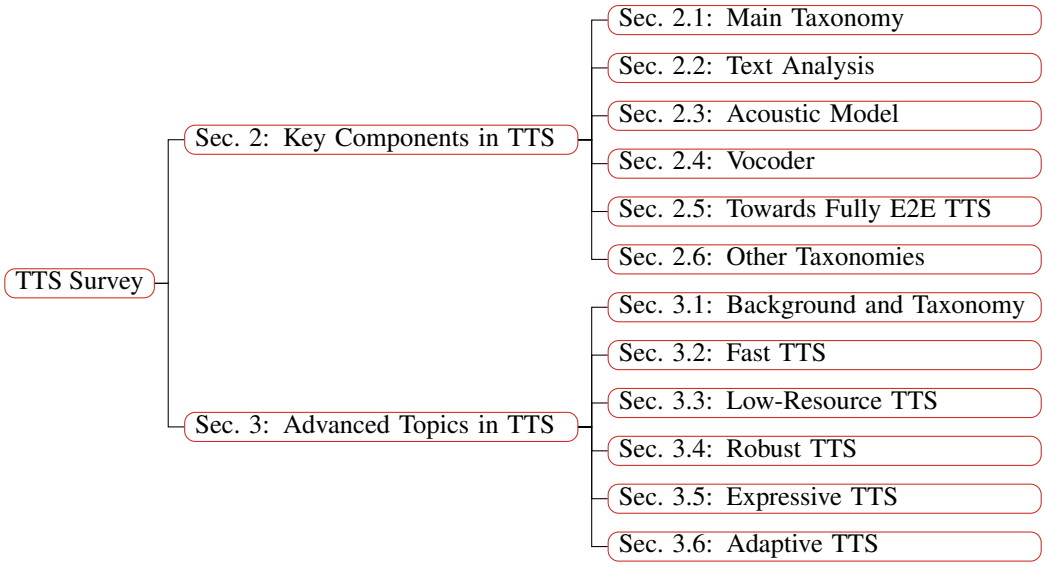


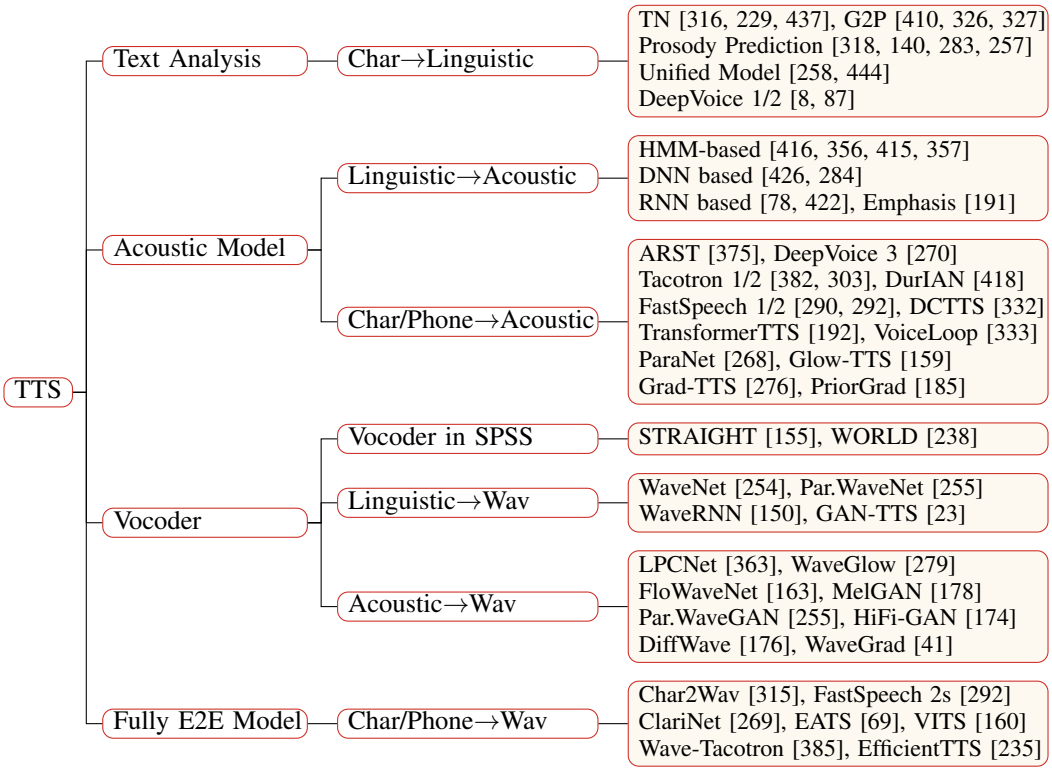
Figure 2: Organization of this survey paper.

Advanced Topics in TTS Besides the key components of neural TTS, we further review several advanced topics in neural TTS, which push the frontier of TTS research and address practical challenges in TTS product. For example, as TTS is a typical sequence to sequence generation task and the output sequence is usually very long, how to speed up the autoregressive generation and reduce the model size for fast speech synthesis are hot research topics (Section 3.2). A good TTS system should generate both natural and intelligible speech and a lot of TTS research works aim to improve the intelligibility and naturalness of speech synthesis. For example, in low-resource scenarios where the data to train a TTS model is insufficient, the synthesized speech may be of both low intelligibility and naturalness. Therefore, a lot of works aim to build data-efficient TTS models under low-resource settings (Section 3.3). Since TTS models are facing robustness issues where word skipping and repeating problems in generated speech affect the speech quality, a lot of works aim to improve the robustness of speech synthesis (Section 3.4). To improve the naturalness and expressiveness, a lot of works model, control, and transfer the style/prosody of speech in order to generate expressive speech (Section 3.5). Adapting a TTS model to support the voice of any target speakers is very helpful for the broad usage of TTS. Therefore, efficient voice adaptation with limited adaptation data and parameters is critical for practical TTS applications (Section 3.6).

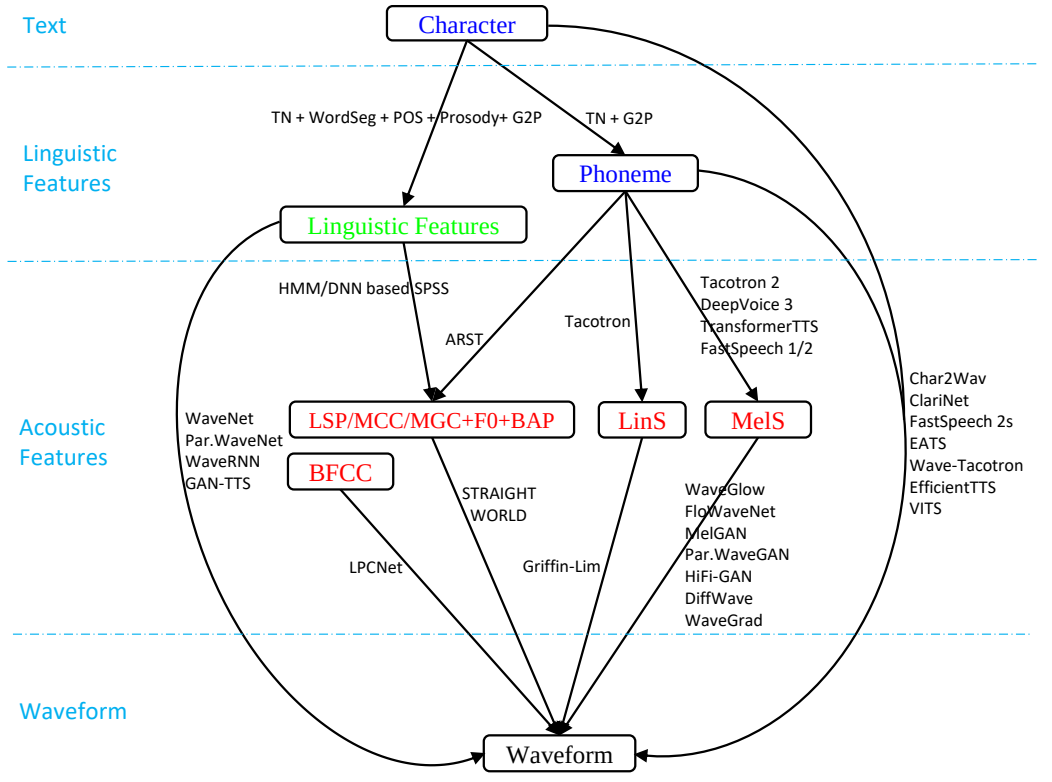
To further enrich this survey, we summarize TTS related resources including open-source implementations, corpora, and other useful resources in Section 4. We summarize this survey and discuss future research directions in Section 5.

2 Key Components in TTS

In this section, we review the research works from the perspective of the key components (text analysis, acoustic models, and vocoders) in neural TTS. We first introduce the main taxonomy under this perspective in Section 2.1, and then introduce the three TTS components in Section 2.2, Section 2.3, and Section 2.4, respectively. Furthermore, we review the works towards fully end-to-end TTS in Section 2.5. Besides the main taxonomy, we also introduce more taxonomies such as autoregressive/non-autoregressive sequence generation, generative model, network structure, as well as the timeline of representative research works on TTS in Section 2.6.



(a) A taxonomy of neural TTS.



(b) The data flows from text to waveform.

Figure 3: A taxonomy of neural TTS from the perspectives of key components and data flows.

2.1 Main Taxonomy

We categorize the works on neural TTS mainly from the perspective of basic TTS components: text analysis, acoustic models, vocoders[6], and fully end-to-end models, as shown in Figure 3a. We find this taxonomy is consistent with the data conversion flow from text to waveform: 1) Text analysis converts character into phoneme or linguistic features; 2) Acoustic models generate acoustic features, from either linguistic features or characters/phonemes; 3) Vocoders generate waveform from either linguistic features or acoustic features; 4) Fully end-to-end models directly convert characters/phonemes into waveform.

We re-organize the TTS works according to the data flow from text to waveform, as shown in Figure 3b. There are several data representations in the process of text to speech conversion: 1) Characters, which are the raw format of text. 2) Linguistic features, which are obtained through text analysis and contain rich context information about pronunciation and prosody. Phonemes are one of the most important elements in linguistic features, and are usually used alone to represent text in neural based TTS models. 3) Acoustic features, which are abstractive representations of speech waveform. In statistical parametric speech synthesis [416, 356, 415, 425, 357], LSP (line spectral pairs) [135], MCC (mel-cepstral coefficients) [82], MGC (mel-generalized coefficients) [355], F0 and BAP (band aperiodicities) [156, 157] are used as acoustic features, which can be easily converted into waveform through vocoders such as STRAIGHT [155] and WORLD [238]. In neural based end-to-end TTS models, mel-spectrograms or linear-spectrograms are usually used as acoustic features, which are converted into waveform using neural based vocoders. 4) Waveform, the final format of speech. As can be seen from Figure 3b, there can be different data flows from text to waveform, including: 1) character $\rightarrow$ linguistic features $\rightarrow$ acoustic features $\rightarrow$ waveform; 2) character $\rightarrow$ phoneme $\rightarrow$ acoustic features $\rightarrow$ waveform; 3) character $\rightarrow$ linguistic features $\rightarrow$ waveform; 4) character $\rightarrow$ phoneme $\rightarrow$ acoustic features $\rightarrow$ waveform; 5) character $\rightarrow$ phoneme $\rightarrow$ waveform, or character $\rightarrow$ waveform.

2.2 Text Analysis

Text analysis, also called frontend in TTS, transforms input text into linguistic features that contain rich information about pronunciation and prosody to ease the speech synthesis. In statistic parametric synthesis, text analysis is used to extract a sequence of linguistic feature vectors [357], and contains several functionalities such as text normalization [316, 439], word segmentation [400], part-of-speech (POS) tagging [298], prosody prediction [51], and grapheme-to-phoneme conversion [410]. In end-to-end neural TTS, due to the large modeling capacity of neural based models, the character or phoneme sequences are directly taken as input for synthesis, and thus the text analysis module is largely simplified. In this scenario, text normalization is still needed to get standard word format from character input, and grapheme-to-phoneme conversion is further needed to get phonemes from standard word format. Although some TTS models claim fully end-to-end synthesis that directly generates waveform from text, text normalization is still needed to handle raw text with any possible non-standard formats for practical usage. Besides, some end-to-end TTS models incorporate conventional text analysis functions. For example, Char2Wav [315] and DeepVoice 1/2 [8, 87] implement the character-to-linguistic feature conversion into its pipeline, purely based on neural networks, and some works[321] explicitly predict prosody features with text encoder. In the remaining of this subsection, we first introduce the typical tasks for text analysis in statistic parametric synthesis, and then discuss the development of text analysis in end-to-end TTS models.

We summarize some typical tasks in text analysis in Table 1, and introduce some representative works for each task as follows.

- Text normalization. The raw written text (non-standard words) should be converted into spoken-form words through text normalization, which can make the words easy to pronounce for TTS models. For example, the year "1989" is normalized into "nineteen eighty nine", "Jan. 24" is normalized into "Janunary twenty-fourth". Early works on text normalization are rule based [317],

[6]Note that some neural TTS models such as WaveNet [254] and WaveRNN [150] are first introduced to directly generate waveform from linguistic features. From this perspective, WaveNet can be regarded as a combination of an acoustic model and a vocoder. Following works usually leverage WaveNet and WaveRNN as a vocoder by taking mel-spectrograms as input to generate waveform. Therefore, we categorize WaveNet/WaveRNN into vocoders and introduce in Section 2.4.

Table 1: Typical tasks in text analysis (i.e., TTS fronend, character→linguistic).

Task	Research Work
Text Normalization	Rule-based [317], Neural-based [316, 229, 413, 437], Hybrid [439]
Word Segmentation	[400, 451, 267]
POS Tagging	[298, 329, 227, 451, 138]
Prosody Prediction	[51, 412, 318, 190, 140, 328, 283, 64, 447, 216, 218, 3]
Grapheme to Phoneme	N-gram [42, 24], Neural-based [410, 289, 33, 326]
- - Polyphone Disambiguation	[448, 398, 230, 301, 327, 29, 263]

and then neural networks are leveraged to model text normalization as a sequence to sequence task where the source and target sequences are non-standard words and spoken-form words respectively [316, 229, 437]. Recently, some works [439] propose to combine the advantages of both rule-based and neural-based models to further improve the performance of text normalization.

- Word segmentation. For character-based languages such as Chinese, word segmentation [400, 451, 267] is necessary to detect the word boundary from raw text, which is important to ensure the accuracy for later POS tagging, prosody prediction, and grapheme-to-phoneme conversion process.
- Part-of-speech tagging. The part-of-speech (POS) of each word, such as noun, verb, preposition, is also important for grapheme-to-phoneme conversion and prosody prediction in TTS. Several works have investigated POS tagging in speech synthesis [298, 329, 227, 451, 138].
- Prosody prediction. The prosody information, such as rhythm, stress, and intonation of speech, corresponds to the variations in syllable duration, loudness and pitch, which plays an important perceptual role in human speech communication. Prosody prediction relies on tagging systems to label each kind of prosody. Different languages have different prosody tagging systems and tools [307, 294, 345, 112, 249]. For English, ToBI (tones and break indices) [307, 294] is a popular tagging system, which describes the tags for tones (e.g., pitch accents, phrase accents, and boundary tones) and break (how strong the break is between words). For example, in this sentence "Mary went to the store ?", "Mary" and "store" can be emphasized, and this sentence is raising tone. A lot of works [318, 190, 140, 283] investigate different models and features to predict the prosody tags based on ToBI. For Chinese speech synthesis, the typical prosody boundary labels consist of prosodic word (PW), prosodic phrase (PPH) and intonational phrase (IPH), which can construct a three-layer hierarchical prosody tree [51, 328, 64]. Some works [51, 3, 64, 328, 216, 218] investigate different model structures such as CRF [180], RNN [114], and self-attention [368] for prosody prediction in Chinese.
- Grapheme-to-phoneme (G2P) conversion. Converting character (grapheme) into pronunciation (phoneme) can greatly ease speech synthesis. For example, the word "speech" is converted into "s p iy ch". A manually collected grapheme-to-phoneme lexicon is usually leveraged for conversion. However, for alphabetic languages like English, lexicon cannot cover the pronunciations of all the words. Thus, the G2P conversion for English is mainly responsible to generate the pronunciations of out-of-vocabulary words [42, 24, 410, 289, 33, 326]. For languages like Chinese, although the lexicon can cover nearly all the characters, there are a lot of polyphones that can be only decided according to the context of a character[7]. Thus, G2P conversion in this kind of languages is mainly responsible for polyphone disambiguation, which decides the appropriate pronunciation based on the current word context [448, 398, 230, 301, 327, 29, 263].

After the above text analyses, we can further construct linguistic features and take them as input to the later part of TTS pipeline, e.g., acoustic models in SPSS or vocoders [254]. Usually, we can construct linguistic features by aggregating the results of text analysis from different levels including phoneme, syllable, word, phrase and sentence levels [357].

Discussions Although text analysis seems to receive less attention in neural TTS compared to SPSS, it has been incorporated into neural TTS in various ways: 1) Multi-task and unified frontend model. Recently, Pan et al. [258], Zhang et al. [444] design unified models to cover all the tasks in text analysis in a multi-task paradigm and achieve good results. 2) Prosody prediction. Prosody

[7] A lot of languages including English have polyphones. For example, "resume" in English can be pronounced as "ri′zju:m′" (means to go on or continue after interruption) or "′rezjumei" (means curriculum vitae).

is critical for the naturalness of speech synthesis. Although neural TTS models simplify the text analysis module, some features for prosody prediction are incorporated into text encoder, such as the prediction of pitch [292], duration [290], phrase break [206], breath or filled pauses [404] are built on top of the text (character or phoneme) encoder in TTS models. Some other ways to incorporate prosody features include 1) reference encoders that learn the prosody representations from reference speech; 2) text pre-training that learns good text representations with implicit prosody information through self-supervised pre-training [104, 98]; and 3) incorporating syntax information through dedicated modeling methods such as graph networks [208].

2.3 Acoustic Models

In this section, we review the works on acoustic models, which generate acoustic features from linguistic features or directly from phonemes or characters. As the development of TTS, different kinds of acoustic models have been adopted, including the early HMM and DNN based models in statistical parametric speech synthesis (SPSS) [416, 356, 426, 284, 78, 424], and then the sequence to sequence models based on encoder-attention-decoder framework (including LSTM, CNN and self-attention) [382, 303, 270, 192], and the latest feed-forward networks (CNN or self-attention) [290, 268] for parallel generation.

Acoustic models aim to generate acoustic features that are further converted into waveform using vocoders. The choice of acoustic features largely determines the types of TTS pipeline. Different kinds of acoustic features have been tried, such as mel-cepstral coefficients (MCC) [82], mel-generalized coefficients (MGC) [355], band aperiodicity (BAP) [156, 157], fundamental frequency (F0), voiced/unvoiced (V/UV), bark-frequency cepstral coefficients (BFCC), and the most widely used mel-spectrograms. Accordingly, we can divide the acoustic models into two periods: 1) acoustic models in SPSS, which typically predict acoustic features such as MGC, BAP and F0 from linguistic features, and 2) acoustic models in neural based end-to-end TTS, which predict acoustic features such as mel-spectrograms from phonemes or characters.

2.3.1 Acoustic Models in SPSS

In SPSS [425, 357], statistical models such as HMM [416, 356], DNN [426, 284] or RNN [78, 424] are leveraged to generate acoustic features (speech parameters) from linguistic features, where the generated speech parameters are converted into speech waveform using a vocoder such as STRAIGHT [155] and WORLD [238]. The developments of these acoustic models are driven by several considerations: 1) taking more context information as input; 2) modeling the correlation between output frames; 3) better combating the over-smoothing prediction problem [425], since the mapping from linguistic features to acoustic features is one-to-many. We briefly review some works as follows.

HMM [286] is leveraged to generate speech parameters in Yoshimura et al. [416], Tokuda et al. [356], where the observation vectors of HMM consist of spectral parameter vectors such as mel-cepstral coefficients (MCC) and F0. Compared to previous concatenative speech synthesis, HMM-based parametric synthesis is more flexible in changing speaker identities, emotions, and speaking styles [356]. Readers can refer to Zen [422], Zen et al. [425], Tokuda et al. [357] for some analyses on the advantages and drawbacks of HMM-based SPSS. One major drawback of HMM-based SPSS is that the quality of the synthesized speech is not good enough [425, 357], mainly due to two reasons: 1) the accuracy of acoustic models is not good, and the predicted acoustic features are over-smoothing and lack of details, and 2) the vocoding techniques are not good enough. The first reason is mainly due to the lack of modeling capacity in HMM. Thus, DNN-based acoustic models [426] are proposed in SPSS, which improve the synthesized quality of HMM-based models. Later, in order to better model the long time span contextual effect in a speech utterance, LSTM-based recurrent neural networks [78] are leveraged to better model the context dependency. As the development of deep learning, some advanced network structure such as CBHG [382] are leveraged to better predict acoustic features [191]. VoiceLoop [333] adopts a working memory called phonological loop to generate acoustic features (e.g., F0, MGC, BAP) from phoneme sequence, and then uses a WORLD [238] vocoder to synthesize waveform from this acoustic features. Yang et al. [409] leverage GAN [90] to improve the generation quality of acoustic features. Wang et al. [375] explore a more end-to-end way that leverages an attention-based recurrent sequence transducer model to directly generate acoustic features from phoneme sequence, which can avoid the frame-by-frame alignments

required in previous neural network-based acoustic models. Wang et al. [379] conduct thorough experimental studies on different acoustic models. Some acoustic models in SPSS are summarized in Table 2.

Table 2: A list of acoustic models and their corresponding characteristics. "Ling" stands for linguistic features, "Ch" stands for character, "Ph" stands for phoneme, "MCC" stands for mel-cepstral coefficients [82], "MGC" stands for mel-generalized coefficients [355], "BAP" stands for band aperiodicities [156, 157], "LSP" stands for line spectral pairs [135], "LinS" stands for linear-spectrograms, and "MelS" stands for mel-spectrograms. "NAR*" means the model uses autoregressive structures upon non-autoregressive structures and is not fully parallel.

Acoustic Model	Input→Output	AR/NAR	Modeling	Structure
HMM-based [416, 356]	Ling→MCC+F0	/	/	HMM
DNN-based [426]	Ling→MCC+BAP+F0	NAR	/	DNN
LSTM-based [78]	Ling→LSP+F0	AR	/	RNN
EMPHASIS [191]	Ling→LinS+CAP+F0	AR	/	Hybrid
ARST [375]	Ph→LSP+BAP+F0	AR	Seq2Seq	RNN
VoiceLoop [333]	Ph→MGC+BAP+F0	AR	/	hybrid
Tacotron [382]	Ch→LinS	AR	Seq2Seq	Hybrid/RNN
Tacotron 2 [303]	Ch→MelS	AR	Seq2Seq	RNN
DurIAN [418]	Ph→MelS	AR	Seq2Seq	RNN
Non-Att Tacotron [304]	Ph→MelS	AR	/	Hybrid/CNN/RNN
Para. Tacotron 1/2 [74, 75]	Ph→MelS	NAR	/	Hybrid/Self-Att/CNN
MelNet [367]	Ch→MelS	AR	/	RNN
DeepVoice [8]	Ch/Ph→MelS	AR	/	CNN
DeepVoice 2 [87]	Ch/Ph→MelS	AR	/	CNN
DeepVoice 3 [270]	Ch/Ph→MelS	AR	Seq2Seq	CNN
ParaNet [268]	Ph→MelS	NAR	Seq2Seq	CNN
DCTTS [332]	Ch→MelS	AR	Seq2Seq	CNN
SpeedySpeech [361]	Ph→MelS	NAR	/	CNN
TalkNet 1/2 [19, 18]	Ch→MelS	NAR	/	CNN
TransformerTTS [192]	Ph→MelS	AR	Seq2Seq	Self-Att
MultiSpeech [39]	Ph→MelS	AR	Seq2Seq	Self-Att
FastSpeech 1/2 [290, 292]	Ph→MelS	NAR	Seq2Seq	Self-Att
AlignTTS [429]	Ch/Ph→MelS	NAR	Seq2Seq	Self-Att
JDI-T [197]	Ph→MelS	NAR	Seq2Seq	Self-Att
FastPitch [181]	Ph→MelS	NAR	Seq2Seq	Self-Att
AdaSpeech 1/2/3 [40, 403, 404]	Ph→MelS	NAR	Seq2Seq	Self-Att
DenoiSpeech [434]	Ph→MelS	NAR	Seq2Seq	Self-Att
DeviceTTS [126]	Ph→MelS	NAR	/	Hybrid/DNN/RNN
LightSpeech [220]	Ph→MelS	NAR	/	Hybrid/Self-Att/CNN
Flow-TTS [234]	Ch/Ph→MelS	NAR*	Flow	Hybrid/CNN/RNN
Glow-TTS [159]	Ph→MelS	NAR	Flow	Hybrid/Self-Att/CNN
Flowtron [366]	Ph→MelS	AR	Flow	Hybrid/RNN
EfficientTTS [235]	Ch→MelS	NAR	Flow	Hybrid/CNN
GMVAE-Tacotron [119]	Ph→MelS	AR	VAE	Hybrid/RNN
VAE-TTS [443]	Ph→MelS	AR	VAE	Hybrid/RNN
BVAE-TTS [187]	Ph→MelS	NAR	VAE	CNN
GAN exposure [99]	Ph→MelS	AR	GAN	Hybrid/RNN
TTS-Stylization [224]	Ch→MelS	AR	GAN	Hybrid/RNN
Multi-SpectroGAN [186]	Ph→MelS	NAR	GAN	Hybrid/Self-Att/CNN
Diff-TTS [141]	Ph→MelS	NAR*	Diffusion	Hybrid/CNN
Grad-TTS [276]	Ph→MelS	NAR	Diffusion	Hybrid/Self-Att/CNN
PriorGrad [185]	Ph→MelS	NAR	Diffusion	Hybrid/Self-Att/CNN

2.3.2 Acoustic Models in End-to-End TTS

Acoustic models in neural-based end-to-end TTS have several advantages compared to those in SPSS: 1) Conventional acoustic models require alignments between linguistic and acoustic features, while sequence to sequence based neural models implicitly learn the alignments through attention or predict the duration jointly, which are more end-to-end and require less preprocessing. 2) As the increasing modeling power of neural networks, the linguistic features are simplified into only character or phoneme sequence, and the acoustic features have changed from low-dimensional and condensed cepstrums (e.g., MGC) to high-dimensional mel-spectrograms or even more high-dimensional linear-spectrograms. In the following paragraphs, we introduce some representative acoustic models in neural TTS[8], and provide a comprehensive list of acoustic models in Table 2.

RNN-based Models (e.g., Tacotron Series) Tacotron [382] leverages an encoder-attention-decoder framework and takes characters as input[9] and outputs linear-spectrograms, and uses Griffin-Lim algorithm [95] to generate waveform. Tacotron 2 [303] is proposed to generate mel-spectrograms and convert mel-spectrograms into waveform using an additional WaveNet [254] model. Tacotron 2 greatly improves the voice quality over previous methods including concatenative TTS, parametric TTS, neural TTS such as Tacotron.

Later, a lot of works improve Tacotron from different aspects: 1) Using a reference encoder and style tokens to enhance the expressiveness of speech synthesis, such as GST-Tacotron [383] and Ref-Tacotron [309]. 2) Removing the attention mechanism in Tacotron, and instead using a duration predictor for autoregressive prediction, such as DurIAN [418] and Non-attentative Tacotron [304]. 3) Changing the autoregressive generation in Tacotron to non-autoregressive generation, such as Parallel Tacotron 1/2 [74, 75]. 4) Building end-to-end text-to-waveform models based on Tacotron, such as Wave-Tacotron [385].

CNN-based Models (e.g., DeepVoice Series) DeepVoice [8] is actually an SPSS system enhanced with convolutional neural networks. After obtaining linguistic features through neural networks, DeepVoice leverages a WaveNet [254] based vocoder to generate waveform. DeepVoice 2 [87] follows the basic data conversion flow of DeepVoice and enhances DeepVoice with improved network structures and multi-speaker modeling. Furthermore, DeepVoice 2 also adopts a Tacotron + WaveNet model pipeline, which first generates linear-spectrograms using Tacotron and then generates waveform using WaveNet. DeepVoice 3 [270] leverage a fully-convolutional network structure for speech synthesis, which generates mel-spectrograms from characters and can scale up to real-word multi-speaker datasets. DeepVoice 3 improves over previous DeepVoice 1/2 systems by using a more compact sequence-to-sequence model and directly predicting mel-spectrograms instead of complex linguistic features.

Later, ClariNet [269] is proposed to generate waveform from text in a fully end-to-end way. ParaNet [268] is a fully convolutional based non-autoregressive model that can speed up the mel-spectrogram generation and obtain reasonably good speech quality. DCTTS [332] shares similar data conversion pipeline with Tacotron, and leverages a fully convolutional based encoder-attention-decoder network to generate mel-spectrograms from character sequence. It then uses a spectrogram super-resolution network to obtain linear-spectrograms, and synthesizes waveform using Griffin-Lim [95].

Transformer-based Models (e.g., FastSpeech Series) TransformerTTS [192] leverages Transformer [368] based encoder-attention-decoder architecture to generate mel-spectrograms from

[8] We mainly review the acoustic models according to different network structures such as RNN, CNN and Transformer (self-attention), while review the vocoders according to different generative models such as autoregressive-based, flow-based, GAN-based, Diffusion-based, as shown in Section 2.4. However, it is not the only perspective, since acoustic models also cover different generative models while vocoders also cover different network structures.

[9] Although either characters or phonemes are taken as input in neural TTS, we do not explicitly differentiate them mainly for two considerations: 1) To ensure high pronunciation accuracy for product usage, phonemes are necessary especially for those languages (e.g., Chinese) where graphemes and phonemes have large difference. 2) For the models directly taking characters as input, there is no specific design for characters input, and thus one can easily change characters into phonemes. It is worth to mention that there are some works [270, 268, 154] using mixed representations of characters and phonemes as input to address the data sparsity problem.

phonemes. They argue that RNN-based encoder-attention-decoder models like Tacotron 2 suffer from the following two issues: 1) Due to the recurrent nature, both the RNN-based encoder and decoder cannot be trained in parallel, and the RNN-based encoder cannot be parallel in inference, which affects the efficiency both in training and inference. 2) Since the text and speech sequences are usually very long, RNN is not good at modeling the long dependency in these sequences. TransformerTTS adopts the basic model structure of Transformer and absorbs some designs from Tacotron 2 such as decoder pre-net/post-net and stop token prediction. It achieves similar voice quality with Tacotron 2 but enjoys faster training time. However, compared with RNN-based models such as Tacotron that leverage stable attention mechanisms such as location-sensitive attention, the encoder-decoder attentions in Transformer are not robust due to parallel computation. Thus, some works propose to enhance the robustness of Transformer-based acoustic models. For example, MultiSpeech [39] improves the robustness of the attention mechanism through encoder normalization, decoder bottleneck, and diagonal attention constraint, and RobuTrans [194] leverages duration prediction to enhance the robustness in autoregressive generation.

Previous neural-based acoustic models such as Tacotron 1/2 [382, 303], DeepVoice 3 [270] and TransformerTTS [192] all adopt autoregressive generation, which suffer from several issues: 1) Slow inference speed. The autoregressive mel-spectrogram generation is slow especially for long speech sequence (e.g., for 1 second speech, there are nearly 500 frames of mel-spectrogram if hop size is 10ms, which is a long sequence). 2) Robust issues. The generated speech usually has a lot of word skipping and repeating and issues, which is mainly caused by the inaccurate attention alignments between text and mel-spectrograms in encoder-attention-decoder based autoregressive generation. Thus, FastSpeech [290] is proposed to solve these issues: 1) It adopts a feed-forward Transformer network to generate mel-spectrograms in parallel, which can greatly speed up inference. 2) It removes the attention mechanism between text and speech to avoid word skipping and repeating issues and improve robustness. Instead, it uses a length regulator to bridge the length mismatch between the phoneme and mel-spectrogram sequences. The length regulator leverages a duration predictor to predict the duration of each phoneme and expands the phoneme hidden sequence according to the phoneme duration, where the expanded phoneme hidden sequence can match the length of mel-spectrogram sequence and facilitate the parallel generation. FastSpeech enjoys several advantages [290]: 1) extremely fast inference speed (e.g., 270x inference speedup on mel-spectrogram generation, 38x speedup on waveform generation); 2) robust speech synthesis without word skipping and repeating issues; and 3) on par voice quality with previous autoregressive models. FastSpeech has been deployed in Microsoft Azure Text to Speech Service[10] to support all the languages and locales in Azure TTS[11].

FastSpeech leverages an explicit duration predictor to expand the phoneme hidden sequence to match to the length of mel-spectrograms. How to get the duration label to train the duration predictor is critical for the prosody and quality of generated voice. We briefly review the TTS models with duration prediction in Section 3.4.2. In the next, we introduce some other improvements based on FastSpeech. FastSpeech 2 [292] is proposed to further enhance FastSpeech, mainly from two aspects: 1) Using ground-truth mel-spectrograms as training targets, instead of distilled mel-spectrograms from an autoregressive teacher model. This simplifies the two-stage teacher-student distillation pipeline in FastSpeech and also avoids the information loss in target mel-spectrograms after distillation. 2) Providing more variance information such as pitch, duration, and energy as decoder input, which eases the one-to-many mapping problem [139, 84, 382, 456] in text to speech[12]. FastSpeech 2 achieves better voice quality than FastSpeech and maintains the advantages of fast, robust, and controllable speech synthesis in FastSpeech[13]. FastPitch [181] improves FastSpeech by using pitch information as decoder input, which shares similar idea of variance predictor in FastSpeech 2.

Other Acoustic Models (e.g., Flow, GAN, VAE, Diffusion) Besides the above acoustic models, there are a lot of other acoustic models [367, 22, 126, 187, 55], as shown in Table 2. Flow-based models have long been used in neural TTS. After the early successful applications on vocoders (e.g.,

[10] https://azure.microsoft.com/en-us/services/cognitive-services/text-to-speech/

[11] https://techcommunity.microsoft.com/t5/azure-ai/neural-text-to-speech-extends-support-to-15-more-languages-with/ba-p/1505911

[12] One-to-many mapping in TTS refers to that there are multiple possible speech sequences corresponding to a text sequence due to variations in speech, such as pitch, duration, sound volume, and prosody, etc.

[13] FastSpeech 2s [292] is proposed together with FastSpeech 2. Since it is a fully end-to-end text-to-waveform model, we introduce it in Section 2.5.

Parallel WaveNet [255], WaveGlow [279], FloWaveNet [163]), flow-based models are also applied in acoustic models, such as Flowtron [366] that is an autoregressive flow-based mel-spectrogram generation model, Flow-TTS [234] and Glow-TTS [159] that leverage generative flow for non-autoregressive mel-spectrogram generation. Besides flow-based models, other generative models have also been leveraged in acoustic models. For example, 1) GMVAE-Tacotron [119], VAE-TTS [443], and BVAE-TTS [187] are based on VAE [168]; 2) GAN exposure [99], TTS-Stylization [224], and Multi-SpectroGAN [186] are based on GAN [90]; 3) Diff-TTS [141], Grad-TTS [276], and PriorGrad [185] are based on diffusion model [310, 113].

Table 3: A list of vocoders and their corresponding characteristics.

Vocoder	Input	AR/NAR	Modeling	Architecture
WaveNet [254]	Linguistic Feature	AR	/	CNN
SampleRNN [233]	/	AR	/	RNN
WaveRNN [150]	Linguistic Feature	AR	/	RNN
LPCNet [363]	BFCC	AR	/	RNN
Univ. WaveRNN [215]	Mel-Spectrogram	AR	/	RNN
SC-WaveRNN [265]	Mel-Spectrogram	AR	/	RNN
MB WaveRNN [418]	Mel-Spectrogram	AR	/	RNN
FFTNet [145]	Cepstrum	AR	/	CNN
Par. WaveNet [255]	Linguistic Feature	NAR	Flow	CNN
WaveGlow [279]	Mel-Spectrogram	NAR	Flow	Hybrid/CNN
FloWaveNet [163]	Mel-Spectrogram	NAR	Flow	Hybrid/CNN
WaveFlow [271]	Mel-Spectrogram	AR	Flow	Hybrid/CNN
SqueezeWave [433]	Mel-Spectrogram	NAR	Flow	CNN
WaveGAN [68]	/	NAR	GAN	CNN
GELP [149]	Mel-Spectrogram	NAR	GAN	CNN
GAN-TTS [23]	Linguistic Feature	NAR	GAN	CNN
MelGAN [178]	Mel-Spectrogram	NAR	GAN	CNN
Par. WaveGAN [402]	Mel-Spectrogram	NAR	GAN	CNN
HiFi-GAN [174]	Mel-Spectrogram	NAR	GAN	Hybrid/CNN
VocGAN [408]	Mel-Spectrogram	NAR	GAN	CNN
GED [96]	Linguistic Feature	NAR	GAN	CNN
Fre-GAN [161]	Mel-Spectrogram	NAR	GAN	CNN
Wave-VAE [268]	Mel-Spectrogram	NAR	VAE	CNN
WaveGrad [41]	Mel-Spectrogram	NAR	Diffusion	Hybrid/CNN
DiffWave [176]	Mel-Spectrogram	NAR	Diffusion	Hybrid/CNN
PriorGrad [185]	Mel-Spectrogram	NAR	Diffusion	Hybrid/CNN

2.4 Vocoders

Roughly speaking, the development of vocoders can be categorized into two stages: the vocoders used in statistical parametric speech synthesis (SPSS) [155, 238, 3], and the neural network-based vocoders [254, 315, 150, 279, 163]. Some popular vocoders in SPSS include STRAIGHT [155] and WORLD [238]. We take the WORLD vocoder as an example, which consists of vocoder analysis and vocoder synthesis steps. In vocoder analysis, it analyzes the speech and gets acoustic features such as mel-cepstral coefficients [82], band aperiodicity [156, 157] and F0. In vocoder synthesis, it generates speech waveform from these acoustic features. In this section, we mainly review the works on neural-based vocoders due to their high voice quality.

Early neural vocoders such as WaveNet [254, 255], Char2Wav [315], WaveRNN [150] directly take linguistic features as input and generate waveform. Later, Prenger et al. [279], Kim et al. [163], Kumar et al. [178], Yamamoto et al. [402] take mel-spectrograms as input and generate waveform. Since speech waveform is very long, autoregressive waveform generation takes much inference time. Thus, generative models such as Flow [65, 169, 167], GAN [90], VAE [168], and DDPM (Denoising Diffusion Probabilistic Model, Diffusion for short) [310, 113] are used in waveform generation. Accordingly, we divide the neural vocoders into different categories: 1) Autoregressive vocoders,

2) Flow-based vocoders, 3) GAN-based vocoders, 4) VAE-based vocoders, and 5) Diffusion-based vocoders. We list some representative vocoders in Table 3 and describe them as follows.

Autoregressive Vocoders WaveNet [254] is the first neural-based vocoder, which leverages dilated convolution to generate waveform points autoregressively. Unlike the vocoder analysis and synthesis in SPSS [82, 355, 156, 135, 155, 238], WaveNet incorporates almost no prior knowledge about audio signals, and purely relies on end-to-end learning. The original WaveNet, as well as some following works that leverage WaveNet as vocoder [8, 87], generate speech waveform conditioned on linguistic features, while WaveNet can be easily adapted to condition on linear-spectrograms [87] and mel-spectrograms [336, 270, 303]. Although WaveNet achieves good voice quality, it suffers from slow inference speed. Therefore, a lot of works [256, 117, 233] investigate lightweight and fast vocoders. SampleRNN [233] leverages a hierarchical recurrent neural network for unconditional waveform generation, and it is further integrated into Char2Wav [315] to generate waveform conditioned on acoustic features. Further, WaveRNN [448] is developed for efficient audio synthesis, using a recurrent neural network and leveraging several designs including dual softmax layer, weight pruning, and subscaling techniques to reduce the computation. Lorenzo-Trueba et al. [215], Paul et al. [265], Jiao et al. [144] further improve the robustness and universality of the vocoders. LPCNet [363, 364] introduces conventional digital signal processing into neural networks, and uses linear prediction coefficients to calculate the next waveform point while leveraging a lightweight RNN to compute the residual. LPCNet generates speech waveform conditioned on BFCC (bark-frequency cepstral coefficients) features, and can be easily adapted to condition on mel-spectrograms. Some following works further improve LPCNet from different perspectives, such as reducing complexity for speedup [370, 275, 151], and improving stability for better quality [129].

Flow-based Vocoders Normalizing flow [65, 66, 293, 169, 167] is a kind of generative model. It transforms a probability density with a sequence of invertible mappings [293]. Since we can get a standard/normalized probability distribution (e.g., Gaussion) through the sequence of invertible mappings based on the change-of-variables rules, this kind of flow-based generative model is called as a normalizing flow. During sampling, it generates data from a standard probability distribution through the inverse of these transforms. The flow-based models used in neural TTS can be divided into two categories according to the two different techniques [262]: 1) autoregressive transforms [169] (e.g., inverse autoregressive flow used in Parallel WaveNet [255]), and 2) bipartite transforms (e.g., Glow [167] used in WaveGlow [279], and RealNVP [66] used in FloWaveNet [163]), as shown in Table 4.

Table 4: Several representative flow-based models and their formulations [271].

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]	$z_b = x_b \cdot \sigma_b(x_a;\theta) + \mu_b(x_a;\theta)$	$x_b = \frac{z_b - \mu_b(x_a;\theta)}{\sigma_b(x_a;\theta)}$

- Autoregressive transforms, e.g., inverse autoregressive flow (IAF) [169]. IAF can be regarded as a dual formulation of autoregressive flow (AF) [261, 124]. The training of AF is parallel while the sampling is sequential. In contrast, the sampling in IAF is parallel while the inference for likelihood estimation is sequential. Parallel WaveNet [255] leverages probability density distillation to marry the efficient sampling of IAF with the efficient training of AR modeling. It uses an autoregressive WaveNet as the teacher network to guide the training of the student network (Parallel WaveNet) to approximate the data likelihood. Similarly, ClariNet [269] uses IAF and teacher distillation, and leverages a closed-form KL divergence to simplify and stabilize the distillation process. Although Parallel Wavenet and ClariNet can generate speech in parallel, it relies on sophisticated teacher-student training and still requires large computation.
- Bipartite transforms, e.g., Glow [167] or RealNVP [66]. To ensure the transforms to be invertible, bipartite transforms leverage the affine coupling layers that ensure the output can be computed from

the input and vice versa. Some vocoders based on bipartite transforms include WaveGlow [279] and FloWaveNet [163], which achieve high voice quality and fast inference speed.

Both autoregressive and bipartite transforms have their advantages and disadvantages [271]: 1) Autoregressive transforms are more expressive than bipartite transforms by modeling dependency between data distribution x and standard probability distribution z, but require teacher distillation that is complicated in training. 2) Bipartite transforms enjoy much simpler training pipeline, but usually require larger number of parameters (e.g., deeper layers, larger hidden size) to reach comparable capacities with autoregressive models. To combine the advantages of both autoregressive and bipartite transforms, WaveFlow [271] provides a unified view of likelihood-based models for audio data to explicitly trade inference parallelism for model capacity. In this way, WaveNet, WaveGlow, and FloWaveNet can be regarded as special cases of WaveFlow.

GAN-based Vocoders Generative adversarial networks (GANs) [90] have been widely used in data generation tasks, such as image generation [90, 455], text generation [419], and audio generation [68]. GAN consists a generator for data generation, and a discriminator to judge the authenticity of the generated data. A lot of vocoders leverage GAN to ensure the audio generation quality, including WaveGAN [68], GAN-TTS [23], MelGAN [178], Parallel WaveGAN [402], HiFi-GAN [174], and other GAN-based vocoders [401, 391, 312, 417, 372, 137].

Table 5: Several representative GAN based vocoders and their characteristics.

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

We summarize the characteristics according to the generators, discriminators, and losses used in each vocoder in Table 5.

- Generator. Most GAN-based vocoders use dilated convolution to increase the receptive field to model the long-dependency in waveform sequence, and transposed convolution to upsample the condition information (e.g., linguistic features or mel-spectrograms) to match the length of waveform sequence. Yamamoto et al. [402] choose to upsample the conditional information one time, and then perform dilated convolution to ensure model capacity. However, this kind of upsampling increases the sequence length too early, resulting larger computation cost. Therefore, some vocoders [178, 174] choose to iteratively upsample the condition information and perform dilated convolution, which can avoid too long sequence in the lower layers. Specifically, VocGAN [408] proposes a multi-scale generator that can gradually output waveform sequence at different scales, from coarse-grained to fine-grained. HiFi-GAN [174] processes different patterns of various lengths in parallel through a multi-receptive field fusion module, and also has the flexibility to trade off between synthesis efficiency and sample quality.
- Discriminator. The research efforts [23, 178, 174, 408] on discriminators focus on how to design models to capture the characteristics of waveform, in order to provide better guiding signal for the generators. We review these efforts as follows: 1) Random window discriminators, proposed in GAN-TTS [23], which use multiple discriminators, where each is feeding with different random windows of waveform with and without conditional information. Random window discriminators

have several benefits, such as evaluating audios in different complementary way, simplifying the true/false judgements compared with full audio, and acting as a data augmentation effect, etc. 2) Multi-scale discriminators, proposed in MelGAN [178], which use multiple discriminators to judge audios in different scales (different downsampling ratios compared with original audio). The advantage of multi-scale discriminators is that the discriminator in each scale can focus on the characteristics in different frequency ranges. 3) Multi-period discriminators, proposed in HiFi-GAN [174], which leverage multiple discriminators, where each accepts equally spaced samples of an input audio with a period. Specifically, the 1D waveform sequence with a length of T is reshaped into a 2D data $[p, T/p]$ where p is the period, and processed by a 2D convolution. Multi-period discriminators can capture different implicit structures by looking at different parts of an input audio in different periods. 4) Hierarchical discriminators, leveraged in VocGAN [408] to judge the generated waveform in different resolutions from coarse-grained to fine-grained, which can guide the generator to learn the mapping between the acoustic features and waveform in both low and high frequencies.

- Loss. Except for the regular GAN losses such as WGAN-GP [97], hinge-loss GAN [198], and LS-GAN [231], other specific losses such as STFT loss [10, 401] and feature matching loss [182] are also leveraged. These additional losses can improve the stability and efficiency of adversarial training [402], and improve the perceptual audio quality. Gritsenko et al. [96] propose a generalized energy distance with a repulsive term to better capture the multi-modal waveform distribution.

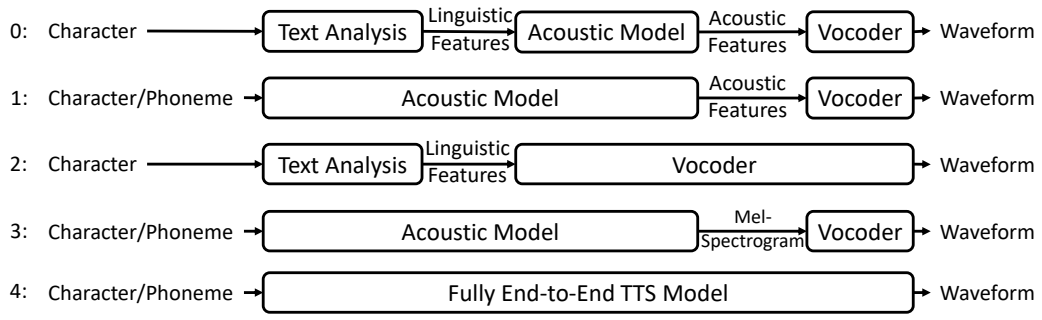
Diffusion-based Vocoders Recently, there are some works leveraging denoising diffusion probabilistic models (DDPM or Diffusion) [113] for vocoders, such as DiffWave [176], WaveGrad [41], and PriorGrad [185]. The basic idea is to formulate the mapping between data and latent distributions with diffusion process and reverse process: in the diffusion process, the waveform data sample is gradually added with some random noises and finally becomes Gaussion noise; in the reverse process, the random Gaussion noise is gradually denoised into waveform data sample step by step. Diffusion-based vocoders can generate speech with very high voice quality, but suffer from slow inference speed due to the long iterative process. Thus, a lot of works on diffusion models [313, 185, 384, 175] are investigating how to reduce inference time while maintaining generation quality.

Other Vocoders Some works leverage neural-based source-filter model for waveform generation [381, 380, 377, 213, 149, 148, 77, 311, 414], aiming to achieve high voice quality while maintaining controllable speech generation. Govalkar et al. [91] conduct a comprehensive study on different kinds of vocoders. Hsu et al. [118] study the robustness of vocoders by evaluating several common vocoders with comprehensive experiments.

Discussions We summarize the characteristics of different kinds of generative models used in vocoders, as shown in Table 6: 1) In terms of mathematical simplicity, autoregressive (AR) based models are easier than other generative models such as VAE, Flow, Diffusion, and GAN. 2) All the generative models except AR can support parallel speech generation. 3) Except for AR models, all generative models can support latent manipulations to some extent (some GAN-based vocoders do not take random Gaussian noise as model input, and thus cannot support latent manipulation). 4) GAN-based models cannot estimate the likelihood of data samples, while other models enjoy this benefit.

Table 6: Some characteristics of several representative generative models used in vocoders.

Generative Model	AR	VAE	Flow/AR	Flow/Bipartite	Diffusion	GAN
Vocoder (e.g.)	WaveNet	WaveVAE	Par.WaveNet	WaveGlow	DiffWave	MelGAN
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



Stage	Models
0	SPSS [416, 356, 415, 425, 357]
1	ARST [375]
2	WaveNet [254], DeepVoice 1/2 [8, 87], Par. WaveNet [255], WaveRNN [150], HiFi-GAN [23]
3	DeepVoice 3 [270], Tacotron 2 [303], FastSpeech 1/2 [290, 292], WaveGlow [279], FloWaveNet [163]
4	Char2Wav [315], ClariNet [269], FastSpeech 2s [292], EATS [69], Wave-Tacotron [385], VITS [160]

Figure 4: The progressively end-to-end process for TTS models.

2.5 Towards Fully End-to-End TTS

Fully end-to-end TTS models can generate speech waveform from character or phoneme sequence directly, which have the following advantages: 1) It requires less human annotation and feature development (e.g., alignment information between text and speech); 2) The joint and end-to-end optimization can avoid error propagation in cascaded models (e.g., Text Analysis + Acoustic Model + Vocoder); 3) It can also reduce the training, development and deployment cost.

However, there are big challenges to train TTS models in an end-to-end way, mainly due to the different modalities between text and speech waveform, as well as the huge length mismatch between character/phoneme sequence and waveform sequence. For example, for a speech with a length of 5 seconds and about 20 words, the length of the phoneme sequence is just about 100, while the length of the waveform sequence is 80k (if the sample rate is 16kHz). It is hard to put the waveform points of the whole utterance into model training, due to the limit of memory. It is hard to capture the context representations if only using a short audio clip for the end-to-end training.

Due to the difficulty of fully end-to-end training, the development of neural TTS follows a progressive process towards fully end-to-end models. Figure 4 illustrates this progressive process starting from early statistic parametric synthesis [416, 356, 415, 425, 357]. The process towards fully end-to-end models typically contains these upgrades: 1) Simplifying text analysis module and linguistic features. In SPSS, text analysis module contains different functionalities such as text normalization, phrase/word/syllable segmentation, POS tagging, prosody prediction, grapheme-to-phoneme conversion (including polyphone disambiguation). In end-to-end models, only the text normalization and grapheme-to-phoneme conversion are retained to convert characters into phonemes, or the whole text analysis module is removed by directly taking characters as input. 2) Simplifying acoustic features, where the complicated acoustic features such as MGC, BAP and F0 used in SPSS are simplified into mel-spectrograms. 3) Replacing two or three modules with a single end-to-end model. For example, the acoustic models and vocoders can be replaced with a single vocoder model such as WaveNet. Accordingly, we illustrate the progressive process in Figure 4 and describe it as follows.

- Stage 0. Statistic parametric synthesis [416, 356, 415, 425, 357] uses three basic modules, where text analysis module converts characters into linguistic features, and acoustic models generate acoustic features from linguistic features (where the target acoustic features are obtained through vocoder analysis), and then vocoders synthesize speech waveform from acoustic features through parametric calculation.

Table 7: A list of fully end-to-end TTS models.

Model	One-Stage Training	AR/NAR	Modeling	Architecture
Char2Wav [315]	N	AR	Seq2Seq	RNN
ClariNet [269]	N	AR	Flow	CNN
FastSpeech 2s [292]	Y	NAR	GAN	Self-Att/CNN
EATS [69]	Y	NAR	GAN	CNN
Wave-Tacotron [385]	Y	AR	Flow	CNN/RNN/Hybrid
EfficientTTS-Wav [235]	Y	NAR	GAN	CNN
VITS [160]	Y	NAR	VAE+Flow	CNN/Self-Att/Hybrid

- Stage 1. Wang et al. [375] in statistic parametric synthesis explore to combine the text analysis and acoustic model into an end-to-end acoustic model that directly generates acoustic features from phoneme sequence, and then uses a vocoder in SPSS to generate waveform.
- Stage 2. WaveNet [254] is first proposed to directly generate speech waveform from linguistic features, which can be regarded as a combination of an acoustic model and a vocoder. This kind of models [254, 255, 150, 23] still require a text analysis module to generate linguistic features.
- Stage 3. Tacotron [382] is further proposed to simplify linguistic and acoustic features, which directly predicts linear-spectrograms from characters/phonemes with an encoder-attention-decoder model, and converts linear-spectrograms into waveform with Griffin-Lim [95]. The following works such as DeepVoice 3 [270], Tacotron 2 [303], TransformerTTS [192], and FastSpeech 1/2 [290, 292] predict mel-spectrograms from characters/phonemes and further use a neural vocoder such as WaveNet [254], WaveRNN [150], WaveGlow [279], FloWaveNet [163], and Parallel WaveGAN [402] to generate waveform.
- Stage 4. Some fully end-to-end TTS models are developed for direct text to waveform synthesis, as listed in Table 7. Char2Wav [315] leverages an RNN-based encoder-attention-decoder model to generate acoustic features from characters, and then uses SampleRNN [233] to generate waveform. The two models are jointly tuned for direct speech synthesis. Similarly, ClariNet [269] jointly tunes an autoregressive acoustic model and a non-autoregressive vocoder for direct waveform generation. FastSpeech 2s [292] directly generate speech from text with a fully parallel structure, which can greatly speed up inference. To alleviate the difficulty of joint text-to-waveform training, it leverages an auxiliary mel-spectrogram decoder to help learn the contextual representations of phoneme sequence. A concurrent work called EATS [69] also directly generates waveform from characters/phonemes, which leverages duration interpolation and soft dynamic time wrapping loss for end-to-end alignment learning. Wave-Tacotron [385] builds a flow-based decoder on Tacotron to directly generate waveform, which uses parallel waveform generation in the flow part but still autoregressive generation in the Tacotron part.

2.6 Other Taxonomies

Besides the main taxonomy from the perspective of key components and data flow as shown in Figure 3, we can also categorize TTS works from several different taxonomies, as shown in Figure 5: 1) **Autoregressive or non-autoregressive**. We can divide these works into autoregressive and non-autoregressive generation models. 2) **Generative model**. Since TTS is a typical sequence generation task and can be modeled through typical generative models, we can categorize in terms of different generative models: normal sequence generation model, flow, GAN, VAE, and diffusion model. 3) **Network structure**. We can divide the works according to their network structures, such as CNN, RNN, self-attention, and hybrid structures (which contains more than one type of structures, such as CNN+RNN, CNN+self-attention).

Evolution of Neural TTS Models In order to better understand the development of various research works on neural TTS and their relationships, we illustrate the evolution of neural TTS models, as shown in Figure 6. Note that we organize the research works according to the time that the paper is open to the public (e.g., put on arXiv), but not formally published later. We choose the early time since we appreciate researchers making their paper public early to encourage knowledge sharing.

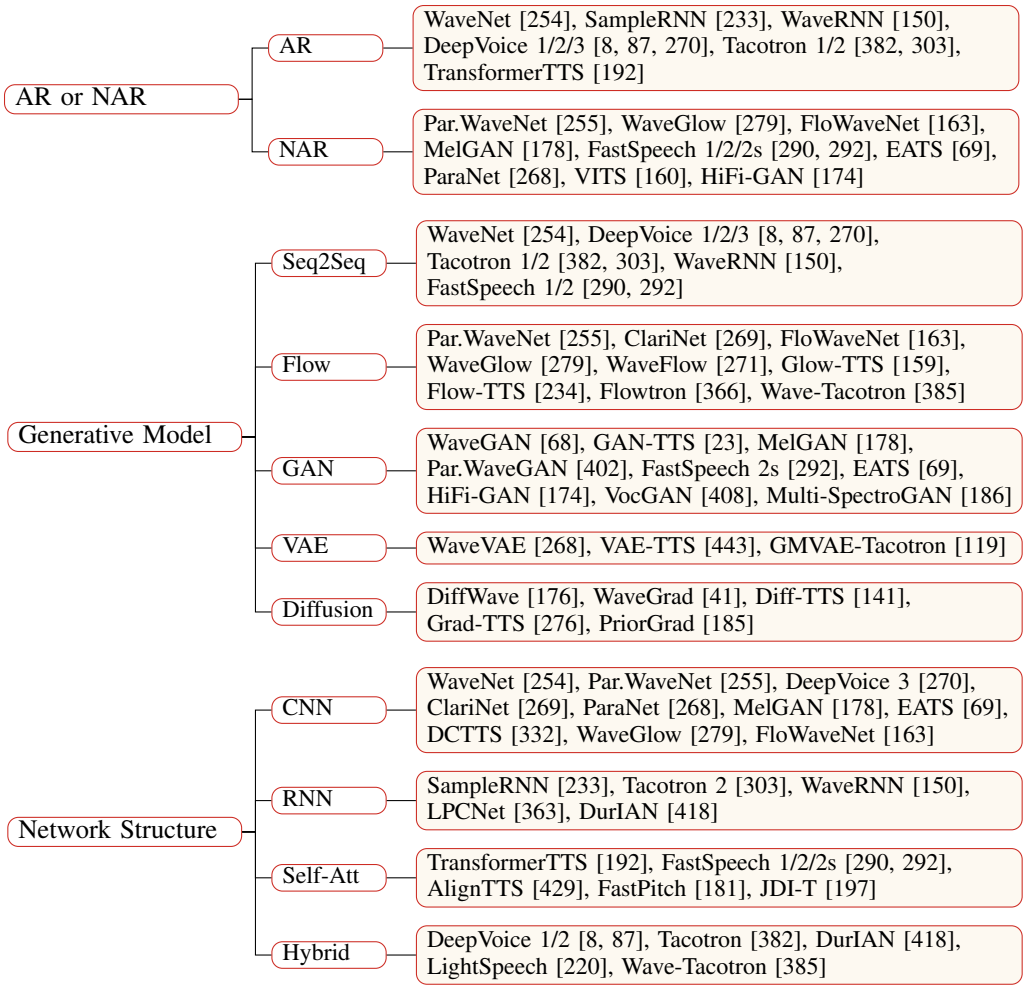


Figure 5: Some other taxonomies of neural TTS from the perspectives of AR/NAR, generative model, and network structure.

Since the research works on neural TTS are so abundant, we only choose some representative works in Figure 6, and list more works in Table 18.

3 Advanced Topics in TTS

3.1 Background and Taxonomy

In previous section, we have introduced neural TTS in terms of basic model components. In this section, we review some advanced topics in neural TTS that aim to push the frontier and cover more practical product usage. Specifically, as TTS is a typical sequence to sequence generation task with slow autoregressive generation, how to speed up the autoregressive generation or reduce the model size for fast speech synthesis is a hot research topic (Section 3.2). A good TTS system should generate both natural and intelligible speech and a lot of TTS research works aim to improve the intelligibility and naturalness of speech synthesis. For example, in low-resource scenarios where the data to train a TTS model is not enough, the synthesized speech may have both low intelligibility and naturalness. Therefore, a lot of works aim to build data efficient TTS models under low-resource settings (Section 3.3). Since TTS models are prone to suffer from robust issues where the generated speech usually has word skipping and repeating problems that affect the intelligibility, a lot of works aim to improve the robustness of speech synthesis (Section 3.4). To improve the naturalness, a lot of works aim to model, control, and transfer the style/prosody of speech in order to generate expressive

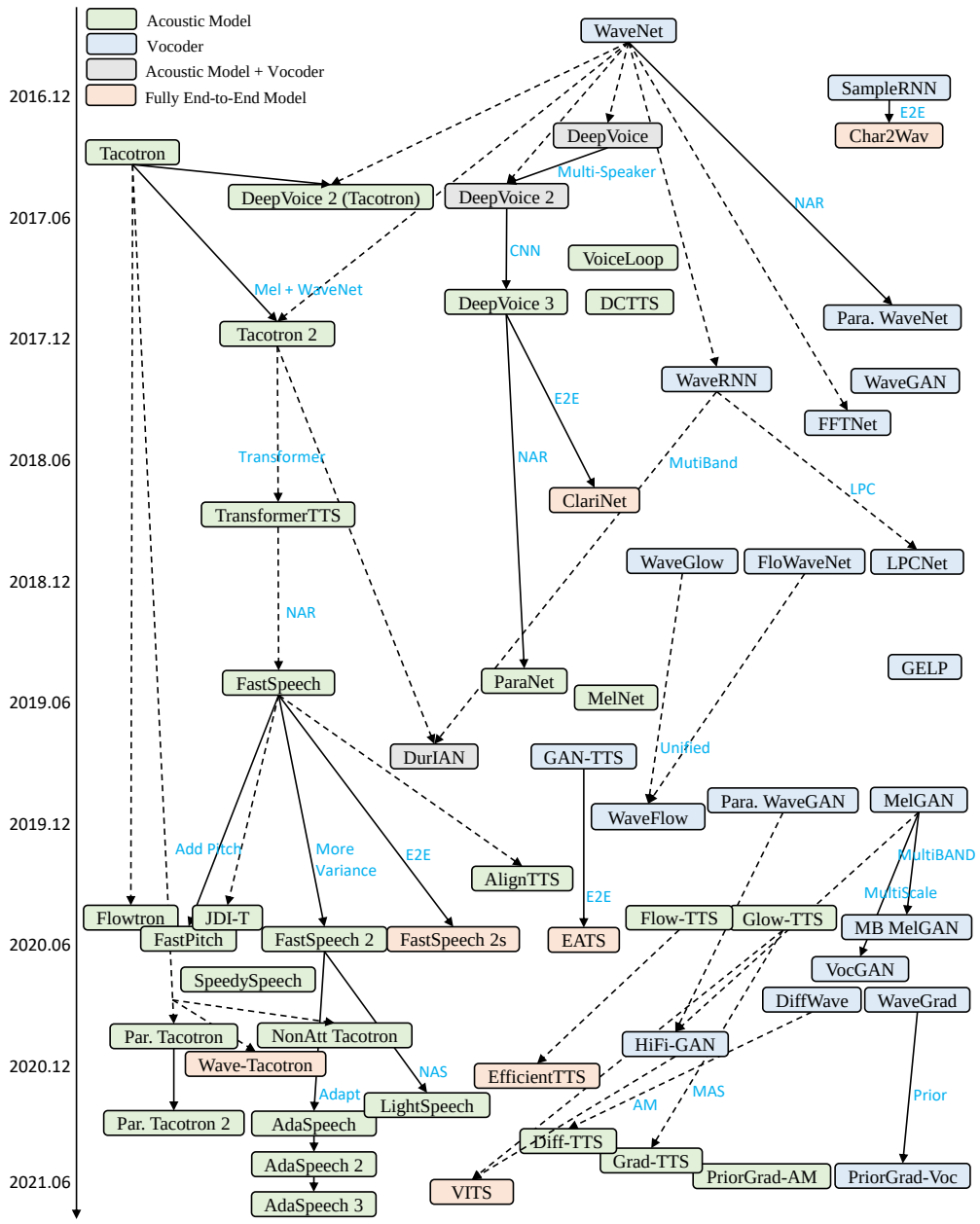


Figure 6: The evolution of neural TTS models.

speech (Section 3.5). Adapting TTS models to support the voice of any target speakers is very helpful for broad usage of TTS. Therefore, efficient voice adaptation with limited adaptation data and parameters and with high-quality voice is critical for practical usage (Section 3.6). A taxonomy of these advanced topics are shown in Figure 7.

3.2 Fast TTS

Text to speech synthesis systems are usually deployed in cloud server or embedded devices, which require fast synthesis speed. However, early neural TTS models usually adopt autoregressive mel-spectrogram and waveform generation, which are very slow considering the long speech sequence (e.g., a 1 second speech usually has 500 mel-spetrograms if hop size is 10ms, and 24k waveform points if sampling rate is 24kHz). To solve this problem, different techniques have been leveraged to speed up the inference of TTS models, including 1) non-autoregressive generation that generates mel-

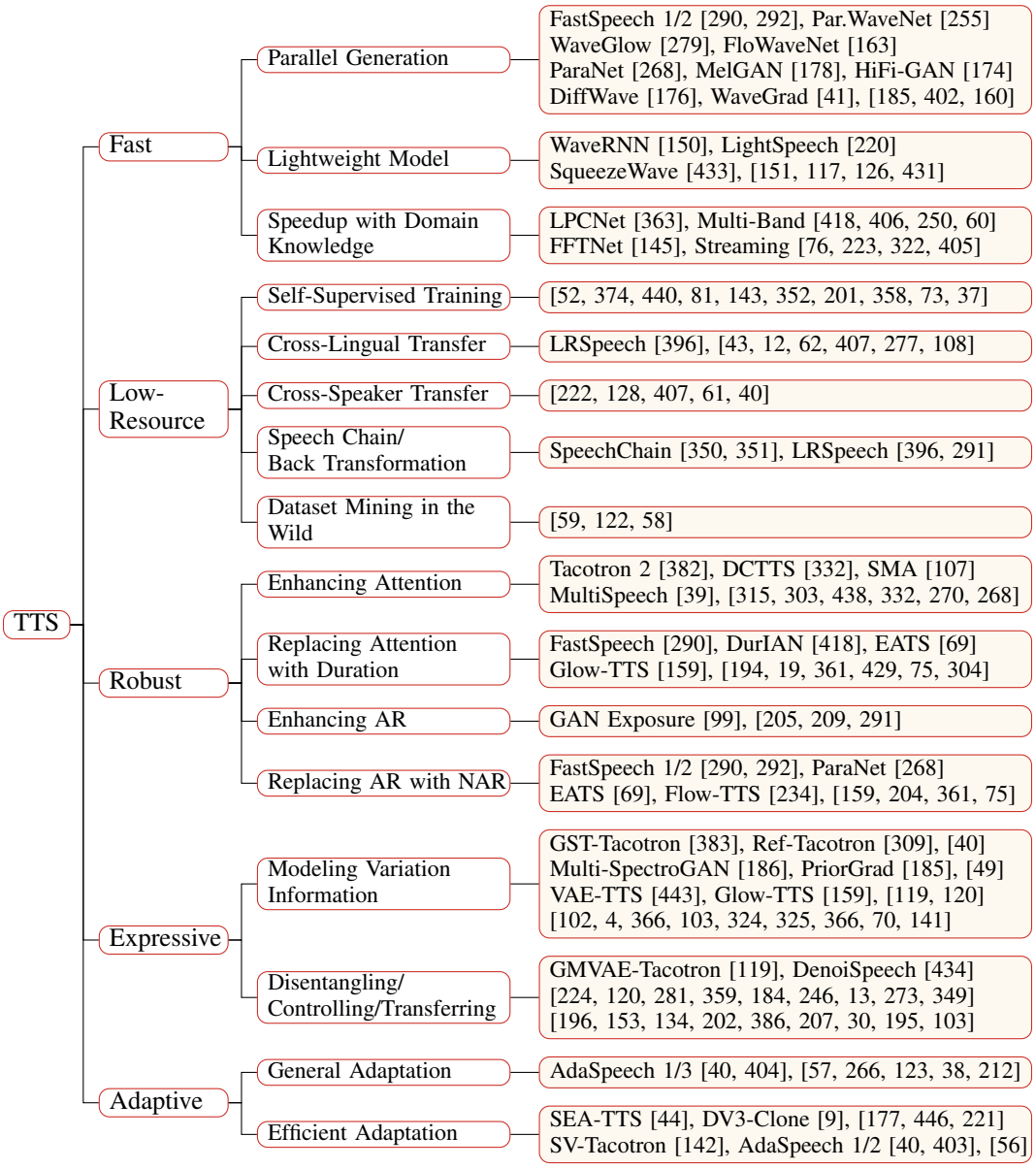


Figure 7: Overview of the advanced topics in neural TTS as described in Section 3.

spectrograms and waveform in parallel; 2) lightweight and efficient model structure; 3) techniques leveraging the domain knowledge of speech for fast speech synthesis. We introduce these techniques as follows.

Table 8: The time complexity of different TTS models in training and inference with regard to sequence length N. T is the number of steps/iterations in flow/diffusion based models.

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

Parallel Generation Table 8 summarizes typical modeling paradigms, the corresponding TTS models, and time complexity in training and inference. As can be seen, TTS models that use RNN-based autoregressive models [382, 303, 233, 363] are slow in both training and inference, with $\mathcal{O}(N)$ computation, where N is the sequence length. To avoid the slow training time caused by RNN structure, DeepVoice 3 [270] and TransformerTTS [192] leverage CNN or self-attention based structure that can support parallel training but still require autoregressive inference. To speed up inference, FastSpeech 1/2 [290, 292] design a feed-forward Transformer that leverages self-attention structure for both parallel training and inference, where the computation is reduced to $\mathcal{O}(1)$. Most GAN-based models for mel-spectrogram and waveform generation [178, 174, 292, 69] are non-autoregressive, with $\mathcal{O}(1)$ computation in both training and inference. Parallel WaveNet [255] and ClariNet [269] leverage inverse autoregressive flow [169], which enable parallel inference but require teacher distillation for parallel training. WaveGlow [279] and FloWaveNet [163] leverage generative flow for parallel training and inference. However, they usually need to stack multiple flow iterations T to ensure the quality of the mapping between data and prior distributions. Similar to flow-based models, diffusion-based models [41, 176, 185, 141, 276] also require multiple diffusion steps T in the forward and reverse process, which increase the computation.

Lightweight Model While non-autoregressive generation can fully leverage the parallel computation for inference speedup, the number of model parameters and total computation cost are not reduced, which make it slow when deploying on the mobile phones or embedded devices since the parallel computation capabilities in these devices are not powerful enough. Therefore, we need to design lightweight and efficient models with less computation cost for inference speedup, even using autoregressive generation. Some widely used techniques for designing lightweight models include pruning, quantization, knowledge distillation [111], and neural architecture search [220, 397], etc. WaveRNN [150] uses techniques like dual softmax, weight pruning, subscale prediction to speed up inference. LightSpeech [220] leverages neural architecture search [457, 219] to find lightweight architectures to further speed up the inference of FastSpeech 2 [292] by 6.5x, while maintaining voice quality. SqueezeWave [433] leverages waveform reshaping to reduce the temporal length and replaces the 1D convolution with depthwise separable convolution to reduce computation cost while achieving similar audio quality. Kanagawa and Ijima [151] compress the model parameters of LPCNet with tensor decomposition. Hsu and Lee [117] propose a heavily compressed flow-based model to reduce computational resources, and a WaveNet-based post-filter to maintain audio quality. DeviceTTS [126] leverages the model structure of DFSMN [441] and mix-resolution decoder to predict multiple frames in one decoding step to speed up inference. LVCNet [431] adopts a location-variable convolution for different waveform intervals, where the convolution coefficients are predicted from mel-spectrograms. It speeds up the Parallel WaveGAN vocoder by 4x without any degradation in sound quality. Wang et al. [373] propose a semi-autoregressive mode for mel-spectrogram generation, where the mel-spectrograms are generated in an autoregressive mode for individual phoneme and non-autoregressive mode for different phonemes.

Speedup with Domain Knowledge Domain knowledge from speech can be leveraged to speed up inference, such as linear prediction [363], multiband modeling [418, 406, 60], subscale predic-

tion [150], multi-frame prediction [427, 382, 373, 126, 210], streaming synthesis [76], etc. LPCNet [363] combines digital signal processing with neural networks, by using linear prediction coefficients to calculate the next waveform and a lightweight model to predict the residual value, which can speed the inference of autoregressive waveform generation. Another technique that is widely used to speed up the inference of vocoders is subband modeling, which divides the waveform into multiple subbands for fast inference. Typical models include DurIAN [418], multi-band MelGAN [406], subband WaveNet [250], and multi-band LPCNet [348, 60]. Bunched LPCNet [370] reduces the computation complexity of LPCNet with sample bunching and bit bunching, achieving more than 2x speedup. Streaming TTS [76, 223, 322, 405, 323, 237] synthesizes speech once some input tokens are comming, without waiting for the whole input sentence, which can also speed up inference. FFTNet [145] uses a simple architecture to mimic the Fast Fourier Transform (FFT), which can generate audio samples in real-time. Okamoto et al. [251] further enhance FFTNet with noise shaping and subband techniques, improving the voice quality while keeping small model size. Popov et al. [274] propose frame splitting and cross-fading to synthesize some parts of the waveform in parallel and then concatenate the synthesized waveforms together to ensure fast synthesis on low-end devices. Kang et al. [152] accelerate DCTTS [332] with network reduction and fidelity improvement techniques such as group highway activation, which can synthesize speech in real time with a single CPU thread.

3.3 Low-Resource TTS

Building high-quality TTS systems usually requires a large amount of high-quality paired text and speech data. However, there are more than 7,000 languages in the world[14], and most languages are lack of training data for developing TTS systems. As a result, popular commercialized speech services[15] can only support dozens of languages for TTS. Supporting TTS for low-resource languages can not only have business value, but is also beneficial for social good. Thus, a lot of research works build TTS system under low data resource scenarios. We summarize some representative techniques for low-resource TTS in Table 9, and introduce these techniques as follows.

Table 9: Some representative techniques for 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. Although paired text and speech data are hard to collect, unpaired speech and text data (especially text data) are relatively easy to obtain. Self-supervised pre-training methods can be leveraged to enhance the language understanding or speech generation capabilities [52, 374, 440, 81]. For example, the text encoder in TTS can be enhanced by the pre-trained BERT models [52, 81, 143], and the speech decoder in TTS can be pre-trained through autoregressive mel-spectrogram prediction [52] or jointed trained with voice conversion task [440]. Besides, speech can be quantized into discrete token sequence to resemble the phoneme or character sequence [352]. In this way, the quantized discrete tokens and the speech can be regarded as pseudo paired data to pre-train a TTS model, which is then fine-tuned on few truly paired text and speech data [201, 358, 436].
- Cross-lingual transfer. Although paired text and speech data is scarce in low-resource languages, it is abundant in rich-resource languages. Since human languages share similar vocal organs, pronunciations [389] and semantic structures [341], pre-training the TTS models on rich-resource languages can help the mapping between text and speech in low-resource languages [43, 396, 12, 62, 101, 342, 442, 247, 407, 435]. Usually, there are different phoneme sets between rich- and low-resource languages. Thus, Chen et al. [43] propose to map the embeddings between the phoneme sets from different languages, and LRSpeech [396] discards the pre-trained phoneme

[14] https://www.ethnologue.com/browse

[15] For example, Microsoft Azure, Google Cloud, and Amazon AWS.

embeddings and initializes the phoneme embeddings from scratch for low-resource languages. International phonetic alphabet (IPA) [109] or byte representation [108] is adopted to support arbitrary texts in multiple languages. Besides, language similarity [341] can also be considered when conducting the cross-lingual transfer.

- Cross-speaker transfer. When a certain speaker has limited speech data, the data from other speakers can be leveraged to improve the synthesis quality of this speaker. This can be achieved by converting the voice of other speakers into this target voice through voice conversion to increase the training data [128], or by adapting the TTS models trained on other voices to this target voice through voice adaptation or voice cloning [44, 40] that are introduced in Section 3.6.
- Speech chain/Back transformation. Text to speech (TTS) and automatic speech recognition (ASR) are two dual tasks [285] and can be leveraged together to improve each other. Techniques like speech chain [350, 351] and back transformation [291, 396] leverage additional unpaired text and speech data to boost the performance of TTS and ASR.
- Dataset mining in the wild. In some scenarios, there may exist some low-quality paired text and speech data in the Web. Cooper [59], Hu et al. [122] propose to mine this kind of data and develop sophisticated techniques to train a TTS model. Some techniques such as speech enhancement [362], denoising [434], and disentangling [383, 120] can be leveraged to improve the quality of the speech data mined in the wild.

3.4 Robust TTS

A good TTS system should be robust to always generate "correct" speech according to text even when encountering corner cases. In neural TTS, robust issues such as word skipping, repeating, and attention collapse[16] often happen in acoustic models[17] when generating mel-spectrogram sequence from character/phoneme sequence. Basically speaking, the causes of these robust issues are from two categories: 1) The difficulty in learning the alignments between characters/phonemes and mel-spectrograms; 2) The exposure bias and error propagation problems incurred in autoregressive generation. Vocoders do not face severely robust issues, since the acoustic features and waveform are already aligned frame-wisely (i.e., each frame of acoustic features correspond to a certain number (hop size) of waveform points). Therefore, existing works on robust TTS address the above two problems respectively[18].

- For the alignment learning between characters/phonemes and mel-spectrograms, the works can be divided into two aspects: 1) enhancing the robustness of attention mechanism [382, 315, 303, 438, 332, 107, 39], and 2) removing attention and instead predicting duration explicitly to bridge the length mismatch between text and speech [290, 418, 69, 75].
- For the exposure bias and error propagation problems in autoregressive generation, the works can also be divided into two aspects: 1) improving autoregressive generation to alleviate the exposure bias and error propagation problems [99, 205, 209, 291], and 2) removing autoregressive generation and instead using non-autoregressive generation [290, 292, 268, 69].

We summarize some popular techniques in these categories to improve robustness, as shown in Table 10. The works addressing the two problems may have overlapping, e.g., some works may enhance the attention mechanism in AR or NAR generation, and similarly, the duration prediction can be applied in both AR and NAR generation. We review these categories as follows.

[16] Attention collapse means the generated speech has unintelligible gibberish, which is usually caused by the not focused attention on a single input token [107].

[17] Robust issues can also happened in neural vocoders, where the generated waveform could have some glitches such as hoarseness, metallic noise, jitter, or pitch breaking. However, they are not so severe as in acoustic models, and the reasons causing these issues are not clear and more likely to be repaired by universal vocoder modeling [215, 265, 137, 144] or sophisticated designs [35]. Thus, we mainly introduce the works addressing the robust issues in acoustic models in this survey.

[18] There are some other reasons that can cause robust issues, such as the test domain is not well covered by the training domain. Research works that scale to unseen domain can alleviate this issue, such as increasing the amount and diversity of the training data [130], adopting relative position encoding to support long sequence unseen in training [17, 430], etc.

Table 10: Categorization of the methods for 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]

3.4.1 Enhancing Attention

In autoregressive acoustic models, a lot of word skipping/repeating and attention collapse issues are caused by the incorrect attention alignments learned in encoder-decoder attention. To alleviate this problem, some properties of the alignments between text (characters/phonemes) sequence and mel-spectrogram sequence are considered [107]: 1) Local: one character/phoneme token can be aligned to one or multiple consecutive mel-spectrogram frames, while one mel-spectrogram frame can only be aligned to a single character/phoneme token, which can avoid the blurry attention and attention collapse; 2) Monotonic: if character A is behind character B, the mel-spectrogram corresponding to A is also behind that corresponding to B, which can avoid word repeating; 3) Complete: each character/phoneme token must be covered by at least one mel-spectrogram frame, which can avoid word skipping. We analyze the techniques to enhance attention (from Table 10) according to whether they satisfy the above three properties and list them in Table 11. We describe these techniques as follows.

Table 11: The techniques on enhancing attention and whether they satisfy the three properties (local/monotonic/complete).

Techniques	Local	Monotonic	Complete
Content-based attention	×	×	×
Location-based attention	×	✓	×
Content/Location hybrid attention	×	✓	×
Monotonic attention	✓	✓	×
Stepwise monotonic attention	✓	✓	✓
Windowing or off-diagonal penalty	×	×	×
Enhancing enc-dec connection	×	×	×
Positional attention	×	×	×
Predicting duration	✓	✓	✓

- Content-based attention. The early attention mechanisms adopted in TTS (e.g. Tacotron [382]) are content-based [14], where the attention distributions are determined by the degree of match between the hidden representations from the encoder and decoder. Content-based attention is suitable for the tasks such as neural machine translation [14, 368] where the alignments between the source and target tokens are purely based semantic meaning (content). However, for the tasks like automatic speech recognition [50, 34, 48] and text to speech synthesis [382], the alignments

between text and speech have some specific properties. For example, in TTS [107], the attention alignments should be local, monotonic, and complete. Therefore, advanced attention mechanisms should be designed to better leverage these properties.

- Location-based attention. Considering the alignments between text and speech are depending on their positions, location-based attention [93, 17] is proposed to leverage the positional information for alignment. Several TTS models such as Char2Wav [315], VoiceLoop [333], and MelNet [367] adopt the location-based attention. As we summarize in Table 11, location-based attention can ensure the monotonicity property if properly handled.
- Content/Location-based hybrid attention. To combine the advantages of content and location based attentions, Chorowski et al. [50], Shen et al. [303] introduce location sensitive attention: when calculating the current attention alignment, the previous attention alignment is used. In this way, the attention would be more stable due to monotonic alignment.
- Monotonic attention. For monotonic attention [288, 47, 107, 411, 347], the attention position is monotonically increasing, which also leverages the prior that the alignments between text and speech are monotonic. In this way, it can avoid the skipping and repeating issues. However, the completeness property cannot be guaranteed in the above monotonic attention. Therefore, He et al. [107] propose stepwise monotonic attention, where in each decoding step, the attention alignment position moves forward at most one step, and is not allowed to skip any input unit.
- Windowing or off-diagonal penalty. Since attention alignments are monotonic and diagonal, Chorowski et al. [50], Tachibana et al. [332], Zhang et al. [438], Ping et al. [270], Chen et al. [39] propose to restrict the attention on the source sequence into a window subset. In this way, the learning flexibility and difficulty are reduced. Chen et al. [39] use penalty loss for off-diagonal attention weights, by constructing a band mask and encouraging the attention weights to be distributed in the diagonal band.
- Enhancing encoder-decoder connection. Since speech has more correlation among adjacent frames, the decoder itself contains enough information to predict next frame, and thus tends to ignore the text information from encoder. Therefore, some works propose to enhance the connection between encoder and decoder, and thus can improve attention alignment. Wang et al. [382], Shen et al. [303] use multi-frame prediction that generates multiple non-overlapping output frames at each decoder step. In this way, in order to predict consecutive frames, the decoder is forced to leverage information from the encoder side, which can improve the alignment learning. Other works also use a large dropout in the prenet before the decoder [382, 303, 39], or a small hidden size in the prenet as a bottleneck [39], which can prevent simply copying the previous speech frame when predicting the current speech frame. The decoder will get more information from the encoder side, which benefits the alignment learning. Ping et al. [270], Chen et al. [39] propose to enhance the connection of the positional information between source and target sequences, which benefits the attention alignment learning. Liu et al. [203] leverage CTC [94] based ASR as a cycle loss to encourage the generated mel-spectrograms to contain text information, which can also enhance the encoder-decoder connection for better attention alignment.
- Positional attention. Some non-autoregressive generation models [268, 234] leverage position information as the query to attend the key and value from the encoder, which is another way to build the connection between encoder and decoder for parallel generation.

3.4.2 Replacing Attention with Duration Prediction

While improving the attention alignments between text and speech can alleviate the robust issues to some extent, it cannot totally avoid them. Thus, some works [290, 418, 159, 69] propose to totally remove the encoder-decoder attention, explicitly predict the duration of each character/phoneme, and expand the text hidden sequence according to the duration to match the length of mel-spectrogram sequence. After that, the model can generate mel-spectrogram sequence in an autoregressive or non-autoregressive manner. It is very interesting that the early SPSS uses duration for alignments, and then the sequence-to-sequence models remove duration but use attention instead, and the later TTS models discard attention and use duration again, which is a kind of technique renaissance.

Existing works to investigate the duration prediction in neural TTS can be categorized from two perspectives: 1) Using external alignment tools or jointly training to get the duration label. 2) Optimizing the duration prediction in an end-to-end way or using ground-truth duration in training

and predicted duration in inference. We summarize the works according to the two perspectives in Table 12, and describe them as follows.

Table 12: A category of neural TTS on duration prediction.

Perspective	Category	Work
External/Internal	External	FastSpeech 1/2 [290, 292], DurIAN [418], TalkNet [19], [361, 74, 304]
	Internal	AlignTTS [429], Glow-TTS [159], EATS [69], [235, 75]
E2E Optimization	Not E2E	[290, 361, 19, 292, 418, 194, 74, 304, 429, 197, 159]
	E2E	EATS [69], Parallel Tacotron 2 [75]

- External alignment. The works leveraging external alignment tools [387, 94, 232, 193] can be divided into several categories according to the used alignment tools: 1) Encoder-decoder attention: FastSpeech [290] obtains the duration label from the attention alignments of an autoregressive acoustic model. SpeedySpeech [361] follows similar pipeline of FastSpeech to extract the duration from an autoregressive teacher model, but replaces the whole network structure with purely CNN. 2) CTC alignment. Beliaev et al. [19] leverages a CTC [94] based ASR model to provide the alignments between phoneme and mel-spectrogram sequence. 3) HMM alignment: FastSpeech 2 [292] leverages the HMM based Montreal forced alignment (MFA) [232] to get the duration. Other works such as DurIAN [418], RobuTrans [194], Parallel Tacotron [74], and Non-Attentive Tacotron [304] use forced alignment or speech recognition tools to get the alignments.
- Internal alignment. AlignTTS [429] follows the basic model structure of FastSpeech, but leverages a dynamic programming based method to learn the alignments between text and mel-spectrogram sequences with multi-stage training. JDI-T [197] follows FastSpeech to extract duration from an autoregressive teacher model, but jointly trains the autoregressive and non-autoregressive models, which does not need two-stage training. Glow-TTS [159] leverages a novel monotonic alignment search to extract duration. EATS [69] leverages the interpolation and soft dynamic time warping (DTW) loss to optimize the duration prediction in a fully end-to-end way.
- Non end-to-end optimization. Typical duration prediction methods [290, 361, 19, 292, 418, 194, 74, 304, 429, 197, 159] usually use duration obtained from external/internal alignment tools for training, and use predicted duration for inference. The predicted duration is not end-to-end optimized by receiving guiding signal (gradients) from the mel-spectrogram loss.
- End-to-end optimization. In order to jointly optimize the duration to achieve better prosody, EATS [69] predicts the duration using an internal module and optimizes the duration end-to-end with the help of duration interpolation and soft DTW loss. Parallel Tacotron 2 [75] follows the practice of EATS to ensure differentiable duration prediction. Non-Attentive Tacotron [304] proposes a semi-supervised learning for duration prediction, where the predicted duration can be used for upsampling if no duration label available.

3.4.3 Enhancing AR Generation

Autoregressive sequence generation usually suffers from exposure bias and error propagation [20, 390]. Exposure bias refers to that the sequence generation model is usually trained by taking previous ground-truth value as input (i.e., teacher-forcing), but generates the sequence autoregressively by taking previous predicted value as input in inference. The mismatch between training and inference can cause error propagation in inference, where the prediction errors can accumulate quickly along the generated sequence.

Some works have investigated different methods to alleviate the exposure bias and error propagation issues. Guo et al. [99] leverage professor forcing [92] to alleviate the mismatch between the different distributions of real and predicted data. Liu et al. [209] conduct teacher-student distillation [111, 164, 343] to reduce the exposure bias problem, where the teacher is trained with teacher-forcing mode, and the student takes the previously predicted value as input and is optimized to reduce the distance of hidden states between the teacher and student models. Considering the right part of the generated mel-spectrogram sequence is usually worse than than that in the left part due to error propagation, some works leverage both left-to-right and right-to-left generations [344] for data augmentation [291] and regularization [452]. Vainer and Dušek [361] leverage some data augmentations to alleviate the exposure bias and error propagation issues, by adding some random Gaussian noises to each

input spectrogram pixel to simulate the prediction errors, and degrading the input spectrograms by randomly replacing several frames with random frames to encourage the model to use temporally more distant frames.

3.4.4 Replacing AR Generation with NAR Generation

Although the exposure bias and error propagation problems in AR generation can be alleviated through the above methods, the problems cannot be addressed thoroughly. Therefore, some works directly adopt non-autoregressive generation to avoid these issues. They can be divided into two categories according to the use of attention or duration prediction. Some works such as ParaNet [268] and Flow-TTS [234] uses positional attention [270] for the text and speech alignment in parallel generation. The remaining works such as FastSpeech [290, 292] and EATS [69] use duration prediction to bridge the length mismatch between text and speech sequences.

Based on the introductions in the above subsections, we have a new category of TTS according to the alignment learning and AR/NAR generation, as shown in Table 13: 1) AR + Attention, such as Tacotron [382, 303], DeepVoice 3 [270], and TransformerTTS [192]. 2) AR + Non-Attention (Duration), such as DurIAN [418], RobuTrans [194], and Non-Attentive Tacotron [304]. 3) Non-AR + Attention, such as ParaNet [268], Flow-TTS [234], and VARA-TTS [204]. 4) Non-AR + Non-Attention, such as FastSpeech 1/2 [290, 292], Glow-TTS [159], and EATS [69].

Table 13: A new category of TTS according to the alignment learning and AR/NAR generation.

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]

3.5 Expressive TTS

The goal of text to speech is to synthesize intelligible and natural speech. The naturalness largely depends on the expressiveness of synthesized voice, which is determined by multiple characteristics, such as content, timbre, prosody, emotion, and style, etc. The research on expressiveness TTS covers broad topics including modeling, disentangling, controlling, and transferring the content, timbre, prosody, style, and emotion, etc. We review those topics in this subsection.

A key for expressive speech synthesis is to handle the problem of one-to-many mapping, which refers to that there are multiple speech variations corresponding to the same text, in terms of duration, pitch, sound volume, speaker style, emotion, etc. Modeling the one-to-many mapping under the regular L1 loss [86, 360] without enough input information will cause over-smoothing mel-spectrogram prediction [353, 334], e,g., predicting the average mel-spectrograms in the dataset instead of capturing the expressiveness of every single speech utterance, which leads to low-quality and less expressive speech. Therefore, providing these variation information as input and better modeling these variation information are important to alleviate this problem and improve the expressiveness of synthesized speech. Furthermore, by providing variation information as input, we can disentangle, control, and transfer the variation information: 1) by adjusting these variation information (any specific speaker timbre, style, accent, speaking rate, etc) in inference, we can control the synthesized speech; 2) by providing the variation information corresponding to another style, we can transfer the voice to this style; 3) in order to achieve fine-grained voice control and transfer, we need to disentangle different variation information, such as content and prosody, timbre and noise, etc.

In the remaining parts of this subsection, we first conduct a comprehensive analysis on these variation information, and then introduce some advanced techniques for modeling, disentangling, controlling, and transferring these variation information.

3.5.1 Categorization of Variation Information

We first categorize the information needed to synthesize a voice into four aspects:

- Text information, which can be characters or phonemes, represents the content of the synthesized speech (i.e., what to say). Some works improve the representation learning of text through enhanced word embeddings or text pre-training [81, 104, 393, 143], aiming to improve the quality and expressiveness of synthesized speech.
- Speaker or timbre information, which represents the characteristics of speakers (i.e., who to say). Some multi-speaker TTS systems explicitly model the speaker representations through a speaker lookup table or speaker encoder [87, 270, 142, 240, 39].
- Prosody, style, and emotion information, which covers the intonation, stress, and rhythm of speech and represents how to say the text [371, 179]. Prosody/style/emotion is the key information to improve the expressiveness of speech and the vast majority of works on expressive TTS focus on improving the prosody/style/emotion of speech [309, 383, 321, 85, 359, 324].
- Recording devices or noise environments, which are the channels to convey speech, and are not related to the content/speaker/prosody of speech, but will affect speech quality. Research works in this area focus on disentangling, controlling, and denoising for clean speech synthesis [120, 40, 434].

3.5.2 Modeling Variation Information

Many methods have been proposed to model different types of variation information in different granularities, as shown in Table 14.

Table 14: Some perspectives of modeling variation information for expressive speech synthesis.

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]

Information Type We can categorize the works according to the types of information being modeled: 1) explicit information, where we can explicitly get the labels of these variation information, and 2) implicit information, where we can only implicitly obtain these variation information.

For explicit information, we directly use them as input to enhance the models for expressive synthesis. We can obtain these information through different ways: 1) Get the language ID, speaker ID, style, and prosody from labeling data [445, 247, 195, 39]. For example, the prosody information can be labeled according to some annotation schemas, such as ToBI [307], AuToBI [294], Tilt [345], INTSINT [112], and SLAM [249]. 2) Extract the pitch and energy information from speech and extract duration from paired text and speech data [290, 292, 181, 158, 239, 365].

In some situations, there are no explicit labels available, or explicit labeling usually causes much human effort and cannot cover the specific or fine-grained variation information. Thus, we can model the variation information implicitly from data. Typical implicit modeling methods include:

- Reference encoder [309, 383, 224, 142, 9, 49, 40, 102]. Skerry-Ryan et al. [309] define the prosody as the variation in speech signals that remains after removing variation due to text content, speaker

timbre, and channel effects, and model prosody through a reference encoder, which does not require explicit annotations. Specifically, it extracts prosody embeddings from a reference audio, and uses it as the input of decoder. During training, a ground-truth reference audio is used, and during inference, another refer audio is used to synthesize speech with similar prosody. Wang et al. [383] extract embeddings from a reference audio and use them as the query to attend (through Q/K/V based attention [368]) a banks of style tokens, and the attention results are used as the prosody condition of TTS models for expressive speech synthesis. The style tokens can increase the capacity and variation of TTS models to learn different kinds of styles, and enable the knowledge sharing across data samples in the dataset. Each token in the style token bank can learn different prosody representations, such as different speaking rates and emotions. During inference, it can use a reference audio to attend and extract prosody representations, or simply pick one or some style tokens to synthesize speech.

- Variational autoencoder [119, 4, 443, 120, 103, 324, 325, 74]. Zhang et al. [443] leverage VAE to model the variance information in the latent space with Gaussian prior as a regularization, which can enable expressive modeling and control on synthesized styles. Some works [4, 120, 2, 74] also leverage the VAE framework to better model the variance information for expressive synthesis.
- Advanced generative models [224, 186, 366, 234, 159, 70, 141, 185]. One way to alleviate the one-to-many mapping problem and combat over-smoothing prediction is to use advanced generative models to implicitly learn the variation information, which can better model the multi-modal distribution.
- Text pre-training [81, 104, 393, 143, 98, 454], which can provide better text representations by using pre-trained word embeddings or model parameters.

Information Granularity Variation information can be modeled in different granularities. We describe these information from coarse-grained to fine-grained levels: 1) Language level and speaker level [445, 247, 39], where multilingual and multispeaker TTS systems use language ID or speaker ID to differentiate languages and speakers. 2) Paragraph level [11, 395, 376], where a TTS model needs to consider the connections between utterances/sentences for long-form reading. 3) Utterance level [309, 383, 142, 321, 207, 40], where a single hidden vector is extracted from the reference speech to represent the timber/style/prosody of this utterance. 4) Word/syllable level [325, 116, 45, 335], which can model the fine-grained style/prosody information that cannot be covered by utterance level information. 5) Character/phoneme level [188, 324, 430, 325, 45, 40, 189], such as duration, pitch or prosody information. 6) Frame level [188, 158, 49, 434], the most fine-grained information. Some corresponding works on different granularities can be found in Table 14.

Furthermore, modeling the variance information with hierarchical structure that covers different granularities is helpful for expressive synthesis. Suni et al. [330] demonstrate that hierarchical structures of prosody intrinsically exist in spoken languages. Kenter et al. [158] predict prosody features from frame and phoneme levels to syllable level, and concatenate with word- and sentence-level features. Hono et al. [116] leverage a multi-grained VAE to obtain different time-resolution latent variables and sample finer-level latent variables from coarser-level ones (e.g., from utterance level to phrase level and then to word level). Sun et al. [325] use VAE to model variance information on both phoneme and word levels and combine them together to feed into the decoder. Chien and Lee [45] study on prosody prediction and propose a hierarchical structure from the word to phoneme level to improve the prosody prediction.

3.5.3 Disentangling, Controlling and Transferring

In this subsection, we review techniques on disentangling [224, 120, 281], controlling [359, 184, 246, 13, 273, 349, 196], and transferring [153, 134, 399, 6] variation information, as shown in Table 15.

Disentangling with Adversarial Training When multiple styles or prosody information are entangled together, it is necessary to disentangle them during training for better expressive speech synthesis and control. Ma et al. [224] enhance the content-style disentanglement ability and controllability with adversarial and collaborative games. Hsu et al. [120] leverage the VAE framework with adversarial training to disentangle noise from speaker information. Qian et al. [281] propose speechflow to disentangle the rhythm, pitch, content, and timbre using three bottleneck reconstructions. Zhang et al. [434] propose to disentangle noise from speaker with frame-level noise modeling and adversarial training.

Table 15: Some representative techniques for disentangling, controlling, and transferring in expressive speech synthesis.

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]

Cycle Consistency/Feedback Loss for Control When providing variance information such as style tag as input, the TTS models are supposed to synthesize speech with the corresponding style. However, if no constraint is added, the TTS models tend to ignore the variance information and the synthesized speech that does not follow the style. To enhance the contollability of the TTS models, some works propose to use cycle consistency or feedback loss to encourage the synthesized speech to contain the variance information in the input. Li et al. [195] conduct controllable emotional transfer by adding an emotion style classifier with feedback cycle, where the classifier encourages the TTS model to synthesize speech with specific emotion. Whitehill et al. [386] use style classifier to provide the feedback loss to encourage the speech synthesis of a given style. Meanwhile, it incorporates adversarial learning between different style classifiers to ensure the preservation of different styles from multiple reference audios. Liu et al. [202] use ASR to provide the feedback loss to train the unmatched text and speech, which aims to reduce the mismatch between training and inference, since random chosen audio is used as the reference in inference. Other works [244, 207, 30, 305, 399, 6] leverage the feedback loss to ensure the controllability on style and speaker embeddings, etc.

Semi-Supervised Learning for Control Some attributes used to control the speech include pitch, duration, energy, prosody, emotion, speaker, noise, etc. If we have the label for each attribute, we can easily control the synthesized speech, by using the tag as input for model training and using the corresponding tag to control the synthesized speech in inference. However, when there is no tag/label available, or only a part is available, how to disentangle and control these attributes are challenging. When partial label is available, Habib et al. [103] propose semi-supervised learning method to learn the latent of VAE model, in order to control attributes such as affect or speaking rate. When no label available, Hsu et al. [119] propose Gaussian mixture VAE models to disentangle different attributes, and Hsu et al. [120], Zhang et al. [434] leverage gradient reversal or adversarial training to disentangle speaker timbre from noise in order to synthesize clean speech for noisy speakers.

Changing Variance Information for Transfer We can transfer the style of synthesized speech by changing the variation information to different styles. If the variation information is provided in the labeled tag, we can use the speech and the corresponding tag in training, and transfer the style with corresponding tags in inference [445, 247, 195, 39]. Alternatively, if we do not have labeled tag for the variation information, we can get the variation information from speech during training, no matter through explicit or implicit modeling as introduced above: Pitch, duration and energy can be explicitly extracted from speech, and some latent representations can be implicitly extracted by reference encoder or VAE. In this way, in order to achieve style transfer in inference, we can obtain the variation information in three ways: 1) extracting from reference speech [309, 383, 142, 443, 49, 40, 399, 6]; 2) predicting from text [321, 290, 324, 430, 292, 40]; 3) obtaining by sampling from the latent space [383, 443, 119].

3.6 Adaptive TTS

Adaptive TTS[19] is an important feature for TTS that can synthesize voice for any user. It is known as different terms in academia and industry, such as voice adaptation [44], voice cloning [9], custom voice [40], etc. Adaptive TTS has been a hot research topic, e,g., a lot of works in statistic parametric speech synthesis have studied voice adaptation [79, 392, 450, 80, 67, 125], and the recent voice cloning challenge also attracts a lot of participants [394, 121, 337, 46]. In adaptive TTS scenario, a source TTS model (usually trained on a multi-speaker speech dataset) is usually adapted with few adaptation data for each target voice.

[19]Here we mainly discuss adaptive TTS for different voices, instead of languages, styles, domains, etc.

We review the works on adaptive TTS from two perspectives: 1) General adaptation setting, which covers the improvements of generalization of source TTS model to support new speakers, and the adaptation to different domains. 2) Efficient adaptation setting, which covers the reduction of adaptation data and adaptation parameters for each target speaker. We summarize the works in the two perspectives in Table 16 and introduce these works as follows.

Table 16: The research works in adaptive TTS from two perspectives.

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]

3.6.1 General Adaptation

Source Model Generalization The works in this category aim to improve the generalization of source TTS model. In source model training, the source text does not contain enough acoustic information such as prosody, speaker timbre, and recording environments to generate target speech. As a result, the TTS model is prone to overfit on the training data and has poor generalization for new speakers in adaptation. Chen et al. [40] propose acoustic condition modeling to provide necessary acoustic information as model input to learn the text-to-speech mapping with better generalization instead of memorizing. Another way to improve the generalization of source TTS model is to increase the amount and diversity of training data. Cooper et al. [57] leverage speaker augmentation to increase the number of speakers when training source TTS model, which can generalize well to unseen speakers in adaptation. Yang and He [407] train a universal TTS model with multiple speakers in 50 language locales, which increase the generalization when adapting to a new speaker.

Cross-Domain Adaptation In adaptive TTS, an important factor is that the adaptation speech has different acoustic conditions or styles with the speech data used to train the source TTS model. In this way, special designs need to be considered to improve the generalization of source TTS model and support the styles in target speakers. AdaSpeech [40] designs acoustic condition modeling to better model the acoustic conditions such as recording devices, environment noise, accents, speaker rates, speaker timbre, etc. In this way, the model tends to generalize instead of memorizing the acoustic conditions, and can be well adapted to the speech data with different acoustic conditions. AdaSpeech 3 [404] adapts a reading-style TTS model to spontaneous style, by designing specific filled pauses adaptation, rhythm adaptation, and timbre adaptation. Some other works [266, 123] consider the adaptation across different speaking styles, such as Lombard [266] or whisper [123]. Some works [445, 38, 212, 449, 110, 319, 225, 453, 109] propose to transfer voices across languages, e.g., synthesize Mandarin speech using an English speaker, where the English speaker does not have any Mandarin speech data.

3.6.2 Efficient Adaptation

Roughly speaking, more adaptation data will result in better voice quality, but incur high data collection cost. For adaptation parameters, the whole TTS model [44, 177], or part of the model (e.g., decoder) [240, 446], or only speaker embedding [9, 44, 40] can be fine-tuned. Similarly, fine-tuning more parameters will result in good voice quality, but increase the memory and deployment cost. In practice, we aim to adapt as few data and parameters as possible while achieving high adaptation voice quality. We divide the works in this category into several aspects: 1) few data adaptation; 2) few parameter adaptation; 3) untranscribed data adaptation; 4) zero-shot adaptation. We introduce these works as follows.

- Few data adaptation. Some works [44, 9, 177, 240, 446, 49, 46, 40, 236] conduct few-shot adaptation that only uses few paired text and speech data, varying from several minutes to several seconds. Chien et al. [46] explore different speaker embeddings for few-shot adaptation. Yue et al. [420] leverage speech chain [350] for few-shot adaptation. Chen et al. [40], Arık et al. [9] compare the voice quality with different amounts of adaptation data and find that voice quality improves quickly with the increase of adaptation data when data size is small (less than 20 sentences) and improves slowly with dozens of adaptation sentences.
- Few parameter adaptation. To support many users/customers, the adaptation parameters need to be small enough for each target speaker to reduce memory usage while maintaining high voice quality. For example, if each user/voice consumes 100MB parameters, the total memory storage equals to 100PB for 1M users, which is a huge memory cost. Some works propose to reduce the adaptation parameter as few as possible, while maintaining the adaptation quality. AdaSpeech [40] proposes conditional layer normalization to generate the scale and bias parameters in layer normalization from the speaker embeddings based on contextual parameter generation [272] and only fine-tune the parameters related to the conditional layer normalization and speaker embeddings to achieve good adaptation quality. Moss et al. [240] propose a fine-tuning method that selects different model hyperparameters for different speakers based on the Bayesian optimization, which achieves the goal of synthesizing the voice of a specific speaker with only a small number of speech samples.
- Untranscribed data adaptation. In many scenarios, only speech data can be collected such as in conversions or online meetings, without the corresponding transcripts. AdaSpeech 2 [403] leverages untranscribed speech data for voice adaptation, with the help of speech reconstruction and latent alignments [221]. Inoue et al. [133] use an ASR model to transcribe the speech data and use the transcribed paired data for voice adaptation.
- Zero-shot adaptation. Some works [9, 44, 142, 56, 32] conduct zero-shot adaptation, which leverage a speaker encoder to extract speaker embeddings given a reference audio. This scenario is quite appealing since no adaptation data and parameters are needed. However, the adaptation quality is not good enough especially when the target speaker is very different from the source speakers.

4 Resources

We collect some resources of TTS, including open-source implementations, TTS tutorials and keynotes, TTS challenges, and TTS corpora, as shown in Table 17.

Table 17: TTS resources.

Open-Source Implementations	
ESPnet-TTS [105]	https://github.com/espnet/espnet
Mozilla-TTS	https://github.com/mozilla/TTS
TensorflowTTS	https://github.com/TensorSpeech/TensorflowTTS
Coqui-TTS	https://github.com/coqui-ai/TTS
Parakeet	https://github.com/PaddlePaddle/Parakeet
NeMo	https://github.com/NVIDIA/NeMo
WaveNet	https://github.com/ibab/tensorflow-wavenet
WaveNet	https://github.com/r9y9/wavenet_vocoder
WaveNet	https://github.com/basveeling/wavenet
SampleRNN	https://github.com/soroushmehr/sampleRNN_ICLR2017
Char2Wav	https://github.com/sotelo/parrot
Tacotron	https://github.com/keithito/tacotron
Tacotron	https://github.com/Kyubyong/tacotron
Tacotron 2	https://github.com/Rayhane-mamah/Tacotron-2
Tacotron 2	https://github.com/NVIDIA/tacotron2
DeepVoice 3	https://github.com/r9y9/deepvoice3_pytorch
TransformerTTS	https://github.com/as-ideas/TransformerTTS
FastSpeech	https://github.com/xcmyz/FastSpeech
FastSpeech 2	https://github.com/ming024/FastSpeech2
MelGAN	https://github.com/descriptinc/melgan-neurips
MelGAN	https://github.com/seungwonpark/melgan
WaveRNN	https://github.com/fatchord/WaveRNN
LPCNet	https://github.com/mozilla/LPCNet
WaveGlow	https://github.com/NVIDIA/WaveGlow
FloWaveNet	https://github.com/ksw0306/FloWaveNet
WaveGAN	https://github.com/chrisdonahue/wavegan
GAN-TTS	https://github.com/r9y9/gantts
Parallel WaveGAN	https://github.com/kan-bayashi/ParallelWaveGAN
HiFi-GAN	https://github.com/jik876/hifi-gan

Glow-TTS	https://github.com/jaywalnut310/glow-tts
Flowtron	https://github.com/NVIDIA/flowtron
DiffWave	https://github.com/lmnt-com/diffwave
WaveGrad	https://github.com/ivanvovk/WaveGrad
VITS	https://github.com/jaywalnut310/vits
TTS Samples	https://github.com/seungwonpark/awesome-tts-samples
Software/Tool for Audio	https://github.com/faroit/awesome-python-scientific-audio

TTS Tutorials & Keynotes

TTS Tutorial at ISCSLP 2014 [282]	https://www.superlectures.com/iscslp2014/tutorial-4-deep-learning-for-speech-generation-and-synthesis
TTS Tutorial at ISCSLP 2016 [200]	http://staff.ustc.edu.cn/~zhling/download/ISCSLP16_tutorial_DLSPSS.pdf
TTS Tutorial at IEICE [378]	https://www.slideshare.net/jyamagis/tutorial-on-endtoend-texttospeech-synthesis-part-1-neural-waveform-modeling
Generative Models for Speech [21]	https://www.youtube.com/watch?v=vEAq_sBf1CA
Generative Model-Based TTS [423]	https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45882.pdf
Keynote at INTERSPEECH [354]	http://www.sp.nitech.ac.jp/~tokuda/INTERSPEECH2019.pdf
TTS Tutorial at ISCSLP 2021 [339]	https://www.microsoft.com/en-us/research/uploads/prod/2021/02/ISCSLP2021-TTS-Tutorial.pdf
TTS Webinar [338]	https://www.youtube.com/watch?v=MA8PCvmr8B0
TTS Tutorial at IJCAI 2021 [340]	https://tts-tutorial.github.io/ijcai2021/

TTS Challenges

Blizzard Challenge	http://www.festvox.org/blizzard/
Zero Resource Speech Challenge	https://www.zerospeech.com/
ICASSP2021 M2VoC	http://challenge.ai.iqiyi.com/detail?raceId=5fb2688224954e0b48431fe0
Voice Conversion Challenge	http://www.vc-challenge.org/

TTS Corpora

Corpus	#Hours	#Speakers	Sampling Rate (kHz)	Language
ARCTIC [173]	7	7	16	English
VCTK [369]	44	109	48	English
Blizzard-2011 [165]	16.6	1	16	English
Blizzard-2013 [166]	319	1	44.1	English
LJSpeech [136]	25	1	22.05	English
LibriSpeech [259]	982	2484	16	English
LibriTTS [428]	586	2456	24	English
VCC 2018 [214]	1	12	22.05	English
HiFi-TTS [16]	300	11	44.1	English
TED-LIUM [295]	118	666	/	English
CALLHOME [31]	60	120	8	English
RyanSpeech [421]	10	1	44.1	English
CSMSC [15]	12	1	48	Mandarin
HKUST [211]	200	2100	8	Mandarin
AISHELL-1 [28]	170	400	16	Mandarin
AISHELL-2 [71]	1000	1991	44.1	Mandarin
AISHELL-3 [305]	85	218	44.1	Mandarin
DiDiSpeech-1 [100]	572	4500	48	Mandarin
DiDiSpeech-2 [100]	227	1500	48	Mandarin
JSUT [314]	10	1	48	Japanese
KazakhTTS [243]	93	2	44.1/48	Kazakh
Ruslan [83]	31	1	44.1	Russian
HUI-Audio-Corpus [280]	326	122	44.1	German
India Corpus [106]	39	253	48	Multilingual
M-AILABS [88]	1000	/	16	Multilingual
MLS [278]	51K	6K	16	Multilingual
CSS10 [264]	140	1	22.05	Multilingual
CommonVoice [7]	2.5K	50K	48	Multilingual

5 Future Directions

In this paper, we conducted a survey on neural text to speech and mainly focused on (1) the basic models of TTS including text analysis, acoustic models, vocoders, and fully end-to-end models, and (2) several advanced topics including fast TTS, low-resource TTS, robust TTS, expressive TTS, and adaptive TTS. As a quick summary, we list representative TTS algorithms in Table 18. Due to page limitations, we only reviewed core algorithms of TTS; readers can refer to other papers for TTS related problems and applications, such as voice conversion [308], singing voice synthesis [115, 217, 35], talking face synthesis [36], etc.

We point out some future research directions on neural TTS, mainly in two categories according to the end goals of TTS.

High-quality speech synthesis The most important goal of TTS is to synthesize high-quality speech. The quality of speech is determined by many aspects that influence the perception of speech, including intelligibility, naturalness, expressiveness, prosody, emotion, style, robustness, controllability, etc. While neural approaches have significantly improved the quality of synthesized speech, there is still large room to make further improvements.

- *Powerful generative models*. TTS is a generation task, including the generation of waveform and/or acoustic features, which can be better handled by powerful generative models. Although advanced generative models based on VAE, GAN, flow, or diffusion have been adopted in acoustic models, vocoders and fully end-to-to models, research efforts on more powerful and efficient generative models are appealing to further improve the quality of synthesized speech.
- *Better representation learning*. Good representations of text and speech are beneficial for neural TTS models, which can improve the quality of synthesized speech. Some initial explorations on text pre-training indicate that better text representations can indeed improve the speech prosody. How to learn powerful representations for text/phoneme sequence and especially for speech sequence through unsupervised/self-supervised learning and pre-training is challenging and worth further explorations.
- *Robust speech synthesis*. While current TTS models eliminate word skipping and repeating issues caused by incorrect attention alignments, they still suffer from robustness issues when encountering corner cases that are not covered in the training set, such as longer text length, different text domains, etc. Improving the generalizability of the TTS model to different domains is critical for robust synthesis.
- *Expressive/controllable/transferrable speech synthesis*. The expressiveness, controllability and transferability of TTS models rely on better variation information modeling. Existing methods leverage reference encoder or explicit prosody features (e.g., pitch, duration, energy) for variation modeling, which enjoys good controllability and transferrability in inference but suffering from training/inference mismatch since ground-truth reference speech or prosody features used in training are usually unavailable in inference. Advanced TTS models capture the variation information implicitly, which enjoy good expressiveness in synthesized speech but perform not good in control and transfer, since sampling from latent space cannot explicitly and precisely control and transfer each prosody feature (e.g., pitch, style). How to design better methods for expressive/controllable/transferrable speech synthesis is also appealing.
- *More human-like speech synthesis*. Current speech recordings used in TTS training are usually in formal reading styles, where no pauses, repeats, changing speeds, varying emotions, and errors are permitted. However, in casual or conversational talking, human seldomly speaks like standard reading. Therefore, better modelling the casual, emotional, and spontaneous styles is critical to improve the naturalness of synthesized speech.

Efficient speech synthesis Once we can synthesize high-quality speech, the next most important task is efficient synthesis, i.e., how to reduce the cost of speech synthesis including the cost of collecting and labeling training data, training and serving TTS models, etc.

- *Data-efficient TTS*. Many low-resource languages are lack of training data. How to leverage unsupervised/semi-supervised learning and cross-lingual transfer learning to help the low-resource languages is an interesting direction. For example, the ZeroSpeech Challenge [432] is a good initiative to explore the techniques to learn only from speech, without any text or linguistic knowledge. Besides, in voice adaptation, a target speaker usually has little adaptation data, which is another application scenario for data-efficient TTS.
- *Parameter-efficient TTS*. Today's neural TTS systems usually employ large neural networks with tens of millions of parameters to synthesize high-quality speech, which block the applications in mobile, IoT and other low-end devices due to their limited memory and power consumption. Designing compact and lightweight models with less memory footprints, power consumption and latency are critical for those application scenarios.
- *Energy-efficient TTS*. Training and serving a high-quality TTS model consume a lot of energy and emit a lot of carbon. Improving energy efficiency, e.g., reducing the FLOPs in TTS training and

inference, is important to let more populations to benefit from advanced TTS techniques while reducing carbon emissions to protect our environment.

Table 18: Overview of TTS models. "AM" represents acoustic models, "Voc" represents vocoders, "E2E" represents fully end-to-end models, "ling" represents linguistic features, "ch" represents characters, "ph" represents phonemes, "ceps" represents cepstrums, "linS" represents linear-spectrograms, "melS" represents mel-spectrograms, "wav" represents waveform, "FF" represents feed-forward, "AR" represents autoregressive, "$\emptyset$" represents no conditional information, "IS" represents INTERSPEECH.

Model	AM/Voc	Data Flow	Publication	Time
WaveNet [254]	Voc	ling $\xrightarrow{\text{AR}}$ wav	SSW16	2016.09
SampleRNN [233]	Voc	$\emptyset$ $\xrightarrow{\text{AR}}$ wav	ICLR17	2016.12
Deep Voice [8]	AM+Voc	ch $\rightarrow$ ph $\rightarrow$ ling $\xrightarrow{\text{AR}}$ wav	ICML17	2017.02
Char2Wav [315]	E2E	ch $\xrightarrow{\text{AR}}$ ceps $\xrightarrow{\text{AR}}$ wav	ICLR17 WS	2017.02
Tacotron [382]	AM	ch/ph $\xrightarrow{\text{AR}}$ linS $\longrightarrow$ wav	IS17	2017.03
Deep Voice 2 [87]	AM+Voc	ch $\rightarrow$ ph $\xrightarrow{\text{FF}}$ ling $\xrightarrow{\text{AR}}$ wav	NIPS17	2017.05
DV2-Tacotron [87]	AM+Voc	ch $\xrightarrow{\text{AR}}$ linS $\xrightarrow{\text{AR}}$ wav	NIPS17	2017.05
VoiceLoop [333]	AM	ph $\rightarrow$ ceps $\longrightarrow$ wav	ICLR18	2017.07
Deep Voice 3 [270]	AM	ch/ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{AR}}$ wav	ICLR18	2017.10
DCTTS [332]	AM	ch $\xrightarrow{\text{AR}}$ melS $\longrightarrow$ wav	ICASSP18	2017.10
Par.WaveNet [255]	Voc	ling $\xrightarrow{\text{FF}}$ wav	ICML18	2017.11
Tacotron 2 [303]	AM	ch/ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{AR}}$ wav	ICASSP18	2017.12
WaveGAN [68]	Voc	$\emptyset$ $\xrightarrow{\text{FF}}$ wav	ICLR19	2018.02
WaveRNN [150]	Voc	ling $\xrightarrow{\text{AR}}$ wav	ICML18	2018.02
DV3-Clone [9]	AM	ch/ph $\xrightarrow{\text{AR}}$ linS $\rightarrow$ wav	NeurIPS18	2018.02
GST-Tacotron [383]	AM	ph $\xrightarrow{\text{AR}}$ melS $\rightarrow$ wav	ICML18	2018.03
Ref-Tacotron [309]	AM	ph $\xrightarrow{\text{AR}}$ melS $\rightarrow$ wav	ICML18	2018.03
FFTNet [145]	Voc	ceps $\xrightarrow{\text{AR}}$ wav	ICASSP18	2018.04
VAE-Loop [4]	AM	ph $\rightarrow$ ceps $\longrightarrow$ wav	IS18	2018.04
SV-Tacotron [142]	AM	ch/ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{AR}}$ wav	NeurIPS18	2018.06
ClariNet [269]	E2E	ch/ph $\xrightarrow{\text{AR}}$ wav	ICLR19	2018.07
ForwardAtt [438]	AM	ph $\xrightarrow{\text{AR}}$ linS $\rightarrow$ wav	ICASSP18	2018.07
MCNN [10]	Voc	linS $\xrightarrow{\text{FF}}$ wav	SPL18	2018.08
TransformerTTS [192]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{AR}}$ wav	AAAI19	2018.09
SEA-TTS [44]	Voc	ling $\xrightarrow{\text{AR}}$ wav	ICLR19	2018.09
GMVAE-Tacotron [119]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{AR}}$ wav	ICLR19	2018.10
LPCNet [363]	Voc	ceps $\xrightarrow{\text{AR}}$ wav	ICASSP19	2018.10
WaveGlow [279]	Voc	melS $\xrightarrow{\text{FF}}$ wav	ICASSP19	2018.10
FloWaveNet [163]	Voc	melS $\xrightarrow{\text{FF}}$ wav	ICML19	2018.11
Univ. WaveRNN [215]	Voc	melS $\xrightarrow{\text{AR}}$ wav	IS19	2018.11
VAE-TTS [443]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{AR}}$ wav	ICASSP19	2018.12
TTS-Stylization [224]	AM	ch $\xrightarrow{\text{AR}}$ melS $\rightarrow$ wav	ICLR19	2018.12
AdVoc [245]	Voc	melS $\xrightarrow{\text{FF}}$ linS $\rightarrow$ wav	IS19	2019.04
GAN Exposure [99]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{AR}}$ wav	IS19	2019.04
GELP [149]	Voc	melS $\xrightarrow{\text{FF}}$ wav	IS19	2019.04
Almost Unsup [291]	AM	ph $\xrightarrow{\text{AR}}$ melS $\rightarrow$ wav	ICML19	2019.05
FastSpeech [290]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	NeurIPS19	2019.05
ParaNet [268]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	ICML20	2019.05
WaveVAE [268]	Voc	melS $\xrightarrow{\text{FF}}$ wav	ICML20	2019.05
MelNet [367]	AM	ch $\xrightarrow{\text{AR}}$ melS $\longrightarrow$ wav	arXiv19	2019.06
StepwiseMA [107]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{AR}$ wav	IS19	2019.06

GAN-TTS [23]	Voc	ling $\xrightarrow{\text{FF}}$ wav	ICLR20	2019.09
DurIAN [418]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{AR}}$ wav	IS20	2019.09
MB WaveRNN [418]	Voc	melS $\xrightarrow{\text{AR}}$ wav	IS20	2019.09
MelGAN [178]	Voc	melS $\xrightarrow{\text{FF}}$ wav	NeurIPS19	2019.10
Para. WaveGAN [402]	Voc	melS $\xrightarrow{\text{FF}}$ wav	ICASSP20	2019.10
DCA-Tacotron [17]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{AR}}$ wav	ICASSP20	2019.10
WaveFlow [271]	Voc	melS $\xrightarrow{\text{AR}}$ wav	ICML20	2019.12
SqueezeWave [433]	Voc	melS $\xrightarrow{\text{FF}}$ wav	arXiv20	2020.01
AlignTTS [429]	AM	ch/ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	ICASSP20	2020.03
RobuTrans [194]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{AR}}$ wav	AAAI20	2020.04
Flow-TTS [234]	AM	ch/ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	ICASSP20	2020.05
Flowtron [366]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{FF}}$ wav	ICLR21	2020.05
Glow-TTS [159]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	NeurIPS20	2020.05
JDI-T [197]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	IS20	2020.05
TalkNet [19]	AM	ch $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	arXiv20	2020.05
MB MelGAN [406]	Voc	melS $\xrightarrow{\text{FF}}$ wav	SLT21	2020.05
MultiSpeech [39]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{FF}}$ wav	IS20	2020.06
FastSpeech 2 [292]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	ICLR21	2020.06
FastSpeech 2s [292]	E2E	ph $\xrightarrow{\text{FF}}$ wav	ICLR21	2020.06
EATS [69]	E2E	ch/ph $\xrightarrow{\text{FF}}$ wav	ICLR21	2020.06
FastPitch [181]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	ICASSP21	2020.06
VocGAN [408]	Voc	melS $\xrightarrow{\text{FF}}$ wav	IS20	2020.07
LRSpeech [396]	AM	ch $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{FF}}$ wav	KDD20	2020.08
SpeedySpeech [361]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	IS20	2020.08
GED [96]	Voc	ling $\xrightarrow{\text{FF}}$ wav	NeurIPS20	2020.08
SC-WaveRNN [265]	Voc	melS $\xrightarrow{\text{AR}}$ wav	IS20	2020.08
WaveGrad [41]	Voc	melS $\xrightarrow{\text{FF}}$ wav	ICLR21	2020.09
DiffWave [176]	Voc	melS $\xrightarrow{\text{FF}}$ wav	ICLR21	2020.09
HiFi-GAN [174]	Voc	melS $\xrightarrow{\text{FF}}$ wav	NeurIPS20	2020.10
NonAtt Tacotron [304]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{AR}}$ wav	arXiv20	2020.10
Para. Tacotron [74]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{AR}}$ wav	arXiv20	2020.10
DeviceTTS [126]	AM	ph $\xrightarrow{\text{AR}}$ Ceps $\rightarrow$ wav	arXiv20	2020.10
Wave-Tacotron [385]	E2E	ch/ph $\xrightarrow{\text{AR}}$ wav	ICASSP21	2020.11
DenoiSpeech [434]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	ICASSP21	2020.12
EfficientTTS [235]	AM	ch $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	ICML21	2020.12
EfficientTTS-Wav [235]	E2E	ch $\xrightarrow{\text{FF}}$ wav	ICML21	2020.12
Multi-SpectroGAN [186]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{FF}}$ wav	AAAI21	2020.12
LightSpeech [220]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	ICASSP21	2021.02
Para. Tacotron 2 [75]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{AR}}$ wav	arXiv21	2021.03
AdaSpeech [40]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	ICLR21	2021.03
BVAE-TTS [187]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	ICLR21	2021.03
PnG BERT [143]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{AR}}$ wav	IS21	2021.03
Fast DCTTS [152]	AM	ch $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{FF}}$ wav	ICASSP21	2021.04
AdaSpeech 2 [403]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	ICASSP21	2021.04
TalkNet 2 [18]	AM	ch $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	arXiv21	2021.04
Triple M [199]	AM+Voc	ch $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{AR}}$ wav	arXiv21	2021.04
Diff-TTS [141]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	arXiv21	2021.04
Grad-TTS [276]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	ICML21	2021.05
Fre-GAN [161]	Voc	melS $\xrightarrow{\text{FF}}$ wav	IS21	2021.06

VITS [160]	E2E	ph $\xrightarrow{\text{FF}}$ wav	ICML21	2021.06
AdaSpeech 3 [404]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	IS21	2021.06
PriorGrad-AM [185]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	arXiv21	2021.06
PriorGrad-Voc [185]	Voc	melS $\xrightarrow{\text{FF}}$ wav	arXiv21	2021.06
Meta-StyleSpeech [236]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	ICML21	2021.06

References

[1] Ronald Brian Adler, George R Rodman, and Alexandre Sévigny. *Understanding human communication*. Holt, Rinehart and Winston Chicago, 1991.

[2] Vatsal Aggarwal, Marius Cotescu, Nishant Prateek, Jaime Lorenzo-Trueba, and Roberto Barra-Chicote. Using vaes and normalizing flows for one-shot text-to-speech synthesis of expressive speech. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6179–6183. IEEE, 2020.

[3] Yang Ai and Zhen-Hua Ling. A neural vocoder with hierarchical generation of amplitude and phase spectra for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:839–851, 2020.

[4] Kei Akuzawa, Yusuke Iwasawa, and Yutaka Matsuo. Expressive speech synthesis via modeling expressions with variational autoencoder. *Proc. Interspeech 2018*, pages 3067–3071, 2018.

[5] Jonathan Allen, Sharon Hunnicutt, Rolf Carlson, and Bjorn Granstrom. Mitalk-79: The 1979 mit text-to-speech system. *The Journal of the Acoustical Society of America*, 65(S1): S130–S130, 1979.

[6] Xiaochun An, Frank K Soong, and Lei Xie. Improving performance of seen and unseen speech style transfer in end-to-end neural tts. *arXiv preprint arXiv:2106.10003*, 2021.

[7] Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4218–4222, 2020.

[8] Sercan Ö Arık, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, et al. Deep voice: Real-time neural text-to-speech. In *International Conference on Machine Learning*, pages 195–204. PMLR, 2017.

[9] Sercan Ö Arık, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. Neural voice cloning with a few samples. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 10040–10050, 2018.

[10] Sercan Ö Arık, Heewoo Jun, and Gregory Diamos. Fast spectrogram inversion using multi-head convolutional neural networks. *IEEE Signal Processing Letters*, 26(1):94–98, 2018.

[11] Adele Aubin, Alessandra Cervone, Oliver Watts, and Simon King. Improving speech synthesis with discourse relations. In *INTERSPEECH*, pages 4470–4474, 2019.

[12] Kurniawati Azizah, Mirna Adriani, and Wisnu Jatmiko. Hierarchical transfer learning for multilingual, multi-speaker, and style transfer dnn-based tts on low-resource languages. *IEEE Access*, 8:179798–179812, 2020.

[13] Jae-Sung Bae, Hanbin Bae, Young-Sun Joo, Junmo Lee, Gyeong-Hoon Lee, and Hoon-Young Cho. Speaking speed control of end-to-end speech synthesis using sentence-level conditioning. *Proc. Interspeech 2020*, pages 4402–4406, 2020.

[14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[15] Data Baker. Chinese standard mandarin speech corpus. https://www.data-baker.com/open_source.html, 2017.

[16] Evelina Bakhturina, Vitaly Lavrukhin, Boris Ginsburg, and Yang Zhang. Hi-fi multi-speaker english tts dataset. *arXiv preprint arXiv:2104.01497*, 2021.

[17] Eric Battenberg, RJ Skerry-Ryan, Soroosh Mariooryad, Daisy Stanton, David Kao, Matt Shannon, and Tom Bagby. Location-relative attention mechanisms for robust long-form speech synthesis. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6194–6198. IEEE, 2020.

[18] Stanislav Beliaev and Boris Ginsburg. Talknet 2: Non-autoregressive depth-wise separable convolutional model for speech synthesis with explicit pitch and duration prediction. *arXiv preprint arXiv:2104.08189*, 2021.

[19] Stanislav Beliaev, Yurii Rebryk, and Boris Ginsburg. Talknet: Fully-convolutional non-autoregressive speech synthesis model. *arXiv preprint arXiv:2005.05514*, 2020.

[20] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 1171–1179, 2015.

[21] Yoshua Bengio. Deep generative models for speech and images. https://www.youtube.com/watch?v=vEAq_sBf1CA, 2017.

[22] Mengxiao Bi, Heng Lu, Shiliang Zhang, Ming Lei, and Zhijie Yan. Deep feed-forward sequential memory networks for speech synthesis. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4794–4798. IEEE, 2018.

[23] Mikołaj Bińkowski, Jeff Donahue, Sander Dieleman, Aidan Clark, Erich Elsen, Norman Casagrande, Luis C Cobo, and Karen Simonyan. High fidelity speech synthesis with adversarial networks. In *International Conference on Learning Representations*, 2019.

[24] Maximilian Bisani and Hermann Ney. Joint-sequence models for grapheme-to-phoneme conversion. *Speech communication*, 50(5):434–451, 2008.

[25] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[26] Alan Black, Paul Taylor, Richard Caley, and Rob Clark. The festival speech synthesis system, 1998.

[27] Alan W Black, Heiga Zen, and Keiichi Tokuda. Statistical parametric speech synthesis. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–1229. IEEE, 2007.

[28] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*, pages 1–5. IEEE, 2017.

[29] Zexin Cai, Yaogen Yang, Chuxiong Zhang, Xiaoyi Qin, and Ming Li. Polyphone disambiguation for mandarin chinese using conditional neural network with multi-level embedding features. *Proc. Interspeech 2019*, pages 2110–2114, 2019.

[30] Zexin Cai, Chuxiong Zhang, and Ming Li. From speaker verification to multispeaker speech synthesis, deep transfer with feedback constraint. *Proc. Interspeech 2020*, pages 3974–3978, 2020.

[31] Alexandra Canavan, Graff David, and Zipperlen George. Callhome american english speech. https://catalog.ldc.upenn.edu/LDC97S42, 2021.

[32] Edresson Casanova, Christopher Shulby, Eren Gölge, Nicolas Michael Müller, Frederico Santos de Oliveira, Arnaldo Candido Junior, Anderson da Silva Soares, Sandra Maria Aluisio, and Moacir Antonelli Ponti. Sc-glowtts: an efficient zero-shot multi-speaker text-to-speech model. *arXiv preprint arXiv:2104.05557*, 2021.

[33] Moon-Jung Chae, Kyubyong Park, Jinhyun Bang, Soobin Suh, Jonghyuk Park, Namju Kim, and Longhun Park. Convolutional sequence to sequence model with non-sequential greedy decoding for grapheme to phoneme conversion. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2486–2490. IEEE, 2018.

[34] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 4960–4964. IEEE, 2016.

[35] Jiawei Chen, Xu Tan, Jian Luan, Tao Qin, and Tie-Yan Liu. Hifisinger: Towards high-fidelity neural singing voice synthesis. *arXiv preprint arXiv:2009.01776*, 2020.

[36] Lele Chen, Guofeng Cui, Ziyi Kou, Haitian Zheng, and Chenliang Xu. What comprises a good talking-head video generation?: A survey and benchmark. *arXiv preprint arXiv:2005.03201*, 2020.

[37] Liping Chen, Yan Deng, Xi Wang, Frank K Soong, and Lei He. Speech bert embedding for improving prosody in neural tts. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6563–6567. IEEE, 2021.

[38] Mengnan Chen, Minchuan Chen, Shuang Liang, Jun Ma, Lei Chen, Shaojun Wang, and Jing Xiao. Cross-lingual, multi-speaker text-to-speech synthesis using neural speaker embedding. *Proc. Interspeech 2019*, pages 2105–2109, 2019.

[39] Mingjian Chen, Xu Tan, Yi Ren, Jin Xu, Hao Sun, Sheng Zhao, and Tao Qin. Multispeech: Multi-speaker text to speech with transformer. In *INTERSPEECH*, pages 4024–4028, 2020.

[40] Mingjian Chen, Xu Tan, Bohan Li, Yanqing Liu, Tao Qin, sheng zhao, and Tie-Yan Liu. Adaspeech: Adaptive text to speech for custom voice. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=Drynvt7gg4L`.

[41] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. In *ICLR*, 2021.

[42] Stanley F Chen. Conditional and joint models for grapheme-to-phoneme conversion. In *Eighth European Conference on Speech Communication and Technology*, 2003.

[43] Yuan-Jui Chen, Tao Tu, Cheng-chieh Yeh, and Hung-Yi Lee. End-to-end text-to-speech for low-resource languages by cross-lingual transfer learning. *Proc. Interspeech 2019*, pages 2075–2079, 2019.

[44] Yutian Chen, Yannis Assael, Brendan Shillingford, David Budden, Scott Reed, Heiga Zen, Quan Wang, Luis C Cobo, Andrew Trask, Ben Laurie, et al. Sample efficient adaptive text-to-speech. In *International Conference on Learning Representations*, 2018.

[45] Chung-Ming Chien and Hung-yi Lee. Hierarchical prosody modeling for non-autoregressive speech synthesis. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 446–453. IEEE, 2021.

[46] Chung-Ming Chien, Jheng-Hao Lin, Chien-yu Huang, Po-chun Hsu, and Hung-yi Lee. Investigating on incorporating pretrained and learnable speaker representations for multi-speaker multi-style text-to-speech. *arXiv preprint arXiv:2103.04088*, 2021.

[47] Chung-Cheng Chiu and Colin Raffel. Monotonic chunkwise attention. In *International Conference on Learning Representations*, 2018.

[48] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778. IEEE, 2018.

[49] Seungwoo Choi, Seungju Han, Dongyoung Kim, and Sungjoo Ha. Attentron: Few-shot text-to-speech utilizing attention-based variable-length embedding. *Proc. Interspeech 2020*, pages 2007–2011, 2020.

[50] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 577–585, 2015.

[51] Min Chu and Yao Qian. Locating boundaries for prosodic constituents in unrestricted mandarin texts. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 6, Number 1, February 2001: Special Issue on Natural Language Processing Researches in MSRA*, pages 61–82, 2001.

[52] Yu-An Chung, Yuxuan Wang, Wei-Ning Hsu, Yu Zhang, and RJ Skerry-Ryan. Semi-supervised training for improving data efficiency in end-to-end speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6940–6944. IEEE, 2019.

[53] Cecil H Coker. A model of articulatory dynamics and control. *Proceedings of the IEEE*, 64(4): 452–460, 1976.

[54] Jian Cong, Shan Yang, Lei Xie, Guoqiao Yu, and Guanglu Wan. Data efficient voice cloning from noisy samples with domain adversarial training. *Proc. Interspeech 2020*, pages 811–815, 2020.

[55] Jian Cong, Shan Yang, Lei Xie, and Dan Su. Glow-wavegan: Learning speech representations from gan-based variational auto-encoder for high fidelity flow-based speech synthesis. *arXiv preprint arXiv:2106.10831*, 2021.

[56] Erica Cooper, Cheng-I Lai, Yusuke Yasuda, Fuming Fang, Xin Wang, Nanxin Chen, and Junichi Yamagishi. Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6184–6188. IEEE, 2020.

[57] Erica Cooper, Cheng-I Lai, Yusuke Yasuda, and Junichi Yamagishi. Can speaker augmentation improve multi-speaker end-to-end tts? *Proc. Interspeech 2020*, pages 3979–3983, 2020.

[58] Erica Cooper, Xin Wang, Yi Zhao, Yusuke Yasuda, and Junichi Yamagishi. Pretraining strategies, waveform model choice, and acoustic configurations for multi-speaker end-to-end speech synthesis. *arXiv preprint arXiv:2011.04839*, 2020.

[59] Erica Lindsay Cooper. *Text-to-speech synthesis using found data for low-resource languages*. PhD thesis, Columbia University, 2019.

[60] Yang Cui, Xi Wang, Lei He, and Frank K Soong. An efficient subband linear prediction for lpcnet-based neural synthesis. In *INTERSPEECH*, pages 3555–3559, 2020.

[61] Dongyang Dai, Li Chen, Yuping Wang, Mu Wang, Rui Xia, Xuchen Song, Zhiyong Wu, and Yuxuan Wang. Noise robust tts for low resource speakers using pre-trained model and speech enhancement. *arXiv preprint arXiv:2005.12531*, 2020.

[62] Marcel de Korte, Jaebok Kim, and Esther Klabbers. Efficient neural speech synthesis for low-resource languages through multilingual modeling. *Proc. Interspeech 2020*, pages 2967–2971, 2020.

[63] Ferdinand De Saussure. *Course in general linguistics*. Columbia University Press, 2011.

[64] Chuang Ding, Lei Xie, Jie Yan, Weini Zhang, and Yang Liu. Automatic prosody prediction for chinese speech synthesis using blstm-rnn and embedding features. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 98–102. IEEE, 2015.

[65] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[66] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[67] Rama Doddipatla, Norbert Braunschweiler, and Ranniery Maia. Speaker adaptation in dnn-based speech synthesis using d-vectors. In *INTERSPEECH*, pages 3404–3408, 2017.

[68] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. In *International Conference on Learning Representations*, 2018.

[69] Jeff Donahue, Sander Dieleman, Mikołaj Bińkowski, Erich Elsen, and Karen Simonyan. End-to-end adversarial text-to-speech. In *ICLR*, 2021.

[70] Chenpeng Du and Kai Yu. Mixture density network for phone-level prosody modelling in speech synthesis. *arXiv preprint arXiv:2102.00851*, 2021.

[71] Jiayu Du, Xingyu Na, Xuechen Liu, and Hui Bu. Aishell-2: Transforming mandarin asr research into industrial scale. *arXiv preprint arXiv:1808.10583*, 2018.

[72] Homer Dudley and Thomas H Tarnoczy. The speaking machine of wolfgang von kempelen. *The Journal of the Acoustical Society of America*, 22(2):151–166, 1950.

[73] Ewan Dunbar, Robin Algayres, Julien Karadayi, Mathieu Bernard, Juan Benjumea, Xuan-Nga Cao, Lucie Miskic, Charlotte Dugrain, Lucas Ondel, Alan W Black, et al. The zero resource speech challenge 2019: Tts without t. *Proc. Interspeech 2019*, pages 1088–1092, 2019.

[74] Isaac Elias, Heiga Zen, Jonathan Shen, Yu Zhang, Ye Jia, Ron Weiss, and Yonghui Wu. Parallel tacotron: Non-autoregressive and controllable tts. *arXiv preprint arXiv:2010.11439*, 2020.

[75] Isaac Elias, Heiga Zen, Jonathan Shen, Yu Zhang, Jia Ye, RJ Ryan, and Yonghui Wu. Parallel tacotron 2: A non-autoregressive neural tts model with differentiable duration modeling. *arXiv preprint arXiv:2103.14574*, 2021.

[76] Nikolaos Ellinas, Georgios Vamvoukakis, Konstantinos Markopoulos, Aimilios Chalamandaris, Georgia Maniati, Panos Kakoulidis, Spyros Raptis, June Sig Sung, Hyoungmin Park, and Pirros Tsiakoulis. High quality streaming speech synthesis with low, sentence-length-independent latency. *Proc. Interspeech 2020*, pages 2022–2026, 2020.

[77] Jesse Engel, Chenjie Gu, Adam Roberts, et al. Ddsp: Differentiable digital signal processing. In *International Conference on Learning Representations*, 2019.

[78] Yuchen Fan, Yao Qian, Feng-Long Xie, and Frank K Soong. Tts synthesis with bidirectional lstm based recurrent neural networks. In *Fifteenth annual conference of the international speech communication association*, 2014.

[79] Yuchen Fan, Yao Qian, Frank K Soong, and Lei He. Multi-speaker modeling and speaker adaptation for dnn-based tts synthesis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4475–4479. IEEE, 2015.

[80] Yuchen Fan, Yao Qian, Frank K Soong, and Lei He. Speaker and language factorization in dnn-based tts synthesis. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5540–5544. IEEE, 2016.

[81] Wei Fang, Yu-An Chung, and James Glass. Towards transfer learning for end-to-end speech synthesis from deep pre-trained language models. *arXiv preprint arXiv:1906.07307*, 2019.

[82] Toshiaki Fukada, Keiichi Tokuda, Takao Kobayashi, and Satoshi Imai. An adaptive algorithm for mel-cepstral analysis of speech. In *Proc. ICASSP*, volume 1, pages 137–140, 1992.

[83] Lenar Gabdrakhmanov, Rustem Garaev, and Evgenii Razinkov. Ruslan: Russian spoken language corpus for speech synthesis. In *International Conference on Speech and Computer*, pages 113–121. Springer, 2019.

[84] Michael Gadermayr, Maximilian Tschuchnig, Laxmi Gupta, Nils Krämer, Daniel Truhn, D Merhof, and Burkhard Gess. An asymmetric cycle-consistency loss for dealing with many-to-one mappings in image translation: a study on thigh mr scans. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 1182–1186. IEEE, 2021.

[85] Yang Gao, Weiyi Zheng, Zhaojun Yang, Thilo Kohler, Christian Fuegen, and Qing He. Interactive text-to-speech via semi-supervised style transfer learning. *arXiv preprint arXiv:2002.06758*, 2020.

[86] Saeed Gazor and Wei Zhang. Speech probability distribution. *IEEE Signal Processing Letters*, 10(7):204–207, 2003.

[87] Andrew Gibiansky, Sercan Ömer Arik, Gregory Frederick Diamos, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou. Deep voice 2: Multi-speaker neural text-to-speech. In *NIPS*, 2017.

[88] Munich Artificial Intelligence Laboratories GmbH. The m-ailabs speech dataset. `https://www.caito.de/2019/01/the-m-ailabs-speech-dataset/`, 2019.

[89] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[90] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[91] Prachi Govalkar, Johannes Fischer, Frank Zalkow, and Christian Dittmar. A comparison of recent neural vocoders for speech signal reconstruction. In *Proc. 10th ISCA Speech Synthesis Workshop*, pages 7–12, 2019.

[92] Anirudh Goyal, Alex Lamb, Ying Zhang, Saizheng Zhang, Aaron Courville, and Yoshua Bengio. Professor forcing: a new algorithm for training recurrent networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4608–4616, 2016.

[93] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[94] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[95] Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

[96] Alexey Gritsenko, Tim Salimans, Rianne van den Berg, Jasper Snoek, and Nal Kalchbrenner. A spectral energy distance for parallel speech synthesis. *Advances in Neural Information Processing Systems*, 33, 2020.

[97] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5769–5779, 2017.

[98] Haohan Guo, Frank K Soong, Lei He, and Lei Xie. Exploiting syntactic features in a parsed tree to improve end-to-end tts. *Proc. Interspeech 2019*, pages 4460–4464, 2019.

[99] Haohan Guo, Frank K Soong, Lei He, and Lei Xie. A new gan-based end-to-end tts training algorithm. *Proc. Interspeech 2019*, pages 1288–1292, 2019.

[100] Tingwei Guo, Cheng Wen, Dongwei Jiang, Ne Luo, Ruixiong Zhang, Shuaijiang Zhao, Wubo Li, Cheng Gong, Wei Zou, Kun Han, et al. Didispeech: A large scale mandarin speech corpus. *arXiv preprint arXiv:2010.09275*, 2020.

[101] Weitong Guo, Hongwu Yang, and Zhenye Gan. A dnn-based mandarin-tibetan cross-lingual speech synthesis. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1702–1707. IEEE, 2018.

[102] Siddharth Gururani, Kilol Gupta, Dhaval Shah, Zahra Shakeri, and Jervis Pinto. Prosody transfer in neural text to speech using global pitch and loudness features. *arXiv preprint arXiv:1911.09645*, 2019.

[103] Raza Habib, Soroosh Mariooryad, Matt Shannon, Eric Battenberg, RJ Skerry-Ryan, Daisy Stanton, David Kao, and Tom Bagby. Semi-supervised generative modeling for controllable speech synthesis. In *International Conference on Learning Representations*, 2019.

[104] Tomoki Hayashi, Shinji Watanabe, Tomoki Toda, Kazuya Takeda, Shubham Toshniwal, and Karen Livescu. Pre-trained text embeddings for enhanced text-to-speech synthesis. *Proc. Interspeech 2019*, pages 4430–4434, 2019.

[105] Tomoki Hayashi, Ryuichi Yamamoto, Katsuki Inoue, Takenori Yoshimura, Shinji Watanabe, Tomoki Toda, Kazuya Takeda, Yu Zhang, and Xu Tan. Espnet-tts: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7654–7658. IEEE, 2020.

[106] Fei He, Shan-Hui Cathy Chu, Oddur Kjartansson, Clara Rivera, Anna Katanova, Alexander Gutkin, Isin Demirsahin, Cibu Johny, Martin Jansche, Supheakmungkol Sarin, et al. Open-source multi-speaker speech corpora for building gujarati, kannada, malayalam, marathi, tamil and telugu speech synthesis systems. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6494–6503, 2020.

[107] Mutian He, Yan Deng, and Lei He. Robust sequence-to-sequence acoustic modeling with stepwise monotonic attention for neural tts. pages 1293–1297, 2019.

[108] Mutian He, Jingzhou Yang, and Lei He. Multilingual byte2speech text-to-speech models are few-shot spoken language learners. *arXiv preprint arXiv:2103.03541*, 2021.

[109] Hamed Hemati and Damian Borth. Using ipa-based tacotron for data efficient cross-lingual speaker adaptation and pronunciation enhancement. *arXiv preprint arXiv:2011.06392*, 2020.

[110] Ivan Himawan, Sandesh Aryal, Iris Ouyang, Sam Kang, Pierre Lanchantin, and Simon King. Speaker adaptation of a multilingual acoustic model for cross-language synthesis. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7629–7633. IEEE, 2020.

[111] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[112] Daniel Hirst. Automatic analysis of prosody for multilingual speech corpora. *Improvements in speech synthesis*, pages 320–327, 2001.

[113] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.

[114] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997.

[115] Yukiya Hono, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda. Singing voice synthesis based on generative adversarial networks. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6955–6959. IEEE, 2019.

[116] Yukiya Hono, Kazuna Tsuboi, Kei Sawada, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda. Hierarchical multi-grained generative model for expressive speech synthesis. *Proc. Interspeech 2020*, pages 3441–3445, 2020.

[117] Po-chun Hsu and Hung-yi Lee. Wg-wavenet: Real-time high-fidelity speech synthesis without gpu. *Proc. Interspeech 2020*, pages 210–214, 2020.

[118] Po-chun Hsu, Chun-hsuan Wang, Andy T Liu, and Hung-yi Lee. Towards robust neural vocoding for speech generation: A survey. *arXiv preprint arXiv:1912.02461*, 2019.

[119] Wei-Ning Hsu, Yu Zhang, Ron J Weiss, Heiga Zen, Yonghui Wu, Yuxuan Wang, Yuan Cao, Ye Jia, Zhifeng Chen, Jonathan Shen, et al. Hierarchical generative modeling for controllable speech synthesis. In *International Conference on Learning Representations*, 2018.

[120] Wei-Ning Hsu, Yu Zhang, Ron J Weiss, Yu-An Chung, Yuxuan Wang, Yonghui Wu, and James Glass. Disentangling correlated speaker and noise for speech synthesis via data augmentation and adversarial factorization. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5901–5905. IEEE, 2019.

[121] Cheng-Hung Hu, Yi-Chiao Wu, Wen-Chin Huang, Yu-Huai Peng, Yu-Wen Chen, Pin-Jui Ku, Tomoki Toda, Yu Tsao, and Hsin-Min Wang. The as-nu system for the m2voc challenge. *arXiv preprint arXiv:2104.03009*, 2021.

[122] Qiong Hu, Erik Marchi, David Winarsky, Yannis Stylianou, Devang Naik, and Sachin Kajarekar. Neural text-to-speech adaptation from low quality public recordings. In *Speech Synthesis Workshop*, volume 10, 2019.

[123] Qiong Hu, Tobias Bleisch, Petko Petkov, Tuomo Raitio, Erik Marchi, and Varun Lakshminarasimhan. Whispered and lombard neural speech synthesis. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 454–461. IEEE, 2021.

[124] Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. In *International Conference on Machine Learning*, pages 2078–2087. PMLR, 2018.

[125] Zhiying Huang, Heng Lu, Ming Lei, and Zhijie Yan. Linear networks based speaker adaptation for speech synthesis. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5319–5323. IEEE, 2018.

[126] Zhiying Huang, Hao Li, and Ming Lei. Devicetts: A small-footprint, fast, stable network for on-device text-to-speech. *arXiv preprint arXiv:2010.15311*, 2020.

[127] Andrew J Hunt and Alan W Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 373–376. IEEE, 1996.

[128] Goeric Huybrechts, Thomas Merritt, Giulia Comini, Bartek Perz, Raahil Shah, and Jaime Lorenzo-Trueba. Low-resource expressive text-to-speech using data augmentation. *arXiv preprint arXiv:2011.05707*, 2020.

[129] Min-Jae Hwang, Eunwoo Song, Ryuichi Yamamoto, Frank Soong, and Hong-Goo Kang. Improving lpcnet-based text-to-speech with linear prediction-structured mixture density network. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7219–7223. IEEE, 2020.

[130] Min-Jae Hwang, Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. Tts-by-tts: Tts-driven data augmentation for fast and high-quality speech synthesis. *arXiv preprint arXiv:2010.13421*, 2020.

[131] Satoshi Imai. Cepstral analysis synthesis on the mel frequency scale. In *ICASSP'83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 8, pages 93–96. IEEE, 1983.

[132] Satoshi Imai, Kazuo Sumita, and Chieko Furuichi. Mel log spectrum approximation (mlsa) filter for speech synthesis. *Electronics and Communications in Japan (Part I: Communications)*, 66(2):10–18, 1983.

[133] Katsuki Inoue, Sunao Hara, Masanobu Abe, Tomoki Hayashi, Ryuichi Yamamoto, and Shinji Watanabe. Semi-supervised speaker adaptation for end-to-end speech synthesis with pretrained models. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7634–7638. IEEE, 2020.

[134] Katsuki Inoue, Sunao Hara, Masanobu Abe, Nobukatsu Hojo, and Yusuke Ijima. Model architectures to extrapolate emotional expressions in dnn-based text-to-speech. *Speech Communication*, 126:35–43, 2021.

[135] Fumitada Itakura. Line spectrum representation of linear predictor coefficients of speech signals. *The Journal of the Acoustical Society of America*, 57(S1):S35–S35, 1975.

[136] Keith Ito. The lj speech dataset. `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[137] Won Jang, Dan Lim, and Jaesam Yoon. Universal melgan: A robust neural vocoder for high-fidelity waveform generation in multiple domains. *arXiv preprint arXiv:2011.09631*, 2020.

[138] Artur Janicki. Application of neural networks for pos tagging and intonation control in speech synthesis for polish. *Soft Computing and Intelligent Systems (SCIS 2004)*, 7, 2004.

[139] Chrisina Jayne, Andreas Lanitis, and Chris Christodoulou. One-to-many neural network mapping techniques for face image synthesis. *Expert Systems with Applications*, 39(10): 9778–9787, 2012.

[140] Je Hun Jeon and Yang Liu. Automatic prosodic events detection using syllable-based acoustic and syntactic features. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4565–4568. IEEE, 2009.

[141] Myeonghun Jeong, Hyeongju Kim, Sung Jun Cheon, Byoung Jin Choi, and Nam Soo Kim. Diff-tts: A denoising diffusion model for text-to-speech. *arXiv preprint arXiv:2104.01409*, 2021.

[142] Ye Jia, Yu Zhang, Ron J Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 4485–4495, 2018.

[143] Ye Jia, Heiga Zen, Jonathan Shen, Yu Zhang, and Yonghui Wu. Png bert: Augmented bert on phonemes and graphemes for neural tts. *arXiv preprint arXiv:2103.15060*, 2021.

[144] Yunlong Jiao, Adam Gabrys, Georgi Tinchev, Bartosz Putrycz, Daniel Korzekwa, and Viacheslav Klimkov. Universal neural vocoding with parallel wavenet. *arXiv preprint arXiv:2102.01106*, 2021.

[145] Zeyu Jin, Adam Finkelstein, Gautham J Mysore, and Jingwan Lu. Fftnet: A real-time speaker-dependent neural vocoder. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2251–2255. IEEE, 2018.

[146] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.

[147] Dan Jurafsky. *Speech & language processing*. Pearson Education India, 2000.

[148] Lauri Juvela, Bajibabu Bollepalli, Vassilis Tsiaras, and Paavo Alku. Glotnet—a raw waveform model for the glottal excitation in statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(6):1019–1030, 2019.

[149] Lauri Juvela, Bajibabu Bollepalli, Junichi Yamagishi, and Paavo Alku. Gelp: Gan-excited linear prediction for speech synthesis from mel-spectrogram. *Proc. Interspeech 2019*, pages 694–698, 2019.

[150] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron Oord, Sander Dieleman, and Koray Kavukcuoglu. Efficient neural audio synthesis. In *International Conference on Machine Learning*, pages 2410–2419. PMLR, 2018.

[151] Hiroki Kanagawa and Yusuke Ijima. Lightweight lpcnet-based neural vocoder with tensor decomposition. *Proc. Interspeech 2020*, pages 205–209, 2020.

[152] Minsu Kang, Jihyun Lee, Simin Kim, and Injung Kim. Fast dctts: Efficient deep convolutional text-to-speech. *arXiv preprint arXiv:2104.00624*, 2021.

[153] Sri Karlapati, Alexis Moinet, Arnaud Joly, Viacheslav Klimkov, Daniel Sáez-Trigueros, and Thomas Drugman. Copycat: Many-to-many fine-grained prosody transfer for neural text-to-speech. *Proc. Interspeech 2020*, pages 4387–4391, 2020.

[154] Kyle Kastner, João Felipe Santos, Yoshua Bengio, and Aaron Courville. Representation mixing for tts synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5906–5910. IEEE, 2019.

[155] Hideki Kawahara. Straight, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds. *Acoustical science and technology*, 27(6):349–353, 2006.

[156] Hideki Kawahara, Ikuyo Masuda-Katsuse, and Alain De Cheveigne. Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds. *Speech communication*, 27(3-4):187–207, 1999.

[157] Hideki Kawahara, Jo Estill, and Osamu Fujimura. Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system straight. In *Second International Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications*, 2001.

[158] Tom Kenter, Vincent Wan, Chun-An Chan, Rob Clark, and Jakub Vit. Chive: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network. In *International Conference on Machine Learning*, pages 3331–3340. PMLR, 2019.

[159] Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon. Glow-tts: A generative flow for text-to-speech via monotonic alignment search. *Advances in Neural Information Processing Systems*, 33, 2020.

[160] Jaehyeon Kim, Jungil Kong, and Juhee Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. *arXiv preprint arXiv:2106.06103*, 2021.

[161] Ji-Hoon Kim, Sang-Hoon Lee, Ji-Hyun Lee, and Seong-Whan Lee. Fre-gan: Adversarial frequency-consistent audio synthesis. *arXiv preprint arXiv:2106.02297*, 2021.

[162] Minchan Kim, Sung Jun Cheon, Byoung Jin Choi, Jong Jin Kim, and Nam Soo Kim. Expressive text-to-speech using style tag. *arXiv preprint arXiv:2104.00436*, 2021.

[163] Sungwon Kim, Sang-Gil Lee, Jongyoon Song, Jaehyeon Kim, and Sungroh Yoon. Flowavenet: A generative flow for raw audio. In *International Conference on Machine Learning*, pages 3370–3378. PMLR, 2019.

[164] Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, 2016.

[165] S. King and V. Karaiskos. The blizzard challenge 2011. In *Blizzard Challenge Workshop*, 2011.

[166] S. King and V. Karaiskos. The blizzard challenge 2013. In *Blizzard Challenge Workshop*, 2013.

[167] Diederik P Kingma and Prafulla Dhariwal. Glow: generative flow with invertible $1\times$ 1 convolutions. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 10236–10245, 2018.

[168] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[169] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in Neural Information Processing Systems*, 29:4743–4751, 2016.

[170] Lawrence E Kinsler, Austin R Frey, Alan B Coppens, and James V Sanders. *Fundamentals of acoustics*. John wiley & sons, 1999.

[171] Dennis H Klatt. Software for a cascade/parallel formant synthesizer. *the Journal of the Acoustical Society of America*, 67(3):971–995, 1980.

[172] Dennis H Klatt. Review of text-to-speech conversion for english. *The Journal of the Acoustical Society of America*, 82(3):737–793, 1987.

[173] John Kominek, Alan W Black, and Ver Ver. Cmu arctic databases for speech synthesis. 2003.

[174] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33, 2020.

[175] Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models. *arXiv preprint arXiv:2106.00132*, 2021.

[176] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *ICLR*, 2021.

[177] Zvi Kons, Slava Shechtman, Alex Sorin, Carmel Rabinovitz, and Ron Hoory. High quality, lightweight and adaptable tts using lpcnet. *Proc. Interspeech 2019*, pages 176–180, 2019.

[178] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. In *NeurIPS*, 2019.

[179] D Robert Ladd. *Intonational phonology*. Cambridge University Press, 2008.

[180] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[181] Adrian Łańcucki. Fastpitch: Parallel text-to-speech with pitch prediction. *arXiv preprint arXiv:2006.06873*, 2020.

[182] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning*, pages 1558–1566. PMLR, 2016.

[183] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553): 436–444, 2015.

[184] Keon Lee, Kyumin Park, and Daeyoung Kim. Styler: Style modeling with rapidity and robustness via speech decomposition for expressive and controllable neural text to speech. *arXiv preprint arXiv:2103.09474*, 2021.

[185] Sang-gil Lee, Heeseung Kim, Chaehun Shin, Xu Tan, Chang Liu, Qi Meng, Tao Qin, Wei Chen, Sungroh Yoon, and Tie-Yan Liu. Priorgrad: Improving conditional denoising diffusion models with data-driven adaptive prior. *arXiv preprint arXiv:2106.06406*, 2021.

[186] Sang-Hoon Lee, Hyun-Wook Yoon, Hyeong-Rae Noh, Ji-Hoon Kim, and Seong-Whan Lee. Multi-spectrogan: High-diversity and high-fidelity spectrogram generation with adversarial style combination for speech synthesis. *arXiv preprint arXiv:2012.07267*, 2020.

[187] Yoonhyung Lee, Joongbo Shin, and Kyomin Jung. Bidirectional variational inference for non-autoregressive text-to-speech. In *International Conference on Learning Representations*, 2020.

[188] Younggun Lee and Taesu Kim. Robust and fine-grained prosody control of end-to-end speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5911–5915. IEEE, 2019.

[189] Yi Lei, Shan Yang, and Lei Xie. Fine-grained emotion strength transfer, control and prediction for emotional speech synthesis. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 423–430. IEEE, 2021.

[190] Gina-Anne Levow. Automatic prosodic labeling with conditional random fields and rich acoustic features. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*, 2008.

[191] Hao Li, Yongguo Kang, and Zhenyu Wang. Emphasis: An emotional phoneme-based acoustic model for speech synthesis system. *Proc. Interspeech 2018*, pages 3077–3081, 2018.

[192] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. Neural speech synthesis with transformer network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6706–6713, 2019.

[193] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, Ming Liu, and Ming Zhou. Moboaligner: A neural alignment model for non-autoregressive tts with monotonic boundary search. *Proc. Interspeech 2020*, pages 3999–4003, 2020.

[194] Naihan Li, Yanqing Liu, Yu Wu, Shujie Liu, Sheng Zhao, and Ming Liu. Robutrans: A robust transformer-based text-to-speech model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8228–8235, 2020.

[195] Tao Li, Shan Yang, Liumeng Xue, and Lei Xie. Controllable emotion transfer for end-to-end speech synthesis. In *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 1–5. IEEE, 2021.

[196] Xiang Li, Changhe Song, Jingbei Li, Zhiyong Wu, Jia Jia, and Helen Meng. Towards multi-scale style control for expressive speech synthesis. *arXiv preprint arXiv:2104.03521*, 2021.

[197] Dan Lim, Won Jang, O Gyeonghwan, Heayoung Park, Bongwan Kim, and Jaesam Yoon. Jdi-t: Jointly trained duration informed transformer for text-to-speech without explicit alignment. *Proc. Interspeech 2020*, pages 4004–4008, 2020.

[198] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

[199] Shilun Lin, Fenglong Xie, Li Meng, Xinhui Li, and Li Lu. Triple m: A practical text-to-speech synthesis system with multi-guidance attention and multi-band multi-time lpcnet. *arXiv preprint arXiv:2102.00247*, 2021.

[200] Zhen-Hua Ling. Deep learning for statistical parametric speech synthesis. 2016.

[201] Alexander H Liu, Tao Tu, Hung-yi Lee, and Lin-shan Lee. Towards unsupervised speech recognition and synthesis with quantized speech representation learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7259–7263. IEEE, 2020.

[202] Da-Rong Liu, Chi-Yu Yang, Szu-Lin Wu, and Hung-Yi Lee. Improving unsupervised style transfer in end-to-end speech synthesis with end-to-end speech recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 640–647. IEEE, 2018.

[203] Peng Liu, Xixin Wu, Shiyin Kang, Guangzhi Li, Dan Su, and Dong Yu. Maximizing mutual information for tacotron. *arXiv preprint arXiv:1909.01145*, 2019.

[204] Peng Liu, Yuewen Cao, Songxiang Liu, Na Hu, Guangzhi Li, Chao Weng, and Dan Su. Vara-tts: Non-autoregressive text-to-speech synthesis based on very deep vae with residual attention. *arXiv preprint arXiv:2102.06431*, 2021.

[205] Renyuan Liu, Jian Yang, and Mengyuan Liu. A new end-to-end long-time speech synthesis system based on tacotron2. In *Proceedings of the 2019 International Symposium on Signal Processing Systems*, pages 46–50, 2019.

[206] Rui Liu, Berrak Sisman, Feilong Bao, Guanglai Gao, and Haizhou Li. Modeling prosodic phrasing with multi-task learning in tacotron-based tts. *IEEE Signal Processing Letters*, 27: 1470–1474, 2020.

[207] Rui Liu, Berrak Sisman, Guanglai Gao, and Haizhou Li. Expressive tts training with frame and style reconstruction loss. *arXiv preprint arXiv:2008.01490*, 2020.

[208] Rui Liu, Berrak Sisman, and Haizhou Li. Graphspeech: Syntax-aware graph attention network for neural speech synthesis. *arXiv preprint arXiv:2010.12423*, 2020.

[209] Rui Liu, Berrak Sisman, Jingdong Li, Feilong Bao, Guanglai Gao, and Haizhou Li. Teacher-student training for robust tacotron-based tts. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6274–6278. IEEE, 2020.

[210] Rui Liu, Berrak Sisman, Yixing Lin, and Haizhou Li. Fasttalker: A neural text-to-speech architecture with shallow and group autoregression. *Neural Networks*, 141:306–314, 2021.

[211] Yi Liu, Pascale Fung, Yongsheng Yang, Christopher Cieri, Shudong Huang, and David Graff. Hkust/mts: A very large scale mandarin telephone speech corpus. In *International Symposium on Chinese Spoken Language Processing*, pages 724–735. Springer, 2006.

[212] Zhaoyu Liu and Brian Mak. Cross-lingual multi-speaker text-to-speech synthesis for voice cloning without using parallel corpus for unseen speakers. *arXiv preprint arXiv:1911.11601*, 2019.

[213] Zhijun Liu, Kuan Chen, and Kai Yu. Neural homomorphic vocoder. *Proc. Interspeech 2020*, pages 240–244, 2020.

[214] Jaime Lorenzo-Trueba, Junichi Yamagishi, Tomoki Toda, Daisuke Saito, Fernando Villavicencio, Tomi Kinnunen, and Zhenhua Ling. The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods. In *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, pages 195–202, 2018.

[215] Jaime Lorenzo-Trueba, Thomas Drugman, Javier Latorre, Thomas Merritt, Bartosz Putrycz, Roberto Barra-Chicote, Alexis Moinet, and Vatsal Aggarwal. Towards achieving robust universal neural vocoding. *Proc. Interspeech 2019*, pages 181–185, 2019.

[216] Chunhui Lu, Pengyuan Zhang, and Yonghong Yan. Self-attention based prosodic boundary prediction for chinese speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7035–7039. IEEE, 2019.

[217] Peiling Lu, Jie Wu, Jian Luan, Xu Tan, and Li Zhou. Xiaoicesing: A high-quality and integrated singing voice synthesis system. *Proc. Interspeech 2020*, pages 1306–1310, 2020.

[218] Yanfeng Lu, Minghui Dong, and Ying Chen. Implementing prosodic phrasing in chinese end-to-end speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7050–7054. IEEE, 2019.

[219] Renqian Luo, Xu Tan, Rui Wang, Tao Qin, Enhong Chen, and Tie-Yan Liu. Neural architecture search with gbdt. *arXiv preprint arXiv:2007.04785*, 2020.

[220] Renqian Luo, Xu Tan, Rui Wang, Tao Qin, Jinzhu Li, Sheng Zhao, Enhong Chen, and Tie-Yan Liu. Lightspeech: Lightweight and fast text to speech with neural architecture search. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021.

[221] Hieu-Thi Luong and Junichi Yamagishi. Nautilus: a versatile voice cloning system. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2967–2981, 2020.

[222] Hieu-Thi Luong, Xin Wang, Junichi Yamagishi, and Nobuyuki Nishizawa. Training multi-speaker neural text-to-speech systems using speaker-imbalanced speech corpora. *Proc. Interspeech 2019*, pages 1303–1307, 2019.

[223] Mingbo Ma, Baigong Zheng, Kaibo Liu, Renjie Zheng, Hairong Liu, Kainan Peng, Kenneth Church, and Liang Huang. Incremental text-to-speech synthesis with prefix-to-prefix framework. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3886–3896, 2020.

[224] Shuang Ma, Daniel Mcduff, and Yale Song. Neural tts stylization with adversarial and collaborative games. In *International Conference on Learning Representations*, 2018.

[225] Soumi Maiti, Erik Marchi, and Alistair Conkie. Generating multilingual voices using speaker space translation based on bilingual speaker data. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7624–7628. IEEE, 2020.

[226] Preranasri Mali. A survey on text to speech translation of multi language. *International Journal of Research In Advanced Engineering Technologies ISSN*, pages 2347–2812, 2014.

[227] Guljamal Mamateli, Askar Rozi, Gulnar Ali, and Askar Hamdulla. Morphological analysis based part-of-speech tagging for uyghur speech synthesis. In *Knowledge Engineering and Management*, pages 389–396. Springer, 2011.

[228] Christopher Manning and Hinrich Schutze. *Foundations of statistical natural language processing*. MIT press, 1999.

[229] Courtney Mansfield, Ming Sun, Yuzong Liu, Ankur Gandhe, and Björn Hoffmeister. Neural text normalization with subword units. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 190–196, 2019.

[230] Xinnian Mao, Yuan Dong, Jinyu Han, Dezhi Huang, and Haila Wang. Inequality maximum entropy classifier with character features for polyphone disambiguation in mandarin tts systems. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–705. IEEE, 2007.

[231] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.

[232] Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*, volume 2017, pages 498–502, 2017.

[233] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio. Samplernn: An unconditional end-to-end neural audio generation model. In *ICLR*, 2017.

[234] Chenfeng Miao, Shuang Liang, Minchuan Chen, Jun Ma, Shaojun Wang, and Jing Xiao. Flow-tts: A non-autoregressive network for text to speech based on flow. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7209–7213. IEEE, 2020.

[235] Chenfeng Miao, Shuang Liang, Zhencheng Liu, Minchuan Chen, Jun Ma, Shaojun Wang, and Jing Xiao. Efficienttts: An efficient and high-quality text-to-speech architecture. *arXiv preprint arXiv:2012.03500*, 2020.

[236] Dongchan Min, Dong Bok Lee, Eunho Yang, and Sung Ju Hwang. Meta-stylespeech: Multi-speaker adaptive text-to-speech generation. *arXiv preprint arXiv:2106.03153*, 2021.

[237] Devang S Ram Mohan, Raphael Lenain, Lorenzo Foglianti, Tian Huey Teh, Marlene Staib, Alexandra Torresquintero, and Jiameng Gao. Incremental text to speech for neural sequence-to-sequence models using reinforcement learning. *Proc. Interspeech 2020*, pages 3186–3190, 2020.

[238] Masanori Morise, Fumiya Yokomori, and Kenji Ozawa. World: a vocoder-based high-quality speech synthesis system for real-time applications. *IEICE TRANSACTIONS on Information and Systems*, 99(7):1877–1884, 2016.

[239] Max Morrison, Zeyu Jin, Justin Salamon, Nicholas J Bryan, and Gautham J Mysore. Controllable neural prosody synthesis. *Proc. Interspeech 2020*, pages 4437–4441, 2020.

[240] Henry B Moss, Vatsal Aggarwal, Nishant Prateek, Javier González, and Roberto Barra-Chicote. Boffin tts: Few-shot speaker adaptation by bayesian optimization. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7639–7643. IEEE, 2020.

[241] Eric Moulines and Francis Charpentier. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech communication*, 9(5-6):453–467, 1990.

[242] Zhaoxi Mu, Xinyu Yang, and Yizhuo Dong. Review of end-to-end speech synthesis technology based on deep learning. *arXiv preprint arXiv:2104.09995*, 2021.

[243] Saida Mussakhojayeva, Aigerim Janaliyeva, Almas Mirzakhmetov, Yerbolat Khassanov, and Huseyin Atakan Varol. Kazakhtts: An open-source kazakh text-to-speech synthesis dataset. *arXiv preprint arXiv:2104.08459*, 2021.

[244] Eliya Nachmani, Adam Polyak, Yaniv Taigman, and Lior Wolf. Fitting new speakers based on a short untranscribed sample. In *International Conference on Machine Learning*, pages 3683–3691. PMLR, 2018.

[245] Paarth Neekhara, Chris Donahue, Miller Puckette, Shlomo Dubnov, and Julian McAuley. Expediting tts synthesis with adversarial vocoding. *Proc. Interspeech 2019*, pages 186–190, 2019.

[246] Paarth Neekhara, Shehzeen Hussain, Shlomo Dubnov, Farinaz Koushanfar, and Julian McAuley. Expressive neural voice cloning. *arXiv preprint arXiv:2102.00151*, 2021.

[247] Tomáš Nekvinda and Ondřej Dušek. One model, many languages: Meta-learning for multilingual text-to-speech. *Proc. Interspeech 2020*, pages 2972–2976, 2020.

[248] Yishuang Ning, Sheng He, Zhiyong Wu, Chunxiao Xing, and Liang-Jie Zhang. A review of deep learning based speech synthesis. *Applied Sciences*, 9(19):4050, 2019.

[249] Nicolas Obin, Julie Beliao, Christophe Veaux, and Anne Lacheret. Slam: Automatic stylization and labelling of speech melody. In *Speech Prosody*, page 246, 2014.

[250] Takuma Okamoto, Kentaro Tachibana, Tomoki Toda, Yoshinori Shiga, and Hisashi Kawai. An investigation of subband wavenet vocoder covering entire audible frequency range with limited acoustic features. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5654–5658. IEEE, 2018.

[251] Takuma Okamoto, Tomoki Toda, Yoshinori Shiga, and Hisashi Kawai. Improving fftnet vocoder with noise shaping and subband approaches. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 304–311. IEEE, 2018.

[252] Takuma Okamoto, Tomoki Toda, Yoshinori Shiga, and Hisashi Kawai. Tacotron-based acoustic model using phoneme alignment for practical neural text-to-speech systems. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 214–221. IEEE, 2019.

[253] Joseph Olive. Rule synthesis of speech from dyadic units. In *ICASSP'77. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 568–570. IEEE, 1977.

[254] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[255] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, et al. Parallel wavenet: Fast high-fidelity speech synthesis. In *International conference on machine learning*, pages 3918–3926. PMLR, 2018.

[256] Tom Le Paine, Pooya Khorrami, Shiyu Chang, Yang Zhang, Prajit Ramachandran, Mark A Hasegawa-Johnson, and Thomas S Huang. Fast wavenet generation algorithm. *arXiv preprint arXiv:1611.09482*, 2016.

[257] Huashan Pan, Xiulin Li, and Zhiqiang Huang. A mandarin prosodic boundary prediction model based on multi-task learning. In *INTERSPEECH*, pages 4485–4488, 2019.

[258] Junjie Pan, Xiang Yin, Zhiling Zhang, Shichao Liu, Yang Zhang, Zejun Ma, and Yuxuan Wang. A unified sequence-to-sequence front-end model for mandarin text-to-speech synthesis. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6689–6693. IEEE, 2020.

[259] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[260] Soumya Priyadarsini Panda, Ajit Kumar Nayak, and Satyananda Champati Rai. A survey on speech synthesis techniques in indian languages. *Multimedia Systems*, 26:453–478, 2020.

[261] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2335–2344, 2017.

[262] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*, 2019.

[263] Kyubyong Park and Seanie Lee. g2pm: A neural grapheme-to-phoneme conversion package for mandarin chinese based on a new open benchmark dataset. *Proc. Interspeech 2020*, pages 1723–1727, 2020.

[264] Kyubyong Park and Thomas Mulc. Css10: A collection of single speaker speech datasets for 10 languages. *Proc. Interspeech 2019*, pages 1566–1570, 2019.

[265] Dipjyoti Paul, Yannis Pantazis, and Yannis Stylianou. Speaker conditional wavernn: Towards universal neural vocoder for unseen speaker and recording conditions. *Proc. Interspeech 2020*, pages 235–239, 2020.

[266] Dipjyoti Paul, Muhammed PV Shifas, Yannis Pantazis, and Yannis Stylianou. Enhancing speech intelligibility in text-to-speech synthesis using speaking style conversion. *Proc. Interspeech 2020*, pages 1361–1365, 2020.

[267] Wenzhe Pei, Tao Ge, and Baobao Chang. Max-margin tensor neural network for chinese word segmentation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 293–303, 2014.

[268] Kainan Peng, Wei Ping, Zhao Song, and Kexin Zhao. Non-autoregressive neural text-to-speech. In *International Conference on Machine Learning*, pages 7586–7598. PMLR, 2020.

[269] Wei Ping, Kainan Peng, and Jitong Chen. Clarinet: Parallel wave generation in end-to-end text-to-speech. In *International Conference on Learning Representations*, 2018.

[270] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. Deep voice 3: 2000-speaker neural text-to-speech. *Proc. ICLR*, pages 214–217, 2018.

[271] Wei Ping, Kainan Peng, Kexin Zhao, and Zhao Song. Waveflow: A compact flow-based model for raw audio. In *International Conference on Machine Learning*, pages 7706–7716. PMLR, 2020.

[272] Emmanouil Antonios Platanios, Mrinmaya Sachan, Graham Neubig, and Tom Mitchell. Contextual parameter generation for universal neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 425–435, 2018.

[273] Adam Polyak, Yossi Adi, Jade Copet, Eugene Kharitonov, Kushal Lakhotia, Wei-Ning Hsu, Abdelrahman Mohamed, and Emmanuel Dupoux. Speech resynthesis from discrete disentangled self-supervised representations. *arXiv preprint arXiv:2104.00355*, 2021.

[274] Vadim Popov, Stanislav Kamenev, Mikhail Kudinov, Sergey Repyevsky, Tasnima Sadekova, Vladimir Kryzhanovskiy Bushaev, and Denis Parkhomenko. Fast and lightweight on-device tts with tacotron2 and lpcnet. *Proc. Interspeech 2020*, pages 220–224, 2020.

[275] Vadim Popov, Mikhail Kudinov, and Tasnima Sadekova. Gaussian lpcnet for multisample speech synthesis. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6204–6208. IEEE, 2020.

[276] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. *arXiv preprint arXiv:2105.06337*, 2021.

[277] KR Prajwal and CV Jawahar. Data-efficient training strategies for neural tts systems. In *8th ACM IKDD CODS and 26th COMAD*, pages 223–227. 2021.

[278] Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. Mls: A large-scale multilingual dataset for speech research. *Proc. Interspeech 2020*, pages 2757–2761, 2020.

[279] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.

[280] Pascal Puchtler, Johannes Wirth, and René Peinl. Hui-audio-corpus-german: A high quality tts dataset. *arXiv preprint arXiv:2106.06309*, 2021.

[281] Kaizhi Qian, Yang Zhang, Shiyu Chang, Mark Hasegawa-Johnson, and David Cox. Unsupervised speech decomposition via triple information bottleneck. In *International Conference on Machine Learning*, pages 7836–7846. PMLR, 2020.

[282] Yao Qian and Frank K Soong. Tts tutorial at iscslp 2014. `https://www.superlectures.com/iscslp2014/tutorial-4-deep-learning-for-speech-generation-and-synthesis`, 2014.

[283] Yao Qian, Zhizheng Wu, Xuezhe Ma, and Frank Soong. Automatic prosody prediction and detection with conditional random field (crf) models. In *2010 7th International Symposium on Chinese Spoken Language Processing*, pages 135–138. IEEE, 2010.

[284] Yao Qian, Yuchen Fan, Wenping Hu, and Frank K Soong. On the training aspects of deep neural network (dnn) for parametric tts synthesis. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3829–3833. IEEE, 2014.

[285] Tao Qin. *Dual Learning*. Springer, 2020.

[286] Lawrence Rabiner and Biinghwang Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

[287] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[288] Colin Raffel, Minh-Thang Luong, Peter J Liu, Ron J Weiss, and Douglas Eck. Online and linear-time attention by enforcing monotonic alignments. In *International Conference on Machine Learning*, pages 2837–2846. PMLR, 2017.

[289] Kanishka Rao, Fuchun Peng, Haşim Sak, and Françoise Beaufays. Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4225–4229. IEEE, 2015.

[290] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech: Fast, robust and controllable text to speech. In *NeurIPS*, 2019.

[291] Yi Ren, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Almost unsupervised text to speech and automatic speech recognition. In *International Conference on Machine Learning*, pages 5410–5419. PMLR, 2019.

[292] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech 2: Fast and high-quality end-to-end text to speech. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=piLPYqxtWuA`.

[293] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538. PMLR, 2015.

[294] Andrew Rosenberg. Autobi-a tool for automatic tobi annotation. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[295] Anthony Rousseau, Paul Deléglise, and Yannick Esteve. Ted-lium: an automatic speech recognition dedicated corpus. In *LREC*, pages 125–129, 2012.

[296] Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 2002.

[297] Yoshinori Sagisaka, Nobuyoshi Kaiki, Naoto Iwahashi, and Katsuhiko Mimura. Atr μ-talk speech synthesis system. In *Second International Conference on Spoken Language Processing*, 1992.

[298] Georg Isaac Schlünz. *The effects of part–of–speech tagging on text–to–speech synthesis for resource–scarce languages*. PhD thesis, North-West University, 2010.

[299] P Seeviour, J Holmes, and M Judd. Automatic generation of control signals for a parallel formant speech synthesizer. In *ICASSP'76. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 690–693. IEEE, 1976.

[300] Christine H Shadle and Robert I Damper. Prospects for articulatory synthesis: A position paper. In *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*, 2001.

[301] Changhao Shan, Lei Xie, and Kaisheng Yao. A bi-directional lstm approach for polyphone disambiguation in mandarin chinese. In *2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 1–5. IEEE, 2016.

[302] Slava Shechtman, Raul Fernandez, and David Haws. Supervised and unsupervised approaches for controlling narrow lexical focus in sequence-to-sequence speech synthesis. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 431–437. IEEE, 2021.

[303] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[304] Jonathan Shen, Ye Jia, Mike Chrzanowski, Yu Zhang, Isaac Elias, Heiga Zen, and Yonghui Wu. Non-attentive tacotron: Robust and controllable neural tts synthesis including unsupervised duration modeling. *arXiv preprint arXiv:2010.04301*, 2020.

[305] Yao Shi, Hui Bu, Xin Xu, Shaoji Zhang, and Ming Li. Aishell-3: A multi-speaker mandarin tts corpus and the baselines. *arXiv preprint arXiv:2010.11567*, 2020.

[306] Desai Siddhi, Jashin M Verghese, and Desai Bhavik. Survey on various methods of text to speech synthesis. *International Journal of Computer Applications*, 165(6):26–30, 2017.

[307] Kim Silverman, Mary Beckman, John Pitrelli, Mori Ostendorf, Colin Wightman, Patti Price, Janet Pierrehumbert, and Julia Hirschberg. Tobi: A standard for labeling english prosody. In *Second international conference on spoken language processing*, 1992.

[308] Berrak Sisman, Junichi Yamagishi, Simon King, and Haizhou Li. An overview of voice conversion and its challenges: From statistical modeling to deep learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.

[309] RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif A Saurous. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. In *international conference on machine learning*, pages 4693–4702. PMLR, 2018.

[310] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[311] Eunwoo Song, Min-Jae Hwang, Ryuichi Yamamoto, Jin-Seob Kim, Ohsung Kwon, and Jae-Min Kim. Neural text-to-speech with a modeling-by-generation excitation vocoder. *Proc. Interspeech 2020*, pages 3570–3574, 2020.

[312] Eunwoo Song, Ryuichi Yamamoto, Min-Jae Hwang, Jin-Seob Kim, Ohsung Kwon, and Jae-Min Kim. Improved parallel wavegan vocoder with perceptually weighted spectrogram loss. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 470–476. IEEE, 2021.

[313] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[314] Ryosuke Sonobe, Shinnosuke Takamichi, and Hiroshi Saruwatari. Jsut corpus: free large-scale japanese speech corpus for end-to-end speech synthesis. *arXiv preprint arXiv:1711.00354*, 2017.

[315] Jose Sotelo, Soroush Mehri, Kundan Kumar, Joao Felipe Santos, Kyle Kastner, Aaron Courville, and Yoshua Bengio. Char2wav: End-to-end speech synthesis. 2017.

[316] Richard Sproat and Navdeep Jaitly. Rnn approaches to text normalization: A challenge. *arXiv preprint arXiv:1611.00068*, 2016.

[317] Richard Sproat, Alan W Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. Normalization of non-standard words. *Computer speech & language*, 15(3):287–333, 2001.

[318] Vivek Kumar Rangarajan Sridhar, Srinivas Bangalore, and Shrikanth Narayanan. Exploiting acoustic and syntactic features for prosody labeling in a maximum entropy framework. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 1–8, 2007.

[319] Marlene Staib, Tian Huey Teh, Alexandra Torresquintero, Devang S Ram Mohan, Lorenzo Foglianti, Raphael Lenain, and Jiameng Gao. Phonological features for 0-shot multilingual speech synthesis. *Proc. Interspeech 2020*, pages 2942–2946, 2020.

[320] William D Stanley, Gary R Dougherty, Ray Dougherty, and H Saunders. Digital signal processing. 1988.

[321] Daisy Stanton, Yuxuan Wang, and RJ Skerry-Ryan. Predicting expressive speaking style from text in end-to-end speech synthesis. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 595–602. IEEE, 2018.

[322] Brooke Stephenson, Laurent Besacier, Laurent Girin, and Thomas Hueber. What the future brings: Investigating the impact of lookahead for incremental neural tts. *Proc. Interspeech 2020*, pages 215–219, 2020.

[323] Brooke Stephenson, Thomas Hueber, Laurent Girin, and Laurent Besacier. Alternate endings: Improving prosody for incremental neural tts with predicted future text input. *arXiv preprint arXiv:2102.09914*, 2021.

[324] Guangzhi Sun, Yu Zhang, Ron J Weiss, Yuan Cao, Heiga Zen, Andrew Rosenberg, Bhuvana Ramabhadran, and Yonghui Wu. Generating diverse and natural text-to-speech samples using a quantized fine-grained vae and autoregressive prosody prior. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6699–6703. IEEE, 2020.

[325] Guangzhi Sun, Yu Zhang, Ron J Weiss, Yuan Cao, Heiga Zen, and Yonghui Wu. Fully-hierarchical fine-grained prosody modeling for interpretable speech synthesis. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6264–6268. IEEE, 2020.

[326] Hao Sun, Xu Tan, Jun-Wei Gan, Hongzhi Liu, Sheng Zhao, Tao Qin, and Tie-Yan Liu. Token-level ensemble distillation for grapheme-to-phoneme conversion. In *INTERSPEECH*, 2019.

[327] Hao Sun, Xu Tan, Jun-Wei Gan, Sheng Zhao, Dongxu Han, Hongzhi Liu, Tao Qin, and Tie-Yan Liu. Knowledge distillation from bert in pre-training and fine-tuning for polyphone disambiguation. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 168–175. IEEE, 2019.

[328] Jingwei Sun, Jing Yang, Jianping Zhang, and Yonghong Yan. Chinese prosody structure prediction based on conditional random fields. In *2009 Fifth International Conference on Natural Computation*, volume 3, pages 602–606. IEEE, 2009.

[329] Ming Sun and Jerome R Bellegarda. Improved pos tagging for text-to-speech synthesis. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5384–5387. IEEE, 2011.

[330] Antti Suni, Juraj Šimko, Daniel Aalto, and Martti Vainio. Hierarchical representation and estimation of prosody using continuous wavelet transform. *Computer Speech & Language*, 45: 123–136, 2017.

[331] Youcef Tabet and Mohamed Boughazi. Speech synthesis techniques. a survey. In *International Workshop on Systems, Signal Processing and their Applications, WOSSPA*, pages 67–70. IEEE, 2011.

[332] Hideyuki Tachibana, Katsuya Uenoyama, and Shunsuke Aihara. Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4784–4788. IEEE, 2018.

[333] Yaniv Taigman, Lior Wolf, Adam Polyak, and Eliya Nachmani. Voiceloop: Voice fitting and synthesis via a phonological loop. In *International Conference on Learning Representations*, 2018.

[334] Shinnosuke Takamichi, Tomoki Toda, Alan W Black, Graham Neubig, Sakriani Sakti, and Satoshi Nakamura. Postfilters to modify the modulation spectrum for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24 (4):755–767, 2016.

[335] Aarne Talman, Antti Suni, Hande Celikkanat, Sofoklis Kakouros, Jörg Tiedemann, Martti Vainio, et al. Predicting prosodic prominence from text with pre-trained contextualized word representations. In *22nd Nordic Conference on Computational Linguistics (NoDaLiDa) Proceedings of the Conference*. Linköping University Electronic Press, 2019.

[336] Akira Tamamori, Tomoki Hayashi, Kazuhiro Kobayashi, Kazuya Takeda, and Tomoki Toda. Speaker-dependent wavenet vocoder. In *Interspeech*, volume 2017, pages 1118–1122, 2017.

[337] Daxin Tan, Hingpang Huang, Guangyan Zhang, and Tan Lee. Cuhk-ee voice cloning system for icassp 2021 m2voc challenge. *arXiv preprint arXiv:2103.04699*, 2021.

[338] Xu Tan. Microsoft research webinar: Pushing the frontier of neural text to speech. `https://www.youtube.com/watch?v=MA8PCvmr8B0`, 2021.

[339] Xu Tan. Tts tutorial at iscslp 2021. `https://www.microsoft.com/en-us/research/uploads/prod/2021/02/ISCSLP2021-TTS-Tutorial.pdf`, 2021.

[340] Xu Tan and Tao Qin. Tts tutorial at ijcai 2021. `https://ijcai-21.org/tutorials/`, 2021.

[341] Xu Tan, Jiale Chen, Di He, Yingce Xia, QIN Tao, and Tie-Yan Liu. Multilingual neural machine translation with language clustering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 962–972, 2019.

[342] Xu Tan, Yichong Leng, Jiale Chen, Yi Ren, Tao Qin, and Tie-Yan Liu. A study of multilingual neural machine translation. *arXiv preprint arXiv:1912.11625*, 2019.

[343] Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu. Multilingual neural machine translation with knowledge distillation. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=S1gUsoR9YX`.

[344] Xu Tan, Yingce Xia, Lijun Wu, and Tao Qin. Efficient bidirectional neural machine translation. *arXiv preprint arXiv:1908.09329*, 2019.

[345] Paul Taylor. The tilt intonation model. In *Fifth International Conference on Spoken Language Processing*, 1998.

[346] Paul Taylor. *Text-to-speech synthesis*. Cambridge university press, 2009.

[347] Qiao Tian, Zewang Zhang, Chao Liu, Heng Lu, Linghui Chen, Bin Wei, Pujiang He, and Shan Liu. Feathertts: Robust and efficient attention based neural tts. *arXiv preprint arXiv:2011.00935*, 2020.

[348] Qiao Tian, Zewang Zhang, Heng Lu, Ling-Hui Chen, and Shan Liu. Featherwave: An efficient high-fidelity neural vocoder with multi-band linear prediction. *Proc. Interspeech 2020*, pages 195–199, 2020.

[349] Noé Tits, Kevin El Haddad, and Thierry Dutoit. Analysis and assessment of controllability of an expressive deep learning-based tts system. *arXiv preprint arXiv:2103.04097*, 2021.

[350] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. Listening while speaking: Speech chain by deep learning. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 301–308. IEEE, 2017.

[351] Andros Tjandra, Sakriani Sakti, and Satoshi Nakamura. Machine speech chain with one-shot speaker adaptation. *Proc. Interspeech 2018*, pages 887–891, 2018.

[352] Andros Tjandra, Berrak Sisman, Mingyang Zhang, Sakriani Sakti, Haizhou Li, and Satoshi Nakamura. Vqvae unsupervised unit discovery and multi-scale code2spec inverter for zerospeech challenge 2019. *Proc. Interspeech 2019*, pages 1118–1122, 2019.

[353] Tomoki Toda and Keiichi Tokuda. A speech parameter generation algorithm considering global variance for hmm-based speech synthesis. *IEICE TRANSACTIONS on Information and Systems*, 90(5):816–824, 2007.

[354] Keiichi Tokuda. Statistical approach to speech synthesis: Past, present and future. In *INTERSPEECH*, 2019.

[355] Keiichi Tokuda, Takao Kobayashi, Takashi Masuko, and Satoshi Imai. Mel-generalized cepstral analysis-a unified approach to speech spectral estimation. In *Third International Conference on Spoken Language Processing*, 1994.

[356] Keiichi Tokuda, Takayoshi Yoshimura, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Speech parameter generation algorithms for hmm-based speech synthesis. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 3, pages 1315–1318. IEEE, 2000.

[357] Keiichi Tokuda, Yoshihiko Nankaku, Tomoki Toda, Heiga Zen, Junichi Yamagishi, and Keiichiro Oura. Speech synthesis based on hidden markov models. *Proceedings of the IEEE*, 101(5):1234–1252, 2013.

[358] Tao Tu, Yuan-Jui Chen, Alexander H Liu, and Hung-yi Lee. Semi-supervised learning for multi-speaker text-to-speech synthesis using discrete speech representation. *Proc. Interspeech 2020*, pages 3191–3195, 2020.

[359] Se-Yun Um, Sangshin Oh, Kyungguen Byun, Inseon Jang, ChungHyun Ahn, and Hong-Goo Kang. Emotional speech synthesis with rich and granularized control. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7254–7258. IEEE, 2020.

[360] Mohammed Usman, Mohammed Zubair, Mohammad Shiblee, Paul Rodrigues, and Syed Jaffar. Probabilistic modeling of speech in spectral domain using maximum likelihood estimation. *Symmetry*, 10(12):750, 2018.

[361] Jan Vainer and Ondřej Dušek. Speedyspeech: Efficient neural speech synthesis. *Proc. Interspeech 2020*, pages 3575–3579, 2020.

[362] Cassia Valentini-Botinhao and Junichi Yamagishi. Speech enhancement of noisy and reverberant speech for text-to-speech. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(8):1420–1433, 2018.

[363] Jean-Marc Valin and Jan Skoglund. Lpcnet: Improving neural speech synthesis through linear prediction. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5891–5895. IEEE, 2019.

[364] Jean-Marc Valin and Jan Skoglund. A real-time wideband neural vocoder at 1.6 kb/s using lpcnet. *Proc. Interspeech 2019*, pages 3406–3410, 2019.

[365] Rafael Valle, Jason Li, Ryan Prenger, and Bryan Catanzaro. Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6189–6193. IEEE, 2020.

[366] Rafael Valle, Kevin Shih, Ryan Prenger, and Bryan Catanzaro. Flowtron: an autoregressive flow-based generative network for text-to-speech synthesis. *arXiv preprint arXiv:2005.05957*, 2020.

[367] Sean Vasquez and Mike Lewis. Melnet: A generative model for audio in the frequency domain. *arXiv preprint arXiv:1906.01083*, 2019.

[368] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[369] Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al. Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. 2016.

[370] Ravichander Vipperla, Sangjun Park, Kihyun Choo, Samin Ishtiaq, Kyoungbo Min, Sourav Bhattacharya, Abhinav Mehrotra, Alberto Gil CP Ramos, and Nicholas D Lane. Bunched lpcnet: Vocoder for low-cost neural text-to-speech systems. *Proc. Interspeech 2020*, pages 3565–3569, 2020.

[371] Michael Wagner and Duane G Watson. Experimental and theoretical advances in prosody: A review. *Language and cognitive processes*, 25(7-9):905–945, 2010.

[372] Congyi Wang, Yu Chen, Bin Wang, and Yi Shi. Improve gan-based neural vocoder using pointwise relativistic leastsquare gan. *arXiv preprint arXiv:2103.14245*, 2021.

[373] Disong Wang, Liqun Deng, Yang Zhang, Nianzu Zheng, Yu Ting Yeung, Xiao Chen, Xunying Liu, and Helen Meng. Fcl-taco2: Towards fast, controllable and lightweight text-to-speech synthesis.

[374] Peilu Wang, Yao Qian, Frank K Soong, Lei He, and Hai Zhao. Word embedding for recurrent neural network based tts synthesis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883. IEEE, 2015.

[375] Wenfu Wang, Shuang Xu, and Bo Xu. First step towards end-to-end parametric tts synthesis: Generating spectral parameters with neural attention. In *Interspeech*, pages 2243–2247, 2016.

[376] Xi Wang, Huaiping Ming, Lei He, and Frank K Soong. s-transformer: Segment-transformer for robust neural speech synthesis. *arXiv preprint arXiv:2011.08480*, 2020.

[377] Xin Wang and Junichi Yamagishi. Neural harmonic-plus-noise waveform model with trainable maximum voice frequency for text-to-speech synthesis. In *Proc. 10th ISCA Speech Synthesis Workshop*, pages 1–6.

[378] Xin Wang and Yusuke Yasuda. Tts tutorial at ieice sp workshop. `https://www.slideshare.net/jyamagis/tutorial-on-endtoend-texttospeech-synthesis-part-1-neural-waveform-modeling`, 2019.

[379] Xin Wang, Jaime Lorenzo-Trueba, Shinji Takaki, Lauri Juvela, and Junichi Yamagishi. A comparison of recent waveform generation and acoustic modeling methods for neural-network-based speech synthesis. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4804–4808. IEEE, 2018.

[380] Xin Wang, Shinji Takaki, and Junichi Yamagishi. Neural source-filter waveform models for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:402–415, 2019.

[381] Xin Wang, Shinji Takaki, and Junichi Yamagishi. Neural source-filter-based waveform model for statistical parametric speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5916–5920. IEEE, 2019.

[382] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. Tacotron: Towards end-to-end speech synthesis. *Proc. Interspeech 2017*, pages 4006–4010, 2017.

[383] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ-Skerry Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A Saurous. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *International Conference on Machine Learning*, pages 5180–5189. PMLR, 2018.

[384] Daniel Watson, Jonathan Ho, Mohammad Norouzi, and William Chan. Learning to efficiently sample from diffusion probabilistic models. *arXiv preprint arXiv:2106.03802*, 2021.

[385] Ron J Weiss, RJ Skerry-Ryan, Eric Battenberg, Soroosh Mariooryad, and Diederik P Kingma. Wave-tacotron: Spectrogram-free end-to-end text-to-speech synthesis. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021.

[386] Matt Whitehill, Shuang Ma, Daniel McDuff, and Yale Song. Multi-reference neural tts stylization with adversarial cycle consistency. *Proc. Interspeech 2020*, pages 4442–4446, 2020.

[387] Colin W Wightman and David T Talkin. The aligner: Text-to-speech alignment using markov models. In *Progress in speech synthesis*, pages 313–323. Springer, 1997.

[388] Wikipedia. Speech synthesis — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Speech%20synthesis&oldid=1020857981`, 2021.

[389] Jan Wind. The evolutionary history of the human speech organs. *Studies in language origins*, 1:173–197, 1989.

[390] Lijun Wu, Xu Tan, Di He, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. Beyond error propagation in neural machine translation: Characteristics of language also matter. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3602–3611, 2018.

[391] Yi-Chiao Wu, Tomoki Hayashi, Takuma Okamoto, Hisashi Kawai, and Tomoki Toda. Quasi-periodic parallel wavegan vocoder: A non-autoregressive pitch-dependent dilated convolution model for parametric speech generation. *Proc. Interspeech 2020*, pages 3535–3539, 2020.

[392] Zhizheng Wu, Pawel Swietojanski, Christophe Veaux, Steve Renals, and Simon King. A study of speaker adaptation for dnn-based speech synthesis. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[393] Yujia Xiao, Lei He, Huaiping Ming, and Frank K Soong. Improving prosody with linguistic and bert derived features in multi-speaker based mandarin chinese neural tts. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6704–6708. IEEE, 2020.

[394] Qicong Xie, Xiaohai Tian, Guanghou Liu, Kun Song, Lei Xie, Zhiyong Wu, Hai Li, Song Shi, Haizhou Li, Fen Hong, et al. The multi-speaker multi-style voice cloning challenge 2021. *arXiv preprint arXiv:2104.01818*, 2021.

[395] Guanghui Xu, Wei Song, Zhengchen Zhang, Chao Zhang, Xiaodong He, and Bowen Zhou. Improving prosody modelling with cross-utterance bert embeddings for end-to-end speech synthesis. *arXiv preprint arXiv:2011.05161*, 2020.

[396] Jin Xu, Xu Tan, Yi Ren, Tao Qin, Jian Li, Sheng Zhao, and Tie-Yan Liu. Lrspeech: Extremely low-resource speech synthesis and recognition. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2802–2812, 2020.

[397] Jin Xu, Xu Tan, Renqian Luo, Kaitao Song, Jian Li, Tao Qin, and Tie-Yan Liu. Nas-bert: Task-agnostic and adaptive-size bert compression with neural architecture search. In *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2021.

[398] Jun Xu, Guohong Fu, and Haizhou Li. Grapheme-to-phoneme conversion for chinese text-to-speech. In *Eighth International Conference on Spoken Language Processing*, 2004.

[399] Liumeng Xue, Shifeng Pan, Lei He, Lei Xie, and Frank K Soong. Cycle consistent network for end-to-end style transfer tts training. *Neural Networks*, 140:223–236, 2021.

[400] Nianwen Xue. Chinese word segmentation as character tagging. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 8, Number 1, February 2003: Special Issue on Word Formation and Chinese Language Processing*, pages 29–48, 2003.

[401] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. Probability density distillation with generative adversarial networks for high-quality parallel waveform generation. *Proc. Interspeech 2019*, pages 699–703, 2019.

[402] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203. IEEE, 2020.

[403] Yuzi Yan, Xu Tan, Bohan Li, Tao Qin, Sheng Zhao, Yuan Shen, and Tie-Yan Liu. Adaspeech 2: Adaptive text to speech with untranscribed data. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021.

[404] Yuzi Yan, Xu Tan, Bohan Li, Guangyan Zhang, Tao Qin, Sheng Zhao, Yuan Shen, Wei-Qiang Zhang, and Tie-Yan Liu. Adaspeech 3: Adaptive text to speech for spontaneous style. In *INTERSPEECH*, 2021.

[405] Tomoya Yanagita, Sakriani Sakti, and Satoshi Nakamura. Neural itts: Toward synthesizing speech in real-time with end-to-end neural text-to-speech framework. In *Proceedings of the 10th ISCA Speech Synthesis Workshop*, pages 183–188, 2019.

[406] Geng Yang, Shan Yang, Kai Liu, Peng Fang, Wei Chen, and Lei Xie. Multi-band melgan: Faster waveform generation for high-quality text-to-speech. *arXiv preprint arXiv:2005.05106*, 2020.

[407] Jingzhou Yang and Lei He. Towards universal text-to-speech. In *INTERSPEECH*, pages 3171–3175, 2020.

[408] Jinhyeok Yang, Junmo Lee, Youngik Kim, Hoon-Young Cho, and Injung Kim. Vocgan: A high-fidelity real-time vocoder with a hierarchically-nested adversarial network. *Proc. Interspeech 2020*, pages 200–204, 2020.

[409] Shan Yang, Lei Xie, Xiao Chen, Xiaoyan Lou, Xuan Zhu, Dongyan Huang, and Haizhou Li. Statistical parametric speech synthesis using generative adversarial networks under a multi-task learning framework. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 685–691. IEEE, 2017.

[410] Kaisheng Yao and Geoffrey Zweig. Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[411] Yusuke Yasuda, Xin Wang, and Junichi Yamagishi. Initial investigation of an encoder-decoder end-to-end tts framework using marginalization of monotonic hard latent alignments. 2019.

[412] Zhiwei Ying and Xiaohua Shi. An rnn-based algorithm to detect prosodic phrase for chinese tts. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, volume 2, pages 809–812. IEEE, 2001.

[413] Sevinj Yolchuyeva, Géza Németh, and Bálint Gyires-Tóth. Text normalization with convolutional neural networks. *International Journal of Speech Technology*, 21(3):589–600, 2018.

[414] Reo Yoneyama, Yi-Chiao Wu, and Tomoki Toda. Unified source-filter gan: Unified source-filter network based on factorization of quasi-periodic parallel wavegan. *arXiv preprint arXiv:2104.04668*, 2021.

[415] Takayoshi Yoshimura. Simultaneous modeling of phonetic and prosodic parameters, and characteristic conversion for hmm-based text-to-speech systems. *PhD diss, Nagoya Institute of Technology*, 2002.

[416] Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis. In *Sixth European Conference on Speech Communication and Technology*, 1999.

[417] Jaeseong You, Dalhyun Kim, Gyuhyeon Nam, Geumbyeol Hwang, and Gyeongsu Chae. Gan vocoder: Multi-resolution discriminator is all you need. *arXiv preprint arXiv:2103.05236*, 2021.

[418] Chengzhu Yu, Heng Lu, Na Hu, Meng Yu, Chao Weng, Kun Xu, Peng Liu, Deyi Tuo, Shiyin Kang, Guangzhi Lei, et al. Durian: Duration informed attention network for speech synthesis. *Proc. Interspeech 2020*, pages 2027–2031, 2020.

[419] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[420] Fengpeng Yue, Yan Deng, Lei He, and Tom Ko. Exploring machine speech chain for domain adaptation and few-shot speaker adaptation. *arXiv preprint arXiv:2104.03815*, 2021.

[421] Rohola Zandie, Mohammad H. Mahoor, Julia Madse, and Eshrat S. Emamian. Ryanspeech: A corpus for conversational text-to-speech synthesis. *arXiv preprint arXiv:2106.08468*, 2021.

[422] Heiga Zen. Acoustic modeling in statistical parametric speech synthesis-from hmm to lstm-rnn. 2015.

[423] Heiga Zen. Generative model-based text-to-speech synthesis. `https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45882.pdf`, 2017.

[424] Heiga Zen and Haşim Sak. Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4470–4474. IEEE, 2015.

[425] Heiga Zen, Keiichi Tokuda, and Alan W Black. Statistical parametric speech synthesis. *speech communication*, 51(11):1039–1064, 2009.

[426] Heiga Zen, Andrew Senior, and Mike Schuster. Statistical parametric speech synthesis using deep neural networks. In *2013 ieee international conference on acoustics, speech and signal processing*, pages 7962–7966. IEEE, 2013.

[427] Heiga Zen, Yannis Agiomyrgiannakis, Niels Egberts, Fergus Henderson, and Przemysław Szczepaniak. Fast, compact, and high quality lstm-rnn based statistical parametric speech synthesizers for mobile devices. *Interspeech 2016*, pages 2273–2277, 2016.

[428] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. Libritts: A corpus derived from librispeech for text-to-speech. *Proc. Interspeech 2019*, pages 1526–1530, 2019.

[429] Zhen Zeng, Jianzong Wang, Ning Cheng, Tian Xia, and Jing Xiao. Aligntts: Efficient feed-forward text-to-speech system without explicit alignment. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6714–6718. IEEE, 2020.

[430] Zhen Zeng, Jianzong Wang, Ning Cheng, and Jing Xiao. Prosody learning mechanism for speech synthesis system without text length limit. *Proc. Interspeech 2020*, pages 4422–4426, 2020.

[431] Zhen Zeng, Jianzong Wang, Ning Cheng, and Jing Xiao. Lvcnet: Efficient condition-dependent modeling network for waveform generation. *arXiv preprint arXiv:2102.10815*, 2021.

[432] ZeroSpeech. Zero resource speech challenge. `https://www.zerospeech.com/`.

[433] Bohan Zhai, Tianren Gao, Flora Xue, Daniel Rothchild, Bichen Wu, Joseph E Gonzalez, and Kurt Keutzer. Squeezewave: Extremely lightweight vocoders for on-device speech synthesis. *arXiv preprint arXiv:2001.05685*, 2020.

[434] Chen Zhang, Yi Ren, Xu Tan, Jinglin Liu, Kejun Zhang, Tao Qin, Sheng Zhao, and Tie-Yan Liu. Denoispeech: Denoising text to speech with frame-level noise modeling. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021.

[435] Chen Zhang, Xu Tan, Yi Ren, Tao Qin, Kejun Zhang, and Tie-Yan Liu. Uwspeech: Speech to speech translation for unwritten languages. In *AAAI*, 2021.

[436] Haitong Zhang and Yue Lin. Unsupervised learning for sequence-to-sequence text-to-speech for low-resource languages. *Proc. Interspeech 2020*, pages 3161–3165, 2020.

[437] Hao Zhang, Richard Sproat, Axel H Ng, Felix Stahlberg, Xiaochang Peng, Kyle Gorman, and Brian Roark. Neural models of text normalization for speech applications. *Computational Linguistics*, 45(2):293–337, 2019.

[438] Jing-Xuan Zhang, Zhen-Hua Ling, and Li-Rong Dai. Forward attention in sequence-to-sequence acoustic modeling for speech synthesis. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4789–4793. IEEE, 2018.

[439] Junhui Zhang, Junjie Pan, Xiang Yin, Chen Li, Shichao Liu, Yang Zhang, Yuxuan Wang, and Zejun Ma. A hybrid text normalization system using multi-head self-attention for mandarin. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6694–6698. IEEE, 2020.

[440] Mingyang Zhang, Xin Wang, Fuming Fang, Haizhou Li, and Junichi Yamagishi. Joint training framework for text-to-speech and voice conversion using multi-source tacotron and wavenet. *Proc. Interspeech 2019*, pages 1298–1302, 2019.

[441] Shiliang Zhang, Ming Lei, Zhijie Yan, and Lirong Dai. Deep-fsmn for large vocabulary continuous speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5869–5873. IEEE, 2018.

[442] Weizhao Zhang, Hongwu Yang, Xiaolong Bu, and Lili Wang. Deep learning for mandarin-tibetan cross-lingual speech synthesis. *IEEE Access*, 7:167884–167894, 2019.

[443] Ya-Jie Zhang, Shifeng Pan, Lei He, and Zhen-Hua Ling. Learning latent representations for style control and transfer in end-to-end speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6945–6949. IEEE, 2019.

[444] Yang Zhang, Liqun Deng, and Yasheng Wang. Unified mandarin tts front-end based on distilled bert model. *arXiv preprint arXiv:2012.15404*, 2020.

[445] Yu Zhang, Ron J Weiss, Heiga Zen, Yonghui Wu, Zhifeng Chen, RJ Skerry-Ryan, Ye Jia, Andrew Rosenberg, and Bhuvana Ramabhadran. Learning to speak fluently in a foreign language: Multilingual speech synthesis and cross-language voice cloning. *Proc. Interspeech 2019*, pages 2080–2084, 2019.

[446] Zewang Zhang, Qiao Tian, Heng Lu, Ling-Hui Chen, and Shan Liu. Adadurian: Few-shot adaptation for neural text-to-speech with durian. *arXiv preprint arXiv:2005.05642*, 2020.

[447] Zhengchen Zhang, Fuxiang Wu, Chenyu Yang, Minghui Dong, and Fugen Zhou. Mandarin prosodic phrase prediction based on syntactic trees. In *SSW*, pages 160–165, 2016.

[448] Zi-Rong Zhang, Min Chu, and Eric Chang. An efficient way to learn rules for grapheme-to-phoneme conversion in chinese. In *International Symposium on Chinese Spoken Language Processing*, 2002.

[449] Shengkui Zhao, Trung Hieu Nguyen, Hao Wang, and Bin Ma. Towards natural bilingual and code-switched speech synthesis based on mix of monolingual recordings and cross-lingual voice conversion. *Proc. Interspeech 2020*, pages 2927–2931, 2020.

[450] Yi Zhao, Daisuke Saito, and Nobuaki Minematsu. Speaker representations for speaker adaptation in multiple speakers blstm-rnn-based speech synthesis. *space*, 5(6):7, 2016.

[451] Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. Deep learning for chinese word segmentation and pos tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 647–657, 2013.

[452] Yibin Zheng, Jianhua Tao, Zhengqi Wen, and Jiangyan Yi. Forward-backward decoding sequence for regularizing end-to-end tts. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(12):2067–2079, 2019.

[453] Xuehao Zhou, Xiaohai Tian, Grandee Lee, Rohan Kumar Das, and Haizhou Li. End-to-end code-switching tts with cross-lingual language model. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7614–7618. IEEE, 2020.

[454] Yixuan Zhou, Changhe Song, Jingbei Li, Zhiyong Wu, and Helen Meng. Dependency parsing based semantic representation learning with graph neural network for enhancing expressiveness of text-to-speech. *arXiv preprint arXiv:2104.06835*, 2021.

[455] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[456] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *Advances in neural information processing systems*, 2017.

[457] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.