Artificial Intelligence: Foundations, Theory, and Algorithms

Xu Tan

Neural Text-to-Speech Synthesis

Springer

Artificial Intelligence: Foundations, Theory, and Algorithms

Artificial Intelligence: Foundations, Theory and Algorithms fosters the dissemination of knowledge, technologies and methodologies that advance developments in artificial intelligence (AI) and its broad applications. It brings together the latest developments in all areas of this multidisciplinary topic, ranging from theories and algorithms to various important applications. The intended readership includes research students and researchers in computer science, computer engineering, electrical engineering, data science, and related areas seeking a convenient way to track the latest findings on the foundations, methodologies, and key applications of artificial intelligence.

This series provides a publication and communication platform for all AI topics, including but not limited to:

- Knowledge representation
- Automated reasoning and inference
- Reasoning under uncertainty
- Planning, scheduling, and problem solving
- Cognition and AI
- Search
- Diagnosis
- Constraint processing
- Multi-agent systems
- Game theory in AI
- Machine learning
- Deep learning
- Reinforcement learning
- Data mining
- Natural language processing
- Computer vision
- Human interfaces
- Intelligent robotics
- Explanation generation
- Ethics in AI
- Fairness, accountability, and transparency in AI

This series includes monographs, introductory and advanced textbooks, state-of-the-art collections, and handbooks. Furthermore, it supports Open Access publication mode.

Xu Tan

Neural Text-to-Speech Synthesis

Springer

Xu Tan
Microsoft Research Asia (China)
Beijing, China

ISSN 2365-3051 ISSN 2365-306X (electronic)
Artificial Intelligence: Foundations, Theory, and Algorithms
ISBN 978-981-99-0826-4 ISBN 978-981-99-0827-1 (eBook)
https://doi.org/10.1007/978-981-99-0827-1

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

I would like to dedicate this book to my family, especially my grandfather and grandmother.

Foreword by Dong Yu

Text-to-speech (TTS) synthesis is an artificial intelligence (AI) technique that renders a preferably naturally sounding speech given an arbitrary text. It is a key technological component in many important applications, including virtual assistants, AI-generated audiobooks, speech-to-speech translation, AI news reporters, audible driving guidance, and digital humans. In the past decade, we have observed significant progress made in TTS. These new developments are mainly attributed to deep learning techniques and are usually referred to as neural TTS. Many neural TTS systems have achieved human quality for the tasks they are designed for.

Although many TTS books have been published, this book is the first of its kind that provides a comprehensive introduction to neural TTS, including but not limited to the key components such as text analysis, acoustic model, and vocoder, the key milestone models such as Tacotron, DeepVoice, FastSpeech, and the more advanced techniques such as expressive and controllable TTS, robust TTS, and efficient TTS. Xu Tan, the author of this book, has contributed significantly to the recent advances in TTS. He has developed several impactful neural TTS systems such as FastSpeech 1/2, DelightfulTTS, and NaturalSpeech, the latter of which has achieved human parity on the TTS benchmark dataset. His knowledge of the domain and his first-hand experience with the topic allow him to organize the contents effectively and make them more accessible to readers, and to describe the key concepts, the basic methods, and the state-of-the-art techniques and their relationships in detail and clearly. I am very glad that he introduced and clarified many key concepts and background knowledge at the beginning of this book so that people with little or no knowledge of TTS can also read and understand the book effectively.

This is a very well-written book and certainly one that provides useful and thoughtful information to readers at various levels. I believe this book is a great reference book for all researchers, practitioners, and students who are interested in

quickly grasping the history, the state-of-the-art, and the future directions of speech synthesis or are interested in gaining insightful ideas on the development of TTS.

ACM/IEEE/ISCA Fellow Dong Yu
Seattle, WA, USA
October, 2022

Foreword by Heiga Zen

For more than 60 years, researchers have studied how to synthesize natural-sounding speech. The focus of speech synthesis research shifted from simulating the physical process of speech production (articulatory speech synthesis) to concatenating basic speech units (concatenative speech synthesis) in the late 1970s. The "unit-selection" method of concatenative synthesis, which was invented in the late 1980s, could synthesize highly natural and intelligible speech on some subset of the sentences. However, because of its "exemplar-based" nature, synthesizing expressive speech, which is essential to achieve human-to-human-level interactions, was still difficult.

In the mid-1990s, a promising new method for speech synthesis, called statistical parametric speech synthesis, was proposed. This method is based on generative models, where statistical models represent the conditional probability distribution of output speech given an input text. Although it offered high intelligibility and flexibility to synthesize a variety of speech, it had limited success until the 2000s due to its inferior naturalness. Thanks to the advancement of signal processing techniques and machine learning in the late 2000s, its quality improved significantly. Furthermore, the introduction of deep learning/neural networks as its generative models in the 2010s completely changed the landscape of text-to-speech research; synthetic speech sounds not only natural and intelligible but also expressive and beyond. Nowadays most of the components in a text-to-speech synthesis system are realized by neural networks. Such text-to-speech synthesis systems are often referred to as "neural text-to-speech".

This book, *Neural Text-to-Speech Synthesis*, is for people who want to understand how modern deep learning/neural network-based text-to-speech synthesis systems are implemented and how they have progressed from traditional concatenative and statistical parametric speech synthesis systems to recent integrated, neural end-to-end text-to-speech systems. Xu Tan, who is one of the leading researchers in neural text-to-speech synthesis, has put together a book based on the progress in neural text-to-speech synthesis over the past decade. The first three chapters of the book give the basics of spoken language processing and deep learning for text-to-speech synthesis. The next three chapters address the problem of converting an input text to a speech waveform in neural text-to-speech synthesis, including both a

cascaded approach consisting of three key modules (text analysis, acoustic models, and vocoders) and an integrated, end-to-end approach. The following four chapters provide a review of advanced topics, which are important to deploy neural text-to-speech synthesis systems to real-world applications, such as data and computational efficiency, controllability, expressiveness, and robustness. This is followed by a chapter describing the relationship with the three related speech synthesis areas. The last chapter gives the concluding remarks and possible future research directions.

This book will be a great help for readers to understand the landscape of neural text-to-speech synthesis and to explore new frontiers in text-to-speech synthesis research.

ISCA Fellow Heiga Zen
Tokyo, Japan
December, 2022

Foreword by Haizhou Li

Text-to-speech synthesis (TTS) enables machines to speak naturally and express human emotions. With the advent of deep learning, the quality of synthesized speech has improved by leaps and bounds.

This is a book like no other. It is the first to provide a comprehensive overview that covers the breadth and depth of neural TTS. It was written by my research friend, Xu Tan, the architect of the widely adopted FastSpeech TTS system. Xu Tan is known as a prominent researcher, a prolific author, and a hands-on engineer. I am fortunate to be the first to read this book, and I must say that this book has fulfilled my longstanding wish of having a more accessible guidebook for neural TTS.

This book provides a unique, historical, and technological perspective on the recent development of neural TTS, that I resonate with. It is very timely given the increasing interest in the research community. It is a book that research students, TTS beginners, and practitioners cannot miss.

IEEE/ISCA Fellow — Haizhou Li
Shenzhen, China
November, 2022

Preface

Speaking is one of the most important language capabilities (the others being listening, reading, and writing) of human beings. Text-to-speech synthesis (TTS for short), which aims to generate intelligible and natural speech from text, plays a key role to enable machines to speak and is an important task in artificial intelligence and natural language/speech processing.

With the development of deep learning and artificial intelligence, neural network-based TTS has significantly improved the quality of synthesized speech in recent years. Considering neural TTS involves multiple disciplines such as speech signal processing and deep learning, and there are abundant of literature in this area. It is very challenging for TTS practitioners especially beginners to understand the landscape of neural TTS. Thus, there is a growing demand for a book dedicated for neural text-to-speech synthesis.

I started my research on neural TTS several years ago. During this period, I was deeply impressed by the rapid progress of neural TTS brought by whole speech synthesis community. Then I started to plan and prepare for this book two years ago. However, it is not an easy thing considering the diverse methodologies and abundant literature in this area. Thus, I divide this difficult job into multiple stages: (1) first give a tutorial on neural TTS at ISCSLP 2021 conference; (2) then write a survey paper on neural speech synthesis based on this tutorial; (3) further enrich the previous tutorial and survey paper gradually and give tutorials at IJCAI 2021, ICASSP 2022, and INTERSPEECH 2022 conferences; (4) and finally write this book based on these survey and tutorials.[1]

This book gives a comprehensive introduction to neural TTS, aiming to provide a good understanding of its basic methods, current research, and future trends. I first introduce the background of speech synthesis and the history of TTS technologies. I then introduce some preliminary knowledge of neural TTS in the first part of this

[1] Please find all these survey and tutorials in this page: https://github.com/tts-tutorial/. Readers can use this page to check updates and initiate discussions on this book: https://github.com/tts-tutorial/book.

book, including the basics of spoken language processing and deep learning. In the second part, I introduce the key components of neural TTS, including text analyses, acoustic models, and vocoders. I further introduce several advanced topics of neural TTS in the third part, including expressive and controllable TTS, robust TTS, model-efficient TTS, data-efficient TTS, and some tasks beyond TTS. At last, I summarize this book and discuss future research directions. I also list some resources related to TTS (e.g., TTS tutorials and talks, open-source implementations, and datasets) in the appendix.

This book is written for researchers, industry practitioners, and graduate/undergraduate students in speech synthesis, speech/language processing, and artificial intelligence.

Beijing, China Xu Tan
October, 2022

Acknowledgements

This book would not have been possible without the contributions of many people.

I would like to thank my colleagues and interns at Microsoft, who have been working together with me on the topic of neural text-to-speech synthesis, including Tao Qin, Sheng Zhao, Tie-Yan Liu, Lei He, Frank Soong, Hsiao-Wuen Hon, Lidong Zhou, Jiang Bian, Qiang Huo, Jun-Wei Gan, Yanqing Liu, Bohan Li, Yi Ren, Renqian Luo, Rui Wang, Kaitao Song, Xi Wang, Gang Wang, Jinzhu Li, Yuanhao Yi, Ruiqing Xue, Runnan Li, Dongxu Han, Xianghao Tang, Yuchao Zhang, Peter Pan, Chen Zhang, Jie Ding, Yangjun Ruan, Chenxu Hu, Jin Xu, Mingjian Chen, Hao Sun, Yichong Leng, Kai Shen, Zeqian Ju, Sang-gil Lee, Zehua Chen, Haohe Liu, Jian Cong, Jiawei Chen, Yuzi Yan, Guangyan Zhang, Yihan Wu, Jian Luan, Peiling Lu, Junliang Guo, Chen Zhang, Qi Meng, and Chang Liu. I would also like to thank my external collaborators including Zhou Zhao, Ruihua Song, Jian Li, Kejun Zhang, Yuan Shen, Wei-Qiang Zhang, Tan Lee, Guihua Wen, Sungroh Yoon, and Danilo Mandic.

I would like to thank Heiga Zen, Haizhou Li, and Dong Yu for providing suggestions and forewords to this book. I want to make special thanks to Heiga Zen for giving me so much help to improve this book. I also want to thank those who gave me permission to directly use or reproduce images/figures from their publications.

At last, I want to thank the people in the whole speech synthesis community for pushing forward TTS technologies.

Contents

Acronyms

AF	Autoregressive Flow
AM	Acoustic Model
AR	Autoregressive
ASR	Automatic Speech Recognition
BAP	Band Aperiodicities
BFCC	Bark-Frequency Cepstral Coefficients
CMOS	Comparative Mean Opinion Score
CNN	Convolutional Neural Network
CTC	Connectionist Temporal Classification
DFT	Discrete Fourier Transform
Diffusion or DDPM	Denoising Diffusion Probabilistic Model
DNN	Dense Neural Network
DTFT	Discrete-Time Fourier Transform
DTW	Dynamic Time Warping
E2E	End-to-End
F0	Fundamental Frequency
FFT	Fast Fourier Transform
Flow	Normalizing Flow
G2P	Grapheme to Phoneme
GAN	Generative Adversarial Network
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model
IAF	Inverse Autoregressive Flow
IPA	International Phonetic Alphabet
LPC	Linear Predictive Coding
LSP	Line Spectral Pairs
LSTM	Long Short-Term Memory
MCC	Mel-Cepstral Coefficients
MCD	Mel-Cepstral Distortion
MFA	Montreal Forced Alignment
MFCC	Mel-Frequency Cepstral Coefficients

MGC	Mel-Generalized Coefficients
MOS	Mean Opinion Score
NAR	Non-Autoregressive
PCM	Pulse-Code Modulation
PESQ	Perceptual Evaluation of Speech Quality
POS	Part of Speech
PPG	Phonetic Posteriorgrams
RNN	Recurrent Neural Network
SDR	Signal-to-Distortion Ratio
SMOS	Similarity Mean Opinion Score
SNR	Signal-to-Noise Ratio
SPSS	Statistical Parametric Speech Synthesis
STFT	Short-Time Fourier Transform
STOI	Short-Time Objective Intelligibility
SVS	Singing Voice Synthesis
TN	Text Normalization
ToBI	Tones and Break Indices
TTS	Text-to-Speech Synthesis
VAE	Variational Auto-Encoder
VC	Voice Conversion
ZCR	Zero-Crossing Rate

About the Author

Xu Tan is a Principal Researcher and Research Manager at Microsoft Research Asia. His research interests cover machine learning and its applications in language/speech/music processing and digital human creation, such as text-to-speech synthesis, neural machine translation, music generation, and talking avatar generation.

He has rich research experience in text-to-speech synthesis. He has developed high-quality TTS systems such as FastSpeech 1/2 (widely used in the TTS community), DelightfulTTS (winning the champion of the Blizzard TTS Challenge), and NauturalSpeech (achieving human-level quality on the TTS benchmark dataset), and transferred many research works to improve the experience of Microsoft Azure TTS services. He has given a series of tutorials on TTS at top conferences such as IJCAI, ICASSP, and INTERSPEECH, and written a comprehensive survey paper on TTS.

Besides speech synthesis, he has designed several popular language models (e.g., MASS) and AI music systems (e.g., Muzic), developed machine translation systems that achieved human parity in Chinese-English translation, and won several champions in WMT machine translation competitions. He has published over 100 papers at prestigious conferences such as ICML, NeurIPS, ICLR, AAAI, IJCAI, ACL, EMNLP, NAACL, ICASSP, INTERSPEECH, KDD, and IEEE/ACM Transactions, and served as the area chair or action editor of some AI conferences and journals (e.g., NeurIPS, AAAI, ICASSP, TMLR).

Chapter 1
Introduction

Abstract In this chapter, we first discuss the motivation of this book and then introduce the history of text-to-speech (TTS) technologies. We further overview neural network-based TTS through different taxonomies, and finally, describe the organization of this book.

Keywords Text-to-speech (TTS) · Neural TTS · TTS history · TTS taxonomy

1.1 Motivation

Speaking is one of the most important language capabilities (the others being listening, reading, and writing) of human beings. Text-to-speech synthesis (TTS or speech synthesis[1] for short), which aims to generate intelligible and natural speech from text [1, 2], plays a key role to enable machines to speak [3], and is an important task in artificial intelligence and natural language/speech processing [4–6]. Developing a TTS system requires knowledge of languages and human speech production, and involves multiple disciplines including linguistics [7], acoustics [8], digital signal processing [9], and machine learning [10, 11].

Previous works usually leverage concatenative [12] or statistical parametric [13] based methods to build TTS systems, which suffer from poor generation quality (e.g., low intelligibility and naturalness, voice artifacts such as muffled, buzzing, noisy, or robotic voice). With the development of deep learning [14, 15], neural network-based TTS (neural TTS for short) has evolved quickly and improved the intelligibility and naturalness of generated speech a lot [16–23]. Neural TTS discards most of the prior knowledge needed in previous TTS systems and conducts end-to-end learning purely from data. Due to the powerful capabilities to learn data representations (representation learning) and model data distributions (generative

[1] Broadly speaking, speech synthesis covers the tasks to generate speech from any information source, such as from text (text-to-speech synthesis) or from another voice (voice conversion). Here we use speech synthesis to denote text-to-speech synthesis from a narrow sense.

X. Tan, *Neural Text-to-Speech Synthesis*, Artificial Intelligence: Foundations, Theory, and Algorithms, https://doi.org/10.1007/978-981-99-0827-1_1

modeling), neural TTS can achieve high voice quality that is as natural as human recordings [24].

A systematic and comprehensive introduction to neural TTS and a good understanding of the current research status and future research trends are very helpful and necessary for people working on TTS. While there is a lot of research work focusing on different aspects of neural TTS and there are also some survey papers on neural TTS [2, 25–31], a comprehensive book to introduce neural TTS is necessary, especially in the era of content creation and metaverse where there is a strong demand on text-to-speech synthesis.

This book originates from our previous survey paper and tutorials:

- *A Survey on Neural Speech Synthesis*, https://arxiv.org/abs/2106.15561.
- *TTS tutorial at ISCSLP 2021*, https://tts-tutorial.github.io/iscslp2021/.
- *TTS tutorial at IJCAI 2021*, https://github.com/tts-tutorial/ijcai2021.
- *TTS tutorial at ICASSP 2022*, https://github.com/tts-tutorial/icassp2022.
- *TTS tutorial at INTERSPEECH 2022*, https://github.com/tts-tutorial/interspeech2022.

Readers can use this GitHub page (https://github.com/tts-tutorial/book) to check updates and initiate discussions on this book.

In the following sections, we briefly review the history of TTS technologies, introduce some basic knowledge of neural TTS, and describe the taxonomy of neural TTS and the organization of this book.

1.2 History of TTS Technology

People have tried to build machines to synthesize human speech dating back to the twelfth century [32]. In the 2nd half of the eighteenth 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 [33]. The first speech synthesis system that was built upon computers came out in the latter half of the twentieth century [32]. The early computer-based speech synthesis methods include articulatory synthesis [34, 35], formant synthesis [36–39], and concatenative synthesis [12, 40–43]. Later, with the development of statistical machine learning, statistical parametric speech synthesis (SPSS) is proposed [13, 44–46], which predicts parameters such as spectrum, fundamental frequency, and duration for speech synthesis. Since the 2010s, neural network-based speech synthesis [16–18, 47–51] has gradually become the dominant method and achieved much better voice quality.

1.2.1 Articulatory Synthesis

Articulatory synthesis [34, 35] produces speech by simulating the behavior of human articulators 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.

1.2.2 Formant Synthesis

Formant synthesis [36–38] 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. 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 challenging to specify rules for synthesis.

1.2.3 Concatenative Synthesis

Concatenative synthesis [12, 40–43] relies on the concatenation of speech segments that are stored in a database. In inference, the concatenative TTS system searches speech segments to match the given input text and produces a speech waveform by concatenating these units together. There are mainly two types of concatenative speech synthesis: diphone synthesis [41] and unit selection synthesis [12]. Diphone synthesis leverages diphones that describe the transitions between phones, and stores a single example of each diphone in the database, while unit selection synthesis leverages speech units ranging from whole sentences to individual phones and stores multiple segments of each unit in the database.

Generally speaking, concatenative TTS can generate audio with high intelligibility and authentic timbre close to the original voice actor. However, concatenative TTS requires a vast 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.

1.2.4 Statistical Parametric Synthesis

To address the drawbacks of concatenative TTS, statistical parametric speech synthesis (SPSS) is proposed [13, 44–46, 52]. The basic idea is that instead of direct generating waveform through concatenation, we can first generate the acoustic parameters [53–55] that are necessary to produce speech and then recover speech from the generated acoustic parameters using some algorithms [56–59]. 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 [60], grapheme-to-phoneme conversion [61], word segmentation, etc. Then it extracts the linguistic features, such as phonemes 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 [53, 54], etc, and are extracted from the speech through vocoder analysis [56, 58, 59]. The vocoders synthesize speech from the predicted acoustic features. SPSS has several advantages over previous TTS systems: (1) flexibility, as it is convenient to modify the parameters to control the generated speech; (2) low data cost, as it requires fewer recordings than concatenative synthesis. However, SPSS also has its drawbacks: (1) the generated speech has lower audio fidelity 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 the 2010s, as neural networks and deep learning have achieved rapid progress, some works first introduce deep neural networks into SPSS, such as deep neural network (DNN) based [16, 47] and recurrent neural network (RNN) based [48, 49, 62]. However, these models replace HMM with neural networks and still predict the acoustic features from linguistic features, which follow the paradigm of SPSS. Later, [50] propose directly generating acoustic features from phoneme sequence instead of linguistic features, which is the first encoder-decoder-based TTS model with a sequence-to-sequence framework. In this book, we focus on neural-based speech synthesis and especially end-to-end models.[2] Since later SPSS also uses neural networks as the acoustic models, we briefly describe these models but do not dive deep into the details.

[2] The term "end-to-end" in TTS has a vague meaning. In early studies, "end-to-end" refers to the text-to-spectrogram model being end-to-end but still using 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 [17] does not use acoustic features but directly generates waveform from linguistic features, and Tacotron [18] does not use linguistic features but directly generates 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.

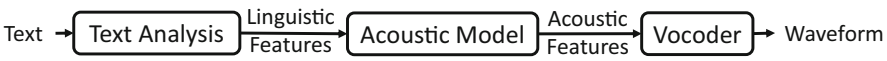


Fig. 1.1 The three key components in neural TTS

1.3 Overview of Neural TTS

1.3.1 TTS in the Era of Deep Learning

With 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 [17] 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 [63, 64] 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., Char2Wav [65], Tacotron 1/2 [18, 19], Deep Voice 3 [21], and FastSpeech 1/2 [23, 66]) 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 [67], FastSpeech 2s [66], EATS [68], and NaturalSpeech [24]. 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.

1.3.2 Key Components of TTS

A neural TTS system consists of three basic components:[3] a text analysis module, an acoustic model, and a vocoder. As shown in Fig. 1.1, 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 introduce the three components of neural TTS in Part II. Specifically, we first introduce the main taxonomy for the basic components of neural TTS and then introduce text analysis, acoustic models, and vocoders in

[3] Although some end-to-end models do not explicitly use text analysis (e.g., Tacotron 2 [19]), acoustic models (e.g., WaveNet [17]), or vocoders (e.g., Tacotron [18]), and some systems only use a single end-to-end model (e.g., FastSpeech 2s [66] and NaturalSpeech [24]), using these components are still popular in current TTS research and product.

Chaps. 4, 5, and 6 respectively. We further introduce fully end-to-end TTS in Chap. 7.

1.3.3 Advanced Topics in TTS

Besides the key components of neural TTS, we further categorize neural TTS according to some advanced topics and introduce the details in Part III: (1) To improve the naturalness and expressiveness, we introduce how to model, control, and transfer the style/prosody of speech in order to generate expressive speech (Chap. 8); (2) Since TTS models are facing robustness issues where word skipping and repeating problems in generated speech affect the speech quality, we introduce how to improve the robustness of speech synthesis (Chap. 9); (3) Since neural TTS is modeled as a sequence-to-sequence generation task that leverages deep neural networks as the model backbone and generates speech in an autoregressive way, it usually requires large inference time and high computation/memory cost. Thus, we introduce how to speed up the autoregressive generation and reduce the model and computation size (Chap. 10); (4) In low data resource scenarios where the data to train a TTS model is insufficient, the synthesized speech may be of both low intelligibility and naturalness. Therefore, we introduce how to build data-efficient TTS models for both new languages and new speakers (Chap. 11); (5) At last, we briefly introduce some tasks beyond text-to-speech synthesis, including singing voice synthesis, voice conversion, speech enhancement, and speech separation (Chap. 12).

1.3.4 Other Taxonomies of TTS

Besides the main taxonomy from the perspective of key components and advanced topics, we can also categorize neural TTS from several different taxonomies, as shown in Fig. 1.2: (1) **Autoregressive or non-autoregressive**. We can divide neural TTS into autoregressive and non-autoregressive generations. (2) **Generative model**. Since TTS is a typical sequence generation task and can be modeled through deep generative models, we categorize it in terms of different generative models: normal sequence generation model, normalizing flows (Flow), generative adversarial networks (GAN), variational auto-encoders (VAE), and denoising diffusion probabilistic models (DDPM or Diffusion). (3) **Network structure**. We can divide neural TTS according to the network structures, such as CNN, RNN, self-attention, and hybrid structures (which contain more than one type of structure, such as CNN+RNN, and CNN+self-attention).

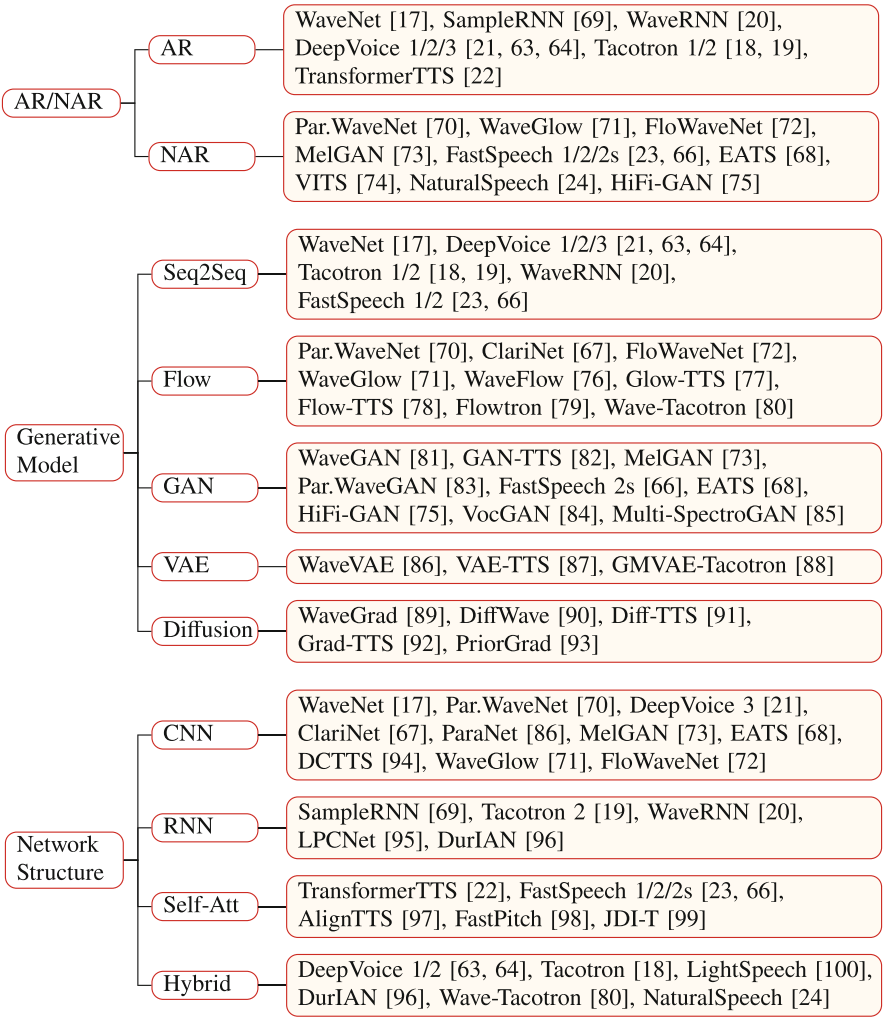


Fig. 1.2 Some other taxonomies of neural TTS from the perspectives of AR/NAR, generative model, and network structure

1.3.5 Evolution of Neural TTS

In order to better understand the development of neural TTS, we illustrate the evolution of neural TTS models, as shown in Fig. 1.3. 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. Since the research works on neural TTS are so abundant, we only choose some representative works in Fig. 1.3, and list more works in Table B.1.

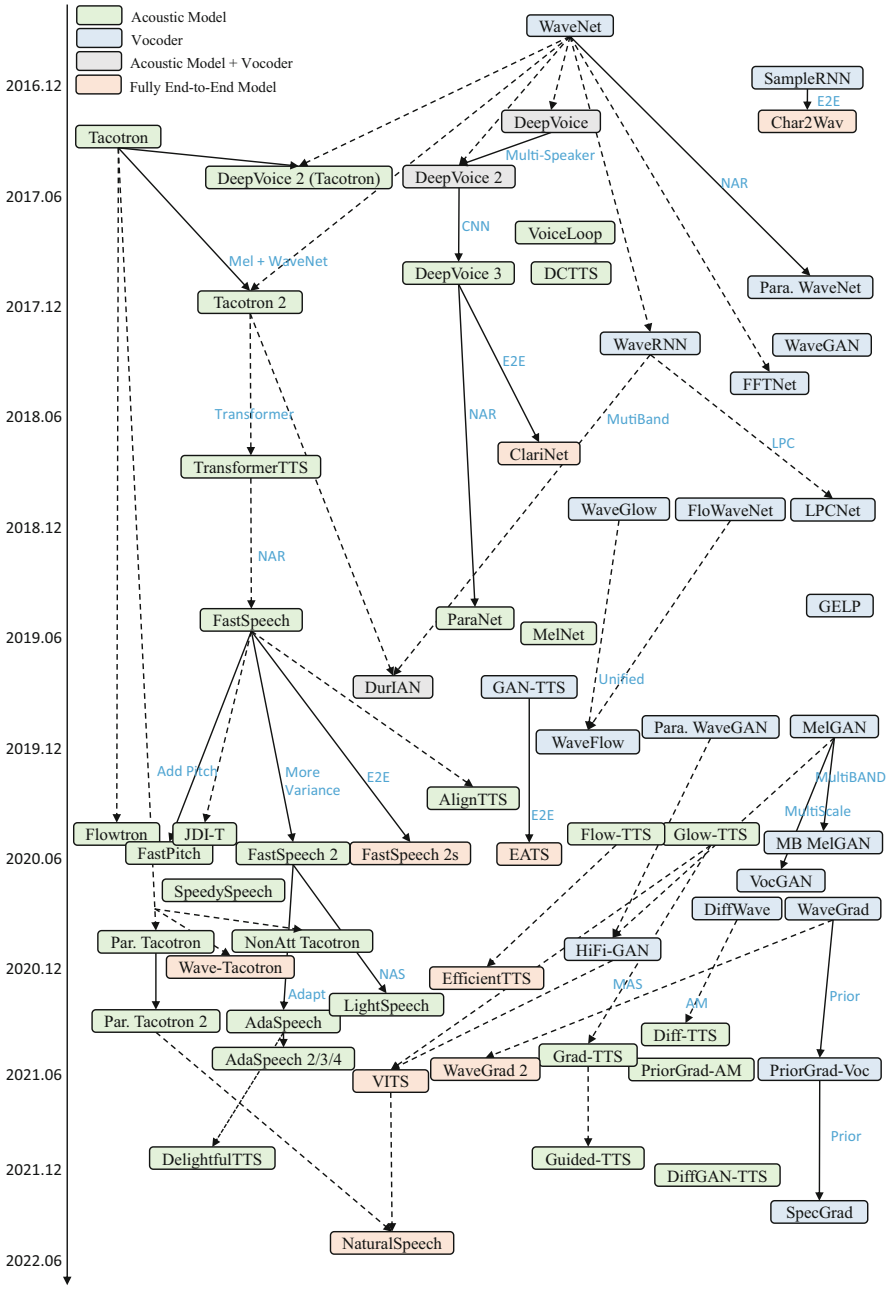


Fig. 1.3 The evolution of neural TTS models

1.4 Organization of This Book

In the introduction (Chap. 1), we describe the motivation of this book, the history of TTS technologies, and the recently developed neural TTS technology, and then introduce some taxonomies about neural TTS. The remaining of this book consists of four parts: Preliminary (Part I), Key Components in TTS (Part II), Advanced Topics in TTS (Part III), and Summary/Outlook (Part IV), as shown in Fig. 1.4. In Part I, some preliminary knowledge about neural TTS is introduced, including the basics of spoken language processing (Chap. 2) and deep learning (Chap. 3) since neural TTS involves the processing of spoken languages and leverages the methods from deep learning. Particularly, since TTS is a data generation task that relies on deep generative models, we cover some background of deep generative models in Chap. 3. In Part II, we introduce several key components in neural TTS, including text analyses (Chap. 4), acoustic models (Chap. 5), vocoders (Chap. 6), and fully end-to-end models (Chap. 7). In Part III, we introduce some advanced topics to address these challenges in neural TTS, including expressive and controllable TTS (Chap. 8), robust TTS (Chap. 9), model-efficient TTS (Chap. 10), data-efficient TTS (Chap. 11), and some tasks beyond TTS (Chap. 12), such as singing voice synthesis, voice conversion, and speech enhancement/separation. In the last part, we summarize this book and discuss future research directions (Chap. 13). To further enrich this book, we summarize TTS-related resources including open-source implementations, corpora, and TTS model list in Appendix.

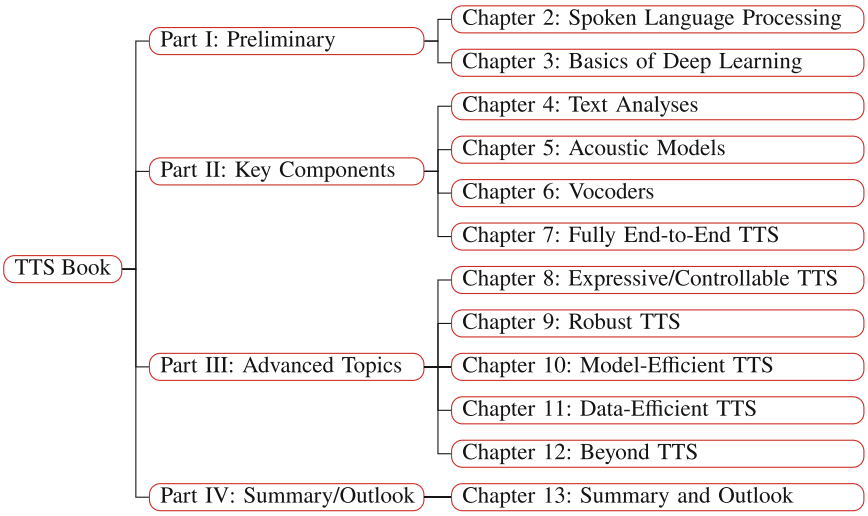


Fig. 1.4 Organization of this book

References

1. Taylor P (2009) Text-to-speech synthesis. Cambridge University Press
2. Tan X, Qin T, Soong F, Liu TY (2021) A survey on neural speech synthesis. Preprint. arXiv:2106.15561
3. Adler RB, Rodman GR, Sévigny A (1991) Understanding human communication. Holt, Rinehart and Winston Chicago
4. Russell S, Norvig P (2020) Artificial intelligence: a modern approach (4th Edition). Pearson. http://aima.cs.berkeley.edu/
5. Manning C, Schutze H (1999) Foundations of statistical natural language processing. MIT Press
6. Jurafsky D (2000) Speech & language processing. Pearson Education India
7. De Saussure F (2011) Course in general linguistics. Columbia University Press
8. Kinsler LE, Frey AR, Coppens AB, Sanders JV (1999) Fundamentals of acoustics. John Wiley & Sons
9. Yuen CK (1978) Review of "Theory and Application of Digital Signal Processing" by Lawrence R. Rabiner and Bernard Gold. IEEE Trans Syst Man Cybern 8(2):146. https://doi.org/10.1109/TSMC.1978.4309918
10. Bishop CM (2006) Pattern recognition and machine learning. Springer
11. Jordan MI, Mitchell TM (2015) Machine learning: trends, perspectives, and prospects. Science 349(6245):255–260
12. Hunt AJ, Black AW (1996) 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, vol 1. IEEE, pp 373–376
13. Zen H, Tokuda K, Black AW (2009) Statistical parametric speech synthesis. Speech Commun 51(11):1039–1064
14. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
15. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press
16. Zen H, Senior A, Schuster M (2013) Statistical parametric speech synthesis using deep neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, pp 7962–7966
17. van den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K (2016) WaveNet: A generative model for raw audio. Preprint. arXiv:1609.03499
18. Wang Y, Skerry-Ryan R, Stanton D, Wu Y, Weiss RJ, Jaitly N, Yang Z, Xiao Y, Chen Z, Bengio S, et al (2017) Tacotron: Towards end-to-end speech synthesis. In: Proc Interspeech 2017, pp 4006–4010
19. Shen J, Pang R, Weiss RJ, Schuster M, Jaitly N, Yang Z, Chen Z, Zhang Y, Wang Y, Skerry-Ryan R, et al (2018) Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4779–4783
20. Kalchbrenner N, Elsen E, Simonyan K, Noury S, Casagrande N, Lockhart E, Stimberg F, Oord A, Dieleman S, Kavukcuoglu K (2018) Efficient neural audio synthesis. In: International conference on machine learning. PMLR, pp 2410–2419
21. Ping W, Peng K, Gibiansky A, Arik SO, Kannan A, Narang S, Raiman J, Miller J (2018) Deep Voice 3: 2000-speaker neural text-to-speech. In: Proc ICLR, pp 214–217
22. Li N, Liu S, Liu Y, Zhao S, Liu M (2019) Neural speech synthesis with Transformer network. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 6706–6713
23. Ren Y, Ruan Y, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2019) FastSpeech: fast, robust and controllable text to speech. In: NeurIPS
24. Tan X, Chen J, Liu H, Cong J, Zhang C, Liu Y, Wang X, Leng Y, Yi Y, He L, et al (2022) NaturalSpeech: End-to-end text to speech synthesis with human-level quality. Preprint. arXiv:2205.04421

25. Tabet Y, Boughazi M (2011) Speech synthesis techniques. a survey. In: International workshop on systems, signal processing and their applications, WOSSPA. IEEE, pp 67–70
26. Mali P (2014) A survey on text to speech translation of multi language. Int J Res Adv Eng Technol. ISSN 2347-2812
27. Siddhi D, Verghese JM, Bhavik D (2017) Survey on various methods of text to speech synthesis. Int J Comput Appl 165(6):26–30
28. Ning Y, He S, Wu Z, Xing C, Zhang LJ (2019) A review of deep learning based speech synthesis. Appl Sci 9(19):4050
29. Hsu Pc, Wang Ch, Liu AT, Lee Hy (2019) Towards robust neural vocoding for speech generation: A survey. Preprint. arXiv:1912.02461
30. Panda SP, Nayak AK, Rai SC (2020) A survey on speech synthesis techniques in Indian languages. Multimedia Syst 26:453–478
31. Mu Z, Yang X, Dong Y (2021) Review of end-to-end speech synthesis technology based on deep learning. Preprint. arXiv:2104.09995
32. Wikipedia (2021) Speech synthesis — Wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Speech%20synthesis&oldid=1020857981
33. Dudley H, Tarnoczy TH (1950) The speaking machine of Wolfgang von Kempelen. J Acoust Soc Am 22(2):151–166
34. Coker CH (1976) A model of articulatory dynamics and control. Proc IEEE 64(4):452–460
35. Shadle CH, Damper RI (2001) Prospects for articulatory synthesis: a position paper. In: 4th ISCA tutorial and research workshop (ITRW) on speech synthesis
36. Seeviour P, Holmes J, Judd M (1976) Automatic generation of control signals for a parallel formant speech synthesizer. In: ICASSP'76. IEEE International conference on acoustics, speech, and signal processing, vol 1. IEEE, pp 690–693
37. Allen J, Hunnicutt S, Carlson R, Granstrom B (1979) MITalk-79: The 1979 MIT text-to-speech system. J Acoust Soc Am 65(S1):S130–S130
38. Klatt DH (1980) Software for a cascade/parallel formant synthesizer. J Acoust Soc Am 67(3):971–995
39. Klatt DH (1987) Review of text-to-speech conversion for English. J Acoust Soc Am 82(3):737–793
40. Olive J (1977) Rule synthesis of speech from dyadic units. In: ICASSP'77. IEEE International conference on acoustics, speech, and signal processing, vol 2. IEEE, pp 568–570
41. Moulines E, Charpentier F (1990) Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. Speech Commun 9(5–6):453–467
42. Sagisaka Y, Kaiki N, Iwahashi N, Mimura K (1992) ATR ν-Talk speech synthesis system. In: Second international conference on spoken language processing
43. Taylor P, Black AW, Caley R (1998) The architecture of the Festival speech synthesis system. In: The Third ESCA/COCOSDA Workshop on Speech Synthesis, Blue Mountains, Australia, November 26–29, 1998. ISCA, pp 147–152. http://www.isca-speech.org/archive_open/ssw3/ssw3_147.html
44. Yoshimura T, Tokuda K, Masuko T, Kobayashi T, Kitamura T (1999) Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In: Sixth European conference on speech communication and technology
45. Tokuda K, Yoshimura T, Masuko T, Kobayashi T, Kitamura T (2000) Speech parameter generation algorithms for HMM-based speech synthesis. In: 2000 IEEE international conference on acoustics, speech, and signal processing. proceedings (Cat. No. 00CH37100), vol 3. IEEE, pp 1315–1318
46. Tokuda K, Nankaku Y, Toda T, Zen H, Yamagishi J, Oura K (2013) Speech synthesis based on hidden Markov models. Proc IEEE 101(5):1234–1252
47. Qian Y, Fan Y, Hu W, Soong FK (2014) 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). IEEE, pp 3829–3833

48. Fan Y, Qian Y, Xie FL, Soong FK (2014) TTS synthesis with bidirectional LSTM based recurrent neural networks. In: Fifteenth annual conference of the international speech communication association
49. Zen H, Sak H (2015) 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). IEEE, pp 4470–4474
50. Wang W, Xu S, Xu B (2016) First step towards end-to-end parametric TTS synthesis: Generating spectral parameters with neural attention. In: Interspeech, pp 2243–2247
51. Li H, Kang Y, Wang Z (2018) EMPHASIS: An emotional phoneme-based acoustic model for speech synthesis system. In: Proc Interspeech 2018, pp 3077–3081
52. Yoshimura T (2002) Simultaneous modeling of phonetic and prosodic parameters, and characteristic conversion for HMM-based text-to-speech systems. PhD diss, Nagoya Institute of Technology
53. Fukada T, Tokuda K, Kobayashi T, Imai S (1992) An adaptive algorithm for mel-cepstral analysis of speech. In: Proc. ICASSP, vol 1, pp 137–140
54. Tokuda K, Kobayashi T, Masuko T, Imai S (1994) Mel-generalized cepstral analysis-a unified approach to speech spectral estimation. In: Third international conference on spoken language processing
55. Kawahara H, Masuda-Katsuse I, De Cheveigne A (1999) 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 Commun 27(3–4):187–207
56. Imai S, Sumita K, Furuichi C (1983) Mel log spectrum approximation (MLSA) filter for speech synthesis. Electron Commun Japan (Part I: Commun) 66(2):10–18
57. Imai S (1983) Cepstral analysis synthesis on the mel frequency scale. In: ICASSP'83. IEEE International conference on acoustics, speech, and signal processing, vol 8. IEEE, pp 93–96
58. Kawahara H (2006) STRAIGHT, exploitation of the other aspect of vocoder: perceptually isomorphic decomposition of speech sounds. Acoust Sci Technol 27(6):349–353
59. Morise M, Yokomori F, Ozawa K (2016) WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. IEICE Trans Inf Syst 99(7):1877–1884
60. Sproat R, Black AW, Chen S, Kumar S, Ostendorf M, Richards C (2001) Normalization of non-standard words. Comput Speech Lang 15(3):287–333
61. Bisani M, Ney H (2008) Joint-sequence models for grapheme-to-phoneme conversion. Speech Commun 50(5):434–451
62. Zen H (2015) Acoustic modeling in statistical parametric speech synthesis-from HMM to LSTM-RNN. In: Proc MLSLP. Invited paper
63. Arık SÖ, Chrzanowski M, Coates A, Diamos G, Gibiansky A, Kang Y, Li X, Miller J, Ng A, Raiman J, et al (2017) Deep Voice: Real-time neural text-to-speech. In: International conference on machine learning, PMLR, pp 195–204
64. Gibiansky A, Arik SÖ, Diamos GF, Miller J, Peng K, Ping W, Raiman J, Zhou Y (2017) Deep Voice 2: Multi-speaker neural text-to-speech. In: NIPS
65. Sotelo J, Mehri S, Kumar K, Santos JF, Kastner K, Courville AC, Bengio Y (2017) Char2wav: End-to-end speech synthesis. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Workshop Track Proceedings. OpenReview.net. https://openreview.net/forum?id=B1VWyySKx
66. Ren Y, Hu C, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2021) FastSpeech 2: fast and high-quality end-to-end text to speech. In: International conference on learning representations. https://openreview.net/forum?id=piLPYqxtWuA
67. Ping W, Peng K, Chen J (2018) ClariNet: parallel wave generation in end-to-end text-to-speech. In: International conference on learning representations
68. Donahue J, Dieleman S, Bińkowski M, Elsen E, Simonyan K (2021) End-to-end adversarial text-to-speech. In: ICLR
69. Mehri S, Kumar K, Gulrajani I, Kumar R, Jain S, Sotelo J, Courville A, Bengio Y (2017) SampleRNN: An unconditional end-to-end neural audio generation model. In: ICLR

70. van den Oord A, Li Y, Babuschkin I, Simonyan K, Vinyals O, Kavukcuoglu K, Driessche G, Lockhart E, Cobo L, Stimberg F, et al (2018) Parallel WaveNet: Fast high-fidelity speech synthesis. In: International conference on machine learning. PMLR, pp 3918–3926
71. Prenger R, Valle R, Catanzaro B (2019) WaveGlow: a flow-based generative network for speech synthesis. In: ICASSP 2019-2019 IEEE International conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 3617–3621
72. Kim S, Lee SG, Song J, Kim J, Yoon S (2019) FloWaveNet: a generative flow for raw audio. In: International conference on machine learning. PMLR, pp 3370–3378
73. Kumar K, Kumar R, de Boissiere T, Gestin L, Teoh WZ, Sotelo J, de Brébisson A, Bengio Y, Courville A (2019) MelGAN: Generative adversarial networks for conditional waveform synthesis. In: NeurIPS
74. Kim J, Kong J, Son J (2021) Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. Preprint. arXiv:2106.06103
75. Kong J, Kim J, Bae J (2020) HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. Adv Neural Inf Process Syst 33:17022
76. Ping W, Peng K, Zhao K, Song Z (2020) WaveFlow: a compact flow-based model for raw audio. In: International conference on machine learning. PMLR, pp 7706–7716
77. Kim J, Kim S, Kong J, Yoon S (2020) Glow-TTS: A generative flow for text-to-speech via monotonic alignment search. Adv Neural Inf Process Syst 33:8067
78. Miao C, Liang S, Chen M, Ma J, Wang S, Xiao J (2020) 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). IEEE, pp 7209–7213
79. Valle R, Shih K, Prenger R, Catanzaro B (2020) Flowtron: an autoregressive flow-based generative network for text-to-speech synthesis. Preprint. arXiv:2005.05957
80. Weiss RJ, Skerry-Ryan R, Battenberg E, Mariooryad S, Kingma DP (2021) Wave-Tacotron: Spectrogram-free end-to-end text-to-speech synthesis. In: 2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
81. Donahue C, McAuley J, Puckette M (2018) Adversarial audio synthesis. In: International conference on learning representations
82. Bińkowski M, Donahue J, Dieleman S, Clark A, Elsen E, Casagrande N, Cobo LC, Simonyan K (2019) High fidelity speech synthesis with adversarial networks. In: International conference on learning representations
83. Yamamoto R, Song E, Kim JM (2020) 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). IEEE, pp 6199–6203
84. Yang J, Lee J, Kim Y, Cho HY, Kim I (2020) VocGAN: A high-fidelity real-time vocoder with a hierarchically-nested adversarial network. In: Proc Interspeech 2020, pp 200–204
85. Lee SH, Yoon HW, Noh HR, Kim JH, Lee SW (2020) Multi-SpectroGAN: High-diversity and high-fidelity spectrogram generation with adversarial style combination for speech synthesis. Preprint. arXiv:2012.07267
86. Peng K, Ping W, Song Z, Zhao K (2020) Non-autoregressive neural text-to-speech. In: International conference on machine learning. PMLR, pp 7586–7598
87. Zhang YJ, Pan S, He L, Ling ZH (2019) 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). IEEE, pp 6945–6949
88. Hsu WN, Zhang Y, Weiss RJ, Zen H, Wu Y, Wang Y, Cao Y, Jia Y, Chen Z, Shen J, et al (2018) Hierarchical generative modeling for controllable speech synthesis. In: International conference on learning representations
89. Chen N, Zhang Y, Zen H, Weiss RJ, Norouzi M, Chan W (2021) WaveGrad: Estimating gradients for waveform generation. In: ICLR
90. Kong Z, Ping W, Huang J, Zhao K, Catanzaro B (2021) DiffWave: A versatile diffusion model for audio synthesis. In: ICLR

91. Jeong M, Kim H, Cheon SJ, Choi BJ, Kim NS (2021) Diff-TTS: A denoising diffusion model for text-to-speech. Preprint. arXiv:2104.01409
92. Popov V, Vovk I, Gogoryan V, Sadekova T, Kudinov M (2021) Grad-TTS: a diffusion probabilistic model for text-to-speech. Preprint. arXiv:2105.06337
93. Lee Sg, Kim H, Shin C, Tan X, Liu C, Meng Q, Qin T, Chen W, Yoon S, Liu TY (2021) PriorGrad: Improving conditional denoising diffusion models with data-driven adaptive prior. Preprint. arXiv:2106.06406
94. Tachibana H, Uenoyama K, Aihara S (2018) 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). IEEE, pp 4784–4788
95. Valin JM, Skoglund J (2019) LPCNet: Improving neural speech synthesis through linear prediction. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 5891–5895
96. Yu C, Lu H, Hu N, Yu M, Weng C, Xu K, Liu P, Tuo D, Kang S, Lei G, et al (2020) DurIAN: Duration informed attention network for speech synthesis. In: Proc Interspeech 2020, pp 2027–2031
97. Zeng Z, Wang J, Cheng N, Xia T, Xiao J (2020) 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). IEEE, pp 6714–6718
98. Łańcucki A (2020) FastPitch: Parallel text-to-speech with pitch prediction. Preprint. arXiv:2006.06873
99. Lim D, Jang W, Gyeonghwan O, Park H, Kim B, Yoon J (2020) JDI-T: Jointly trained duration informed Transformer for text-to-speech without explicit alignment. In: Proc Interspeech 2020, pp 4004–4008
100. Luo R, Tan X, Wang R, Qin T, Li J, Zhao S, Chen E, Liu TY (2021) LightSpeech: Lightweight and fast text to speech with neural architecture search. In: 2021 IEEE International conference on acoustics, speech and signal processing (ICASSP). IEEE

Part I
Preliminary

Neural text-to-speech synthesis relies on spoken language processing and deep neural networks as the building blocks. Thus, to better understand neural text-to-speech synthesis, we introduce some preliminary knowledge on spoken language processing (Chap. 2) and deep learning (Chap. 3). As text-to-speech synthesis is a kind of data generation task, we introduce typical deep generative models that are widely used in speech synthesis (Chap. 3).

Chapter 2
Basics of Spoken Language Processing

Abstract In this chapter, we introduce some basics of spoken language processing (including both speech and natural language), which are fundamental to text-to-speech synthesis. Since speech and language are studied in the discipline of linguistics, we first overview some basic knowledge in linguistics and discuss a key concept called speech chain that is closely related to TTS. Then, we introduce speech signal processing, which covers the topics of digital signal processing, speech processing in the time and frequency domain, cepstrum analysis, linear prediction analysis, and speech parameter estimation. At last, we overview some typical speech processing tasks.

Keywords Spoken language processing · Linguistics · Speech chain · Speech signal processing

In this chapter, we introduce some basics of spoken language processing (both speech and natural language), which are fundamental to text-to-speech synthesis. Since speech and language are studied in the discipline of linguistics, we first overview some basic knowledge in linguistics (Sect. 2.1) and discuss a key concept called speech chain that is closely related to speech synthesis (Sect. 2.2). Then, we introduce speech signal processing, which covers the topics of digital signal processing, speech processing in the time and frequency domain, cepstral analysis, linear prediction analysis, and speech parameter estimation (Sect. 2.3). At last, we overview some typical speech processing tasks.

Prerequisite Knowledge for Reading This Chapter

- Basic knowledge of algebra, calculus, and probability theory.
- Basic knowledge of language and speech processing.

X. Tan, *Neural Text-to-Speech Synthesis*, Artificial Intelligence: Foundations, Theory, and Algorithms, https://doi.org/10.1007/978-981-99-0827-1_2

2.1 Overview of Linguistics

We first introduce some basic concepts of language and speech and their relationship to linguistics.

- Language. Language is a structured system of communication used by humans and consists of different types such as spoken language, written language, and sign language [1]. That is to say, language is the ability to produce and comprehend spoken, written, or sign words.
- Speech. Speech is the vocalized form of human communication and is a way to express spoken language [2]. Here we further illustrate some terminologies related to speech, including sound, voice, and audio: (1) Sound refers to anything that can be heard by humans, and may come from humans (e.g., talking, laughing, clapping hands together), animals (e.g., birdcall, bark), or physical objects (e.g., object collision, slamming a door). (2) Voice is the sound produced by humans or other vertebrates using the lungs and the vocal cords in the larynx or voice box. (3) Speech is the sound produced by humans for communication purposes (some voices from humans are not speech, e.g., the sound of snoring). It can be seen that speech belongs to voice and voice belongs to sound. (4) Audio is used to represent sound electrically and is made of electrical energy (analog or digital signals). As a comparison, the sound is made of mechanical wave energy (longitudinal sound waves). A speech signal can be represented as a mechanical waveform, and can be converted into an electrical waveform (audio) with a microphone and then processed by digital or analog signal processing technologies and converted back into a mechanical waveform with a loudspeaker.
- Linguistics. Linguistics is the scientific study of language [3] and covers several branches: phonetics, phonology, morphology, syntax, semantics, and pragmatics. These branches roughly correspond to different ingredients in linguistic systems, including speech sounds (and gestures, in the case of sign languages), basic units (phonemes, words, morphemes), phrases and sentences, meaning, and language usage, as shown in Fig. 2.1.

  We introduce these branches in linguistics in the following subsections.

2.1.1 Phonetics and Phonology

Phonetics and phonology both deal with speech sounds [4]. Phonetics is the science that studies speech sounds and their production, transmission, and perception, and provides methods for their analysis, classification, and transcription [4]. It consists of three important subdisciplines that correspond to speech production, transmission, and perception respectively [5]: (1) Articulatory phonetics, which studies how the sounds are made with articulators. (2) Acoustic phonetics, which studies the acoustic results of different articulations. (3) Auditory phonetics, which studies how

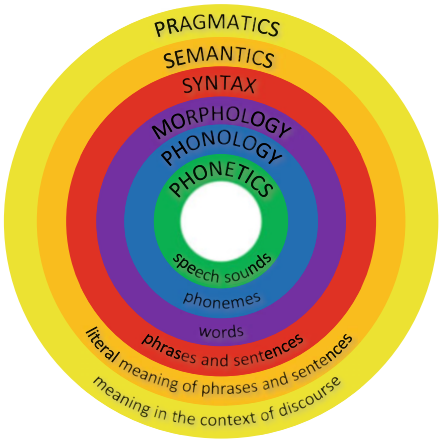


Fig. 2.1 The five linguistics branches (phonetics, phonology, morphology, syntax, semantics, and pragmatics) and their corresponding ingredients in linguistics

the speech sounds are perceived and understood by listeners. Phonology [6] is the science that studies the systematic organization of sounds in languages and studies the systems of phonemes (the smallest set of units that represent distinctive sounds of a language and bring a difference in word meaning) in particular languages. It cares about patterns of sounds in different positions of words or in different languages.

There are key differences between phonetics and phonology [7]: (1) Phonetics is related to phones, while phonology is related to phonemes. (2) Phonetics is about the physical aspect of sounds (human ear), while phonology is the abstract aspect of sounds (human brain). (3) Phonetics studies speech sounds in general, regardless of different languages, while phonology studies speech sounds in particular, in one language or different languages.

2.1.2 Morphology and Syntax

Morphology [8] studies the structure of words, while syntax studies the structure of sentences. In morphology, we understand how words are formed from morphemes, which are the smallest meaningful units of language. Morphemes contain the root, stem, prefix, and suffix of words, such as “cat”, “sat”, “un-”, “-ed”. Morphology determines the meaning of words through these morphemes. In syntax [8], we understand how sentences are developed from words, which are the smallest units in the study of syntax. The syntax looks at the processes and rules of constructing a sentence, the relations between words, and the grammatical structure of sentences, which can determine the meaning of a sentence. For example, “the cat sat” is a simple sentence made of a subject and a verb.

2.1.3 *Semantics and Pragmatics*

Semantics studies the meaning of morphemes, words, phrases, sentences, and their relation, while pragmatics studies the use of language and how people produce and comprehend the meaning in different contexts [9]. Therefore, there are some key differences between semantics and pragmatics: (1) Semantics studies the meaning without considering the context of language usage, while pragmatics studies the meaning by considering the context of language usage. (2) Semantics cares about conceptual meaning based on vocabulary and grammar, and pragmatics cares about the intended meaning based on the context and the inference from readers or listeners when interpreting the sentence. (3) Pragmatics is a broader field compared to semantics.

Table 2.1 lists the characteristics of phonetics, phonology, morphology, syntax, semantics, and pragmatics in linguistics, and their relevance to TTS.

2.2 Speech Chain

An important concept related to speech signal processing is the speech chain [10], which is a model of speech communication between speakers and listeners. The speech chain analyzes the process of converting an intention from a speaker to an understanding of this intention by the listener, as shown in Fig. 2.2. It consists of five levels: (1) Linguistic level, where an intention/message is expressed by selecting and ordering suitable sentences/words/phonemes (related to grammar and phonological coding), which are used to generate the basic sounds of the communication. (2) Physiological level, where the sounds of the linguistics units of the message are generated by the vocal tract components guided by neural and muscular activities (related to articulatory phonetics). (3) Acoustic level, where the sound waves are generated from the lips and nostrils and transmitted to both the speaker through sound feedback and the listener through airborne (related to acoustic phonetics). (4) Physiological level, where the sounds come at the ear of the listener and activate the hearing system, and are perceived by the auditory nerves (related

Table 2.1 Comparison between phonetics, phonology, morphology, syntax, semantics, and pragmatics in linguistics

Branch	Object of study	Granularity	Relevance to TTS
Phonetics	All human sounds	Waveform	******
Phonology	Classified sounds	Phoneme	******
Morphology	Words	Morpheme	****
Syntax	Sentences	Word	***
Semantics	Meaning	Morpheme/Word/Phrase/Sentence	**
Pragmatics	Language use	Morpheme/Word/Phrase/Sentence/Context	*

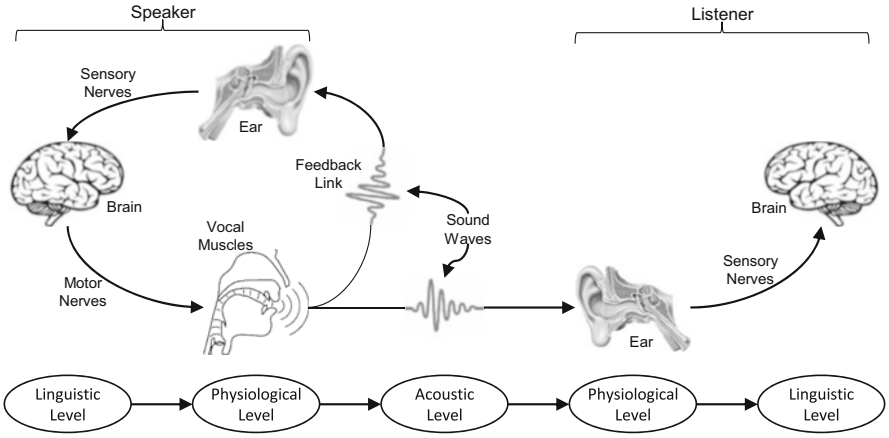


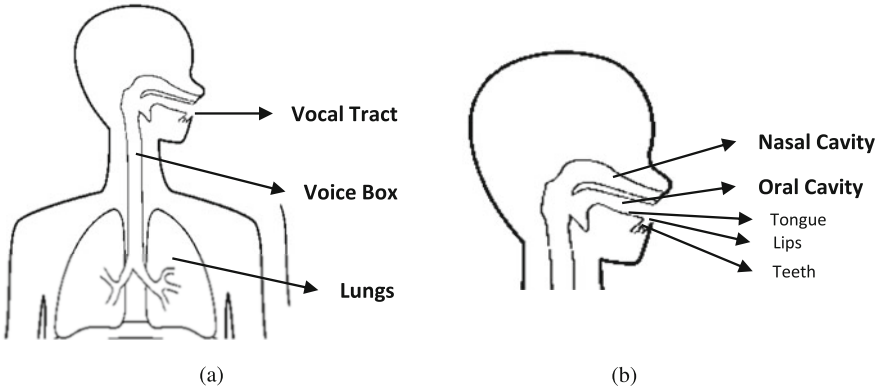
Fig. 2.2 The speech chain: the different forms of a spoken message in its progress from the brain of the speaker to the brain of the listener. (Reproduced from [11])

to auditory phonetics). (5) Linguistic level, where neural activities perceived in the previous stage are recognized as phonemes/words/sentences to understand the intention/message transmitted by the speaker (related to cognitive understanding).

In the next subsections, we introduce the three processes in the speech chain that are closely related to text-to-speech synthesis: speech production, speech transmission, and speech perception, which correspond to the three important subdisciplines of phonetics respectively: articulatory phonetics, acoustic phonetics, and auditory phonetics [5].

2.2.1 Speech Production and Articulatory Phonetics

Articulatory phonetics studies how sounds are made with human articulators. If a human wants to generate speech, the human brain first generates the concept of text and controls the organs of the speech production system to generate the speech waveform corresponding to the text concept. The human speech production system consists of three parts: lungs, voice box, and vocal tract, as shown in Fig. 2.3. The lungs pump air up towards the voice box and vocal tract, as shown in Fig. 2.3a. If the vocal cords in the voice box are tensed, then the airflow causes them to vibrate, which produces voiced sounds or quasi-periodic sounds. If the vocal cords are relaxed, then the airflow just passes through the voice box and enters the vocal tract to produce unvoiced sounds. The nasal cavity and oral cavity (as shown in Fig. 2.3b) in the vocal tract resonate voice to generate sounds such as vowels and consonants.



(a) (b)

Fig. 2.3 The human speech production system. (**a**) Lungs, voice box, and vocal tract. (**b**) Vocal tract

Voiced vs Unvoiced and Vowels vs Consonants

We describe two different classifications of sounds: voiced vs unvoiced, and vowels vs consonants. Voiced and unvoiced sounds depend on whether the vocal cords vibrate or not when producing the sounds: voiced sounds correspond to vibration (e.g., /a/, /m/) while unvoiced sounds correspond to no vibration (e.g., /s/, /t/). The difference between vowels and consonants depends on whether the airflow is constricted by the nasal cavity or oral cavity: vowels are produced by keeping the nasal or oral cavity open (e.g., /a/, /i/), while consonants are produced by constricting the oral or nasal cavity (e.g., /t/, /m/). Consonants can be voiced (e.g., /m/, /b/) or unvoiced (e.g., /p/, /t/). Most of the vowels are voiced, while there exist unvoiced vowels in some languages or in whispers. Table 2.2 shows some examples of vowels and consonants, as well as their voiced and unvoiced property.

Figure 2.4 illustrates the tongue positions of some vowels, where the vertical axis represents the vowel closeness, with close/high vowels at the top of the chart (e.g., i, u) while open/low vowels at the bottom of the chart (e.g., a), and the horizontal axis represents the vowel backness, with front vowels at the left of the chart (e.g., i, a) while back vowels at the right of the chart (e.g., u, ɒ).

Source-Filter Model

The source–filter model describes the process of speech production as a two-stage process, as shown in Fig. 2.5, where a sound source is first generated (e.g., by the vocal cords) and then shaped/filtered by an acoustic filter (e.g., the vocal tract). The sound sources can be either periodic (pulse generator) or aperiodic (noise generator), or a mixture of the two. Basically, a filter is used to selectively let something pass

Table 2.2 Some examples of vowels and consonants, and their voiced and unvoiced property

Vowels/consonants	Sub-types	Unvoiced	Voiced
Vowels	Monophthong (Front)		ɪ iː e æ
	Monophthong (Mid)		ʌ ə ɜː
	Monophthong (Back)		ɑː ɔː uː ɒ ʊ
	Diphthong		aɪ eɪ aʊ əʊ ɔɪ ɪə eə ʊə
Consonants	Plosive	p	b
		t	d
		k	g
	Fricative	f	v
		θ	ð
		s	z
		ʃ	ʒ
		h	
			r
	Affricate	ʧ	ʤ
		tr	dr
		ts	dz
	Nasal		m n ŋ
	Lateral		l
	Semivowel		w j

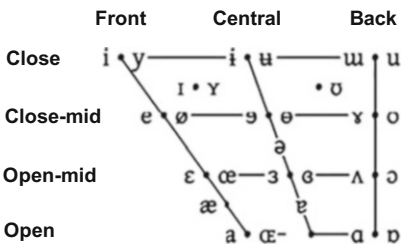


Fig. 2.4 The tongue position of some vowels

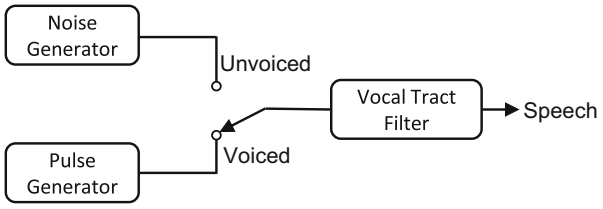


Fig. 2.5 The source-filter model

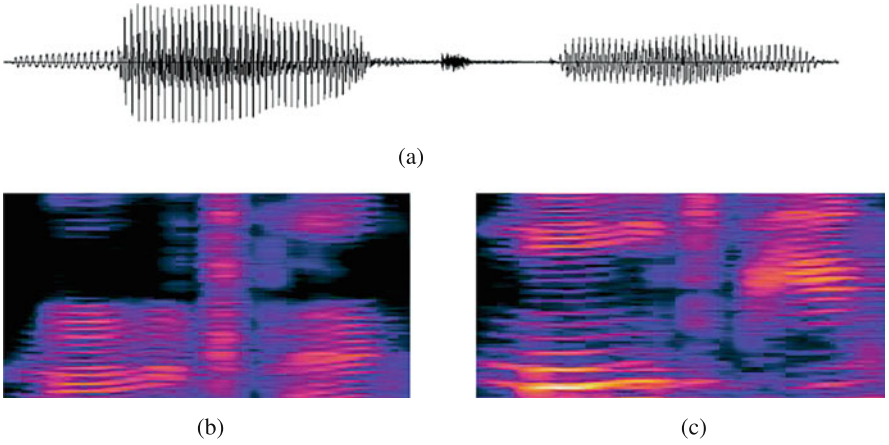
through or block something. The vocal tract filter can allow some frequencies to pass through by reducing the intensity of other frequencies.

The source-filter model can be used to analyze human speech production and artificial speech synthesis systems. In human speech production, the vocal cords produce sound sources as either periodic sounds when vibrating or aperiodic sounds when relaxing, and the vocal tract (i.e., the pharynx, mouth, and nasal cavity) manipulate the sound sources to produce the speech sounds. From the frequency perspective, the vocal cords produce a series of harmonics (including the fundamental frequency) of different amplitudes, and the vocal tract either amplifies or attenuates certain frequencies to generate desired sounds. In artificial speech synthesis, the systems based on the source-filter model usually adopt a periodic impulse train as the source for voiced sound and white noise as the source for unvoiced sound. The filter of the systems is usually implemented with an all-pole filter, and the coefficients of this all-pole filter are learned by linear prediction via minimizing the errors between the synthesized speech and ground-truth speech [12]. The filter can be also implemented with neural networks [13] in the era of deep learning.

2.2.2 Speech Transmission and Acoustic Phonetics

After the speech is generated by a human articulator in the speech production process, we can get the speech sound. In the subsection, we describe the speech transmission part and introduce acoustic phonetics, which studies the physical properties of speech produced by articulators, such as amplitudes, frequencies, and durations. We introduce several physical properties as follows:

- Amplitude. It measures the degree of fluctuations in the air particles caused by sound waves. An example of speech sound waves is shown in Fig. 2.6a where the horizon axis represents the time and the vertical axis represents the amplitude. Amplitude is related to several terms: sound energy, sound power, sound intensity, and sound pressure. We explain these terminologies as follows: (1) Sound energy measures the ability/capacity of sound to move the air particles, and is proportional to the square of the amplitude. The unit of sound energy is the joule (J). (2) Sound power measures the energy created by a sound source every second, and it maintains constant no matter how far the sound travels from the source. The unit of sound power is the watt ($W = J/S$). (3) Sound intensity is the flow of energy (power) through a unit area in one second, measured in power/area. The unit of sound intensity is watts/square meter (W/m^2). As the sound travels from the source, the sound intensity level decreases in a proportion to the square of the distance from the source. (4) Sound pressure is the amount of force caused by the vibration of particles at a unit area. The unit of sound pressure is the Pascal (Pa), where $Pa = N/m^2$. The human ear usually responds to sound pressure in a large range and in a logarithmic scale. Thus, we define sound



(a)

(b) (c)

Fig. 2.6 An example of speech waveform and spectrogram. (**a**) Waveform. (**b**) Linear-spectrogram. (**c**) Mel-spectrogram

pressure level as $SPL = 10\log_{10}\frac{P^2}{P_0^2} = 20\log_{10}\frac{P}{P_0}$ in decibel (dB) scale, where P_0 is a reference value that usually set as the sound pressure at the threshold of hearing $2 * 10^{-5}$ Pa.

- Frequency. It measures the number of cycles on a repeating/periodic waveform per unit of time. The unit of measurement is Hertz (Hz). The higher frequency of a speech waveform is, the higher the pitch human can perceive.
- Fundamental frequency and harmonics. The fundamental frequency is defined as the lowest frequency of a periodic waveform, denoted as F_0, and harmonics are the higher frequency components that are integer multiples of the fundamental frequency. Fundamental frequency usually has a higher amplitude than harmonics and plays an important role in speech processing. For the sound of a musical note, the fundamental frequency is the pitch of this note.
- Sine waves and complex waves. Most speech sounds are complex waves, which consist of two or more simple sine waves. The lowest frequency of these sine waves is the fundamental frequency of this complex wave, which represents the number of times the vocal cords vibrate in a unit of time. The harmonics of this complex wave can be regarded as the amplified frequencies by the human vocal tract under a mechanism called natural resonances, where the frequency of the harmonic that is most augmented by this resonance is called a formant.
- Periodic and aperiodic waves. The voiced and unvoiced sounds described in Sect. 2.2.1 correspond to the periodic and aperiodic waves here. The periodic waves are generated under the vibration of vocal cords and contain fundamental frequency and harmonics. The aperiodic waves are generated without the vibration of vocal cords, which means they do not repeat in a regular pattern and do not contain a fundamental frequency.

- Spectrograms. It is obtained by converting a time-domain waveform signal into the frequency domain and can be regarded as a 2D image, where the horizontal axis of the spectrograms measures time while the vertical axis measures the frequency, and the value of each position in this image is the magnitude of this frequency in a certain time. An example of a spectrogram is shown in Figs. 2.6b and 2.6c.

2.2.3 Speech Perception and Auditory Phonetics

In this subsection, we introduce how speech sound is perceived by humans, i.e., auditory phonetics.

How Human Perceives Sound

Auditory phonetics studies how speech sounds are perceived and understood by listeners. When the speech sounds arrive at the ear of the listener, the auditory system perceives the speech sounds as the following steps: (1) Acoustic-to-neural converter, where the acoustic signals are converted into neural representation through the outer, middle, and inner ears. Specifically, the outer ear receives sound in the ear canal, and the tympanic membrane in the middle ear converts the acoustical sound waves into mechanical vibrations, which are further fired as neural (electrical) impulses in the inner ear. (2) Neural transduction, where neural signals fired by the inner ear are transmitted in the auditory nerve, which acts as the neural pathway to the brain. (3) Neural processing, where the neural firing signals are perceived and processed by the human brain.

Difference Between Auditory Perceptions and Physical Property of Sound

In speech perception, we care about several aspects of the perceived sound, such as the sound pressure level, frequency, and spectral structure (formants). However, there are differences between psychophysical observations from humans and the physical attributes of sounds. Table 2.3 lists the different but corresponding concepts between physical property and auditory perception:

- Loudness is the sound pressure perceived by humans and is measured by the pressure of a sound relative to the pressure at the threshold of hearing, i.e., sound pressure level. It is denoted as $SPL = 10\log_{10}\frac{P^2}{P_0^2} = 20\log_{10}\frac{P}{P_0}$ dB, where $P_0 = 20\,\mu$Pa.
- Pitch is the sensation of sound frequency by humans. Pitch is related to the fundamental frequency of sounds but not exactly the same. Although the pitch is monotonously increasing with frequency, the relationship is not linear,

Table 2.3 Different concepts between physical properties and auditory perceptions

Physical properties	Auditory perceptions
Amplitude	Loudness/sound pressure level
Fundamental frequency	Pitch/tone
Spectrum/harmonic/envelope/formant	Content/timbre

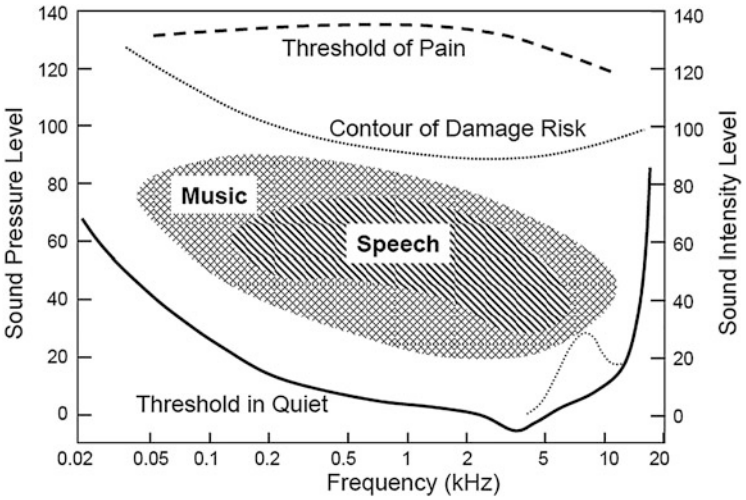


Fig. 2.7 The threshold of sound pressure level (loudness) and frequency perceived by human ear

since humans have different sensibilities on different frequencies. Usually, the relationship between perceived pitch and physical frequency is characterized by the mel-scale frequency: Pitch (mels) $= 2595 \log_{10}(1 + F/700) = 1127 \ln(1 + F/700)$. An example of linear-scale and mel-scale spectrogram is shown in Figs. 2.6b and 2.6c.

- The content and timbre of a speech waveform are largely determined by the details of the spectrum, such as harmonic, envelope, and formant.

The human ear has a range of sound pressure levels (loudness) and frequencies, from the threshold of hearing to the threshold of pain, as shown in Fig. 2.7.

Evaluation Metrics for Speech Perception

We list some metrics to evaluate how good the speech is perceived by humans, including both objective metrics and subjective metrics, as shown in Table 2.4.

For objective metrics: (1) Mel-Cepstral Distortion (MCD) measures the difference between two speech sequences in terms of the mel-cepstra. The smaller the MCD between the synthesized speech and natural (ground-truth) speech, the closer

Table 2.4 Evaluation metrics for speech perception

Type	Metric
Objective metrics	Mel-Cepstral distortion (MCD)
	Signal-to-distortion ratio (SDR)
	Perceptual evaluation of speech quality (PESQ)
	Short-time objective intelligibility (STOI)
Subjective metrics	Intelligibility score (IS)
	Mean opinion score (MOS)
	Comparative MOS (CMOS)
	Similarity MOS (SMOS)

the synthesized speech is to the natural speech. (2) Signal-to-Distortion Ratio (SDR) measures the logarithmic ratio between the power of the ground-truth speech and the error between the synthesized speech and ground-truth speech (i.e., distortion). A larger SDR means better speech synthesis quality. (3) Perceptual Evaluation of Speech Quality (PESQ) [14] and Short-Time Objective Intelligibility (STOI) [15] are two metrics to measure the quality and intelligibility of the synthesized speech.

For subjective metrics: (1) Intelligibility Score (IS) measures how a speech sentence or word can be understandable by listeners. (2) Mean Opinion Score (MOS) measures the speech quality on a 5-point scale. (3) Comparison Mean Opinion Score (CMOS) measures the speech quality by comparing samples from two systems head by head. (4) Similarity Mean Opinion Score (SMOS) measures the speaker similarity between the synthesized speech and the reference speech.

2.3 Speech Signal Processing

Speech signal processing is to represent, transform, analyze, and understand speech signals. We mainly focus on speech signal processing in the digital/discrete form (i.e., digital speech signal processing), which can be easily processed by computer and deep learning technology. We first introduce some basic knowledge about digital signal processing and introduce speech processing in the time and frequency domain. Then, we introduce some advanced topics in speech processing, such as cepstral analysis and speech parameter estimation.

2.3.1 Analog-to-Digital Conversion

When a sound is captured by a microphone, it becomes a continuous-time and continuous-amplitude analog signal, which is further converted into a discrete-time

and discrete-amplitude digital signal for digital signal processing. This analog-to-digital conversion process involves sampling and quantization of the analog signal.

Sampling

A discrete digital signal $x(n)$ can be sampled from continuous analog signal: $x(n) = x_a(t)$, where $t = nT$, $x_a(t)$ is the analog signal, t is the time, and T is the sampling period. If the sampling rate $1/T$ is greater than twice the bands of the continuous signal, then this continuous signal can be reconstructed perfectly according to Nyquist–Shannon sampling theorem [16]. The half of the sampling rate (frequency) is called Nyquist frequency. The bandwidth of an analog-to-digital conversion module is determined by the sampling rate and Nyquist frequency. When the bandwidth (or highest frequency) of a signal is above the Nyquist frequency, the sampled discrete signal will suffer from artifacts, which are known as aliasing. Aliasing refers to the phenomenon that different signals are indistinguishable or aliased together after being sampled, and as a consequence, there are artifacts when the signal is reconstructed from samples.

Quantization

Quantization is used to convert continuous values of a signal into discrete values, which is also called pulse-code modulation (PCM). For example, for a signal with a value range of $[-1, 1]$, we can quantize the values into integers in a range of $[0, 255]$, which is stored in binary with 8 bits. Thus, the resolution of the quantization is 256, or 8 bits. Since the resolution is not infinitely large, quantization usually introduces a small amount of error/noise to the signal, which can be measured by the signal-to-noise ratio (SNR).

There are different types of PCM: (1) linear PCM, where the quantization is conducted in linearly uniform with the amplitude; (2) non-linear PCM, where the quantization levels vary with the amplitude according to different algorithms, such as μ-law (in North America and Japan) or A-law algorithm (in Europe). For an input value x, the output of the μ-law encoding is $F(x) = sgn(x)\frac{\ln(1+\mu|x|)}{\ln(1+\mu)}$, where $-1 \leq x \leq 1$, $\mu = 255$ in North America and Japan, and $sgn(x)$ is the sign function. The output of the A-law encoding is $F(x) = sgn(x)\begin{cases} \frac{A|x|}{1+\ln(A)}, & |x| < \frac{1}{A} \\ \frac{1+\ln(A|x|)}{1+\ln(A)}, & \frac{1}{A} \leq |x| < 1 \end{cases}$, where A is the compression parameter, and $A = 87.6$ in Europe.

2.3.2 Time to Frequency Domain Transformation

We introduce several methods to transform the audio signal from the time domain to the frequency domain, including Discrete-Time Fourier Transform (DTFT), Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT), and Short-Time Fourier Transform (STFT).

Discrete-Time Fourier Transform (DTFT)

The DTFT of a digital signal $x(n)$ is denoted as

$$X(w) = \Sigma_{n=-\infty}^{+\infty} x(n) e^{-jwn}, \tag{2.1}$$

where n is the index of the discrete data $x(n)$, j is the imaginary unit of Euler's formula ($e^{jx} = \cos x + j \sin x$), and w is the digital frequency and its unit is radian/sample. $w = 2\pi f_\Omega / f = \Omega / f = \Omega T$, where f_Ω is the frequency of the continuous analog signal $x_a(t)$, Ω is the analog radian frequency and its unit is radian/second, f and T are the sampling frequency and sampling period of the discrete digital signal $x(n)$. The inverse DTFT is denoted as

$$x(n) = \frac{1}{2\pi} \int_{-\pi/2}^{\pi/2} X(w) e^{jwn} dw. \tag{2.2}$$

Discrete Fourier Transform (DFT)

Since Eq. 2.2 has an integral operation to convert the signal in the frequency domain into the time domain, it is not practical for computation. In practice, we only sample N points in the frequency domain. Then DTFT becomes DFT (the term "time" is omitted since we do not need to emphasize "time" alone considering both time and frequency are discrete). DFT is formulated as

$$X(k) = \Sigma_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn}, \tag{2.3}$$

where $0 \leq k < N - 1$. The inverse of DFT is formulated as

$$x(n) = \frac{1}{N} \Sigma_{k=0}^{N-1} X(k) e^{-j\frac{2\pi}{N}nk}, \tag{2.4}$$

where $0 \leq n < N - 1$, $e^{-j\frac{2\pi}{N}n}$ is the fundamental frequency of the time series, $e^{-j\frac{2\pi}{N}nk}$ is the k-th harmonics, and there are totally N harmonics, since $e^{-\frac{jwn}{N}(k+N)} = e^{-\frac{jwn}{N}k}$.

Fast Fourier Transform (FFT)

FFT has the same mathematical formulation as DFT, but with fast implementation with optimization in the computation, resulting in $\frac{N}{\log_2 N}$ times speedup. Meanwhile, the FFT size (N) should be a power of two to ensure this fast speed.

Short-Time Fourier Transform (STFT)

For unstable signals like speech, DFT/FFF are not suitable. Therefore, we usually divide speech into frames, where each frame lasts for dozens of milliseconds and can be regarded as a stable signal. Then DFT/FFT can be applied. This process is called STFT, which usually consists of several steps:

- Framing, which divides the waveform points into frames.
- Windowing, which adds a window to each frame.
- Transformation, which converts the time-domain signal in each frame into a frequency-domain signal (i.e., spectrum) using DFT or FFT.
- Concatenation, which concatenates the spectrum of each frame together along the time axis to get the spectrograms of the speech signal.

Besides the above processing steps, we usually have some preprocessing and postprocessing steps for STFT:

- Pre-emphasis and de-emphasis. To avoid the distortions caused by the noisy signal, pre-emphasis is usually applied before the framing and windowing to boost the frequency range that is susceptible to noise. After the transmission or processing of this signal, de-emphasis is applied to transform it back to the original signal. In this way, the noise added to the susceptible frequency range in transmission or processing is attenuated.
- Linear-to-mel scale transform. As we introduced in Sect. 2.2.3, human sensibility in terms of frequency is not in linear scale but in mel scale. Therefore, to make the spectrograms more consistent with human sensitivity, we further convert the linear-scale spectrograms obtained by STFT into mel-scale spectrograms using mel filters, such as triangular overlapping windows or cosine overlapping windows.

2.3.3 Cepstral Analysis

We first show a segment of waveform sequence in Fig. 2.8a and its spectrum after applying FFT in Fig. 2.8b. By looking at the spectrum of this speech segment, we can locate the formants and the spectral envelope (which can be regarded as a smoothing curve that connects the formants together), which contain important information about the speech. The high-frequency peaks in the spectrum represent

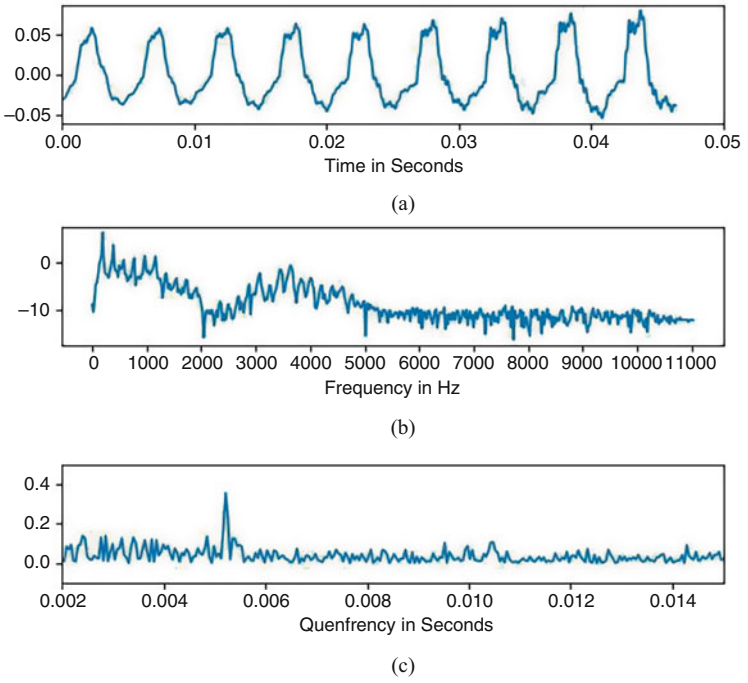


Fig. 2.8 A segment of (**a**) speech waveform, (**b**) its spectrum, and (**c**) its cepstrum

the harmonics, and the low-frequency peaks denote the resonances. We can conduct cepstral analysis by applying the inverse Fourier transform of the logarithm of the spectrum, and get the cepstrum of this speech segment as shown in Fig. 2.8c. The term "cepstrum" is derived from the term "spectrum", by reversing the first letters "spec". Cepstral analysis is to explore the periodic structures in frequency spectra, which has many applications, such as pitch detection and spectral envelope analysis.

We can apply the inverse of DFT of the logarithm magnitude of the spectrum $X(k)$ to get the cepstrum:

$$\hat{x}(n) = \frac{1}{N}\Sigma_{k=0}^{N-1} \ln X(k) e^{-j\frac{2\pi}{N}nk}, \tag{2.5}$$

where $0 \leq n < N-1$. By looking at the resulting cepstrum as shown in Fig. 2.8c, the x-axis represents time quefrency (by swapping the letters "fre" and "que" in "frequency"), where larger time represents lower frequency, while lower time represents higher frequency.

Note that there is a peak around 0.005 s in quefrency (nearly 200 Hz), which equals to vocal cord excitation period, i.e., inverse of fundamental frequency. We can also infer from Fig. 2.8a that this segment of speech waveform has a fundamental frequency of about 200 Hz (there are nearly 10 periods in 0.05 s,

and thus the period is nearly 0.005 s and frequency is about 200 Hz). There is an empirical way to locate fundamental frequency from the cepstrum: we can assume the fundamental frequency of human voice is in the range from 80 to 450 Hz, and then the corresponding peak in the cepstrum should lie in the quefrency range from 0.0022 to 0.0125 s.

2.3.4 Linear Predictive Coding/Analysis

Linear predictive coding (LPC) [12] is a method for compressed coding in speech processing, which leverages a linear predictive model to represent the spectral envelope of a speech signal. LPC is based on the source-filter model, where a sound source $e(n)$ (from vocal cords) goes through a filter $h(n)$ (vocal tract) to formulate a signal $x(n) = h(n) * e(n)$. We usually have the resulting signal $x(n)$, but need to estimate the filter $h(n)$ and source signal $e(n)$.

In LPC, the filter is formulated as a p-th order all-pole filter. Let us recall the digital filter design in digital signal processing. Each pole in a discrete-time system corresponds to a time delay, and thus the output signal $x(n)$ at time step n depends on the current input $e(n)$ and the previous samples $x(n-k), k \in [1, p]$, i.e., $x(n) = \Sigma_{k=1}^{p} a_k * x(n-k) + e(n)$. It is in accordance with the intuition that a speech signal at any time step can be approximated by the linear combination of the past samples. The coefficients can be obtained by minimizing the error $e(n)$ between the ground-truth speech samples and the linear-predicted ones. Given N samples, where $N \gg p$, we can have N equations to determine the coefficient $a_k, k \in [1, p]$, which can be solved in a mathematical way [12]. After we have estimated the filter $h(n)$ (i.e., $a_k, k \in [1, p]$), we can estimate the source signal $e(n)$ by $x(n) - \Sigma_{k=1}^{p} a_k * x(n-k)$. The source signal $e(n)$ can either be an impulse train of a pitch frequency (voiced) or random noise (unvoiced), or a mixture of both voiced and unvoiced sound.

We then describe how LPC can be used for compressed coding in speech processing. Usually, the compression is done on each frame of speech signal due to its time-varying property. We take an example to illustrate compression. For a frame of speech signal with 200 waveform samples, if choosing $p = 16$, we have 16 coefficients a_k and a variance $e(n)$. Thus, we reduce the data size from 200 samples to 17 samples, with a reduction ratio of about 12.

2.3.5 Speech Parameter Estimation

In this subsection, we introduce several methods to estimate the speech parameters such as voiced/unvoiced/silent speech detection, fundamental frequency (usually denoted as F0) detection, and formant estimation.

Voiced/Unvoiced/Silent Speech Detection

Voice activity detection is to detect speech segments and non-speech segments from a speech utterance (can be either clean or noisy speech). For clean speech, it is relatively easy for voice activity detection, e.g., using some energy-based method. However, for noisy speech, it is challenging. Therefore, we need to design sophisticated methods for voice activity detection.

There are different methods for voice activity detection, such as traditional feature-based and learning-based methods. For feature-based methods, we usually extract some features that can determine speech or non-speech segments. Some useful features for voice activity detection include: (1) Speech energy. When the signal-to-noise ratio (SNR) is large, we can assume that the energy of the voiced part is larger than that of the unvoiced part and larger than that of silence. However, when SNR is small, e.g., the energy of speech is similar to that of noise, we cannot tell apart it is speech (voiced or unvoiced) or non-speech (noisy). (2) Frequency or spectrum feature. We can apply a short-time Fourier transform to get the spectrogram features of speech and judge speech and non-speech segments accordingly. (3) Cepstrum feature. The cepstrum feature can determine the pitch of the speech, which can decide whether a segment is voiced speech or not. (4) Zero-crossing rate (ZCR), which measures the ratio between the number of times that the speech waveform is crossing the zero point and the number of total speech waveform points in a certain time (e.g., a speech frame). Voiced speech is produced by periodic vibration and usually has a low zero-crossing rate. Unvoiced speech is produced by the high-frequency noisy source signal and usually has a high zero-crossing rate. The zero-crossing rate of silent speech is usually lower than that of unvoiced speech. (5) Autocorrelation coefficient at unit sample delay, which calculates the correlation between adjacent speech samples. Since voiced speech usually has a low frequency, adjacent waveform points of voiced speech are highly correlated while those of unvoiced speech are weakly correlated.

After getting these features, we can judge voiced/unvoiced/silent speech either by some rules/thresholds or by training a machine learning model with labeled training data to infer voiced/unvoiced/silent speech. For example, we can simply use energy to separate voiced speech from unvoiced speech and silence and then use a zero-crossing rate to separate unvoiced speech from silence.

F0 Detection

After we determine whether a speech segment is voiced, unvoiced, or silent, we can further detect the pitch/F0 of the voiced speech. There are different methods to detect F0, and we introduce a few methods as follows: (1) Autocorrelation for F0 estimation. We shift the speech waveform sequence x by n points to get a new waveform sequence x^n, and we calculate the correlation between x and x^n. The period (inverse of frequency) of this speech waveform is the n which has the largest and peak correlation. (2) DIO algorithm [17] used in WORLD [18] vocoder. We

can first use different low-pass filters, each with a different cutoff frequency to get filtered waveforms. For each waveform, we can calculate four types of intervals: the interval between two adjacent positive/negative zero-crossing points, and the intervals between two adjacent peaks and valleys. If the four interval values are close to each other, then the average of the intervals is more likely to be the period of the waveform. (3) F0 detection based on cepstral analysis. We can first compute the cepstrum of each speech frame and search for the cepstrum peak in a reasonable range.[1] The peak can be usually found for a voiced speech, while not for an unvoiced speech.

Formant Estimation

Formant estimation refers to the estimation of the formant parameters such as resonant frequency and bandwidth. A typical method to estimate formant parameters is to use a low-pass lifter (inverse of filter) to filter the cepstrum to get the spectral envelope, and further locate the maximal value to get the resonant frequency and also the bandwidth.

2.3.6 Overview of Speech Processing Tasks

In the previous subsections, we have introduced some basic knowledge and methods in speech processing. Here we give a brief overview of some typical speech processing tasks, as shown in Fig. 2.9. There are some basic speech processing tasks, such as speech recognition, speaker recognition, speech synthesis, speech enhancement, speech separation, speech translation, etc. For speech synthesis, there are also several subtasks, such as text-to-speech synthesis, singing voice synthesis, voice conversion, EEG (electroencephalography) based speech synthesis, etc. The difference among these subtasks is the source input (e.g., text, music score, speech, or EEG signal). However, other subtasks such as singing voice synthesis, voice conversion, and EEG-based speech synthesis share some or most methodologies with text-to-speech synthesis. Thus, in this book, we mainly focus on text-to-speech synthesis.

[1] Usually we can assume that F0 is in the range of 80–450 Hz, then the quefrency peak in cepstrum should lie in a quefrency (inverse of frequency) of 2.2–12.5 ms.

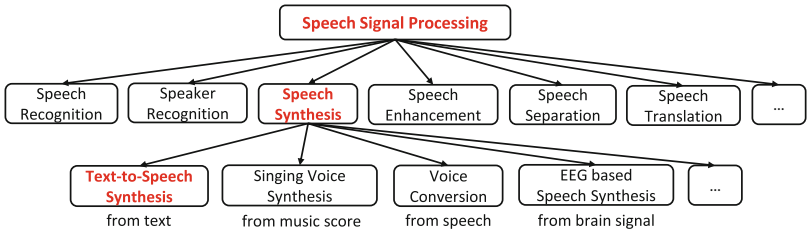


Fig. 2.9 Overview of speech processing tasks

References

1. Manning C, Schutze H (1999) Foundations of statistical natural language processing. MIT Press
2. Dance FEX, Larson C (1985) The functions of human communication. Inf Behav 1(1):62–75
3. Akmajian A, Farmer AK, Bickmore L, Demers RA, Harnish RM (2017) Linguistics: An introduction to language and communication. MIT Press
4. Crystal D (2011) A dictionary of linguistics and phonetics. John Wiley & Sons
5. Stevens KN (1997) Articulatory-acoustic-auditory relationships. Handbook of phonetic sciences. Oxford
6. Lass R (1984) Phonology: an introduction to basic concepts. Cambridge University Press
7. Blumstein SE (1991) The relation between phonetics and phonology. Phonetica 48(2–4):108–119
8. Borer H (2017) Morphology and syntax. The handbook of morphology, pp 149–190
9. Cruse A (2006) Glossary of semantics and pragmatics. Edinburgh University Press
10. Denes PB, Denes P, Pinson E (1993) The speech chain. Macmillan
11. Denes PB, Pinson EN (2015) The speech chain: the physics and biology of spoken language. Waveland Press
12. Makhoul J (1975) Linear prediction: A tutorial review. Proc IEEE 63(4):561–580
13. Wang X, Takaki S, Yamagishi J (2019) Neural source-filter waveform models for statistical parametric speech synthesis. IEEE/ACM Trans Audio Speech Lang Process 28:402–415
14. Cernak M, Rusko M (2005) An evaluation of synthetic speech using the PESQ measure. In Proc. European congress on acoustics, pp 2725–2728
15. Taal CH, Hendriks RC, Heusdens R, Jensen J (2011) An algorithm for intelligibility prediction of time-frequency weighted noisy speech. IEEE Trans Audio Speech Lang Process 19(7):2125–2136
16. Shannon CE (1949) Communication in the presence of noise. Proc IRE 37(1):10–21
17. Morise M, Kawahara H, Katayose H (2009) Fast and reliable f0 estimation method based on the period extraction of vocal fold vibration of singing voice and speech. In Audio engineering society conference: 35th international conference: audio for games. Audio Engineering Society
18. Morise M, Yokomori F, Ozawa K (2016) WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. IEICE Trans Inf Syst 99(7):1877–1884

Chapter 3
Basics of Deep Learning

Abstract This chapter introduces some basics of deep learning, including some learning paradigms and key components of machine learning, and some representative model structures and frameworks in deep learning. Since TTS is a typical data generation task, we also introduce some representative deep generative models.

Keywords Machine learning · Deep learning · Deep generative model

This chapter introduces some basics of deep learning [1], including some learning paradigms and key components of machine learning [2], and some representative model structures and frameworks in deep learning. Since TTS is a typical data generation task, we also introduce some representative deep generative models.

Prerequisite Knowledge for Reading This Chapter
- Basic knowledge of algebra, calculus, and probability theory.
- Basic knowledge of machine learning and deep learning.

3.1 Machine Learning Basics

Since deep learning is a branch of machine learning, we first introduce some basics of machine learning including learning paradigms and key components of machine learning [3].

3.1.1 Learning Paradigms

Machine learning covers different learning paradigms. The most fundamental learning paradigms are supervised learning, unsupervised learning, and reinforcement learning.

X. Tan, *Neural Text-to-Speech Synthesis*, Artificial Intelligence: Foundations, Theory, and Algorithms, https://doi.org/10.1007/978-981-99-0827-1_3

Supervised Learning

Supervised learning aims to learn a mapping function (the optimal solution in the hypothesis space) that converts the input to the output based on a set of training examples, where a training example is a pair of input and output. A supervised learning algorithm analyzes and detects the underlying patterns and relationships between the input and output of the training examples, and can generate output for unseen examples. Supervised learning is one of the most basic learning paradigms, and is widely used for regression and classification. It is easy to understand, train, and control what to learn by providing the training inputs and outputs. Some disadvantages of supervised learning include it is expensive and time-consuming to collect labeled training data, and it is difficult to learn information beyond the training pairs.

Unsupervised Learning

Different from supervised learning which needs input and output pairs for training, unsupervised learning analyzes and discoveries data patterns from purely unlabeled data. Typical unsupervised learning approaches include clustering [4], anomaly detection [5], dimensionality reduction [6], latent variable models [7], etc. A benefit of unsupervised learning is that it does not rely on labeled data that is costly to collect. Some advantages of unsupervised learning include it is inaccurate and time-consuming for model learning.

Reinforcement Learning

Different from supervised learning which requires labeled input and output data pairs for training or unsupervised learning that purely relies on unlabeled data to learn data patterns, reinforcement learning regards the feedback from the environment as the learning signal [8]. It is usually used in scenarios where an intelligent agent takes action in an environment to maximize its cumulative reward. Instead of finding an exact solution to map the input to the output as in supervised learning, reinforcement learning focuses on finding a balance between exploration (of uncharted territory) and exploitation (of current knowledge). Reinforcement learning is widely used in games, self-driving cars, finance, business management, and scenarios where exact labeled data is hard to get but feedback signal is easy to get.

Beyond the three basic learning paradigms, there are a lot of learning paradigms that are practical when applying machine learning in different scenarios, including semi-supervised learning, self-supervised learning, pre-training/fine-tuning, transfer learning, curriculum learning, and active learning, etc.

Semi-supervised Learning

Semi-supervised learning lies in between supervised learning and unsupervised learning. It usually combines a small amount of labeled data with a large amount of unlabeled data for training. In practice, it first trains a model on the labeled data and then uses this model to predict the target of the unlabeled data. The unlabeled data with pseudo labels are combined with the labeled data to train the model. This process can be repeated several times for better learning accuracy.

Self-supervised Learning

Self-supervised learning is to learn from unlabeled data by constructing labels from the unlabeled data itself. It belongs to supervised learning since it relies on input-output data pairs for training. It can be also regarded as unsupervised learning since it does not need explicit data labels. However, it is also different from unsupervised learning since that learning is not achieved using inherent data patterns. Self-supervised learning has been widely used in natural language processing, speech, and computer vision in recent years, such as language modeling, masked language modeling, masked auto-encoder, contrastive learning, etc. Some prominent work include BERT [9], GPT [10–12], MASS [13] in natural language processing, and Wav2Vec [14, 15] in speech processing, and SimCLR [16] and MAE [17] in computer vision.

Pre-training/Fine-Tuning

Pre-training and fine-tuning is a learning paradigm that first pre-trains a model with pretext tasks on large-scale datasets, and then fine-tunes the model on downstream tasks. Both supervised learning and self-supervised learning can be used in the pre-training stage. Most self-supervised learning methods introduced in the previous paragraph adopt the pre-training and fine-tuning paradigm.

Transfer Learning

Transfer learning aims to reuse or transfer the knowledge learned in previous tasks/problems for learning or solving different but related tasks/problems, which can improve the efficiency of training data. Pre-training and fine-tuning are also regarded as a type of transfer learning.

For other learning paradigms such as curriculum learning [18], active learning [19] or lifelong learning [20], we skip the introduction here and readers can find more details from some related machine learning materials.

As text-to-speech synthesis is a typical machine learning task, most of the learning paradigms are applied to this task to enable speech synthesis in multi-lingual,

multi-speaker, and multi-domain scenarios. We will cover some popular learning paradigms such as supervised learning, unsupervised learning, semi-supervised learning, self-supervised learning, pre-training/fine-tuning, transfer learning, and curriculum learning in later chapters.

3.1.2 Key Components of Machine Learning

Basically speaking, there are three key ingredients in machine learning: model, strategy, and algorithm [21]. The model is defined as a hypothesis space that covers all possible mappings from input to output, where this hypothesis space is usually constructed based on human prior. Machine learning methods try to find an optimal solution in this hypothesis space (i.e., to find the optimal parameters of the model). The strategy is a criterion or a method (e.g., a loss function) to judge whether a solution is optimal in this hypothesis space (i.e., whether the model parameters are optimal). Since the strategy can usually convert the goal of machine learning into solving the minimum of a loss function, the algorithm is a method (e.g., gradient descent) to get the minimum of the loss function.

We take logistic regression [22] as an example to illustrate the three key ingredients of machine learning. The model of logistic regression maps the input x into output $f(x) = \frac{1}{1+e^{-wx}}$, where w is the model parameter. Denote y as the label of the input x. We choose cross-entropy loss function as the criterion (i.e., strategy): $\mathcal{L}(x, y; w) = y \log \frac{1}{1+e^{-wx}} + (1-y)\log(1 - \frac{1}{1+e^{-wx}})$. Finally, we can use stochastic gradient descent (SGD) [23] as the algorithm to find the minimum of the cross-entropy loss function.

3.2 Deep Learning Basics

As a branch of machine learning, deep learning leverages deep neural networks to learn good representations for data understanding or learn good data distributions for data generation. Deep learning adopts similar learning paradigms and components in machine learning. We further introduce some distinctive model structures and frameworks in deep learning.

3.2.1 Model Structures: DNN/CNN/RNN/Self-attention

Deep learning has a more complicated hypothesis space with deeper and wider models than traditional machine learning. How to construct the model to form a good hypothesis space is important in deep learning. We introduce several typical building blocks for models in deep learning: dense neural network (DNN), convo-

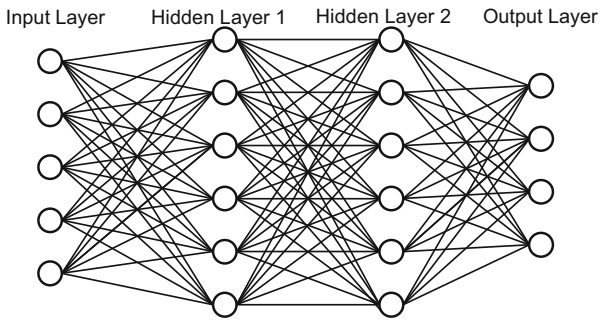


Fig. 3.1 DNN

lutional neural network (CNN), recurrent neural network (RNN), and Transformer (i.e., self-attention based networks).

DNN

A dense neural network (DNN), also called a fully connected network (FCN), consists of multiple densely connected layers, or multi-layer perceptron, as shown in Fig. 3.1. Each node (neuron) i in layer $l+1$ connects to each node j in layer l with a weight w_{ij}, and then followed with an activation function, such as tanh, sigmoid, ReLU and its variants.

CNN

A convolutional neural network (CNN) consists of convolution kernels or filters that slide along input features with shared weights and provide responses (i.e., feature maps), as shown in Fig. 3.2a. CNN is inspired by the biological process in visual perception that the connectivity pattern between neurons in CNN resembles the organization of the animal visual cortex, where each cortical neuron only responds to the stimuli in a restricted region of the visual field (i.e., receptive field). CNN can be regarded as a regularized version of DNN (multi-layer perceptron), by restricting each node i in layer $l+1$ to only connect to neighboring nodes in layer l.

RNN

In a recurrent neural network (RNN), there is a concept of "memory" to store the states or information of previous inputs and influence the generation of current output. Recurrent neural networks are used to process temporal sequences, where the data in a time step depends on the data points in previous time steps, as shown

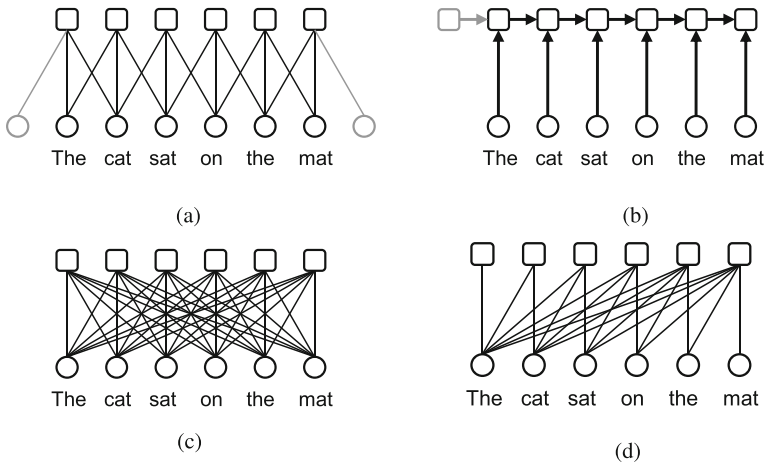


Fig. 3.2 Illustration of different model structures [25–27] when taking a text sequence as input. (**a**) Convolution. (**b**) Recurrence. (**c**) Self-attention. (**d**) Causal self-attention

in Fig. 3.2b. Since vanilla RNNs are prone to gradient vanishing or exploding problems, a lot of variants such as GRU (gated recurrent unit) [24] and LSTM (long short-term memory) [25] are proposed to solve this problem.

Self-attention

Different from CNN organizes the neuron connections in a local region with shared weights, or RNN organizes the neuron connections in a recurrent way, self-attention proposed in Transformer [26] (as shown in Fig. 3.2c) organizes the connections with a self-attention mechanism based on similarity measurement. After the cross-position information aggregations through self-attention, a feed-forward network is leveraged for position-wise information processing. When self-attention is used for autoregressive sequence generation, a causal attention mask is usually used to force the model to only attend to hidden states in previous steps. Thus, this attention mechanism is called causal self-attention, as shown in Fig. 3.2d. Self-attention as well as Transformer has been widely used as the backbone models in natural language processing, speech, and computer vision.

Comparison Between Different Structures

We compare DNN, CNN, RNN, and self-attention in Table 3.1 in terms of several characteristics: (1) computation complexity: the computation cost for each model structure; (2) sequential operations: how many sequential operations are needed to process a sequence; (3) maximal path length: the maximal length (in terms of

Table 3.1 Some statistics between DNN, CNN, RNN, and self-attention

Type	Computation complexity	Sequential operations	Maximal path length
DNN	$O(n^2 \cdot d^2)$	$O(1)$	$O(1)$
CNN	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(\log_k(n))$
RNN	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Self-attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$

the number of tokens) of the path that connects any two tokens in a sequence. We assume the sequence has n tokens and the model hidden dimension is h. As can be seen from Table 3.1, DNN has the largest computation cost (n^2d^2) since it uses dense connections between different tokens and different neurons to calculate the hidden of each token. CNN and RNN have the computation proportional to the sequence length. Self-attention has a computation quadratic to the sequence length.[1] In terms of sequential operation, RNN has sequential operations of n while other structures have parallel computation. The maximal path length captures the interaction efficiency between any two tokens in a sequence. For DNN, since it has dense connections between any two tokens, the maximal path length is 1. For CNN, if the kernel size is k, we need to stack $\log_k(n)$ layers (in the case of dilated convolutions [28, 29]) to reach a receptive field of n tokens, i.e., two farthest tokens can be processed in a convolutional kernel. Thus, the maximal path length is $\log_k(n)$. Due to the sequential operations in RNN, the maximal path length is n. In self-attention, since any two tokens can be connected through the self-attention mechanism, the maximal path length is 1. As can be seen, self-attention has advantages over other model structures in either computation complexity, sequential operations, or maximal path length.

3.2.2 Model Frameworks: Encoder/Decoder/Encoder-Decoder

Since speech synthesis is a typical sequence processing task, we introduce some typical frameworks for sequence modeling. Generally speaking, there are three kinds of sequence tasks: (1) many-to-one, where a sequence is mapped to a single label, such as sequence classification; (2) one-to-many, where a sequence is generated based on a single label, such as conditional sequence generation; (3) many-to-many, where an input sequence is converted into another sequence, either in the same length (e.g., sequence tagging), or different lengths (e.g., machine translation, text summarization, speech-to-text recognition, text-to-speech synthesis). Based on this categorization, we introduce the corresponding framework for each kind of

[1] When calculating the computation of self-attention, we omit the computation of the matrix multiplication by W_Q, W_K, W_V, and W_O in self-attention and just measure the Q-K-V mechanism. We also omit the feed-forward networks after self-attention.

sequence task: Encoder, Decoder, and Encoder-Decoder. Although we categorize different model frameworks for different sequence tasks, these frameworks are model structure agnostic, which means each framework can be implemented with DNN, RNN, CNN, or self-attention.

Encoder

An encoder usually takes a sequence of tokens as input and outputs a sequence of hidden representations. After that, these hidden representations are further used for classification or regression. Early works use RNN or CNN as the basic model structures for encoders, while recently Transformer with self-attention has been a popular structure for encoders. The most prominent one is BERT [9], which pre-trains a Transformer encoder on large-scale language corpus and fine-tunes it in downstream tasks to achieve state-of-the-art performance on sequence classification and language understanding. To enable classification or prediction, BERT introduces a special "CLS" token, which aggregates information across different token positions and is used as the prediction head in downstream tasks.

Decoder

A decoder usually models sequence with a next token prediction task.[2] It is usually trained in a teacher-forcing way, where the previous ground-truth tokens are taken as the input to generate the current token. When generating the current token in inference, it takes the previously generated token as input, i.e., in an autoregressive way. An example is GPT [10–12], which pre-trains on large-scale language corpus and can be used for language modeling and generation. This autoregressive generation would cause error propagation, where the errors that occurred in previous tokens would affect the generation of later tokens.

Encoder-Decoder

An encoder-decoder framework is usually used for sequence-to-sequence tasks, where the encoder takes the source sequence as input and outputs a sequence of hidden representations, and the decoder generates target tokens conditioned on source hidden representations. In the early encoder-decoder framework, the decoder only takes the last hidden of source representations as condition [30, 31], while

[2] There are different ways to model a sequence in the decoder, such as autoregressive generation, non-autoregressive generation, or iterative (e.g., normalizing flow or diffusion model as introduced in Sect. 3.3) ways. Here we briefly describe the traditional autoregressive way and leave other advanced methods to later sections.

later attention mechanism [32] is introduced into the encoder-decoder framework to extract information from source hidden representations for target token generation.

3.3 Deep Generative Models

Intuitively speaking, a generative model describes how data is generated by capturing the probability of data through statistical learning. Then new data can be generated by sampling from the probability distribution learned by this model. A deep generative model follows the basic definition of generative models, is parameterized by a deep neural network θ, and is trained to maximize its likelihood given the training data D. In this way, it learns a data distribution P_θ that is similar to the true data distribution P_D and can generate new data from the learned data distribution P_θ. Therefore, the key problem in deep generative models is probability/density estimation: to estimate an unobservable underlying probability/density function based on observed data. After getting a good density estimation, we can sample new data points (i.e., data generation), predict the density/probability of a data point (i.e., density estimation), etc.

Depending on the types of density learned in model training, deep generative models can be divided into different categories, as shown in Fig. 3.3.

- Exact density. They estimate data density by explicitly defining the density and designing specific model structures to compute it in a tractable way, such as autoregressive models and normalizing flows [33–37].
- Approximate density. They model the tractable approximations of the data density, such as variational auto-encoders [38] and denoising diffusion probabilistic models [39, 40].
- Unnormalized density. They learn the unnormalized density due to the intractability of the partition function, such as energy-based models [41].

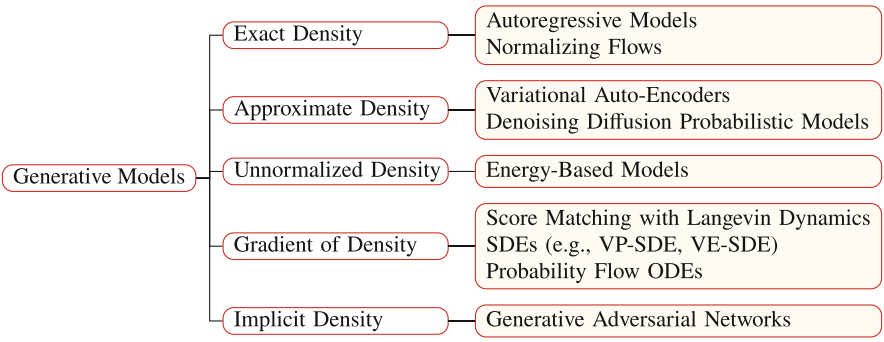


Fig. 3.3 A taxonomy of generative models according to the ways of density estimation

- Gradient of density. They estimate the gradient of the log probability density and use some methods such as Langevin dynamics or differential equations to generate data, such as score matching with Langevin dynamics [42, 43], stochastic differential equations (SDEs) [43], and probability flow ordinary differential equations (ODEs) [43, 44].
- Implicit density. They learn the data density by comparing it with the true data density without explicitly defining it, such as generative adversarial networks [45].

Text-to-speech synthesis is a kind of data generation task, which can leverage deep generative models for better speech generation. Therefore, in this section, we introduce some representative generative models that are commonly used in text-to-speech synthesis, including autoregressive models (AR), normalizing flows (Flow), variational auto-encoders (VAE), denoising diffusion probabilistic models (Diffusion), score matching with Langevin Dynamics (SMLD), SDE/ODE, and generative adversarial networks (GAN), etc.

3.3.1 Autoregressive Models

The term "autoregressive" comes from the term "autoregression", which means it is a regression of the sequence element against itself. For example, a linear autoregressive model of order p is formulated as $y_t = c + \Sigma_{i=1}^{p} w_i y_{t-i} + \epsilon_t$, where $w_i, i \in [1, p]$ are the model parameters, c is a constant, and ϵ_t is white noise. In deep autoregressive models, w_i is implemented with non-linear deep neural networks, such as CNN, RNN, or self-attention.

Autoregressive models can estimate the probability of data x in an exact way, by assuming each output in the current step depends on the data in the previous time steps and using the chain rule to decompose the probability of a sequence data x into a product of conditional probability in each step:

$$P(x) = \Sigma_{i=1}^{n} P(x_i | x_{<i}), \tag{3.1}$$

where $x_{<i}$ represents the elements before position i in sequence x, and n is the sequence length.

Autoregressive models are widely used in data generation, such as the RNN language model and GPT [10–12] for text generation, WaveNet [28] for speech synthesis, and PixelRNN [46]/PixelCNN [47] for image generation. Specifically, WaveNet [28] is the first neural-based vocoder for speech synthesis, which leverages dilated convolution to generate waveform points autoregressively.

By regarding the data sample as a sequence of data points and modeling the dependency among data points in an autoregressive way, autoregressive models can better learn the complex internal structure dependency in data and provide exact data probability estimation. A side effect is that they suffer from slow generation speed

due to autoregressive modeling. Therefore, a lot of generative models are proposed to speed up the generation process by removing the autoregressive dependency among data points, such as normalizing flows, variational auto-encoders, and generative adversarial networks. Among them, normalizing flows can still provide the exact probability estimation as in autoregressive models.

3.3.2 Normalizing Flows

Normalizing flows [33–37] are a kind of generative models that transform a data point from a standard distribution into a data point following complex data distribution with a sequence of invertible mapping functions [35]: $x = f_0 \circ f_1 \circ ... f_k(z)$, where $z \sim \mathcal{N}(0, 1)$ is a data point from a standard Gaussian distribution, and x is a data point that follows the distribution that we want to generate, f_i where $i \in [1, k]$ denotes the invertible mapping function. Since we can get a normalized (standard) probability distribution (e.g., Gaussian) from the data distribution through a flow (sequence) of invertible mapping functions: $z = f_k^{-1} \circ f_{k-1}^{-1} \circ ... f_0^{-1}(x)$, this kind of flow-based generative models is called as normalizing flows.

We can train a flow-based model by maximizing the log-likelihood of its model parameters given the data based on the rules of change of variables:

$$\log p(x) = \log p(z) + \log \det(dz/dx) = \log p(z) + \Sigma_{i=1}^{k} \log |\det(J(f_i^{-1}(x)))|, \tag{3.2}$$

where J denotes the Jacobian matrix of $f_i^{-1}(x)$, and $\det(\cdot)$ is the determinant of the matrix. Therefore, normalizing flows can estimate the data probability in an exact way, as in autoregressive models.

To maximize the log-likelihood in an easy way based on Eq. 3.2, the transformation function f should satisfy two requirements: (1) it is easily invertible; (2) its Jacobian determinant is easy to compute.[3]

To reduce the complexity of Jacobian determinant, previous works either carefully design the model architectures with low-rank (e.g., Planar NF [35], Sylvester NF [48]), coupling (NICE [33], RealNVP [34], Glow [37]), or autoregressive (e.g., inverse AF [36], neural AF [49], Masked AF [50]) technologies, or design stochastic estimator of free-form Jacobian (e.g., FFJORD [51], Residual Flows [52]). We mainly introduce coupling (bipartite) and autoregressive technologies, which can ensure the invertible functions have triangular Jacobians so that the determinant can be easily calculated from the diagonal elements. Bipartite transformation and autoregressive transformation use two different granularities to split data x for dependency modeling: In autoregressive transformation, x is split into each step

[3] Since it usually takes $O(n^3)$ to compute the determinant where n is the dimension of x, it is not scalable in high dimension.

Table 3.2 Several representative flow-based models and their formulations [53]

Flow		Evaluation $z = f^{-1}(x)$	Synthesis $x = f(z)$
AR	AF [50]	$z_t = \frac{x_t - \mu_t(x_{<t})}{\sigma_t(x_{<t})}$	$x_t = z_t \cdot \sigma_t(x_{<t}) + \mu_t(x_{<t})$
	IAF [36]	$z_t = x_t \cdot \sigma_t(z_{<t}) + \mu_t(z_{<t})$	$x_t = \frac{z_t - \mu_t(z_{<t})}{\sigma_t(z_{<t})}$
Bipartite	RealNVP [34]	$z_a = x_a,$	$x_a = z_a,$
	Glow [37]	$z_b = x_b \cdot \sigma_b(x_a; \theta) + \mu_b(x_a; \theta)$	$x_b = \frac{z_b - \mu_b(x_a;\theta)}{\sigma_b(x_a;\theta)}$

x_t where $t \in [1, n]$ and n is the length of x, while in bipartite transformation, x is split into two parts x_a and x_b. We introduce different normalized flows based on these transformation functions, as shown in Table 3.2.

Based on autoregressive transformations, there are two types of flows: autoregressive flow (AF) and inverse autoregressive flow (IAF). AF can be regarded as a basic autoregressive model parameterized with a single Gaussian:

$$x_t = z_t \cdot \sigma_t(x_{<t}) + \mu_t(x_{<t}), \tag{3.3}$$

where $\mu_t(x_{<t})$ and $\sigma_t(x_{<t})$ are the shifting and scaling variables modeled by autoregressive models like PixelCNN [47] or WaveNet [28]. The inverse mapping is

$$z_t = \frac{x_t - \mu_t(x_{<t})}{\sigma_t(x_{<t})}. \tag{3.4}$$

Note that the training (evaluating z) can be parallel since z_t does not depend on $z_{<t}$, but the inference (generating x) is autoregressive since x_t depends on $x_{<t}$. Instead of modeling the generation of x in an autoregressive manner like in AF, IAF models the generation of x in parallel but models the evaluation of z in an autoregressive manner:

$$x_t = \frac{z_t - \mu_t(z_{<t})}{\sigma_t(z_{<t})}, \quad z_t = x_t \cdot \sigma_t(z_{<t}) + \mu_t(z_{<t}). \tag{3.5}$$

IAF can be regarded as a dual formulation of autoregressive flow (AF) [49, 50]. The training of AF is parallel while the sampling is sequential. In contrast, the sampling in IAF is parallel while the inference for probability estimation is sequential.

Based on bipartite transformations, NICE [33], RealNVP [34], and Glow [37] leverage coupling layer to ensure the output can be computed from the input and vice versa. The inverse mapping is:

$$z_a = x_a, \quad z_b = x_b \cdot \sigma_b(z_a) + \mu_b(z_a), \tag{3.6}$$

where $\mu_b(z_a)$ and $\sigma_b(z_a)$ are the shifting and scaling variables modeled by feed-forward networks. The forward mapping to generate x from z is:

$$x_a = z_a, \quad x_b = \frac{z_b - \mu_b(z_a)}{\sigma_b(z_a)}. \tag{3.7}$$

Different from autoregressive transformation-based flows, bipartite transformation-based flows can enable parallel computation in both the training (evaluating z) and inference (generating x).

There are some variations in bipartite transformation-based flows. Some works only use additive coupling layers, where the scaling term $\sigma_b(z_a)$ is set to 1, i.e., only additive term for shifting but no scaling, like that in NICE [33]. Some works use affine coupling layers, where the $\sigma_b(z_a)$ is usually not 1. In one coupling layer, some dimensions of x (i.e., x_a) are unchanged. If these dimensions are always unchanged in all the coupling layers, it will greatly limit the model capacity. Thus, the order of dimension of x is reversed in each coupling layer, to ensure that all the dimensions have the chance to be changed. In Glow [37], this reverse operation is replaced by an invertible $1 * 1$ convolution, which can be regarded as a generalization of any permutation of the dimension order.

Both autoregressive and bipartite transforms have their advantages and disadvantages [53]: (1) Autoregressive transforms are more expressive than bipartite transforms in modeling dependency between data distribution x and standard probability distribution z but require teacher distillation that is complicated in training. (2) Bipartite transforms enjoy a much simpler training pipeline, but usually require a larger number of parameters (e.g., deeper layers, larger hidden size) to reach comparable capacity with autoregressive transforms.

Generally speaking, normalizing flows have several advantages: (1) The training process is very stable and much easier to converge, as simple as autoregressive models, unlike other generative models (e.g., VAEs and GANs) that require careful tuning. (2) Normalizing flows can estimate the data density in an exact way. Some disadvantages of normalizing flows include: (1) the model expressiveness is limited due to the requirement of invertible (bijective) mapping; (2) the bijective mapping requires a high dimensional latent space, which is usually difficult to interpret.

3.3.3 Variational Auto-encoders

Variational auto-encoders (VAEs) [38] are a kind of generative model that compress the input data into a constrained/regularized multivariate latent distribution through an encoder and reconstruct the input data as accurately as possible through a decoder. The latent distribution is regularized to some prior distribution during training to ensure that it is easy to sample from the latent space to generate new data. The term "variational" means that there is some relationship between the regularization and variational inference [54].

We first explain why we need regularization in VAEs by discussing naive auto-encoder. In auto-encoder, the encoder converts the data into latent representations, and the decoder converts the latent representations back into data, and

the encoder and decoder are trained by minimizing the reconstruction error, e.g., $\|x - \text{dec}(\text{enc}(x))\|^2$. When using an auto-encoder for data generation, we usually sample a random point from the latent space and decode it to obtain new data. Since no regularization on the latent space of the auto-encoder and the high complexity of the encoder and decoder in the auto-encoder, there would be overfitting, which means that some points in the latent space (corresponding to the training data) are highly optimized with low reconstruction loss, while other points (not covered in training) will correspond to high reconstruction loss with meaningless data. As a consequence, the latent space is extremely irregular and non-smoothing, which means some points that are close to each other in the latent space can decode to data points that are very different in the data space, and some points in the latent space can decode to data points that are meaningless in the data space. Thus, to enable data generation with good generalization, we need to regularize the latent space.

A VAE also consists of an encoder and a decoder similar to an auto-encoder and is trained to minimize the reconstruction error. However, different from the auto-encoder that converts a data input into a single point in the latent space, the VAE encoder converts a data input x into a distribution $p(z|x)$ over the latent space, where a point is sampled from this distribution ($z \sim p(z|x)$) and decoded into a data output by the VAE decoder. We usually add a regularization term on the latent space to ensure the distribution encoded by the encoder to facilitate data generation. Thus, the loss function of VAEs is as follows:

$$L = \|x - \text{dec}(z)\|^2 + KL(\mathcal{N}(\mu_x, \sigma_x)||\mathcal{N}(0, 1)), \tag{3.8}$$

where $z \sim N(\mu_x, \sigma_x)$ and $\mu_x = \text{enc}_\mu(x), \sigma_x = \text{enc}_\sigma(x)$. Generally speaking, the regularization should satisfy two properties: continuity and completeness. Continuity means two points that are close to each other in the latent space should be decoded into data in the data space that are also close to each other. Completeness means a point sampled from the distribution of the latent space should be decoded into meaningful data in the data space. We regularize both the mean and covariance matrix of the distributions generated by the encoder to be close to those of a standard Gaussian distribution. In this way, by constraining the mean to be close to 0, we can ensure the distribution is near to each other, and by constraining the covariance matrix to be close to the identity, we can ensure a smooth distribution instead of a punctual distribution. In this way, we encourage the encoder to encode data to be close to and overlapped with each other in the latent space, better satisfying the continuity and completeness conditions.

We can also interpret the loss formulation of VAEs from the perspective of variational inference. Considering the data generation is depending on some random process involved by random variable z, we first sample z from $p(z)$ and then sample x from $p(x|z;\theta)$, where θ and z are unknown to us. The log-likelihood of its model

parameter θ given data x can be formulated as

$$\begin{aligned}\log p(x) &= \log \int p(x|z)p(z)dz = \log \int q(z|x)\frac{p(x|z)p(z)}{q(z|x)}dz \\ &= \log \mathbb{E}_{z\sim q(z|x)}\frac{p(x|z)p(z)}{q(z|x)} \geq \mathbb{E}_{z\sim q(z|x)} \log \frac{p(x|z)p(z)}{q(z|x)} \\ &= \mathbb{E}_{z\sim q(z|x)} \log p(x|z) - KL(q(z|x)||p(z)),\end{aligned} \tag{3.9}$$

where the $\geq$ in the above equation is derived based on Jensen inequality [55]. To maximize the log-likelihood $\log p(x)$, instead, we can maximize its lower bound $\mathbb{E}_{z\sim q(z|x)} \log p(x|z) - KL(q(z|x)||p(z))$. Thus, the loss function becomes

$$L(x;\theta,\phi) = -\mathbb{E}_{z\sim q(z|x;\phi)} \log p(x|z;\theta) + KL(q(z|x;\phi)||p(z)) \tag{3.10}$$

VAEs have some advantages: (1) the latent code is generated by data x itself, so the correspondence between latent z and data x in the VAE decoder is stronger; (2) the latent space is regularized with continuity and completeness, so the generated data is more regular than auto-encoders. However, since VAEs use point-wise loss without global information, it is blurry with lower quality. VAEs can only obtain the lower bound of the log-likelihood, not like autoregressive models and normalizing flows that can obtain exact likelihood.

3.3.4 Denoising Diffusion Probabilistic Models

Denoising diffusion probabilistic models (DDPMs or Diffusion Models for short) [39, 40] are a kind of generative model that consists of two processes: a forward process and a backward process. The forward process is a Markov chain that transforms the data x_0 into a prior x_T (usually standard Gaussian) by iteratively injecting Gaussian noise into x_0 according to a pre-defined noise schedule β with $0 < \beta_1 < ... < \beta_T < 1$, as follows:

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}), \quad q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I), \tag{3.11}$$

where $q(x_t|x_{t-1})$ denotes the transition probability at time step t. We can obtain the noisy distribution of x_t by $q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$, where $\alpha_t := 1-\beta_t$, and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$ represents the noise level at time step t. $q(x_T|x_0)$ converges to standard Gaussian distribution $\mathcal{N}(x_T; 0, I)$ if $\bar{\alpha}_T$ is small enough. The reverse

process gradually transforms the prior noise from $x_T \sim \mathcal{N}(0, I)$ to data x_0 through

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \quad p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{3.12}$$

where $p(x_T) = \mathcal{N}(x_T; 0, I)$, and $p_\theta(x_{t-1}|x_t)$ is the transition probability in each reverse step, parameterized by μ_θ and Σ_θ. The network θ is trained by maximizing the evidence lower bound (ELBO) of the likelihood of $p_\theta(x_0)$. To deduce the ELBO, we can first write the log-likelihood as

$$\begin{aligned} \log p(x_0) &= \log \int p(x_{0:T}) dx_{1:T} = \log \int q(x_{1:T}|x_0) \frac{p(x_{0:T})}{q(x_{1:T}|x_0)} dx_{1:T} \\ &= \log \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0))} \frac{p(x_{0:T})}{q(x_{1:T}|x_0)} \geq \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0))} \log \frac{p(x_{0:T})}{q(x_{1:T}|x_0)} \\ &= ELBO. \end{aligned} \tag{3.13}$$

Then we can obtain the ELBO as

$$\begin{aligned} ELBO &= \mathbb{E}_{x_{1:T} \sim q(x_{1:T}|x_0))} \log \frac{p(x_{0:T})}{q(x_{1:T}|x_0)} \\ &= -\mathbb{E}_q \Bigg[\underbrace{KL(q(x_T|x_0)||p(x_T))}_{L_T} + \sum_{t=2}^{T} \underbrace{KL(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))}_{L_{t-1}} \\ &\quad - \underbrace{\log p_\theta(x_0|x_1)}_{L_0} \Bigg]. \end{aligned} \tag{3.14}$$

Since the variance β_t is fixed to constants in the forward process, the posterior $q(x_T|x_0)$ has no learnable parameters and $p(x_T)$ is a standard Gaussian, and thus L_T is a constant during training and can be ignored. For the term L_{t-1}, we first rewrite the forward process posteriors $q(x_{t-1}|x_t, x_0)$ by Bayes rule as:

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I), \tag{3.15}$$

$$\tilde{\mu}_t(x_t, x_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} x_0 + \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} x_t, \quad \tilde{\beta}_t := \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t. \tag{3.16}$$

We usually set $\Sigma_\theta(x_t, t) = \tilde{\beta}_t I$ so that we do not need to learn the variance term. To learn the mean term $\mu_\theta(x_t, t)$, since $x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t(x_0, \epsilon) - \sqrt{1 - \bar{\alpha}_t}\epsilon)$, we can

rewrite $\tilde{\mu}_t(x_t, x_0)$ in above equation as $\frac{1}{\sqrt{\alpha_t}}(x_t(x_0, \epsilon) - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon)$. Instead of directly setting $\tilde{\mu}_t(x_t, x_0)$ as the learning target, we usually set the standard Gaussian noise ϵ as the learning target, parameterized by $\epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon), t)$. Then the ELBO becomes

$$-ELBO = C + \sum_{t=1}^{T} \mathbb{E}_{x_0,\epsilon}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|^2\right]. \tag{3.17}$$

Ho et al. [40] further demonstrate the effectiveness to drop the weighting factor in the above loss functions and use a simplified training objective as follows:

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t,x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right]. \tag{3.18}$$

As a summary, the training and inference procedure of diffusion models are shown in Algorithms 1 and 2.

Algorithm 1 Training

repeat
  Sample $x_0 \sim q_{data}, \epsilon \sim \mathcal{N}(0, I)$
  Sample $t \sim \mathcal{U}(\{1, \cdots, T\})$
  $\mathcal{L} = \|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|^2$
  Update θ with $\nabla_\theta\mathcal{L}$
until converged

Algorithm 2 Sampling

Sample $x_T \sim \mathcal{N}(0, I)$
for $t = T, T-1, \cdots, 1$ **do**
  Sample $z \sim \mathcal{N}(0, I)$ if $t > 1$; else $z = 0$
  $x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)) + \sigma_t z$
end for
return x_0

3.3.5 *Score Matching with Langevin Dynamics, SDEs, and ODEs*

Score matching with Langevin dynamics (SMLD) [42, 43] is similar to diffusion models in that both of them involve the process that corrupts data by gradually adding noises and generates the data by gradually removing the noises. The term "score" denotes the gradient of the log probability density with regard to the data.

SMLD estimates the score of the data at different noise levels with a learnable score function and leverages Langevin dynamics to generate data sequentially.

Similar to diffusion models, we first denote the noise perturbation (forward) process $q_\sigma(x_t|x_0)$ that transforms the data x_0 into x_t:

$$q_\sigma(x_t|x_0) = \mathcal{N}(x_t; x_0, \sigma_t^2 I), \tag{3.19}$$

where σ is a sequence of positive noise scales satisfying $\sigma_1 < ... < \sigma_t < ... < \sigma_T$. σ_1 is small enough so that $q_\sigma(x_1|x_0) \approx p(x_0)$ which is the data distribution, and σ_T is large enough so that $q_\sigma(x_T|x_0) \approx \mathcal{N}(0, \sigma_T^2 I)$.

We can obtain the score $\nabla_x \log q_\sigma(x_t|x_0)$ in training in an analytic form, and train a noise conditional score network [42] $s_\theta(x_t, \sigma_t)$ to match the score $\nabla_x \log q_\sigma(x_t|x_0)$ using the following loss function:

$$\arg\min_\theta \Sigma_{t=1}^T \sigma_t^2 \mathbb{E}_{p(x_0)} \mathbb{E}_{q_\sigma(x_t|x_0)} ||s_\theta(x_t, \sigma_t) - \nabla_x \log q_\sigma(x_t|x_0)||_2^2. \tag{3.20}$$

After obtaining the score function $s_\theta(x_t, \sigma_t)$, we can sample data using Langevin dynamics (Markov chain Monte Carlo) [42]:

$$x_t^m = x_t^{m-1} + \epsilon_t * s_\theta(x_t^{m-1}, \sigma_t) + \sqrt{2\epsilon_t} z_t^m, m = 1, 2, ..., M, \tag{3.21}$$

where ϵ_t is the step size and z_t^m is standard Gaussian. The sampling process is repeated from $t = N, N-1, ..., 1$, with $x_N^0 \sim \mathcal{N}(0, \sigma_T^2 I)$ and $x_t^0 = x_{t+1}^M$ when $t < N$. As $M \to \infty$ and $\epsilon_t \to 0$, $X_1^M \sim p(x_0)$.

We can further extend the discrete time steps to the continuous time variable t. In this way, we use $x(t)$ and $\sigma(t)$ to denote x_t and σ_t. $\sigma(t)$ has an infinite number of noise scales, and the distribution of the perturbed data $x(t) \sim q_\sigma(x(t)|x(0))$ evolves as a stochastic differential equation (SDE) [43]:

$$dx = f(x, t)dt + g(t)dw, \tag{3.22}$$

where w is the standard Brownian motion, $f(\cdot, t)$ is called the drift coefficient of $x(t)$, and $g(t)$ is the diffusion coefficient of $x(t)$. Similarly, we can obtain a corresponding SDE for the reverse/denoising process [43, 56]:

$$dx = [f(x, t) - g(t)^2 \nabla_x \log p_t(x)]dt + g(t)d\bar{w}, \tag{3.23}$$

where $\bar{w}$ is a standard Brownian motion as the time flows from T to 0.

According to [43], for all diffusion and denoising processes in stochastic versions as introduced above, there is a corresponding deterministic process whose diffusion and denoising trajectories have the same marginal probability densities $\{p_t(x)\}_{t=o}^T$. This deterministic process can be formulated as a probability flow

ordinary differential equation (ODE) [43]:

$$dx = [f(x,t) - \frac{1}{2}g(t)^2 \nabla_x \log p_t(x)]dt. \tag{3.24}$$

For the details of how to train the score function to estimate the score $\nabla_x \log p_t(x)$ and how to sample data according to the reverse SDE and ODE processes, readers can refer to [43].

3.3.6 Generative Adversarial Networks

Generative adversarial networks (GANs) [45] is a kind of generative model that generate data from a random vector sampled from a simple distribution, and are widely used in many data generation tasks, such as image generation [45, 57, 58], speech and audio generation [59–61], text generation [62], etc. Before introducing the mathematical formulation of GANs, we introduce the intuitive idea behind GANs.

Generating data can be regarded as a process to generate an N-dimensional vector that follows the distribution of the data. One way is to transform a simple N-dimensional random vector into this desired vector using a complex function, i.e., to generate a vector following a specific data distribution given a random vector. Due to the complexity of data distribution, we cannot explicitly express this transformation function, and instead, formulate it as a deep neural network (a generator) and learn it from data. The key is how to train the generator to generate data that matches the true data distribution. There are different methods to train the generators. One straightforward way is to compare the distribution generated by the generator and the true data distribution directly based on samples, and backpropagate the distribution matching error to adjust the weights of the generator. However, it is difficult to directly compare two probability distributions based on data samples. Therefore, some works propose indirect ways to match the two distributions, where GANs are one of the most prominent methods.

GANs introduce a discriminator upon the generator to form an adversarial game: the discriminator takes the generated data and true data and tries to classify them as well as possible; the generator tries to fool the discriminator as much as possible, i.e., generating data that can be classified as true by the discriminator. GANs do not estimate the data density explicitly like that in autoregressive models or normalizing flows. Instead, they learn the transformation parameterized by the generator from a simple distribution (e.g. standard Gaussian) to the training data distribution. During inference, they sample a random noise from the standard Gaussian distribution and use the generator to transform it into a data point in the training distribution. The

loss function of GANs can be formulated as follows

$$\min_{\theta}\max_{\phi}\mathbb{E}_{x\sim p_{\text{data}}}\log D(x;\phi)+\mathbb{E}_{x\sim p_z}\log(1-D(G(z;\theta);\phi)), \tag{3.25}$$

where θ and ϕ denote the parameter of generator and discriminator respectively, and p_{data} and p_z denote the true data distribution and standard Gaussian distribution.

3.3.7 Comparisons of Deep Generative Models

Data generation based on deep generative models can be regarded as a process to transform a random variable z from a simple prior distribution $p(z)$ (usually a simple distribution like standard Gaussian distribution) into the desired complex data x following a distribution of $p(x)$.[4] We illustrate different deep generative models to achieve this data transformation in Fig. 3.4. For example, GAN achieves this by its generator, VAE achieves this by its decoder, Flow achieves this by a series of inverse mapping $f_k^{-1}...f_1^{-1}$, and Diffusion achieves this by the iterative inverse process $x_T\rightarrow\cdots\rightarrow x_t\rightarrow\cdots\rightarrow x_0$.

We further explain why we need these sophisticated designs in different generative models. Let us first check a naive way to learn a generative model to achieve this data transformation: we can randomly sample a latent variable z from $p(z)$ and randomly pick a data x from $p(x)$, and try to optimize the generative model to align the two data by $x=f(z;\theta)$. However, this simple learning and optimization will never work since there is no consistent relationship between latent z and data x. For example, for a data point x optimized in different training iterations, different z may be sampled. A reasonable way is to assign each data point x with a consistent latent variable z that would not change over the training iterations. In this way, the generative model will try its best to learn (overfit) the mapping between z and x. However, since the random variables are randomly assigned to each data point, the learned generative model will not have a generalization ability. It is because we may sample two close latent variables z for two data points that are very different, or the latent variables are very different for two similar data points. To ensure the generative models can be optimized and generalized well, we need to associate the data point x to the random variable z in a consistent and generalized way. Thus, we need an inverse mapping function from data to latent $z=f(x;\phi)$.

We introduce how the inverse mapping function is implemented in different deep generative models. (1) In VAE, the encoder maps the data point x into latent variable $z\sim q(z|x;\phi)$ and reconstructs the data point x based on this z. To achieve data generation from latent variable z randomly sampled from prior

[4] Naive autoregressive models do not satisfy this process since they generate data in an autoregressive way, without taking simple prior distribution as input. However, a Gaussian version of autoregressive model ($x_t=z_t\cdot\sigma_t(x_{<t})+\mu_t(x_{<t})$) denoted in Eq. 3.3 satisfies this process.

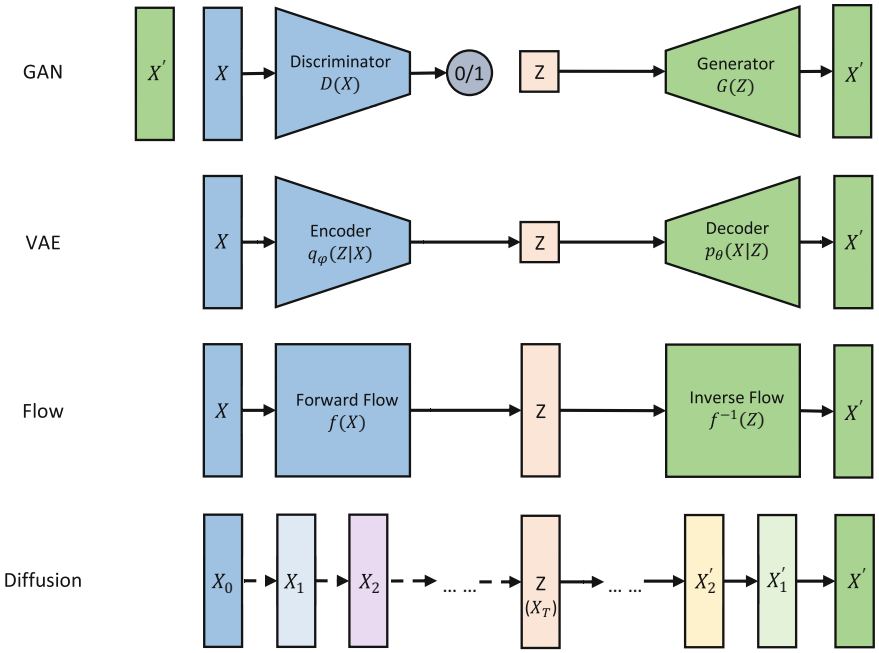


Fig. 3.4 Comparison among different deep generative models

distribution $p(z)$, the posterior distribution $q(z|x;\phi)$ is regularized by the prior distribution. (2) In Flow, the data x is transformed by the forward mapping $f(\cdot)$ into $z \sim p(z)$. Since the forward mapping function is invertible, the generation from z to x can be achieved by the inverse mapping $f^{-1}(\cdot)$. (3) In Diffusion, the data x is destroyed into random Gaussian noise z with a series of diffusion steps $q(x_t|x_{t-1})$. The data generation from z to x is achieved by the inverse/denoising process $p_\theta(x_{t-1}|x_t)$, which is optimized by maximizing the evidence lower bound of the log-likelihood. (4) GAN solves this problem from a very different perspective: its discriminator learns to distinguish the data generated by its generator from the real data, and the generator and discriminator play an adversarial game. In this way, GAN just randomly samples a latent variable z and guides the generation of x using a distribution loss with a discriminator, instead of a point-wise loss as in VAE. Thus, the adversarial game can automatically guarantee that the data point x and the random variable z are associated in a consistent and generalized way.

We list the characteristics of different kinds of generative models, as shown in Table 3.3.[5] We summarize as follows:

[5] The * in the table means there are some special situations and not always being Y or N for this item.

Table 3.3 Some characteristics of different deep generative models

Generative models	AR	Flow	VAE	Diffusion	SMLD	SDE	ODE	GAN
High-quality	Y	N	N	Y	Y	Y	Y	Y
Fast sampling	N	Y*	Y	N	N	N	N	Y
Mode diversity	Y	Y	Y	Y	Y	Y	Y	N
Density estimation	Y	Y	Y*	Y*	N	N	Y	N
Latent manipulation	N	Y	Y	Y*	Y*	Y*	Y*	Y*
Error propagation	Y	N*	N	Y	Y	Y	Y	N
Stable training	Y	Y	N*	Y	Y	Y	Y	N

- High-Quality Generation. Normalizing flows have to ensure the invertibility of the mapping function to enable exact likelihood estimation and data sampling, with a sacrifice of model capacity and thus sampling quality. VAEs usually adopt L1/L2 loss for data reconstruction and thus result in blurred generation results. Other generative models can usually generate high-quality samples.
- Fast Sampling. Autoregressive models generate samples token by token, which suffer from slow generation speed. Other iterative-based models such as diffusion models, score matching with Langevin dynamics (SMLD), SDEs, and ODEs also suffer from slow inference speed due to multiple time steps used to guarantee high sampling quality. VAEs and GANs support parallel/non-autoregressive generation, with fast sampling speed. For flow-based models, they either leverage bipartite factorization that requires stacking multiple mapping steps or autoregressive factorization that requires autoregressive training/inference, both of which have slow inference speeds.
- Model Diversity. All these models except GANs can generate samples with diverse modes. GANs usually suffer from model collapse problems where the model can only capture and generate a part of data distribution.
- Density Estimation. Autoregressive models use the chain rule to decompose the probability of data into a product of multiple single-step distributions and thus can estimate the probability in an exact way. Normalizing flows perform tractable density estimation of the data points and can give exact data probability. Unlike autoregressive models and normalizing flows, both VAEs and diffusion models can only give a lower bound of the data probability since they maximize the evidence lower bound of the likelihood (the true likelihood $p(x) = \int p(x|z)p(z)dz$ is very hard to be computed since it is intractable to calculate all possible values of the latent variable z). For SMLD and SDEs, they only care the score function instead of the data density, and thus cannot estimate the data probability. For probability flow ODEs, they can estimate the data probability in an exact way. GANs model the probability of data samples in an implicit way, and thus they cannot estimate the data probability.
- Latent Manipulation. Except for autoregressive models, all generative models can support latent manipulations to some extent. Some GAN-based models used in

TTS do not take random Gaussian noise as model input and thus cannot support latent manipulation.
- Error Propagation. Iterative-based methods, such as autoregressive models, diffusion models, SMLD, SDEs, and ODEs suffer from error propagation, as they typically use the ground-truth intermediate data for training (i.e., teacher-forcing) while the generated data for inference.
- Stable Training. The training of VAEs requires the tradeoff between the data reconstruction loss and the KL loss, which is not stable if loss weights are not properly used. GANs also suffer from notoriously unstable training due to the tradeoff between the generator and the discriminator.

References

1. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press
2. Bishop CM (2006) Pattern recognition and machine learning. Springer
3. Mitchell TM, Mitchell TM (1997) Machine learning, vol 1. McGraw-hill, New York
4. Xu R, Wunsch D (2005) Survey of clustering algorithms. IEEE Trans Neural Netw 16(3):645–678
5. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: A survey. ACM Comput Surv (CSUR) 41(3):1–58
6. Van Der Maaten L, Postma E, Van den Herik J, et al (2009) Dimensionality reduction: a comparative review. J Mach Learn Res 10(66–71):13
7. Bishop CM (1998) Latent variable models. In: Learning in graphical models. Springer, pp 371–403
8. Sutton RS, Barto AG (2018) Reinforcement learning: an introduction. MIT Press
9. Devlin J, Chang MW, Lee K, Toutanova K (2018) BERT: Pre-training of deep bidirectional Transformers for language understanding. Preprint. arXiv:1810.04805
10. Radford A, Narasimhan K, Salimans T, Sutskever I, et al (2018) Improving language understanding by generative pre-training
11. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I, et al (2019) Language models are unsupervised multitask learners. OpenAI Blog 1(8):9
12. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, et al (2020) Language models are few-shot learners. Adv Neural Inf Process Syst 33:1877–1901
13. Song K, Tan X, Qin T, Lu J, Liu TY (2019) Mass: masked sequence to sequence pre-training for language generation. In: International conference on machine learning. PMLR, pp 5926–5936
14. Schneider S, Baevski A, Collobert R, Auli M (2019) wav2vec: unsupervised pre-training for speech recognition. In: Proc Interspeech 2019, pp 3465–3469
15. Baevski A, Zhou Y, Mohamed A, Auli M (2020) wav2vec 2.0: A framework for self-supervised learning of speech representations. Adv Neural Inf Process Syst 33:12449–12460
16. Chen T, Kornblith S, Norouzi M, Hinton G (2020) A simple framework for contrastive learning of visual representations. In: International conference on machine learning. PMLR, pp 1597–1607
17. He K, Chen X, Xie S, Li Y, Dollár P, Girshick R (2022) Masked autoencoders are scalable vision learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 16000–16009
18. Bengio Y, Louradour J, Collobert R, Weston J (2009) Curriculum learning. In: Proceedings of the 26th annual international conference on machine learning, pp 41–48

19. Settles B (2009) Active learning literature survey. University of Wisconsin-Madison Department of Computer Sciences
20. Parisi GI, Kemker R, Part JL, Kanan C, Wermter S (2019) Continual lifelong learning with neural networks: a review. Neural Netw 113:54–71
21. Li H (2012) Statistical learning methods. Tsinghua University Press, pp 95–115
22. Kleinbaum DG, Dietz K, Gail M, Klein M, Klein M (2002) Logistic regression. Springer
23. Bottou L (2012) Stochastic gradient descent tricks. In: Neural networks: tricks of the trade. Springer, pp 421–436
24. Cho K, van Merriënboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: encoder–decoder approaches. In: Proceedings of SSST-8, eighth workshop on syntax, semantics and structure in statistical translation, pp 103–111
25. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
26. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
27. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
28. van den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K (2016) WaveNet: A generative model for raw audio. Preprint. arXiv:1609.03499
29. Kalchbrenner N, Espeholt L, Simonyan K, Oord Avd, Graves A, Kavukcuoglu K (2016) Neural machine translation in linear time. Preprint. arXiv:1610.10099
30. Cho K, van Merrienboer B, Gülçehre Ç, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: EMNLP
31. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ (eds) Advances in neural information processing systems 27: Annual conference on neural information processing systems 2014, December 8–13 2014, Montreal, Quebec, Canada, pp 3104–3112. https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html
32. Bahdanau D, Cho KH, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015
33. Dinh L, Krueger D, Bengio Y (2014) NICE: Non-linear independent components estimation. Preprint. arXiv:1410.8516
34. Dinh L, Sohl-Dickstein J, Bengio S (2016) Density estimation using Real NVP. Preprint. arXiv:1605.08803
35. Rezende D, Mohamed S (2015) Variational inference with normalizing flows. In: International conference on machine learning. PMLR, pp 1530–1538
36. Kingma DP, Salimans T, Jozefowicz R, Chen X, Sutskever I, Welling M (2016) Improved variational inference with inverse autoregressive flow. Adv Neural Inf Process Syst 29:4743–4751
37. Kingma DP, Dhariwal P (2018) Glow: generative flow with invertible 1×1 convolutions. In: Proceedings of the 32nd international conference on neural information processing systems, pp 10236–10245
38. Kingma DP, Welling M (2013) Auto-encoding variational Bayes. Preprint. arXiv:1312.6114
39. Sohl-Dickstein J, Weiss E, Maheswaranathan N, Ganguli S (2015) Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. PMLR, pp 2256–2265
40. Ho J, Jain A, Abbeel P (2020) Denoising diffusion probabilistic models. Preprint. arXiv:2006.11239
41. LeCun Y, Chopra S, Hadsell R, Ranzato MA, Huang FJ (2006) A tutorial on energy-based learning. To appear in Predict Struct Data 1(0)
42. Song Y, Ermon S (2019) Generative modeling by estimating gradients of the data distribution. In: Wallach HM, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox EB, Gar-

nett R (eds) Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada, pp 11895–11907. https://proceedings.neurips.cc/paper/2019/hash/3001ef257407d5a371a96dcd947c7d93-Abstract.html
43. Song Y, Sohl-Dickstein J, Kingma DP, Kumar A, Ermon S, Poole B (2020) Score-based generative modeling through stochastic differential equations. In: International conference on learning representations
44. Song J, Meng C, Ermon S (2020) Denoising diffusion implicit models. Preprint. arXiv:2010.02502
45. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: NIPS
46. van den Oord A, Kalchbrenner N, Kavukcuoglu K (2016) Pixel recurrent neural networks. In: International conference on machine learning. PMLR, pp 1747–1756
47. van den Oord A, Kalchbrenner N, Vinyals O, Espeholt L, Graves A, Kavukcuoglu K (2016) Conditional image generation with PixelCNN decoders. In: Proceedings of the 30th international conference on neural information processing systems, pp 4797–4805
48. van der Berg R, Hasenclever L, Tomczak JM, Welling M (2018) Sylvester normalizing flows for variational inference. Preprint. arXiv:180305649
49. Huang CW, Krueger D, Lacoste A, Courville A (2018) Neural autoregressive flows. In: International conference on machine learning. PMLR, pp 2078–2087
50. Papamakarios G, Pavlakou T, Murray I (2017) Masked autoregressive flow for density estimation. In: Proceedings of the 31st international conference on neural information processing systems, pp 2335–2344
51. Grathwohl W, Chen RT, Bettencourt J, Sutskever I, Duvenaud D (2018) FFJORD: Free-form continuous dynamics for scalable reversible generative models. In: International conference on learning representations
52. Chen RTQ, Behrmann J, Duvenaud D, Jacobsen J-H (2019) Residual flows for invertible generative modeling. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, pp 9916–9926
53. Ping W, Peng K, Zhao K, Song Z (2020) WaveFlow: a compact flow-based model for raw audio. In: International conference on machine learning. PMLR, pp 7706–7716
54. Blei DM, Kucukelbir A, McAuliffe JD (2017) Variational inference: A review for statisticians. J Am Stat Assoc 112(518):859–877
55. Jensen JLWV (1906) Sur les fonctions convexes et les inégalités entre les valeurs moyennes. Acta Math 30(1):175–193
56. Anderson BD (1982) Reverse-time diffusion equation models. Stoch Process Appl 12(3):313–326
57. Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision, pp 2223–2232
58. Karras T, Laine S, Aila T (2019) A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4401–4410
59. Donahue C, McAuley J, Puckette M (2018) Adversarial audio synthesis. In: International conference on learning representations
60. Kumar K, Kumar R, de Boissiere T, Gestin L, Teoh WZ, Sotelo J, de Brébisson A, Bengio Y, Courville A (2019) MelGAN: Generative adversarial networks for conditional waveform synthesis. In: NeurIPS
61. Kong J, Kim J, Bae J (2020) HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. Adv Neural Inf Process Syst 33:17022
62. Yu L, Zhang W, Wang J, Yu Y (2017) SeGAN: Sequence generative adversarial nets with policy gradient. In: Proceedings of the AAAI conference on artificial intelligence, vol 31

Part II
Key Components in TTS

In this part, we introduce the key components of a neural TTS system. Generally, there are three components in a neural TTS system:[1] a text analysis module, an acoustic model, and a vocoder. We first introduce each component individually in Chaps. 4, 5, and 6, and then introduce some methods that combine two or more components together to build end-to-end TTS systems in Chap. 7.

We categorize neural TTS from the perspective of basic TTS components: text analyses, acoustic models, vocoders,[2] and fully end-to-end models, as shown in Fig. 1a. We find this taxonomy is consistent with the data conversion flow from text to waveform: (1) text analyses convert characters into phonemes 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.[3]

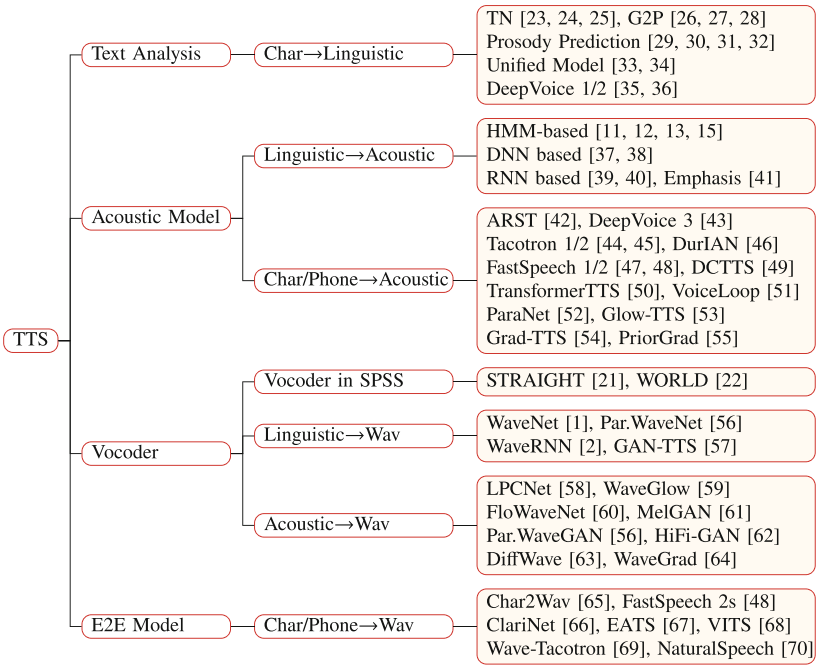
We can also categorize neural TTS according to the data flow from text to the waveform, as shown in Fig. 1b. There are several data representations in the process of text-to-speech conversion: (1) Characters, which are the raw format of text. (2) Linguistic features, which are obtained through text analysis and

[1] The three components are not specific in neural TTS and they also exist in statistical parametric speech synthesis (SPSS). However, neural networks are used as both acoustic models and vocoders in neural TTS while only as acoustic models in SPSS.

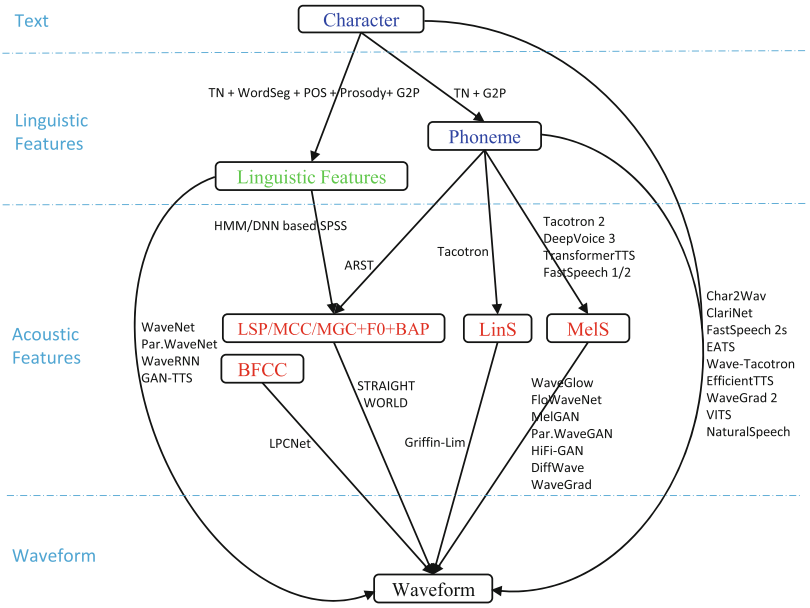
[2] Note that some neural TTS models such as WaveNet [1] and WaveRNN [2] directly generate waveform from linguistic features. From this perspective, WaveNet can be regarded as a combination of an acoustic model and a vocoder. The following works usually leverage WaveNet and WaveRNN as a vocoder by taking mel-spectrograms as input to generate the waveform. Therefore, we categorize WaveNet/WaveRNN into vocoders and introduce them in Chap. 6.

[3] Note that some recent TTS systems [3–6] learn intermediate representations such as continuous vectors [3, 4] or discrete tokens [5, 6] instead of traditional mel-spectrograms, using a neural audio codec [7–10] to bridge the mapping between text and speech. They leverage an acoustic model or a language model to generate these continuous vectors or discrete tokens and leverage the codec decoder to generate waveforms from these intermediate representations. Since these methods are explored at the early stage, we do not cover them in detail in this book.

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 TTS. (3) Acoustic features, which are abstractive representations of the speech waveform. In statistical parametric speech synthesis [11–15], LSP (line spectral pairs) [16], MCC (mel-cepstral coefficients) [17], MGC (mel-generalized coefficients) [18], F0 and BAP (band aperiodicities) [19, 20] are used as acoustic features, which can be easily converted into waveforms through vocoders such as STRAIGHT [21] and WORLD [22]. In neural-based TTS models, mel-spectrograms are usually used as acoustic features, which are converted into waveforms using neural-based vocoders. (4) Waveform, the final form of speech. As can be seen from Fig. 1b, there can be different data flows from text to waveform, including (1) character → linguistic features → acoustic features → waveform; (2) character → phoneme → acoustic features → waveform; (3) character → linguistic features → waveform; (4) character → phoneme → acoustic features → waveform; (5) character → phoneme → waveform, or character → waveform.



(a)



(b)

Fig. 1 A taxonomy of neural TTS from the perspectives of key components. (**a**) A taxonomy of neural TTS. (**b**) The data flows from text to waveform

Chapter 4
Text Analyses

Abstract Through text analyses, we can transform input text into linguistic features, which contain rich information about pronunciation and prosody that can ease speech synthesis. Text analyses consist of several components: (1) text processing, which processes raw text from documents, normalizes the text from the written form into spoken form, and conducts some linguistic analyses; (2) phonetic analysis, which converts text into phonetic symbols, including polyphone disambiguation and grapheme-to-phoneme conversion; (3) prosodic analysis, which analyzes some prosodic features such as pitch, duration, loudness, stress, and pauses. In this chapter, we first introduce these components in the first three sections and then discuss the development of text analysis in TTS in the last section.

Keywords Text analysis · Text processing · Phonetic analysis · Prosodic analysis

Through text analyses, we can transform input text into linguistic features, which contain rich information about pronunciation and prosody that can ease speech synthesis. Text analyses consist of several components, as shown in Fig. 4.1: (1) text processing, which processes raw text from documents, normalizes the text from a written form into a spoken form, and conducts some linguistic analyses; (2) phonetic analysis, which converts text into phonetic symbols, including polyphone disambiguation and grapheme-to-phoneme conversion; (3) prosodic analysis, which analyzes some prosodic features such as pitch, duration, loudness, stress, and pauses. In this chapter, we first introduce these components in Sects. 4.1, 4.2, and 4.3, and then discuss the development of text analysis in TTS in Sect. 4.4.

Prerequisite Knowledge for Reading This Chapter

- Basic knowledge of spoken language processing, such as phonetics, phonology, morphology, syntax, and semantics.

X. Tan, *Neural Text-to-Speech Synthesis*, Artificial Intelligence: Foundations, Theory, and Algorithms, https://doi.org/10.1007/978-981-99-0827-1_4

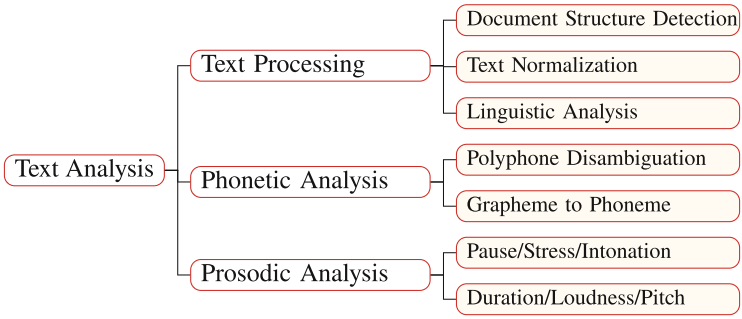


Fig. 4.1 The components in text analysis

4.1 Text Processing

Text processing is to extract, simplify, normalize, and analyze text to make it suitable for phonetic analysis and prosodic analysis. The processes include (1) document structure detection, which helps to locate the document structure of a sentence that is useful for later processing; (2) text normalization, which converts the text from a nonorthographic form into an orthographic form to ease phonetic conversion; and (3) linguistic analysis, which processes the syntactic and semantic features that are helpful for later phonetic and prosodic analyses.

4.1.1 Document Structure Detection

The input text of TTS systems is usually not single sentences by itself, but from documents (e.g., book, article, e-mail, web page, conversation, dialog) that can provide context for these sentences. Different structures, such as chapter/section headers, lists, paragraphs, sentences, e-mails, and dialogue turns can provide different indications for intonational contours, pitch assignments, and prosodic styles. A typical task for document structure detection is sentence breaking since a knowledge of the sentence unit is important for correct pronunciation and prosodic breaking.

4.1.2 Text Normalization

Non-standard words usually contain a lot of nonorthographic forms or semiotic classes [71], such as (1) abbreviations and acronyms (e.g., TTS for Text to Speech, OPEC for the Organization of Petroleum Exporting Countries); (2) number formats (e.g., phone number 716-123-4568, date 03/15/2022, time 2:18 pm, money and

currency $32, account numbers 6217-9062, ordinal numbers 1st, cardinal numbers 3728); (3) scientific formula (e.g., mathematical formula $\frac{\sqrt{x}}{y}$, chemical formula H_2O); (4) Web and Internet address (e.g., https://www.microsoft.com/); (5) special symbols (e.g., emotion ":-)" means smiley). To make the text suitable for subsequent phonetic conversion and easy to pronounce for TTS systems, text normalization is leveraged to convert text from the nonorthographic form (written form) into the orthographic form (speakable form). For example, the date "Jan. 24, 1989" is normalized into "January twenty-fourth nineteen eighty-nine". Early works on text normalization are rule-based [71], 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 [23–25]. Some works [72] propose to combine the advantages of both rule-based and neural-based models to further improve the performance of text normalization.

4.1.3 Linguistic Analysis

Linguistic analysis is to extract structural and semantic information from sentences through syntactic and semantic parsing. Linguistic analysis has several usages in TTS: (1) it can provide additional grammar information to help determine the pronunciation of a word in different senses or abstraction inflections (e.g., "read" is pronounced as /ri: d/ in present tense and /red/ in past tense); (2) it can provide additional information to differentiate sentences that are the same after text normalization; (3) it can provide useful information to determine the prosodic structure that influences the duration and pitch contour (e.g., the syntactic type of a sentence, yes/no question, and wh-question have different duration and pitch contours although both are marked with a question symbol /?/). Some basic syntactic and semantic parsing include sentence type detection, word/phrase/sentence segmentation, part of speech (POS) tagging, word sense disambiguation, and homograph disambiguation. We introduce these syntactic and semantic parsing tasks in the following paragraphs.

Sentence Breaking and Type Detection

Sentence breaking from raw documents is important for correct pronunciation and prosodic breaking and can be implemented in some rules based on punctuation. Different sentence types, such as declarative sentence, yes/no question, and wh-question, have different prosodic lines in synthesized speech. We can use some rules to detect the type of a sentence by checking some keywords (e.g., please, what/how, is, isn't) and punctuations (e.g., /./, /!/, /?/).

Word/Phrase Segmentation

For languages like Chinese, Thai, and Japanese, word segmentation [73–75] is necessary to detect the word boundary from raw text, which is important to ensure the accuracy for POS tagging, phonetic and prosodic analysis. The segmentations of phrases such as noun phrases and clauses are important for prosodic analysis, especially in determining the pitch, duration, and pause of phrases to make a sentence more intelligible and natural. Word segmentation and phrase parsing are well supported by popular parsing toolkits, such as Jieba[1] and Stanford CoreNLP.[2]

Part-of-Speech Tagging

The part-of-speech (POS) of each word, such as noun, verb, and preposition, is important for the phonetic and prosodic analysis in TTS. Several works have investigated POS tagging in speech synthesis [74, 76–79].

Homograph and Word Sense Disambiguation

Homographs represent words that have the same written form (spelling) but have different meanings. For example, "bear" is either a noun that represents a kind of animal or a verb that is similar to "tolerate". Thus, "bear" is regarded as a homograph. A similar concept is word sense, which means one of the meanings of a word. Disambiguating homographs or word senses is helpful to understand the semantics and syntax of a sentence better, which is helpful for later prosodic analysis.

4.2 Phonetic Analysis

Phonetic analysis can ease the speech synthesis process by providing information about how a word should be pronounced. It involves the study of the pronunciation of a word and the conversion from its grapheme sequence (lexical orthographic symbols) into phoneme sequence, with possible diacritic information (e.g., stress placement) for precise pronunciation. Phonetic analysis includes two tasks: polyphone disambiguation to determine different pronunciations of the same word in different word contexts, and grapheme-to-phoneme conversion to generate the phoneme sequence of a word.

[1] https://github.com/fxsjy/jieba.

[2] https://github.com/stanfordnlp/CoreNLP.

4.2.1 Polyphone Disambiguation

A polyphone refers to a word that can be pronounced in two or more different ways, where each way represents a different word sense. Many languages have polyphones. For example, "resume" in English can be pronounced as /riˈzju:mˈ/ (a verb, means to go on or continue after interruption) or /ˈrezjumei/ (a noun, means curriculum vitae), "奇" in Chinese can be pronounced as jī (means odd or odd number) or qí (means strange). Polyphone disambiguation is to decide the appropriate pronunciation based on the context of this word/character [26, 27, 80–83]. Note that polyphones are different from homographs since a polyphone has multiple different pronunciations while a homograph has multiple different meanings but not necessarily have multiple pronunciations.

4.2.2 Grapheme-to-Phoneme Conversion

After polyphone disambiguation, we further conduct grapheme-to-phoneme conversion to transform characters (graphemes) into pronunciations (phonemes) (e.g., the word "speech" is converted into "s p iy ch"), which can greatly ease speech synthesis. For alphabetic languages with simple and clear relationships between graphemes and phonemes (phonetic languages, e.g., Spanish), grapheme-to-phoneme conversion should be easy and can be well processed by handcrafted rules. For alphabetic languages with complicated relationships between graphemes and phonemes (non-phonetic language, e.g., English), handcrafted rules cannot cover all words. Thus, grapheme-to-phoneme conversion models are usually developed to generate the pronunciations of out-of-vocabulary words [26, 27, 80–83]. For some languages like Arabic and Hebrew, the vowel information is not available in written text and needed to be determined/predicted from the text (it can be also regarded as a polyphone disambiguation task). For non-alphabetic languages (e.g., Chinese), a manually collected grapheme-to-phoneme lexicon is usually leveraged for conversion, which can cover nearly all the characters.

4.3 Prosodic Analysis

To make the synthesized speech sound natural, we need to conduct a prosodic analysis of the sentences properly. The prosodic analysis involves the analysis of prosody information such as rhythm, stress, and intonation of speech, which correspond to the variations in phoneme duration, loudness, and pitch, and play an important perceptual role in human speech communication. In the following subsections, we

introduce how to analyze the pause, stress, and intonation information, as well as the pitch, duration, and loudness of the speech.[3]

4.3.1 Pause, Stress, and Intonation

Prosody analysis relies on tagging systems to label each kind of prosody, such as pause, stress, and intonation. Different languages have different prosody tagging systems and tools [84–88]. For English, ToBI (tones and break indices) [84, 85] 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 [29–31, 89] investigate different models and features to predict the prosody tags based on ToBI. For Chinese speech synthesis, the typical prosody boundary labels consist of the prosodic word (PW), the prosodic phrase (PPH), and the intonational phrase (IPH), which can construct a three-layer hierarchical prosody tree [90–92]. Some works [90–95] investigate different model structures such as conditional random field [96], RNN [97], and self-attention [98] for prosody prediction in Chinese.

4.3.2 Pitch, Duration, and Loudness

Pause, stress, and intonation are fine-grained features to determine the prosody of a speech. Alternatively, we can also use pitch, duration, and loudness features to determine the prosody of speech. The differences between pause/stress/intonation and pitch/duration/loudness may be that pauses/stresses/intonations are more comprehensible by listeners or users, while pitch/duration/loudness is more like basis features and capture the characteristics of speech prosody in a data-driven way. Any kind of prosody such as pause/stress/intonation can be obtained by varying pitch/duration/loudness. Due to the fundamental and basic representations brought by pitch/duration/loudness, several neural TTS models [47, 48, 99] leverage them as the prosody features (variance information) to improve the quality of synthesized speech.

[3] Strictly speaking, duration, pitch, and loudness are acoustic/auditory characteristics and may not belong to text analyses. However, we describe them together with pause, stress, and intonation in this chapter for ease of understanding.

4.4 Text Analysis from a Historic Perspective

In this section, we briefly overview the development progress of text analysis in TTS, from conventional statistical parametric speech synthesis to neural speech synthesis.

4.4.1 Text Analysis in SPSS

In statistical parametric speech synthesis (SPSS), text analysis is used to extract a sequence of linguistic feature vectors [15], which are taken as input to the later part of the TTS pipeline, e.g., acoustic models in SPSS or neural vocoders [1]. Therefore, text analysis nearly consists of all the modules in text processing (e.g., text normalization [23, 72], word segmentation [73], part-of-speech (POS) tagging [76], phonetic analysis (e.g., grapheme-to-phoneme conversion [26]), and prosodic analysis (e.g., prosody prediction [90])). Usually, we can construct linguistic features by aggregating the results of text analysis from different levels including phoneme, syllable, word, phrase, and sentence levels [15].

- Phoneme level: the phonetic symbols of the previous before the previous, the previous, the current, the next, or the next after the next; the forward or backward distance of the current phoneme within the syllable.
- Syllable level: whether the previous, the current, or the next syllable is stressed; the number of phonemes contained in the previous, the current, or the next syllable; the forward or the backward distance of the current syllable within the word or phrase; the number of the stressed syllables before or after the current syllable within the phrase; the distance from the current syllable to the forward or backward most nearest stressed syllable; the vowel phonetics of the current syllable.
- Word level: the part of speech (POS) of the previous, the current, or the next word; the number of syllables of the previous, the current, or the next word; the forward or backward position of the current word in the phrase; the forward or backward content word of the current word within the phrase; the distance from the current word to the forward or backward nearest content word; the POS of the previous, the current or the next word.
- Phrase level: the number of syllables of the previous, the current, or the next phrase; the number of words of the previous, the current, or the next phrase; the forward or backward position of the current phrase in the sentence; the prosodic annotation of the current phrase.
- Sentence level: The number of syllables, words, or phrases in the current sentence.

4.4.2 *Text Analysis in Neural TTS*

In neural TTS, due to the large modeling capacity of neural-based models, the character or phoneme sequences are directly taken as input for synthesis. In this way, only text normalization is needed to convert raw text with a non-standard format to a standard word format if the character is taken as input, and grapheme-to-phoneme conversion is further needed to get phonemes from standard word format if phonemes are taken as input. Thus, the text analysis module is largely simplified.

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) Neural network-based text analysis module. Char2Wav [65] and DeepVoice 1/2 [35, 36] implement the character-to-linguistic feature conversion into its pipeline, purely based on neural networks. Pan et al. [33] and Zhang et al. [34] designed unified models to cover all the tasks in text analysis in a multi-task paradigm and achieve good results. (2) Prosody prediction. Prosody 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 the text encoder, such as the prediction of pitch [48], duration [47], phrase break [100], breath or filled pauses [101], or prosodic style features [102] are built on top of the text (character or phoneme) encoder in TTS models. (3) Reference encoders. Some works [103–105] use reference encoders to learn the prosody representations from reference speech. (4) Text pre-training. Some works learn good text representations with implicit prosody information through self-supervised pre-training [70, 106–108]. (5) Incorporating syntax information through dedicated modeling methods such as graph networks [109].

References

1. van den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K (2016) WaveNet: a generative model for raw audio. Preprint. arXiv:1609.03499
2. Kalchbrenner N, Elsen E, Simonyan K, Noury S, Casagrande N, Lockhart E, Stimberg F, Oord A, Dieleman S, Kavukcuoglu K (2018) Efficient neural audio synthesis. In: International conference on machine learning. PMLR, pp 2410–2419
3. Liu Y, Xue R, He L, Tan X, Zhao S (2022) DelightfulTTS 2: End-to-end speech synthesis with adversarial vector-quantized auto-encoders. Preprint. arXiv:2207.04646
4. Cong J, Yang S, Xie L, Su D (2021) Glow-WaveGAN: learning speech representations from GAN-based variational auto-encoder for high fidelity flow-based speech synthesis. Preprint. arXiv:2106.10831
5. Hayashi T, Watanabe S (2020) DiscreTalk: Text-to-speech as a machine translation problem. Preprint. arXiv:2005.05525
6. Du C, Guo Y, Chen X, Yu K (2022) VQTTS: High-fidelity text-to-speech synthesis with self-supervised VQ acoustic feature. Preprint. arXiv:2204.00768

7. van den Oord A, Vinyals O, Kavukcuoglu K (2017) Neural discrete representation learning. In: Proceedings of the 31st international conference on neural information processing systems, pp 6309–6318
8. Razavi A, van den Oord A, Vinyals O (2019) Generating diverse high-fidelity images with VQ-VAE-2. In: Advances in neural information processing systems, pp 14866–14876
9. Zeghidour N, Luebs A, Omran A, Skoglund J, Tagliasacchi M (2021) SoundStream: An end-to-end neural audio codec. IEEE/ACM Trans Audio Speech Lang Process 30:495
10. Défossez A, Copet J, Synnaeve G, Adi Y (2022) High fidelity neural audio compression. Preprint. arXiv:2210.13438
11. Yoshimura T, Tokuda K, Masuko T, Kobayashi T, Kitamura T (1999) Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In: Sixth European conference on speech communication and technology
12. Tokuda K, Yoshimura T, Masuko T, Kobayashi T, Kitamura T (2000) Speech parameter generation algorithms for HMM-based speech synthesis. In: 2000 IEEE international conference on acoustics, speech, and signal processing. proceedings (Cat. No. 00CH37100). IEEE, vol 3, pp 1315–1318
13. Yoshimura T (2002) Simultaneous modeling of phonetic and prosodic parameters, and characteristic conversion for HMM-based text-to-speech systems. PhD diss, Nagoya Institute of Technology
14. Zen H, Tokuda K, Black AW (2009) Statistical parametric speech synthesis. Speech Commun 51(11):1039–1064
15. Tokuda K, Nankaku Y, Toda T, Zen H, Yamagishi J, Oura K (2013) Speech synthesis based on hidden Markov models. Proc IEEE 101(5):1234–1252
16. Itakura F (1975) Line spectrum representation of linear predictor coefficients of speech signals. J Acoust Soc Am 57(S1):S35–S35
17. Fukada T, Tokuda K, Kobayashi T, Imai S (1992) An adaptive algorithm for mel-cepstral analysis of speech. In: Proc. ICASSP, vol 1, pp 137–140
18. Tokuda K, Kobayashi T, Masuko T, Imai S (1994) Mel-generalized cepstral analysis-a unified approach to speech spectral estimation. In: Third international conference on spoken language processing
19. Kawahara H, Masuda-Katsuse I, De Cheveigne A (1999) 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 Commun 27(3–4):187–207
20. Kawahara H, Estill J, Fujimura O (2001) 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
21. Kawahara H (2006) STRAIGHT, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds. Acoust Sci Technol 27(6):349–353
22. Morise M, Yokomori F, Ozawa K (2016) WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. IEICE Trans Inf Syst 99(7):1877–1884
23. Sproat R, Jaitly N (2016) RNN approaches to text normalization: A challenge. Preprint. arXiv:1611.00068
24. Mansfield C, Sun M, Liu Y, Gandhe A, Hoffmeister B (2019) 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), pp 190–196
25. Zhang H, Sproat R, Ng AH, Stahlberg F, Peng X, Gorman K, Roark B (2019) Neural models of text normalization for speech applications. Comput Linguist 45(2):293–337
26. Yao K, Zweig G (2015) Sequence-to-sequence neural net models for grapheme-to-phoneme conversion. In: Sixteenth annual conference of the International Speech Communication Association

27. Sun H, Tan X, Gan JW, Liu H, Zhao S, Qin T, Liu TY (2019) Token-level ensemble distillation for grapheme-to-phoneme conversion. In: INTERSPEECH
28. Sun H, Tan X, Gan JW, Zhao S, Han D, Liu H, Qin T, Liu TY (2019) Knowledge distillation from BERT in pre-training and fine-tuning for polyphone disambiguation. In: 2019 IEEE automatic speech recognition and understanding workshop (ASRU). IEEE, pp 168–175
29. Sridhar VKR, Bangalore S, Narayanan S (2007) 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, pp 1–8
30. Jeon JH, Liu Y (2009) Automatic prosodic events detection using syllable-based acoustic and syntactic features. In: 2009 IEEE international conference on acoustics, speech and signal processing. IEEE, pp 4565–4568
31. Qian Y, Wu Z, Ma X, Soong F (2010) Automatic prosody prediction and detection with conditional random field (CRF) models. In: 2010 7th International symposium on Chinese spoken language processing. IEEE, pp 135–138
32. Pan H, Li X, Huang Z (2019) A Mandarin prosodic boundary prediction model based on multi-task learning. In: INTERSPEECH, pp 4485–4488
33. Pan J, Yin X, Zhang Z, Liu S, Zhang Y, Ma Z, Wang Y (2020) 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). IEEE, pp 6689–6693
34. Zhang Y, Deng L, Wang Y (2020) Unified Mandarin TTS front-end based on distilled BERT model. Preprint. arXiv:2012.15404
35. Arık SÖ, Chrzanowski M, Coates A, Diamos G, Gibiansky A, Kang Y, Li X, Miller J, Ng A, Raiman J, et al (2017) Deep Voice: real-time neural text-to-speech. In: International conference on machine learning. PMLR, pp 195–204
36. Gibiansky A, Arik SÖ, Diamos GF, Miller J, Peng K, Ping W, Raiman J, Zhou Y (2017) Deep Voice 2: Multi-speaker neural text-to-speech. In: NIPS
37. Zen H, Senior A, Schuster M (2013) Statistical parametric speech synthesis using deep neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, pp 7962–7966
38. Qian Y, Fan Y, Hu W, Soong FK (2014) 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). IEEE, pp 3829–3833
39. Fan Y, Qian Y, Xie FL, Soong FK (2014) TTS synthesis with bidirectional LSTM based recurrent neural networks. In: Fifteenth annual conference of the international speech communication association
40. Zen H (2015) Acoustic modeling in statistical parametric speech synthesis-from HMM to LSTM-RNN. In: Proc MLSLP. Invited paper
41. Li H, Kang Y, Wang Z (2018) EMPHASIS: An emotional phoneme-based acoustic model for speech synthesis system. In: Proc Interspeech 2018, pp 3077–3081
42. Wang W, Xu S, Xu B (2016) First step towards end-to-end parametric TTS synthesis: Generating spectral parameters with neural attention. In: Interspeech, pp 2243–2247
43. Ping W, Peng K, Gibiansky A, Arik SO, Kannan A, Narang S, Raiman J, Miller J (2018) Deep Voice 3: 2000-speaker neural text-to-speech. In: Proc ICLR, pp 214–217
44. Wang Y, Skerry-Ryan R, Stanton D, Wu Y, Weiss RJ, Jaitly N, Yang Z, Xiao Y, Chen Z, Bengio S, et al (2017) Tacotron: towards end-to-end speech synthesis. In: Proc Interspeech 2017, pp 4006–4010
45. Shen J, Pang R, Weiss RJ, Schuster M, Jaitly N, Yang Z, Chen Z, Zhang Y, Wang Y, Skerry-Ryan R, et al (2018) Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4779–4783

46. Yu C, Lu H, Hu N, Yu M, Weng C, Xu K, Liu P, Tuo D, Kang S, Lei G, et al (2020) DurIAN: Duration informed attention network for speech synthesis. In: Proc Interspeech 2020, pp 2027–2031
47. Ren Y, Ruan Y, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2019) FastSpeech: fast, robust and controllable text to speech. In: NeurIPS
48. Ren Y, Hu C, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2021) FastSpeech 2: fast and high-quality end-to-end text to speech. In: International conference on learning representations
49. Tachibana H, Uenoyama K, Aihara S (2018) 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). IEEE, pp 4784–4788
50. Li N, Liu S, Liu Y, Zhao S, Liu M (2019) Neural speech synthesis with transformer network. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 6706–6713
51. Taigman Y, Wolf L, Polyak A, Nachmani E (2018) VoiceLoop: voice fitting and synthesis via a phonological loop. In: International conference on learning representations
52. Peng K, Ping W, Song Z, Zhao K (2020) Non-autoregressive neural text-to-speech. In: International conference on machine learning. PMLR, pp 7586–7598
53. Kim J, Kim S, Kong J, Yoon S (2020) Glow-TTS: A generative flow for text-to-speech via monotonic alignment search. Adv Neural Inf Process Syst 33, 8067
54. Popov V, Vovk I, Gogoryan V, Sadekova T, Kudinov M (2021) Grad-TTS: A diffusion probabilistic model for text-to-speech. Preprint. arXiv:2105.06337
55. Lee Sg, Kim H, Shin C, Tan X, Liu C, Meng Q, Qin T, Chen W, Yoon S, Liu TY (2021) PriorGrad: Improving conditional denoising diffusion models with data-driven adaptive prior. Preprint. arXiv:2106.06406
56. van den Oord A, Li Y, Babuschkin I, Simonyan K, Vinyals O, Kavukcuoglu K, Driessche G, Lockhart E, Cobo L, Stimberg F, et al (2018) Parallel WaveNet: fast high-fidelity speech synthesis. In: International conference on machine learning. PMLR, pp 3918–3926
57. Bińkowski M, Donahue J, Dieleman S, Clark A, Elsen E, Casagrande N, Cobo LC, Simonyan K (2019) High fidelity speech synthesis with adversarial networks. In: International conference on learning representations
58. Valin JM, Skoglund J (2019) LPCNet: Improving neural speech synthesis through linear prediction. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 5891–5895
59. Prenger R, Valle R, Catanzaro B (2019) WaveGlow: a flow-based generative network for speech synthesis. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 3617–3621
60. Kim S, Lee SG, Song J, Kim J, Yoon S (2019) FloWaveNet: A generative flow for raw audio. In: International conference on machine learning. PMLR, pp 3370–3378
61. Kumar K, Kumar R, de Boissiere T, Gestin L, Teoh WZ, Sotelo J, de Brébisson A, Bengio Y, Courville A (2019) MelGAN: Generative adversarial networks for conditional waveform synthesis. In: NeurIPS
62. Kong J, Kim J, Bae J (2020) HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. Adv Neural Inf Process Syst 33, 17022
63. Kong Z, Ping W, Huang J, Zhao K, Catanzaro B (2021) DiffWave: a versatile diffusion model for audio synthesis. In: ICLR
64. Chen N, Zhang Y, Zen H, Weiss RJ, Norouzi M, Chan W (2021) WaveGrad: Estimating gradients for waveform generation. In: ICLR
65. Sotelo J, Mehri S, Kumar K, Santos JF, Kastner K, Courville AC, Bengio Y (2017) Char2wav: End-to-end speech synthesis. In: 5th international conference on learning representations, ICLR 2017, Toulon, France, April 24–26, 2017, Workshop Track Proceedings. OpenReview.net. https://openreview.net/forum?id=B1VWyySKx
66. Ping W, Peng K, Chen J (2018) ClariNet: Parallel wave generation in end-to-end text-to-speech. In: International conference on learning representations
67. Donahue J, Dieleman S, Bińkowski M, Elsen E, Simonyan K (2021) End-to-end adversarial text-to-speech. In: ICLR

68. Kim J, Kong J, Son J (2021) Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. Preprint. arXiv:2106.06103
69. Weiss RJ, Skerry-Ryan R, Battenberg E, Mariooryad S, Kingma DP (2021) Wave-Tacotron: spectrogram-free end-to-end text-to-speech synthesis. In: 2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
70. Tan X, Chen J, Liu H, Cong J, Zhang C, Liu Y, Wang X, Leng Y, Yi Y, He L, et al (2022) NaturalSpeech: end-to-end text to speech synthesis with human-level quality. Preprint. arXiv:2205.04421
71. Sproat R, Black AW, Chen S, Kumar S, Ostendorf M, Richards C (2001) Normalization of non-standard words. Computer Speech Lang 15(3):287–333
72. Zhang J, Pan J, Yin X, Li C, Liu S, Zhang Y, Wang Y, Ma Z (2020) 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). IEEE, pp 6694–6698
73. Xue N (2003) 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, pp 29–48
74. Zheng X, Chen H, Xu T (2013) Deep learning for Chinese word segmentation and POS tagging. In: Proceedings of the 2013 conference on empirical methods in natural language processing, pp 647–657
75. Pei W, Ge T, Chang B (2014) 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), pp 293–303
76. Schlünz GI (2010) The effects of part-of-speech tagging on text–to–speech synthesis for resource–scarce languages. PhD thesis, North-West University
77. Sun M, Bellegarda JR (2011) Improved POS tagging for text-to-speech synthesis. In: 2011 IEEE International conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 5384–5387
78. Mamateli G, Rozi A, Ali G, Hamdulla A (2011) Morphological analysis based part-of-speech tagging for uyghur speech synthesis. In: Knowledge engineering and management. Springer, pp 389–396
79. Janicki A (2004) Application of neural networks for POS tagging and intonation control in speech synthesis for polish. Soft Comput Intell Syst (SCIS 2004) 7
80. Chen SF (2003) Conditional and joint models for grapheme-to-phoneme conversion. In: Eighth European conference on speech communication and technology
81. Bisani M, Ney H (2008) Joint-sequence models for grapheme-to-phoneme conversion. Speech Commun 50(5):434–451
82. Rao K, Peng F, Sak H, Beaufays F (2015) Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. In: 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4225–4229
83. Chae MJ, Park K, Bang J, Suh S, Park J, Kim N, Park L (2018) 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). IEEE, pp 2486–2490
84. Silverman K, Beckman M, Pitrelli J, Ostendorf M, Wightman C, Price P, Pierrehumbert J, Hirschberg J (1992) ToBI: a standard for labeling English prosody. In: Second international conference on spoken language processing
85. Rosenberg A (2010) AuToBI-A tool for automatic ToBI annotation. In: Eleventh annual conference of the International Speech Communication Association
86. Taylor P (1998) The Tilt intonation model. In: Fifth international conference on spoken language processing
87. Hirst D (2001) Automatic analysis of prosody for multilingual speech corpora. In: Improvements in speech synthesis, pp 320–327

88. Obin N, Beliao J, Veaux C, Lacheret A (2014) SLAM: automatic stylization and labelling of speech melody. In: Speech prosody, p 246
89. Levow GA (2008) 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
90. Chu M, Qian Y (2001) Locating boundaries for prosodic constituents in unrestricted Mandarin texts. In: International journal of computational linguistics & Chinese language processing, vol 6, no 1, February 2001: Special Issue on Natural Language Processing Researches in MSRA, pp 61–82
91. Sun J, Yang J, Zhang J, Yan Y (2009) Chinese prosody structure prediction based on conditional random fields. In: 2009 Fifth international conference on natural computation. IEEE, vol 3, pp 602–606
92. Ding C, Xie L, Yan J, Zhang W, Liu Y (2015) Automatic prosody prediction for Chinese speech synthesis using BLSTM-RNN and embedding features. In: 2015 IEEE workshop on automatic speech recognition and understanding (ASRU). IEEE, pp 98–102
93. Ai Y, Ling ZH (2020) A neural vocoder with hierarchical generation of amplitude and phase spectra for statistical parametric speech synthesis. IEEE/ACM Trans Audio Speech Lang Process 28:839–851
94. Lu C, Zhang P, Yan Y (2019) Self-attention based prosodic boundary prediction for Chinese speech synthesis. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 7035–7039
95. Lu Y, Dong M, Chen Y (2019) Implementing prosodic phrasing in Chinese end-to-end speech synthesis. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 7050–7054
96. Lafferty JD, McCallum A, Pereira FCN (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Brodley CE, Danyluk AP (eds) Proceedings of the eighteenth international conference on machine learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 – July 1, 2001. Morgan Kaufmann, pp 282–289
97. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
98. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
99. Łańcucki A (2020) FastPitch: Parallel text-to-speech with pitch prediction. Preprint. arXiv:2006.06873
100. Liu R, Sisman B, Bao F, Gao G, Li H (2020) Modeling prosodic phrasing with multi-task learning in Tacotron-based TTS. IEEE Signal Process Lett 27:1470–1474
101. Yan Y, Tan X, Li B, Zhang G, Qin T, Zhao S, Shen Y, Zhang WQ, Liu TY (2021) AdaSpeech 3: adaptive text to speech for spontaneous style. In: INTERSPEECH
102. Stanton D, Wang Y, Skerry-Ryan R (2018) Predicting expressive speaking style from text in end-to-end speech synthesis. In: 2018 IEEE spoken language technology workshop (SLT). IEEE, pp 595–602
103. Wang Y, Stanton D, Zhang Y, Skerry-Ryan R, Battenberg E, Shor J, Xiao Y, Jia Y, Ren F, Saurous RA (2018) Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In: International conference on machine learning. PMLR, pp 5180–5189
104. Skerry-Ryan R, Battenberg E, Xiao Y, Wang Y, Stanton D, Shor J, Weiss R, Clark R, Saurous RA (2018) Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron. In: International conference on machine learning. PMLR, pp 4693–4702
105. Chen M, Tan X, Li B, Liu Y, Qin T, Zhao S, Liu TY (2021) AdaSpeech: Adaptive text to speech for custom voice. In: International conference on learning representations.
106. Hayashi T, Watanabe S, Toda T, Takeda K, Toshniwal S, Livescu K (2019) Pre-trained text embeddings for enhanced text-to-speech synthesis. In: Proc Interspeech 2019, pp 4430–4434

107. Guo H, Soong FK, He L, Xie L (2019) Exploiting syntactic features in a parsed tree to improve end-to-end TTS. In: Proc Interspeech 2019, pp 4460–4464
108. Zhang G, Song K, Tan X, Tan D, Yan Y, Liu Y, Wang G, Zhou W, Qin T, Lee T, et al (2022) Mixed-Phoneme BERT: Improving BERT with mixed phoneme and sup-phoneme representations for text to speech. Preprint. arXiv:2203.17190
109. Liu R, Sisman B, Li H (2020) Graphspeech: Syntax-aware graph attention network for neural speech synthesis. Preprint. arXiv:2010.12423

Chapter 5
Acoustic Models

Abstract In this chapter, we introduce acoustic models, which generate acoustic features from linguistic features or directly from phonemes or characters. With the development of TTS, different kinds of acoustic models have been adopted, including the early hidden Markov models and deep neural networks in statistical parametric speech synthesis, and then the sequence-to-sequence models based on an encoder-attention-decoder framework (including RNN, CNN, and Transformer), and the latest feed-forward models (CNN or Transformer) and advanced generative models (GAN, Flow, VAE, and Diffusion).

Keywords Acoustic model · Tacotron · DeepVoice · FastSpeech · Generative model

In this chapter, we introduce acoustic models, which generate acoustic features from linguistic features or directly from phonemes or characters. With the development of TTS, different kinds of acoustic models have been adopted, including the early hidden Markov models and deep neural networks in statistical parametric speech synthesis (SPSS) [1–6], and then the sequence-to-sequence models based on an encoder-attention-decoder framework (including RNN, CNN, and Transformer) [7–10], and the latest feed-forward networks (CNN or Transformer) [11, 12] and advanced generative models (GAN, Flow, VAE, and Diffusion) [13–17].

Prerequisite Knowledge for Reading This Chapter

- Language and speech processing, such as waveform, mel-spectrogram, phoneme, pitch, duration.
- Model structures of deep neural networks, such as RNN, CNN, and Transformer.
- Generative models, such as Autoregressive Models, Normalizing Flows, Variational Auto-Encoders, Denoising Diffusion Probabilistic Models, and Generative Adversarial Networks.

X. Tan, *Neural Text-to-Speech Synthesis*, Artificial Intelligence: Foundations, Theory, and Algorithms, https://doi.org/10.1007/978-981-99-0827-1_5

5.1 Acoustic Models from a Historic Perspective

Acoustic models aim to generate acoustic features, which are further converted into waveforms using vocoders. The choice of acoustic features largely determines the types of TTS pipelines. Different kinds of acoustic features have been tried, such as mel-cepstral coefficients (MCC) [18], mel-generalized coefficients (MGC) [19], band aperiodicity (BAP) [20, 21], 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 TTS, which predict acoustic features such as mel-spectrograms from phonemes or characters.

5.1.1 Acoustic Models in SPSS

In statistical parametric speech synthesis (SPSS) [22, 23], statistical models such as HMM [1, 2], DNN [3, 4] or RNN [5, 6] 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 [24] or WORLD [25]. The developments of these acoustic models are driven by several requirements: (1) taking more context information as input; (2) modeling the correlation between output frames; (3) better combating the over-smoothing prediction problem [22] since the mapping from linguistic features to acoustic features is one-to-many. We briefly review some works as follows.

HMM [26] is leveraged to generate speech parameters in [1, 2], where the observation vectors of HMM consist of spectral parameter vectors such as mel-cepstral coefficients (MCC) and F0. Compared to concatenative speech synthesis, HMM-based parametric synthesis is more flexible in changing speaker identities, emotions, and speaking styles [2]. Readers can refer to [22, 23, 27] 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 [22, 23], mainly due to two reasons: (1) the accuracy of acoustic models is not good enough, and the predicted acoustic features are over-smoothing and a lack of details, and (2) the vocoding techniques are not good enough. The first reason is mainly due to a lack of modeling capacity in HMM. Thus, DNN-based acoustic models [3] 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 [5] are leveraged to better model the' context dependency. As the development of deep learning, some advanced network structures such as CBHG [7] are leveraged to better predict acoustic features [28]. VoiceLoop [29] adopts a working memory called phonological loop to generate acoustic features (e.g., F0, MGC, BAP) from phoneme sequence, and then uses a WORLD [25] vocoder to synthesize waveform

from this acoustic features. Yang et al. [30] leverage GAN [31] to improve the generation quality of acoustic features. Wang et al. [32] 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. [33] conduct thorough experimental studies on different acoustic models. Some acoustic models in SPSS are summarized in Table 5.1.

5.1.2 Acoustic Models in Neural TTS

Acoustic models enhanced with deep neural networks 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 cepstrum (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,[1] and provide a comprehensive list of acoustic models in Table 5.1.

5.2 Acoustic Models with Different Structures

Different model structures have been adopted in acoustic models, such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), and self-attention in Transformer. In this section, we introduce typical acoustic models with different model structures.

5.2.1 RNN-Based Models (e.g., Tacotron Series)

The inductive bias to use RNNs is that speech signals are inherently in sequential order. We introduce representative RNN-based acoustic models, such as the Tacotron series [7, 8].

[1] We mainly introduce acoustic models according to different network structures such as RNN, CNN, and Transformer (self-attention), while introducing vocoders in Chap. 6 according to different deep generative models (e.g., autoregressive models, normalizing flows, GANs, VAEs, diffusion models). However, it is not the only perspective, since acoustic models also leverage different deep generative models while vocoders also leverage different network structures.

Table 5.1 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 [18], "MGC" stands for mel-generalized coefficients [19], "BAP" stands for band aperiodicities [20, 21], "LSP" stands for line spectral pairs [34], "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 [1, 2]	Ling→MCC+F0	/	/	HMM
DNN-based [3]	Ling→MCC+BAP+F0	NAR	/	DNN
LSTM-based [5]	Ling→LSP+F0	AR	/	RNN
EMPHASIS [28]	Ling→LinS+CAP+F0	AR	/	Hybrid
ARST [32]	Ph→LSP+BAP+F0	AR	Seq2Seq	RNN
VoiceLoop [29]	Ph→MGC+BAP+F0	AR	/	Hybrid
Tacotron [7]	Ch→LinS	AR	Seq2Seq	Hybrid/RNN
Tacotron 2 [8]	Ch→MelS	AR	Seq2Seq	RNN
DurIAN [35]	Ph→MelS	AR	Seq2Seq	RNN
Non-Att Tacotron [36]	Ph→MelS	AR	/	Hybrid/CNN/RNN
MelNet [37]	Ch→MelS	AR	/	RNN
DeepVoice [38]	Ch/Ph→MelS	AR	/	CNN
DeepVoice 2 [39]	Ch/Ph→MelS	AR	/	CNN
DeepVoice 3 [9]	Ch/Ph→MelS	AR	Seq2Seq	CNN
ParaNet [12]	Ph→MelS	NAR	Seq2Seq	CNN
DCTTS [40]	Ch→MelS	AR	Seq2Seq	CNN
SpeedySpeech [41]	Ph→MelS	NAR	/	CNN
TalkNet 1/2 [42, 43]	Ch→MelS	NAR	/	CNN
TransformerTTS [10]	Ph→MelS	AR	Seq2Seq	Self-Att
MultiSpeech [44]	Ph→MelS	AR	Seq2Seq	Self-Att
FastSpeech 1/2 [11, 45]	Ph→MelS	NAR	Seq2Seq	Self-Att
AlignTTS [46]	Ch/Ph→MelS	NAR	Seq2Seq	Self-Att
JDI-T [47]	Ph→MelS	NAR	Seq2Seq	Self-Att
FastPitch [48]	Ph→MelS	NAR	Seq2Seq	Self-Att
AdaSpeech 1/2/3 [49–51]	Ph→MelS	NAR	Seq2Seq	Self-Att
DenoiSpeech [52]	Ph→MelS	NAR	Seq2Seq	Self-Att
DeviceTTS [53]	Ph→MelS	NAR	/	Hybrid/DNN/RNN
LightSpeech [54]	Ph→MelS	NAR	/	Hybrid/Self-Att/CNN
Flow-TTS [55]	Ch/Ph→MelS	NAR*	Flow	Hybrid/CNN/RNN
Glow-TTS [56]	Ph→MelS	NAR	Flow	Hybrid/Self-Att/CNN
Flowtron [57]	Ph→MelS	AR	Flow	Hybrid/RNN
EfficientTTS [58]	Ch→MelS	NAR	Flow	Hybrid/CNN
GMVAE-Tacotron [15]	Ph→MelS	AR	VAE	Hybrid/RNN

(continued)

Table 5.1 (continued)

Acoustic model	Input→Output	AR/NAR	Modeling	Structure
VAE-TTS [59]	Ph→MelS	AR	VAE	Hybrid/RNN
BVAE-TTS [60]	Ph→MelS	NAR	VAE	CNN
Para. Tacotron 1/2 [61, 62]	Ph→MelS	NAR	VAE	Hybrid/Self-Att/CNN
GAN exposure [63]	Ph→MelS	AR	GAN	Hybrid/RNN
TTS-Stylization [64]	Ch→MelS	AR	GAN	Hybrid/RNN
Multi-SpectroGAN [13]	Ph→MelS	NAR	GAN	Hybrid/Self-Att/CNN
Diff-TTS [65]	Ph→MelS	NAR*	Diffusion	Hybrid/CNN
Grad-TTS [66]	Ph→MelS	NAR	Diffusion	Hybrid/Self-Att/CNN
PriorGrad [67]	Ph→MelS	NAR	Diffusion	Hybrid/Self-Att/CNN

Tacotron

Tacotron [7] leverages an encoder-attention-decoder framework and takes characters (with text normalization) as input[2] and outputs linear-spectrograms, and uses Griffin-Lim algorithm [69] to generate the waveform. The model structure of Tacotron is shown in Fig. 5.1. The encoder consists of a Pre-net that pre-processes the character embedding into a sequence of hidden representations, and a CBHG module [7, 70] that transforms the hidden representations into the encoder output representations. Specifically, the CBHG module consists of (1) several 1D convolution filters with different kernel sizes k to emphasize k-gram in phoneme sequence, (2) a highway network [71] to extract high-level features, and (3) a bidirectional gated recurrent unit (GRU) [72] to generate final representations of the character sequence. The decoder consists of (1) a Pre-net to pre-process the mel-spectrogram sequence, (2) an attention RNN to attend to the encoder representations, (3) a decoder RNN to generate the mel-spectrogram sequence, and (4) a post-processing network using CBHG module to predict linear-spectrograms from mel-spectrograms. Finally, Tacotron leverages the Griffin-Lim algorithm [69] to synthesize waveform from linear-spectrograms. A useful trick is to predict multiple frames at each autoregressive step, which helps the learning of attention alignment [7].

[2] 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 a 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 mentioning that there are some works [9, 12, 68] using mixed representations of characters and phonemes as input to address the data sparsity problem.

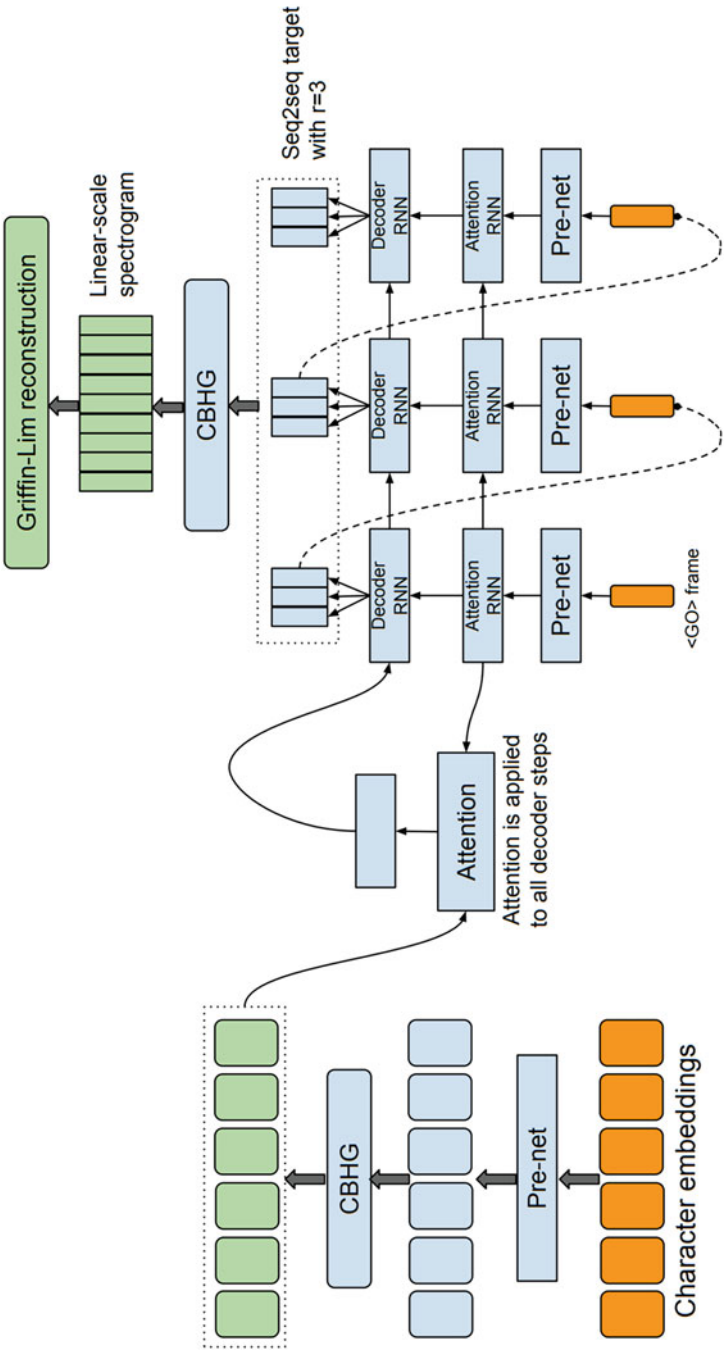


Fig. 5.1 The model structure of Tacotron. This figure is taken from [7] with permission

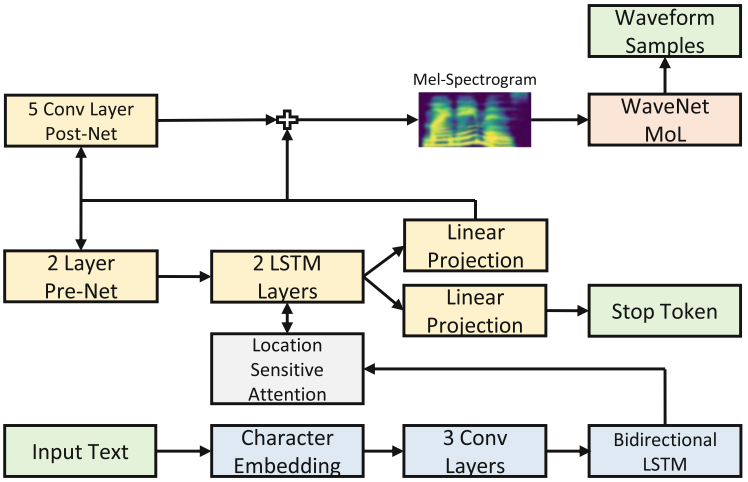


Fig. 5.2 The model structure of Tacotron 2. (Reproduced from [8])

Tacotron 2

Since the Griffin-Lim algorithm used in Tacotron leads to artifacts and worse quality than neural-based vocoders such as WaveNet[73], Tacotron 2 [8] is proposed to generate mel-spectrograms and convert mel-spectrograms into a waveform using WaveNet [73] model. As shown in Fig. 5.2, Tacotron 2 adopts an encoder-attention-decoder framework. The encoder consists of a Pre-net with 3 convolution layers to convert the character embeddings into hidden representations, and a bidirectional LSTM to generate the hidden representations of the encoder. The decoder consists of a Pre-net to pre-process mel-spectrograms and a 2-layer LSTM with a location-sensitive attention [74] to generate mel-spectrograms in an autoregressive way, with a Post-net to further enhance the quality of the predicted mel-spectrograms. Tacotron 2 greatly improves the voice quality over previous methods including concatenative TTS, parametric TTS, and neural TTS such as Tacotron.

Other Tacotron Related Acoustic Models

A lot of works improve Tacotron 1/2 from different aspects: (1) Using a reference encoder and style tokens to enhance the expressiveness of the speech synthesis, such as GST-Tacotron [75] and Ref-Tacotron [76]. (2) Removing the attention mechanism in Tacotron, and instead using a duration predictor for autoregressive prediction, such as DurIAN [35] and Non-Attentive Tacotron [36]. (3) Changing the autoregressive generation in Tacotron to non-autoregressive generation, such as Parallel Tacotron 1/2 [61, 62] which do not use RNN structure anymore. (4)

Building end-to-end text-to-waveform models based on Tacotron, such as Wave-Tacotron [77].

RNN-based acoustic models can well model the text and speech sequence into a sequential nature but suffer from both slow training and slow inference. To this end, parallel structures such as CNN and self-attention-based models are further leveraged in acoustic models.

5.2.2 CNN-Based Models (e.g., DeepVoice Series)

The inductive bias to use CNNs is that speech signals are locally dependent, i.e., a speech frame is mostly influenced by its adjacent speech frames.

DeepVoice [38] is actually an SPSS system enhanced with convolutional neural networks. After obtaining linguistic features through neural networks, DeepVoice leverages a WaveNet [73] based vocoder to generate the waveform. DeepVoice 2 [39] 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 [9] leverages a fully-convolutional network structure for speech synthesis as shown in Fig. 5.3, 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.

Besides the DeepVoice series, there are several other CNN-based acoustic models: (1) Based on DeepVoice 3, ClariNet [78] is proposed to generate waveform

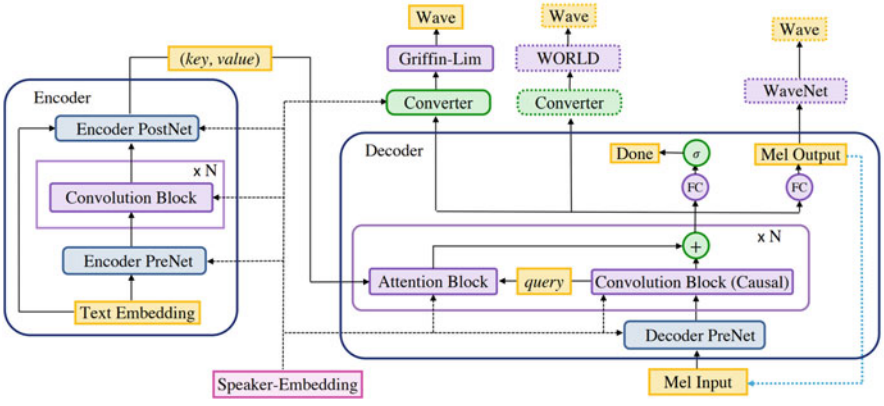


Fig. 5.3 The model structure of DeepVoice 3. This figure is taken from [9] with permission

from a text in a fully end-to-end way. (2) ParaNet [12] is a fully convolutional-based non-autoregressive model that can speed up the mel-spectrogram generation and obtain reasonably good speech quality. (3) DCTTS [40] shares a similar data conversion pipeline with Tacotron and leverages a fully convolutional-based encoder-attention-decoder network to generate mel-spectrograms from character sequences. It then uses a spectrogram super-resolution network to obtain linear-spectrograms and synthesizes waveform using Griffin-Lim [69].

5.2.3 Transformer-Based Models (e.g., FastSpeech Series)

Transformer [79] removes the sequential or local bias in RNNs and CNNs and models the sequence through the self-attention mechanism, where the attention weights are automatically learned from data. Transformer demonstrates strong capacity in sequence modeling.

TransformerTTS

TransformerTTS [10] (Fig. 5.4) leverages Transformer [79] based encoder-attention-decoder architecture to generate mel-spectrograms from phonemes. The authors in TransformerTTS [10] 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, RNNs are 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 to 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 attention in Transformer is not robust due to parallel computation. Thus, some works propose to enhance the robustness of Transformer-based acoustic models. For example, MultiSpeech [44] improves the robustness of the attention mechanism through several technologies including encoder normalization, decoder bottleneck, and diagonal attention constraint, and RobuTrans [80] leverages duration prediction to enhance the robustness in an autoregressive generation.

FastSpeech

Previous neural-based acoustic models such as Tacotron 1/2 [7, 8], DeepVoice 3 [9], and TransformerTTS [10] all adopt autoregressive generation, which suffer

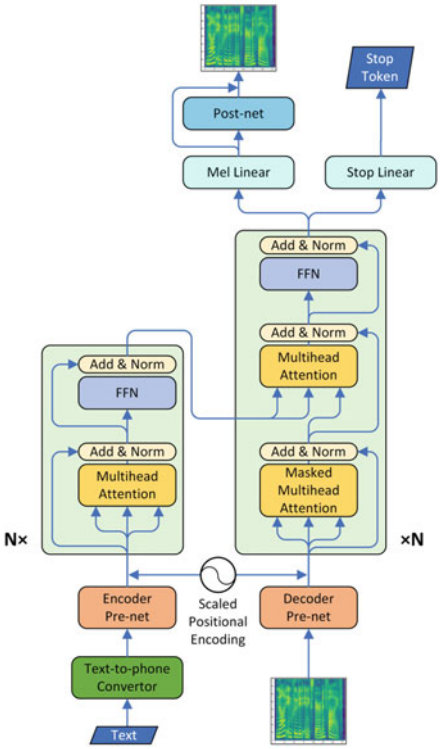


Fig. 5.4 The model structure of TransformerTTS. This figure is taken from [10] with permission

from several issues: (1) Slow inference speed. The autoregressive mel-spectrogram generation is slow, especially for long speech sequences (e.g., for a 5 s speech, there are nearly 500 frames of mel-spectrograms if hop size is 10 ms, which is a long sequence). (2) Robust issues. The generated speech usually has a lot of word skipping and repeating issues, which are mainly caused by the inaccurate attention alignments between text and mel-spectrogram sequences in an encoder-attention-decoder-based autoregressive generation. Thus, FastSpeech [11] (Fig. 5.5) 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[3] 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 the mel-spectrogram sequence and facilitate the parallel generation.

[3] 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 Chap. 9.

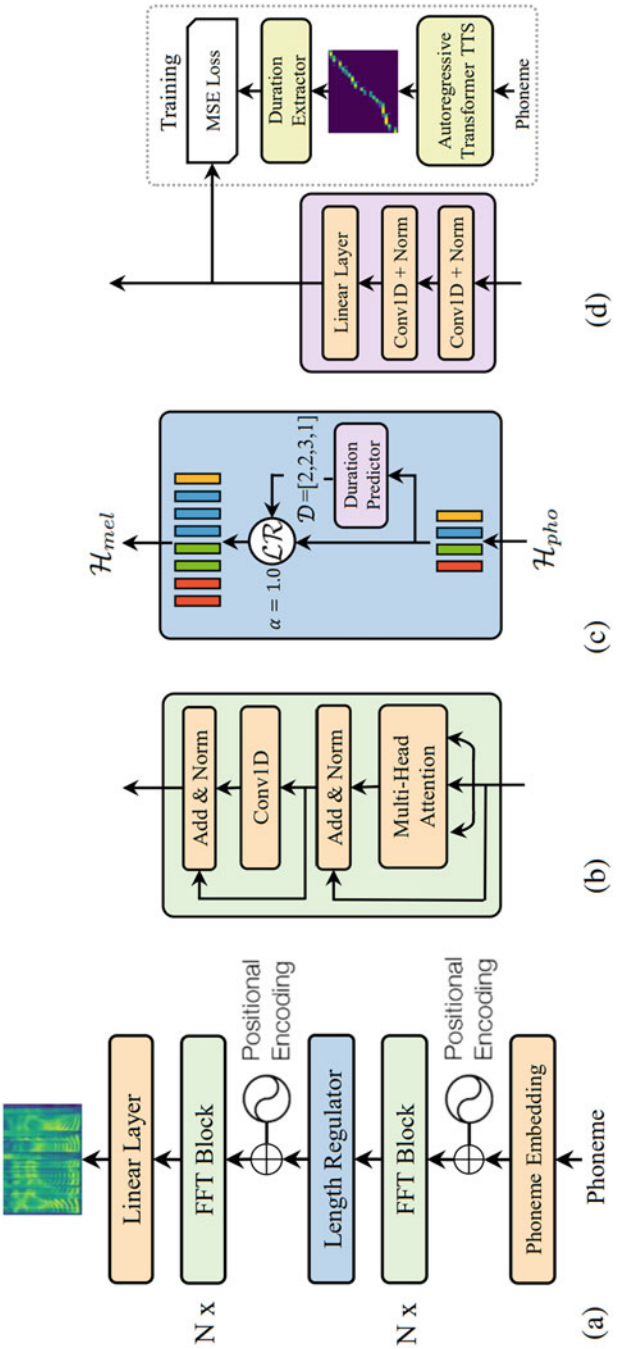


Fig. 5.5 The model structure of FastSpeech. This figure is taken from [11] with permission

FastSpeech enjoys several advantages [11]:[4] (1) extremely fast inference speed (e.g., 270× inference speedup on mel-spectrogram generation, 38× speedup on waveform generation), (2) robust speech synthesis without word skipping and repeating issues, and (3) on par or even better voice quality with previous autoregressive models.

FastSpeech 2

FastSpeech 2 [45] (Fig. 5.6) is proposed to further enhance FastSpeech, mainly from two aspects: (1) Using ground-truth mel-spectrograms as training targets, instead of the distilled mel-spectrograms from an autoregressive teacher model in FastSpeech. 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 [7, 81–83] in text-to-speech synthesis.[5] FastSpeech 2 achieves better voice quality than FastSpeech and maintains the advantages of fast, robust, and controllable speech synthesis in FastSpeech.[6]

To provide an overall comparison between different acoustic models, we summarize each component in the encoder and decoder in acoustic models as well as the vocoder used in each acoustic model in Table 5.2, and also summarize the time complexity of different model structures in training and inference in Table 5.3.

5.2.4 Advanced Generative Models (GAN/Flow/VAE/Diffusion)

Acoustic models learn the mapping between linguistic features (e.g., phoneme sequence) and acoustic features (e.g., mel-spectrogram sequence). The mapping between mel-spectrogram and phoneme sequences are one-to-many since multiple mel-spectrogram variations (e.g., different speed, pitches, volumes) can correspond to the same phoneme sequence. Thus, the mel-spectrogram data conditioned on

[4] FastSpeech has been deployed in Microsoft Azure Text-to-Speech Service (https://azure.microsoft.com/en-us/services/cognitive-services/text-to-speech/) to support all the languages and locales in Azure TTS.

[5] 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.

[6] FastSpeech 2s [45] is proposed together with FastSpeech 2. Since it is a fully end-to-end text-to-waveform model, we introduce it in Chap. 7.

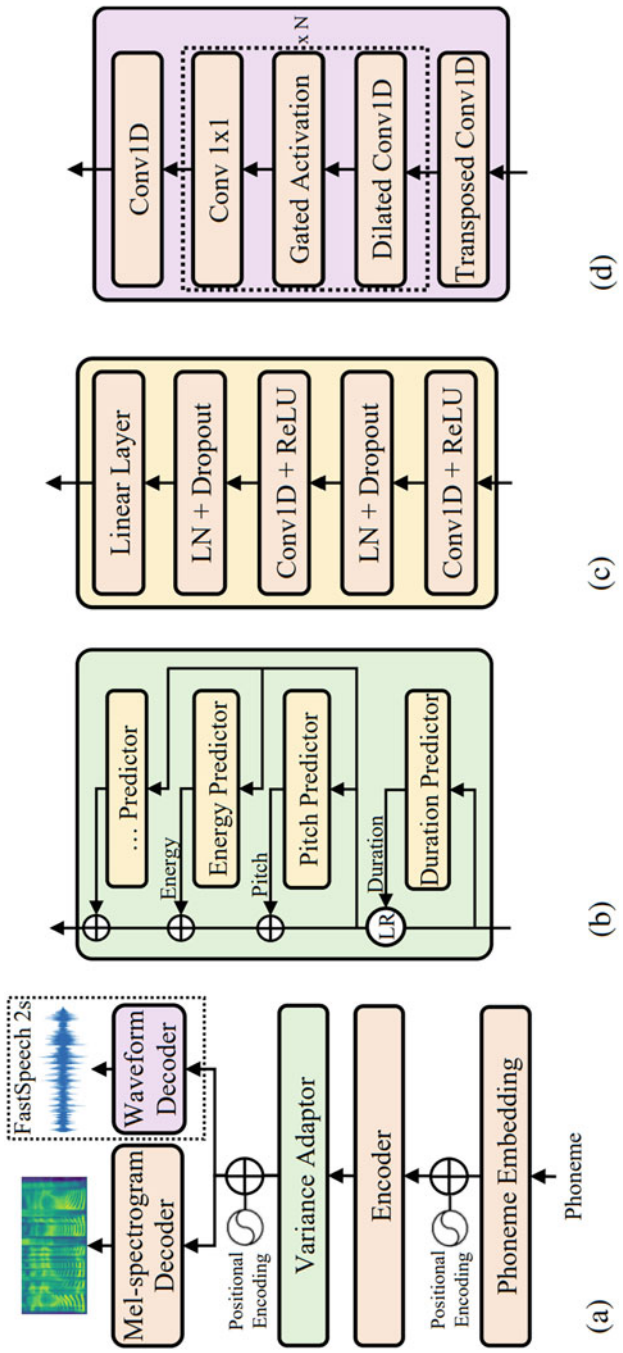


Fig. 5.6 The model structure of FastSpeech 2. This figure is taken from [45] with permission

Table 5.2 A summary of different components used in different acoustic models, where "Enc" and "Dec" denote encoder and decoder respectively, "Dense" denotes dense connection, "Att" denotes attention, "GL" denotes Griffin-Lim, "PWG" denotes Parallel WaveGAN, "N/A" means this acoustic model does not use this component

Component		Tacotron	Tacotron 2	Deep Voice 3	TransformerTTS	FastSpeech 1/2
Enc	Pre-net	Dense	Conv	Dense	Conv	N/A
	Enc	CBHG	Bi-LSTM	Conv	Self-Att	Self-Att
Dec	Pre-net	Dense	Dense	Dense	Dense	N/A
	Enc-Dec Att	Att	Att	Att	Att	Duration
	Dec	GRU	LSTM	Conv	Self-Att	Self-Att
	Post-net	CBHG	Conv	N/A	Conv	N/A
Vocoder		GL	WaveNet	GL/WaveNet	WaveNet	PWG/WaveGlow

Table 5.3 The time complexity of different model structures used in acoustic models in both training and inference with regard to sequence length N. We also list the interaction length (path length) between two elements/tokens in a sequence in different structures, where k is the kernel size of CNN

Structure	Acoustic model	Training	Inference	Path length
RNN	Tacotron 1/2	$O(N)$	$O(N)$	$O(N)$
CNN	DeepVoice 3	$O(1)$	$O(N)$	$O(\log_k N)$
Self-attention	TransformerTTS	$O(1)$	$O(N)$	$O(1)$
CNN (NAR)	ParaNet	$O(1)$	$O(1)$	$O(\log_k N)$
Self-attention (NAR)	FastSpeech 1/2	$O(1)$	$O(1)$	$O(1)$

phoneme is distributional-wise, instead of point-wise, and are usually modeled by generative models, such as autoregressive models, normalizing flows (Flow), variational auto-encoders (VAE), generative adversarial networks (GAN), and denoising diffusion probabilistic models (Diffusion). In this section, besides the basic generative models introduced in the previous subsections, we introduce some advanced generative models used in acoustic models, such as Flow, GAN, VAE, and Diffusion.[7] We just give a brief overview of the application of generative models

[7] Although we first introduce these advanced generative models used in acoustic models in this chapter and then introduced those used in vocoders in the next chapter, some generative models such as normalizing flows and diffusion models have been applied earlier on vocoders than in acoustic models (e.g., Parallel WaveNet [14] as a normalized flow-based model, WaveGrad/DiffWave [16, 17] as diffusion-based models). This trend is mainly due to several reasons: (1) Waveform sequence is extremely long and autoregressive generation is slow. Thus, exploring non-autoregressive generative models such as Flow, GAN, and Diffusion are more needed. (2) Simple L1/L2 loss can work quite well in mel-spectrogram prediction, and more efforts are taken to provide more variance information (such as pitch, duration, prosody) as input to solve the one-to-many mapping problem in acoustic models. However, L1/L2 loss does not work well in waveform prediction and there is not much variance information that can be leveraged in vocoders. Thus, advanced generative models are explored earlier on vocoders.

in acoustic models in this chapter. For the detailed formulation of deep generative models, please refer to Sect. 3.3.

GAN-Based Models

GANs have been widely used in acoustic models, such as GAN exposure [63], TTS-Stylization [64], and Multi-SpectroGAN [13]. Since simple L1/L2 loss can help the acoustic models to predict mel-spectrograms in a reasonable quality, GANs are usually used as an auxiliary loss to compensate for the L1/L2 mel-spectrogram loss to achieve better prediction quality.

Flow-Based Models

Normalizing flows have been leveraged in acoustic models to better model the conditional mel-spectrogram distributions. As introduced in Sect. 3.3.2, normalizing flows can be divided into two categories: autoregressive flows and bipartite flows. Accordingly, both types of flows are applied to acoustic models. Flowtron [57] is an autoregressive flow-based mel-spectrogram generation model, while Flow-TTS [55] and Glow-TTS [56] leverage bipartite flows for a non-autoregressive mel-spectrogram generation.

VAE-Based Models

Early work [75, 76] leverage reference encoders and style tokens to model the variance information from a reference speech, which can be regarded as an auto-encoder, and here the reference encoder is the encoder of the auto-encoder. Later, variational auto-encoders (VAEs) [84] are leveraged in acoustic models, such as GMVAE-Tacotron [15], VAE-TTS [59], BVAE-TTS [60], and Para. Tacotron 1/2 [61, 62]. Different from the standard unconditional VAEs as formulated in Eq. 3.10, the VAEs used in acoustic models are conditioned on source text or phoneme sequence, which are formulated as follows:

$$L(x;\theta,\phi) = -\mathbb{E}_{z\sim q(z|x;\phi)} \log p(x|z,y;\theta) + KL(q(z|x;\phi)||p(z)), \tag{5.1}$$

where y represents the text or phoneme sequence, θ represents the parameters of both the phoneme encoder and mel-spectrogram decoder, ϕ represents the parameters of the VAE encoder (posterior encoder), and $p(z)$ is the prior distribution, which is chosen as standard Gaussian distribution.

There is also another formulation of conditional VAEs for acoustic models. Instead of choosing standard Gaussian as the prior distribution, we leverage a prior encoder θ_{pri} to predict the prior distribution from the conditional phoneme sequence y. Thus, this kind of conditional VAE can be formulated as follows:

$$L(x;\theta_{\text{dec}},\theta_{\text{pri}},\phi) = -\mathbb{E}_{z\sim q(z|x;\phi)}\log p(x|z;\theta_{\text{dec}}) + KL(q(z|x;\phi)||p(z|y;\theta_{\text{pri}})), \tag{5.2}$$

where θ_{dec} represents the parameters of the mel-spectrogram decoder (VAE decoder), θ_{pri} represents the parameters of the prior encoder, ϕ represents the parameters of the VAE encoder (posterior encoder), and $p(z|y;\theta_{\text{pri}})$ is the prior distribution predicted from the phoneme sequence, instead of standard Gaussian distribution. Some models such as VITS [85] and NaturalSpeech [86] leverage this kind of conditional VAEs, which are introduced in Chap. 7.

Diffusion-Based Models

After diffusion models [87, 88] are first leveraged in vocoders [16, 17], a lot of works have applied diffusion models into acoustic models, such as Diff-TTS [65], Grad-TTS [66], and PriorGrad [67]. Diffusion models are good at generating high-quality mel-spectrograms with many fine-grained details, but at the cost of slow inference speed due to a large number of iteration steps. Thus, a lot of work has tried to reduce the number of iteration steps to speed up inference. For example, PriorGrad [67] provides a more informative prior distribution which is calculated from the conditional phoneme sequence and is closer to the mel-spectrogram distribution, which can reduce the difficulty of the data generation and thus result in faster training and inference.

References

1. Yoshimura T, Tokuda K, Masuko T, Kobayashi T, Kitamura T (1999) Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In: Sixth European conference on speech communication and technology
2. Tokuda K, Yoshimura T, Masuko T, Kobayashi T, Kitamura T (2000) Speech parameter generation algorithms for HMM-based speech synthesis. In: 2000 IEEE international conference on acoustics, speech, and signal processing. proceedings (Cat. No. 00CH37100), vol 3. IEEE, pp 1315–1318
3. Zen H, Senior A, Schuster M (2013) Statistical parametric speech synthesis using deep neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, pp 7962–7966
4. Qian Y, Fan Y, Hu W, Soong FK (2014) 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). IEEE, pp 3829–3833

5. Fan Y, Qian Y, Xie FL, Soong FK (2014) TTS synthesis with bidirectional LSTM based recurrent neural networks. In: Fifteenth annual conference of the international speech communication association
6. Zen H, Sak H (2015) 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). IEEE, pp 4470–4474
7. Wang Y, Skerry-Ryan R, Stanton D, Wu Y, Weiss RJ, Jaitly N, Yang Z, Xiao Y, Chen Z, Bengio S, et al (2017) Tacotron: towards end-to-end speech synthesis. In: Proc Interspeech 2017, pp 4006–4010
8. Shen J, Pang R, Weiss RJ, Schuster M, Jaitly N, Yang Z, Chen Z, Zhang Y, Wang Y, Skerry-Ryan R, et al (2018) Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4779–4783
9. Ping W, Peng K, Gibiansky A, Arik SO, Kannan A, Narang S, Raiman J, Miller J (2018) Deep Voice 3: 2000-speaker neural text-to-speech. In: Proc ICLR, pp 214–217
10. Li N, Liu S, Liu Y, Zhao S, Liu M (2019) Neural speech synthesis with transformer network. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 6706–6713
11. Ren Y, Ruan Y, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2019) FastSpeech: fast, robust and controllable text to speech. In: NeurIPS
12. Peng K, Ping W, Song Z, Zhao K (2020) Non-autoregressive neural text-to-speech. In: International conference on machine learning. PMLR, pp 7586–7598
13. Lee SH, Yoon HW, Noh HR, Kim JH, Lee SW (2020) Multi-SpectroGAN: high-diversity and high-fidelity spectrogram generation with adversarial style combination for speech synthesis. Preprint. arXiv:2012.07267
14. van den Oord A, Li Y, Babuschkin I, Simonyan K, Vinyals O, Kavukcuoglu K, Driessche G, Lockhart E, Cobo L, Stimberg F, et al (2018) Parallel WaveNet: fast high-fidelity speech synthesis. In: International conference on machine learning. PMLR, pp 3918–3926
15. Hsu WN, Zhang Y, Weiss RJ, Zen H, Wu Y, Wang Y, Cao Y, Jia Y, Chen Z, Shen J, et al (2018) Hierarchical generative modeling for controllable speech synthesis. In: International conference on learning representations
16. Kong Z, Ping W, Huang J, Zhao K, Catanzaro B (2021) DiffWave: a versatile diffusion model for audio synthesis. In: ICLR
17. Chen N, Zhang Y, Zen H, Weiss RJ, Norouzi M, Chan W (2021) WaveGrad: Estimating gradients for waveform generation. In: ICLR
18. Fukada T, Tokuda K, Kobayashi T, Imai S (1992) An adaptive algorithm for mel-cepstral analysis of speech. In: Proc. ICASSP, vol 1, pp 137–140
19. Tokuda K, Kobayashi T, Masuko T, Imai S (1994) Mel-generalized cepstral analysis-a unified approach to speech spectral estimation. In: Third international conference on spoken language processing
20. Kawahara H, Masuda-Katsuse I, De Cheveigne A (1999) 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 Commun 27(3–4):187–207
21. Kawahara H, Estill J, Fujimura O (2001) 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
22. Zen H, Tokuda K, Black AW (2009) Statistical parametric speech synthesis. Speech Commun 51(11):1039–1064
23. Tokuda K, Nankaku Y, Toda T, Zen H, Yamagishi J, Oura K (2013) Speech synthesis based on hidden Markov models. Proc IEEE 101(5):1234–1252
24. Kawahara H (2006) STRAIGHT, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds. Acoust Sci Technol 27(6):349–353
25. Morise M, Yokomori F, Ozawa K (2016) WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. IEICE Trans Inf Syst 99(7):1877–1884

26. Rabiner L, Juang B (1986) An introduction to hidden Markov models. IEEE ASSP Mag 3(1):4–16
27. Zen H (2015) Acoustic modeling in statistical parametric speech synthesis-from HMM to LSTM-RNN. In: Proc MLSLP. Invited paper
28. Li H, Kang Y, Wang Z (2018) EMPHASIS: an emotional phoneme-based acoustic model for speech synthesis system. In: Proc Interspeech 2018, pp 3077–3081
29. Taigman Y, Wolf L, Polyak A, Nachmani E (2018) VoiceLoop: voice fitting and synthesis via a phonological loop. In: International conference on learning representations
30. Yang S, Xie L, Chen X, Lou X, Zhu X, Huang D, Li H (2017) Statistical parametric speech synthesis using generative adversarial networks under a multi-task learning framework. In: 2017 IEEE automatic speech recognition and understanding workshop (ASRU). IEEE, pp 685–691
31. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: NIPS
32. Wang W, Xu S, Xu B (2016) First step towards end-to-end parametric TTS synthesis: generating spectral parameters with neural attention. In: Interspeech, pp 2243–2247
33. Wang X, Lorenzo-Trueba J, Takaki S, Juvela L, Yamagishi J (2018) 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). IEEE, pp 4804–4808
34. Itakura F (1975) Line spectrum representation of linear predictor coefficients of speech signals. J Acoust Soc Am 57(S1):S35–S35
35. Yu C, Lu H, Hu N, Yu M, Weng C, Xu K, Liu P, Tuo D, Kang S, Lei G, et al (2020) DurIAN: duration informed attention network for speech synthesis. In: Proc Interspeech 2020, pp 2027–2031
36. Shen J, Jia Y, Chrzanowski M, Zhang Y, Elias I, Zen H, Wu Y (2020) Non-attentive Tacotron: robust and controllable neural TTS synthesis including unsupervised duration modeling. Preprint. arXiv:2010.04301
37. Vasquez S, Lewis M (2019) MelNet: a generative model for audio in the frequency domain. Preprint. arXiv:1906.01083
38. Arık SÖ, Chrzanowski M, Coates A, Diamos G, Gibiansky A, Kang Y, Li X, Miller J, Ng A, Raiman J, et al (2017) Deep Voice: real-time neural text-to-speech. In: International conference on machine learning. PMLR, pp 195–204
39. Gibiansky A, Arik SÖ, Diamos GF, Miller J, Peng K, Ping W, Raiman J, Zhou Y (2017) Deep Voice 2: multi-speaker neural text-to-speech. In: NIPS
40. Tachibana H, Uenoyama K, Aihara S (2018) 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). IEEE, pp 4784–4788
41. Vainer J, Dušek O (2020) SpeedySpeech: efficient neural speech synthesis. In: Proc Interspeech 2020, pp 3575–3579
42. Beliaev S, Rebryk Y, Ginsburg B (2020) TalkNet: Fully-convolutional non-autoregressive speech synthesis model. Preprint. arXiv:2005.05514
43. Beliaev S, Ginsburg B (2021) TalkNet 2: Non-autoregressive depth-wise separable convolutional model for speech synthesis with explicit pitch and duration prediction. Preprint. arXiv:2104.08189
44. Chen M, Tan X, Ren Y, Xu J, Sun H, Zhao S, Qin T (2020) MultiSpeech: Multi-speaker text to speech with transformer. In: INTERSPEECH, pp 4024–4028
45. Ren Y, Hu C, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2021) FastSpeech 2: fast and high-quality end-to-end text to speech. In: International conference on learning representations. https://openreview.net/forum?id=piLPYqxtWuA
46. Zeng Z, Wang J, Cheng N, Xia T, Xiao J (2020) 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). IEEE, pp 6714–6718

47. Lim D, Jang W, Gyeonghwan O, Park H, Kim B, Yoon J (2020) JDI-T: Jointly trained duration informed transformer for text-to-speech without explicit alignment. In: Proc Interspeech 2020, pp 4004–4008
48. Łańcucki A (2020) FastPitch: parallel text-to-speech with pitch prediction. Preprint. arXiv:2006.06873
49. Chen M, Tan X, Li B, Liu Y, Qin T, sheng zhao, Liu TY (2021) AdaSpeech: adaptive text to speech for custom voice. In: International conference on learning representations. https://openreview.net/forum?id=Drynvt7gg4L
50. Yan Y, Tan X, Li B, Qin T, Zhao S, Shen Y, Liu TY (2021) AdaSpeech 2: Adaptive text to speech with untranscribed data. In: 2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
51. Yan Y, Tan X, Li B, Zhang G, Qin T, Zhao S, Shen Y, Zhang WQ, Liu TY (2021) AdaSpeech 3: adaptive text to speech for spontaneous style. In: INTERSPEECH
52. Zhang C, Ren Y, Tan X, Liu J, Zhang K, Qin T, Zhao S, Liu TY (2021) DenoiSpeech: denoising text to speech with frame-level noise modeling. In: 2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
53. Huang Z, Li H, Lei M (2020) DeviceTTS: A small-footprint, fast, stable network for on-device text-to-speech. Preprint. arXiv:2010.15311
54. Luo R, Tan X, Wang R, Qin T, Li J, Zhao S, Chen E, Liu TY (2021) LightSpeech: lightweight and fast text to speech with neural architecture search. In: 2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
55. Miao C, Liang S, Chen M, Ma J, Wang S, Xiao J (2020) 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). IEEE, pp 7209–7213
56. Kim J, Kim S, Kong J, Yoon S (2020) Glow-TTS: a generative flow for text-to-speech via monotonic alignment search. Adv Neural Inf Process Syst 33, 8067
57. Valle R, Shih K, Prenger R, Catanzaro B (2020) Flowtron: an autoregressive flow-based generative network for text-to-speech synthesis. Preprint. arXiv:2005.05957
58. Miao C, Liang S, Liu Z, Chen M, Ma J, Wang S, Xiao J (2020) EfficientTTS: an efficient and high-quality text-to-speech architecture. Preprint. arXiv:2012.03500
59. Zhang YJ, Pan S, He L, Ling ZH (2019) 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). IEEE, pp 6945–6949
60. Lee Y, Shin J, Jung K (2020) Bidirectional variational inference for non-autoregressive text-to-speech. In: International conference on learning representations
61. Elias I, Zen H, Shen J, Zhang Y, Jia Y, Weiss R, Wu Y (2020) Parallel Tacotron: non-autoregressive and controllable TTS. Preprint. arXiv:2010.11439
62. Elias I, Zen H, Shen J, Zhang Y, Ye J, Skerry-Ryan R, Wu Y (2021) Parallel Tacotron 2: a non-autoregressive neural TTS model with differentiable duration modeling. Preprint. arXiv:2103.14574
63. Guo H, Soong FK, He L, Xie L (2019) A new GAN-based end-to-end TTS training algorithm. In: Proc Interspeech 2019, pp 1288–1292
64. Ma S, Mcduff D, Song Y (2018) Neural TTS stylization with adversarial and collaborative games. In: International conference on learning representations
65. Jeong M, Kim H, Cheon SJ, Choi BJ, Kim NS (2021) Diff-TTS: a denoising diffusion model for text-to-speech. Preprint. arXiv:2104.01409
66. Popov V, Vovk I, Gogoryan V, Sadekova T, Kudinov M (2021) Grad-TTS: a diffusion probabilistic model for text-to-speech. Preprint. arXiv:2105.06337
67. Lee S, Kim H, Shin C, Tan X, Liu C, Meng Q, Qin T, Chen W, Yoon S, Liu TY (2021) PriorGrad: Improving conditional denoising diffusion models with data-driven adaptive prior. Preprint. arXiv:2106.06406
68. Kastner K, Santos JF, Bengio Y, Courville A (2019) Representation mixing for TTS synthesis. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 5906–5910

69. Griffin D, Lim J (1984) Signal estimation from modified short-time Fourier transform. IEEE Trans Acoust Speech Signal Process 32(2):236–243
70. Lee J, Cho K, Hofmann T (2017) Fully character-level neural machine translation without explicit segmentation. Trans Assoc Comput Linguist 5:365–378
71. Srivastava RK, Greff K, Schmidhuber J (2015) Highway networks. Preprint. arXiv:1505.00387
72. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. Preprint. arXiv:1412.3555
73. van den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K (2016) WaveNet: a generative model for raw audio. Preprint. arXiv:1609.03499
74. Chorowski J, Bahdanau D, Serdyuk D, Cho K, Bengio Y (2015) Attention-based models for speech recognition. In: Proceedings of the 28th international conference on neural information processing systems-volume 1, pp 577–585
75. Wang Y, Stanton D, Zhang Y, Skerry-Ryan R, Battenberg E, Shor J, Xiao Y, Jia Y, Ren F, Saurous RA (2018) Style tokens: unsupervised style modeling, control and transfer in end-to-end speech synthesis. In: International conference on machine learning. PMLR, pp 5180–5189
76. Skerry-Ryan R, Battenberg E, Xiao Y, Wang Y, Stanton D, Shor J, Weiss R, Clark R, Saurous RA (2018) Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron. In: International conference on machine learning. PMLR, pp 4693–4702
77. Weiss RJ, Skerry-Ryan R, Battenberg E, Mariooryad S, Kingma DP (2021) Wave-Tacotron: spectrogram-free end-to-end text-to-speech synthesis. In: 2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
78. Ping W, Peng K, Chen J (2018) ClariNet: parallel wave generation in end-to-end text-to-speech. In: International conference on learning representations
79. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
80. Li N, Liu Y, Wu Y, Liu S, Zhao S, Liu M (2020) RobuTrans: a robust transformer-based text-to-speech model. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 8228–8235
81. Jayne C, Lanitis A, Christodoulou C (2012) One-to-many neural network mapping techniques for face image synthesis. Expert Syst Appl 39(10):9778–9787
82. Gadermayr M, Tschuchnig M, Gupta L, Krämer N, Truhn D, Merhof D, Gess B (2021) 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). IEEE, pp 1182–1186
83. Zhu J-Y, Zhang R, Pathak D, Darrell T, Efros AA, Wang O, Shechtman E (2017) Toward multimodal image-to-image translation. In: Guyon I, von Luxburg U, Bengio S, Wallach HM, Fergus R, Vishwanathan SVN, Garnett R (eds) Advances in neural information processing systems 30: Annual conference on neural information processing systems 2017, December 4–9, 2017, Long Beach, CA, USA, pp 465–476. https://proceedings.neurips.cc/paper/2017/hash/819f46e52c25763a55cc642422644317-Abstract.html
84. Kingma DP, Welling M (2013) Auto-encoding variational bayes. Preprint. arXiv:1312.6114
85. Kim J, Kong J, Son J (2021) Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. Preprint. arXiv:2106.06103
86. Tan X, Chen J, Liu H, Cong J, Zhang C, Liu Y, Wang X, Leng Y, Yi Y, He L, et al (2022) NaturalSpeech: end-to-end text to speech synthesis with human-level quality. Preprint. arXiv:2205.04421
87. Sohl-Dickstein J, Weiss E, Maheswaranathan N, Ganguli S (2015) Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. PMLR, pp 2256–2265
88. Ho J, Jain A, Abbeel P (2020) Denoising diffusion probabilistic models. Preprint. arXiv:2006.11239

Chapter 6
Vocoders

Abstract In this chapter, we introduce vocoders, which generate waveforms from acoustic features or directly from linguistic features. With the development of TTS, different kinds of vocoders have been adopted, including the vocoders in statistical parametric speech synthesis (SPSS), and neural network-based vocoders. We first view vocoders from a historic perspective, covering vocoders in SPSS and neural TTS, and then introduce the vocoders in neural TTS, mainly from the perspective of different deep generative models used.

Keywords Vocoder · Autoregressive model · Normalizing flow · GAN · Diffusion model

In this chapter, we introduce vocoders, which generate waveforms from acoustic features or directly from linguistic features. With the development of TTS, different kinds of vocoders have been adopted, including signal processing-based vocoders [1–3], and neural network-based vocoders [4–8]. We first view vocoders from a historic perspective, and then introduce the vocoders in neural TTS, mainly from the perspective of different deep generative models used.

Prerequisite Knowledge for Reading This Chapter

- Language and speech processing, such as waveform, mel-spectrogram.
- Model structures of deep neural networks, such as RNN, and CNN.
- Deep generative models, such as Autoregressive Models, Normalizing Flows, Variational Auto-Encoders, Denoising Diffusion Probabilistic Models, and Generative Adversarial Networks.

6.1 Vocoders from a Historic Perspective

Roughly speaking, the development of vocoders can be categorized into two stages: the vocoders using signal processing technologies [1–3, 9, 10], and the vocoders using neural networks [4–8].

X. Tan, *Neural Text-to-Speech Synthesis*, Artificial Intelligence: Foundations, Theory, and Algorithms, https://doi.org/10.1007/978-981-99-0827-1_6

6.1.1 Vocoders in Signal Processing

Some popular vocoders using signal processing include STRAIGHT [1], WORLD [2], and Griffin-Lim algorithm [11]. 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 a spectral envelope, aperiodicity, and F0, which can be further postprocessed to extract their lower-dimensional representation such as mel-cepstrum [12] and band aperiodicity [13, 14] to be modeled by acoustic models. In vocoder synthesis, it generates speech waveforms from these acoustic features.

6.1.2 Vocoders in Neural TTS

Early neural vocoders such as WaveNet [4], Parallel WaveNet [21], WaveRNN [6] directly take linguistic features as input and generate the waveform. Later, [7, 8, 27, 28] take mel-spectrograms as input and generate the waveform. Since speech waveform is very long, autoregressive waveform generation takes much inference time. Thus, other deep generative models (as introduced in Sect. 3.3) such as normalizing flows (Flow) [38–40], generative adversarial networks (GAN) [41], variational auto-encoders (VAE) [42], and denoising diffusion probabilistic model (DDPM or Diffusion for short) [43, 44] 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 neural vocoders in Table 6.1 and introduce some typical vocoders in the following sections.

6.2 Vocoders with Different Generative Models

6.2.1 Autoregressive Vocoders (e.g., WaveNet)

WaveNet [4] is the first neural-based vocoder, which leverages dilated convolution to generate waveform points autoregressively, as shown in Fig. 6.1. Unlike the vocoder analysis and synthesis in signal processing [1, 2, 12, 13, 45, 46], 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 [47, 48], generate speech waveform conditioned on linguistic features, while WaveNet can be easily adapted to condition on linear-spectrograms [48] and mel-spectrograms [49–51]. Although WaveNet achieves good voice quality, it suffers from slow inference speed. Therefore, a lot of works [15, 52, 53] investigate lightweight and fast vocoders. SampleRNN [15]

Table 6.1 A list of neural vocoders and their corresponding characteristics

Vocoder	Input	AR/NAR	Modeling	Architecture
WaveNet [4]	Linguistic feature	AR	/	CNN
SampleRNN [15]	/	AR	/	RNN
WaveRNN [6]	Linguistic feature	AR	/	RNN
LPCNet [16]	BFCC	AR	/	RNN
Univ. WaveRNN [17]	Mel-Spectrogram	AR	/	RNN
SC-WaveRNN [18]	Mel-Spectrogram	AR	/	RNN
MB WaveRNN [19]	Mel-Spectrogram	AR	/	RNN
FFTNet [20]	Cepstrum	AR	/	CNN
Par. WaveNet [21]	Linguistic feature	NAR	Flow	CNN
WaveGlow [7]	Mel-Spectrogram	NAR	Flow	Hybrid/CNN
FloWaveNet [8]	Mel-Spectrogram	NAR	Flow	Hybrid/CNN
WaveFlow [22]	Mel-Spectrogram	AR	Flow	Hybrid/CNN
SqueezeWave [23]	Mel-Spectrogram	NAR	Flow	CNN
WaveGAN [24]	/	NAR	GAN	CNN
GELP [25]	Mel-Spectrogram	NAR	GAN	CNN
GAN-TTS [26]	Linguistic feature	NAR	GAN	CNN
MelGAN [27]	Mel-Spectrogram	NAR	GAN	CNN
Par. WaveGAN [28]	Mel-Spectrogram	NAR	GAN	CNN
HiFi-GAN [29]	Mel-Spectrogram	NAR	GAN	Hybrid/CNN
VocGAN [30]	Mel-Spectrogram	NAR	GAN	CNN
GED [31]	Linguistic feature	NAR	GAN	CNN
Fre-GAN [32]	Mel-Spectrogram	NAR	GAN	CNN
Wave-VAE [33]	Mel-Spectrogram	NAR	VAE	CNN
WaveGrad [34]	Mel-Spectrogram	NAR	Diffusion	Hybrid/CNN
DiffWave [35]	Mel-Spectrogram	NAR	Diffusion	Hybrid/CNN
PriorGrad [36]	Mel-Spectrogram	NAR	Diffusion	Hybrid/CNN
SpecGrad [37]	Mel-Spectrogram	NAR	Diffusion	Hybrid/CNN

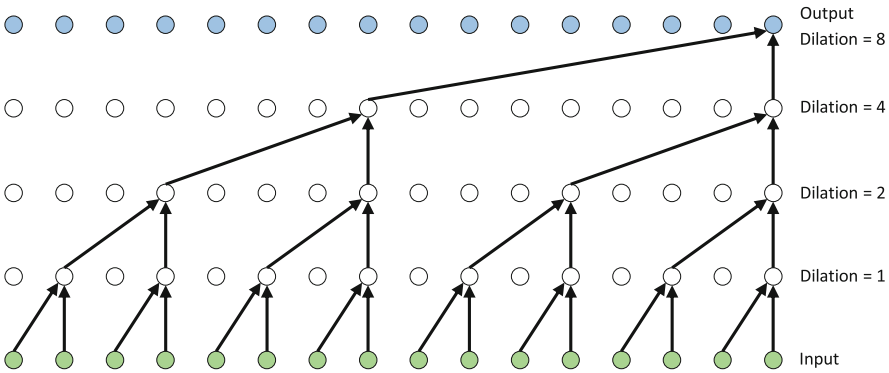


Fig. 6.1 Illustration of dilated convolution in WaveNet [4]. (Reproduced from [4])

leverages a hierarchical recurrent neural network for unconditional waveform generation, and it is further integrated into Char2Wav [5] to generate waveform conditioned on acoustic features. Further, WaveRNN [54] 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. [17], Paul et al. [18], and Jiao et al. [55] further improve the robustness and universality of the vocoders. LPCNet [16, 56] 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 mel-spectrograms. Some following works further improve LPCNet from different perspectives, such as reducing complexity for speedup [57–59], and improving stability for better quality [60].

6.2.2 *Flow-Based Vocoders (e.g., Parallel WaveNet, WaveGlow)*

Normalizing flows [38–40, 61, 62] are a kind of generative model. It transforms a probability density with a sequence of invertible mappings [62]. Since we can get a standard/normalized probability distribution (e.g., Gaussian) through the sequence of invertible mappings based on the change-of-variables rules, this kind of flow-based generative model is called a normalizing flow. During sampling, it generates data from a standard probability distribution through the inverse of these transforms. As described in Sect. 3.3.2 and shown in Table 6.2, there are two categories of normalizing flows according to the two different techniques [63]: (1) autoregressive transforms [39] (e.g., inverse autoregressive flow used in Parallel WaveNet [21]), and (2) bipartite transforms (e.g., Glow [40] used in WaveGlow [7], and RealNVP [61] used in FloWaveNet [8]).

- Autoregressive transforms, e.g., inverse autoregressive flow (IAF) [39]. IAF can be regarded as a dual formulation of autoregressive flow (AF) [64, 65]. 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 [21] leverages probability density distillation

Table 6.2 Several representative flow-based models and their formulations [22]

Flow		Evaluation $z = f^{-1}(x)$	Synthesis $x = f(z)$
AR	AF [64]	$z_t = x_t \cdot \sigma_t(x_{<t};\theta) + \mu_t(x_{<t};\theta)$	$x_t = \frac{z_t - u_t(x_{<t};\theta)}{\sigma_t(x_{<t};\theta)}$
	IAF [39]	$z_t = \frac{x_t - \mu_t(z_{<t};\theta)}{\sigma_t(z_{<t};\theta)}$	$x_t = z_t \cdot \sigma_t(z_{<t};\theta) + \mu_t(z_{<t};\theta)$
Bipartite	RealNVP [61]	$z_a = x_a,$	$x_a = z_a,$
	Glow [40]	$z_b = x_b \cdot \sigma_b(x_a;\theta) + \mu_b(x_a;\theta)$	$x_b = \frac{z_b - \mu_b(x_a;\theta)}{\sigma_b(x_a;\theta)}$

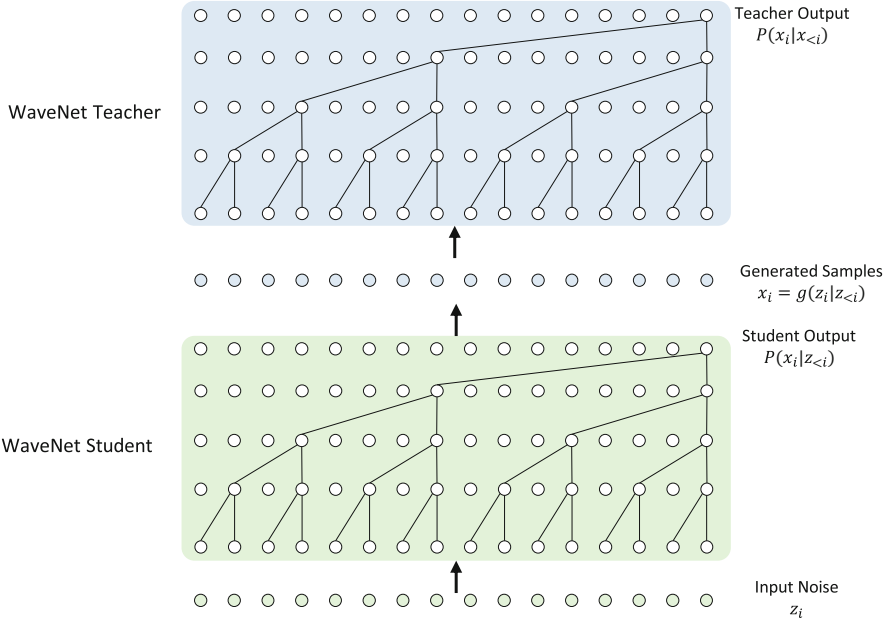


Fig. 6.2 The teacher-student knowledge distillation used in Parallel WaveNet [21]. (Reproduced from [21])

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, as shown in Fig. 6.2. Similarly, ClariNet [66] 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 [40] or RealNVP [61]. To ensure the transforms are 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 [7] and FloWaveNet [8], which achieve high voice quality and fast inference speed. An illustration of the affine coupling layer used in WaveGlow is shown in Fig. 6.3, where the $\sigma_b(x_a; \theta)$ and $\mu_b(x_a; \theta)$ in the last line of Table 6.2 are implemented with a WaveNet based model structure, denoted as "WN" in Fig. 6.3.

Both autoregressive and bipartite transforms have their advantages and disadvantages [22]: (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

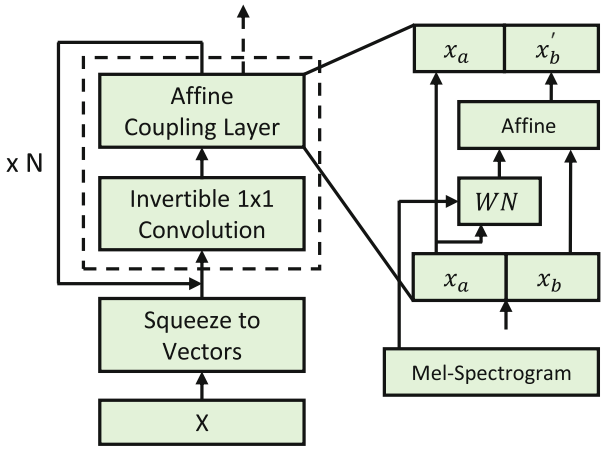


Fig. 6.3 Illustration of the affine coupling layer used in WaveGlow [7], where "WN" denotes a WaveNet based model, and "N" is the number of flow layers. (Reproduced from [7])

training. (2) Bipartite transforms enjoy a much simpler training pipeline, but usually require a 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 [22] 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.

6.2.3 GAN-Based Vocoders (e.g., MelGAN, HiFiGAN)

Generative adversarial networks (GANs) [41] have been widely used in data generation tasks, such as image generation [41, 67], text generation [68], and audio generation [24]. As introduced in Sect. 3.3.6, GAN consists of a generator for data generation, and a discriminator to judge the authenticity of the generated data and is optimized with an adversarial loss function, formulated as follows:

$$\min_{\theta} \max_{\phi} \mathbb{E}_{x \sim p_{\text{data}}} \log D(x; \phi) + \mathbb{E}_{x \sim p_z} \log(1 - D(G(z; \theta); \phi)), \tag{6.1}$$

where θ and ϕ denote the parameter of generator and discriminator respectively, and p_{data} and p_z denote the true data distribution and standard Gaussian distribution.

A lot of vocoders leverage GAN to ensure the audio generation quality, including WaveGAN [24], GAN-TTS [26], MelGAN [27], Parallel WaveGAN [28], HiFi-GAN [29], and other GAN-based vocoders [69–74]. We summarize the

Table 6.3 Several representative GAN-based vocoders and their characteristics

GAN	Generator	Discriminator	Loss
WaveGAN [24]	DCGAN [80]	/	WGAN-GP [75]
GAN-TTS [26]	/	Random window D	Hinge-loss GAN [76]
MelGAN [27]	/	Multi-Scale D	LS-GAN [77] Feature Matching Loss [79]
Par.WaveGAN [28]	WaveNet [4]	/	LS-GAN, Multi-STFT Loss
HiFi-GAN [29]	Multi-Receptive Field Fusion	Multi-Period D, Multi-Scale D	LS-GAN, STFT Loss, Feature Matching Loss
VocGAN [30]	Multi-Scale G	Hierarchical D	LS-GAN, Multi-STFT Loss, Feature Matching Loss
GED [31]	/	Random Window D	Hinge-Loss GAN, Repulsive loss

characteristics according to the generators, discriminators, and losses used in each vocoder in Table 6.3, and introduce them accordingly.

- 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. [28] 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 in a larger computation cost. Therefore, some vocoders [27, 29] choose to iteratively upsample the condition information and perform dilated convolution, which can avoid too long a sequence in the lower layers. Specifically, VocGAN [30] proposes a multi-scale generator that can gradually output waveform sequences at different scales, from coarse-grained to fine-grained. HiFi-GAN [29] 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. Some efforts [26, 27, 29, 30] on discriminators focus on how to design models to capture the characteristics of a waveform, in order to provide a better guiding signal for the generators. We introduce these efforts as follows: (1) Random window discriminators, proposed in GAN-TTS [26], 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 ways, simplifying the true/false judgments compared with full audio, and acting as a data augmentation effect, etc. (2) Multi-scale discriminators, proposed in MelGAN [27], 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 [29], which leverage multiple discriminators, where each accepts equally spaced samples of 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 input audio in different periods. (4) Hierarchical discriminators, leveraged in VocGAN [30] 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 [75], hinge-loss GAN [76], and LS-GAN [77], other specific losses such as STFT loss [69, 78] and feature matching loss [79] are also leveraged. These additional losses can improve the stability and efficiency of adversarial training [28], and improve the perceptual audio quality. Gritsenko et al. [31] propose a generalized energy distance with a repulsive term to better capture the multi-modal waveform distribution.

6.2.4 *Diffusion-Based Vocoders (e.g., WaveGrad, DiffWave)*

Denoising diffusion probabilistic models (DDPM or Diffusion) [44] are leveraged in vocoders, such as WaveGrad [34], DiffWave [35]. As introduced in Sect. 3.3.4, the basic idea of the diffusion model is to formulate the mapping between data and latent distributions with a diffusion process and a reverse process, as shown in Fig. 6.4: in the diffusion process, the waveform data sample is gradually added with some random noises and finally becomes Gaussian noise; in the reverse process, the random Gaussian 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 [36, 37, 81–83] are investigating how to reduce

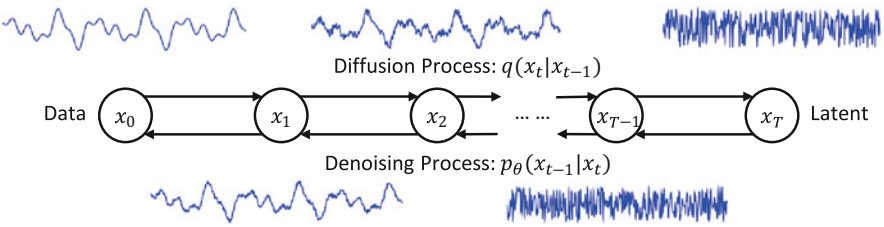


Fig. 6.4 The diffusion and denoising processes in diffusion-based vocoders [34, 35]. (Reproduced from [34])

inference time while maintaining generation quality. For example, PriorGrad [36] and SpecGrad [37] reduce the number of iterations by providing a more informative prior distribution that is close to the data distribution. They have a nice connection between signal processing and neural network-based methods by incorporating signal processing knowledge into the model. InferGrad [84] reduces the mismatch between training and inference in the diffusion model (i.e., a large iteration step is used in training while a small iteration step in inference) by taking the inference process into training, which can achieve good voice quality with a small number of iteration steps.

As shown in Sects. 3.3.4 and 3.3.5, we introduce several kinds of generative models, including denoising diffusion probabilistic models (Diffusion) [43, 44], score matching with Langevin dynamics (SMLG) [85], stochastic differential equation (SDE) [86], and ordinary differential equation (ODE) [86]. Actually, they have very close connections: (1) Diffusion and SMLG have very similar formulations and only differ in scale in their loss functions [87]. (2) SDE extends the discrete time step in Diffusion and SMLG to continuous time variables and formulates the diffusion and denoising processes as stochastic differential equations. (3) ODE is a corresponding deterministic process of SDE and has the same marginal probability densities in the diffusion and denoising trajectories. These generative models have been used in the literature of speech synthesis: DiffWave [35] is based on the formulation of the original diffusion model, while WaveGrad [34] is based on the formulation of both diffusion model and score matching with Langevin dynamics. In this chapter, we just use the term "diffusion model" to represent these generative models in speech synthesis.

6.2.5 Other Vocoders

Some works leverage neural-based source-filter model for waveform generation [25, 88–95], aiming to achieve high voice quality while maintaining controllable speech generation. Govalkar et al. [96] conduct a comprehensive study on different kinds of vocoders. Hsu et al. [97] study the robustness of vocoders by evaluating several common vocoders with comprehensive experiments.

References

1. Kawahara H (2006) STRAIGHT, exploitation of the other aspect of vocoder: perceptually isomorphic decomposition of speech sounds. Acoust Sci Technol 27(6):349–353
2. Morise M, Yokomori F, Ozawa K (2016) WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. IEICE Trans Inf Syst 99(7):1877–1884

3. Ai Y, Ling ZH (2020) A neural vocoder with hierarchical generation of amplitude and phase spectra for statistical parametric speech synthesis. IEEE/ACM Trans Audio Speech Lang Process 28:839–851
4. van den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K (2016) WaveNet: a generative model for raw audio. Preprint. arXiv:1609.03499
5. Sotelo J, Mehri S, Kumar K, Santos JF, Kastner K, Courville A, Bengio Y (2017) Char2wav: end-to-end speech synthesis
6. Kalchbrenner N, Elsen E, Simonyan K, Noury S, Casagrande N, Lockhart E, Stimberg F, Oord A, Dieleman S, Kavukcuoglu K (2018) Efficient neural audio synthesis. In: International conference on machine learning. PMLR, pp 2410–2419
7. Prenger R, Valle R, Catanzaro B (2019) WaveGlow: a flow-based generative network for speech synthesis. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 3617–3621
8. Kim S, Lee SG, Song J, Kim J, Yoon S (2019) FloWaveNet: A generative flow for raw audio. In: International conference on machine learning. PMLR, pp 3370–3378
9. Imai S (1983) Cepstral analysis synthesis on the mel frequency scale. In: ICASSP'83. IEEE international conference on acoustics, speech, and signal processing. IEEE, vol 8, pp 93–96
10. Yoshimura T, Tokuda K, Masuko T, Kobayashi T, Kitamura T (2001) Mixed excitation for HMM-based speech synthesis. In: Seventh European conference on speech communication and technology
11. Griffin D, Lim J (1984) Signal estimation from modified short-time Fourier transform. IEEE Trans Acoust Speech Signal Process 32(2):236–243
12. Fukada T, Tokuda K, Kobayashi T, Imai S (1992) An adaptive algorithm for mel-cepstral analysis of speech. In: Proc. ICASSP, vol 1, pp 137–140
13. Kawahara H, Masuda-Katsuse I, De Cheveigne A (1999) 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 Commun 27(3–4):187–207
14. Kawahara H, Estill J, Fujimura O (2001) 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
15. Mehri S, Kumar K, Gulrajani I, Kumar R, Jain S, Sotelo J, Courville A, Bengio Y (2017) SampleRNN: an unconditional end-to-end neural audio generation model. In: ICLR
16. Valin JM, Skoglund J (2019) LPCNet: Improving neural speech synthesis through linear prediction. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 5891–5895
17. Lorenzo-Trueba J, Drugman T, Latorre J, Merritt T, Putrycz B, Barra-Chicote R, Moinet A, Aggarwal V (2019) Towards achieving robust universal neural vocoding. In: Proc Interspeech 2019, pp 181–185
18. Paul D, Pantazis Y, Stylianou Y (2020) Speaker conditional WaveRNN: towards universal neural vocoder for unseen speaker and recording conditions. In: Proc Interspeech 2020, pp 235–239
19. Yu C, Lu H, Hu N, Yu M, Weng C, Xu K, Liu P, Tuo D, Kang S, Lei G, et al (2020) DurIAN: Duration informed attention network for speech synthesis. In: Proc Interspeech 2020, pp 2027–2031
20. Jin Z, Finkelstein A, Mysore GJ, Lu J (2018) FFTNet: A real-time speaker-dependent neural vocoder. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 2251–2255
21. van den Oord A, Li Y, Babuschkin I, Simonyan K, Vinyals O, Kavukcuoglu K, Driessche G, Lockhart E, Cobo L, Stimberg F, et al. (2018) Parallel WaveNet: fast high-fidelity speech synthesis. In: International conference on machine learning. PMLR, pp 3918–3926
22. Ping W, Peng K, Zhao K, Song Z (2020) WaveFlow: a compact flow-based model for raw audio. In: International conference on machine learning. PMLR, pp 7706–7716

23. Zhai B, Gao T, Xue F, Rothchild D, Wu B, Gonzalez JE, Keutzer K (2020) SqueezeWave: extremely lightweight vocoders for on-device speech synthesis. Preprint. arXiv:200105685
24. Donahue C, McAuley J, Puckette M (2018) Adversarial audio synthesis. In: International conference on learning representations
25. Juvela L, Bollepalli B, Yamagishi J, Alku P (2019) GELP: GAN-excited linear prediction for speech synthesis from mel-spectrogram. In: Proc Interspeech 2019, pp 694–698
26. Bińkowski M, Donahue J, Dieleman S, Clark A, Elsen E, Casagrande N, Cobo LC, Simonyan K (2019) High fidelity speech synthesis with adversarial networks. In: International conference on learning representations
27. Kumar K, Kumar R, de Boissiere T, Gestin L, Teoh WZ, Sotelo J, de Brébisson A, Bengio Y, Courville A (2019) MelGAN: generative adversarial networks for conditional waveform synthesis. In: NeurIPS
28. Yamamoto R, Song E, Kim JM (2020) 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). IEEE, pp 6199–6203
29. Kong J, Kim J, Bae J (2020) HiFi-GAN: generative adversarial networks for efficient and high fidelity speech synthesis. Adv Neural Inf Process Syst 33, 17022
30. Yang J, Lee J, Kim Y, Cho HY, Kim I (2020) VocGAN: a high-fidelity real-time vocoder with a hierarchically-nested adversarial network. In: Proc Interspeech 2020, pp 200–204
31. Gritsenko A, Salimans T, van den Berg R, Snoek J, Kalchbrenner N (2020) A spectral energy distance for parallel speech synthesis. Adv Neural Inf Process Syst 33:13062
32. Kim JH, Lee SH, Lee JH, Lee SW (2021) Fre-GAN: Adversarial frequency-consistent audio synthesis. Preprint. arXiv:2106.02297
33. Peng K, Ping W, Song Z, Zhao K (2020) Non-autoregressive neural text-to-speech. In: International conference on machine learning. PMLR, pp 7586–7598
34. Chen N, Zhang Y, Zen H, Weiss RJ, Norouzi M, Chan W (2021) WaveGrad: Estimating gradients for waveform generation. In: ICLR
35. Kong Z, Ping W, Huang J, Zhao K, Catanzaro B (2021) DiffWave: a versatile diffusion model for audio synthesis. In: ICLR
36. Lee Sg, Kim H, Shin C, Tan X, Liu C, Meng Q, Qin T, Chen W, Yoon S, Liu TY (2021) PriorGrad: improving conditional denoising diffusion models with data-driven adaptive prior. Preprint. arXiv:2106.06406
37. Koizumi Y, Zen H, Yatabe K, Chen N, Bacchiani M (2022) SpecGrad: diffusion probabilistic model based neural vocoder with adaptive noise spectral shaping. Preprint. arXiv:2203.16749
38. Dinh L, Krueger D, Bengio Y (2014) NICE: Non-linear independent components estimation. Preprint. arXiv:14108516
39. Kingma DP, Salimans T, Jozefowicz R, Chen X, Sutskever I, Welling M (2016) Improved variational inference with inverse autoregressive flow. Adv Neural Inf Process Syst 29:4743–4751
40. Kingma DP, Dhariwal P (2018) Glow: generative flow with invertible 1×1 convolutions. In: Proceedings of the 32nd international conference on neural information processing systems, pp 10236–10245
41. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: NIPS
42. Kingma DP, Welling M (2013) Auto-encoding variational bayes. Preprint. arXiv:1312.6114
43. Sohl-Dickstein J, Weiss E, Maheswaranathan N, Ganguli S (2015) Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. PMLR, pp 2256–2265
44. Ho J, Jain A, Abbeel P (2020) Denoising diffusion probabilistic models. Preprint. arXiv:2006.11239
45. Tokuda K, Kobayashi T, Masuko T, Imai S (1994) Mel-generalized cepstral analysis-a unified approach to speech spectral estimation. In: Third international conference on spoken language processing

46. Itakura F (1975) Line spectrum representation of linear predictor coefficients of speech signals. J Acoust Soc Am 57(S1):S35–S35
47. Arık SÖ, Chrzanowski M, Coates A, Diamos G, Gibiansky A, Kang Y, Li X, Miller J, Ng A, Raiman J, et al (2017) Deep Voice: real-time neural text-to-speech. In: International conference on machine learning. PMLR, pp 195–204
48. Gibiansky A, Arik SÖ, Diamos GF, Miller J, Peng K, Ping W, Raiman J, Zhou Y (2017) Deep Voice 2: multi-speaker neural text-to-speech. In: NIPS
49. Tamamori A, Hayashi T, Kobayashi K, Takeda K, Toda T (2017) Speaker-dependent WaveNet vocoder. In: Interspeech, vol 2017, pp 1118–1122
50. Ping W, Peng K, Gibiansky A, Arik SO, Kannan A, Narang S, Raiman J, Miller J (2018) Deep Voice 3: 2000-speaker neural text-to-speech. In: Proc ICLR, pp 214–217
51. Shen J, Pang R, Weiss RJ, Schuster M, Jaitly N, Yang Z, Chen Z, Zhang Y, Wang Y, Skerry-Ryan R, et al (2018) Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4779–4783
52. Paine TL, Khorrami P, Chang S, Zhang Y, Ramachandran P, Hasegawa-Johnson MA, Huang TS (2016) Fast WaveNet generation algorithm. Preprint. arXiv:1611.09482
53. Hsu Pc, Lee Hy (2020) WG-WaveNet: real-time high-fidelity speech synthesis without GPU. In: Proc Interspeech 2020, pp 210–214
54. Zhang ZR, Chu M, Chang E (2002) An efficient way to learn rules for grapheme-to-phoneme conversion in Chinese. In: International symposium on Chinese spoken language processing
55. Jiao Y, Gabrys A, Tinchev G, Putrycz B, Korzekwa D, Klimkov V (2021) Universal neural vocoding with parallel WaveNet. Preprint. arXiv:2102.01106
56. Valin JM, Skoglund J (2019) A real-time wideband neural vocoder at 1.6 kb/s using LPCNet. In: Proc Interspeech 2019, pp 3406–3410
57. Vipperla R, Park S, Choo K, Ishtiaq S, Min K, Bhattacharya S, Mehrotra A, Ramos AGC, Lane ND (2020) Bunched LPCNet: vocoder for low-cost neural text-to-speech systems. In: Proc Interspeech 2020, pp 3565–3569
58. Popov V, Kudinov M, Sadekova T (2020) Gaussian LPCNet for multisample speech synthesis. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 6204–6208
59. Kanagawa H, Ijima Y (2020) Lightweight LPCNet-based neural vocoder with tensor decomposition. In: Proc Interspeech 2020, pp 205–209
60. Hwang MJ, Song E, Yamamoto R, Soong F, Kang HG (2020) 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). IEEE, pp 7219–7223
61. Dinh L, Sohl-Dickstein J, Bengio S (2016) Density estimation using real NVP. Preprint. arXiv:160508803
62. Rezende D, Mohamed S (2015) Variational inference with normalizing flows. In: International conference on machine learning. PMLR, pp 1530–1538
63. Papamakarios G, Nalisnick E, Rezende DJ, Mohamed S, Lakshminarayanan B (2019) Normalizing flows for probabilistic modeling and inference. Preprint. arXiv:1912.02762
64. Papamakarios G, Pavlakou T, Murray I (2017) Masked autoregressive flow for density estimation. In: Proceedings of the 31st international conference on neural information processing systems, pp 2335–2344
65. Huang CW, Krueger D, Lacoste A, Courville A (2018) Neural autoregressive flows. In: International conference on machine learning. PMLR, pp 2078–2087
66. Ping W, Peng K, Chen J (2018) ClariNet: Parallel wave generation in end-to-end text-to-speech. In: International conference on learning representations
67. Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision, pp 2223–2232

68. Yu L, Zhang W, Wang J, Yu Y (2017) SeGAN: sequence generative adversarial nets with policy gradient. In: Proceedings of the AAAI conference on artificial intelligence, vol 31
69. Yamamoto R, Song E, Kim JM (2019) Probability density distillation with generative adversarial networks for high-quality parallel waveform generation. In: Proc Interspeech 2019, pp 699–703
70. Wu YC, Hayashi T, Okamoto T, Kawai H, Toda T (2020) Quasi-periodic parallel WaveGAN vocoder: a non-autoregressive pitch-dependent dilated convolution model for parametric speech generation. In: Proc Interspeech 2020, pp 3535–3539
71. Song E, Yamamoto R, Hwang MJ, Kim JS, Kwon O, Kim JM (2021) Improved parallel WaveGAN vocoder with perceptually weighted spectrogram loss. In: 2021 IEEE spoken language technology workshop (SLT). IEEE, pp 470–476
72. You J, Kim D, Nam G, Hwang G, Chae G (2021) GAN vocoder: multi-resolution discriminator is all you need. Preprint. arXiv:2103.05236
73. Wang C, Chen Y, Wang B, Shi Y (2021) Improve GAN-based neural vocoder using pointwise relativistic leastsquare GAN. Preprint. arXiv:210314245
74. Jang W, Lim D, Yoon J (2020) Universal MelGAN: a robust neural vocoder for high-fidelity waveform generation in multiple domains. Preprint. arXiv:2011.09631
75. Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville A (2017) Improved training of wasserstein GANs. In: Proceedings of the 31st international conference on neural information processing systems, pp 5769–5779
76. Lim JH, Ye JC (2017) Geometric GAN. Preprint. arXiv:1705.02894
77. Mao X, Li Q, Xie H, Lau RY, Wang Z, Paul Smolley S (2017) Least squares generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision, pp 2794–2802
78. Arık SÖ, Jun H, Diamos G (2018) Fast spectrogram inversion using multi-head convolutional neural networks. IEEE Signal Process Lett 26(1):94–98
79. Larsen ABL, Sønderby SK, Larochelle H, Winther O (2016) Autoencoding beyond pixels using a learned similarity metric. In: International conference on machine learning. PMLR, pp 1558–1566
80. Radford A, Metz L, Chintala S (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. Preprint. arXiv:1511.06434
81. Song J, Meng C, Ermon S (2020) Denoising diffusion implicit models. Preprint. arXiv:201002502
82. Watson D, Ho J, Norouzi M, Chan W (2021) Learning to efficiently sample from diffusion probabilistic models. Preprint. arXiv:210603802
83. Kong Z, Ping W (2021) On fast sampling of diffusion probabilistic models. Preprint. arXiv:2106.00132
84. Chen Z, Tan X, Wang K, Pan S, Mandic D, He L, Zhao S (2022) InferGrad: Improving diffusion models for vocoder by considering inference in training. In: ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 8432–8436
85. Song Y, Ermon S (2019) Generative modeling by estimating gradients of the data distribution. In: Wallach HM, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox EB, Garnett R (eds) Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada, pp 11895–11907. https://proceedings.neurips.cc/paper/2019/hash/3001ef257407d5a371a96dcd947c7d93-Abstract.html
86. Song Y, Sohl-Dickstein J, Kingma DP, Kumar A, Ermon S, Poole B (2020) Score-based generative modeling through stochastic differential equations. In: International conference on learning representations
87. Luo C (2022) Understanding diffusion models: a unified perspective. Preprint. arXiv:2208.11970

88. Wang X, Takaki S, Yamagishi J (2019) 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). IEEE, pp 5916–5920
89. Wang X, Takaki S, Yamagishi J (2019) Neural source-filter waveform models for statistical parametric speech synthesis. IEEE/ACM Trans Audio Speech Lang Process 28:402–415
90. Wang X, Yamagishi J (2019) Neural harmonic-plus-noise waveform model with trainable maximum voice frequency for text-to-speech synthesis. In: Proc. 10th ISCA Speech Synthesis Workshop, pp 1–6
91. Liu Z, Chen K, Yu K (2020) Neural homomorphic vocoder. In: Proc Interspeech 2020, pp 240–244
92. Juvela L, Bollepalli B, Tsiaras V, Alku P (2019) GlotNet – a raw waveform model for the glottal excitation in statistical parametric speech synthesis. IEEE/ACM Trans Audio Speech Lang Process 27(6):1019–1030
93. Engel J, Gu C, Roberts A, et al (2019) DDSP: differentiable digital signal processing. In: International conference on learning representations
94. Song E, Hwang MJ, Yamamoto R, Kim JS, Kwon O, Kim JM (2020) Neural text-to-speech with a modeling-by-generation excitation vocoder. In: Proc Interspeech 2020, pp 3570–3574
95. Yoneyama R, Wu YC, Toda T (2021) Unified source-filter GAN: unified source-filter network based on factorization of quasi-periodic parallel WaveGAN. Preprint. arXiv:210404668
96. Govalkar P, Fischer J, Zalkow F, Dittmar C (2019) A comparison of recent neural vocoders for speech signal reconstruction. In: Proc. 10th ISCA speech synthesis workshop, pp 7–12
97. Hsu Pc, Wang Ch, Liu AT, Lee Hy (2019) Towards robust neural vocoding for speech generation: a survey. Preprint. arXiv:191202461

Chapter 7
Fully End-to-End TTS

Abstract Fully end-to-end TTS models can generate speech waveforms from character or phoneme sequences directly. However, there are big challenges to training 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. Thus, the development of end-to-end TTS is progressive. In this chapter, we first review the progressively end-to-end process in TTS from a historical perspective and then introduce some fully end-to-end TTS models.

Keywords End-to-end TTS · Two-stage training · One-stage training · NaturalSpeech

Fully end-to-end TTS models can generate speech waveform from character or phoneme sequence directly, which have the following advantages: (1) They require 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) They can also reduce the training, development, and deployment cost.

However, there are big challenges to training 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 s 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 16 kHz). 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. Thus, the development of end-to-end TTS is progressive. In this chapter, we first review the progressively end-to-end process in TTS from a historical perspective and then introduce some fully end-to-end TTS models.

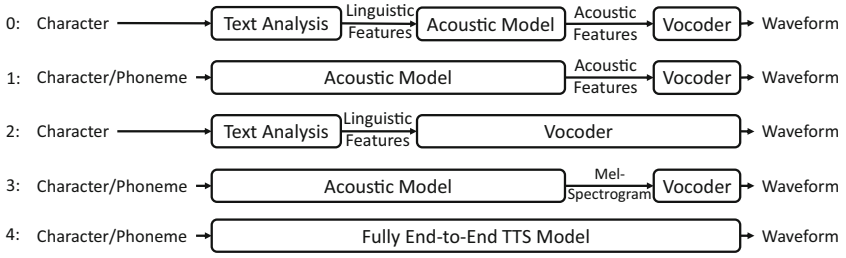
X. Tan, *Neural Text-to-Speech Synthesis*, Artificial Intelligence: Foundations, Theory, and Algorithms, https://doi.org/10.1007/978-981-99-0827-1_7

7.1 Prerequisite Knowledge for Reading This Chapter

- Language and speech processing.
- Deep generative models, such as Autoregressive Models, Normalizing Flows, Variational Auto-Encoders, Denoising Diffusion Probabilistic Models, and Generative Adversarial Networks.
- The key components in neural TTS, including acoustic models and vocoders.

7.2 End-to-End TTS from a Historic Perspective

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 7.1 illustrates this progressive process starting from early statistical parametric synthesis [1–5]. The process towards fully end-to-end models typically contains these upgrades: (1) Simplifying text analysis module and linguistic features. In SPSS, the text analysis module contains different functionalities such as text normalization, phrase/word/syllable segmentation, POS tagging, prosody prediction, and 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 grapheme-to-phoneme conversion 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



Stage	Models
0	SPSS [1, 2, 3, 4, 5]
1	ARST [6]
2	WaveNet [7], DeepVoice 1/2 [8, 9], Par. WaveNet [10], WaveRNN [11], HiFi-GAN [12]
3	DeepVoice 3 [13], Tacotron 2 [14], FastSpeech 1/2 [15, 16], WaveGlow [17], FloWaveNet [18]
4	Char2Wav [19], ClariNet [20], FastSpeech 2s [16], EATS [21], VITS [22], NaturalSpeech [23]

Fig. 7.1 The progressively end-to-end process for TTS models

and vocoders can be replaced with a single vocoder model such as WaveNet. Accordingly, we illustrate the progressive process in Fig. 7.1 and describe it as follows.

7.2.1 *Stage 0: Character→Linguistic→Acoustic→Waveform*

Statistical parametric synthesis [1–5] uses three basic modules, where text analyses convert 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.

7.2.2 *Stage 1: Character/Phoneme→Acoustic→Waveform*

ARST [6] in statistical parametric synthesis combines 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 the waveform.

7.2.3 *Stage 2: Character→Linguistic→Waveform*

Some works [24–26] in SPSS propose to directly generate speech waveform from linguistic features. Later, WaveNet [7] learns the mapping between linguistic features and speech waveform with a deep neural network, which can be regarded as a combination of an acoustic model and a vocoder. This kind of model [7, 10–12] still requires a text analysis module to generate linguistic features.

7.2.4 *Stage 3: Character/Phoneme→Spectrogram→Waveform*

Tacotron [27] 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 [28]. The following works such as DeepVoice 3 [13], Tacotron 2 [14], TransformerTTS [29], and FastSpeech 1/2 [15, 16] predict mel-spectrograms from characters/phonemes and further use a neural vocoder such as WaveNet [7], WaveRNN [11], WaveGlow [17], FloWaveNet [18], and Parallel WaveGAN [30] to generate the waveform.

Table 7.1 A list of fully end-to-end TTS models

Model	One-stage training	AR/NAR	Modeling	Architecture
Char2Wav [19]	N	AR	Seq2Seq	RNN
ClariNet [20]	N	AR	Flow	CNN
FastSpeech 2s [16]	Y	NAR	GAN	Self-Att/CNN
EATS [21]	Y	NAR	GAN	CNN
Wave-Tacotron [31]	Y	AR	Flow	CNN/RNN/Hybrid
EfficientTTS-Wav [32]	Y	NAR	GAN	CNN
WaveGrad 2 [33]	Y	NAR	Diffusion	CNN/RNN/Hybrid
VITS [22]	Y	NAR	VAE+Flow+GAN	CNN/Self-Att/Hybrid
NaturalSpeech [23]	Y	NAR	VAE+Flow+GAN	CNN/Self-Att/Hybrid

7.2.5 Stage 4: Character/Phoneme→Waveform

In recent years, some fully end-to-end TTS models are developed for direct text-to-waveform synthesis, as listed in Table 7.1. We will introduce these models in the next section.

7.3 Fully End-to-End Models

In this section, we introduce some representative fully end-to-end TTS models from several aspects: (1) two-stage training which first trains acoustic models and/or vocoder separately and then jointly optimizes them. (2) Using generative models for end-to-end training such as GAN, Flow, or Diffusion. (3) Hybrid system that combines several different generative models including VAE, Flow, and GAN. (4) A fully end-to-end TTS system that achieves human-level quality.

7.3.1 Two-Stage Training (e.g., Char2Wav, ClariNet)

In the early investigation, researchers usually cascade acoustic models and vocoders together and optimize them jointly to build fully end-to-end models, such as Char2Wav and Clarinet. Char2Wav [19] leverages an RNN-based encoder-attention-decoder model to generate acoustic features from characters and then uses SampleRNN [34] to generate the waveform. The two models are jointly tuned for direct speech synthesis. Similarly, ClariNet [20] jointly tunes an autoregressive acoustic model and a non-autoregressive vocoder for direct waveform generation.

7.3.2 One-Stage Training (e.g., FastSpeech 2s, EATS, VITS)

FastSpeech 2s [16] directly generates 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 sequences. A concurrent work called EATS [21] 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 [31] builds a flow-based decoder on Tacotron to directly generate a waveform, which uses parallel waveform generation in the flow part but still an autoregressive generation in the Tacotron part. WaveGrad 2 [33] leverages a diffusion model to iteratively denoise the waveform conditioned on a phoneme encoder. VITS [22] leverages a VAE to reconstruct the waveform sequence and uses a GAN-based loss to help optimize the waveform generation. It also uses leverage a Glow-TTS [35] based module to predict the prior distribution from the phoneme sequence to fulfill phoneme to waveform generation.

7.3.3 Human-Level Quality (e.g., NaturalSpeech)

NaturalSpeech [23] is the first TTS model that achieves comparable voice quality with human recordings on a benchmarking dataset (i.e., LJSpeech [36]). As shown in Fig. 7.2, NaturalSpeech leverages a variational auto-encoder (VAE) [37] to compress the high-dimensional speech (x) into continuous frame-level representations (denoted as posterior $q(z|x)$), which are used to reconstruct the waveform (denoted as $p(x|z)$). The corresponding prior (denoted as $p(z|y)$) is obtained from the text

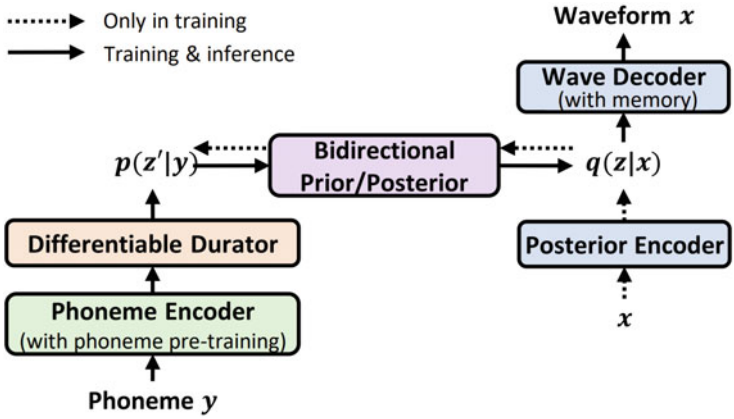


Fig. 7.2 The model structure of NaturalSpeech. This figure is taken from [23] with permission

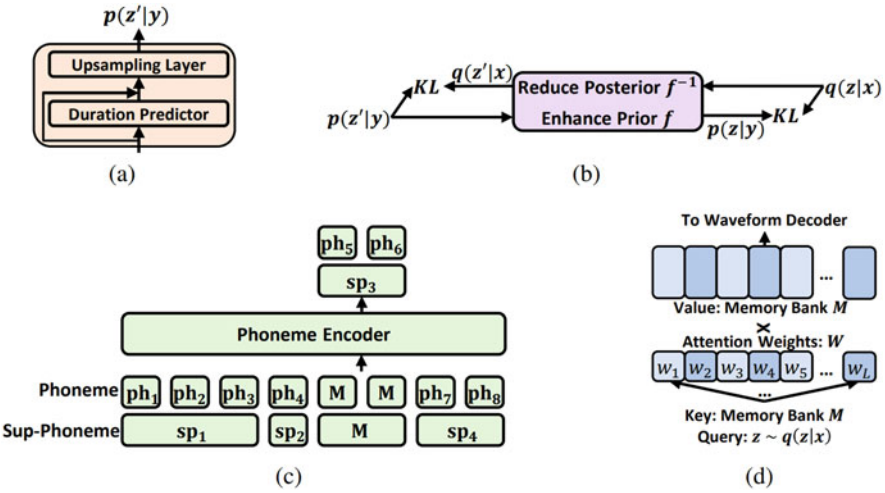


Fig. 7.3 The modules designed in NaturalSpeech. This figure is taken from [23] with permission. (**a**) Differentiable durator. (**b**) Bidirectional prior/posterior. (**c**) Phoneme pre-training. (**d**) Memory mechanism in VAE

sequence y. Considering the posterior from the speech is more complicated than the prior from text, NaturalSpeech designs several modules (see Fig. 7.3) to match the posterior and prior as close to each other as possible, to enable text-to-speech synthesis through $p(z|y) \rightarrow p(x|z)$: (1) a large-scale pre-training on the phoneme encoder to extract better representations from phoneme sequence; (2) a fully differentiable durator [23] that consists of a duration predictor and an upsampling layer to improve the duration modeling; (3) a bidirectional prior/posterior module based on flow models [38–40] to further enhance the prior $p(z|y)$ and reduce the complexity of posterior $q(z|x)$; and (4) a memory-based VAE to reduce the complexity of the posterior needed to reconstruct the waveform. NaturalSpeech achieves a CMOS of -0.01 compared to recordings, with a Wilcoxon signed rank test [41] at p-level $p \gg 0.05$, which shows that NaturalSpeech generates speech with no statistically significant difference from human recordings.

References

1. Yoshimura T, Tokuda K, Masuko T, Kobayashi T, Kitamura T (1999) Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In: sixth european conference on speech communication and technology
2. Tokuda K, Yoshimura T, Masuko T, Kobayashi T, Kitamura T (2000) Speech parameter generation algorithms for HMM-based speech synthesis. In: 2000 IEEE international conference on acoustics, speech, and signal processing. proceedings (Cat. No. 00CH37100), vol 3. IEEE, pp 1315–1318

3. Yoshimura T (2002) Simultaneous modeling of phonetic and prosodic parameters, and characteristic conversion for HMM-based text-to-speech systems. Ph.D Dissertation, Nagoya Institute of Technology
4. Zen H, Tokuda K, Black AW (2009) Statistical parametric speech synthesis. Speech Commun 51(11):1039–1064
5. Tokuda K, Nankaku Y, Toda T, Zen H, Yamagishi J, Oura K (2013) Speech synthesis based on hidden Markov models. Proc IEEE 101(5):1234–1252
6. Wang W, Xu S, Xu B (2016) First step towards end-to-end parametric TTS synthesis: Generating spectral parameters with neural attention. In: Interspeech, pp 2243–2247
7. van den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K (2016) WaveNet: a generative model for raw audio. Preprint. arXiv:1609.03499
8. Arık SÖ, Chrzanowski M, Coates A, Diamos G, Gibiansky A, Kang Y, Li X, Miller J, Ng A, Raiman J, et al. (2017) Deep voice: real-time neural text-to-speech. In: International conference on machine learning (PMLR), pp 195–204
9. Gibiansky A, Arik SÖ, Diamos GF, Miller J, Peng K, Ping W, Raiman J, Zhou Y (2017) Deep voice 2: multi-speaker neural text-to-speech. In: NIPS
10. van den Oord A, Li Y, Babuschkin I, Simonyan K, Vinyals O, Kavukcuoglu K, Driessche G, Lockhart E, Cobo L, Stimberg F et al (2018) Parallel WaveNet: fast high-fidelity speech synthesis. In: International conference on machine learning (PMLR), pp 3918–3926
11. Kalchbrenner N, Elsen E, Simonyan K, Noury S, Casagrande N, Lockhart E, Stimberg F, van den Oord A, Dieleman S, Kavukcuoglu K (2018) Efficient neural audio synthesis. In: International conference on machine learning (PMLR), pp 2410–2419
12. Bińkowski M, Donahue J, Dieleman S, Clark A, Elsen E, Casagrande N, Cobo LC, Simonyan K (2019) High fidelity speech synthesis with adversarial networks. In: International conference on learning representations
13. Ping W, Peng K, Gibiansky A, Arik SO, Kannan A, Narang S, Raiman J, Miller J (2018) Deep voice 3: 2000-speaker neural text-to-speech. In: Proceedings of the International conference on learning representation, pp 214–217
14. Shen J, Pang R, Weiss RJ, Schuster M, Jaitly N, Yang Z, Chen Z, Zhang Y, Wang Y, Skerry-Ryan R et al (2018) Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4779–4783
15. Ren Y, Ruan Y, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2019) FastSpeech: fast, robust and controllable text to speech. In: NeurIPS
16. Ren Y, Hu C, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2021) FastSpeech 2: fast and high-quality end-to-end text to speech. In: International conference on learning representations. https://openreview.net/forum?id=piLPYqxtWuA
17. Prenger R, Valle R, Catanzaro B (2019) WaveGlow: a flow-based generative network for speech synthesis. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 3617–3621
18. Kim S, Lee SG, Song J, Kim J, Yoon S (2019) FloWaveNet: a generative flow for raw audio. In: International conference on machine learning (PMLR), pp 3370–3378
19. Sotelo J, Mehri S, Kumar K, Santos JF, Kastner K, Courville AC, Bengio Y (2017) Char2wav: end-to-end speech synthesis. In: 5th international conference on learning representations, ICLR 2017, Toulon, France, April 24–26, 2017, Workshop Track Proceedings. OpenReview.net. https://openreview.net/forum?id=B1VWyySKx
20. Ping W, Peng K, Chen J (2018) ClariNet: parallel wave generation in end-to-end text-to-speech. In: International conference on learning representations
21. Donahue J, Dieleman S, Bińkowski M, Elsen E, Simonyan K (2021) End-to-end adversarial text-to-speech. In: International conference on learning representations
22. Kim J, Kong J, Son J (2021) Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. Preprint. arXiv:2106.06103

23. Tan X, Chen J, Liu H, Cong J, Zhang C, Liu Y, Wang X, Leng Y, Yi Y, He L et al (2022) NaturalSpeech: end-to-end text to speech synthesis with human-level quality. Preprint. arXiv:2205.04421
24. Maia R, Zen H, Gales MJ (2010) Statistical parametric speech synthesis with joint estimation of acoustic and excitation model parameters. In: SSW, pp 88–93
25. Tokuday K, Zen H (2015) Directly modeling speech waveforms by neural networks for statistical parametric speech synthesis. In: 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4215–4219
26. Tokuda K, Zen H (2016) Directly modeling voiced and unvoiced components in speech waveforms by neural networks. In: 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 5640–5644
27. Wang Y, Skerry-Ryan R, Stanton D, Wu Y, Weiss RJ, Jaitly N, Yang Z, Xiao Y, Chen Z, Bengio S et al (2017) Tacotron: towards end-to-end speech synthesis. In: Proceedings of the Interspeech 2017, pp 4006–4010
28. Griffin D, Lim J (1984) Signal estimation from modified short-time Fourier transform. IEEE Trans Acoust Speech Signal Process 32(2):236–243
29. Li N, Liu S, Liu Y, Zhao S, Liu M (2019) Neural speech synthesis with Transformer network. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 6706–6713
30. Yamamoto R, Song E, Kim JM (2020) 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). IEEE, pp 6199–6203
31. Weiss RJ, Skerry-Ryan R, Battenberg E, Mariooryad S, Kingma DP (2021) Wave-Tacotron: spectrogram-free end-to-end text-to-speech synthesis. In: 2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
32. Miao C, Liang S, Liu Z, Chen M, Ma J, Wang S, Xiao J (2020) EfficientTTS: an efficient and high-quality text-to-speech architecture. Preprint. arXiv:2012.03500
33. Chen N, Zhang Y, Zen H, Weiss RJ, Norouzi M, Dehak N, Chan W (2021) WaveGrad 2: iterative refinement for text-to-speech synthesis. Preprint. arXiv:210609660
34. Mehri S, Kumar K, Gulrajani I, Kumar R, Jain S, Sotelo J, Courville A, Bengio Y (2017) SampleRNN: an unconditional end-to-end neural audio generation model. In: The international conference on learning representations
35. Kim J, Kim S, Kong J, Yoon S (2020) Glow-TTS: a generative flow for text-to-speech via monotonic alignment search. Adv Neural Inf Proces Syst 33
36. Ito K (2017) The LJ speech dataset. https://keithito.com/LJ-Speech-Dataset/
37. Kingma DP, Welling M (2013) Auto-encoding variational bayes. Preprint. arXiv:1312.6114
38. Dinh L, Krueger D, Bengio Y (2014) NICE: non-linear independent components estimation. Preprint. arXiv:1410.8516
39. Kingma DP, Salimans T, Jozefowicz R, Chen X, Sutskever I, Welling M (2016) Improved variational inference with inverse autoregressive flow. Adv Neural Inf Process Syst 29:4743–4751
40. Kingma DP, Dhariwal P (2018) Glow: generative flow with invertible 1×1 convolutions. In: Proceedings of the 32nd international conference on neural information processing systems, pp 10236–10245
41. Wilcoxon F (1992) Individual comparisons by ranking methods. In: Breakthroughs in statistics. Springer, pp 196–202

Part III
Advanced Topics in TTS

In the second part of this book, we have introduced neural TTS in terms of basic model components. We further introduce neural TTS according to several advanced topics in Part III. Specifically, to improve the naturalness and expressiveness, we need to model, control, and transfer the style/prosody of speech in order to generate expressive speech (Chap. 8). Since some neural TTS models are facing robustness issues where word skipping and repeating problems in generated speech affect the speech quality, we need to improve the robustness of speech synthesis (Chap. 9). Since 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 are hot research topics (Chap. 10). In low data resource scenarios where the data to train a TTS model is insufficient, the synthesized speech may be of both low intelligibility and naturalness. Therefore, how to build data-efficient TTS models for new languages and new speakers are important (Chap. 11). We also briefly overview some speech synthesis tasks beyond TTS in Chap. 12. A taxonomy of these topics is shown in Fig. 1.

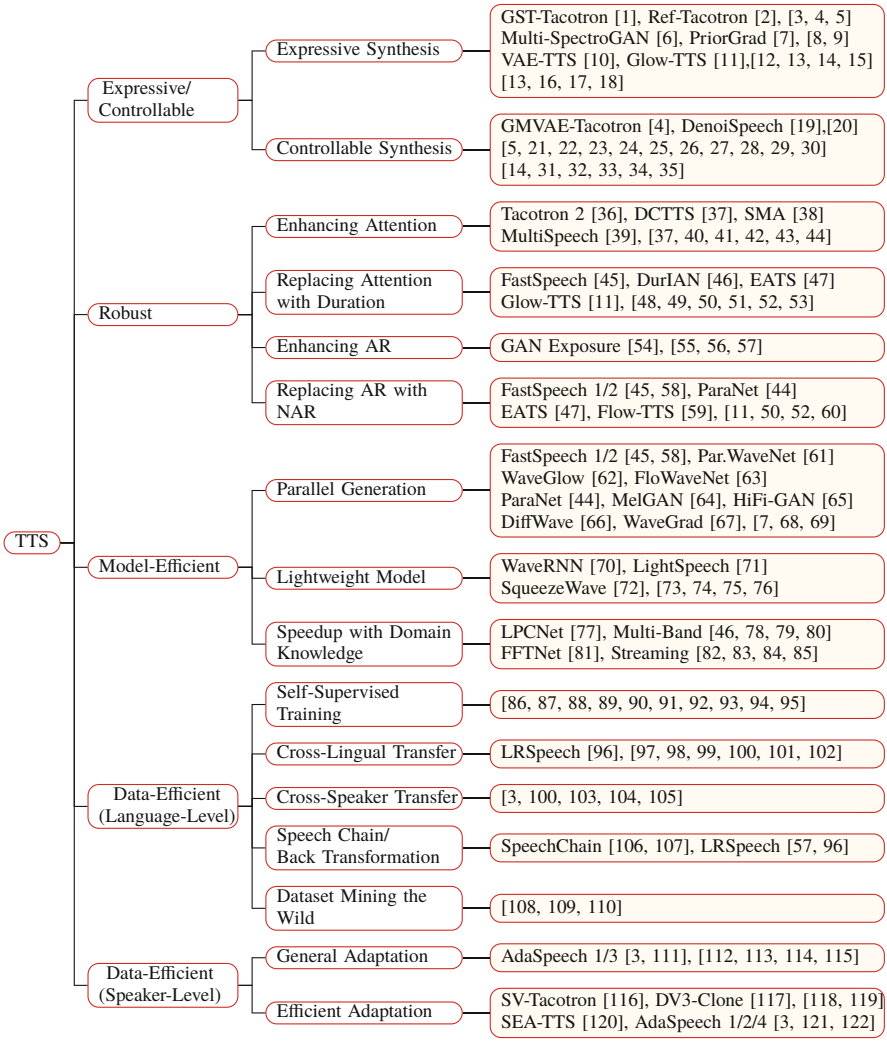


Fig. 1 Overview of the advanced topics in neural TTS described in Part III

Chapter 8
Expressive and Controllable TTS

Abstract Expressive and controllable TTS covers broad topics including modeling, disentangling, controlling, and transferring the content, timbre, prosody, style, emotion, etc. In this chapter, we first conduct a comprehensive analysis of this variation information, and then introduce some technologies for expressive and controllable TTS, including modeling, disentangling, controlling, and transferring this variation information.

Keywords Expressive TTS · Controllable TTS · Variation information · Disentanglement · Style transfer

The goal of text-to-speech synthesis is to generate intelligible and natural speech, where the naturalness largely depends on the expressiveness of the synthesized voice. Generally speaking, expressiveness is determined by multiple characteristics, such as content, timbre, emotion, and style, etc. 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 [123, 124] without enough input information will cause over-smoothing mel-spectrogram prediction [125, 126], 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 this variation information as input and better modeling this variation information is important to alleviate this problem and improve the expressiveness of synthesized speech.

Furthermore, by providing variation information as input, we can achieve controllable speech synthesis. Specifically, we can disentangle, control, and transfer the variation information as follows: (1) by adjusting this 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

X. Tan, *Neural Text-to-Speech Synthesis*, Artificial Intelligence: Foundations, Theory, and Algorithms, https://doi.org/10.1007/978-981-99-0827-1_8

fine-grained voice control and transfer, we need to disentangle different variation information, such as content and prosody, timbre and noise, etc.

Therefore, expressive and controllable TTS covers broad topics including modeling, disentangling, controlling, and transferring the content, timbre, style, and emotion, etc. In this chapter, we first conduct a comprehensive analysis of this variation information (Sect. 8.1), and then introduce some technologies for expressive (Sect. 8.2) and controllable (Sect. 8.3) TTS, including modeling, disentangling, controlling, and transferring this variation information.

8.1 Categorization of Variation Information in Speech

We first categorize the information needed for speech synthesis into four aspects as follows.

8.1.1 Text/Content Information

Text information can be characters or phonemes, and 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 [89, 90, 127–129], aiming to improve the quality and expressiveness of synthesized speech.

8.1.2 Speaker/Timbre Information

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 [39, 43, 116, 130, 131].

8.1.3 Style/Emotion Information

Style and emotion information, which is determined by the prosody, intonation, stress, and rhythm of speech and represents how to say the text [132, 133]. Style and emotion are important to improve the expressiveness of speech and the vast majority of works on expressive TTS focus on improving the style/emotion of speech [1, 2, 15, 22, 134, 135].

8.1.4 Recording Devices or Noise Environments

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 [3, 5, 19].

8.2 Modeling Variation Information for Expressive Synthesis

Many methods have been proposed to model different types of variation information in different granularities, as shown in Table 8.1. We introduce them in the following subsections.

8.2.1 Explicit or Implicit Modeling

We can categorize the works according to the types of information being modeled: (1) explicit information, where we can explicitly get the labels of this variation

Table 8.1 Some perspectives of modeling variation information for expressive speech synthesis

Perspective	Category	Description	Work
Information type	Explicit	Language/style/speaker ID	[35, 39, 136–138]
		Pitch/duration/energy	[45, 58, 139–142]
	Implicit	Reference encoder	[1–3, 8, 20, 95, 116, 117]
		VAE	[4, 5, 10, 12, 15, 16, 143]
		GAN/flow/diffusion	[6, 11, 13, 18, 20, 59]
Information granularity	Language/speaker level	Multi-lingual/speaker TTS	[39, 136, 137]
	Paragraph level	Long-form reading	[144–146]
	Utterance level	Timbre/prosody/noise	[1–3, 33, 116, 134, 147, 148]
	Word/syllable level	Fine-grained information	[16, 149–151]
	Character/phoneme level		[3, 15, 16, 150, 152–154]
	Frame level		[8, 19, 140, 152]

information; (2) implicit information, where we can only implicitly obtain this variation information.

For explicit information, we directly use them as input to enhance the models for expressive synthesis. We can obtain this information through different ways: (1) Get the language ID, speaker ID, style, and prosody from labeling data [35, 39, 136, 137]. For example, the prosody information can be labeled according to some annotation schemas, such as ToBI [155], AuToBI [156], Tilt [157], INTSINT [158], and SLAM [159]. (2) Extract the pitch and energy information from speech and extract duration from paired text and speech data [45, 58, 139–142].

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

- Reference encoder [1–3, 8, 9, 20, 116, 117], which extracts latent information from reference input as the latent variables. Skerry-Ryan et al. [2] 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 reference audio and uses it as the input of the decoder. During training, ground-truth reference audio is used, and during inference, another refer audio is used to synthesize speech with similar prosody. Wang et al. [1] extract embeddings from reference audio and use them as the query to attend (through Q/K/V based attention [160]) a bank 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 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 reference audio to attend and extract prosody representations, or simply pick one or some style tokens to synthesize speech.
- Variational autoencoder [4, 5, 10, 12, 14–16, 143] is a typical latent variable model. Zhang et al. [10] 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 [5, 12, 143, 161, 162] also leverage the VAE framework to better model the variance information for expressive synthesis.
- Advanced generative models [6, 7, 11, 13, 17, 18, 20, 59]. One way to alleviate the one-to-many mapping problem and combat over-smoothing prediction is to use advanced generative models (such as GAN, Flow, and Diffusion introduced in Sect. 3.3), which introduce latent variables (z in GAN/Flow, x_t in Diffusion) to implicitly learn the variation information and model the multi-modal distribution.

8.2.2 Modeling in Different Granularities

Variation information can be modeled in different granularities. We describe this information from coarse-grained to fine-grained levels:

- Language level and speaker level [39, 136, 137], where multilingual and multi-speaker TTS systems use language ID or speaker ID to differentiate languages and speakers.
- Paragraph level [144–146], where a TTS model needs to consider the connections between utterances/sentences for long-form reading.
- Utterance level [1–3, 33, 116, 134], where a single hidden vector is extracted from the reference speech to represent the timber/prosody of this utterance.
- Word/syllable level [16, 149–151], which can model the fine-grained style/prosody information that cannot be covered by utterance level information.
- Character/phoneme level [3, 15, 16, 150, 152–154], such as duration, pitch or prosody information.
- Frame level [8, 19, 140, 152], the most fine-grained information.

Some corresponding works on different granularities can be found in Table 8.1.

Furthermore, modeling the variance information with a hierarchical structure that covers different granularities is helpful for expressive synthesis. Suni et al. [163] demonstrate that hierarchical structures of prosody intrinsically exist in spoken languages. Kenter et al. [140] predict prosody features from the frame and phoneme levels to syllable level, and concatenate with the word- and sentence-level features. Hono et al. [149] 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. [16] use VAE to model variance information on both phoneme and word levels and combine them together to feed into the decoder. Chien and Lee [150] study on prosody prediction and propose a hierarchical structure from the word to phoneme level to improve the prosody prediction.

8.3 Modeling Variation Information for Controllable Synthesis

In this subsection, we introduce technologies on disentangling [5, 20, 21], controlling [22–28], and transferring [29, 30, 164, 165] variation information, as shown in Table 8.2.

Table 8.2 Some representative techniques for disentangling, controlling, and transferring in expressive speech synthesis

Technology	Description	Work
Disentangling for control	Adversarial training	[5, 19–21]
	Semi-supervised learning	[4, 5, 14, 19, 166]
Improving controllability	Cycle consistency/feedback loss	[31–35]
Transferring with control	Changing variance information in inference	[1–3, 10, 116, 167]

8.3.1 Disentangling for Control

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. [20] enhance the content-style disentanglement ability and controllability with adversarial and collaborative games. Hsu et al. [5] leverage the VAE framework with adversarial training to disentangle noise from speaker information. Qian et al. [21] propose speechflow to disentangle the rhythm, pitch, content, and timbre using three bottleneck reconstructions. Zhang et al. [19] propose to disentangle noise from speakers with frame-level noise modeling and adversarial training.

Disentangling with Semi-Supervised Learning 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 is challenging. When the partial label is available, [14] propose a semi-supervised learning method to learn the latent of the VAE model, in order to control attributes such as affect or speaking rate. When no label is available, [4] propose Gaussian mixture VAE models to disentangle different attributes, and [5, 19] leverage gradient reversal or adversarial training to disentangle speaker timbre from noise in order to synthesize clean speech for noisy speakers.

8.3.2 Improving Controllability

When providing variance information such as style tags 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 controllability 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. [35] conduct controllable emotional transfer by adding an emotion style

classifier with a feedback cycle, where the classifier encourages the TTS model to synthesize speech with a specific emotion. Whitehill et al. [32] 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. [31] 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 randomly chosen audio is used as the reference in inference. Other works [33, 34, 164, 165, 168, 169] leverage the feedback loss to ensure the controllability on style and speaker embeddings, etc.

8.3.3 Transfering with Control

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 [35, 39, 136, 137]. Alternatively, if we do not have a 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 [1–3, 8, 10, 116, 164, 165]; (2) predicting from text [3, 15, 45, 58, 134, 153, 167]; (3) obtaining by sampling from the latent space [1, 4, 10].

References

1. Wang Y, Stanton D, Zhang Y, Skerry-Ryan R, Battenberg E, Shor J, Xiao Y, Jia Y, Ren F, Saurous RA (2018) Style tokens: unsupervised style modeling, control and transfer in end-to-end speech synthesis. In: International conference on machine learning (PMLR), pp 5180–5189
2. Skerry-Ryan R, Battenberg E, Xiao Y, Wang Y, Stanton D, Shor J, Weiss R, Clark R, Saurous RA (2018) Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron. In: International conference on machine learning (PMLR), pp 4693–4702
3. Chen M, Tan X, Li B, Liu Y, Qin T, sheng zhao, Liu TY (2021) AdaSpeech: adaptive text to speech for custom voice. In: International conference on learning representations. https://openreview.net/forum?id=Drynvt7gg4L
4. Hsu WN, Zhang Y, Weiss RJ, Zen H, Wu Y, Wang Y, Cao Y, Jia Y, Chen Z, Shen J et al (2018) Hierarchical generative modeling for controllable speech synthesis. In: International conference on learning representations

5. Hsu WN, Zhang Y, Weiss RJ, Chung YA, Wang Y, Wu Y, Glass J (2019) 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). IEEE, pp 5901–5905
6. Lee SH, Yoon HW, Noh HR, Kim JH, Lee SW (2020) Multi-SpectroGAN: high-diversity and high-fidelity spectrogram generation with adversarial style combination for speech synthesis. Preprint. arXiv:2012.07267
7. Lee S-G, Kim H, Shin C, Tan X, Liu C, Meng Q, Qin T, Chen W, Yoon S, Liu TY (2021) PriorGrad: improving conditional denoising diffusion models with data-driven adaptive prior. Preprint. arXiv:2106.06406
8. Choi S, Han S, Kim D, Ha S (2020) Attentron: Few-shot text-to-speech utilizing attention-based variable-length embedding. In: Proceedigs of the Interspeech 2020, pp 2007–2011
9. Gururani S, Gupta K, Shah D, Shakeri Z, Pinto J (2019) Prosody transfer in neural text to speech using global pitch and loudness features. Preprint, arXiv:1911.09645
10. Zhang YJ, Pan S, He L, Ling ZH (2019) 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). IEEE, pp 6945–6949
11. Kim J, Kim S, Kong J, Yoon S (2020) Glow-TTS: a generative flow for text-to-speech via monotonic alignment search. Adv Neural Inf Process Syst 33
12. Akuzawa K, Iwasawa Y, Matsuo Y (2018) Expressive speech synthesis via modeling expressions with variational autoencoder. In: Proceedings of the Interspeech 2018, pp 3067–3071
13. Valle R, Shih K, Prenger R, Catanzaro B (2020) Flowtron: an autoregressive flow-based generative network for text-to-speech synthesis. Preprint. arXiv:2005.05957
14. Habib R, Mariooryad S, Shannon M, Battenberg E, Skerry-Ryan R, Stanton D, Kao D, Bagby T (2019) Semi-supervised generative modeling for controllable speech synthesis. In: International conference on learning representations
15. Sun G, Zhang Y, Weiss RJ, Cao Y, Zen H, Rosenberg A, Ramabhadran B, Wu Y (2020) 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). IEEE, pp 6699–6703
16. Sun G, Zhang Y, Weiss RJ, Cao Y, Zen H, Wu Y (2020) Fully-hierarchical fine-grained prosody modeling for interpretable speech synthesis. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 6264–6268
17. Du C, Yu K (2021) Mixture density network for phone-level prosody modelling in speech synthesis. Preprint. arXiv:2102.00851
18. Jeong M, Kim H, Cheon SJ, Choi BJ, Kim NS (2021) Diff-TTS: a denoising diffusion model for text-to-speech. Preprint. arXiv:2104.01409
19. Zhang C, Ren Y, Tan X, Liu J, Zhang K, Qin T, Zhao S, Liu TY (2021) DenoiSpeech: denoising text to speech with frame-level noise modeling. In: 2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
20. Ma S, Mcduff D, Song Y (2018) Neural TTS stylization with adversarial and collaborative games. In: International conference on learning representations
21. Qian K, Zhang Y, Chang S, Hasegawa-Johnson M, Cox D (2020) Unsupervised speech decomposition via triple information bottleneck. In: International conference on machine learning (PMLR), pp 7836–7846
22. Um S-Y, Oh S, Byun K, Jang I, Ahn C, Kang HG (2020) Emotional speech synthesis with rich and granularized control. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 7254–7258
23. Lee K, Park K, Kim D (2021) Styler: style modeling with rapidity and robustness via speech decomposition for expressive and controllable neural text to speech. Preprint, arXiv:2103.09474
24. Neekhara P, Hussain S, Dubnov S, Koushanfar F, McAuley J (2021) Expressive neural voice cloning. Preprint. arXiv:2102.00151

25. Bae J-S, Bae H, Joo YS, Lee J, Lee GH, Cho HY (2020) Speaking speed control of end-to-end speech synthesis using sentence-level conditioning. In: Proceedings of the Interspeech 2020, pp 4402–4406
26. Polyak A, Adi Y, Copet J, Kharitonov E, Lakhotia K, Hsu WN, Mohamed A, Dupoux E (2021) Speech resynthesis from discrete disentangled self-supervised representations. Preprint. arXiv:2104.00355
27. Tits N, Haddad KE, Dutoit T (2021) Analysis and assessment of controllability of an expressive deep learning-based TTS system. Preprint. arXiv:2103.04097
28. Li X, Song C, Li J, Wu Z, Jia J, Meng H (2021) Towards multi-scale style control for expressive speech synthesis. Preprint. arXiv:2104.03521
29. Karlapati S, Moinet A, Joly A, Klimkov V, Sáez-Trigueros D, Drugman T (2020) Copycat: many-to-many fine-grained prosody transfer for neural text-to-speech. In: Proceedings of the Interspeech 2020, pp 4387–4391
30. Inoue K, Hara S, Abe M, Hojo N, Ijima Y (2021) Model architectures to extrapolate emotional expressions in DNN-based text-to-speech. Speech Commun 126:35–43
31. Liu DR, Yang CY, Wu SL, Lee HY (2018) Improving unsupervised style transfer in end-to-end speech synthesis with end-to-end speech recognition. In: 2018 IEEE spoken language technology workshop (SLT). IEEE, pp 640–647
32. Whitehill M, Ma S, McDuff D, Song Y (2020) Multi-reference neural TTS stylization with adversarial cycle consistency. In: Proceedings of the Interspeech 2020, pp 4442–4446
33. Liu R, Sisman B, Gao G, Li H (2020) Expressive TTS training with frame and style reconstruction loss. Preprint. arXiv:2008.01490
34. Cai Z, Zhang C, Li M (2020) From speaker verification to multispeaker speech synthesis, deep transfer with feedback constraint. In: Proceedings of the Interspeech 2020, pp 3974–3978
35. Li T, Yang S, Xue L, Xie L (2021) Controllable emotion transfer for end-to-end speech synthesis. In: 2021 12th international symposium on chinese spoken language processing (ISCSLP). IEEE, pp 1–5
36. Wang Y, Skerry-Ryan R, Stanton D, Wu Y, Weiss RJ, Jaitly N, Yang Z, Xiao Y, Chen Z, Bengio S et al (2017) Tacotron: towards end-to-end speech synthesis. In: Proceedings of the Interspeech 2017, pp 4006–4010
37. Tachibana H, Uenoyama K, Aihara S (2018) 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). IEEE, pp 4784–4788
38. He M, Deng Y, He L (2019) Robust sequence-to-sequence acoustic modeling with stepwise monotonic attention for neural TTS. In: Proceedings of the Interspeech 2019, pp 1293–1297
39. Chen M, Tan X, Ren Y, Xu J, Sun H, Zhao S, Qin T (2020) MultiSpeech: multi-speaker text to speech with Transformer. In: INTERSPEECH, pp 4024–4028
40. Sotelo J, Mehri S, Kumar K, Santos JF, Kastner K, Courville AC, Bengio Y (2017) Char2wav: end-to-end speech synthesis. In: 5th international conference on learning representations, ICLR 2017, Toulon, France, April 24–26, 2017, Workshop Track Proceedings. OpenReview.net. https://openreview.net/forum?id=B1VWyySKx
41. Shen J, Pang R, Weiss RJ, Schuster M, Jaitly N, Yang Z, Chen Z, Zhang Y, Wang Y, Skerry-Ryan R et al (2018) Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4779–4783
42. Zhang JX, Ling ZH, Dai LR (2018) Forward attention in sequence-to-sequence acoustic modeling for speech synthesis. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4789–4793
43. Ping W, Peng K, Gibiansky A, Arik SO, Kannan A, Narang S, Raiman J, Miller J (2018) Deep voice 3: 2000-speaker neural text-to-speech. In: Proceedings of the international conference on learning representations, pp 214–217
44. Peng K, Ping W, Song Z, Zhao K (2020) Non-autoregressive neural text-to-speech. In: International conference on machine learning (PMLR), pp 7586–7598

45. Ren Y, Ruan Y, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2019) FastSpeech: fast, robust and controllable text to speech. In: NeurIPS
46. Yu C, Lu H, Hu N, Yu M, Weng C, Xu K, Liu P, Tuo D, Kang S, Lei G et al (2020) DurIAN: duration informed attention network for speech synthesis. In: Proceedings of the Interspeech 2020, pp 2027–2031
47. Donahue J, Dieleman S, Bińkowski M, Elsen E, Simonyan K (2021) End-to-end adversarial text-to-speech. In: International conference on learning representations
48. Li N, Liu Y, Wu Y, Liu S, Zhao S, Liu M (2020) RobuTrans: a robust Transformer-based text-to-speech model. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 8228–8235
49. Beliaev S, Rebryk Y, Ginsburg B (2020) TalkNet: fully-convolutional non-autoregressive speech synthesis model. Preprint. arXiv:2005.05514
50. Vainer J, Dušek O (2020) SpeedySpeech: efficient neural speech synthesis. In: Proceedings of the Interspeech 2020, pp 3575–3579
51. Zeng Z, Wang J, Cheng N, Xia T, Xiao J (2020) 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). IEEE, pp 6714–6718
52. Elias I, Zen H, Shen J, Zhang Y, Ye J, Skerry-Ryan R, Wu Y (2021) Parallel Tacotron 2: a non-autoregressive neural TTS model with differentiable duration modeling. Preprint. arXiv:2103.14574
53. Shen J, Jia Y, Chrzanowski M, Zhang Y, Elias I, Zen H, Wu Y (2020) Non-attentive Tacotron: robust and controllable neural TTS synthesis including unsupervised duration modeling. Preprint. arXiv:2010.04301
54. Guo H, Soong FK, He L, Xie L (2019) A new GAN-based end-to-end TTS training algorithm. In: Proceedings of the Interspeech 2019, pp 1288–1292
55. Liu R, Yang J, Liu M (2019) A new end-to-end long-time speech synthesis system based on Tacotron2. In: Proceedings of the 2019 international symposium on signal processing systems, pp 46–50
56. Liu R, Sisman B, Li J, Bao F, Gao G, Li H (2020) Teacher-student training for robust Tacotron-based TTS. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 6274–6278
57. Ren Y, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2019) Almost unsupervised text to speech and automatic speech recognition. In: International conference on machine learning (PMLR), pp 5410–5419
58. Ren Y, Hu C, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2021) FastSpeech 2: fast and high-quality end-to-end text to speech. In: International conference on learning representations. https://openreview.net/forum?id=piLPYqxtWuA
59. Miao C, Liang S, Chen M, Ma J, Wang S, Xiao J (2020) 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). IEEE, pp 7209–7213
60. Liu P, Cao Y, Liu S, Hu N, Li G, Weng C, Su D (2021) VARA-TTS: non-autoregressive text-to-speech synthesis based on very deep VAE with residual attention. Preprint. arXiv:2102.06431
61. van der Oord A, Li Y, Babuschkin I, Simonyan K, Vinyals O, Kavukcuoglu K, Driessche G, Lockhart E, Cobo L, Stimberg F et al (2018) Parallel WaveNet: fast high-fidelity speech synthesis. In: International conference on machine learning (PMLR), pp 3918–3926
62. Prenger R, Valle R, Catanzaro B (2019) WaveGlow: a flow-based generative network for speech synthesis. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 3617–3621
63. Kim S, Lee SG, Song J, Kim J, Yoon S (2019) FloWaveNet: a generative flow for raw audio. In: International conference on machine learning (PMLR), pp 3370–3378
64. Kumar K, Kumar R, de Boissiere T, Gestin L, Teoh WZ, Sotelo J, de Brébisson A, Bengio Y, Courville A (2019) MelGAN: generative adversarial networks for conditional waveform synthesis. In: NeurIPS

65. Kong J, Kim J, Bae J (2020) HiFi-GAN: generative adversarial networks for efficient and high fidelity speech synthesis. Adv Neural Inf Process Syst 33
66. Kong Z, Ping W, Huang J, Zhao K, Catanzaro B (2021) DiffWave: a versatile diffusion model for audio synthesis. In: International conference on learning representations
67. Chen N, Zhang Y, Zen H, Weiss RJ, Norouzi M, Chan W (2021) WaveGrad: estimating gradients for waveform generation. In: International conference on learning representations
68. Yamamoto R, Song E, Kim JM (2020) 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). IEEE, pp 6199–6203
69. Kim J, Kong J, Son J (2021) Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. Preprint. arXiv:2106.06103
70. Kalchbrenner N, Elsen E, Simonyan K, Noury S, Casagrande N, Lockhart E, Stimberg F, van den Oord A, Dieleman S, Kavukcuoglu K (2018) Efficient neural audio synthesis. In: International conference on machine learning (PMLR), pp 2410–2419
71. Luo R, Tan X, Wang R, Qin T, Li J, Zhao S, Chen E, Liu TY (2021) LightSpeech: lightweight and fast text to speech with neural architecture search. In: 2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
72. Zhai B, Gao T, Xue F, Rothchild D, Wu B, Gonzalez JE, Keutzer K (2020) SqueezeWave: extremely lightweight vocoders for on-device speech synthesis. Preprint. arXiv:2001.05685
73. Kanagawa H, Ijima Y (2020) Lightweight LPCNet-based neural vocoder with tensor decomposition. In: Proceedings of the Interspeech 2020, pp 205–209
74. Hsu Pc, Lee Hy (2020) WG-WaveNet: real-time high-fidelity speech synthesis without gpu. In: Proceedings of the Interspeech 2020, pp 210–214
75. Huang Z, Li H, Lei M (2020) DeviceTTS: a small-footprint, fast, stable network for on-device text-to-speech. Preprint. arXiv:2010.15311
76. Zeng Z, Wang J, Cheng N, Xiao J (2021) Lvcnet: efficient condition-dependent modeling network for waveform generation. Preprint. arXiv:2102.10815
77. Valin JM, Skoglund J (2019) LPCNet: improving neural speech synthesis through linear prediction. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 5891–5895
78. Yang G, Yang S, Liu K, Fang P, Chen W, Xie L (2020) Multi-band MelGAN: faster waveform generation for high-quality text-to-speech. Preprint. arXiv:2005.05106
79. Okamoto T, Tachibana K, Toda T, Shiga Y, Kawai H (2018) 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). IEEE, pp 5654–5658
80. Cui Y, Wang X, He L, Soong FK (2020) An efficient subband linear prediction for LPCNet-based neural synthesis. In: INTERSPEECH, pp 3555–3559
81. Jin Z, Finkelstein A, Mysore GJ, Lu J (2018) FFTNet: a real-time speaker-dependent neural vocoder. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 2251–2255
82. Ellinas N, Vamvoukakis G, Markopoulos K, Chalamandaris A, Maniati G, Kakoulidis P, Raptis S, Sung JS, Park H, Tsiakoulis P (2020) High quality streaming speech synthesis with low, sentence-length-independent latency. In: Proceedings of the Interspeech 2020, pp 2022–2026
83. Ma M, Zheng B, Liu K, Zheng R, Liu H, Peng K, Church K, Huang L (2020) Incremental text-to-speech synthesis with prefix-to-prefix framework. In: Proceedings of the 2020 conference on empirical methods in natural language processing: findings, pp 3886–3896
84. Stephenson B, Besacier L, Girin L, Hueber T (2020) What the future brings: investigating the impact of lookahead for incremental neural TTS. In: Proceedings of the Interspeech 2020, pp 215–219

85. Yanagita T, Sakti S, Nakamura S (2019) 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, pp 183–188
86. Chung YA, Wang Y, Hsu WN, Zhang Y, Skerry-Ryan R (2019) 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). IEEE, pp 6940–6944
87. Wang P, Qian Y, Soong FK, He L, Zhao H (2015) Word embedding for recurrent neural network based TTS synthesis. In: 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4879–4883
88. Zhang M, Wang X, Fang F, Li H, Yamagishi J (2019) Joint training framework for text-to-speech and voice conversion using multi-source Tacotron and WaveNet. In: Proceedings of the Interspeech 2019, pp 1298–1302
89. Fang W, Chung YA, Glass J (2019) Towards transfer learning for end-to-end speech synthesis from deep pre-trained language models. Preprint. arXiv:1906.07307
90. Jia Y, Zen H, Shen J, Zhang Y, Wu Y (2021) PnG BERT: augmented BERT on phonemes and graphemes for neural TTS. Preprint. arXiv:2103.15060
91. Tjandra A, Sisman B, Zhang M, Sakti S, Li H, Nakamura S (2019) VQVAE unsupervised unit discovery and multi-scale Code2Spec inverter for zerospeech challenge 2019. In: Proceedings of the Interspeech 2019, pp 1118–1122
92. Liu AH, Tu T, Lee Hy, Lee L-S (2020) 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). IEEE, pp 7259–7263
93. Tu T, Chen YJ, Liu AH, Lee Hy (2020) Semi-supervised learning for multi-speaker text-to-speech synthesis using discrete speech representation. In: Proceedings of the Interspeech 2020, pp 3191–3195
94. Dunbar E, Algayres R, Karadayi J, Bernard M, Benjumea J, Cao XN, Miskic L, Dugrain C, Ondel L, Black AW et al (2019) The zero resource speech challenge 2019: TTS without T. In: Proceedings of the Interspeech 2019, pp 1088–1092
95. Chen L, Deng Y, Wang X, Soong FK, He L (2021) Speech BERT embedding for improving prosody in neural TTS. In: ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 6563–6567
96. Xu J, Tan X, Ren Y, Qin T, Li J, Zhao S, Liu TY (2020) LRSpeech: extremely low-resource speech synthesis and recognition. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery and data mining, pp 2802–2812
97. Chen Y-J, Tu T, Yeh C-C, Lee H-Y (2019) End-to-end text-to-speech for low-resource languages by cross-lingual transfer learning. In: Proceedings of the Interspeech 2019, pp 2075–2079
98. Azizah K, Adriani M, Jatmiko W (2020) Hierarchical transfer learning for multilingual, multi-speaker, and style transfer DNN-based TTS on low-resource languages. IEEE Access 8:179798–179812
99. de Korte M, Kim J, Klabbers E (2020) Efficient neural speech synthesis for low-resource languages through multilingual modeling. In: Proceedings of the Interspeech 2020, pp 2967–2971
100. Yang J, He L (2020) Towards universal text-to-speech. In: INTERSPEECH, pp 3171–3175
101. Prajwal K, Jawahar C (2021) Data-efficient training strategies for neural TTS systems. In: 8th ACM IKDD CODS and 26th COMAD, pp 223–227
102. He M, Yang J, He L (2021) Multilingual Byte2Speech text-to-speech models are few-shot spoken language learners. Preprint. arXiv:2103.03541
103. Luong HT, Wang X, Yamagishi J, Nishizawa N (2019) Training multi-speaker neural text-to-speech systems using speaker-imbalanced speech corpora. In: Proceedings of the Interspeech 2019, pp 1303–1307

104. Huybrechts G, Merritt T, Comini G, Perz B, Shah R, Lorenzo-Trueba J (2020) Low-resource expressive text-to-speech using data augmentation. Preprint. arXiv:2011.05707
105. Dai D, Chen L, Wang Y, Wang M, Xia R, Song X, Wu Z, Wang Y (2020) Noise robust TTS for low resource speakers using pre-trained model and speech enhancement. Preprint. arXiv:2005.12531
106. Tjandra A, Sakti S, Nakamura S (2017) Listening while speaking: Speech chain by deep learning. In: 2017 IEEE automatic speech recognition and understanding workshop (ASRU). IEEE, pp 301–308
107. Tjandra A, Sakti S, Nakamura S (2018) Machine speech chain with one-shot speaker adaptation. In: Proceedings of the Interspeech 2018, pp 887–891
108. Cooper EL (2019) Text-to-speech synthesis using found data for low-resource languages. Ph.D. Thesis, Columbia University
109. Hu Q, Marchi E, Winarsky D, Stylianou Y, Naik D, Kajarekar S (2019) Neural text-to-speech adaptation from low quality public recordings. In: Speech Synthesis Workshop, vol 10
110. Cooper E, Wang X, Zhao Y, Yasuda Y, Yamagishi J (2020) Pretraining strategies, waveform model choice, and acoustic configurations for multi-speaker end-to-end speech synthesis. Preprint. arXiv:2011.04839
111. Yan Y, Tan X, Li B, Zhang G, Qin T, Zhao S, Shen Y, Zhang WQ, Liu TY (2021) AdaSpeech 3: adaptive text to speech for spontaneous style. In: INTERSPEECH
112. Cooper E, Lai CI, Yasuda Y, Yamagishi J (2020) Can speaker augmentation improve multi-speaker end-to-end TTS? In: Proceedings of the Interspeech 2020, pp 3979–3983
113. Paul D, Shifas MP, Pantazis Y, Stylianou Y (2020) Enhancing speech intelligibility in text-to-speech synthesis using speaking style conversion. In: Proceedings of the Interspeech 2020, pp 1361–1365
114. Hu Q, Bleisch T, Petkov P, Raitio T, Marchi E, Lakshminarasimhan V (2021) Whispered and lombard neural speech synthesis. In: 2021 IEEE spoken language technology workshop (SLT). IEEE, pp 454–461
115. Chen M, Chen M, Liang S, Ma J, Chen L, Wang S, Xiao J (2019) Cross-lingual, multi-speaker text-to-speech synthesis using neural speaker embedding. In: Proceedings of the Interspeech 2019, pp 2105–2109
116. Jia Y, Zhang Y, Weiss RJ, Wang Q, Shen J, Ren F, Chen Z, Nguyen P, Pang R, Moreno IL et al (2018) Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In: Proceedings of the 32nd international conference on neural information processing systems, pp 4485–4495
117. Arık SÖ, Chen J, Peng K, Ping W, Zhou Y (2018) Neural voice cloning with a few samples. In: Proceedings of the 32nd international conference on neural information processing systems, pp 10040–10050
118. Kons Z, Shechtman S, Sorin A, Rabinovitz C, Hoory R (2019) High quality, lightweight and adaptable TTS using LPCNet. In: Proceedings of the Interspeech 2019, pp 176–180
119. Zhang Z, Tian Q, Lu H, Chen LH, Liu S (2020) AdaDurIAN: few-shot adaptation for neural text-to-speech with durian. Preprint. arXiv:2005.05642
120. Chen Y, Assael Y, Shillingford B, Budden D, Reed S, Zen H, Wang Q, Cobo LC, Trask A, Laurie B et al (2018) Sample efficient adaptive text-to-speech. In: International conference on learning representations
121. Yan Y, Tan X, Li B, Qin T, Zhao S, Shen Y, Liu TY (2021) AdaSpeech 2: adaptive text to speech with untranscribed data. In: 2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
122. Wu Y, Tan X, Li B, He L, Zhao S, Song R, Qin T, Liu TY (2022) Adaspeech 4: adaptive text to speech in zero-shot scenarios. In: INTERSPEECH
123. Gazor S, Zhang W (2003) Speech probability distribution. IEEE Signal Process Lett 10(7):204–207
124. Usman M, Zubair M, Shiblee M, Rodrigues P, Jaffar S (2018) Probabilistic modeling of speech in spectral domain using maximum likelihood estimation. Symmetry 10(12):750

125. Toda T, Tokuda K (2007) A speech parameter generation algorithm considering global variance for HMM-based speech synthesis. IEICE Trans Inf Syst 90(5):816–824
126. Takamichi S, Toda T, Black AW, Neubig G, Sakti S, Nakamura S (2016) Postfilters to modify the modulation spectrum for statistical parametric speech synthesis. IEEE/ACM Trans Audio Speech Lang Process 24(4):755–767
127. Hayashi T, Watanabe S, Toda T, Takeda K, Toshniwal S, Livescu K (2019) Pre-trained text embeddings for enhanced text-to-speech synthesis. In: Proc Interspeech 2019, pp 4430–4434
128. Xiao Y, He L, Ming H, Soong FK (2020) 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). IEEE, pp 6704–6708
129. Zhang G, Song K, Tan X, Tan D, Yan Y, Liu Y, Wang G, Zhou W, Qin T, Lee T et al (2022) Mixed-phoneme BERT: improving BERT with mixed phoneme and sup-phoneme representations for text to speech. Preprint. arXiv:2203.17190
130. Gibiansky A, Arik SÖ, Diamos GF, Miller J, Peng K, Ping W, Raiman J, Zhou Y (2017) Deep voice 2: multi-speaker neural text-to-speech. In: Proceedings of the neural information processing systems
131. Moss HB, Aggarwal V, Prateek N, González J, Barra-Chicote R (2020) BOFFIN TTS: few-shot speaker adaptation by Bayesian optimization. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 7639–7643
132. Wagner M, Watson DG (2010) Experimental and theoretical advances in prosody: a review. Lang Cogn Process 25(7–9):905–945
133. Ladd DR (2008) Intonational phonology. Cambridge University Press, Cambridge
134. Stanton D, Wang Y, Skerry-Ryan R (2018) Predicting expressive speaking style from text in end-to-end speech synthesis. In: 2018 IEEE spoken language technology workshop (SLT). IEEE, pp 595–602
135. Gao Y, Zheng W, Yang Z, Kohler T, Fuegen C, He Q (2020) Interactive text-to-speech via semi-supervised style transfer learning. Preprint arXiv:2002.06758
136. Zhang Y, Weiss RJ, Zen H, Wu Y, Chen Z, Skerry-Ryan R, Jia Y, Rosenberg A, Ramabhadran B (2019) Learning to speak fluently in a foreign language: multilingual speech synthesis and cross-language voice cloning. In: Proceedings of the Interspeech 2019, pp 2080–2084
137. Nekvinda T, Dušek O (2020) One model, many languages: meta-learning for multilingual text-to-speech. In: Proceedings of the Interspeech 2020, pp 2972–2976
138. Kim M, Cheon SJ, Choi BJ, Kim JJ, Kim NS (2021) Expressive text-to-speech using style tag. Preprint arXiv:2104.00436
139. Łańcucki A (2020) FastPitch: parallel text-to-speech with pitch prediction. Preprint. arXiv:2006.06873
140. Kenter T, Wan V, Chan CA, Clark R, Vit J (2019) Chive: varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network. In: International conference on machine learning (PMLR), pp 3331–3340
141. Morrison M, Jin Z, Salamon J, Bryan NJ, Mysore GJ (2020) Controllable neural prosody synthesis. In: Proceedings of the Interspeech 2020, pp 4437–4441
142. Valle R, Li J, Prenger R, Catanzaro B (2020) 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). IEEE, pp 6189–6193
143. Elias I, Zen H, Shen J, Zhang Y, Jia Y, Weiss R, Wu Y (2020) Parallel Tacotron: non-autoregressive and controllable TTS. Preprint, arXiv:2010.11439
144. Aubin A, Cervone A, Watts O, King S (2019) Improving speech synthesis with discourse relations. In: INTERSPEECH, pp 4470–4474
145. Xu G, Song W, Zhang Z, Zhang C, He X, Zhou B (2020) Improving prosody modelling with cross-utterance BERT embeddings for end-to-end speech synthesis. Preprint. arXiv:2011.05161

146. Wang X, Ming H, He L, Soong FK (2020) s-Transformer: segment-transformer for robust neural speech synthesis. Preprint. arXiv:2011.08480
147. Liu Y, Xu Z, Wang G, Chen K, Li B, Tan X, Li J, He L, Zhao S (2021) DelightfulTTS: the microsoft speech synthesis system for Blizzard challenge 2021. Preprint. arXiv:2110.12612
148. Liu Y, Xue R, He L, Tan X, Zhao S (2022) DelightfulTTS 2: end-to-end speech synthesis with adversarial vector-quantized auto-encoders. Preprint. arXiv:2207.04646
149. Hono Y, Tsuboi K, Sawada K, Hashimoto K, Oura K, Nankaku Y, Tokuda K (2020) Hierarchical multi-grained generative model for expressive speech synthesis. In: Proceedings of the Interspeech 2020, pp 3441–3445
150. Chien CM, Lee Hy (2021) Hierarchical prosody modeling for non-autoregressive speech synthesis. In: 2021 IEEE spoken language technology workshop (SLT). IEEE, pp 446–453
151. Talman A, Suni A, Celikkanat H, Kakouros S, Tiedemann J, Vainio M et al (2019) 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
152. Lee Y, Kim T (2019) 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). IEEE, pp 5911–5915
153. Zeng Z, Wang J, Cheng N, Xiao J (2020) Prosody learning mechanism for speech synthesis system without text length limit. In: Proceedings of the Interspeech 2020, pp 4422–4426
154. Lei Y, Yang S, Xie L (2021) Fine-grained emotion strength transfer, control and prediction for emotional speech synthesis. In: 2021 IEEE spoken language technology workshop (SLT). IEEE, pp 423–430
155. Silverman K, Beckman M, Pitrelli J, Ostendorf M, Wightman C, Price P, Pierrehumbert J, Hirschberg J (1992) ToBI: a standard for labeling English prosody. In: Second international conference on spoken language processing
156. Rosenberg A (2010) AuToBI-a tool for automatic ToBI annotation. In: Eleventh annual conference of the international speech communication association
157. Taylor P (1998) The Tilt intonation model. In: Fifth international conference on spoken language processing
158. Hirst D (2001) Automatic analysis of prosody for multilingual speech corpora. In: Improvements in speech synthesis, pp 320–327
159. Obin N, Beliao J, Veaux C, Lacheret A (2014) SLAM: automatic stylization and labelling of speech melody. In: Speech prosody, p 246
160. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
161. Aggarwal V, Cotescu M, Prateek N, Lorenzo-Trueba J, Barra-Chicote R (2020) 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). IEEE, pp 6179–6183
162. Tan X, Chen J, Liu H, Cong J, Zhang C, Liu Y, Wang X, Leng Y, Yi Y, He L et al (2022) NaturalSpeech: end-to-end text to speech synthesis with human-level quality. Preprint. arXiv:2205.04421
163. Suni A, Šimko J, Aalto D, Vainio M (2017) Hierarchical representation and estimation of prosody using continuous wavelet transform. Comput Speech Lang 45:123–136
164. Xue L, Pan S, He L, Xie L, Soong FK (2021) Cycle consistent network for end-to-end style transfer TTS training. Neural Netw 140:223–236
165. An X, Soong FK, Xie L (2021) Improving performance of seen and unseen speech style transfer in end-to-end neural TTS. Preprint. arXiv:2106.10003
166. Shechtman S, Fernandez R, Haws D (2021) Supervised and unsupervised approaches for controlling narrow lexical focus in sequence-to-sequence speech synthesis. In: 2021 IEEE spoken language technology workshop (SLT). IEEE, pp 431–437

167. Guo Z, Leng Y, Wu Y, Zhao S, Tan X (2022) PromptTTS: Controllable text-to-speech with text descriptions. Preprint. arXiv:2211.12171
168. Nachmani E, Polyak A, Taigman Y, Wolf L (2018) Fitting new speakers based on a short untranscribed sample. In: International conference on machine learning (PMLR), pp 3683–3691
169. Shi Y, Bu H, Xu X, Zhang S, Li M (2020) AISHELL-3: A multi-speaker Mandarin TTS corpus and the baselines. Preprint. arXiv:2010.11567

Chapter 9
Robust TTS

Abstract In this chapter, we introduce how to address the robustness issues in TTS. We summarize some popular techniques to improve robustness, including enhancing attention, replacing attention with duration prediction, enhancing autoregressive generation, and replacing autoregressive generation with non-autoregressive generation.

Keywords Robust TTS · Generalization · Text-speech alignment · Attention · Autoregressive generation · Non-autoregressive generation

A good TTS system should be robust to always generate "correct" speech according to text even when encountering corner cases. However, in neural TTS, robustness issues such as word skipping, repeating, and attention collapse[1] often happen in acoustic models when generating mel-spectrogram sequence from character/phoneme sequence with encoder-decoder-attention and autoregressive generation, or some glitches such as hoarseness, metallic noise, jitter, or pitch breaking often happen in vocoders when generating waveform from mel-spectrogram sequence.

Basically speaking, the causes of these robustness issues can be categorized as follows:

- A lack of data coverage and generalization ability can cause robustness issues. For example, when the test domain is not well covered by the training domain, the TTS models may not perform well and cause robustness issues when generating speech.
- The difficulty in learning the alignments between characters/phonemes and mel-spectrograms. Vocoders do not face severe robustness issues due to this reason, since the acoustic features and waveform are already aligned frame-wisely (i.e.,

[1] Attention collapse means the generated speech has unintelligible gibberish, which is usually caused by the not focused attention on a single input token [1].

X. Tan, *Neural Text-to-Speech Synthesis*, Artificial Intelligence: Foundations, Theory, and Algorithms, https://doi.org/10.1007/978-981-99-0827-1_9

Table 9.1 Categorization of the methods for robust TTS

Category	Technology	Work
Improving generalization	Improving data coverage	[2–8]
Enhancing attention	Content-based attention	[9, 10]
	Location-based attention	[7, 11–13]
	Content/location hybrid attention	[14]
	Monotonic attention	[1, 15, 16]
	Windowing or off-diagonal penalty	[15, 17–19]
	Enhancing enc-dec connection	[9, 14, 18–20]
	Positional attention	[21–23]
Replacing attention with duration prediction	Label from encoder-decoder attention	[24–27]
	Label from CTC alignment	[28]
	Label from HMM alignment	[29–34]
	Dynamic programming	[35–37]
	Monotonic alignment search	[38]
	Monotonic interpolation with soft DTW	[39, 40]
Enhancing AR	Professor forcing	[41, 42]
	Reducing training/inference gap	[25]
	Knowledge distillation	[43]
	Bidirectional regularization	[44, 45]
Replacing AR with NAR	Parallel generation	[21, 24, 29, 39]

each frame of acoustic features corresponds to a certain number (hop size) of waveform points).

- The exposure bias and error propagation problems incurred in an autoregressive generation.

In this chapter, we introduce how to address these robustness issues according to their causes in different categories. We summarize some popular techniques in these categories to improve robustness, as shown in Table 9.1. The works addressing these 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 introduce these technologies in Sects. 9.1, 9.2, and 9.3.

9.1 Improving Generalization Ability

To alleviate the robustness issues caused by a lack of data coverage and generalization ability, some works propose to scale to the unseen domain, such as using pre-trained word embeddings or model parameters [8, 46–51] to provide better text representations, increasing the amount and diversity of the training

data [6], adopting relative position encoding to support long sequence unseen in training [7, 52], or train universal acoustic models or vocoders [2–5] (usually in multi-lingual, multi-speaker, and multi-style), or sophisticated designs [53].

9.2 Improving Text-Speech Alignment

There are different methods to improve the alignment learning between characters/phonemes and mel-spectrograms, in order to alleviate the robustness issues: (1) enhancing the robustness of attention mechanism [1, 9, 11, 14, 15, 17, 19], and (2) removing attention and instead predicting duration explicitly to bridge the length mismatch between text and speech [24, 30, 39, 40]. We introduce them respectively in Sects. 9.2.1 and 9.2.2.

9.2.1 Enhancing Attention

In attention-based 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 [1]: (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 9.1) according to whether they satisfy the above three properties and list them in Table 9.2. We describe these techniques as follows.

- Content-based attention. The early attention mechanisms adopted in TTS (e.g. Tacotron [9]) are content-based [54], 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 [54, 55] where the alignments between the source and target tokens are purely based on semantic meaning (content). However, for the tasks like automatic speech recognition [56–58] and text-to-speech synthesis [9], the alignments between text and speech have some specific properties. For example, in TTS [1], the attention alignments should be local, monotonic, and complete. Therefore, advanced attention mechanisms should be designed to better leverage these properties.

Table 9.2 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	✓	✓	✓

- Location-based attention. Considering the alignments between text and speech are depending on their positions, location-based attention [7, 59] is proposed to leverage the positional information for alignment. Several TTS models such as Char2Wav [11], VoiceLoop [12], and MelNet [13] adopt the location-based attention. As we summarize in Table 9.2, 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 attention, [14, 56] 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 [1, 16, 60–62], 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 skipping and repeating issues. However, the completeness property cannot be guaranteed in the above monotonic attention. Therefore, [1] 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, [15, 17–19, 56] 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. [19] 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 the next frame and thus tends to ignore the text information from the encoder. Therefore, some works propose to enhance the connection between the encoder and decoder and thus can improve attention alignment. Wang et al. [9] and Shen et al. [14] 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 Pre-net before the decoder [9, 14, 19], or a small hidden size in the Pre-net as a bottleneck [19], 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. [18, 19] propose to enhance the connection of the positional information between source and target sequences, which benefits attention alignment learning. Liu et al. [20] leverage connectionist temporal classification (CTC) [63] based automatic speech recognition (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 [21, 22] 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 a parallel generation.

9.2.2 Replacing Attention with Duration Prediction

While improving the attention alignments between text and speech can alleviate the robustness issues to some extent, it cannot totally avoid them. Thus, some works [24, 30, 38, 39] 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 the mel-spectrogram sequence. After that, the model can generate a 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 9.3 and describe them as follows.

- External alignment. The works leveraging external alignment tools [36, 63–65] can be divided into several categories according to the used alignment tools: (1) Encoder-decoder attention: FastSpeech [24] obtains the duration label from the attention alignments of an autoregressive acoustic model. SpeedySpeech [25] follows a similar pipeline of FastSpeech to extract the duration from an autoregressive teacher model but replaces the whole network structure with purely

Table 9.3 A category of neural TTS on duration prediction

Perspective	Category	Work
External/internal	External	FastSpeech 1/2 [24, 29], DurIAN [30], TalkNet [28], [25, 33, 34]
	Internal	AlignTTS [35], Glow-TTS [38], EATS [39], [37, 40]
E2E optimization	Not E2E	[24–26, 28–31, 33–35, 38]
	E2E	EATS [39], Parallel Tacotron 2 [40]

CNN. (2) CTC alignment. Beliaev et al. [28] leverages a CTC [63] based ASR model to provide the alignments between phoneme and mel-spectrogram sequence. (3) HMM alignment: FastSpeech 2 [29] leverages the HMM-based Montreal forced alignment (MFA) [65] to get the duration. Other works such as DurIAN [30], RobuTrans [31], Parallel Tacotron [33], and Non-Attentive Tacotron [34] use forced alignment or speech recognition tools to get the alignments.

- Internal alignment. AlignTTS [35] 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 [26] 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 [38] leverages a novel monotonic alignment search to extract duration. EATS [39] 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 [24–26, 28–31, 33–35, 38] 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 a guiding signal (gradients) from the mel-spectrogram loss.
- End-to-end optimization. In order to jointly optimize the duration to achieve better prosody, EATS [39] 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 [40] follows the practice of EATS to ensure differentiable duration prediction. NaturalSpeech [66] also leverages a differentiable durator for end-to-end duration modeling. Non-Attentive Tacotron [34] proposes a semi-supervised learning for duration prediction, where the predicted duration can be used for upsampling if no duration label is available.

9.3 Improving Autoregressive Generation

For the exposure bias and error propagation problems in the autoregressive generation, the works can also be divided into two aspects: (1) improving autoregressive

generation to alleviate the exposure bias and error propagation problems [41–44], and (2) removing autoregressive generation and instead using non-autoregressive generation [21, 24, 29, 39].

9.3.1 Enhancing AR Generation

Autoregressive sequence generation usually suffers from exposure bias and error propagation [67, 68]. Exposure bias refers to that the sequence generation model is usually trained by taking the previous ground-truth value as input (i.e., teacher-forcing), but generates the sequence autoregressively by taking the 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 exposure bias and error propagation issues. Guo et al. [41] leverage professor forcing [69] to alleviate the mismatch between the different distributions of real and predicted data. Liu et al. [43] conduct teacher-student distillation [70–72] 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 that in the left part due to error propagation, some works leverage both left-to-right and right-to-left generations [73] for data augmentation [44] and regularization [45]. Vainer and Dušek [25] 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. It is worthing mention that the bottleneck structure and dropout of the Pre-net introduced in Tacotron [9] and later analyzed by MultiSpeech [19] can also alleviate the exposure bias by reducing the information needed from the previous frames.

9.3.2 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 [21] and Flow-TTS [22] use positional attention [18] for the text and speech alignment in a parallel generation.

Table 9.4 A new category of TTS according to the alignment learning and AR/NAR generation

Attention?	AR?	
	AR	Non-AR
Attention	Tacotron 2 [14], DeepVoice 3 [18]	ParaNet [21], Flow-TTS [22]
Non-attention	DurIAN [30], Non-Att Tacotron [34]	FastSpeech [24, 29], EATS [39]

The remaining works such as FastSpeech [24, 29] and EATS [39] 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 9.4: (1) AR + Attention, such as Tacotron [9, 14], DeepVoice 3 [18], and TransformerTTS [10]. (2) AR + Non-Attention (Duration), such as DurIAN [30], RobuTrans [31], and Non-Attentive Tacotron [34]. (3) Non-AR + Attention, such as ParaNet [21], Flow-TTS [22], and VARA-TTS [23]. (4) Non-AR + Non-Attention, such as FastSpeech 1/2 [24, 29], Glow-TTS [38], and EATS [39].

References

1. He M, Deng Y, He L (2019) Robust sequence-to-sequence acoustic modeling with stepwise monotonic attention for neural TTS. In: Proceedings of the Interspeech 2019, pp 1293–1297
2. Lorenzo-Trueba J, Drugman T, Latorre J, Merritt T, Putrycz B, Barra-Chicote R, Moinet A, Aggarwal V (2019) Towards achieving robust universal neural vocoding. In: Proceedings of the Interspeech 2019, pp 181–185
3. Paul D, Pantazis Y, Stylianou Y (2020) Speaker conditional WaveRNN: towards universal neural vocoder for unseen speaker and recording conditions. In: Proceedings of the Interspeech 2020, pp 235–239
4. Jang W, Lim D, Yoon J (2020) Universal MelGAN: a robust neural vocoder for high-fidelity waveform generation in multiple domains. Preprint. arXiv:2011.09631
5. Jiao Y, Gabrys A, Tinchev G, Putrycz B, Korzekwa D, Klimkov V (2021) Universal neural vocoding with parallel WaveNet. Preprint. arXiv:2102.01106
6. Hwang MJ, Yamamoto R, Song E, Kim JM (2020) TTS-by-TTS: TTS-driven data augmentation for fast and high-quality speech synthesis. Preprint. arXiv:2010.13421
7. Battenberg E, Skerry-Ryan R, Mariooryad S, Stanton D, Kao D, Shannon M, Bagby T (2020) Location-relative attention mechanisms for robust long-form speech synthesis. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 6194–6198
8. Zhang G, Song K, Tan X, Tan D, Yan Y, Liu Y, Wang G, Zhou W, Qin T, Lee T, et al. (2022) Mixed-phoneme BERT: improving BERT with mixed phoneme and sup-phoneme representations for text to speech. Preprint. arXiv:2203.17190
9. Wang Y, Skerry-Ryan R, Stanton D, Wu Y, Weiss RJ, Jaitly N, Yang Z, Xiao Y, Chen Z, Bengio S et al (2017) Tacotron: towards end-to-end speech synthesis. In: Proceedings of the Interspeech 2017, pp 4006–4010
10. Li N, Liu S, Liu Y, Zhao S, Liu M (2019) Neural speech synthesis with Transformer network. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 6706–6713
11. Sotelo J, Mehri S, Kumar K, Santos JF, Kastner K, Courville A, Bengio Y (2017) Char2wav: end-to-end speech synthesis

12. Taigman Y, Wolf L, Polyak A, Nachmani E (2018) VoiceLoop: voice fitting and synthesis via a phonological loop. In: International conference on learning representations
13. Vasquez S, Lewis M (2019) MelNet: a generative model for audio in the frequency domain. Preprint. arXiv:1906.01083
14. Shen J, Pang R, Weiss RJ, Schuster M, Jaitly N, Yang Z, Chen Z, Zhang Y, Wang Y, Skerry-Ryan R et al (2018) Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4779–4783
15. Zhang JX, Ling ZH, Dai LR (2018) Forward attention in sequence-to-sequence acoustic modeling for speech synthesis. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4789–4793
16. Yasuda Y, Wang X, Yamagishi J (2019) Initial investigation of an encoder-decoder end-to-end TTS framework using marginalization of monotonic hard latent alignments. In: Proceedings of the 10th ISCA Speech Synthesis Workshop
17. Tachibana H, Uenoyama K, Aihara S (2018) 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). IEEE, pp 4784–4788
18. Ping W, Peng K, Gibiansky A, Arik SO, Kannan A, Narang S, Raiman J, Miller J (2018) Deep voice 3: 2000-speaker neural text-to-speech. In: Proceedings of the international conference on learning representations, pp 214–217
19. Chen M, Tan X, Ren Y, Xu J, Sun H, Zhao S, Qin T (2020) MultiSpeech: multi-speaker text to speech with Transformer. In: INTERSPEECH, pp 4024–4028
20. Liu P, Wu X, Kang S, Li G, Su D, Yu D (2019) Maximizing mutual information for Tacotron. Preprint. arXiv:1909.01145
21. Peng K, Ping W, Song Z, Zhao K (2020) Non-autoregressive neural text-to-speech. In: International conference on machine learning (PMLR), pp 7586–7598
22. Miao C, Liang S, Chen M, Ma J, Wang S, Xiao J (2020) 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). IEEE, pp 7209–7213
23. Liu P, Cao Y, Liu S, Hu N, Li G, Weng C, Su D (2021) VARA-TTS: non-autoregressive text-to-speech synthesis based on very deep VAE with residual attention. Preprint. arXiv:2102.06431
24. Ren Y, Ruan Y, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2019) FastSpeech: fast, robust and controllable text to speech. In: NeurIPS
25. Vainer J, Dušek O (2020) SpeedySpeech: efficient neural speech synthesis. In: Proceedings of the Interspeech 2020, pp 3575–3579
26. Lim D, Jang W, Gyeonghwan O, Park H, Kim B, Yoon J (2020) JDI-T: jointly trained duration informed transformer for text-to-speech without explicit alignment. In: Proceedings of the Interspeech 2020, pp 4004–4008
27. Łańcucki A (2020) FastPitch: parallel text-to-speech with pitch prediction. Preprint, arXiv:2006.06873
28. Beliaev S, Rebryk Y, Ginsburg B (2020) TalkNet: fully-convolutional non-autoregressive speech synthesis model. Preprint. arXiv:2005.05514
29. Ren Y, Hu C, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2021) FastSpeech 2: fast and high-quality end-to-end text to speech. In: International conference on learning representations. https://openreview.net/forum?id=piLPYqxtWuA
30. Yu C, Lu H, Hu N, Yu M, Weng C, Xu K, Liu P, Tuo D, Kang S, Lei G et al (2020) DurIAN: duration informed attention network for speech synthesis. In: Proceedings of the Interspeech 2020, pp 2027–2031
31. Li N, Liu Y, Wu Y, Liu S, Zhao S, Liu M (2020) RobuTrans: a robust transformer-based text-to-speech model. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 8228–8235
32. Okamoto T, Toda T, Shiga Y, Kawai H (2019) Tacotron-based acoustic model using phoneme alignment for practical neural text-to-speech systems. In: 2019 IEEE automatic speech recognition and understanding workshop (ASRU). IEEE, pp 214–221

33. Elias I, Zen H, Shen J, Zhang Y, Jia Y, Weiss R, Wu Y (2020) Parallel Tacotron: non-autoregressive and controllable TTS. Preprint. arXiv:2010.11439
34. Shen J, Jia Y, Chrzanowski M, Zhang Y, Elias I, Zen H, Wu Y (2020) Non-attentive Tacotron: robust and controllable neural TTS synthesis including unsupervised duration modeling. Preprint. arXiv:2010.04301
35. Zeng Z, Wang J, Cheng N, Xia T, Xiao J (2020) 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). IEEE, pp 6714–6718
36. Li N, Liu S, Liu Y, Zhao S, Liu M, Zhou M (2020) MoBoAligner: a neural alignment model for non-autoregressive TTS with monotonic boundary search. In: Proceedings of the Interspeech 2020, pp 3999–4003
37. Miao C, Liang S, Liu Z, Chen M, Ma J, Wang S, Xiao J (2020) EfficientTTS: an efficient and high-quality text-to-speech architecture. Preprint. arXiv:2012.03500
38. Kim J, Kim S, Kong J, Yoon S (2020) Glow-TTS: a generative flow for text-to-speech via monotonic alignment search. Adv Neural Inf Process Syst 33
39. Donahue J, Dieleman S, Bińkowski M, Elsen E, Simonyan K (2021) End-to-end adversarial text-to-speech. In: International conference on learning representations
40. Elias I, Zen H, Shen J, Zhang Y, Ye J, Skerry-Ryan R, Wu Y (2021) Parallel Tacotron 2: a non-autoregressive neural TTS model with differentiable duration modeling. Preprint. arXiv:2103.14574
41. Guo H, Soong FK, He L, Xie L (2019) A new GAN-based end-to-end TTS training algorithm. In: Proceedings of the Interspeech 2019, pp 1288–1292
42. Liu R, Yang J, Liu M (2019) A new end-to-end long-time speech synthesis system based on Tacotron2. In: Proceedings of the 2019 international symposium on signal processing systems, pp 46–50
43. Liu R, Sisman B, Li J, Bao F, Gao G, Li H (2020) Teacher-student training for robust Tacotron-based TTS. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 6274–6278
44. Ren Y, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2019) Almost unsupervised text to speech and automatic speech recognition. In: International conference on machine learning (PMLR), pp 5410–5419
45. Zheng Y, Tao J, Wen Z, Yi J (2019) Forward-backward decoding sequence for regularizing end-to-end tts. IEEE/ACM Trans Audio Speech Lang Process 27(12):2067–2079
46. Fang W, Chung YA, Glass J (2019) Towards transfer learning for end-to-end speech synthesis from deep pre-trained language models. Preprint. arXiv:1906.07307
47. Hayashi T, Watanabe S, Toda T, Takeda K, Toshniwal S, Livescu K (2019) Pre-trained text embeddings for enhanced text-to-speech synthesis. In: Proceedings of the Interspeech 2019, pp 4430–4434
48. Xiao Y, He L, Ming H, Soong FK (2020) 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). IEEE, pp 6704–6708
49. Jia Y, Zen H, Shen J, Zhang Y, Wu Y (2021) PnG BERT: augmented BERT on phonemes and graphemes for neural TTS. Preprint. arXiv:2103.15060
50. Guo H, Soong FK, He L, Xie L (2019) Exploiting syntactic features in a parsed tree to improve end-to-end TTS. In: Proceedings of the Interspeech 2019, pp 4460–4464
51. Zhou Y, Song C, Li J, Wu Z, Meng H (2021) Dependency parsing based semantic representation learning with graph neural network for enhancing expressiveness of text-to-speech. Preprint. arXiv:2104.06835
52. Zeng Z, Wang J, Cheng N, Xiao J (2020) Prosody learning mechanism for speech synthesis system without text length limit. In: Proceedings of the Interspeech 2020, pp 4422–4426
53. Chen J, Tan X, Luan J, Qin T, Liu TY (2020) HiFiSinger: towards high-fidelity neural singing voice synthesis. Preprint. arXiv:2009.01776

54. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. Preprint. arXiv:1409.0473
55. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
56. Chorowski J, Bahdanau D, Serdyuk D, Cho K, Bengio Y (2015) Attention-based models for speech recognition. In: Proceedings of the 28th international conference on neural information processing systems, vol 1, pp 577–585
57. Chan W, Jaitly N, Le Q, Vinyals O (2016) Listen, attend and spell: a neural network for large vocabulary conversational speech recognition. In: 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4960–4964
58. Chiu CC, Sainath TN, Wu Y, Prabhavalkar R, Nguyen P, Chen Z, Kannan A, Weiss RJ, Rao K, Gonina E et al (2018) State-of-the-art speech recognition with sequence-to-sequence models. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4774–4778
59. Graves A (2013) Generating sequences with recurrent neural networks. Preprint. arXiv:1308.0850
60. Raffel C, Luong MT, Liu PJ, Weiss RJ, Eck D (2017) Online and linear-time attention by enforcing monotonic alignments. In: International conference on machine learning (PMLR), pp 2837–2846
61. Chiu CC, Raffel C (2018) Monotonic chunkwise attention. In: International conference on learning representations
62. Tian Q, Zhang Z, Liu C, Lu H, Chen L, Wei B, He P, Liu S (2020) FeatherTTS: robust and efficient attention based neural TTS. Preprint. arXiv:2011.00935
63. Graves A, Fernández S, Gomez F, Schmidhuber J (2006) Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd international conference on Machine learning, pp 369–376
64. Wightman CW, Talkin DT (1997) The aligner: Text-to-speech alignment using Markov models. In: Progress in speech synthesis. Springer, pp 313–323
65. McAuliffe M, Socolof M, Mihuc S, Wagner M, Sonderegger M (2017) Montreal forced aligner: trainable text-speech alignment using kaldi. In: Interspeech, vol 2017, pp 498–502
66. Tan X, Chen J, Liu H, Cong J, Zhang C, Liu Y, Wang X, Leng Y, Yi Y, He L et al (2022) NaturalSpeech: end-to-end text to speech synthesis with human-level quality. Preprint. arXiv:2205.04421
67. Bengio S, Vinyals O, Jaitly N, Shazeer N (2015) Scheduled sampling for sequence prediction with recurrent neural networks. In: Proceedings of the 28th international conference on neural information processing systems, vol 1, pp 1171–1179
68. Wu L, Tan X, He D, Tian F, Qin T, Lai J, Liu TY (2018) 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, pp 3602–3611
69. Goyal A, Lamb A, Zhang Y, Zhang S, Courville A, Bengio Y (2016) Professor forcing: a new algorithm for training recurrent networks. In: Proceedings of the 30th international conference on neural information processing systems, pp 4608–4616
70. Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. Preprint. arXiv:1503.02531
71. Kim Y, Rush AM (2016) Sequence-level knowledge distillation. In: Proceedings of the 2016 conference on empirical methods in natural language processing, pp 1317–1327
72. Tan X, Ren Y, He D, Qin T, Liu TY (2019) Multilingual neural machine translation with knowledge distillation. In: International conference on learning representations. https://openreview.net/forum?id=S1gUsoR9YX
73. Tan X, Xia Y, Wu L, Qin T (2019) Efficient bidirectional neural machine translation. Preprint. arXiv:1908.09329

Chapter 10
Model-Efficient TTS

Abstract Text-to-speech synthesis systems are usually deployed in different environments such as cloud-edge-end devices, which require fast synthesis speed, small memory storage, and low computation cost. However, early neural TTS models usually leverage deep neural networks that are usually with large computation/memory/time costs. In this chapter, we introduce the technologies for model-efficient TTS. We categorize these technologies according to the methods they use as follows: (1) Parallel generation, which can increase the parallelism of the computation and improve the inference (or training) speed. (2) Lightweight modeling, which aims to develop lightweight and efficient models with small model sizes, low computation, and fast inference speed. (3) Efficient modeling with domain knowledge, which designs efficient models by leveraging the domain knowledge of speech.

Keywords Model-efficient TTS · Parallel generation · Lightweight modeling · Domain knowledge

Text-to-speech synthesis systems are usually deployed in different environments with various cloud-edge-end devices, which require fast synthesis speed, small memory storage, and low computation cost. However, early neural TTS models are usually with large computation/memory/time costs. In this chapter, we introduce the technologies for model-efficient TTS.

Since technologies can reduce the cost of computation, memory, or inference time at the same time, we do not categorize the technologies according to the aspects of computation/memory/time. Instead, we categorize according to the methods they use as follows: (1) Parallel generation, as introduced in Sect. 10.1, which can increase the parallelism of the computation and improve the inference (or training) speed. (2) Lightweight modeling, as introduced in Sect. 10.2, which aims to develop lightweight and efficient models with small model sizes, low computation, and fast inference speed. (3) Efficient modeling with domain knowledge, as introduced in Sect. 10.3, which designs efficient models by leveraging the domain knowledge of speech.

X. Tan, *Neural Text-to-Speech Synthesis*, Artificial Intelligence: Foundations, Theory, and Algorithms, https://doi.org/10.1007/978-981-99-0827-1_10

10.1 Parallel Generation

Previous neural TTS models usually adopt autoregressive mel-spectrogram and waveform generation, which are very slow considering the long speech sequence (e.g., a 5-s speech has 500 mel-spectrograms if hop size is 10 ms, and 120 k waveform points if the sampling rate is 24 kHz). To solve this problem, parallel generation has been leveraged to speed up the inference of TTS models.

Table 10.1 summarizes some typical modeling paradigms for sequence generation, the corresponding TTS models, and time complexity in training and inference. As can be seen, TTS models that use RNN-based autoregressive models [1–4] are slow in both training and inference, with $O(N)$ computation, where N is the sequence length. To avoid the slow training time caused by RNN structure, DeepVoice 3 [5] and TransformerTTS [6] leverage CNN or self-attention-based structure that can support parallel training but still require autoregressive inference. In the following subsections, we introduce different parallel generation methods, including (1) parallel generation with CNN/Transformer (Sect. 10.1.1), which modifies CNN/Transformer from autoregressive generation into non-autoregressive generation; (2) parallel generation with GAN/VAE/Flow (Sect. 10.1.2), which leverages generative models with either a non-iterative (GAN/VAE) or iterative process (Flow) for a parallel generation; (3) parallel generation with Diffusion (Sect. 10.1.3), which leverages diffusion models with an iterative process for a parallel generation.

10.1.1 Non-Autoregressive Generation with CNN or Transformer

To speed up inference, FastSpeech 1/2 [7, 8] design a feed-forward Transformer that leverages self-attention structure for both parallel training and inference, where the computation is reduced to $O(1)$. ParaNet [9] is a fully convolutional-based

Table 10.1 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. $O(T)^*$ means that there are some methods to speed up the inference of diffusion model, as introduced in Sect. 10.1.3

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

non-autoregressive model that can speed up the mel-spectrogram generation. To generate speech sequence in parallel with good voice quality, FastSpeech leverages an autoregressive teacher model for data distillation, similar to that used in Parallel WaveNet [10] and ClariNet [11]. FastSpeech and Parallel WaveNet can both be regarded as an inverse autoregressive flow [12], with parallel inference but require teacher distillation for parallel training, as described in Sect. 6.2.2. FastSpeech 2 [8] improves FastSpeech by modeling more variance information, without the need for teacher distillation.

Besides pure non-autoregressive generation, there are other methods that combine autoregressive and non-autoregressive generation. For example, [13] proposes a semi-autoregressive mode for mel-spectrogram generation, where the mel-spectrograms are generated in an autoregressive manner inside each phoneme while in a non-autoregressive mode for different phonemes.

10.1.2 Non-Autoregressive Generation with GAN, VAE, or Flow

Most GAN-based [8, 14–16] and VAE-based [17, 18] models for mel-spectrogram and waveform generation are non-autoregressive, with $O(1)$ computation in both training and inference. As introduced in Sect. 3.3.2, normalizing flows based on bipartite transforms such as WaveGlow [19] and FloWaveNet [20] can ensure 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.

10.1.3 Iterative Generation with Diffusion

Diffusion models [21–25] require multiple diffusion steps T in the forward and reverse process, which increases the computation. However, its training complexity is $O(1)$ according to Algorithm 1 in Sect. 3.3.4 since we only need to forward the model once for a sampling step and data. Note that some diffusion models such as InferGrad [26] take inference procedure into training, and thus its training complexity is $O(T)$.

The inference complexity of the diffusion model is usually $O(T)$. However, there are a lot of methods to speed up the inference of the diffusion model. Here we just briefly overview some methods as follows:

- Improving prior distribution. By choosing a prior distribution that is closer to the data distribution, such as [23, 25, 27], the diffusion and denoising processes can be easier in training and inference, resulting in fewer sampling steps, and thus faster sampling speed.

- Improving the forward/diffusion or reverse/denoising process. By learning the diffusion process [28], choosing a better schedule in reverse process [29], or taking the non-Markov assumption [30], the inference steps can be reduced.
- Combining diffusion model with other methods. We can further combine the diffusion model with other methods such as GAN [31] and VAE [32].
- Speedup with SDE/ODE solver. We can formulate a diffusion model as an SDE or ODE process, and speed up the inference with some numerical SDE/ODE solvers [33].

10.2 Lightweight Modeling

While non-autoregressive generation can fully leverage the parallel computation for inference speedup, the number of model parameters and total computation cost is not reduced, which makes it costly when deploying on mobile phones or embedded devices since the parallel computation capabilities and memory storage in these devices are not powerful enough. Therefore, we need to design lightweight and efficient models with small model sizes and less computation costs to satisfy different devices, even using autoregressive generation. Some widely used techniques for designing lightweight models include model compression (e.g., pruning, quantization, knowledge distillation [34]) and neural architecture search [35, 36], etc. We also introduce other modeling tricks for lightweight model design.

10.2.1 Model Compression

WaveRNN [37] uses techniques like dual softmax, weight pruning, and subscale prediction to speed up inference. SqueezeWave [38] 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 [39] compress the model parameters of LPCNet with tensor decomposition. Hsu and Lee [40] proposes a heavily compressed flow-based model to reduce computational resources, and a WaveNet-based post-filter to maintain audio quality.

10.2.2 Neural Architecture Search

LightSpeech [35] leverages neural architecture search [41, 42] to find lightweight architectures to further speed up the inference of FastSpeech 2 [8] by 6.5× while maintaining voice quality.

10.2.3 Other Technologies

DeviceTTS [43] leverages the model structure of DFSMN [44] and mix-resolution decoder to predict multiple frames in one decoding step to speed up inference. LVC-Net [45] 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 4× without any degradation in sound quality.

10.3 Efficient Modeling with Domain Knowledge

Domain knowledge from speech can be leveraged to speed up inference, such as linear prediction [4], multiband modeling [46–48], subscale prediction [37], multi-frame prediction [1, 13, 43, 49, 50], streaming synthesis [51], etc.

10.3.1 Linear Prediction

LPCNet [4] 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.

10.3.2 Multiband Modeling

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 [46], multi-band MelGAN [47], subband WaveNet [52], and multi-band LPCNet [48, 53].

10.3.3 Subscale Prediction

WaveRNN [37] leverages subscale prediction to speed up inference.

10.3.4 Multi-Frame Prediction

In Tacotron 1/2, to help the attention alignment learning [1] between character/phoneme sequence and mel-spectrogram sequence, the decoder predicts multiple frames at each autoregressive step [54]. As a by-product, it can also help speed up the inference process.

10.3.5 Streaming or Chunk-Wise Synthesis

Streaming TTS [51, 55–61] synthesizes speech once some input tokens are coming, without waiting for the whole input sentence, which can also speed up inference.

10.3.6 Other Technologies

Bunched LPCNet [62] reduces the computation complexity of LPCNet with sample bunching and bit bunching, achieving more than 2× speedup. FFTNet [63] uses a simple architecture to mimic the Fast Fourier Transform (FFT), which can generate audio samples in real time. Okamoto et al. [64] further enhances FFTNet with noise shaping and subband techniques, improving the voice quality while keeping a small model size. Popov et al. [65] 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. [66] accelerate DCTTS [67] with network reduction and fidelity improvement techniques such as group highway activation, which can synthesize speech in real time with a single CPU thread.

References

1. Wang Y, Skerry-Ryan R, Stanton D, Wu Y, Weiss RJ, Jaitly N, Yang Z, Xiao Y, Chen Z, Bengio S et al (2017) Tacotron: Towards end-to-end speech synthesis. In: Proceedings of the Interspeech 2017, pp 4006–4010
2. Shen J, Pang R, Weiss RJ, Schuster M, Jaitly N, Yang Z, Chen Z, Zhang Y, Wang Y, Skerry-Ryan R et al (2018) Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4779–4783
3. Mehri S, Kumar K, Gulrajani I, Kumar R, Jain S, Sotelo J, Courville A, Bengio Y (2017) SampleRNN: an unconditional end-to-end neural audio generation model. In: International conference on learning representations

4. Valin JM, Skoglund J (2019) LPCNet: improving neural speech synthesis through linear prediction. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 5891–5895
5. Ping W, Peng K, Gibiansky A, Arik SO, Kannan A, Narang S, Raiman J, Miller J (2018) Deep voice 3: 2000-speaker neural text-to-speech. In: Proceedings of the international conference on learning representations, pp 214–217
6. Li N, Liu S, Liu Y, Zhao S, Liu M (2019) Neural speech synthesis with Transformer network. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 6706–6713
7. Ren Y, Ruan Y, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2019) FastSpeech: fast, robust and controllable text to speech. In: NeurIPS
8. Ren Y, Hu C, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2021) FastSpeech 2: fast and high-quality end-to-end text to speech. In: International conference on learning representations. https://openreview.net/forum?id=piLPYqxtWuA
9. Peng K, Ping W, Song Z, Zhao K (2020) Non-autoregressive neural text-to-speech. In: International conference on machine learning (PMLR), pp 7586–7598
10. van der Oord A, Li Y, Babuschkin I, Simonyan K, Vinyals O, Kavukcuoglu K, Driessche G, Lockhart E, Cobo L, Stimberg F et al (2018) Parallel WaveNet: fast high-fidelity speech synthesis. In: International conference on machine learning (PMLR), pp 3918–3926
11. Ping W, Peng K, Chen J (2018) ClariNet: parallel wave generation in end-to-end text-to-speech. In: International conference on learning representations
12. Kingma DP, Salimans T, Jozefowicz R, Chen X, Sutskever I, Welling M (2016) Improved variational inference with inverse autoregressive flow. Adv Neural Inf Process Syst 29:4743–4751
13. Wang D, Deng L, Zhang Y, Zheng N, Yeung YT, Chen X, Liu X, Meng H (2021) FCL-TACO2: towards fast, controllable and lightweight text-to-speech synthesis. In: ICASSP 2021 - 2021 IEEE international conference on acoustics, speech and signal processing
14. Kumar K, Kumar R, de Boissiere T, Gestin L, Teoh WZ, Sotelo J, de Brébisson A, Bengio Y, Courville A (2019) MelGAN: generative adversarial networks for conditional waveform synthesis. In: NeurIPS
15. Kong J, Kim J, Bae J (2020) HiFi-GAN: generative adversarial networks for efficient and high fidelity speech synthesis. Adv Neural Inf Process Syst 33
16. Donahue J, Dieleman S, Bińkowski M, Elsen E, Simonyan K (2021) End-to-end adversarial text-to-speech. In: International conference on learning representations
17. Kim J, Kong J, Son J (2021) Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. Preprint. arXiv:2106.06103
18. Tan X, Chen J, Liu H, Cong J, Zhang C, Liu Y, Wang X, Leng Y, Yi Y, He L, et al. (2022) NaturalSpeech: end-to-end text to speech synthesis with human-level quality. Preprint. arXiv:2205.04421
19. Prenger R, Valle R, Catanzaro B (2019) WaveGlow: a flow-based generative network for speech synthesis. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 3617–3621
20. Kim S, Lee SG, Song J, Kim J, Yoon S (2019) FloWaveNet: a generative flow for raw audio. In: International conference on machine learning (PMLR), pp 3370–3378
21. Chen N, Zhang Y, Zen H, Weiss RJ, Norouzi M, Chan W (2021) WaveGrad: estimating gradients for waveform generation. In: International conference on learning representations
22. Kong Z, Ping W, Huang J, Zhao K, Catanzaro B (2021) DiffWave: a versatile diffusion model for audio synthesis. In: International conference on learning representations
23. Lee S-G, Kim H, Shin C, Tan X, Liu C, Meng Q, Qin T, Chen W, Yoon S, Liu TY (2021) PriorGrad: improving conditional denoising diffusion models with data-driven adaptive prior. Preprint. arXiv:2106.06406
24. Jeong M, Kim H, Cheon SJ, Choi BJ, Kim NS (2021) Diff-TTS: a denoising diffusion model for text-to-speech. Preprint. arXiv:2104.01409
25. Popov V, Vovk I, Gogoryan V, Sadekova T, Kudinov M (2021) Grad-TTS: a diffusion probabilistic model for text-to-speech. Preprint. arXiv:2105.06337

26. Chen Z, Tan X, Wang K, Pan S, Mandic D, He L, Zhao S (2022) InferGrad: improving diffusion models for vocoder by considering inference in training. In: ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 8432–8436
27. Koizumi Y, Zen H, Yatabe K, Chen N, Bacchiani M (2022) SpecGrad: diffusion probabilistic model based neural vocoder with adaptive noise spectral shaping. Preprint. arXiv:2203.16749
28. Kingma D, Salimans T, Poole B, Ho J (2021) Variational diffusion models. Adv Neural Inf Process Syst 34:21696–21707
29. Nichol AQ, Dhariwal P (2021) Improved denoising diffusion probabilistic models. In: International conference on machine learning (PMLR), pp 8162–8171
30. Song J, Meng C, Ermon S (2020) Denoising diffusion implicit models. Preprint. arXiv:2010.02502
31. Xiao Z, Kreis K, Vahdat A (2021) Tackling the generative learning trilemma with denoising diffusion GANs. In: International conference on learning representations
32. Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B (2022) High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10684–10695
33. Lu C, Zhou Y, Bao F, Chen J, Li C, Zhu J (2022) DPM-Solver: a fast ODE solver for diffusion probabilistic model sampling in around 10 steps. Preprint. arXiv:2206.00927
34. Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. Preprint. arXiv:1503.02531
35. Luo R, Tan X, Wang R, Qin T, Li J, Zhao S, Chen E, Liu TY (2021) LightSpeech: lightweight and fast text to speech with neural architecture search. In: 2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
36. Xu J, Tan X, Luo R, Song K, Li J, Qin T, Liu TY (2021) 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 and data mining
37. Kalchbrenner N, Elsen E, Simonyan K, Noury S, Casagrande N, Lockhart E, Stimberg F, van der Oord A, Dieleman S, Kavukcuoglu K (2018) Efficient neural audio synthesis. In: International conference on machine learning (PMLR), pp 2410–2419
38. Zhai B, Gao T, Xue F, Rothchild D, Wu B, Gonzalez JE, Keutzer K (2020) SqueezeWave: extremely lightweight vocoders for on-device speech synthesis. Preprint. arXiv:2001.05685
39. Kanagawa H, Ijima Y (2020) Lightweight LPCNet-based neural vocoder with tensor decomposition. In: Proceedings of the Interspeech 2020, pp 205–209
40. Hsu P-C, Lee Hy (2020) WG-WaveNet: real-time high-fidelity speech synthesis without gpu. In: Proceedings of the Interspeech 2020, pp 210–214
41. Zoph B, Le QV (2016) Neural architecture search with reinforcement learning. Preprint. arXiv:1611.01578
42. Luo R, Tan X, Wang R, Qin T, Chen E, Liu TY (2020) Neural architecture search with GBDT. Preprint. arXiv:2007.04785
43. Huang Z, Li H, Lei M (2020) DeviceTTS: a small-footprint, fast, stable network for on-device text-to-speech. Preprint. arXiv:2010.15311
44. Zhang S, Lei M, Yan Z, Dai L (2018) Deep-FSMN for large vocabulary continuous speech recognition. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 5869–5873
45. Zeng Z, Wang J, Cheng N, Xiao J (2021) Lvcnet: Efficient condition-dependent modeling network for waveform generation. Preprint. arXiv:2102.10815
46. Yu C, Lu H, Hu N, Yu M, Weng C, Xu K, Liu P, Tuo D, Kang S, Lei G et al (2020) DurIAN: duration informed attention network for speech synthesis. In: Proceedings of the Interspeech 2020, pp 2027–2031
47. Yang G, Yang S, Liu K, Fang P, Chen W, Xie L (2020) Multi-band MelGAN: faster waveform generation for high-quality text-to-speech. Preprint. arXiv:2005.05106
48. Cui Y, Wang X, He L, Soong FK (2020) An efficient subband linear prediction for LPCNet-based neural synthesis. In: INTERSPEECH, pp 3555–3559

49. Zen H, Agiomyrgiannakis Y, Egberts N, Henderson F, Szczepaniak P (2016) Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizers for mobile devices. In: Proceedings of the Interspeech 2016, pp 2273–2277
50. Liu R, Sisman B, Lin Y, Li H (2021) FastTalker: a neural text-to-speech architecture with shallow and group autoregression. Neural Netw 141:306–314
51. Ellinas N, Vamvoukakis G, Markopoulos K, Chalamandaris A, Maniati G, Kakoulidis P, Raptis S, Sung JS, Park H, Tsiakoulis P (2020) High quality streaming speech synthesis with low, sentence-length-independent latency. In: Proceedings of the Interspeech 2020, pp 2022–2026
52. Okamoto T, Tachibana K, Toda T, Shiga Y, Kawai H (2018) 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). IEEE, pp 5654–5658
53. Tian Q, Zhang Z, Lu H, Chen LH, Liu S (2020) Featherwave: an efficient high-fidelity neural vocoder with multi-band linear prediction. In: Proceedings of the Interspeech 2020, pp 195–199
54. Chen M, Tan X, Ren Y, Xu J, Sun H, Zhao S, Qin T (2020) MultiSpeech: multi-speaker text to speech with Transformer. In: INTERSPEECH, pp 4024–4028
55. Ma M, Zheng B, Liu K, Zheng R, Liu H, Peng K, Church K, Huang L (2020) Incremental text-to-speech synthesis with prefix-to-prefix framework. In: Proceedings of the 2020 conference on empirical methods in natural language processing: findings, pp 3886–3896
56. Stephenson B, Besacier L, Girin L, Hueber T (2020) What the future brings: Investigating the impact of lookahead for incremental neural TTS. In: Proceedings of the Interspeech 2020, pp 215–219
57. Yanagita T, Sakti S, Nakamura S (2019) 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, pp 183–188
58. Stephenson B, Hueber T, Girin L, Besacier L (2021) Alternate endings: improving prosody for incremental neural TTS with predicted future text input. Preprint. arXiv:2102.09914
59. Mohan DSR, Lenain R, Foglianti L, Teh TH, Staib M, Torresquintero A, Gao J (2020) Incremental text to speech for neural sequence-to-sequence models using reinforcement learning. In: Proceedings of the Interspeech 2020, pp 3186–3190
60. Saeki T, Takamichi S, Saruwatari H (2021) Low-latency incremental text-to-speech synthesis with distilled context prediction network. In: 2021 IEEE automatic speech recognition and understanding workshop (ASRU). IEEE, pp 749–756
61. Saeki T, Takamichi S, Saruwatari H (2021) Incremental text-to-speech synthesis using pseudo lookahead with large pretrained language model. IEEE Signal Process Lett 28:857–861
62. Vipperla R, Park S, Choo K, Ishtiaq S, Min K, Bhattacharya S, Mehrotra A, Ramos AGC, Lane ND (2020) Bunched LPCNet: vocoder for low-cost neural text-to-speech systems. In: Proceedings of the Interspeech 2020, pp 3565–3569
63. Jin Z, Finkelstein A, Mysore GJ, Lu J (2018) FFTNet: a real-time speaker-dependent neural vocoder. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 2251–2255
64. Okamoto T, Toda T, Shiga Y, Kawai H (2018) Improving FFTNet vocoder with noise shaping and subband approaches. In: 2018 IEEE spoken language technology workshop (SLT). IEEE, pp 304–311
65. Popov V, Kamenev S, Kudinov M, Repyevsky S, Sadekova T, Bushaev VK, Parkhomenko D (2020) Fast and lightweight on-device TTS with Tacotron2 and LPCNet. In: Proceedings of the Interspeech 2020, pp 220–224
66. Kang M, Lee J, Kim S, Kim I (2021) Fast DCTTS: efficient deep convolutional text-to-speech. Preprint. arXiv:2104.00624
67. Tachibana H, Uenoyama K, Aihara S (2018) 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). IEEE, pp 4784–4788

Chapter 11
Data-Efficient TTS

Abstract Deep learning-based TTS models require a large amount of training data to learn the mapping patterns between text and speech as well as the complicated distributions in speech. However, in many scenarios, the training data is usually limited, which hinders the effective learning of neural TTS models. In this chapter, we introduce data-efficient methods to solve the low data resource issue in TTS. Generally speaking, there are different scenarios with low data resources: (1) language level, where there lack of training data when we want to build TTS models for a language, and (2) speaker level, where there lack of training data when we want to build TTS models for a speaker. Thus, we mainly introduce data-efficient TTS methods from the two scenarios in this chapter.

Keywords Data-efficient TTS · Self/semi/un-supervised learning · Cross-lingual transfer · Adaptive TTS · Few/zero-shot learning

Deep learning-based TTS models require a large amount of training data to learn the mapping patterns between text and speech as well as the complicated distributions in speech. However, in many scenarios, the training data is usually limited, which hinders the effective learning of neural TTS models. In this chapter, we introduce data-efficient methods to solve the low data resource issue in TTS.

Generally speaking, there are different scenarios with low data resources: (1) language level, where there is a lack of training data when we want to build TTS models for a language, and (2) speaker level, where there is a lack of training data when we want to build TTS models for a speaker. Thus, we mainly introduce data-efficient TTS methods from the two scenarios in Sects. 11.1 and 11.2.

X. Tan, *Neural Text-to-Speech Synthesis*, Artificial Intelligence: Foundations, Theory, and Algorithms, https://doi.org/10.1007/978-981-99-0827-1_11

Table 11.1 Some representative techniques for low-resource TTS

Techniques	Data	Work
Self-supervised training	Unpaired text or speech	[1–9]
Cross-lingual transfer	Paired text and speech	[10–16]
Semi-supervised training	Unpaired text or speech	[11, 17–19]
Dataset mining in the wild	Paired text and speech	[20–22]
Purely unsupervised learning	Unpaired text and speech	[23–25]

11.1 Language-Level Data-Efficient TTS

Building a high-quality TTS system for a new language usually requires a large amount of high-quality paired text and speech data. However, there are more than 7000 languages in the world,[1] and most languages lack training data for developing TTS systems. As a result, popular commercialized speech services[2] can only support dozens of (or one hundred) 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 systems under low data resource scenarios.

We summarize some representative techniques for low-resource TTS in Table 11.1 and introduce these techniques as follows.

11.1.1 Self-Supervised Training

Although paired text and speech data is 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 language understanding or speech generation capabilities [1–4]. For example, the text encoder in TTS can be enhanced by the pre-trained BERT models [1, 4, 5], and the speech decoder in TTS can be pre-trained through autoregressive mel-spectrogram prediction [1] or jointly trained with voice conversion task [3]. Besides, speech can be quantized into a discrete token sequence to resemble the phoneme or character sequence [6]. 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 a few truly paired text and speech data [7, 8, 26].

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

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

11.1.2 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 [27], and semantic structures [28], pre-training the TTS models on rich-resource languages can help the mapping between text and speech in low-resource languages [10–14, 29–33]. Usually, there are different phoneme sets between rich- and low-resource languages. Thus, [10] proposes to map the embeddings between the phoneme sets from different languages, and LRSpeech [11] discards the pre-trained phoneme embeddings and initializes the phoneme embeddings from scratch for low-resource languages. International phonetic alphabet (IPA) [34] or byte representation [16] is adopted to support arbitrary texts in multiple languages. Besides, language similarity [28] can also be considered when conducting the cross-lingual transfer.

11.1.3 Semi-Supervised Training

Text-to-speech (TTS) and automatic speech recognition (ASR) are two dual tasks [35] and can be leveraged together to improve each other. Techniques like speech chain [18, 19] and back transformation [11, 17, 36] leverage additional unpaired text and speech data to boost the performance of TTS and ASR.

11.1.4 Mining Dataset in the Wild

In some scenarios, there may exist some low-quality paired text and speech data on the Web. Cooper [20] and Hu et al. [21] propose to mine this kind of data and develop sophisticated techniques to train a TTS model. Some techniques such as speech enhancement [37], denoising [38], and disentangling [39, 40] can be leveraged to improve the quality of the speech data mined in the wild.

11.1.5 Purely Unsupervised Learning

There are some methods [23–25] to build TTS systems without any paired text/speech data, but only unpaired text/speech data. They typically leverage an unsupervised speech recognizer with good speech representation learned in a self-supervised way to recognize pseudo transcripts, which is guided to match the distribution of true transcripts by the generator-discriminator framework [41]. After that, they leverage the pseudo transcripts and the corresponding speech to train a

TTS model. This unsupervised learning method relies on the progress in large-scale unsupervised learning on speech representation, which can provide good pseudo transcripts as the paired text/speech data.

11.2 Speaker-Level Data-Efficient TTS

When a speaker lacks training data, we usually adapt a TTS model that is trained on multiple speakers to this speaker, i.e., adaptive TTS. Adaptive TTS[3] is an important feature of TTS that can synthesize voice for any user. It is known by different terms in academia and industry, such as voice adaptation [42], voice cloning [43], custom voice [44], etc. Adaptive TTS has been a hot research topic, e,g., a lot of works in statistical parametric speech synthesis have studied voice adaptation [45–52], and the voice cloning challenge also attracts a lot of participants [53–56]. In an adaptive TTS scenario, 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 [57], or by adapting the TTS models trained on a multi-speaker speech dataset to this target speaker through adaptation or cloning [42, 44]. We mainly introduce the latter since the former involves another voice conversion model.

Generally speaking, when considering adaptive TTS, more adaptation data and parameters will result in better voice quality but incur high data collection cost and model deployment costs. In practice, we aim to adapt as few data and parameters as possible (the ideal case is no adaptation data and parameters are needed) while achieving high adaptation voice quality. To achieve this, on the one hand, we should improve the generalization of the source TTS model and take different adaptation domains into consideration. On the other hand, we should consider the methods with few or no adaptation data and parameters.

Accordingly, we introduce adaptive TTS from several perspectives: (1) Improving generalization for adaptation, which enhances the generalization of the source TTS model to support new speakers. (2) Cross-domain adaption, which adapts the source TTS models to different acoustic conditions, styles, and languages. (3) Few-data adaptation, which uses few data to adapt to a target speaker. (4) Few-parameter adaption, which uses few model parameters to adapt to a target speaker. (5) Zero-shot adaptation, which generalizes the source TTS model to a target speaker without any adaptation data or parameters. We summarize the works in each perspective in Table 11.2 and introduce them in the following subsections.

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

Table 11.2 Speaker-level data-efficient TTS from different perspectives

Category	Topic	Work
Improving generalization	Modeling variation information	[44]
	Increasing data coverage	[13, 58]
Cross-domain adaption	Cross-acoustic adaptation	[44, 59]
	Cross-style adaptation	[60–62]
	Cross-lingual adaptation	[63–65]
Few-data adaption	Transcribed data adaptation	[42–44, 66–70]
	Untranscribed data adaptation	[71–73]
Few-parameter adaptation	–	[42–44]
Zero-shot adaptation	–	[42, 43, 74–76]

11.2.1 Improving Generalization

A key factor in speaker-level data-efficient TTS is to improve the generalization of the 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. [44] 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 the source TTS model is to increase the amount and diversity of training data. Cooper et al. [58] 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 [13] train a universal TTS model with multiple speakers in 50 language locales, which increases the generalization when adapting to a new speaker.

11.2.2 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 the source TTS model and support the styles of target speakers. AdaSpeech [44] 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 [60] adapts a reading-style TTS model to spontaneous style, by designing specific filled pauses adaptation,

rhythm adaptation, and timbre adaptation. Some other works [61, 62] consider the adaptation across different speaking styles, such as Lombard [61] or whisper [62]. Some works [34, 63–65, 77–81] 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.

11.2.3 Few-Data Adaptation

The adaptation data is usually limited for a target speaker, and sometimes there are even only untranscribed speech data (without text transcripts) available. Accordingly, we introduce the methods for few-data adaptation, under the transcribed and untranscribed data settings.

Some works [42–44, 56, 66–70] conduct few-shot adaptation that only uses few paired text and speech data, varying from several minutes to several seconds. Chien et al. [56] explore different speaker embeddings for few-shot adaptation. Yue et al. [82] leverage speech chain [18] for few-shot adaptation. Chen et al. [44] and Arık et al. [43] 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.

In many scenarios, only speech data can be collected such as in conversions or online meetings, without the corresponding transcripts. AdaSpeech 2 [71] leverages untranscribed speech data for voice adaptation, with the help of speech reconstruction and latent alignments [73]. Inoue et al. [72] use an ASR model to transcribe the speech data and use the transcribed paired data for voice adaptation.

11.2.4 Few-Parameter Adaptation

For adaptation parameters, the whole TTS model [42, 66], or part of the model (e.g., decoder) [67, 68], or only speaker embedding [42–44] can be fine-tuned. Similarly, fine-tuning more parameters will result in good voice quality but increase memory and deployment costs. 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 100 MB parameters, the total memory storage equals to 100 PB for 1M users, which is a huge memory cost. Some works propose to reduce the number of adaptation parameters to as few as possible while maintaining the adaptation quality. AdaSpeech [44] proposes conditional layer normalization to generate the scale and bias parameters in layer normalization from the speaker embeddings based on contextual parameter generation [83] and only fine-tune the parameters related to the conditional layer normalization and speaker embeddings to achieve good adaptation quality. Moss et

al. [67] 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.

11.2.5 Zero-Shot Adaptation

Some works [42, 43, 74–76, 84] conduct zero-shot adaptation, which leverage a speaker encoder to extract speaker embeddings given 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.

References

1. Chung YA, Wang Y, Hsu WN, Zhang Y, Skerry-Ryan R (2019) 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). IEEE, pp 6940–6944
2. Wang P, Qian Y, Soong FK, He L, Zhao H (2015) Word embedding for recurrent neural network based TTS synthesis. In: 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4879–4883
3. Zhang M, Wang X, Fang F, Li H, Yamagishi J (2019) Joint training framework for text-to-speech and voice conversion using multi-source Tacotron and WaveNet. In: Proceedings of the Interspeech 2019, pp 1298–1302
4. Fang W, Chung YA, Glass J (2019) Towards transfer learning for end-to-end speech synthesis from deep pre-trained language models. Preprint. arXiv:1906.07307
5. Jia Y, Zen H, Shen J, Zhang Y, Wu Y (2021) PnG BERT: augmented BERT on phonemes and graphemes for neural TTS. Preprint. arXiv:2103.15060
6. Tjandra A, Sisman B, Zhang M, Sakti S, Li H, Nakamura S (2019) VQVAE unsupervised unit discovery and multi-scale Code2Spec inverter for zerospeech challenge 2019. In: Proceedings of the Interspeech 2019, pp 1118–1122
7. Liu AH, Tu T, Lee H-Y, Lee L-S (2020) 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). IEEE, pp 7259–7263
8. Tu T, Chen YJ, Liu AH, Lee Hy (2020) Semi-supervised learning for multi-speaker text-to-speech synthesis using discrete speech representation. In: Proceedings of the Interspeech 2020, pp 3191–3195
9. Dunbar E, Algayres R, Karadayi J, Bernard M, Benjumea J, Cao XN, Miskic L, Dugrain C, Ondel L, Black AW et al (2019) The zero resource speech challenge 2019: TTS without T. In: Proceedings of the Interspeech 2019, pp 1088–1092
10. Chen YJ, Tu T, Yeh C-C, Lee HY (2019) End-to-end text-to-speech for low-resource languages by cross-lingual transfer learning. In: Proceedings of the Interspeech 2019, pp 2075–2079
11. Xu J, Tan X, Ren Y, Qin T, Li J, Zhao S, Liu TY (2020) LRSpeech: extremely low-resource speech synthesis and recognition. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery and data mining, pp 2802–2812

12. Azizah K, Adriani M, Jatmiko W (2020) Hierarchical transfer learning for multilingual, multi-speaker, and style transfer DNN-based TTS on low-resource languages. IEEE Access 8:179798–179812
13. Yang J, He L (2020) Towards universal text-to-speech. In: INTERSPEECH, pp 3171–3175
14. de Korte M, Kim J, Klabbers E (2020) Efficient neural speech synthesis for low-resource languages through multilingual modeling. In: Proceedings of the Interspeech 2020, pp 2967–2971
15. Prajwal K, Jawahar C (2021) Data-efficient training strategies for neural TTS systems. In: 8th ACM IKDD CODS and 26th COMAD, pp 223–227
16. He M, Yang J, He L (2021) Multilingual Byte2Speech text-to-speech models are few-shot spoken language learners. Preprint. arXiv:2103.03541
17. Ren Y, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2019) Almost unsupervised text to speech and automatic speech recognition. In: International conference on machine learning (PMLR), pp 5410–5419
18. Tjandra A, Sakti S, Nakamura S (2017) Listening while speaking: speech chain by deep learning. In: 2017 IEEE automatic speech recognition and understanding workshop (ASRU). IEEE, pp 301–308
19. Tjandra A, Sakti S, Nakamura S (2018) Machine speech chain with one-shot speaker adaptation. In: Proceedings of the Interspeech 2018, pp 887–891
20. Cooper EL (2019) Text-to-speech synthesis using found data for low-resource languages. Ph.D Thesis, Columbia University
21. Hu Q, Marchi E, Winarsky D, Stylianou Y, Naik D, Kajarekar S (2019) Neural text-to-speech adaptation from low quality public recordings. In: Speech synthesis workshop, vol 10
22. Cooper E, Wang X, Zhao Y, Yasuda Y, Yamagishi J (2020) Pretraining strategies, waveform model choice, and acoustic configurations for multi-speaker end-to-end speech synthesis. Preprint. arXiv:2011.04839
23. Liu AH, Lai CIJ, Hsu WN, Auli M, Baevskiv A, Glass J (2022) Simple and effective unsupervised speech synthesis. Preprint. arXiv:2204.02524
24. Ni J, Wang L, Gao H, Qian K, Zhang Y, Chang S, Hasegawa-Johnson M (2022) Unsupervised text-to-speech synthesis by unsupervised automatic speech recognition. Preprint. arXiv:2203.15796
25. Lian J, Zhang C, Anumanchipalli GK, Yu D (2022) UTTS: Unsupervised TTS with conditional disentangled sequential variational auto-encoder. Preprint. arXiv:2206.02512
26. Zhang H, Lin Y (2020) Unsupervised learning for sequence-to-sequence text-to-speech for low-resource languages. In: Proceedings of the Interspeech 2020, pp 3161–3165
27. Wind J (1989) The evolutionary history of the human speech organs. Stud Lang Origins 1:173–197
28. Tan X, Chen J, He D, Xia Y, Tao Q, Liu TY (2019) 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), pp 962–972
29. Guo W, Yang H, Gan Z (2018) A DNN-based Mandarin-Tibetan cross-lingual speech synthesis. In: 2018 Asia-Pacific signal and information processing association annual summit and conference (APSIPA ASC). IEEE, pp 1702–1707
30. Tan X, Leng Y, Chen J, Ren Y, Qin T, Liu TY (2019) A study of multilingual neural machine translation. Preprint. arXiv:1912.11625
31. Zhang W, Yang H, Bu X, Wang L (2019) Deep learning for Mandarin-Tibetan cross-lingual speech synthesis. IEEE Access 7:167884–167894
32. Nekvinda T, Dušek O (2020) One model, many languages: meta-learning for multilingual text-to-speech. In: Proceedings of the Interspeech 2020 pp 2972–2976
33. Zhang C, Tan X, Ren Y, Qin T, Zhang K, Liu TY (2021) UWSpeech: speech to speech translation for unwritten languages. In: AAAI association for the advancement of artificial intelligence

34. Hemati H, Borth D (2020) Using IPA-based Tacotron for data efficient cross-lingual speaker adaptation and pronunciation enhancement. Preprint. arXiv:2011.06392
35. Qin T (2020) Dual learning. Springer
36. Chen J, Tan X, Leng Y, Xu J, Wen G, Qin T, Liu TY (2021) Speech-T: transducer for text to speech and beyond. Adv Neural Inf Process Syst 34:6621–6633
37. Valentini-Botinhao C, Yamagishi J (2018) Speech enhancement of noisy and reverberant speech for text-to-speech. IEEE/ACM Trans Audio Speech Lang Process 26(8):1420–1433
38. Zhang C, Ren Y, Tan X, Liu J, Zhang K, Qin T, Zhao S, Liu TY (2021) DenoiSpeech: denoising text to speech with frame-level noise modeling. In: 2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
39. Wang Y, Stanton D, Zhang Y, Skerry-Ryan R, Battenberg E, Shor J, Xiao Y, Jia Y, Ren F, Saurous RA (2018) Style tokens: unsupervised style modeling, control and transfer in end-to-end speech synthesis. In: International conference on machine learning (PMLR), pp 5180–5189
40. Hsu WN, Zhang Y, Weiss RJ, Chung YA, Wang Y, Wu Y, Glass J (2019) 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). IEEE, pp 5901–5905
41. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: NIPS
42. Chen Y, Assael Y, Shillingford B, Budden D, Reed S, Zen H, Wang Q, Cobo LC, Trask A, Laurie B et al (2018) Sample efficient adaptive text-to-speech. In: International conference on learning representations
43. Arık SÖ, Chen J, Peng K, Ping W, Zhou Y (2018) Neural voice cloning with a few samples. In: Proceedings of the 32nd international conference on neural information processing systems, pp 10040–10050
44. Chen M, Tan X, Li B, Liu Y, Qin T, Zhao S, Liu TY (2021) AdaSpeech: Adaptive text to speech for custom voice. In: International conference on learning representations. https://openreview.net/forum?id=Drynvt7gg4L
45. Tamura M, Masuko T, Tokuda K, Kobayashi T (1998) Speaker adaptation for HMM-based speech synthesis system using MLLR. In: The third ESCA/COCOSDA workshop (ETRW) on speech synthesis
46. Yamagishi J, Nose T, Zen H, Ling ZH, Toda T, Tokuda K, King S, Renals S (2009) Robust speaker-adaptive HMM-based text-to-speech synthesis. IEEE Trans Audio Speech Lang Process 17(6):1208–1230
47. Fan Y, Qian Y, Soong FK, He L (2015) Multi-speaker modeling and speaker adaptation for DNN-based TTS synthesis. In: 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4475–4479
48. Wu Z, Swietojanski P, Veaux C, Renals S, King S (2015) A study of speaker adaptation for DNN-based speech synthesis. In: Sixteenth annual conference of the international speech communication association
49. Zhao Y, Saito D, Minematsu N (2016) Speaker representations for speaker adaptation in multiple speakers BLSTM-RNN-based speech synthesis. Space 5(6):7
50. Fan Y, Qian Y, Soong FK, He L (2016) Speaker and language factorization in DNN-based TTS synthesis. In: 2016 IEEE International conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 5540–5544
51. Doddipatla R, Braunschweiler N, Maia R (2017) Speaker adaptation in DNN-based speech synthesis using d-vectors. In: INTERSPEECH, pp 3404–3408
52. Huang Z, Lu H, Lei M, Yan Z (2018) Linear networks based speaker adaptation for speech synthesis. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 5319–5323
53. Xie Q, Tian X, Liu G, Song K, Xie L, Wu Z, Li H, Shi S, Li H, Hong F et al (2021) The multi-speaker multi-style voice cloning challenge 2021. Preprint. arXiv:2104.01818
54. Hu CH, Wu YC, Huang WC, Peng YH, Chen YW, Ku PJ, Toda T, Tsao Y, Wang HM (2021) The AS-NU system for the M2VoC challenge. Preprint. arXiv:2104.03009

55. Tan D, Huang H, Zhang G, Lee T (2021) CUHK-EE voice cloning system for ICASSP 2021 M2VoC challenge. Preprint. arXiv:2103.04699
56. Chien CM, Lin JH, Huang C-Y, Hsu P-C, Lee H-Y (2021) Investigating on incorporating pretrained and learnable speaker representations for multi-speaker multi-style text-to-speech. Preprint. arXiv:2103.04088
57. Huybrechts G, Merritt T, Comini G, Perz B, Shah R, Lorenzo-Trueba J (2020) Low-resource expressive text-to-speech using data augmentation. Preprint. arXiv:2011.05707
58. Cooper E, Lai CI, Yasuda Y, Yamagishi J (2020) Can speaker augmentation improve multi-speaker end-to-end TTS? In: Proceedings of the Interspeech 2020, pp 3979–3983
59. Cong J, Yang S, Xie L, Yu G, Wan G (2020) Data efficient voice cloning from noisy samples with domain adversarial training. In: Proceedings of the Interspeech 2020, pp 811–815
60. Yan Y, Tan X, Li B, Zhang G, Qin T, Zhao S, Shen Y, Zhang WQ, Liu TY (2021) AdaSpeech 3: adaptive text to speech for spontaneous style. In: INTERSPEECH
61. Paul D, Shifas MP, Pantazis Y, Stylianou Y (2020) Enhancing speech intelligibility in text-to-speech synthesis using speaking style conversion. In: Proceedings of the Interspeech 2020, pp 1361–1365
62. Hu Q, Bleisch T, Petkov P, Raitio T, Marchi E, Lakshminarasimhan V (2021) Whispered and lombard neural speech synthesis. In: 2021 IEEE spoken language technology workshop (SLT). IEEE, pp 454–461
63. Zhang Y, Weiss RJ, Zen H, Wu Y, Chen Z, Skerry-Ryan R, Jia Y, Rosenberg A, Ramabhadran B (2019) Learning to speak fluently in a foreign language: multilingual speech synthesis and cross-language voice cloning. In: Proceedings of the Interspeech 2019, pp 2080–2084
64. Chen M, Chen M, Liang S, Ma J, Chen L, Wang S, Xiao J (2019) Cross-lingual, multi-speaker text-to-speech synthesis using neural speaker embedding. In: Proceedings of the Interspeech 2019, pp 2105–2109
65. Liu Z, Mak B (2019) Cross-lingual multi-speaker text-to-speech synthesis for voice cloning without using parallel corpus for unseen speakers. Preprint. arXiv:1911.11601
66. Kons Z, Shechtman S, Sorin A, Rabinovitz C, Hoory R (2019) High quality, lightweight and adaptable TTS using LPCNet. In: Proceedings of the Interspeech 2019, pp 176–180
67. Moss HB, Aggarwal V, Prateek N, González J, Barra-Chicote R (2020) BOFFIN TTS: few-shot speaker adaptation by Bayesian optimization. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 7639–7643
68. Zhang Z, Tian Q, Lu H, Chen LH, Liu S (2020) AdaDurIAN: few-shot adaptation for neural text-to-speech with durian. Preprint. arXiv:2005.05642
69. Choi S, Han S, Kim D, Ha S (2020) Attentron: few-shot text-to-speech utilizing attention-based variable-length embedding. In: Proceedings of the Interspeech 2020, pp 2007–2011
70. Min D, Lee DB, Yang E, Hwang SJ (2021) Meta-StyleSpeech: multi-speaker adaptive text-to-speech generation. Preprint. arXiv:2106.03153
71. Yan Y, Tan X, Li B, Qin T, Zhao S, Shen Y, Liu TY (2021) AdaSpeech 2: adaptive text to speech with untranscribed data. In: 2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
72. Inoue K, Hara S, Abe M, Hayashi T, Yamamoto R, Watanabe S (2020) 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). IEEE, pp 7634–7638
73. Luong HT, Yamagishi J (2020) Nautilus: a versatile voice cloning system. IEEE/ACM Trans Audio Speech Lang Process 28:2967–2981
74. Jia Y, Zhang Y, Weiss RJ, Wang Q, Shen J, Ren F, Chen Z, Nguyen P, Pang R, Moreno IL et al (2018) Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In: Proceedings of the 32nd international conference on neural information processing systems, pp 4485–4495

75. Cooper E, Lai CI, Yasuda Y, Fang F, Wang X, Chen N, Yamagishi J (2020) 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). IEEE, pp 6184–6188
76. Wu Y, Tan X, Li B, He L, Zhao S, Song R, Qin T, Liu TY (2022) Adaspeech 4: adaptive text to speech in zero-shot scenarios. In: INTERSPEECH
77. Zhao S, Nguyen TH, Wang H, Ma B (2020) Towards natural bilingual and code-switched speech synthesis based on mix of monolingual recordings and cross-lingual voice conversion. In: Proceedings of the Interspeech 2020 pp 2927–2931
78. Himawan I, Aryal S, Ouyang I, Kang S, Lanchantin P, King S (2020) 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). IEEE, pp 7629–7633
79. Staib M, Teh TH, Torresquintero A, Mohan DSR, Foglianti L, Lenain R, Gao J (2020) Phonological features for 0-shot multilingual speech synthesis. In: Proceedings of the Interspeech 2020, pp 2942–2946
80. Maiti S, Marchi E, Conkie A (2020) 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), IEEE, pp 7624–7628
81. Zhou X, Tian X, Lee G, Das RK, Li H (2020) 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). IEEE, pp 7614–7618
82. Yue F, Deng Y, He L, Ko T (2021) Exploring machine speech chain for domain adaptation and few-shot speaker adaptation. Preprint. arXiv:2104.03815
83. Platanios EA, Sachan M, Neubig G, Mitchell T (2018) Contextual parameter generation for universal neural machine translation. In: Proceedings of the 2018 conference on empirical methods in natural language processing, pp 425–435
84. Casanova E, Shulby C, Gölge E, Müller NM, de Oliveira FS, Junior AC, da Silva Soares A, Aluisio SM, Ponti MA (2021) SC-GlowTTS: an efficient zero-shot multi-speaker text-to-speech model. Preprint. arXiv:2104.05557

Chapter 12
Beyond Text-to-Speech Synthesis

Abstract In this chapter, we briefly introduce other speech tasks that are related to TTS and discuss their relationships. The closest task to text-to-speech synthesis is singing voice synthesis (SVS). Actually, SVS can be regarded as a subtask of TTS, as SVS shares similar input/output, model pipeline, and methodology with TTS. Voice conversion (VC) is also closely related to TTS since they both aim to synthesize speech. The difference is that TTS synthesizes speech from the text while VC synthesizes speech from another speech. Other related tasks include speech enhancement and separation, which separate different signals from source speech (e.g., separate clean signal from noise in speech enhancement).

Keywords Singing voice synthesis · Voice conversion · Speech enhancement · Speech separation

In this chapter, we briefly introduce other speech tasks that are related to TTS and discuss their relationships. The closest task to text-to-speech synthesis is singing voice synthesis (SVS) [1–5]. Actually, SVS can be regarded as a subtask of TTS, as SVS shares similar input/output, model pipeline, and methodology with TTS. Voice conversion (VC) [6–8] is also closely related to TTS since they both aim to synthesize speech. The difference is that TTS synthesizes speech from the text while VC synthesizes speech from another speech. Other related tasks include speech enhancement and separation [9], which separate different signals from source speech (e.g., separate clean signal from noise in speech enhancement).

12.1 Singing Voice Synthesis

Singing voice synthesis (SVS) is similar to text-to-speech synthesis in that they share similar input/output, modeling pipeline, and methodology. They both take text or discrete symbols (music score for singing voice synthesis, which can be

X. Tan, *Neural Text-to-Speech Synthesis*, Artificial Intelligence: Foundations, Theory, and Algorithms, https://doi.org/10.1007/978-981-99-0827-1_12

roughly regarded as text, pitch, and duration as in TTS) as input, and generate mel-spectrogram or waveform as output. Singing voice synthesis nearly adopts the same pipeline as in TTS that leverages either cascaded acoustic models and vocoders or fully end-to-end models. However, singing voice synthesis has distinctive characteristics that make it challenging to model. We introduce the distinctive challenges in singing voice synthesis and some representative models for singing voice synthesis.

12.1.1 Challenges in Singing Voice Synthesis

We list some challenges in SVS that are unique to TTS: (1) Diverse pitch/duration/energy and text/speaker pairs. We can roughly factorize the elements in speech and singing voice into different aspects: linguistic content, pitch, duration, energy, and speaker identity. As we can see, in TTS, a word or phoneme for a certain speaker usually has a relatively fixed range of pitch, duration, and energy, while in SVS, a word or phoneme for a certain speaker can be combined with different pitch, duration, and energy, which are pre-defined by the music score. As a result, the singing voice has a much large space of compositionality in terms of linguistic content, speaker identity, pitch, duration, and energy, which makes singing voice synthesis challenging. (2) Singing voice is mainly to express emotion while speech voice is mainly to express content. Thus, singing voice requires more expressive singing skills and also high-fidelity audio to convey the expressiveness. As a result, we need to model different singing skills that lack training data, and also model singing voice using a high sampling rate (e.g., 48 kHz), which is challenging.

12.1.2 Representative Models for Singing Voice Synthesis

We just list some works on singing voice synthesis that address the unique challenges in singing voice synthesis: (1) XiaoiceSing [3] collects a large corpus of singing data with over 70 h, which can largely alleviate the diverse pattern distribution. (2) DeepSinger [4] mines a lot of data from the Web. Although the mined data is noisy, it is abundant and helpful to cover a wide range of singing patterns. (3) HiFiSinger [5] models singing voice in 48 kHz sampling rate with sophisticated model designs, which improves the expressiveness of synthesized singing voice. You are encouraged to read more works in the SVS literature [1, 2, 10–14].

12.2 Voice Conversion

Voice conversion (VC) is to convert the speaker identity of a speech utterance from one to another, without modifying the linguistic content. In a broad sense, voice conversion can convert any aspect of a speech, not limited to speaker identity. Both voice conversion and text-to-speech can be regarded as a subtask of speech synthesis.

12.2.1 Brief Overview of Voice Conversion

Traditional voice conversion adopts an analysis-mapping-reconstruction pipeline [8], which first converts the source speech into acoustic features (analysis), maps the features from the source speaker to the target speaker (mapping), and then reconstructs the speech waveform from the acoustic features of target speaker (reconstruction). With the development of neural networks and deep learning, the voice conversion pipeline is more and more neural-based and end-to-end: (1) The acoustic features are simplified, e.g., using mel-spectrograms. (2) The mapping function is implemented with neural networks. (3) The reconstruction is achieved by neural vocoders instead of traditional statistical vocoders. (4) The whole conversion pipeline can be replaced by a fully end-to-end neural model.

An important taxonomy of voice conversion is based on the training data: parallel data or non-parallel data. Since paired training data is usually hard to collect, achieving voice conversion with non-parallel data is attractive and also challenging. Thus, we mainly introduce voice conversion methods based on non-parallel data.

12.2.2 Representative Methods for Voice Conversion

There are different modeling pipelines for voice conversion: (1) PPG-VC, where the source speech is first encoded into a sequence of phonetic posteriorgrams (PPGs) [15] and then converted into target speech [16]; (2) ASR-TTS, where the source speech is first recognized into text using an automatic speech recognition (ASR) model, and then synthesized into target speech using a TTS model [17, 18]; (3) Auto-Encoder, where the source speech is first encoded into a sequence of hidden representations, and then reconstructed into the original speech [19, 20]. To ensure the voice can be converted, the hidden representations are disentangled into speaker-dependent and speaker-independent representations. In inference, we concatenate the speaker-independent representations of the source speech and the speaker-dependent representations of the target speaker and use them to generate the target speech; (4) GAN-based methods, such as CycleGAN-VC [21] or StarGAN-VC [22].

The key to voice conversion is to convert the speaker identity while maintaining the linguistic content. Thus, disentangling the speaker identity and linguistic content is critical for voice conversion. The above methods achieve disentanglement in different ways: (1) Explicit way, including PPG-VC and ASR-TTS, which explicitly obtain the linguistic content from source speech through PPG extraction or text recognition, and then add the target speaker identity when generating target speech. (2) Implicit way, including Auto-Encoder and GAN-based methods, which use some loss constraints such as adversarial loss or cycle-consistency loss to achieve disentanglement on hidden representations and ensure the target speech does not contain source speaker identity information.

12.3 Speech Enhancement/Separation

Speech enhancement and separation [9] can also be regarded as a general form of speech synthesis: generate clean from noisy speech or generate separated speech from mixed speech. Speech enhancement can be regarded as a special case of speech separation tasks since speech enhancement separates clean speech and noise from noisy speech.

Different from TTS, SVS, and voice conversion, speech enhancement and separation do not generate speech from scratch or modify the speaker identity. They just need to separate different signals apart from the source speech. Thus, masking methods are widely used in speech enhancement and separation, where the model predicts a mask, which is added or multiplied on the source speech spectrogram to get the separated speech.

References

1. Nishimura M, Hashimoto K, Oura K, Nankaku Y, Tokuda K (2016) Singing voice synthesis based on deep neural networks. In: Proceedings of the Interspeech, pp 2478–2482
2. Lee J, Choi HS, Jeon CB, Koo J, Lee K (2019) Adversarially trained end-to-end korean singing voice synthesis system. In: Proceedings of the Interspeech 2019, pp 2588–2592
3. Lu P, Wu J, Luan J, Tan X, Zhou L (2020) XiaoiceSing: a high-quality and integrated singing voice synthesis system. In: Proceedings of the Interspeech, 2020 pp 1306–1310
4. Ren Y, Tan X, Qin T, Luan J, Zhao Z, Liu TY (2020) Deepsinger: singing voice synthesis with data mined from the web. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1979–1989
5. Chen J, Tan X, Luan J, Qin T, Liu TY (2020) HiFiSinger: towards high-fidelity neural singing voice synthesis. Preprint. arXiv:2009.01776
6. Stylianou Y (2009) Voice transformation: a survey. In: 2009 IEEE international conference on acoustics, speech and signal processing. IEEE, pp 3585–3588
7. Mohammadi SH, Kain A (2017) An overview of voice conversion systems. Speech Commun 88:65–82

8. Sisman B, Yamagishi J, King S, Li H (2020) An overview of voice conversion and its challenges: from statistical modeling to deep learning. IEEE/ACM Trans Audio Speech Lang Process 29:132–157
9. Wang D, Chen J (2018) Supervised speech separation based on deep learning: an overview. IEEE/ACM Trans Audio Speech Lang Process 26(10):1702–1726
10. Chandna P, Blaauw M, Bonada J, Gómez E (2019) Wgansing: a multi-voice singing voice synthesizer based on the wasserstein-gan. In: 2019 27th European signal processing conference (EUSIPCO). IEEE, pp 1–5
11. Gu Y, Yin X, Rao Y, Wan Y, Tang B, Zhang Y, Chen J, Wang Y, Ma Z (2021) ByteSing: a Chinese singing voice synthesis system using duration allocated encoder-decoder acoustic models and WaveRNN vocoders. In: 2021 12th international symposium on chinese spoken language processing (ISCSLP). IEEE, pp 1–5
12. Zhang Y, Cong J, Xue H, Xie L, Zhu P, Bi M (2022) VISinger: variational inference with adversarial learning for end-to-end singing voice synthesis. In: ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 7237–7241
13. Liu J, Li C, Ren Y, Chen F, Zhao Z (2022) DiffSinger: singing voice synthesis via shallow diffusion mechanism. In: Proceedings of the AAAI conference on artificial intelligence, vol 36, pp 11020–11028
14. Zhang Z, Zheng Y, Li X, Lu L (2022) WeSinger 2: fully parallel singing voice synthesis via multi-singer conditional adversarial training. Preprint. arXiv:2207.01886
15. Sun L, Li K, Wang H, Kang S, Meng H (2016) Phonetic posteriorgrams for many-to-one voice conversion without parallel data training. In: 2016 IEEE international conference on multimedia and expo (ICME). IEEE, pp 1–6
16. Tian X, Chng ES, Li H (2019) A speaker-dependent WaveNet for voice conversion with non-parallel data. In: Proceedings of the Interspeech, pp 201–205
17. Miyoshi H, Saito Y, Takamichi S, Saruwatari H (2017) Voice conversion using sequence-to-sequence learning of context posterior probabilities. In: Proceedings of the Interspeech 2017, pp 1268–1272
18. Biadsy F, Weiss RJ, Moreno PJ, Kanvesky D, Jia Y (2019) Parrotron: an end-to-end speech-to-speech conversion model and its applications to hearing-impaired speech and speech separation. In: Proceedings of the Interspeech 2019, pp 4115–4119
19. Mohammadi SH, Kain A (2014) Voice conversion using deep neural networks with speaker-independent pre-training. In: 2014 IEEE spoken language technology workshop (SLT). IEEE, pp 19–23
20. Qian K, Zhang Y, Chang S, Yang X, Hasegawa-Johnson M (2019) AutoVC: zero-shot voice style transfer with only autoencoder loss. In: International conference on machine learning (PMLR), pp 5210–5219
21. Kaneko T, Kameoka H (2018) CycleGAN-VC: non-parallel voice conversion using cycle-consistent adversarial networks. In: 2018 26th European signal processing conference (EUSIPCO). IEEE, pp 2100–2104
22. Kameoka H, Kaneko T, Tanaka K, Hojo N (2018) StarGAN-VC: non-parallel many-to-many voice conversion using star generative adversarial networks. In: 2018 IEEE spoken language technology workshop (SLT). IEEE, pp 266–273

Part IV
Summary and Outlook

Chapter 13
Summary and Outlook

Abstract In this chapter, we summarize the content we introduce in this book, including the basic model components of TTS and several advanced topics in TTS. We point out some future research directions on neural TTS.

Keywords High-quality speech synthesis · Diverse speech synthesis · Efficient speech synthesis

13.1 Summary

In this book, we introduced neural text-to-speech synthesis, with a focus on (1) the basic model components of TTS including text analyses, acoustic models, vocoders, and fully end-to-end models; (2) several advanced topics in TTS including expressive and controllable TTS, robust TTS, model-efficient TTS, data-efficient TTS, and some tasks beyond TTS. As a quick summary, we list representative TTS models in Table B.1. Due to page limitations, we only introduced core algorithms of TTS; readers can refer to other papers for TTS-related problems and applications, such as singing voice synthesis [1–3], voice conversion [4], speech enhancement/separation [5], and talking face synthesis [6], etc.

13.2 Future Directions

We point out some future research directions on neural TTS according to the end goals of TTS.

X. Tan, *Neural Text-to-Speech Synthesis*, Artificial Intelligence: Foundations, Theory, and Algorithms, https://doi.org/10.1007/978-981-99-0827-1_13

13.2.1 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 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 exploration.
- *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 transferability in inference but suffers 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 well 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, or errors are permitted. However, in casual or conversational talking, human seldom speaks like standard reading. Therefore, better modeling the casual, emotional, and spontaneous styles are critical to improving the naturalness of synthesized speech.

13.2.2 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 lack 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 [7] 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 blocks the applications in mobile, IoT, and other low-end devices due to their limited memory and power consumption. Designing compact and lightweight models with fewer 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 benefit from advanced TTS techniques while reducing carbon emissions to protect our environment.

References

1. Hono Y, Hashimoto K, Oura K, Nankaku Y, Tokuda K (2019) Singing voice synthesis based on generative adversarial networks. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 6955–6959
2. Lu P, Wu J, Luan J, Tan X, Zhou L (2020) XiaoiceSing: a high-quality and integrated singing voice synthesis system. In: Proceedings of the Interspeech 2020, pp 1306–1310
3. Chen J, Tan X, Luan J, Qin T, Liu TY (2020) HiFiSinger: towards high-fidelity neural singing voice synthesis. Preprint. arXiv:2009.01776
4. Sisman B, Yamagishi J, King S, Li H (2020) An overview of voice conversion and its challenges: from statistical modeling to deep learning. IEEE/ACM Trans Audio Speech Lang Process 29:132–157
5. Wang D, Chen J (2018) Supervised speech separation based on deep learning: an overview. IEEE/ACM Trans Audio Speech Lang Process 26(10):1702–1726
6. Chen L, Cui G, Kou Z, Zheng H, Xu C (2020) What comprises a good talking-head video generation? A survey and benchmark. Preprint. arXiv:2005.03201
7. ZeroSpeech, Zero resource speech challenge. (2021) https://www.zerospeech.com/

Appendix A
Resources of TTS

We collect some resources on TTS, including TTS tutorials and keynotes, open-source implementations, TTS challenges, and TTS corpora, as shown in Table A.1.

X. Tan, *Neural Text-to-Speech Synthesis*, Artificial Intelligence: Foundations, Theory, and Algorithms, https://doi.org/10.1007/978-981-99-0827-1

Table A.1 TTS resources

TTS Tutorials and keynotes	
TTS survey paper [1]	https://github.com/tts-tutorial/survey
TTS tutorial at INTERSPEECH 2022 [2]	https://github.com/tts-tutorial/interspeech2022
TTS tutorial at ICASSP 2022 [3]	https://github.com/tts-tutorial/icassp2022
TTS tutorial at IJCAI 2021 [4]	https://github.com/tts-tutorial/ijcai2021
TTS tutorial at ISCSLP 2021 [5]	https://tts-tutorial.github.io/iscslp2021/
TTS tutorial at ISCSLP 2016 [6]	http://staff.ustc.edu.cn/~zhling/download/ISCSLP16_tutorial_DLSPSS.pdf
TTS tutorial at ISCSLP 2014 [7]	https://www.superlectures.com/iscslp2014/tutorial-4-deep-learning-for-speech-generation-and-synthesis
TTS tutorial at IEICE [8]	https://www.slideshare.net/jyamagis/tutorial-on-endtoend-texttospeech-synthesis-part-1-neural-waveform-modeling
Generative models for speech [9]	https://www.youtube.com/watch?v=vEAq_sBf1CA
Generative model-based TTS [10]	https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45882.pdf
Keynote at INTERSPEECH [11]	http://www.sp.nitech.ac.jp/~tokuda/INTERSPEECH2019.pdf
TTS Webinar [12]	https://www.youtube.com/watch?v=MA8PCvmr8B0
Open-source implementations	
ESPnet-TTS [13]	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

(continued)

Table A.1 (continued)

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 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 corpor*a

Corpus	#Hours	#Speakers	SR (kHz)	Language	License
ARCTIC [14]	7	7	16	English	BSD
VCTK [15]	44	109	48	English	CC BY 4.0
Blizzard-2011 [16]	16.6	1	16	English	Non-commercial
Blizzard-2013 [17]	319	1	44.1	English	Non-commercial
LJSpeech [18]	25	1	22.05	English	CC0 1.0
LibriSpeech [19]	982	2484	16	English	CC BY 4.0
LibriTTS [20]	586	2456	24	English	CC BY 4.0
VCC 2016 [21]	2	10	16	Engllish	CC BY 4.0
VCC 2018 [22]	1	12	22.05	English	CC BY 4.0
HiFi-TTS [23]	300	11	44.1	English	CC BY 4.0
TED-LIUM [24]	118	666	/	English	CC BY-NC-ND 3.0
CALLHOME [25]	60	120	8	English	LDC
RyanSpeech [26]	10	1	44.1	English	CC BY-NC-ND
CSMSC [27]	12	1	48	Mandarin	Non-commercial
HKUST [28]	200	2100	8	Mandarin	LDC

(continued)

Table A.1 (continued)

AISHELL-1 [29]	170	400	16	Mandarin	Apache license 2.0
AISHELL-2 [30]	1000	1991	44.1	Mandarin	Apache license 2.0
AISHELL-3 [31]	85	218	44.1	Mandarin	Apache license 2.0
DiDiSpeech-1 [32]	572	4500	48	Mandarin	Apache license 2.0
DiDiSpeech-2 [32]	227	1500	48	Mandarin	Apache license 2.0
JSUT [33]	10	1	48	Japanese	CC-BY-SA 4.0
JVS corpus [34]	30	100	24	Japanese	CC BY-SA 4.0
Korean single speaker [35]	12	1	44.1	Korean	CC BY-NC-SA 4.0
KazakhTTS [36]	93	2	44.1/48	Kazakh	CC BY 4.0
Ruslan [37]	31	1	44.1	Russian	CC BY-NC-SA 4.0
HUI-audio-corpus [38]	326	122	44.1	German	CC0
SIWIS [39]	10	1	44.1	French	CC BY 4.0
India corpus [40]	39	253	48	Multilingual	CC BY-SA 4.0
M-AILABS [41]	1000	/	16	Multilingual	BSD
MLS [42]	51K	6K	16	Multilingual	CC BY 4.0
CSS10 [43]	140	1	22.05	Multilingual	CC0 1.0
CommonVoice [44]	2.5K	50K	48	Multilingual	CC0 1.0

Appendix B
TTS Model List

We list some representative TTS models in Table B.1.

X. Tan, *Neural Text-to-Speech Synthesis*, Artificial Intelligence: Foundations, Theory, and Algorithms, https://doi.org/10.1007/978-981-99-0827-1

Table B.1 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, "Ø" represents no conditional information, "IS" represents INTERSPEECH

Model	AM/Voc	Data flow	Publication	Time
WaveNet [45]	Voc	ling $\xrightarrow{AR}$ wav	arXiv16	2016.09
SampleRNN [46]	Voc	Ø $\xrightarrow{AR}$ wav	ICLR17	2016.12
Deep voice [47]	AM+Voc	ch→ ph→ling $\xrightarrow{AR}$ wav	ICML17	2017.02
Char2Wav [48]	E2E	ch $\xrightarrow{AR}$ ceps $\xrightarrow{AR}$ wav	ICLR17 WS	2017.02
Tacotron [49]	AM	ch/ph $\xrightarrow{AR}$ linS ⟶ wav	IS17	2017.03
Deep voice 2 [50]	AM+Voc	ch→ph $\xrightarrow{FF}$ ling $\xrightarrow{AR}$ wav	NIPS17	2017.05
DV2-Tacotron [50]	AM+Voc	ch $\xrightarrow{AR}$ linS $\xrightarrow{AR}$ wav	NIPS17	2017.05
VoiceLoop [51]	AM	ph→ceps ⟶ wav	ICLR18	2017.07
Deep voice 3 [52]	AM	ch/ph $\xrightarrow{AR}$ melS $\xrightarrow{AR}$ wav	ICLR18	2017.10
DCTTS [53]	AM	ch $\xrightarrow{AR}$ melS ⟶ wav	ICASSP18	2017.10
Par.WaveNet [54]	Voc	ling $\xrightarrow{FF}$ wav	ICML18	2017.11
Tacotron 2 [55]	AM	ch/ph $\xrightarrow{AR}$ melS $\xrightarrow{AR}$ wav	ICASSP18	2017.12
WaveGAN [56]	Voc	Ø $\xrightarrow{FF}$ wav	ICLR19	2018.02
WaveRNN [57]	Voc	ling $\xrightarrow{AR}$ wav	ICML18	2018.02
DV3-Clone [58]	AM	ch/ph $\xrightarrow{AR}$ linS→wav	NeurIPS18	2018.02
GST-Tacotron [59]	AM	ph $\xrightarrow{AR}$ melS→wav	ICML18	2018.03
Ref-Tacotron [60]	AM	ph $\xrightarrow{AR}$ melS→wav	ICML18	2018.03
FFTNet [61]	Voc	ceps $\xrightarrow{AR}$ wav	ICASSP18	2018.04
VAE-Loop [62]	AM	ph→ceps ⟶ wav	IS18	2018.04
SV-Tacotron [63]	AM	ch/ph $\xrightarrow{AR}$ melS $\xrightarrow{AR}$ wav	NeurIPS18	2018.06
ClariNet [64]	E2E	ch/ph $\xrightarrow{AR}$ wav	ICLR19	2018.07
ForwardAtt [65]	AM	ph $\xrightarrow{AR}$ linS→wav	ICASSP18	2018.07
MCNN [66]	Voc	linS $\xrightarrow{FF}$ wav	SPL18	2018.08
TransformerTTS [67]	AM	ph $\xrightarrow{AR}$ melS $\xrightarrow{AR}$ wav	AAAI19	2018.09
SEA-TTS [68]	Voc	ling $\xrightarrow{AR}$ wav	ICLR19	2018.09
GMVAE-Tacotron [69]	AM	ph $\xrightarrow{AR}$ melS $\xrightarrow{AR}$ wav	ICLR19	2018.10
LPCNet [70]	Voc	ceps $\xrightarrow{AR}$ wav	ICASSP19	2018.10
WaveGlow [71]	Voc	melS $\xrightarrow{FF}$ wav	ICASSP19	2018.10
FloWaveNet [72]	Voc	melS $\xrightarrow{FF}$ wav	ICML19	2018.11
Univ. WaveRNN [73]	Voc	melS $\xrightarrow{AR}$ wav	IS19	2018.11
VAE-TTS [74]	AM	ph $\xrightarrow{AR}$ melS $\xrightarrow{AR}$ wav	ICASSP19	2018.12
TTS-Stylization [75]	AM	ch $\xrightarrow{AR}$ melS→wav	ICLR19	2018.12

(continued)

Table B.1 (continued)

AdVoc [76]	Voc	melS $\xrightarrow{\text{FF}}$ linS $\rightarrow$ wav	IS19	2019.04
GAN exposure [77]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{AR}}$ wav	IS19	2019.04
GELP [78]	Voc	melS $\xrightarrow{\text{FF}}$ wav	IS19	2019.04
Almost unsup [79]	AM	ph $\xrightarrow{\text{AR}}$ melS $\rightarrow$ wav	ICML19	2019.05
FastSpeech [80]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	NeurIPS19	2019.05
ParaNet [81]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	ICML20	2019.05
WaveVAE [81]	Voc	melS $\xrightarrow{\text{FF}}$ wav	ICML20	2019.05
MelNet [82]	AM	ch $\xrightarrow{\text{AR}}$ melS $\longrightarrow$ wav	arXiv19	2019.06
StepwiseMA [83]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{AR}$ wav	IS19	2019.06
GAN-TTS [84]	Voc	ling $\xrightarrow{\text{FF}}$ wav	ICLR20	2019.09
DurIAN [85]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{AR}}$ wav	IS20	2019.09
MB WaveRNN [85]	Voc	melS $\xrightarrow{\text{AR}}$ wav	IS20	2019.09
MelGAN [86]	Voc	melS $\xrightarrow{\text{FF}}$ wav	NeurIPS19	2019.10
Para. WaveGAN [87]	Voc	melS $\xrightarrow{\text{FF}}$ wav	ICASSP20	2019.10
DCA-Tacotron [88]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{AR}}$ wav	ICASSP20	2019.10
WaveFlow [89]	Voc	melS $\xrightarrow{\text{AR}}$ wav	ICML20	2019.12
SqueezeWave [90]	Voc	melS $\xrightarrow{\text{FF}}$ wav	arXiv20	2020.01
AlignTTS [91]	AM	ch/ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	ICASSP20	2020.03
RobuTrans [92]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{AR}}$ wav	AAAI20	2020.04
Flow-TTS [93]	AM	ch/ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	ICASSP20	2020.05
Flowtron [94]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{FF}}$ wav	ICLR21	2020.05
Glow-TTS [95]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	NeurIPS20	2020.05
JDI-T [96]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	IS20	2020.05
TalkNet [97]	AM	ch $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	IS21	2020.05
MB MelGAN [98]	Voc	melS $\xrightarrow{\text{FF}}$ wav	SLT21	2020.05
MultiSpeech [99]	AM	ph $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{FF}}$ wav	IS20	2020.06
FastSpeech 2 [100]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	ICLR21	2020.06
FastSpeech 2s [100]	E2E	ph $\xrightarrow{\text{FF}}$ wav	ICLR21	2020.06
EATS [101]	E2E	ch/ph $\xrightarrow{\text{FF}}$ wav	ICLR21	2020.06
FastPitch [102]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	ICASSP21	2020.06
VocGAN [103]	Voc	melS $\xrightarrow{\text{FF}}$ wav	IS20	2020.07
LRSpeech [104]	AM	ch $\xrightarrow{\text{AR}}$ melS $\xrightarrow{\text{FF}}$ wav	KDD20	2020.08
SpeedySpeech [105]	AM	ph $\xrightarrow{\text{FF}}$ melS $\xrightarrow{\text{FF}}$ wav	IS20	2020.08
GED [106]	Voc	ling $\xrightarrow{\text{FF}}$ wav	NeurIPS20	2020.08
SC-WaveRNN [107]	Voc	melS $\xrightarrow{\text{AR}}$ wav	IS20	2020.08
WaveGrad [108]	Voc	melS $\xrightarrow{\text{FF}}$ wav	ICLR21	2020.09

(continued)

Table B.1 (continued)

DiffWave [109]	Voc	$\text{melS} \xrightarrow{\text{FF}} \text{wav}$	ICLR21	2020.09
HiFi-GAN [110]	Voc	$\text{melS} \xrightarrow{\text{FF}} \text{wav}$	NeurIPS20	2020.10
NonAtt tacotron [111]	AM	$\text{ph} \xrightarrow{\text{AR}} \text{melS} \xrightarrow{\text{AR}} \text{wav}$	arXiv20	2020.10
Para. tacotron [112]	AM	$\text{ph} \xrightarrow{\text{FF}} \text{melS} \xrightarrow{\text{AR}} \text{wav}$	ICASSP21	2020.10
DeviceTTS [113]	AM	$\text{ph} \xrightarrow{\text{AR}} \text{Ceps} \rightarrow \text{wav}$	arXiv20	2020.10
Wave-Tacotron [114]	E2E	$\text{ch/ph} \xrightarrow{\text{AR}} \text{wav}$	ICASSP21	2020.11
DenoiSpeech [115]	AM	$\text{ph} \xrightarrow{\text{FF}} \text{melS} \xrightarrow{\text{FF}} \text{wav}$	ICASSP21	2020.12
EfficientTTS [116]	AM	$\text{ch} \xrightarrow{\text{FF}} \text{melS} \xrightarrow{\text{FF}} \text{wav}$	ICML21	2020.12
EfficientTTS-Wav [116]	E2E	$\text{ch} \xrightarrow{\text{FF}} \text{wav}$	ICML21	2020.12
Multi-SpectroGAN [117]	AM	$\text{ph} \xrightarrow{\text{AR}} \text{melS} \xrightarrow{\text{FF}} \text{wav}$	AAAI21	2020.12
LightSpeech [118]	AM	$\text{ph} \xrightarrow{\text{FF}} \text{melS} \xrightarrow{\text{FF}} \text{wav}$	ICASSP21	2021.02
Para. Tacotron 2 [119]	AM	$\text{ph} \xrightarrow{\text{FF}} \text{melS} \xrightarrow{\text{AR}} \text{wav}$	IS21	2021.03
AdaSpeech [120]	AM	$\text{ph} \xrightarrow{\text{FF}} \text{melS} \xrightarrow{\text{FF}} \text{wav}$	ICLR21	2021.03
BVAE-TTS [121]	AM	$\text{ph} \xrightarrow{\text{FF}} \text{melS} \xrightarrow{\text{FF}} \text{wav}$	ICLR21	2021.03
PnG BERT [122]	AM	$\text{ph} \xrightarrow{\text{AR}} \text{melS} \xrightarrow{\text{AR}} \text{wav}$	IS21	2021.03
Fast DCTTS [123]	AM	$\text{ch} \xrightarrow{\text{AR}} \text{melS} \xrightarrow{\text{FF}} \text{wav}$	ICASSP21	2021.04
AdaSpeech 2 [124]	AM	$\text{ph} \xrightarrow{\text{FF}} \text{melS} \xrightarrow{\text{FF}} \text{wav}$	ICASSP21	2021.04
TalkNet 2 [125]	AM	$\text{ch} \xrightarrow{\text{FF}} \text{melS} \xrightarrow{\text{FF}} \text{wav}$	arXiv21	2021.04
Triple M [126]	AM+Voc	$\text{ch} \xrightarrow{\text{AR}} \text{melS} \xrightarrow{\text{AR}} \text{wav}$	IS21	2021.04
Diff-TTS [127]	AM	$\text{ph} \xrightarrow{\text{FF}} \text{melS} \xrightarrow{\text{FF}} \text{wav}$	IS21	2021.04
Grad-TTS [128]	AM	$\text{ph} \xrightarrow{\text{FF}} \text{melS} \xrightarrow{\text{FF}} \text{wav}$	ICML21	2021.05
Fre-GAN [129]	Voc	$\text{melS} \xrightarrow{\text{FF}} \text{wav}$	IS21	2021.06
VITS [130]	E2E	$\text{ph} \xrightarrow{\text{FF}} \text{wav}$	ICML21	2021.06
AdaSpeech 3 [131]	AM	$\text{ph} \xrightarrow{\text{FF}} \text{melS} \xrightarrow{\text{FF}} \text{wav}$	IS21	2021.06
PriorGrad-AM [132]	AM	$\text{ph} \xrightarrow{\text{FF}} \text{melS} \xrightarrow{\text{FF}} \text{wav}$	ICLR22	2021.06
PriorGrad-Voc [132]	Voc	$\text{melS} \xrightarrow{\text{FF}} \text{wav}$	ICLR22	2021.06
Meta-StyleSpeech [133]	AM	$\text{ph} \xrightarrow{\text{FF}} \text{melS} \xrightarrow{\text{FF}} \text{wav}$	ICML21	2021.06
WaveGrad 2 [134]	E2E	$\text{ph} \xrightarrow{\text{FF}} \text{wav}$	IS21	2021.06
InferGrad [135]	Voc	$\text{melS} \xrightarrow{\text{FF}} \text{wav}$	ICASSP22	2022.02
SpecGrad [136]	Voc	$\text{melS} \xrightarrow{\text{FF}} \text{wav}$	IS22	2022.03
NaturalSpeech [137]	E2E	$\text{ph} \xrightarrow{\text{FF}} \text{wav}$	arXiv22	2022.05

References

1. Tan X, Qin T, Soong F, Liu TY (2021) A survey on neural speech synthesis. Preprint. arXiv:2106.15561
2. Tan X, Lee Hy (2022) TTS tutorial at INTERSPEECH 2022. https://www.interspeech2022.org/program/tutorials.php

3. Tan X, Qin T (2022) TTS tutorial at ICASSP 2022. https://2022.ieeeicassp.org/tutorials.php
4. Tan X, Qin T (2021) TTS tutorial at IJCAI 2021. https://ijcai-21.org/tutorials/
5. Tan X (2021) TTS tutorial at ISCSLP 2021. https://www.microsoft.com/en-us/research/uploads/prod/2021/02/ISCSLP2021-TTS-Tutorial.pdf
6. Ling ZH (2016) Deep learning for statistical parametric speech synthesis. http://staff.ustc.edu.cn/~zhling/download/ISCSLP16_tutorial_DLSPSS.pdf
7. Qian Y, Soong FK (2014) TTS tutorial at ISCSLP 2014. https://www.superlectures.com/iscslp2014/tutorial-4-deep-learning-for-speech-generation-and-synthesis
8. Wang X, Yasuda Y (2019) TTS tutorial at IEICE SP workshop. https://www.slideshare.net/jyamagis/tutorial-on-endtoend-texttospeech-synthesis-part-1-neural-waveform-modeling
9. Bengio Y (2017) Deep generative models for speech and images. https://www.youtube.com/watch?v=vEAq_sBf1CA
10. Zen H (2017) Generative model-based text-to-speech synthesis. https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45882.pdf
11. Tokuda K (2019) Statistical approach to speech synthesis: past, present and future. In: INTERSPEECH
12. Tan X (2021) Microsoft research webinar: pushing the frontier of neural text to speech. https://www.youtube.com/watch?v=MA8PCvmr8B0
13. Hayashi T, Yamamoto R, Inoue K, Yoshimura T, Watanabe S, Toda T, Takeda K, Zhang Y, Tan X (2020) 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). IEEE, pp 7654–7658
14. Kominek J (2003) CMU ARCTIC databases for speech synthesis. CMU-LTI, Carnegie Mellon University
15. Veaux C, Yamagishi J, MacDonald K et al (2016) Superseded-CSTK VCTK corpus: English multi-speaker corpus for CSTK voice cloning toolkit
16. King S, Karaiskos V (2011) The Blizzard challenge 2011. In: Blizzard challenge workshop
17. King S, Karaiskos V (2013) The Blizzard challenge 2013. In: Blizzard challenge workshop
18. Ito K (2017) The LJ speech dataset. https://keithito.com/LJ-Speech-Dataset/
19. Panayotov V, Chen G, Povey D, Khudanpur S (2015) LibriSpeech: an ASR corpus based on public domain audio books. In: 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 5206–5210
20. Zen H, Dang V, Clark R, Zhang Y, Weiss RJ, Jia Y, Chen Z, Wu Y (2019) LibriTTS: a corpus derived from librispeech for text-to-speech. In: Proceedings of the Interspeech 2019, pp 1526–1530
21. Toda T, Chen LH, Saito D, Villavicencio F, Wester M, Wu Z, Yamagishi J (2016) The voice conversion challenge 2016. In: Interspeech, pp 1632–1636
22. Lorenzo-Trueba J, Yamagishi J, Toda T, Saito D, Villavicencio F, Kinnunen T, Ling Z (2018) The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods. In: Proceedings of the Odyssey 2018 the speaker and language recognition workshop, pp 195–202
23. Bakhturina E, Lavrukhin V, Ginsburg B, Zhang Y (2021) Hi-Fi multi-speaker English TTS dataset. Preprint. arXiv:2104.01497
24. Rousseau A, Deléglise P, Esteve Y (2012) TED-LIUM: an automatic speech recognition dedicated corpus. In: The international conference on language resources and evaluation, pp 125–129
25. Canavan A, David G, George Z (2021) CALLHOME american English speech. https://catalog.ldc.upenn.edu/LDC97S42
26. Zandie R, Mahoor MH, Madse J, Emamian ES (2021) RyanSpeech: a corpus for conversational text-to-speech synthesis. Preprint. arXiv:2106.08468
27. Baker D (2017) Chinese standard Mandarin speech corpus. https://www.data-baker.com/open_source.html
28. Liu Y, Fung P, Yang Y, Cieri C, Huang S, Graff D (2006) HKUST/MTS: a very large scale Mandarin telephone speech corpus. In: International symposium on Chinese spoken language processing. Springer, pp 724–735

29. Bu H, Du J, Na X, Wu B, Zheng H (2017) 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). IEEE, pp 1–5
30. Du J, Na X, Liu X, Bu H (2018) AISHELL-2: transforming Mandarin ASR research into industrial scale. Preprint. arXiv:1808.10583
31. Shi Y, Bu H, Xu X, Zhang S, Li M (2020) AISHELL-3: a multi-speaker Mandarin TTS corpus and the baselines. Preprint. arXiv:2010.11567
32. Guo T, Wen C, Jiang D, Luo N, Zhang R, Zhao S, Li W, Gong C, Zou W, Han K, et al. (2020) DiDiSpeech: a large scale Mandarin speech corpus. Preprint. arXiv:2010.09275
33. Sonobe R, Takamichi S, Saruwatari H (2017) JSUT corpus: free large-scale Japanese speech corpus for end-to-end speech synthesis. Preprint. arXiv:1711.00354
34. Takamichi S, Mitsui K, Saito Y, Koriyama T, Tanji N, Saruwatari H (2019) JVS corpus: free Japanese multi-speaker voice corpus. Preprint. arXiv:1908.06248
35. Park K (2018) KSS dataset: Korean single speaker speech dataset. https://kaggle.com/bryanpark/korean-single-speaker-speech-dataset
36. Mussakhojayeva S, Janaliyeva A, Mirzakhmetov A, Khassanov Y, Varol HA (2021) KazakhTTS: an open-source kazakh text-to-speech synthesis dataset. Preprint. arXiv:2104.08459
37. Gabdrakhmanov L, Garaev R, Razinkov E (2019) Ruslan: Russian spoken language corpus for speech synthesis. In: International conference on speech and computer. Springer, pp 113–121
38. Puchtler P, Wirth J, Peinl R (2021) HUI-Audio-Corpus-German: a high quality TTS dataset. Preprint. arXiv:2106.06309
39. Yamagishi J, Honnet PE, Garner P, Lazaridis A et al (2017) The SIWIS French speech synthesis database
40. He F, Chu SHC, Kjartansson O, Rivera C, Katanova A, Gutkin A, Demirsahin I, Johny C, Jansche M, Sarin S et al (2020) 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, pp 6494–6503
41. GmbH MAIL (2019) The M-AILABS speech dataset. https://www.caito.de/2019/01/the-m-ailabs-speech-dataset/
42. Pratap V, Xu Q, Sriram A, Synnaeve G, Collobert R (2020) MLS: a large-scale multilingual dataset for speech research. In: Proceedings of the Interspeech 2020, pp 2757–2761
43. Park K, Mulc T (2019) CSS10: a collection of single speaker speech datasets for 10 languages. In: Proceedings of the Interspeech 2019, pp 1566–1570
44. Ardila R, Branson M, Davis K, Kohler M, Meyer J, Henretty M, Morais R, Saunders L, Tyers F, Weber G (2020) Common voice: a massively-multilingual speech corpus. In: Proceedings of The 12th language resources and evaluation conference, pp 4218–4222
45. van den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K (2016) WaveNet: a generative model for raw audio. Preprint. arXiv:1609.03499
46. Mehri S, Kumar K, Gulrajani I, Kumar R, Jain S, Sotelo J, Courville A, Bengio Y (2017) SampleRNN: an unconditional end-to-end neural audio generation model. In: International conference on learning representations
47. Arık SÖ, Chrzanowski M, Coates A, Diamos G, Gibiansky A, Kang Y, Li X, Miller J, Ng A, Raiman J et al (2017) Deep voice: real-time neural text-to-speech. In: International conference on machine learning (PMLR), pp 195–204
48. Sotelo J, Mehri S, Kumar K, Santos JF, Kastner K, Courville AC, Bengio Y (2017) Char2wav: end-to-end speech synthesis. In: 5th international conference on learning representations, ICLR 2017, Toulon, France, April 24–26, 2017, Workshop Track Proceedings. OpenReview.net. https://openreview.net/forum?id=B1VWyySKx
49. Wang Y, Skerry-Ryan R, Stanton D, Wu Y, Weiss RJ, Jaitly N, Yang Z, Xiao Y, Chen Z, Bengio S et al (2017) Tacotron: towards end-to-end speech synthesis. In: Proceedings of the Interspeech 2017, pp 4006–4010

50. Gibiansky A, Arik SÖ, Diamos GF, Miller J, Peng K, Ping W, Raiman J, Zhou Y (2017) Deep voice 2: multi-speaker neural text-to-speech. In: NIPS
51. Taigman Y, Wolf L, Polyak A, Nachmani E (2018) VoiceLoop: voice fitting and synthesis via a phonological loop. In: International conference on learning representations
52. Ping W, Peng K, Gibiansky A, Arik SO, Kannan A, Narang S, Raiman J, Miller J (2018) Deep Voice 3: 2000-speaker neural text-to-speech. In: Proceedings of the International conference on learning representations, pp ,214–217
53. Tachibana H, Uenoyama K, Aihara S (2018) 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). IEEE, pp 4784–4788
54. Oord Avd, Li Y, Babuschkin I, Simonyan K, Vinyals O, Kavukcuoglu K, Driessche G, Lockhart E, Cobo L, Stimberg F et al (2018) Parallel WaveNet: fast high-fidelity speech synthesis. In: International conference on machine learning (PMLR), pp 3918–3926
55. Shen J, Pang R, Weiss RJ, Schuster M, Jaitly N, Yang Z, Chen Z, Zhang Y, Wang Y, Skerry-Ryan R et al (2018) Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4779–4783
56. Donahue C, McAuley J, Puckette M (2018) Adversarial audio synthesis. In: International conference on learning representations
57. Kalchbrenner N, Elsen E, Simonyan K, Noury S, Casagrande N, Lockhart E, Stimberg F, Oord A, Dieleman S, Kavukcuoglu K (2018) Efficient neural audio synthesis. In: International conference on machine learning (PMLR), pp 2410–2419
58. Arık SÖ, Chen J, Peng K, Ping W, Zhou Y (2018) Neural voice cloning with a few samples. In: Proceedings of the 32nd international conference on neural information processing systems, pp 10040–10050
59. Wang Y, Stanton D, Zhang Y, Skerry-Ryan R, Battenberg E, Shor J, Xiao Y, Jia Y, Ren F, Saurous RA (2018) Style tokens: unsupervised style modeling, control and transfer in end-to-end speech synthesis. In: International conference on machine learning (PMLR), pp 5180–5189
60. Skerry-Ryan R, Battenberg E, Xiao Y, Wang Y, Stanton D, Shor J, Weiss R, Clark R, Saurous RA (2018) Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron. In: International conference on machine learning (PMLR), pp 4693–4702
61. Jin Z, Finkelstein A, Mysore GJ, Lu J (2018) FFTNet: a real-time speaker-dependent neural vocoder. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 2251–2255
62. Akuzawa K, Iwasawa Y, Matsuo Y (2018) Expressive speech synthesis via modeling expressions with variational autoencoder. In: Proceedings of The Interspeech 2018, pp 3067–3071
63. Jia Y, Zhang Y, Weiss RJ, Wang Q, Shen J, Ren F, Chen Z, Nguyen P, Pang R, Moreno IL et al (2018) Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In: Proceedings of the 32nd international conference on neural information processing systems, pp 4485–4495
64. Ping W, Peng K, Chen J (2018) ClariNet: parallel wave generation in end-to-end text-to-speech. In: International conference on learning representations
65. Zhang JX, Ling ZH, Dai LR (2018) Forward attention in sequence-to-sequence acoustic modeling for speech synthesis. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 4789–4793
66. Arık SÖ, Jun H, Diamos G (2018) Fast spectrogram inversion using multi-head convolutional neural networks. IEEE Signal Process Lett 26(1):94–98
67. Li N, Liu S, Liu Y, Zhao S, Liu M (2019) Neural speech synthesis with transformer network. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 6706–6713
68. Chen Y, Assael Y, Shillingford B, Budden D, Reed S, Zen H, Wang Q, Cobo LC, Trask A, Laurie B et al (2018) Sample efficient adaptive text-to-speech. In: International conference on learning representations

69. Hsu WN, Zhang Y, Weiss RJ, Zen H, Wu Y, Wang Y, Cao Y, Jia Y, Chen Z, Shen J et al (2018) Hierarchical generative modeling for controllable speech synthesis. In: International conference on learning representations
70. Valin JM, Skoglund J (2019) LPCNet: improving neural speech synthesis through linear prediction. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 5891–5895
71. Prenger R, Valle R, Catanzaro B (2019) WaveGlow: a flow-based generative network for speech synthesis. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 3617–3621
72. Kim S, Lee SG, Song J, Kim J, Yoon S (2019) FloWaveNet: a generative flow for raw audio. In: International conference on machine learning (PMLR), pp 3370–3378
73. Lorenzo-Trueba J, Drugman T, Latorre J, Merritt T, Putrycz B, Barra-Chicote R, Moinet A, Aggarwal V (2019) Towards achieving robust universal neural vocoding. In: Proceedings of the Interspeech 2019, pp 181–185
74. Zhang YJ, Pan S, He L, Ling ZH (2019) 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). IEEE, pp 6945–6949
75. Ma S, Mcduff D, Song Y (2018) Neural TTS stylization with adversarial and collaborative games. In: International conference on learning representations
76. Neekhara P, Donahue C, Puckette M, Dubnov S, McAuley J (2019) Expediting TTS synthesis with adversarial vocoding. In: Proceedings of the Interspeech 2019, pp 186–190
77. Guo H, Soong FK, He L, Xie L (2019) A new GAN-based end-to-end TTS training algorithm. In: Proceedings of the Interspeech 2019, pp 1288–1292
78. Juvela L, Bollepalli B, Yamagishi J, Alku P (2019) GELP: GAN-excited linear prediction for speech synthesis from mel-spectrogram. In: Proceedings of the Interspeech 2019, pp 694–698
79. Ren Y, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2019) Almost unsupervised text to speech and automatic speech recognition. In: International conference on machine learning (PMLR), pp 5410–5419
80. Ren Y, Ruan Y, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2019) FastSpeech: fast, robust and controllable text to speech. In: NeurIPS
81. Peng K, Ping W, Song Z, Zhao K (2020) Non-autoregressive neural text-to-speech. In: International conference on machine learning (PMLR), pp 7586–7598
82. Vasquez S, Lewis M (2019) MelNet: a generative model for audio in the frequency domain. Preprint. arXiv:1906.01083
83. He M, Deng Y, He L (2019) Robust sequence-to-sequence acoustic modeling with stepwise monotonic attention for neural TTS. In: Proceedings of the Interspeech 2019, pp 1293–1297
84. Bińkowski M, Donahue J, Dieleman S, Clark A, Elsen E, Casagrande N, Cobo LC, Simonyan K (2019) High fidelity speech synthesis with adversarial networks. In: International conference on learning representations
85. Yu C, Lu H, Hu N, Yu M, Weng C, Xu K, Liu P, Tuo D, Kang S, Lei G et al (2020) DurIAN: Duration informed attention network for speech synthesis. In: Proceedings of the Interspeech 2020 pp 2027–2031
86. Kumar K, Kumar R, de Boissiere T, Gestin L, Teoh WZ, Sotelo J, de Brébisson A, Bengio Y, Courville A (2019) MelGAN: generative adversarial networks for conditional waveform synthesis. In: NeurIPS
87. Yamamoto R, Song E, Kim JM (2020) 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). IEEE, pp 6199–6203
88. Battenberg E, Skerry-Ryan R, Mariooryad S, Stanton D, Kao D, Shannon M, Bagby T (2020) Location-relative attention mechanisms for robust long-form speech synthesis. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 6194–6198

89. Ping W, Peng K, Zhao K, Song Z (2020) WaveFlow: a compact flow-based model for raw audio. In: International conference on machine learning (PMLR), pp 7706–7716
90. Zhai B, Gao T, Xue F, Rothchild D, Wu B, Gonzalez JE, Keutzer K (2020) SqueezeWave: extremely lightweight vocoders for on-device speech synthesis. Preprint. arXiv:2001.05685
91. Zeng Z, Wang J, Cheng N, Xia T, Xiao J (2020) 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). IEEE, pp 6714–6718
92. Li N, Liu Y, Wu Y, Liu S, Zhao S, Liu M (2020) RobuTrans: a robust Transformer-based text-to-speech model. In: Proceedings of the AAAI conference on artificial intelligence, vol 34, pp 8228–8235
93. Miao C, Liang S, Chen M, Ma J, Wang S, Xiao J (2020) 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). IEEE, pp 7209–7213
94. Valle R, Shih K, Prenger R, Catanzaro B (2020) Flowtron: an autoregressive flow-based generative network for text-to-speech synthesis. Preprint. arXiv:2005.05957
95. Kim J, Kim S, Kong J, Yoon S (2020) Glow-TTS: a generative flow for text-to-speech via monotonic alignment search. Adv Neural Inf Process Syst 33
96. Lim D, Jang W, Gyeonghwan O, Park H, Kim B, Yoon J (2020) JDI-T: jointly trained duration informed transformer for text-to-speech without explicit alignment. In: Proceedings of the Interspeech 2020, pp 4004–4008
97. Beliaev S, Rebryk Y, Ginsburg B (2020) TalkNet: fully-convolutional non-autoregressive speech synthesis model. Preprint. arXiv:2005.05514
98. Yang G, Yang S, Liu K, Fang P, Chen W, Xie L (2020) Multi-band MelGAN: faster waveform generation for high-quality text-to-speech. Preprint. arXiv:2005.05106
99. Chen M, Tan X, Ren Y, Xu J, Sun H, Zhao S, Qin T (2020) MultiSpeech: multi-speaker text to speech with transformer. In: INTERSPEECH, pp 4024–4028
100. Ren Y, Hu C, Tan X, Qin T, Zhao S, Zhao Z, Liu TY (2021) FastSpeech 2:fast and high-quality end-to-end text to speech. In: International conference on learning representations. https://openreview.net/forum?id=piLPYqxtWuA
101. Donahue J, Dieleman S, Bińkowski M, Elsen E, Simonyan K (2021) End-to-end adversarial text-to-speech. In: International conference on learning representations
102. Łańcucki A (2020) FastPitch: parallel text-to-speech with pitch prediction. Preprint. arXiv:2006.06873
103. Yang J, Lee J, Kim Y, Cho HY, Kim I (2020) VocGAN: a high-fidelity real-time vocoder with a hierarchically-nested adversarial network. In: Proceedings of the Interspeech 2020 pp 200–204
104. Xu J, Tan X, Ren Y, Qin T, Li J, Zhao S, Liu TY (2020) LRSpeech: extremely low-resource speech synthesis and recognition. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery and data mining, pp 2802–2812
105. Vainer J, Dušek O (2020) SpeedySpeech: efficient neural speech synthesis. In: Proceedings of the Interspeech 2020, pp 3575–3579
106. Gritsenko A, Salimans T, van den Berg R, Snoek J, Kalchbrenner N (2020) A spectral energy distance for parallel speech synthesis. Adv Neural Inf Process Syst 33
107. Paul D, Pantazis Y, Stylianou Y (2020) Speaker conditional WaveRNN: towards universal neural vocoder for unseen speaker and recording conditions. In: Proceedings of the Interspeech 2020, pp 235–239
108. Chen N, Zhang Y, Zen H, Weiss RJ, Norouzi M, Chan W (2021) WaveGrad: estimating gradients for waveform generation. In: International conference on learning representations
109. Kong Z, Ping W, Huang J, Zhao K, Catanzaro B (2021) DiffWave: a versatile diffusion model for audio synthesis. In: International conference on learning representations
110. Kong J, Kim J, Bae J (2020) HiFi-GAN: generative adversarial networks for efficient and high fidelity speech synthesis. Adv Neural Inf Process Syst 33
111. Shen J, Jia Y, Chrzanowski M, Zhang Y, Elias I, Zen H, Wu Y (2020) Non-attentive Tacotron: robust and controllable neural TTS synthesis including unsupervised duration modeling. Preprint. arXiv:2010.04301

112. Elias I, Zen H, Shen J, Zhang Y, Jia Y, Weiss R, Wu Y (2020) Parallel Tacotron: non-autoregressive and controllable TTS. Preprint. arXiv:2010.11439
113. Huang Z, Li H, Lei M (2020) DeviceTTS: a small-footprint, fast, stable network for on-device text-to-speech. Preprint. arXiv:2010.15311
114. Weiss RJ, Skerry-Ryan R, Battenberg E, Mariooryad S, Kingma DP (2021) Wave-Tacotron: spectrogram-free end-to-end text-to-speech synthesis. In: 2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
115. Zhang C, Ren Y, Tan X, Liu J, Zhang K, Qin T, Zhao S, Liu TY (2021) DenoiSpeech: denoising text to speech with frame-level noise modeling. In: 2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
116. Miao C, Liang S, Liu Z, Chen M, Ma J, Wang S, Xiao J (2020) EfficientTTS: An efficient and high-quality text-to-speech architecture. Preprint. arXiv:2012.03500
117. Lee SH, Yoon HW, Noh HR, Kim JH, Lee SW (2020) Multi-SpectroGAN: high-diversity and high-fidelity spectrogram generation with adversarial style combination for speech synthesis. Preprint. arXiv:2012.07267
118. Luo R, Tan X, Wang R, Qin T, Li J, Zhao S, Chen E, Liu TY (2021) LightSpeech: lightweight and fast text to speech with neural architecture search. In: 2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
119. Elias I, Zen H, Shen J, Zhang Y, Ye J, Skerry-Ryan R, Wu Y (2021) Parallel Tacotron 2: a non-autoregressive neural TTS model with differentiable duration modeling. Preprint. arXiv:2103.14574
120. Chen M, Tan X, Li B, Liu Y, Qin T, Zhao S, Liu TY (2021) AdaSpeech: Adaptive text to speech for custom voice. In: International conference on learning representations. https://openreview.net/forum?id=Drynvt7gg4L
121. Lee Y, Shin J, Jung K (2020) Bidirectional variational inference for non-autoregressive text-to-speech. In: International conference on learning representations
122. Jia Y, Zen H, Shen J, Zhang Y, Wu Y (2021) PnG BERT: augmented BERT on phonemes and graphemes for neural TTS. Preprint. arXiv:2103.15060
123. Kang M, Lee J, Kim S, Kim I (2021) Fast DCTTS: efficient deep convolutional text-to-speech. Preprint. arXiv:2104.00624
124. Yan Y, Tan X, Li B, Qin T, Zhao S, Shen Y, Liu TY (2021) AdaSpeech 2: adaptive text to speech with untranscribed data. In: 2021 IEEE International conference on acoustics, speech and signal processing (ICASSP). IEEE
125. Beliaev S, Ginsburg B (2021) TalkNet 2: Non-autoregressive depth-wise separable convolutional model for speech synthesis with explicit pitch and duration prediction. Preprint. arXiv:2104.08189
126. Lin S, Xie F, Meng L, Li X, Lu L (2021) Triple M: a practical text-to-speech synthesis system with multi-guidance attention and multi-band multi-time LPCNet. Preprint. arXiv:2102.00247
127. Jeong M, Kim H, Cheon SJ, Choi BJ, Kim NS (2021) Diff-TTS: a denoising diffusion model for text-to-speech. Preprint. arXiv:2104.01409
128. Popov V, Vovk I, Gogoryan V, Sadekova T, Kudinov M (2021) Grad-TTS: a diffusion probabilistic model for text-to-speech. Preprint. arXiv:2105.06337
129. Kim JH, Lee SH, Lee JH, Lee SW (2021) Fre-GAN: adversarial frequency-consistent audio synthesis. Preprint. arXiv:2106.02297
130. Kim J, Kong J, Son J (2021) Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. Preprint. arXiv:2106.06103
131. Yan Y, Tan X, Li B, Zhang G, Qin T, Zhao S, Shen Y, Zhang WQ, Liu TY (2021) AdaSpeech 3: adaptive text to speech for spontaneous style. In: INTERSPEECH
132. Lee Sg, Kim H, Shin C, Tan X, Liu C, Meng Q, Qin T, Chen W, Yoon S, Liu TY (2021) PriorGrad: improving conditional denoising diffusion models with data-driven adaptive prior. Preprint. arXiv:2106.06406
133. Min D, Lee DB, Yang E, Hwang SJ (2021) Meta-StyleSpeech: multi-speaker adaptive text-to-speech generation. Preprint. arXiv:2106.03153

134. Chen N, Zhang Y, Zen H, Weiss RJ, Norouzi M, Dehak N, Chan W (2021) WaveGrad 2: iterative refinement for text-to-speech synthesis. Preprint. arXiv:2106.09660
135. Chen Z, Tan X, Wang K, Pan S, Mandic D, He L, Zhao S (2022) InferGrad: improving diffusion models for vocoder by considering inference in training. In: ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 8432–8436
136. Koizumi Y, Zen H, Yatabe K, Chen N, Bacchiani M (2022) SpecGrad: diffusion probabilistic model based neural vocoder with adaptive noise spectral shaping. Preprint. arXiv:2203.16749
137. Tan X, Chen J, Liu H, Cong J, Zhang C, Liu Y, Wang X, Leng Y, Yi Y, He L et al (2022) NaturalSpeech: end-to-end text to speech synthesis with human-level quality. Preprint. arXiv:2205.04421