Neural Vocoder Based on Generative Adversarial Networks Considering Speech Production Mechanism

Reo Yoneyama

Contents

Abstract

A vocoder is a system that analyzes and synthesizes speech signals based on a mathematical or statistical model, enabling efficient representation and generation of speech. Conventional vocoders are built upon the source-filter theory, where speech is generated by filtering an excitation signal through a vocal-tract filter. By explicitly modeling and estimating the source and filter components, these source-filter vocoders offer interpretable and flexible control of speech parameters such as pitch and timbre. However, because they rely on simplified mathematical assumptions such as linearity and short-time stationarity, they struggle to fully capture the complex aspects of speech, limiting achievable synthesis quality.

In contrast, neural vocoders learn a nonlinear, data-driven mapping from acoustic features to speech waveforms. These approaches do not require explicit modeling of the speech production mechanism, and the entire generation process is instead learned in a purely data-driven manner. Among these, neural vocoders based on generative adversarial networks (GANs) have become a prominent paradigm, offering a favorable balance between speech quality and computational efficiency. Nevertheless, neural vocoders still exhibit fundamental limitations. Because their internal mechanisms are learned without explicit physical structure, they operate as black boxes, which restricts interpretability, controllability, and generalization to unseen conditions. Furthermore, their discrete-time, nonlinear processing can give rise to non-physical artifacts, high-

lighting a fundamental gap between continuous-time speech production and neural waveform synthesis.

Thus, signal-processing-based vocoders and neural vocoders exhibit complementary strengths and weaknesses, and integrating the advantages of both paradigms has emerged as a promising direction. Although prior studies have explored such hybrid approaches, they often encounter a trade-off between physical interpretability and modeling flexibility, in part due to the linear or simplified assumptions embedded in the physically motivated components.

To overcome these limitations, this study proposes design principles that effectively leverage the nonlinear modeling capacity of neural networks, while guiding the model to incorporate key physical aspects of speech production. These ideas are organized into two complementary perspectives: (1) Functional perspective: reflecting the functional aspects of speech production based on the source-filter theory by encouraging an implicit decomposition within a single generator network; (2) Signal-processing perspective: exploring signal representations and processing schemes that respect the continuous-time nature of the actual speech production process.

Guided by these principles, this study investigates three themes that embody and validate the proposed design philosophy. (i) Unified source-filter modeling, which introduces implicit source-filter decomposition within a single GAN generator through regularization on intermediate signals, achieving a favorable balance between speech quality and fundamental frequency controllability. Experimental results demonstrate that unified source-filter modeling can simultaneously achieve speech quality comparable to modern neural vocoders and controllability competitive with classical source-filter approaches. (ii) Efficient unified source-filter modeling, which applies the same principle to an efficient multi-stage upsampling-based GAN generator, demonstrating its

effectiveness across different architectures and enabling real-time waveform generation on a single CPU. Experimental results indicate that the unified source-filter modeling remains effective even in upsampling-based architectures, leading to improved perceptual quality and real-time inference performance. (iii) Aliasing-free neural waveform synthesis, which provides a theoretical analysis of aliasing, one of the most prominent non-physical artifacts in neural vocoders, and introduces a practical formulation that avoids aliasing and demonstrates performance improvement in multiple aspects. Experimental results show that aliasing-free waveform synthesis improves robustness in unseen fundamental frequency conditions while significantly reducing computational cost. Collectively, these studies explore the intersection between the physical speech production process and neural waveform synthesis, contributing to the development of physically grounded neural vocoders.

1 Introduction

This chapter presents the background of this thesis, the landscape of related technologies, and the research objectives. Section 1.1 describes the fundamental framework of speech synthesis and the role of vocoders. Section 1.2 reviews conventional signal-processing-based vocoders and summarizes their characteristics and limitations. Section 1.3 discusses the development of neural vocoders, along with their advantages and remaining challenges. Section 1.4 surveys prior studies closely related to this dissertation on neural vocoders inspired by the physical mechanisms of speech production. Finally, Section 1.5 states the objective of this research, and Section 1.6 outlines the overall structure of this dissertation.

1.1 Research Background

A vocoder (voice coder) is a technology designed to analyze (encode) and synthesize (decode) speech signals. The concept of the vocoder was proposed by Dudley [1], and it was originally developed to achieve efficient bandwidth compression and speech coding for telecommunications. Instead of transmitting the raw speech waveform, Dudley's system extracts and transmits its physical characteristics, such as the spectral envelope in each frequency band, and reconstructs the waveform on the receiver side. This concept of analyzing and re-synthesizing speech signals later evolved beyond telecommunications and became a fundamental framework for speech synthesis. The

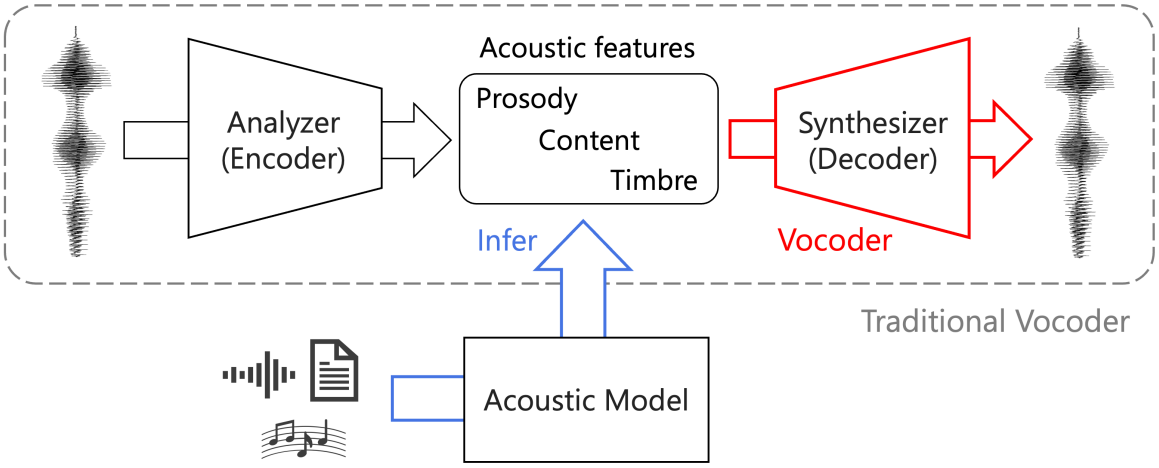

Figure 1.1: *A traditional vocoder consists of an analyzer (encoder) that decomposes a speech waveform into acoustic features, and a synthesizer (decoder) that reconstructs the waveform from these features. An acoustic model infers such features from high-level representations, such as text, musical scores, or speech waveforms, thereby forming a complete speech generation pipeline, including text-to-speech (TTS), singing voice synthesis (SVS), and voice conversion (VC) tasks. In modern terminology, the term vocoder broadly refers to the waveform synthesis (decoder) module.*

analysis-synthesis principle established by the early vocoder remains a core idea in modern speech modeling.

As illustrated in Fig. 1.1, modern speech synthesis systems typically adopt a two-stage framework consisting of (1) acoustic feature estimation by the acoustic model and (2) waveform generation by the vocoder. This pipeline enables a wide range of applications, including text-to-speech (TTS), singing voice synthesis (SVS), and voice conversion (VC). In this framework, the acoustic model predicts acoustic features, such as prosody, linguistic content, and timbre, from various types of inputs (e.g., text, musical scores, or audio waveforms), and the vocoder subsequently generates the corresponding time-domain speech waveform. Unlike traditional vocoders, which perform both analysis (encoding) and synthesis (decoding), modern vocoders serve

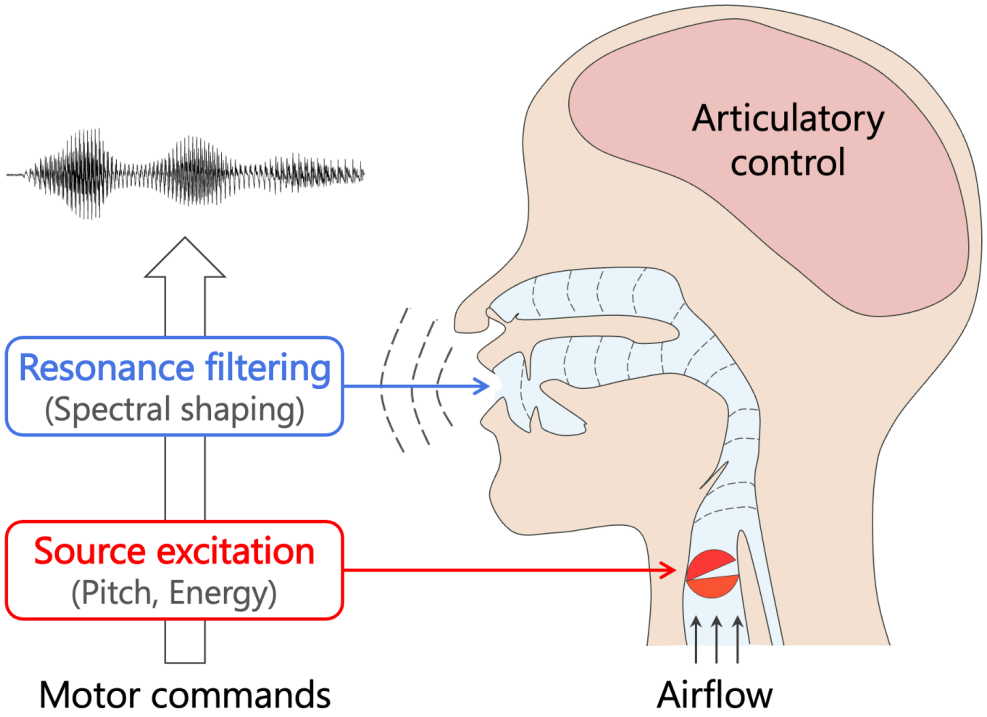

Figure 1.2: *Schematic overview of the source-filter theory of speech production.*

solely as decoders, focusing exclusively on waveform generation from acoustic features. This thesis focuses on the design and development of modern vocoders that serve as decoders.

Over the past decade, the rapid advancement of deep learning has revolutionized speech synthesis, driving major progress in both acoustic modeling and vocoder design. Processes that once relied on hand-crafted features or simplified mathematical models [2,3] are now learned automatically through data-driven optimization. The introduction of the neural vocoder [4–6], in combination with neural acoustic modeling, represented a pivotal moment that transformed the methodology of speech synthesis. Owing to their powerful modeling capacity, neural models can represent complex waveform structures and acoustic patterns that traditional signal-processing-based approaches could not capture, enabling the generation of speech that is perceptually

indistinguishable from natural recordings. As a result, expressive and high-quality speech generation has been achieved across various applications. The next section introduces the classical signal-processing-based vocoders, which laid the theoretical and structural foundation for modern approaches.

1.2 Source-Filter Models

Traditionally, vocoders have been developed as mathematical models that explicitly represent the human speech production process. According to the source-filter theory [7] (Fig. 1.2), speech can be interpreted as the combination of two primary components: a sound source and a resonance filter. The sound source corresponds to the airflow and vibration generated by the vocal folds, which act as a periodic excitation during voiced sounds and as a turbulent noise source during unvoiced sounds. The resonance filter represents the shape of the vocal tract, including the mouth, throat, and nasal cavities, which determine the resonant characteristics of the speech signal. Although the vocal-tract shape continuously changes due to articulator motion, these variations are typically modeled as a time-varying transfer function. Thus, speech production can be interpreted as a process in which the excitation signal generated by the vocal folds is shaped by the vocal tract and radiated as audible sound.

Vocoders designed according to this theory are referred to as source-filter vocoders. In typical implementations, the source is modeled as a periodic pulse train that simulates vocal-fold vibration, or as a mixed excitation signal that combines periodic pulses with noise (the pulse-plus-noise model). The vocal-tract filter is usually represented as a short-time, linear, time-invariant system. This structure enables a parametric separation between the source and filter contributions, allowing independent control of each component. For instance, modifying the source periodicity affects pitch and

intensity, while adjusting the filter's resonant characteristics alters phonetic content and timbre. Such a decompositional framework offers both interpretability and controllability, making it a powerful foundation for analyzing and manipulating speech. Even today, source-filter vocoders remain a core technology in applications that require flexible control. In particular, precise fundamental frequency (F_0) control is essential for reproducing prosody and melodic contours, playing a vital role in expressive speech and singing voice synthesis.

However, source-filter vocoders have inherent limitations. Because they rely on simplified excitation models and assumptions, such as linearity and short-term stationarity, they cannot accurately capture the complex, nonlinear, and stochastic nature of real speech signals. Consequently, the synthesized speech often lacks natural temporal and spectral variations and tends to exhibit perceptual artifacts, both of which degrade the overall speech realism. In addition, both the analysis and synthesis stages require manually designed acoustic features, which makes automatic optimization difficult and increases reliance on expert knowledge.

1.3 Neural Vocoders

In response to these limitations, neural vocoders have emerged as a new generation of speech synthesis models. Most modern neural vocoders are formulated as probabilistic generative models, in which deep neural networks (DNNs) directly learns the mapping from acoustic features to speech waveforms (or their conditional distributions). Unlike traditional vocoders, these approaches require no explicit design of source or filter components; instead, the parameters are learned automatically in a data-driven manner using large-scale speech corpora. Representative examples include the autoregressive model WaveNet [4], which generates waveforms sample by sample, and

the non-autoregressive model Parallel WaveGAN (PWG) [8], which achieves parallel waveform generation through adversarial learning. These models have achieved speech quality surpassing that of conventional source-filter vocoders, and further refinements have pushed synthesis performance to a level nearly indistinguishable from natural human speech. Among these approaches, neural vocoders based on generative adversarial networks (GAN) [9] have become particularly dominant because they offer an effective balance between speech quality and computational efficiency.

Nonetheless, neural vocoders still face several challenges. First, unlike signal-processing-based models, they require large amounts of data and computational resources for training. Second, their internal mechanisms function as a black box, making it difficult to identify which network components correspond to specific acoustic features and thereby limiting controllability. This limitation not only hinders fine-grained speech control but also affects generalization, the ability to adapt to unseen speakers or to F_0 ranges beyond the training data. Third, although neural vocoders achieve high representational power through extensive nonlinear transformations, these operations are applied to discretely sampled signals, which can introduce artificial distortions that deviate from the continuous and physically grounded speech production process [10,11].

The third issue warrants particular attention. Neural networks operate as digital signal processing systems, and according to the Nyquist-Shannon sampling theorem [12,13], the frequency range representable in discrete time is fundamentally limited. In such systems, nonlinear operations can introduce frequency components beyond the Nyquist limit, which are folded back into the baseband as aliasing. Aliasing introduces issues absent from the continuous physical process, leading to noisy artifacts and breaking the shift invariance of convolutional neural networks (CNNs). In contrast, conventional source-filter models do not include nonlinear operations that give

rise to aliasing. Even when nonlinearity is introduced, it is typically done in a controlled and physically interpretable manner, such as waveform shaping or excitation modeling, thereby preserving consistency with the assumptions of the speech production mechanism. Consequently, aliasing has not been a practical concern in traditional signal-processing-based approaches.

1.4 Physically Oriented Neural Vocoders

As discussed in the previous section, signal-processing-based and neural vocoders exhibit complementary strengths and weaknesses. To overcome this trade-off, a number of studies have explored neural vocoders physically oriented with the speech production mechanism that integrate the strengths of both paradigms. In this context, physically oriented refers to model designs that implicitly or explicitly incorporate structural assumptions inspired by aspects of the physical speech production process. This section reviews representative research trends in this direction and discusses the remaining challenges.

A major line of research focuses on incorporating linear filters to represent vocal-tract resonances while integrating them with data-driven neural components. The key idea is to design hybrid architectures that divide the modeling task between analytically defined components and neural networks, determining which aspects of speech production should be represented by explicit signal-processing models and which should be learned from data. Within this framework, several modeling strategies have emerged. One approach generates the excitation signal using DNNs while employing analytically defined linear filters for vocal-tract modeling, as demonstrated by [14–23]. Another approach adopts the opposite strategy, using parametric excitation models while estimating the filter coefficients through DNNs, as in [24–27]. More data-driven approaches

pursue a fully integrated design in which both the excitation and linear filtering processes are jointly learned within a unified neural network, as seen in [28–32]. Across these variations, linear filters serve as principled and interpretable models of vocal-tract resonances, while neural networks flexibly learn nonlinear characteristics that are difficult to capture analytically. This integration offers a structured path to combine the physical interpretability with the flexibility of deep learning.

Another research trend focuses on incorporating F_0-driven mechanisms to achieve robust pitch control and improve generalization beyond the training distribution. Since the F_0 characterizes the periodicity of vocal fold vibration, explicitly conditioning neural vocoders on F_0 enables controllable synthesis. Two main strategies have been explored in this context. The first introduces analytic F_0-based periodic signals [33–35], such as sine waves or pulse trains generated from the target F_0, similar to the excitation modeling in classical source-filter vocoders. This approach reduces the burden of learning harmonic phase progression purely from data and provides stable and accurate pitch reproduction. The second strategy employs pitch-dependent network architectures [36–38], where convolutional layers dynamically adjust their receptive fields according to the instantaneous F_0. These structures act as inductive biases that align the network's receptive field with the quasi-periodic structure of speech, enhancing F_0 controllability. While these mechanisms substantially improve pitch controllability, challenges remain in handling unseen F_0 values and in integrating them as a coherent component of the source–filter modeling framework, which directly motivates the models proposed in this thesis.

Another line of research addresses the continuous-discrete time gap inherent in digital signal processing. As discussed in the previous section, one of the most prominent issues arising from discrete-time computation is aliasing. To mitigate this problem, Lee

et al. [11] introduced anti-aliased nonlinear operations that temporarily expand the signal bandwidth to suppress spectral folding within neural vocoder architectures. Cho et al. [39] identified aliasing as a major factor that breaks the shift-equivariance property of CNNs and proposed training schemes to promote this property, thereby indirectly reducing aliasing. Shang et al. [40] further developed a training framework that quantifies and penalizes spectral asymmetry to explicitly suppress frequency-folding artifacts during optimization. Because aliasing arises from the discrete-time implementation of neural networks, it cannot be fundamentally eliminated, and existing countermeasures require extra filtering or oversampling, increasing computational cost.

1.5 Research Objective

Speech generation is a continuous-time physical process that is well described by the source-filter theory, whereas neural vocoders operate as discrete-time black-box systems trained through data-driven optimization. Although modern neural vocoders have achieved remarkable improvements in speech quality, two fundamental challenges remain. First, their internal mechanisms do not explicitly account for the actual speech production process, which in turn limits controllability and generalizability. Second, there exists a fundamental mismatch between the continuous-time nature of speech production and the discrete-time dynamics learned by neural vocoders, which inevitably introduces signal-processing inconsistencies such as aliasing.

At the same time, deep learning provides a powerful framework for discovering underlying structures and relations through high-dimensional nonlinear mappings learned directly from data. The central viewpoint of this dissertation is that combining such data-driven learning capabilities with physically oriented modeling can address the challenges of existing neural vocoders. To this end, this study proposes design prin-

ciples organized around two complementary perspectives. (1) Functional perspective: reflecting the functional aspects of the speech production based on the source-filter theory by encouraging an implicit decomposition of excitation generation and resonance filtering within a single generator network. (2) Signal processing perspective: exploring signal representations and processing schemes in neural networks that respect the continuous-time characteristics of the speech production process. Guided by these perspectives, the subsequent chapters investigate neural vocoders through three distinct themes.

- Theme 1: Unified Source-Filter Modeling

  This theme focuses on modeling both the excitation source and the vocal-tract filter within a single unified framework, leading to a unified source-filter GAN (uSFGAN). While conventional approaches often design either the source or the filter using signal-processing modules, the proposed method constructs a unified neural network in which both components are represented through learnable nonlinear transformations. Whereas prior studies impose a source-filter decomposition through parametric formulations, this framework introduces an inductive bias that guides the neural network toward representations that respect the physical speech production mechanism. This enables high-quality speech generation together with robust F_0 controllability.

- Theme 2: Efficient Unified Source-Filter Modeling

  This theme proposes SiFi-GAN (Source-Filter HiFi-GAN), which integrates the efficient upsampling-based generator architecture with the unified source-filter modeling. Such upsampling-based generators offer a more favorable balance between modeling capacity and computational efficiency than the PWG-style architecture used in uSFGAN. SiFi-GAN demonstrates that the unified source-filter

modeling can be incorporated into this efficient generator architecture, yielding improved perceptual quality while substantially reducing computational cost, enabling real-time generation even on a single CPU.

- Theme 3: Aliasing-Free Neural Waveform Synthesis

  The final theme investigates the internal behavior of neural vocoders from a signal-processing perspective, with particular focus on aliasing. Neural vocoders typically operate directly on discrete-time waveforms using convolutional neural networks, which inevitably introduce aliasing. To address this issue, this theme explores a time-frequency-domain formulation in which the generator predicts a complex spectrogram, a redundant representation of the waveform. Empirical evaluations demonstrate that the proposed time-frequency-domain vocoder, Wavehax, intrinsically avoids aliasing and yields improvements across multiple aspects of performance.

Through these investigations, this dissertation explores the connection between the physical speech production process and neural vocoders. It provides theoretical insights and design principles for developing physically grounded neural vocoders, thereby supporting continued advances in speech synthesis technology.

1.6 Thesis Overview

This thesis consists of eight chapters and an appendix. An overview of the remaining chapters is illustrated in Fig. 1.3. Chapter 2 introduces WORLD [3], a signal-processing-based vocoder based on the source-filter theory. The chapter begins with a detailed explanation of the source-filter theory, which provides a mathematical approximation of the speech production process, and then describes the analysis and

synthesis procedures that form the basis of the WORLD system. Chapter 3 reviews the foundations and development of neural vocoders. It organizes the design principles and architectural characteristics of vocoders derived from various deep generative models and explains how these differing approaches have shaped current trends in neural waveform generation. This background highlights why GAN-based vocoders have emerged as the prominent choice in practical applications. Chapter 4 surveys research efforts that incorporate explicit formulations of the physical speech production mechanism into neural vocoder design. By embedding physically motivated inductive biases into network architectures and training objectives, these approaches aim to improve vocoder performance from multiple perspectives. Chapters 5, 6, and 7 develop the three central themes of this dissertation, as outlined in the previous section. Chapter 5 presents uSFGAN, corresponding to the first theme. Chapter 6 introduces SiFi-GAN, addressing the second theme. Chapter 7 proposes Wavehax, which constitutes the third theme. Chapter 8 summarizes the contributions of this dissertation and discusses its limitations. Finally, Appendix 9 provides supplementary experiments and proofs.

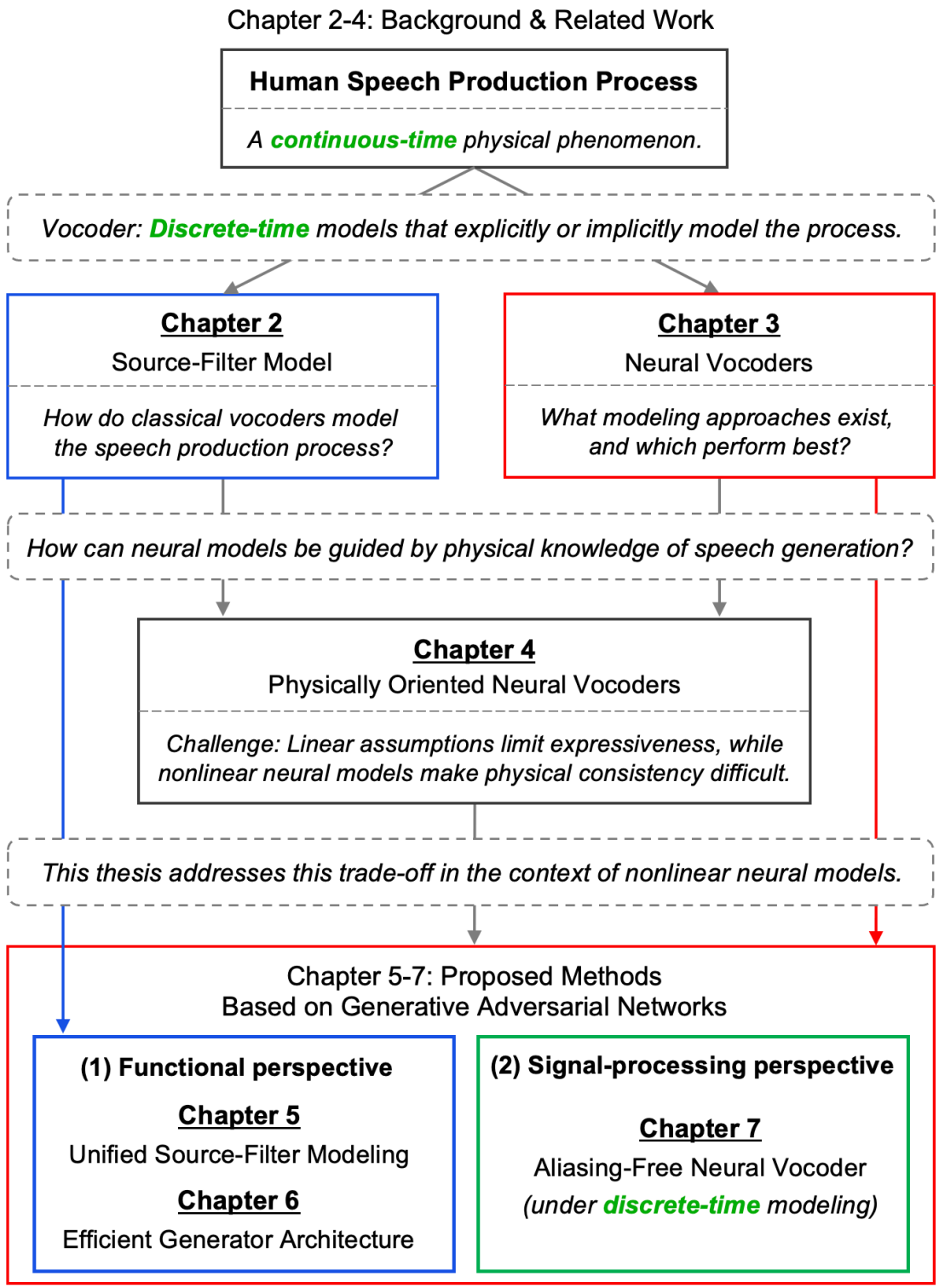

Figure 1.3: *Overview of the thesis structure.*

2 Source-Filter Model

This chapter introduces the theoretical foundation and signal-processing framework underlying speech analysis and synthesis technologies. Section 2.1 first describes the source-filter theory [7], a fundamental concept that mathematically models the speech production mechanism. Section 2.2 then introduces WORLD [3], a representative vocoder based on the source-filter theory and widely used in modern speech research and applications. Finally, Section 2.3 concludes the chapter with a brief summary.

2.1 Source-Filter Theory

2.1.1 Overview

The source-filter theory provides both an intuitive understanding of the speech production process and a theoretical foundation for speech analysis and synthesis algorithms. Speech waveforms are extremely long, complex, and nonstationary signals in the time domain. However, within a sufficiently short time window (typically less than about 10 ms), the parameters describing the speech signal can be regarded as approximately constant. Under this quasi-stationarity assumption, the speech production process can be locally modeled as a short-time linear time-invariant system, allowing the speech signal to be interpreted as the combination of a source excitation component and a resonant filter component.

The process of speech generation begins with an excitation signal produced by vocal-fold vibration and glottal airflow. This excitation is shaped by the resonant, damping, and radiation characteristics of the vocal tract before being emitted as audible sound. In the source-filter theory, this process is approximated as a linear convolution system and expressed as

$$s(t) = (h * e)(t), \tag{2.1}$$

where $e(t)$ denotes the excitation signal, $h(t)$ represents the impulse response of the vocal tract, and $s(t)$ is the observed speech waveform.

Applying the Fourier transform converts this relationship into a multiplicative form in the frequency domain:

$$S(\omega) = H(\omega)E(\omega), \tag{2.2}$$

where $S(\omega)$, $H(\omega)$, and $E(\omega)$ denote the complex spectra of the speech signal, the vocal-tract transfer function, and the excitation signal, respectively. This formulation shows that the speech spectrum can be expressed as the product of the excitation characteristics $E(\omega)$ and the vocal-tract characteristics $H(\omega)$. Taking the logarithm of the magnitude spectrum converts this multiplicative relationship into an additive one:

$$\log|S(\omega)| = \log|E(\omega)| + \log|H(\omega)|. \tag{2.3}$$

This formulation provides the basis for treating the excitation and filtering processes as separable components, an idea that underlies cepstral analysis and the spectral envelope estimation methods discussed later.

This separation is supported by the markedly different spectral variation scales of the two components: the excitation spectrum $E(\omega)$ contains sharp periodic peaks spaced according to the fundamental frequency F_0, whereas the vocal-tract filter response $H(\omega)$ exhibits a smooth spectral envelope shaped by resonant formant structures. These contrasting spectral characteristics enable the approximate independent estimation of the source and filter contributions.

2.1.2 Source Modeling

This section describes how the excitation signal of speech is mathematically modeled. In the source-filter theory, the excitation signal corresponds to the glottal airflow produced by vocal-fold vibration and serves as the input to the vocal-tract filter. The following subsections introduce the periodic excitation model and its extension to a mixed excitation representation.

Periodic Excitation Model

In voiced speech segments, periodic excitation generated by vocal fold vibration propagates through the vocal tract and is shaped by its resonant characteristics. Ideally, this excitation signal $e(t)$ can be approximated as a sequence of impulses with a constant fundamental period T_0 within a short time frame:

$$e(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_0), \tag{2.4}$$

where $\delta(t)$ denotes the Dirac delta function.

The Fourier transform of this signal is given by

$$E(\omega) = \frac{2\pi}{T_0} \sum_{k=-\infty}^{\infty} \delta(\omega - k\omega_0), \tag{2.5}$$

where $\omega_0 = 2\pi/T_0 = 2\pi F_0$, and $F_0 = 1/T_0$ represents the fundamental frequency. Accordingly, $E(\omega)$ exhibits sharp peaks at integer multiples of ω_0, corresponding to frequency positions at integer multiples of F_0, thereby forming a harmonic structure. The fundamental frequency F_0 reflects the periodic opening and closing of the vocal folds and is a key physical parameter governing the excitation characteristics. Moreover, perceived pitch is strongly related to F_0, although the human auditory system exhibits an approximately logarithmic sensitivity to frequency.

Mixed Excitation Model

Speech is not composed solely of periodic components. It also contains aperiodic components generated by unvoiced segments and consonants. Consonants, in particular, are produced when the airflow in the vocal tract is constricted or blocked, which creates turbulent noise dominated by aperiodic energy. Therefore, it is natural and effective to model the excitation signal as a mixed excitation source comprising both periodic and aperiodic components.

To represent differences in periodicity across frequency bands, the mixed excitation source introduces an independent periodicity parameter $\alpha_b(\omega)$ for each frequency band b. This parameter can be defined either as a binary variable indicating voiced or unvoiced states [41], or as a continuous value representing the degree of periodicity [42]. The excitation spectrum in each band is then expressed as a weighted combination of

harmonic and noise components:

$$E_b(\omega) = \alpha_b(\omega) E_b^{(\mathrm{h})}(\omega) + \big(1 - \alpha_b(\omega)\big) E_b^{(\mathrm{n})}(\omega), \tag{2.6}$$

where $\alpha_b(\omega) \in [0, 1]$ denotes the periodicity ratio in band b. Here, $\alpha_b(\omega) = 1$ corresponds to fully periodic, whereas $\alpha_b(\omega) = 0$ represents purely aperiodic.

This formulation enables excitation to be described with frequency-dependent periodicity rather than a single global parameter. Consequently, both the strongly periodic low-frequency regions and the highly aperiodic high-frequency regions can be handled naturally within a unified framework. Furthermore, mixed-excitation models have proven highly useful in speech synthesis and modification. By controlling the band-wise periodicity parameter α_b, we can flexibly manipulate the perceived voice quality of synthesized speech. For example, reducing α_b increases the relative contribution of aperiodic components, yielding a breathy or whisper-like timbre.

2.1.3 Filter Modeling

This section describes the modeling of the vocal-tract filter. In the source-filter framework, the vocal-tract filter shapes the spectral envelope of the excitation signal. In digital signal processing, it is typically implemented as either a finite impulse response (FIR) system or an infinite impulse response (IIR) system.

Finite Impulse Response Model

The cepstrum is obtained by applying the inverse Fourier transform $\mathcal{F}$ to the logarithmic amplitude spectrum of a speech signal. It represents the periodic structure of the spectrum on a time-like axis called the quefrency. For a speech signal $s(t)$ with

spectrum $S(\omega)$, the cepstrum is defined as

$$c_s(\tau) = \mathcal{F}^{-1}\{\log|S(\omega)|\}, \tag{2.7}$$

Through this transformation, the fine periodic structure of the spectrum is mapped to the high-quefrency region, whereas the smooth spectral envelope appears in the low-quefrency region.

According to the source-filter theory, the logarithmic amplitude spectrum of speech can be expressed as Eq. (2.3), which leads to an additive relationship in the cepstral domain:

$$c_s(\tau) = c_e(\tau) + c_h(\tau), \tag{2.8}$$

where $c_e(\tau)$ and $c_h(\tau)$ denote the cepstral components of the excitation source and the vocal-tract filter, respectively. Because these two components occupy different quefrency regions, the source-related structure appearing at high quefrencies and the vocal-tract envelope at low quefrencies, they can be separated by retaining only the low-quefrency coefficients. This liftering operation, implemented by multiplying the cepstrum by a rectangular window, is equivalent (via the convolution theorem) to smoothing the logarithmic amplitude spectrum. As a result, sharp harmonic structures are attenuated and a smooth spectral envelope corresponding to vocal-tract resonances can be obtained.

For implementation as an FIR filter, the low-quefrency cepstrum $c_s^{(\mathrm{low})}(\tau)$ is transformed back into the frequency domain and exponentiated to reconstruct the spectral envelope:

$$\hat{H}(\omega) = \exp\big(\mathcal{F}\{c_s^{(\mathrm{low})}(\tau)\}\big), \tag{2.9}$$

where $\mathcal{F}\{\cdot\}$ denotes the Fourier transform. The resulting $\hat{H}(\omega)$ represents the frequency response of the vocal-tract filter, and its inverse Fourier transform yields a finite-length impulse response $\hat{h}(t) = \mathcal{F}^{-1}\{\hat{H}(\omega)\}$.

Cepstrum-based vocal-tract modeling offers numerical stability and is straightforward to implement. However, its accuracy is highly sensitive to the choice of the quefrency threshold used for low-pass liftering. A cutoff that is too low causes excessive smoothing of the spectral envelope, while one that is too high fails to sufficiently suppress harmonic structures originating from the excitation source. To address these limitations, advanced source-filter models such as STRAIGHT [2] and WORLD [3] employ F_0-adaptive techniques, as explained in Section 2.2.3.

Infinite Impulse Response Model

Whereas the FIR model approximates the spectral envelope through nonrecursive smoothing in the frequency domain, the IIR model captures the vocal-tract characteristics by employing a recursive structure in the time domain. A representative and widely used implementation of this approach is linear predictive coding (LPC) [43, 44].

In LPC, the speech signal is modeled as an autoregressive process, assuming that the current sample is a linear combination of past samples plus an excitation term:

$$s(t) = \sum_{k=1}^{p} a_k s(t-k) + e(t), \tag{2.10}$$

where $s(t)$ denotes the speech waveform, a_k are the linear prediction coefficients, and p is the model order. The residual signal $e(t)$ represents the component that cannot be linearly predicted from past samples. Under the quasi-stationarity assumption of speech, the coefficients a_k are treated as constant within each short analysis frame

and are typically estimated by minimizing the mean-square prediction error. This recursive formulation effectively models the resonant characteristics and provides a compact, smooth representation of the spectral envelope.

Applying the z-transform to the above relation yields

$$S(z) = \left(\sum_{k=1}^{p} a_k z^{-k} \right) S(z) + E(z), \tag{2.11}$$

which can be rearranged as

$$S(z) \left(1 - \sum_{k=1}^{p} a_k z^{-k} \right) = E(z), \tag{2.12}$$

and thus,

$$S(z) = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}} E(z). \tag{2.13}$$

This formulation has the same structure as the source-filter model $S(z) = H(z)E(z)$, where the vocal-tract transfer function given by

$$H(z) = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}}. \tag{2.14}$$

Accordingly, the LPC formulation models the vocal tract as an all-pole system, with the excitation signal $E(z)$ serving as its input.

In the time domain, the corresponding residual signal is defined as

$$e(t) = s(t) - \sum_{k=1}^{p} a_k s(t-k), \tag{2.15}$$

which aligns with the source-filter relation $s(t) = (h * e)(t)$. Thus, LPC provides a unified representation of both glottal excitation and stochastic noise as the residual

signal that drives the vocal-tract filter.

However, to ensure filter stability, all poles of the system must lie inside the unit circle. Moreover, an all-pole model such as LPC cannot explicitly represent spectral zeros (anti-formants), which correspond to frequency regions where acoustic cancellations in the vocal tract attenuate the spectrum.

2.2 WORLD: Conventional Source-Filter Model

2.2.1 Overview

The WORLD vocoder [3] is a representative implementation of the source-filter model for speech analysis and synthesis. As shown in Fig. 2.1, WORLD consists of three main analysis algorithms: Harvest [45] for F_0 estimation, CheapTrick [46] for spectral envelope estimation, and D4C [47] for band-wise aperiodicity estimation. The speech waveform is then reconstructed from these parameters using a synthesis algorithm. The following subsections describe each algorithm in detail.

2.2.2 Harvest: Fundamental Frequency Estimation

Overview

The fundamental frequency is the most essential parameter for representing the periodic structure of voiced sounds. Accurate estimation of F_0 is crucial because subsequent analyses, such as spectral envelope and aperiodicity estimation, are performed in an F_0-synchronous manner. However, natural speech often contains non-periodic components, such as breath noise and background noise, which easily cause false F_0 detections when simple periodicity-based detectors are used. To address this problem,

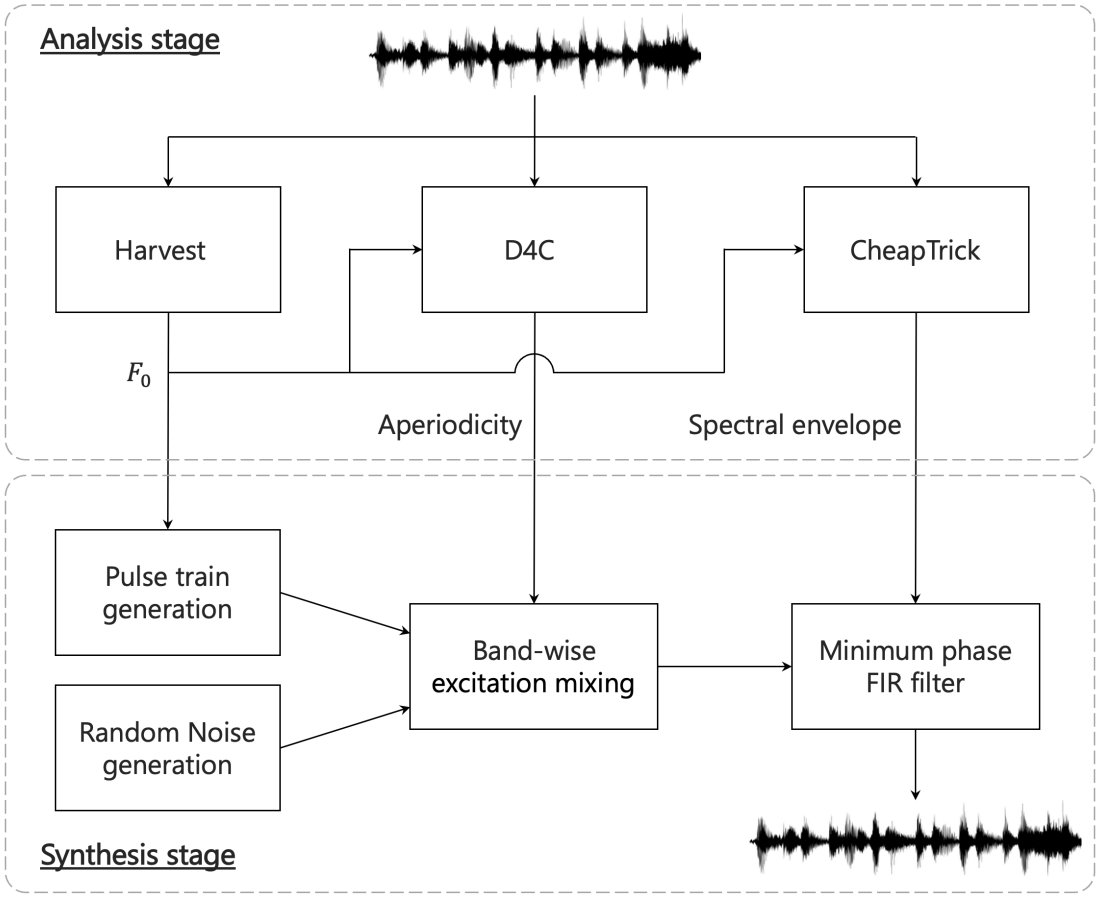

Figure 2.1: *Block diagram of the analysis–synthesis framework of the WORLD vocoder [3], where three speech parameters are extracted by separate analysis algorithms and used for the synthesis algorithm based on the source–filter theory [7].*

the Harvest algorithm [45] was proposed, a high-performance F_0 estimation algorithm that achieves both accuracy and robustness. Its reliability-driven framework and temporally consistent design make it a key component of the WORLD vocoder system.

The Harvest algorithm consists of two major stages. In the first stage, multiple F_0 candidates are extracted from the input waveform using a bank of bandpass filters. By employing multi-band filtering and overlapping strategies, Harvest maintains stable candidate extraction even for speech signals contaminated by non-periodic components.

In the second stage, each candidate is refined and evaluated based on its reliability, which is computed using instantaneous frequency analysis to measure the harmonic consistency of each candidate. A temporally coherent F_0 contour is then constructed by integrating the reliability scores with dynamic smoothing constraints. This separation between candidate extraction and temporal refinement enables both diversity and continuity in the F_0 trajectory, resulting in stable and perceptually consistent F_0 estimation.

Step1: Candidate Extraction and Refinement

The input speech signal is first decomposed using a bank of narrow-bandpass filters with logarithmically spaced center frequencies ω_c (e.g., 40 channels per octave). Each filter is designed to isolate a limited frequency region centered at ω_c and outputs a quasi-sinusoidal waveform dominated by the spectral component around that center frequency. From each filtered waveform, a preliminary estimate of the local fundamental period is obtained using measures such as zero-crossing or peak-to-peak intervals. The average of these local period estimates within a short time frame is then calculated, and its reciprocal is taken as the instantaneous F_0 value for that band. To ensure that each filter output properly represents the target spectral region, only those whose dominant frequency lies within $\pm 10\%$ of the center frequency ω_c are retained. Outputs deviating beyond this range are considered unreliable, as they likely contain interference from adjacent harmonics or noise components.

To suppress spurious detections, the remaining local period estimates are compared across adjacent frequency bands. Only the values that are consistent across multiple neighboring bands are retained as valid F_0 candidates. This cross-band consistency check effectively removes false periodicities caused by strong harmonic overtones or

transient noise, resulting in a set of temporally dense and spectrally consistent F_0 candidates.

Each F_0 candidate is then refined using the instantaneous frequency, which is defined as the temporal derivative of the instantaneous phase of the short-time spectrum. The instantaneous phase represents the time-varying argument of the complex-valued spectrum $S(\omega, t)$, and its derivative provides the true local angular frequency of each spectral component. Harvest employs Flanagan's formula to compute the instantaneous frequency because it allows direct computation from the real and imaginary parts of the short-time spectrum. For a candidate with angular frequency ω_0, the instantaneous frequency at the k-th harmonic, denoted by $\omega_i(k\omega_0, t)$, is calculated as:

$$\omega_i(\omega, t) = \frac{\Re[S(\omega,t)]\Im[\partial S(\omega,t)/\partial t] - \Im[S(\omega,t)]\Re[\partial S(\omega,t)/\partial t]}{|S(\omega,t)|^2},$$

where $S(\omega, t)$ is the short-time spectrum obtained using a Blackman window of width $3T_0$, and $T_0 = 2\pi/\omega_0$ is the candidate period.

The refined fundamental frequency $\hat{\omega}_0$ is then computed as the weighted average of the instantaneous frequencies of its harmonic components:

$$\hat{\omega}_0 = \frac{\sum_{k=1}^{K} |S(k\omega_0, t)|\, \omega_i(k\omega_0, t)}{\sum_{k=1}^{K} k\, |S(k\omega_0, t)|}, \tag{2.16}$$

where K denotes the number of harmonics used for refinement. Since speech has strong harmonics only in the low-to-mid frequency range, using about six harmonics achieves a good balance between capturing the periodic structure and avoiding unreliable high-frequency components.

Finally, the reliability of each candidate is evaluated by

$$r = \frac{K}{\sum_{k=1}^{K} \left| \frac{\omega_i(k\omega_0, t)}{k} - \omega_0 \right|}. \tag{2.17}$$

This measure represents the degree of harmonic consistency among the K instantaneous frequencies. When the instantaneous frequencies $\omega_i(k\omega_0, t)$ closely follow the expected integer-multiple relationship $k\omega_0$, the deviations in the denominator become small, resulting in a larger r value. Thus, a higher r indicates stronger harmonic regularity and greater confidence in the corresponding F_0 candidate.

Step2: Construction of F_0 Trajectory

In the second stage, Harvest constructs a continuous and reliable F_0 contour from the frame-wise candidates and their associated reliability scores obtained in the previous step. The process focuses on temporal coherence and robustness to spurious or unstable detections. First, unreliable candidates with low reliability are removed, leaving only those that represent stable periodicity in the waveform. Next, temporal continuity is enforced by discarding candidates that exhibit abrupt frequency changes between adjacent frames, ensuring that the resulting trajectory evolves smoothly over time. Short and isolated voiced segments likely caused by transient noise are excluded, while small gaps between reliable voiced regions are bridged by interpolation to maintain continuity. After connecting the voiced regions, the F_0 trajectory is smoothed using a low-pass filter to suppress residual fluctuations and to produce a perceptually natural F_0 contour.

2.2.3 CheapTrick: Spectral Envelope Estimation

Overview

The spectral envelope $|H(\omega)|$ characterizes the acoustic response of the vocal tract and thus largely determines the timbre and formant structure of speech. In practical signal processing, however, estimates of the envelope often vary with window placement, local harmonic interference, and non-periodic noise components. The CheapTrick algorithm [46] was developed to provide a pitch-synchronous, temporally stable, and computationally efficient spectral envelope estimator. In its design, CheapTrick seeks to suppress harmonic interference via a pitch-synchronous window, enforce temporal consistency through smoothing in the frequency domain, and remove residual time-varying components via cepstral liftering.

Step 1: Pitch-Synchronous Windowing

CheapTrick performs a pitch-synchronous analysis to eliminate time-varying components in the power spectrum. For each analysis frame, the fundamental period $T_0 = 1/F_0$ obtained from the estimated F_0 is used to determine the window length. A Hanning window with a length of three pitch periods ($L = 3T_0$) is applied to the waveform centered at the analysis time t_c:

$$x_w(t) = x(t)\, w(t - t_c), \quad w(\tau) = 0.5 \left(1 - \cos \frac{2\pi\tau}{L}\right).$$

When a waveform is extracted using a Hanning window of three fundamental periods, the total power of the windowed signal becomes almost constant regardless of the

extraction timing, as expressed by

$$\int_0^{3T_0} \big(y(t)w(t)\big)^2 dt = 1.125 \int_0^{T_0} y^2(t)dt.$$

The power spectrum is then calculated as

$$P(\omega) = |X_w(\omega)|^2.$$

If adjacent harmonics interfere through convolution with the window spectrum, the resulting spectral overlap causes time-dependent variations in the power spectrum. The pitch-synchronous Hanning window used in CheapTrick is designed so that its spectral zeros coincide with integer multiples of the fundamental angular frequency, $n\omega_0$ ($\omega_0 = 2\pi/T_0$). These zeros eliminate such interference, making the power at $n\omega_0$ temporally static.

Step 2: Spectral Smoothing

The power spectrum $P(\omega)$ obtained in Step 1 can contain spectral zeros, which result in numerical instability when taking the logarithm in the subsequent cepstral liftering process. To prevent this problem, CheapTrick performs F_0-synchronous smoothing in the frequency domain by convolving $P(\omega)$ with a rectangular window whose width is determined from the fundamental angular frequency $\omega_0 = 2\pi/T_0$:

$$P_s(\omega) = \frac{3}{2\omega_0} \int_{-\omega_0/3}^{\omega_0/3} P(\omega + \lambda)\, d\lambda.$$

The smoothing bandwidth of $2\omega_0/3$ is empirically chosen to ensure numerical stability while minimizing inter-harmonic interference and avoiding excessive spectral smoothing, which can otherwise blur formant structures. This step stabilizes the power spec-

trum and provides a robust input for the subsequent cepstral analysis.

Step 3: Liftering and Spectral Recovery

After the spectral smoothing in Step 2, CheapTrick performs a cepstral-domain processing to remove residual time-varying components and to reconstruct the spectral envelope. A logarithmic transform and inverse Fourier transform yield the cepstrum:

$$p_s(\tau) = \mathcal{F}^{-1}\{\log P_s(\omega)\}.$$

Two liftering functions are applied to the cepstrum: a smoothing lifter $l_s(\tau)$ to suppress rapid cepstral fluctuations and a recovery lifter $l_q(\tau)$ to restore the spectral resolution reduced by the smoothing process. The overall operation is expressed as

$$P_l(\omega) = \exp\{\mathcal{F}[\, l_s(\tau)\, l_q(\tau)\, p_s(\tau)\,]\}.$$

The smoothing lifter $l_s(\tau)$ is designed using a sinc function:

$$l_s(\tau) = \frac{\sin(\pi f_0 \tau)}{\pi f_0 \tau},$$

which removes time-varying components that concentrate around multiples of the fundamental period T_0, while simultaneously performing spectral smoothing. The recovery lifter $l_q(\tau)$ compensates for the amplitude attenuation caused by the smoothing process and is given by

$$l_q(\tau) = q_0 + 2q_1 \cos(2\pi f_0 \tau),$$

where q_0 and q_1 are empirically determined constants that restore the overall spectral

level. Finally, the refined power spectrum is obtained as

$$P_l(\omega) = \exp\left\{q_0 \log P(\omega) + q_1 \left(\log P(\omega+\omega_0)P(\omega-\omega_0)\right)\right\},$$

and the spectral envelope is estimated by taking the square root:

$$|H(\omega)| = \sqrt{P_l(\omega)}.$$

This liftering procedure successfully removes time-varying components while recovering the spectral shape of periodic signals, yielding a smooth yet accurate estimate of the vocal-tract spectral envelope.

2.2.4 D4C: Band-Aperiodicity Estimation

Overview

The D4C (Definitive Decomposition Derived Dirt-Cheap) algorithm estimates aperiodicity, defined as the ratio of periodic to aperiodic (non-periodic) energy in the excitation signal. This parameter characterizes breathiness and naturalness of speech. A key motivation for using group delay in D4C is that it provides a representation in which the harmonic structure is clearly emphasized. D4C takes advantage of this property by constructing a parameter derived from group delay, from which an almost purely periodic signal can be reconstructed. The deviation between this ideal periodic component and the original speech signal then reflects the aperiodic energy, enabling a robust estimation of band-wise aperiodicity. The D4C algorithm consists of three main steps: (1) extraction of temporally static parameters based on group delay, (2) parameter shaping to emphasize the fundamental periodic component, and (3) estimation of band-wise aperiodicity using a spectral power ratio.

Step 1: Temporally Static Group-Delay

Let $s(t)$ be a speech signal and let $S(\omega)$ denote its complex spectrum. In general, group delay is defined as the frequency derivative of the phase,

$$\tau_g(\omega) = -\frac{d\phi(\omega)}{d\omega}, \tag{2.18}$$

but this conventional definition is numerically unstable. An equivalent formulation expresses group delay in terms of the real $\Re\{\cdot\}$ and imaginary $\Im\{\cdot\}$ parts of $S(\omega)$, its frequency derivative $S'(\omega) = \mathcal{F}\{-jt\,s(t)\}$, and the power spectrum $|S(\omega)|^2$:

$$\tau_g(\omega) = \frac{\Re\{S'(\omega)\}\Im\{S(\omega)\} - \Re\{S(\omega)\}\Im\{S'(\omega)\}}{|S(\omega)|^2}. \tag{2.19}$$

D4C analyzes aperiodicity using a parameter derived from the group-delay formulation in Eq. (2.19). Let τ denote the center time of an analysis frame, and let $w(t)$ be a window function. The windowed signal is defined as $y(t,\tau) = w(t-\tau)\,s(t)$, and its spectrum as $S(\omega,\tau) = \mathcal{F}\{y(t,\tau)\}$. In general, group-delay computation from windowed frames is highly sensitive to the temporal position τ; even slight shifts can lead to variations in the extracted parameters and consequently the synthesized speech. D4C addresses this issue by constructing temporally static parameters that are robust to the window position.

To suppress the dependence on τ, D4C employs pitch-synchronous windows, similarly to the CheapTrick algorithm. The numerator of Eq. (2.19) is computed using a Blackman window of length $4T_0$, which provides favorable characteristics for evaluating the group-delay numerator. Using this window, the numerator is modified so that its dependence on the analysis time τ is effectively canceled, yielding a temporally static counterpart of the original group-delay numerator. In contrast, the denominator of

Eq. (2.19) is obtained from the power spectrum computed using a Hanning window of length $4T_0$. Because the Hanning window has a main-lobe bandwidth of $\omega_0/2$, each harmonic component at $n\omega_0$ can be analyzed with minimal interference from adjacent harmonics. For additional numerical stability, the resulting power spectrum is further smoothed in the frequency domain by convolution with a rectangular window of width ω_0.

Step 2: Parameter Shaping

The group-delay parameter $\tau_g(\omega)$ obtained in Step 1 exhibits a periodic structure with period $\omega_0 = 2\pi/T_0$. However, the corresponding time-domain signal obtained via inverse Fourier transform contains higher-order peaks at nT_0 $(n > 1)$ as well as slowly varying DC components, which arise from the windowing process. To isolate only the fundamental periodic structure, D4C smooths $\tau_g(\omega)$ via frequency domain processing so that these higher-order components are attenuated and the resulting waveform approximates a single sinusoid with period T_0.

First, the components corresponding to $2nT_0$ $(n > 1)$ are reduced using the spectral smoothing

$$\tau_{gs}(\omega) = \frac{2}{\omega_0} \int_{-\omega_0/4}^{\omega_0/4} \tau_g(\omega + \lambda)\, d\lambda. \tag{2.20}$$

which corresponds to multiplication by a sinc-shaped function in the time domain. This sinc function has zeros at $2nT_0$ $(n \in \mathbb{N})$, thereby suppressing higher-order harmonic components at these positions while preserving the fundamental component at T_0. Although this smoothing does not perfectly eliminate all components above $2T_0$, the remaining higher-order terms have sufficiently low power, which can be ignored in practical use. To further remove the DC component, an additional smoothing is

applied:

$$\tau_{gb}(\omega) = \frac{1}{\omega_0} \int_{-\omega_0/2}^{\omega_0/2} \tau_{gs}(\omega + \lambda)\, d\lambda, \tag{2.21}$$

Finally, the fundamental component is obtained by subtracting the slowly varying DC component from the smoothed parameter:

$$\tau_D(\omega) = \tau_{gs}(\omega) - \tau_{gb}(\omega). \tag{2.22}$$

The resulting parameter $\tau_D(\omega)$ approximates a sinusoid with frequency ω_0, corresponding to the fundamental periodic structure of the signal. This parameter serves as the basis for the subsequent aperiodicity estimation.

Step 3: Band-Wise Aperiodicity Estimation

In the final stage, D4C estimates the band-wise aperiodicity using the temporally static parameter $\tau_D(\omega)$. For each center frequency $\omega_c = 2\pi f_c$, $\tau_D(\omega)$ is multiplied by a low-side-lobe Nuttall window $w(\omega - \omega_c)$, and the windowed spectrum is transformed into the time domain:

$$p(t, \omega_c) = \mathcal{F}^{-1}\left[w(\omega)\, \tau_D\left(\omega - \left(\omega_c - \frac{w_l}{2}\right)\right)\right]. \tag{2.23}$$

The power envelope $|p(t, \omega_c)|^2$ is then sorted in descending order so that the concentrated harmonic energy appears first, followed by the aperiodic components. The cumulative power function is defined as

$$p_c(t, \omega_c) = 1 - \int_0^t p_s(\lambda, \omega_c)\, d\lambda, \tag{2.24}$$

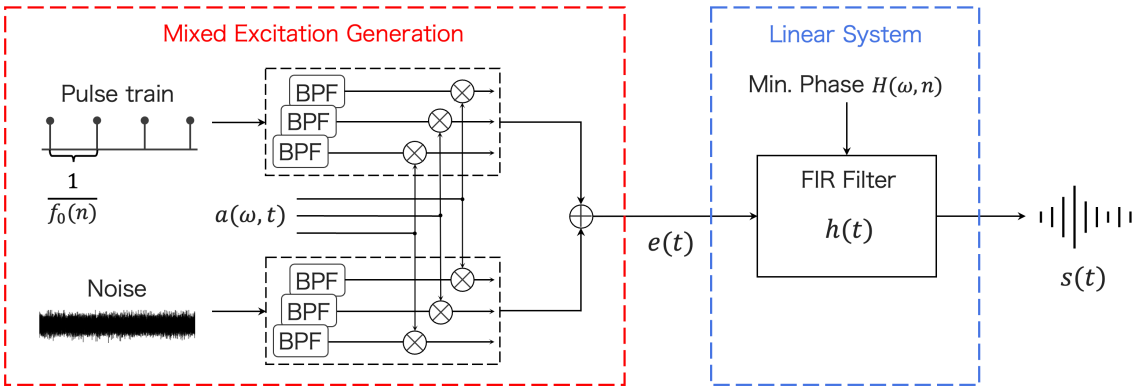

Figure 2.2: *Schematic overview of the WORLD vocoder synthesis algorithm [3]. BPE denotes a band-pass filter.*

where $p_s(t, \omega_c)$ denotes the sorted power values.

Finally, the band-wise aperiodicity $\mathrm{ap}(\omega_c)$ is computed as the ratio between the total power and the power contained in the main-lobe region associated with the harmonic component:

$$\mathrm{ap}(\omega_c) = -10 \log_{10}\big(p_c(2w_{\mathrm{bw}}, \omega_c)\big), \tag{2.25}$$

where w_{bw} denotes the main-lobe bandwidth of the window function.

2.2.5 Speech Synthesis Algorithm

The WORLD vocoder reconstructs a speech waveform from three frame-level parameters estimated in the analysis stage: the fundamental frequency $f_0(n)$, the spectral envelope $|H(\omega, n)|$, and the aperiodicity $a(\omega, n)$. An overview of the synthesis algorithm is shown in Fig. 2.2.

In the synthesis stage, a mixed excitation signal $e(t)$ is first generated. For frames in which voiced speech is detected, a periodic pulse train is generated with a fundamental

period of $1/f_0(n)$, and a noise component is added according to the aperiodicity parameter $a(\omega, n)$. For unvoiced frames, the excitation consists solely of noise. In this way, the excitation is constructed as a mixture of an F_0-synchronous periodic component and an aperiodic noise component, with the mixing ratio controlled by the aperiodicity parameter. The excitation is assumed to have an approximately flat spectral envelope, and its phase characteristics are determined solely by the positions of the pulses. Because the pulses align the phase in a manner similar to natural glottal excitation, they contribute to producing natural-sounding speech.

The excitation signal is then passed through a synthesis filter $h(t)$ obtained from the spectral envelope $|H(\omega, n)|$ estimated by CheapTrick. Because CheapTrick provides only the amplitude spectrum, a minimum-phase spectrum consistent with the estimated envelope is first constructed. This minimum-phase spectrum is then transformed into a time-domain impulse response by applying the inverse Fourier transform. Finally, the filtered excitation in each frame is combined with its neighboring frames using the overlap-add technique, which provides smooth frame transitions and prevents discontinuities in the synthesized waveform. Through this process, WORLD achieves natural speech synthesis within a purely signal-processing framework while maintaining high interpretability and computational efficiency.

2.3 Summary

This chapter has described the source-filter theory of speech production and the signal-processing vocoder WORLD. Speech generation can be modeled as the convolution of a source excitation signal and a vocal-tract filter. The excitation signal is characterized by different excitation models, voiced, unvoiced, or mixed excitation, while the resonance of the vocal tract can be represented by a linear filter, such as the

FIR or IIR system. Under this assumption, the source and filter components are additively separable in the logarithmic spectral domain, forming the theoretical foundation for practical speech-analysis algorithms in the cepstral domain.

The source-filter model implemented in WORLD employs three analysis algorithms: Harvest, CheapTrick, and D4C. Harvest estimates F_0 robustly and accurately by evaluating signal periodicity through a bank of bandpass filters and integrating reliability across multiple frequency bands. CheapTrick uses a pitch-synchronous analysis window based on the estimated F_0 to obtain short-time spectra, and estimates a temporally stable spectral envelope via spectral smoothing and cepstral-domain processing. D4C analyzes the group-delay characteristics of the signal to estimate the band aperiodicity precisely for each frequency band, thereby improving the quality of the excitation model. In WORLD's synthesis stage, an excitation signal is generated from these estimated parameters, filtered through a minimum-phase FIR filter, and the speech waveform is reconstructed via overlap-add synthesis to maintain temporal continuity.

Source-filter vocoders, such as WORLD, provide a signal-processing framework that mathematically models the human speech production process. They allow independent control of acoustic parameters such as F_0, spectral envelope, and aperiodicity, making them highly effective in applications that require flexible manipulation of speech characteristics, such as voice conversion and singing-voice synthesis. However, because these systems typically employ minimum-phase models without explicitly reconstructing the phase spectrum, their representational capacity is limited. Furthermore, as they rely on empirically designed algorithms, approximation errors and structural constraints may affect the synthesis, which sometimes result in perceptually unnatural speech.

3 Neural Vocoder

This chapter provides an overview of research trends in neural vocoders, especially from the perspective of deep generative models. Additionally, we discuss four key aspects to evaluate the performance of neural vocoders. Section 3.1 provides an overview of the main frameworks of probabilistic generative models. Sections 3.2 through 3.4 present representative neural vocoders developed within these frameworks, summarizing their characteristics. Section 3.5 organizes the evaluation criteria for neural vocoders, which serve as guidelines for assessing the proposed methods in the subsequent chapters. Finally, Section 3.6 provides a summary of this chapter.

3.1 Neural Vocoders Based on Generative Models

Conventional signal-processing-based vocoders [2, 3] are typically designed on the basis of source-filter theory [7], in which speech signals are explicitly represented in terms of source and filter components. While this framework offers high interpretability and controllability aligned with the human speech production mechanism, it relies on idealized assumptions such as linearity, time invariance, and independence between components.

In contrast, neural vocoders dispense with an explicit parametric formulation of the speech production process and instead learn the probability distribution of the speech waveform $\boldsymbol{x}$ directly from large amounts of data (Fig. 3.1). This data-driven modeling

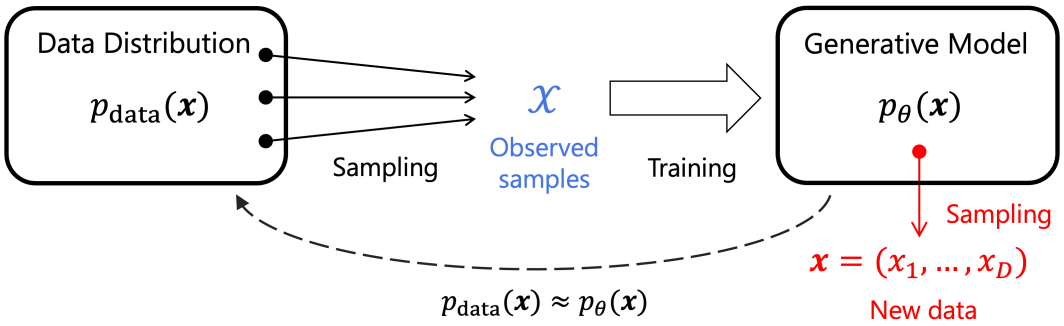

Figure 3.1: *Conceptual illustration of a generative model. A generative model* $p_\theta(\boldsymbol{x})$ *is trained to approximate the true data distribution* $p_{\text{data}}(\boldsymbol{x})$ *from observed samples* $\boldsymbol{x}$*. After training, new data can be synthesized by sampling from the learned distribution.*

paradigm enables the model to learn underlying patterns directly from observed data rather than from predefined assumptions. Consequently, neural vocoders can capture latent structures and dependencies that remain beyond the descriptive capacity of conventional frameworks.

Furthermore, by formulating the model as a conditional probability distribution $p(\boldsymbol{x} \mid \boldsymbol{c})$, neural vocoders enable conditional speech generation based on arbitrary inputs such as mel-spectrograms. This formulation also allows conditioning on higher-level attributes, including speaker identity, emotional labels, and phonetic representations, thereby providing flexible and high-level control over diverse speech characteristics. In summary, neural vocoders circumvent the structural constraints imposed by traditional source-filter assumptions and offer a more powerful and versatile framework of speech modeling.

Deep generative models encompass a range of paradigms, including autoregressive models, normalizing flows, and GANs, each offering distinct strengths and limitations. Their key characteristics and differences are summarized in Fig. 3.2. Since neural vocoders are constructed on top of these diverse frameworks, the choice of the underlying generative model plays a critical role in determining their performance. As the

	Autoregressive models <Sec. 3.2>	Normalizing flows <Sec. 3.3>	GANs <Sec. 3.4>
Probability density estimation	Possible	Possible	Not possible
Structural constraints	Sequential generation	Invertibility, Jacobian	—
Inverse transformation	—	Deterministic	Not possible
Assumption on parametric distribution	Conditional probability	Latent variables only	Latent variables only
Learning criterion	Likelihood maximization	Likelihood maximization	Min–max game
Minimized divergence	KLD	KLD	JSD*
Practical challenges	Sequential generation	Model size	Sample diversity

Figure 3.2: *Comparison of major generative model families in terms of their probabilistic modeling properties and learning objectives. KLD denotes the Kullback–Leibler divergence. GANs correspond to minimizing the Jensen–Shannon divergence (JSD) only under the assumption of a cross-entropy loss for the discriminator, as in the original GAN formulation [9].*

theoretical foundation of this dissertation, the following sections survey representative methods from the major families of deep generative models and distill their characteristics and design principles. This overview clarifies the fundamental frameworks that have supported the diverse development of neural vocoder research.

3.2 Autoregressive Models

3.2.1 General Formulation

Autoregressive models are probabilistic generative models that represent a joint distribution by decomposing it into a sequence of conditional distributions. In speech

modeling, the waveform is naturally treated as a time-ordered sequence, and generation is typically assumed to proceed sample by sample along the temporal axis. For a speech waveform $\boldsymbol{x} = (x_1, x_2, \ldots, x_T)$ and conditioning features $\boldsymbol{c}$, an autoregressive model specifies the following factorization of the conditional joint distribution:

$$p(\boldsymbol{x} \mid \boldsymbol{c}) = \prod_{t=1}^{T} p(x_t \mid x_{<t}, \boldsymbol{c}), \tag{3.1}$$

where $x_{<t} = (x_1, \ldots, x_{t-1})$ denotes all preceding samples. Each conditional distribution $p(x_t \mid x_{<t}, \boldsymbol{c})$ is assumed to belong to a parametric family $p_\theta(\cdot)$, whose parameters (e.g., a mean, variance, or a set of categorical probabilities) are predicted by a neural network f_θ as:

$$p(x_t \mid x_{<t}, \boldsymbol{c}) = p_\theta\big(x_t \mid f_\theta(x_{<t}, \boldsymbol{c})\big). \tag{3.2}$$

In other words, the neural network takes the past samples and conditioning features as input and outputs the parameters of the assumed conditional distribution at each time step. By learning the sequence of conditional distributions, the autoregressive model can generate the entire waveform through sequentially sampling each value.

3.2.2 Architectures

Since the emergence of WaveNet [4], neural vocoders have undergone rapid and diverse development. WaveNet demonstrated that deep neural networks can generate high-quality speech waveforms, revealing the potential of the fully data-driven approach. This breakthrough established the foundation for subsequent neural vocoders,

including both autoregressive and non-autoregressive models. [1]

WaveNet is built on dilated causal convolutions and is widely recognized as the first neural architecture capable of generating raw waveforms with high perceptual quality. Dilated convolution expands the receptive field by inserting gaps between kernel elements, enabling the model to capture long-range temporal dependencies efficiently. For a layer l with dilation rate $d_l = 2^{l-1}$, the receptive field across L stacked layers is expressed as

$$R = 1 + \sum_{l=1}^{L}(k_l - 1)\prod_{i=1}^{l-1} d_i, \tag{3.3}$$

where k_l denotes the kernel size at layer l. Through this exponentially growing receptive field, WaveNet can condition each generated sample on thousands of past values, effectively modeling long-term temporal structure.

Each layer employs a gated activation unit:

$$\boldsymbol{x}' = \tanh(W_f * \boldsymbol{x}) \odot \sigma(W_g * \boldsymbol{x}), \tag{3.4}$$

where $\odot$ denotes element-wise multiplication. External conditioning features, such as acoustic or linguistic representations, are added as bias terms in each layer, realizing the conditional generation process $p(x_t \mid x_{<t}, \boldsymbol{c})$. This structural component, often referred to as the WaveNet block, became a foundational design unit for subsequent vocoders [8, 48, 49].

Although WaveNet achieved high-fidelity waveform generation, its strictly sequential sampling process results in extremely slow inference, limiting its practicality. This

[1] Although the original WaveNet [4] was introduced as a TTS model conditioned on linguistic and acoustic features, it was later adapted as a vocoder, referred to as the WaveNet vocoder [5, 6], which synthesizes speech from acoustic features predicted by a separate acoustic model.

limitation motivated the development of a variety of more efficient autoregressive architectures aimed at improving generation speed. Among these efforts, models such as WaveRNN [50] and FFTNet [51] achieve real-time synthesis even in CPU-only environments while substantially reducing computational cost.

WaveRNN improves efficiency by replacing WaveNet's deeply stacked dilated convolutions with a compact recurrent neural network (RNN). Its architecture is built upon lightweight gated recurrent units that take the previous sample and conditioning features as inputs, capturing long-term dependencies through recurrent state updates rather than through an expanded convolutional receptive field. Furthermore, WaveRNN introduces a dual softmax output scheme that decomposes each 16-bit sample into high- and low-8-bit components, avoiding the large-scale softmax operations required for full-resolution waveform prediction.

On the other hand, FFTNet employs a different strategy, drawing inspiration from the divide-and-conquer structure of the fast Fourier transform (FFT). Its layers form a reversed binary-tree configuration in which each block combines two input streams corresponding to temporal positions separated by powers of two. Using simple 1×1 convolutions, each layer aggregates information from increasingly distant past samples, allowing the receptive field to expand exponentially with depth. This design provides an efficient alternative to dilated convolutions, capturing long-range dependencies with shallow convolutional layers.

3.2.3 Output Distribution Modeling

In many autoregressive vocoders, the output distribution of each waveform sample is modeled as a categorical distribution, following the design choice popularized by WaveNet. WaveNet applies μ-law quantization with $\mu = 255$ to transform each

continuous sample into a discrete label:

$$y_t = \text{Quantize}\left(\frac{\text{sign}(x_t)\ln(1+\mu|x_t|)}{\ln(1+\mu)}\right), \quad \mu = 255. \tag{3.5}$$

The μ-law function compresses the amplitude dynamic range, allocating denser quantization levels to low-amplitude regions where human auditory sensitivity is higher. This strategy reduces perceptually salient quantization errors, allowing an effective categorical representation with a manageable number of classes.

The neural network outputs the logits over these quantized labels:

$$p(x_t \mid x_{<t}) = \text{Categorical}(y_t; \boldsymbol{p}_t), \quad \boldsymbol{p}_t = \text{Softmax}(f_\theta(x_{<t})), \tag{3.6}$$

and the model parameters are optimized via cross-entropy:

$$\mathcal{L}_{\text{CE}} = -\sum_t \log p(x_t = y_t \mid x_{<t}). \tag{3.7}$$

In contrast, WaveRNN introduced a more efficient and expressive output representation, the dual softmax formulation. The model first predicts the distribution of the high bits $p(x_t^{(h)} \mid x_{<t})$, and then, conditioned on that prediction, estimates the distribution of the low bits $p(x_t^{(l)} \mid x_t^{(h)}, x_{<t})$:

$$p(x_t \mid x_{<t}) = p(x_t^{(h)} \mid x_{<t})\, p(x_t^{(l)} \mid x_t^{(h)}, x_{<t}). \tag{3.8}$$

This two-stage hierarchical parameterization preserves the expressiveness required for high-resolution waveform modeling while reducing the computational cost.

In addition to such discrete formulations, some autoregressive models employ continuous output distributions. By modeling waveform samples as continuous probability densities rather than quantized μ-law labels, these models reduce quantization artifacts

and enable more precise waveform reconstruction. Continuous output distributions are used in several autoregressive vocoders [16, 22] and also serve as teacher distributions in distillation-based neural vocoders such as [52, 53], discussed in Section 3.3.

3.2.4 Limitations and Challenges

Autoregressive models inherently rely on a strictly sequential generation process, which requires running the network once for every output sample. Because each sample depends on all previously generated ones, the model cannot exploit parallel computation during inference. Although several architectures have been proposed to improve efficiency and enable real-time synthesis on practical hardware, the need for sample-by-sample generation remains an inherent bottleneck, making autoregressive vocoders fundamentally inefficient for large-scale waveform generation.

A second limitation arises during training. Autoregressive models are typically optimized using teacher forcing, where the ground-truth sample is used as input when predicting the next time step. This stabilizes learning and allows full parallelization during training, since all time steps can be computed from the ground-truth sequence. However, it also induces a mismatch between training and inference conditions, known as exposure bias [54], because the model must rely on its own predictions during generation. This discrepancy often causes error accumulation and degradation of the synthesized waveform.

In summary, while autoregressive vocoders can produce high-fidelity speech waveforms, their reliance on sequential sampling and the degradation caused by exposure bias pose significant challenges for practical deployment. To overcome these limitations, subsequent research has explored non-autoregressive generative models, including normalizing flows and GANs.

3.3 Normalizing Flow

3.3.1 General Formulation

Normalizing flows transform samples $\boldsymbol{x}$ drawn from the data distribution $p_{\text{data}}(\boldsymbol{x})$ into latent variables $\boldsymbol{z} = f_\theta(\boldsymbol{x})$ through an invertible mapping $f_\theta = f_{1,\theta_1} \circ \cdots \circ f_{K,\theta_K}$. Each component transformation f_{k,θ_k} is designed to be bijective, ensuring that the overall inverse mapping

$$\boldsymbol{x} = f_\theta^{-1}(\boldsymbol{z}) = f_{K,\theta_K}^{-1} \circ \cdots \circ f_{1,\theta_1}^{-1}(\boldsymbol{z}) \tag{3.9}$$

is explicitly defined and provides a one-to-one correspondence between the data space and the latent space. In practice, each transformation f_{k,θ_k} is implemented via a neural network parameterized by θ_k.

The invertibility of f_θ allows the probability density in the data space to be computed exactly via the change-of-variables formula. Letting $\boldsymbol{x}_K = \boldsymbol{x}$ and $\boldsymbol{x}_{k-1} = f_{k,\theta_k}(\boldsymbol{x}_k)$ denote the intermediate variables at each layer, the resulting latent variable is $\boldsymbol{z} = \boldsymbol{x}_0$, and the data density $p_{\boldsymbol{x}}(\boldsymbol{x})$ is given by

$$p_{\boldsymbol{x}}(\boldsymbol{x}) = p_{\boldsymbol{z}}(\boldsymbol{z}) \prod_{k=1}^{K} \left| \det \frac{\partial f_{k,\theta_k}(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_k} \right|. \tag{3.10}$$

Here, the Jacobian determinant captures the local expansion or contraction of volume induced by each transformation. Since the latent distribution $p_{\boldsymbol{z}}(\boldsymbol{z})$ is typically chosen to be analytically tractable, such as a standard normal distribution, the model can describe complex data distributions without requiring an explicit parametric form for p_{data}.

Training is performed by maximizing the log-likelihood of the model distribution,

$$\log p_{\boldsymbol{x}}(\boldsymbol{x}) = \log p_{\boldsymbol{z}}(\boldsymbol{z}) + \sum_{k=1}^{K} \log \left| \det \frac{\partial f_{k,\theta_k}(\boldsymbol{x}_k)}{\partial \boldsymbol{x}_k} \right|, \tag{3.11}$$

which is equivalent to minimizing the Kullback-Leibler divergence between the model distribution $p_{\boldsymbol{x}}$ and the data distribution p_{data}.

On the other hand, computing Jacobian determinants generally requires $O(D^3)$ time for a waveform of dimensionality D, making direct application to high-dimensional data impractical. To overcome this limitation, practical flow models employ transformations, such as coupling layers and autoregressive transformations, discussed in the next sections, whose Jacobian matrices have a triangular structure, thereby reducing the computational cost of the determinant to $O(D)$.

However, computing Jacobian determinants naively requires $O(D^3)$ operations, making direct application to high-dimensional data, such as speech, impractical. To resolve this issue, practical flow architectures employ specially structured transformations, such as coupling layers and autoregressive transformations, whose Jacobian matrices are triangular, thereby reducing the computational cost of the determinant to $O(D)$. Moreover, when the invertible transformations are designed without autoregressive dependencies, the forward mapping $\boldsymbol{x} = f_\theta(\boldsymbol{z})$ becomes fully parallelizable, enabling fast parallel waveform generation in contrast to autoregressive models.

3.3.2 Coupling-Based Models

A major family of flow-based neural vocoders is built upon coupling-layer architectures, exemplified by FloWaveNet [55] and WaveGlow [48]. These models adapt the Real NVP [56] and Glow [57] frameworks to waveform generation, benefiting from their

parallelizable forward and inverse transformations.

A coupling layer splits the input $\boldsymbol{x}$ into two parts, $\boldsymbol{x} = (\boldsymbol{x}_a, \boldsymbol{x}_b)$, and transforms only one part while keeping the other unchanged. For an affine coupling layer, the forward transformation is defined as

$$\boldsymbol{y}_a = \boldsymbol{x}_a, \quad \boldsymbol{y}_b = \boldsymbol{x}_b \odot \exp\big(s_\theta(\boldsymbol{x}_a)\big) + t_\theta(\boldsymbol{x}_a), \tag{3.12}$$

where $s_\theta(\cdot)$ and $t_\theta(\cdot)$ are neural networks that output elementwise scale and shift parameters. Since the Jacobian of this transformation is triangular, its log-determinant reduces to

$$\log \left| \det \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} \right| = \sum_i s_{\theta,i}(\boldsymbol{x}_a), \tag{3.13}$$

ensuring that likelihood computation remains tractable. This structural property enables exact maximum-likelihood training and allows waveform samples to be generated fully in parallel.

However, flow-based vocoders must impose structural constraints on each invertible transformation to ensure that the Jacobian determinant remains tractable. As a result, many layers must be stacked to model the complexity of speech waveforms, and architectures such as FloWaveNet and WaveGlow typically require a substantially larger number of parameters, over several times more than WaveNet, leading to increased model size and computational cost.

These models also employ a squeeze operation that reshapes the one-dimensional audio signal into multiple channels by folding the temporal dimension. This rearrangement groups adjacent samples into jointly processed channels, allowing coupling layers and convolutional blocks to expand their effective receptive fields more efficiently.

However, the operation can introduce artifacts, known as fixed-frequency noise, which degrade the perceptual quality of the generated speech.

To address the trade-off between model expressiveness and computational efficiency in flow-based vocoders, WaveFlow [58] proposes a compact flow-based architecture that provides a unified view of autoregressive and bipartite flows. After applying a squeeze operation to reshape the one-dimensional waveform into a two-dimensional matrix, WaveFlow defines an autoregressive flow along the channel dimension, using 2D dilated convolutions. In this formulation, likelihood evaluation remains fully parallel, whereas synthesis requires only a fixed number of sequential steps equal to the number of channels. By adjusting this channel dimension, WaveFlow explicitly trades off sampling parallelism against model capacity, introducing sequential dependency as a tunable design dimension for balancing expressiveness, parameter size, and inference speed.

3.3.3 Distillation-Based Models

Another approach to applying normalizing flows to neural vocoders is based on the inverse autoregressive flow (IAF) [59], which underlies models such as Parallel WaveNet [52] and ClariNet [53]. IAF models enable fully parallel generation and, unlike coupling-layer flows, are not constrained by a fixed partition of the input, allowing more flexible transformations. However, likelihood-based training remains inherently sequential due to the structural duality between autoregressive and IAF models: autoregressive models support parallel likelihood computation but generate sequentially, whereas IAF models generate in parallel but require sequential computation for likelihood evaluation.

To overcome this generation-training trade-off, Parallel WaveNet (PWN) [52] and ClariNet [53] adopt a two-stage training strategy known as probability density distillation. In this framework, a high-capacity autoregressive teacher model is first trained,

and a non-autoregressive student model then learns to approximate the teacher distribution through distillation. In PWN, the autoregressive teacher uses a mixture of logistic distributions to model 16-bit waveform samples with high fidelity, whereas the student is implemented as an IAF model with a single logistic distribution to reduce computational overhead. ClariNet, in contrast, employs a single Gaussian output distribution for both the teacher and the student, enabling a closed-form Kullback-Leibler (KL) divergence and thereby simplifying the distillation procedure. The distillation objective is formulated as

$$\mathcal{L}_{\text{distill}} = D_{\text{KL}}\left(p_{\text{s}}(\boldsymbol{x}) \parallel p_{\text{t}}(\boldsymbol{x})\right), \tag{3.14}$$

which encourages the student to reproduce the teacher's predictive distribution, enabling fully parallel generation while achieving speech quality comparable to that of autoregressive models.

This two-stage procedure, however, introduces several practical challenges. First, training requires maintaining both the teacher and student models simultaneously, which increases memory usage and computational cost compared with single-model training. Second, because the distillation loss minimizes the reverse Kullback-Leibler divergence (Eq. 3.14), the student model tends to underestimate the variability of the teacher distribution, a phenomenon often associated with mode-collapse. In practice, this can lead to degraded perceptual quality, and prior studies have reported the need for additional spectral or statistical loss terms to stabilize training.

3.3.4 CNF-Based Models

To alleviate the structural constraints and computational burden present in discrete normalizing flows, the continuous normalizing flows (CNFs) framework was introduced by neural ordinary differential equations (ODEs) [60]. CNFs generalize a sequence of invertible transformations into a continuous-time ODE, and the generative process is defined as

$$\frac{d\boldsymbol{z}(t)}{dt} = f_\theta(\boldsymbol{z}(t), t), \quad \boldsymbol{z}(0) \sim p_{\boldsymbol{z}}(\boldsymbol{z}). \tag{3.15}$$

The evolution of the log probability density is given by

$$\frac{d \log p(\boldsymbol{z}(t))}{dt} = -\mathrm{Tr}\left(\frac{\partial f_\theta(\boldsymbol{z}(t), t)}{\partial \boldsymbol{z}(t)}\right). \tag{3.16}$$

This formulation treats the flow transformations as an accumulation of infinitesimal invertible mappings in continuous time and can be interpreted as an ODE governing the evolution of probability density. In CNFs, the neural network defines a continuous-time transformation field, removing the explicit Jacobian computations and invertibility constraints of discrete flows. However, computing the trace of the Jacobian remains costly in high-dimensional spaces. Neural ODEs still require $\mathcal{O}(D^2)$ operations, and free-form Jacobian of reversible dynamics (FFJORD) employs the Hutchinson estimator to approximate this trace stochastically, reducing the complexity to $\mathcal{O}(D)$.

Training in CNFs involves solving the ODE backward in time to evaluate likelihoods. During generation, the model integrates forward from $\boldsymbol{z}$ to $\boldsymbol{x}$. During training, it integrates backward from the observation $\boldsymbol{x}$ to the latent variable $\boldsymbol{z}$. This guarantees a unique inverse mapping provided that the vector field f_θ satisfies mild smoothness conditions such as Lipschitz continuity.

Building on the CNF formulation, models such as WaveNODE [61] and WaveF-FJORD [62] adapt continuous-time flows to neural vocoders. By removing the structural constraints imposed on transformation functions in discrete flows, CNF-based vocoders can employ much more flexible mappings, which enables high-fidelity waveform generation with substantially fewer parameters than the coupling-based architectures. WaveNODE achieves speech quality comparable to FloWaveNet and WaveGlow, while using significantly fewer parameters. This demonstrates that continuous time formulations improve the trade-off between model size and speech quality.

3.3.5 Limitations and Challenges

Normalizing flow models provide a powerful generative framework that combines a rigorous likelihood-based training objective with an inference structure capable of fully parallel generation. However, the structural constraints imposed on each layer to ensure invertibility and the computational cost of evaluating Jacobian determinants for density estimation significantly limit the expressive capacity of these models. As a result, a clear trade-off arises among generation quality, model size, and computational efficiency, which remains a major obstacle to achieving high-quality and efficient speech synthesis.

Continuous Normalizing Flow (CNF) was introduced to relax the structural and computational constraints inherent in discrete normalizing flows. By formulating discrete invertible transformations as continuous-time ordinary differential equations, CNFs provide substantially greater architectural flexibility. CNFs have also served as a theoretical foundation for the more recent flow matching framework, establishing an important conceptual connection between normalizing flows and probability flows. Despite these strengths, CNFs still suffer from practical challenges, including the computational

overhead of numerical ODE integration as well as stability and convergence issues during training. Consequently, in practical applications, CNFs are often less favorable than the implicit generative models discussed in the following section.

3.4 Generative Adversarial Networks

3.4.1 General Formulation

A generative adversarial network (GAN) consists of two neural networks, a generator $\mathcal{G}$ and a discriminator $\mathcal{D}$, which are trained in an adversarial manner. The generator aims to produce outputs that resemble real data, while the discriminator learns to distinguish between natural samples drawn from the true distribution $p_{\text{data}}(\boldsymbol{x})$ and synthetic samples generated from the model distribution $p_{\mathcal{G}}(\boldsymbol{x})$.

The standard minimax GAN objective is formulated as

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\log \mathcal{D}(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\log\big(1 - \mathcal{D}\big(\mathcal{G}(\boldsymbol{z})\big)\big)], \tag{3.17}$$

where $\boldsymbol{z}$ is a sample drawn from a latent prior distribution (typically a multivariate Gaussian). Under this formulation, if $\mathcal{D}$ is optimized to its theoretical optimum and has sufficient expressive capacity, the resulting objective for $\mathcal{G}$ is equivalent to minimizing the Jensen-Shannon divergence between p_{data} and the model distribution. Because GANs define an implicit probabilistic model, one that generates samples without requiring evaluation of a normalized probability density, they enable the approximation of complex data distributions without relying on explicit likelihood functions or tractable probability densities.

For neural vocoder applications, conditional GANs provide an effective framework

for generating high-quality waveforms from acoustic features. In this setup, the generator receives a random vector $\boldsymbol{z}$ and a conditioning feature $\boldsymbol{c}$ (e.g., a mel-spectrogram), producing a waveform $\hat{\boldsymbol{x}} = \mathcal{G}(\boldsymbol{z}, \boldsymbol{c})$. The discriminator attempts to distinguish generated waveforms $\hat{\boldsymbol{x}}$ from natural waveforms $\boldsymbol{x}$, with optional conditioning on auxiliary features $\boldsymbol{c}$, depending on the specific vocoder design. In many practical vocoders, the explicit noise input is omitted, and the generator learn a deterministic mapping $\hat{\boldsymbol{x}} = \mathcal{G}(\boldsymbol{c})$. For notational convenience, we denote all generator inputs (e.g., $\boldsymbol{z}$, $\boldsymbol{c}$) collectively as $\boldsymbol{z}$ in the following sections. The subsequent sections describe the key components of GAN-based vocoders, including loss functions, generator architectures, and discriminator designs.

3.4.2 Training Objective

Since GANs often exhibited unstable behaviors such as mode collapse and vanishing gradients, GAN-based vocoders typically stabilize training by combining multiple loss terms. The generator is optimized not only with the adversarial loss, but also with auxiliary losses such as feature matching loss $\mathcal{L}_{\text{fm}}$ and spectral reconstruction loss $\mathcal{L}_{\text{spec}}$, which improve stability and perceptual quality. The overall generator objective is generally formulated as

$$\mathcal{L}_{\mathcal{G}} = \lambda_{\text{adv}}\mathcal{L}_{\text{adv}} + \lambda_{\text{fm}}\mathcal{L}_{\text{fm}} + \lambda_{\text{spec}}\mathcal{L}_{\text{spec}}, \tag{3.18}$$

where each weighting factor λ_* controls the contribution of the associated loss term.

GAN-based vocoders generally do not use the original cross-entropy minimax formulation (Eq.3.17). Instead, they adopt alternative adversarial objectives such as the hinge GAN [63] or least-squares GAN (LS-GAN) [64] losses, which alleviate gradient

saturation and improve training stability. The LS-GAN loss is defined as

$$\mathcal{L}_{\mathcal{D}} = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[(\mathcal{D}(\boldsymbol{x}) - 1)^2] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\mathcal{D}(\mathcal{G}(\boldsymbol{z}))^2], \tag{3.19}$$

$$\mathcal{L}_{\mathcal{G}} = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[(\mathcal{D}(\mathcal{G}(\boldsymbol{z})) - 1)^2], \tag{3.20}$$

while the hinge formulation is given by

$$\mathcal{L}_{\mathcal{D}} = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}[\max(0, 1 - \mathcal{D}(\boldsymbol{x}))] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\max(0, 1 + \mathcal{D}(\mathcal{G}(\boldsymbol{z})))], \tag{3.21}$$

$$\mathcal{L}_{\mathcal{G}} = \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[-\mathcal{D}(\mathcal{G}(\boldsymbol{z}))]. \tag{3.22}$$

The feature matching loss compares intermediate activations of the discriminator between real and generated waveforms. To further stabilize learning, the feature matching loss compares intermediate discriminator activations for natural and generated waveforms:

$$\mathcal{L}_{\text{fm}} = \mathbb{E}\left[\sum_i \frac{1}{N_i}\|\mathcal{D}^{(i)}(\boldsymbol{x}) - \mathcal{D}^{(i)}(\mathcal{G}(\boldsymbol{z}))\|_1\right], \tag{3.23}$$

where $\mathcal{D}^{(i)}(\cdot)$ denotes the output of the i-th discriminator layer and N_i is the number of elements in that layer. By matching the discriminator's intermediate feature representations for real and generated waveforms, the generator is encouraged to produce signals whose characteristics resemble those of natural speech.

To enforce consistency between the generated waveform $\hat{\boldsymbol{x}}$ and the target waveform

$\boldsymbol{x}$ in both temporal and spectral domains, GAN-based vocoders often incorporate a multi-resolution short-time Fourier transform (STFT) loss. This loss captures fine- and coarse-grained spectral structures as well as partial phase information. Following the formulation used in Parallel WaveGAN (PWG) [8], the multi-resolution STFT loss is defined as the average of spectral losses computed with M different sets of STFT analysis parameters (e.g., window length, FFT points, and frame shift):

$$\mathcal{L}_{\text{MR-STFT}} = \frac{1}{M} \sum_{m=1}^{M} \mathcal{L}_{\text{STFT}}^{(m)}, \tag{3.24}$$

$$\mathcal{L}_{\text{STFT}} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{z}} \left[\mathcal{L}_{\text{sc}}(\boldsymbol{x}, \mathcal{G}(\boldsymbol{z})) + \mathcal{L}_{\text{mag}}(\boldsymbol{x}, \mathcal{G}(\boldsymbol{z})) \right], \tag{3.25}$$

where $\mathcal{L}_{\text{sc}}^{(m)}$ and $\mathcal{L}_{\text{mag}}^{(m)}$ denote the spectral convergence and log-magnitude losses at resolution m, respectively:

$$\mathcal{L}_{\text{sc}}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{\| \, |\text{STFT}(\boldsymbol{x})| - |\text{STFT}(\hat{\boldsymbol{x}})| \, \|_{\text{F}}}{\| \, |\text{STFT}(\boldsymbol{x})| \, \|_{\text{F}}}, \tag{3.26}$$

$$\mathcal{L}_{\text{mag}}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{N} \| \log |\text{STFT}(\boldsymbol{x})| - \log |\text{STFT}(\hat{\boldsymbol{x}})| \, \|_1, \tag{3.27}$$

where $|| \cdot ||_{\text{F}}$ is the Frobenius norm, $|\,\text{STFT}(\cdot)\,|$ denotes the STFT magnitude, and N is the number of elements in the spectrogram. Because STFT resolution depends on analysis parameters, employing multiple resolutions enables the model to capture diverse spectral patterns and prevents it from overfitting to a particular time-frequency scale.

A widely used variant of spectral loss is the mel-spectrogram loss, which evaluates

discrepancies in a perceptually motivated frequency domain. This loss measures the L1 distance between mel-spectrograms converted from real and synthesized waveforms:

$$\mathcal{L}_{\text{mel}} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{z}} \left[\frac{1}{N} \|\mathcal{M}(\boldsymbol{x}) - \mathcal{M}(\mathcal{G}(\boldsymbol{z}))\|_1 \right], \tag{3.28}$$

where $\mathcal{M}(\cdot)$ denotes the mel transform and N is the number of mel bins. The mel-spectrogram provides a frequency-warped representation that reflects key characteristics of human auditory perception. Humans exhibit high frequency resolution at low frequencies, where formants and other perceptually salient speech structures reside, but substantially lower resolution at higher frequencies. By applying a bank of triangular filters that widen with increasing frequency, the mel transform performs a weighted integration of spectral magnitudes over broader high-frequency bands. This reduces the influence of stochastic high-frequency components, whose fine-grained variations are inherently difficult to predict. Consequently, this loss provides a coarse but perceptually meaningful objective, allowing the adversarial and feature matching losses to focus on modeling the fine-grained details.

3.4.3 Generator

The generator in a GAN-based vocoder directly converts conditional features into a time-domain waveform. Unlike autoregressive models that generate samples sequentially, GANs produce the entire waveform in a single forward pass. This one-shot generation capability offers significant advantages for large-scale parallel synthesis, real-time speech generation, and streaming applications. Another important property is that GANs do not require an explicit likelihood formulation, which frees them from many

structural constraints imposed on models such as normalizing flows and enables more flexible and expressive network designs. GAN generators also differ from other probabilistic generative models in that they do not rely on a latent variable $\boldsymbol{z}$ as the basis of the generative process. The latent prior $p(\boldsymbol{z})$ is a core assumption in models such as variational autoencoders [65] and normalizing flows. Because these models depend on this assumed prior, any mismatch between the assumed prior and the true underlying data structure can degrade synthesis quality. In contrast, GANs learn to approximate the data distribution through adversarial training without requiring explicit likelihood estimation or the specification of any latent prior. In practical neural vocoders based on conditional GANs, the role of the latent vector becomes even more limited. Although a random vector is often injected into the generator, it serves primarily to introduce sample-level stochastic variation rather than to represent a structured latent space, as the conditioning features almost completely determine the output.

GAN-based vocoders employ generator architectures constructed primarily from dilated convolutional neural networks, which enable fully parallel waveform generation. Several designs have become the foundation for many subsequent models, including Parallel WaveGAN, MelGAN, and HiFi-GAN. Parallel WaveGAN adopts a non-autoregressive WaveNet-style generator consisting of stacks of residual blocks that use dilated convolutions and gated activation units, as described in Section 3.2.2. The generator takes both a noise signal with the same shape of the output waveform and conditioning acoustic features as inputs. The noise provides sample-level stochastic variation, and the conditioning features are injected into each residual block to guide the waveform generation. Because GAN generators do not impose the architectural constraints unlike normalizing flows, PWG employs a much lighter generator and achieves faster waveform generation while maintaining competitive speech quality

with IAF-based model [53]. MelGAN employs a hierarchical upsampling generator that maps acoustic features directly to a speech waveform without using an explicit noise input. The conditioning features are progressively upsampled through a sequence of transposed-convolution layers, each followed by convolutional blocks that refine the intermediate representations. Because the generator operates at lower temporal resolutions in the earlier layers and only reaches the full waveform resolution at the final stage, it can model long-range temporal structure efficiently while avoiding expensive convolutions at the raw-waveform sampling rate. This makes forward computation more efficient than PWG generators. HiFi-GAN further advances the hierarchical upsampling approach of MelGAN while sharing its basic design principle of generating the waveform directly from acoustic features without using an explicit noise input. Similar to MelGAN, the conditioning features are progressively upsampled through a series of transposed-convolution stages to reach the waveform sampling rate. However, HiFi-GAN significantly refines this architecture by introducing a multi-receptive-field (MRF) module after each upsampling stage. The MRF consists of several residual blocks with different kernel sizes and dilation rates operating in parallel, and their outputs are combined across multiple temporal resolutions.

3.4.4 Discriminator

The discriminator is responsible for distinguishing generated waveforms from natural ones, thereby guiding the generator during adversarial training. To effectively capture the complex temporal structures of speech, most GAN vocoders employ multiple discriminators that operate on different aspects of the signal and at different levels of granularity.

The multi-scale discriminator (MSD) introduced in MelGAN [66] processes input

waveforms at multiple downsampling rates, allowing the model to evaluate signal characteristics across diverse temporal scales. Each sub-discriminator analyzes a different resolution, ranging from the raw waveform (high resolution) to heavily downsampled versions (low resolution). Coarse scales enable the discriminator to capture global temporal envelopes and energy contours, whereas finer scales focus on high-frequency components and micro-level periodic patterns. Because low-frequency structures such as formants and the fundamental frequency are perceptually dominant, the MSD promotes accurate modeling of these components while preserving the naturalness of high-frequency details. As a result, MSD contributes to enhanced temporal smoothness and perceptually consistent high-frequency rendering.

Building on this idea, HiFi-GAN [67] introduces the multi-period discriminator (MPD), which reshapes the waveform into two-dimensional representations based on predefined periods (period folding) and applies 2D convolutions for discrimination. Each sub-discriminator corresponds to a specific period (e.g., 2, 3, 5, 7, or 11 samples), explicitly modeling diverse pitch periodicities and harmonic structures inherent in speech. This design enables the MPD to learn periodic patterns associated with fundamental frequencies, their harmonics, and their temporal variations. By enforcing periodic consistency through adversarial training, the MPD improves pitch naturalness, waveform continuity, and phase stability. While the MSD enforces macro-level temporal coherence, the MPD complements it by capturing micro-level periodic coherence, resulting in a discriminator set that aligns more closely with human perceptual criteria for natural speech.

More recently, discriminators utilizing time-frequency representations have been proposed. A representative example is the multi-resolution spectrogram discriminator (MRD) introduced in UnivNet [68]. The MRD computes multiple spectrograms from

the same waveform using different STFT configurations (varying FFT sizes, window lengths, and hop sizes), and each spectrogram is processed by a dedicated sub-discriminator. This enables adversarial learning of spectral consistency across multiple time-frequency resolutions, effectively mitigating high-frequency noise and excessive spectral smoothing in generated audio. Unlike waveform-domain discriminators, the MRD directly operates in the spectrogram domain and thus serves as a complementary extension to MSD and MPD.

These discriminator designs have consistently produced high perceptual quality across a range of generator architectures [69]. This robustness arises because discriminators extract perceptually salient speech features and convey them to the generator through adversarial and feature-matching losses. Consequently, the combination of MSD, MPD, and MRD has become the de facto standard discriminator configuration in modern GAN-based speech synthesis systems.

3.4.5 Limitations and Challenges

Compared with other generative models, GAN-based vocoders are particularly effective at achieving both high-quality and real-time speech synthesis, leading them to become widely adopted in modern speech generation systems. However, they still present several limitations. First, they often suffer from limited diversity in the generated samples. This issue arises because the conditioning features provided to the generator are high-dimensional and highly structured, which causes the model to ignore the stochastic noise input and instead learn an almost deterministic mapping [70]. Second, GAN-based vocoders tend to lack the speech controllability that traditional signal-processing-based vocoders [2,3] often provide. Their internal generative process, learned in a purely data-driven manner, does not explicitly account for the physical

mechanisms of speech production. From this perspective, subsequent studies, discussed in Chapter 4, have explored incorporating prior knowledge about the speech production mechanism to enhance controllability, as well as improving computational efficiency.

3.5 Evaluation of Vocoders

This section discusses the key perspectives for evaluating the performance of vocoders. Although the specific requirements vary depending on the application and operating environment, vocoders are generally assessed along four primary dimensions: naturalness, controllability, computational complexity, and lightweightness. These dimensions are not independent and often exhibit trade-off relationships. Therefore, the goal of vocoder design is not to optimize any single criterion, but to achieve a well-balanced improvement across all of them.

3.5.1 Naturalness

Naturalness refers to the perceptual quality of synthesized speech, in particular, whether it sounds smooth, easy to understand, and free of artificial or metallic noises. Subjective listening tests are the standard evaluation method. For instance, metrics such as the mean opinion score (MOS) represent overall perceived quality, and ABX tests assess how easily synthesized speech can be distinguished from natural reference samples. In addition to subjective evaluation, objective metrics based on acoustic or perceptual distance have also been widely used. Traditional measures such as perceptual evaluation of speech quality (PESQ) and short-time objective intelligibility (STOI) were originally designed for speech enhancement and telecommunication codecs, and although they are not ideally suited for speech synthesis, they have been used in

some vocoder evaluation studies. More recently, perceptual metrics derived from self-supervised speech representations have gained attention. For example, UTMOS [71] predicts MOS by leveraging self-supervised speech representations [72] together with features from an automatic speech recognition model [73], and its predictions have been shown to correlate well with human MOS ratings. Naturalness is a crucial prerequisite for the social acceptability of speech synthesis systems. In applications such as call center automation or conversational agents, it plays a central role in ensuring a comfortable and stress-free user experience.

3.5.2 Controllability

Controllability in vocoders refers to how accurately they can reproduce speech attributes, such as pitch, timbre, and breathiness, based on the acoustic features provided by the upstream acoustic model. Typically, the vocoder itself does not directly manipulate these attributes; instead, it converts predicted features such as F_0, spectral features, and speaker-identical representations into time-domain waveforms. From this perspective, controllability measures how faithfully the generated waveform reflects the intended acoustic characteristics encoded in the input features. Accordingly, objective metrics grounded in physical quantities are commonly used. Measures such as voiced-unvoiced (VUV) error rate and F_0 root mean square error (RMSE) evaluate the accuracy of excitation-related features, including voicing patterns and pitch. In contrast, STFT-based spectral distance and mel-cepstral distortion (MCD) assess the fidelity of spectral characteristics, capturing how accurately timbral attributes and phonetic information encoded in the input features are reconstructed. High controllability is essential for applications requiring precise and expressive vocal rendering, including emotional speech synthesis and singing voice synthesis. Imposing strong structural con-

straints, as in classical source-filter models, can enhance controllability. However, such constraints restrict the range of acoustic phenomena the model can represent, which in turn limits the realism of the generated speech. This creates a trade-off between naturalness and controllability.

3.5.3 Computational Complexity

Computational complexity refers to the amount of arithmetic computation required during speech generation, typically measured in floating-point operations (FLOPs) or multiply-accumulate operations (MACs). Because a speech waveform is a high-dimensional time-domain signal comprising tens of thousands of samples per second, generating it demands substantial computation, and the intrinsic complexity of a model strongly influences its practical deployability. The computational load directly affects both the real-time factor (RTF) and the latency of speech synthesis. RTF represents system throughput and is defined as the processing time required to synthesize one second of audio, where values below one indicate real-time capability. Latency, in contrast, denotes the delay until the model produces the first output sample. Note that latency is also influenced by architectural factors, such as sequential dependencies and the granularity of block processing. The impact of computational cost becomes particularly pronounced on CPUs or other environments with limited parallelism, where higher complexity more strongly constrains throughput. Achieving high perceptual quality while reducing computational cost remains a central challenge, since model expressiveness generally increases with computational complexity.

3.5.4 Lightweightness

Lightweightness refers to the compactness of a vocoder, primarily determined by its model size, which is typically measured by the number of trainable parameters. A smaller parameter count reduces memory consumption and lowers the cost of loading model parameters during inference, which is especially beneficial on devices with limited memory bandwidth or storage capacity. Although compact models are easier to deploy, parameter reduction limits expressive capacity and can degrade naturalness or controllability. Consequently, designing vocoders that remain lightweight while preserving high speech quality is an important challenge. Since speech is increasingly used as a universal interface across a wide variety of devices, from mobile terminals to embedded systems, the demand for lightweight models continues to grow.

3.6 Summary

This chapter presented an overview of the development of neural vocoders from the perspective of probabilistic generative modeling. Neural vocoders learn to represent the speech waveform, or its probability distribution, using deep neural networks conditioned on acoustic features, allowing a data-driven formulation that does not depend on explicitly modeling the speech production mechanism, as in the classical signal-processing vocoders.

Autoregressive models were first introduced as the foundational framework for neural waveform generation. By parameterizing the distribution of each sample conditioned on preceding samples, they achieve high-fidelity synthesis. However, their inherently sequential sampling leads to substantial computational overhead, limiting their practicality in many real-world applications. This drawback has motivated a shift toward

models capable of fully parallel generation.

Normalizing flow models provide exact probabilistic modeling and enable fully parallel waveform generation through bijective transformations. Yet, guaranteeing invertibility and efficient Jacobian computation imposes structural constraints, creating trade-offs among modeling capacity, model size, and the degree of autoregressiveness. To ease these restrictions, continuous normalizing flows reformulate flows as continuous-time ODE-based transformations, providing greater architectural flexibility. Nevertheless, both training and inference require numerical integration, which remains computationally expensive.

GAN-based approaches, in contrast, implicitly approximate data distributions through adversarial learning without relying on explicit likelihoods. This enables lightweight generators that perform one-shot waveform generation, making GAN vocoders generally more deployable in practical speech synthesis systems than vocoders based on other probabilistic models. The introduction of auxiliary losses utilizing target speech signals is essential for training stability, while advances in discriminator design have improved speech quality. The use of auxiliary losses derived from target speech signals is crucial for training stability, while advances in discriminator design have improved speech quality.

This chapter also outlined four key evaluation criteria for neural vocoders: naturalness, controllability, computational complexity, and lightweightness. While each criterion reflects a distinct dimension of real-world performance, they are tightly coupled through inherent trade-offs. Together, they determine the practical usability of a vocoder, and overcoming these trade-offs while improving each aspect remains a central challenge in this research field. In this context, the next chapter focuses on neural vocoder approaches that explicitly account for the physical speech production process,

aiming to integrate prior knowledge with data-driven modeling.

4 Physically Oriented Neural Vocoder

This chapter provides an overview of neural vocoder design approaches that are motivated by the physical mechanisms of speech production. This chapter is organized into several complementary perspectives. Section 4.2 introduces neural vocoders that incorporate architectures inspired by the harmonic-plus-noise model. Section 4.3 describes approaches that integrate linear filter models motivated by the source–filter theory. Section 4.4 focuses on F_0-driven mechanisms designed to effectively represent the periodic characteristics of speech signals. Section 4.5 reviews studies that analyze and mitigate distortions arising from the discrete-time nature of digital signal processing. Section 4.6 discusses time–frequency domain modeling approaches that directly learn and generate spectral representations of speech. Finally, Section 4.7 summarizes this chapter.

4.1 Overview

Most neural vocoders do not explicitly account for the physical process of speech production. Instead, they are typically designed as black-box generative systems based on deep neural networks, which often results in limited interpretability and controllability. To address these limitations, a growing body of research has explored neural vocoder architectures inspired by the physical mechanisms of speech production.

This chapter reviews major research directions in neural vocoder design from the perspective of their relationship to the physical speech production mechanism. Figure 4.1 provides an overview of this perspective by summarizing a classification of neural vocoders based on the source–filter modeling framework. The figure organizes existing methods according to their excitation modeling strategies and vocal-tract filtering schemes, and illustrates how representative vocoders, including the proposed methods in this thesis, are positioned within this framework. This classification provides a roadmap for both prior studies and the proposed methods presented in the subsequent chapters of this thesis.

4.2 Harmonic-plus-Noise Model

Speech signals can be represented as a superposition of two components: a periodic component generated by vocal fold vibrations and an aperiodic component produced by turbulent or frictional noise. The harmonic-plus-noise (HN) model [74] explicitly encodes this prior knowledge of the speech production mechanism. In this model, the speech waveform $\boldsymbol{x} \in \mathbb{R}^T$, where T denotes the number of samples, is decomposed into a harmonic component $\boldsymbol{x}^{(\mathrm{h})}$ and a noise component $\boldsymbol{x}^{(\mathrm{n})}$:

$$\boldsymbol{x} = \boldsymbol{x}^{(\mathrm{h})} + \boldsymbol{x}^{(\mathrm{n})}. \tag{4.1}$$

The HN model has long been used in traditional speech analysis-synthesis systems and has also been adopted in neural vocoders.

An HN neural vocoder embeds this decomposition within a neural network framework, in which the periodic and aperiodic components are generated by dedicated

		Parametric modeling	NN modeling	
			AR	Non-AR
Linear	AR		GlotNet, ExcitNet, LPCNet, iLPCNet, LP-WaveNet <Sec. 4.2.2>	QCP-DNN, GlottDNN, GlotGAN <Sec. 4.2.2>
	AR (NN-driven)	GOLF <Sec. 4.2.3>	MWDLP, E2E-LPCNet <Sec. 4.2.3>	
	NAR	STRAIGHT, WORLD <Chap. 2>		GELP, ExcitGlow <Sec. 4.2.2>
	NAR (NN-driven)	DDSP, SawSing <Sec. 4.2.4>		NHV, FIRNet NITECH-E2E-TTS <Sec. 4.2.4>
Nonlinear	Time domain	NSF <Sec. 4.3.2>		uSFGAN, SiFi-GAN <Chap. 5 & 6>
	Time-freq. domain	Wavehax <Chap. 7>		HiFTNet <Sec. 4.5.3>

Figure 4.1: *Classification of vocoders based on the source-filter modeling framework. The columns correspond to source excitation modeling, while the rows correspond to vocal-tract filter modeling. Unlike the description in Section 4.3, the linear filters are categorized by autoregressive (AR) or non-autoregressive (NAR) rather than infinite-impulse-response (IIR) or finite-impulse-response (FIR) filters. The GOLF family [26, 27] behaves in an FIR-like manner and is therefore non-sequential overall. For each representative method, the corresponding section or chapter in this thesis is indicated in the table. The proposed methods in this study are uSFGAN, SiFi-GAN, and Wavehax.*

subnetworks:

$$\boldsymbol{x}^{(\mathrm{h})} = \mathcal{G}^{(\mathrm{h})}(\boldsymbol{z}^{(\mathrm{h})}), \quad \boldsymbol{x}^{(\mathrm{n})} = \mathcal{G}^{(\mathrm{n})}(\boldsymbol{z}^{(\mathrm{n})}), \tag{4.2}$$

where $\boldsymbol{z}^{(\mathrm{h})}$ and $\boldsymbol{z}^{(\mathrm{n})}$ denote acoustic features appropriate for each subnetwork. By explicitly separating the periodic and aperiodic components, the model can naturally incorporate the physical background of human speech production into the network architecture. Furthermore, since each subnetwork specializes in modeling only one component, both periodicity and aperiodicity can be represented with higher precision.

Consequently, HN-based architectures often yield improvements in both speech quality and controllability.

Representative neural vocoders adopting the HN structure include PeriodNet [34], HN-NSF [33], and HN-PWG [35]. Although they share the same fundamental concept, they differ in the way the harmonic and noise components are combined. First, PeriodNet, the simplest structure, where the harmonic and noise components are directly summed in the time domain to generate the final waveform. Second, HN-NSF separately models the harmonic and noise components using low-pass and high-pass filters, respectively, allowing frequency-wise synthesis with physical interpretability. Third, HN-PWG estimates weights dynamically by the neural network for each subband, and a weighted combination is performed across frequency bands, enabling the model to learn frequency-dependent mixing ratios.

The HN architecture is conceptually related to the mixed excitation model in the traditional source-filter framework (Section 2.1.2), in which periodic and noise components are combined to form an excitation signal. However, the HN model does not assume a source-filter structure. Instead, it directly generates the final speech waveform by independently producing and summing the harmonic and noise components.

4.3 Integrating Linear Filters

Revisiting the source-filter theory [7] and its signal-processing implementation offers a valuable perspective for improving neural vocoders. This section provides a systematic review of approaches that integrate linear filters with neural networks, clarifying their motivations and methodologies.

4.3.1 Overview

Human speech production can be effectively described as the convolution of glottal excitation and vocal-tract resonance. The vocal tract acts as a relatively slow, time-varying resonant system that can be well approximated by linear filtering, whereas glottal excitation exhibits rapidly changing and highly nonlinear dynamics. These contrasting behaviors naturally motivate neural vocoders that combine a linear filter with a neural network, allowing the model to learn the remaining nonlinear aspects of speech.

Introducing an explicit linear filter into a neural vocoder provides several advantages. Because the architecture aligns with the source-filter theory, the respective roles of excitation generation and vocal-tract filtering remain interpretable, which improves controllability over the synthesized speech. Moreover, incorporating a linear filter alleviates the burden of modeling vocal-tract characteristics within the neural network itself. This reduction enables the network to be smaller and more computationally efficient.

Two types of linear filters are commonly employed: finite-impulse-response (FIR) filters and infinite-impulse-response (IIR) filters. FIR filters have a nonrecursive and inherently stable structure, making their coefficients straightforward for neural networks to predict. IIR filters, by contrast, use a recursive structure and can represent vocal-tract resonances accurately with far fewer coefficients. Linear prediction [43, 44] is a prominent example of this approach, and its inverse filtering [75] enables direct extraction of the excitation signal, which can serve as teacher data for neural networks.

Linear filters used in neural vocoders are typically implemented as either finite-impulse-response (FIR) or infinite-impulse-response (IIR) filters. FIR filters are nonrecursive and inherently stable, which makes their coefficients straightforward for neural

networks to predict. IIR filters, by contrast, are recursive and can represent vocal-tract resonances accurately with far fewer coefficients than FIR filters. Linear prediction [43, 44] is a prominent example of this approach, and its inverse filtering [75] enables direct extraction of the excitation signal, which can serve as teacher data for neural networks.

Neural networks can interact with the linear filter in several ways. One approach is excitation modeling, in which the network explicitly predicts the excitation signal. Another approach is filter-parameter estimation, in which the network predicts the coefficients that define the linear filter. A hybrid approach combines these two strategies, allowing the network to estimate both the excitation signal and the filter parameters. The following sections review neural vocoder architectures built upon these FIR and IIR frameworks, highlighting their modeling strategies and practical characteristics.

Neural networks can interact with the linear filter in several ways. One approach is excitation modeling, in which the network predicts the excitation signal while an analytical vocal-tract filter is applied separately. Another approach is filter-parameter estimation, in which the network predicts the coefficients of linear filters and employs an analytical excitation model. A third, hybrid approach combines these strategies, allowing the network to estimate both the excitation signal and the filter parameters, including the use of inverse-filtered excitation when available.

Neural networks can interact with linear filters in several ways. The first approach is excitation modeling, where the network predicts only the excitation signal while an analytical vocal-tract filter is applied separately. The second approach is filter-parameter estimation, in which the network predicts the coefficients of linear filters and employs an analytical excitation model. A third, hybrid strategy combines these two approaches, allowing the network to estimate both the excitation signal and the

filter parameters. The following sections review the specific neural vocoders built on these linear filtering frameworks, highlighting their modeling strategies and practical characteristics.

4.3.2 Analytic IIR Filter

In this family of neural vocoders, as in conventional linear predictive coding (LPC), vocal-tract resonances are represented using analytically derived IIR filters. Excitation generation is then handled by either autoregressive or non-autoregressive (NAR) neural networks, allowing flexible and data-driven excitation modeling.

As described in Section 2.1.3, a speech waveform x_t can be decomposed through linear prediction into a predictive (vocal-tract) component and a residual (source-excitation) component:

$$x_t = \sum_{k=1}^{P} a_i x_{t-k} + e_t, \tag{4.3}$$

where P is the number of prediction coefficients, a_k denotes each LPC coefficient, and e_t is the excitation (residual) signal. The vocoders described below incorporate the analytic linear-prediction term $\hat{x}_t = \sum_i a_k x_{t-k}$ into the network input, so that neural networks focus on learning the residual signal $e_t = x_t - \hat{x}_t$. By explicitly separating analytic resonance filtering from data-driven excitation generation, this formulation simplifies the learning objective and leads to improved computational efficiency and reduced parameter count.

Autoregressive Excitation Modeling

Two early representatives of this framework are GlotNet [16] and ExcitNet [76]. Their primary differences lie in the estimation of the vocal-tract filter coefficients and the definition of the excitation output distribution. In GlotNet, the IIR coefficients a_k are estimated using the quasi-closed phase (QCP) method for glottal inverse filtering, which uses a weighted linear prediction that emphasises the glottal closed phase for more accurate vocal-tract estimation. After these coefficients are fixed, the residual (glottal excitation) is modeled by the neural network. The excitation signal $\boldsymbol{e}$ is modeled as a continuous probability density given by a mixture of logistic distributions:

$$p(e_t \mid \boldsymbol{c}) = \sum_{k=1}^{K} \pi_k(\boldsymbol{c}) \,\mathrm{Logistic}\big(e_t \mid \mu_k(\boldsymbol{c}), s_k(\boldsymbol{c})\big), \tag{4.4}$$

where $\boldsymbol{c}$ denotes the conditioning feature including previous excitation samples $e_{<t}$ and acoustic features. ExcitNet, in contrast, employs standard LPC to extract the residual signal e_t, which is represented using 8-bit μ-law quantization and modeled with a categorical softmax distribution, following the original WaveNet [4] design.

LPCNet [21, 23] further extends this line of work with a lightweight variant of the WaveRNN architecture [50]. Instead of relying only on past excitation values, LPCNet computes the LPC-based prediction $\hat{x}_t$, and provides it, together with the previous sample $\hat{x}_t$, the previous excitation e_{t-1} to the network. Together with single categorical distribution modeling on 8-bit μ-law quantization, this RNN-based architecture enables real-time neural waveform generation even on CPU environments.

Building on these foundations, LP-WaveNet [20] and iLPCNet [22] unify the linear prediction with probabilistic generative modeling. Both adopt the linear-prediction mixture density network formulation, where the generative process of speech is modeled

as:

$$p(x_t \mid \boldsymbol{c}) = \sum_k \pi_k(\boldsymbol{c}) \mathcal{N}\left(x_t \,\middle|\, \hat{x}_t + \mu_k(\boldsymbol{c}),\, \sigma_k^2(\boldsymbol{c})\right), \tag{4.5}$$

where $\boldsymbol{c}$ includes previous samples $x_{<t}$ and conditioning acoustic features. Here, the linear prediction term is embedded into the mean of each Gaussian, allowing the neural network to focus on modeling the excitation distribution while maximizing the likelihood of the full waveform. LP-WaveNet and iLPCNet implement this formulation on the basis of WaveNet and LPCNet, respectively. Moreover, by replacing discrete μ-law softmax outputs with continuous mixture-of-Gaussians, these models successfully avoid degradation related to μ-law quantization.

Non-Autoregressive Excitation Modeling

All the autoregressive models described in the previous section combine vocal-tract modeling via an IIR filter with nonlinear excitation generation via an autoregressive network. Because they thus include a double autoregressive structure, they remain limited in inference speed and parallelization. In contrast, the non-autoregressive models discussed in this section remove explicit dependence on past samples in the excitation generator and produce the excitation signal in parallel, thereby alleviating these limitations.

QCP-DNN and GlottDNN employ feed-forward networks to regress short-time excitation signals (residual waveforms) directly, whereas GlotGAN adopts an adversarial formulation in which the discriminator operates directly on excitation signals, encouraging the generator to produce excitation signals. The generated excitation signals are then passed through an LPC filter in a separate synthesis stage to obtain the final speech waveform. These models generate short-frame waveforms that are concatenated

by overlap-add processing to reconstruct the full speech signal. However, this frame-based generation can lead to waveform discontinuities at segment boundaries, which result in degradation of perceptual quality if not carefully controlled or underestimation error. Moreover, because the IIR filter is applied as an external post-processing step, joint optimization of the excitation generation and vocal-tract filtering is difficult.

The early representative models include QCP-DNN [14], GlottDNN [15], and GlotGAN [18]. These approaches estimate the vocal-tract filter (LPC) coefficients using glottal inverse filtering and then convert the excitation waveforms generated by neural networks into speech waveforms through an analytically derived IIR filter. QCP-DNN and GlottDNN employ feed-forward neural networks to regress short-time excitation waveforms directly, whereas GlotGAN adopts an adversarial training in which the discriminator operates on excitation signals to improve the realism. The generated excitation frames are subsequently passed through the LPC filter and concatenated via overlap-add processing to reconstruct the entire waveform. However, because the excitation is generated on a frame-by-frame basis, discontinuities at frame boundaries lead to degradation of perceptual quality. Moreover, the external LPC filtering as post-processing prevents end-to-end training, hindering joint optimization of excitation generation and vocal-tract filtering.

To alleviate this problem, GELP [19] approximates the LPC-based IIR filter in the frequency domain, thereby avoiding recursive computation in the time domain and enabling efficient gradient propagation and faster generation. Specifically, the IIR transfer function (or its all-pole approximation) is converted into a finite-length frequency response, replacing sequential time-domain filtering with element-wise multiplication in the spectral domain. In GELP, the excitation (residual) signal is produced directly as a time-domain waveform by a non-autoregressive CNN-based GAN generator. Excit-

Glow, on the other hand, employs a parallel flow-based model (WaveGlow) to generate the excitation waveform. The generated excitation is then passed through an LPC synthesis filter obtained from linear prediction analysis, and the overall system is trained with a combination of excitation-domain likelihood and multi-resolution STFT loss on the reconstructed speech signal. This formulation integrates the filter into the end-to-end network in a way that allows efficient gradient propagation back to the excitation generator and conditioning networks. Although the long-term temporal response of the original IIR filter cannot be reproduced exactly under this finite-length approximation, the short-term spectral characteristics of speech are preserved sufficiently.

4.3.3 Data-Driven IIR Filter Models

In the above IIR-based vocoders, the filter coefficients are analytically estimated through linear predictive analysis. Subsequent studies have proposed alternative formulations in which the coefficients of linear filters are learned directly by neural networks in a data-driven fashion. In these approaches, the filter parameters are optimized end-to-end based on the training objective of speech waveform generation. The next sections organize existing methods within this framework into (1) autoregressive excitation-generation models and (2) non-autoregressive excitation-generation models.

Autoregressive Excitation Modeling

This category integrates an autoregressive excitation generator with a learnable IIR filter. Representative models such as E2E-LPCNet [29] and MWDLP [28] build on the LPCNet architecture, replacing analytical LPC estimation with neural prediction.

E2E-LPCNet (End-to-End LPCNet) [29] fully integrates the LPC computation into

the neural network, eliminating the need for analytic preprocessing. The network directly predicts reflection coefficients, constraining them to the stable region $(-1, 1)$ using a tanh activation, and analytically converts them into LPC coefficients. To make μ-law quantization differentiable, E2E-LPCNet introduces a linear interpolation-based differentiable embedding layer, allowing waveform-level gradients to propagate back to the frame-level LPC computation. Consequently, the model achieves a fully end-to-end architecture that jointly generates excitation signals and LPC coefficients from acoustic features.

MWDLP (Multiband WaveRNN with Data-driven Linear Prediction) [28] targets μ-law-encoded discrete waveforms and integrates a multiband WaveRNN architecture with data-driven linear prediction. The IIR filter parameters are predicted by the neural network, and the waveform probability distribution is modeled through linear prediction in logit space. In addition, MWDLP employs Gumbel sampling for reparameterization, which enables a differentiable formulation of the multi-resolution STFT loss [8] (see Section 3.4.2) for discrete waveform generation. The incorporation of this spectral loss further improves perceptual quality.

Non-Autoregressive Excitation Modeling

The neural vocoders described in the previous section suffer from limited parallelization because both the autoregressive excitation generation and the recursive nature of linear filtering introduce sample-wise dependencies, resulting in a computational burden. To alleviate these limitations, the following methods combine parametric and parallel excitation generation with data-driven learning of IIR filters.

A representative example is GOLF (Glottal-flow LPC Filter) [26], which models the glottal source using Glottal Flow Wavetables. In GOLF, the LPC coefficients are as-

sumed constant within each short-time frame, and a fixed all-pole IIR filter is applied per frame. Waveforms synthesized independently for each frame are subsequently concatenated using overlap-add processing. By eliminating sample-wise recursion during training, this frame-wise formulation achieves an approximate parallelization of IIR filtering. Filter stability is further ensured by parameterizing each second-order section with reflection coefficients, which constrain the poles to remain inside the unit circle.

Differentiable GOLF [27] discards the frame-wise approximation and introduces a fully recursive, differentiable formulation that applies time-varying LPC coefficients on a sample-by-sample basis. The recursive IIR equations are explicitly incorporated into the gradient computation, enabling the LPC coefficients to be learned end-to-end while accurately capturing temporal dependencies. Although more computationally demanding than the frame-wise approach, this formulation achieves more faithful neural vocoder training through explicit integration of the IIR dynamics.

4.3.4 Data-Driven FIR Filters

Compared with the IIR-based models described in the previous sections, several methods employ FIR filters, providing more stable and efficient training and synthesis. A common feature of these methods is that time-varying FIR filter coefficients are learned as data-driven parameters predicted by neural networks. In this section, we categorize existing methods according to whether the excitation is modeled analytically or in a data-driven manner.

Parametric Excitation Modeling

This family of methods adopts parametric excitation signals based on the traditional source-filter formulation, while neural networks are used to estimate the time-varying spectral shaping parameters.

Differentiable digital signal processing (DDSP) [24] adopts the harmonic-plus-noise synthesis model, described in Section 4.2. The harmonic component is generated by additive synthesis, that is, by explicitly summing sinusoidal partials whose frequencies are aligned with the fundamental frequency F_0. The aperiodic component is obtained by filtering a white-noise signal with a time-varying FIR filter whose coefficients are predicted by a neural network. This filter shapes the spectral envelope of the noise component, transforming the flat spectrum of white noise into the desired aperiodic spectral characteristics. The filtered noise is then combined with the harmonic component to produce the final waveform, optionally followed by a linear filter that applies reverberation. In DDSP, therefore, the FIR filter serves as a spectral shaping filter within the noise generation path.

SawSing [25] is a DDSP-based singing vocoder that replaces DDSP's additive harmonic synthesizer with a subtractive synthesis formulation. Whereas DDSP generates the periodic component by additively summing sinusoidal signals aligned with the fundamental frequency F_0, SawSing instead uses a sawtooth waveform, naturally rich in harmonic overtones, as a fixed excitation signal. A linear time-varying FIR filter, whose coefficients are predicted from the input mel-spectrogram by a neural network, selectively attenuates and preserves frequency components to shape the excitation spectrum into the desired periodic spectral characteristics. Because the excitation is produced by a continuous-phase oscillator and only its spectrum is sculpted by the filter, SawSing naturally maintains phase continuity, which well aligns the human speech. Because the

excitation is generated by a continuous-phase oscillator and only its spectral envelope is shaped by the filter, SawSing inherently maintains phase continuity, closely mirroring the behavior of the human glottal source [25].

Data-Driven Excitation Modeling

In this approach, the excitation signal is generated in a data-driven manner through linear or nonlinear transformations, and the waveform is then produced by applying a data-driven FIR filter as a vocal-tract filter.

NHV [30] is grounded in cepstral analysis, a framework that represents filter characteristics through the complex cepstrum of their impulse responses. In NHV, a neural network predicts this complex cepstrum for each time frame, and it is converted into a linear time-varying filter. Because the complex cepstrum encodes both magnitude- and phase-related spectral characteristics, this formulation allows the filter response to be learned in a fully data-driven manner, enabling flexible modeling of detailed timbral characteristics. For source modeling, NHV analytically generates periodic excitation in the form of a pulse train and aperiodic excitation as noise. These excitation signals are then shaped by the predicted time-varying filters to model vocal-tract resonances and reproduce detailed timbral characteristics in the final waveform.

NITECH E2E-TTS [31] incorporates a neural vocoder into an end-to-end speech synthesis framework by embedding a differentiable FIR-based synthesis filter within the system. In the vocoder, a mixed excitation signal, composed of an impulse train and white noise, is first generated and then processed through a series of neural prenets. The resulting excitation is filtered using a cascade of time-varying FIR filters, whose coefficients are predicted from acoustic features by a neural network. This FIR cascade serves as a fully differentiable spectral shaping module, allowing the excitation

generation and filtering processes to be jointly optimized within the end-to-end model. By integrating the entire filtering operation into the computation graph, the system generates waveforms through differentiable convolution driven by data-driven filter parameters.

In contrast, FIRNet [32] employs a two-stage linear FIR processing architecture for the excitation signal. Each stage consists of a stack of time-varying FIR filters. The first multi-stage FIR block linearly transforms the analytic excitation signal into a residual-like signal, and the second multi-stage block serves as a resonance filter that models vocal-tract characteristics. All FIR coefficients are predicted in a data-driven manner by a neural network, enabling flexible spectral shaping despite the finite filter order, as in NITECH E2E-TTS. Furthermore, FIRNet applies a source-regularization loss [77] to constrain the residual FIR block and prevent excessive modification of the excitation, thereby maintaining a physically plausible source-filter decomposition.

4.4 Fundamental-Frequency-Driven Mechanisms

4.4.1 Overview

Speech waveforms exhibit periodic structures originating from vocal fold vibrations, and this periodicity is governed by the fundamental frequency F_0. If a neural vocoder is designed so that waveform generation is directly controlled by F_0, the model can produce waveforms that follow target F_0 values more faithfully, enabling reliable pitch manipulation and synthesis over a wide frequency range. However, conventional neural vocoders generally lack explicit mechanisms for pitch control. Simply providing F_0 as an auxiliary input is often insufficient because the network typically learns a statistical mapping from the training distribution, rather than behaving in a manner

inherently conditioned on the physical parameter F_0. Consequently, when the target F_0 falls outside the training distribution, such models often exhibit unstable or degraded generation, revealing their weakness in extrapolating to unseen F_0 values.

To achieve consistent and physically interpretable pitch-dependent behavior, it is therefore beneficial to incorporate an explicit F_0-driven mechanism into the model architecture. Existing neural vocoders with such mechanisms can be broadly categorized into two types: (1) methods that use analytically generated periodic signals derived from F_0, and (2) methods in which neural network operations dynamically change according to F_0. The following subsections introduce these F_0-driven mechanisms, which explicitly encode the periodic structure of speech waveforms through formulations tied to F_0.

4.4.2 Analytic Periodic Signals

When a neural vocoder learns speech generation purely in a data-driven manner, the model must internally reproduce periodic structures from scratch. However, the phase of a periodic signal progresses continuously over time according to its instantaneous frequency, and reproducing such phase evolution is particularly challenging for non-autoregressive architectures. Chunked Autoregressive GAN (CARGAN) [78] empirically demonstrated this issue, showing that non-autoregressive CNNs, lacking an explicit mechanism for cumulative phase evolution, often fail to maintain a coherent phase trajectory, leading to periodic instability. Their chunked autoregressive architecture alleviates this problem by introducing an autoregressive inductive bias that facilitates the modeling of phase progression.

Alternatively, for non-autoregressive models, periodic signals analytically generated from F_0, such as sinusoids or pulse trains, can be provided to the network. This relieves

the model from having to learn harmonic phase progression from scratch and greatly simplifies the representation of periodic structure.

This idea was first introduced in the Neural Source-Filter (NSF) model [33], which analytically constructs an F_0-driven periodic signal and interprets it as the excitation. In NSF, the phase is updated sample-by-sample according to the instantaneous F_0, and the periodic waveform is obtained by evaluating a sinusoid with this cumulative phase. For clarity, the excitation corresponding to the fundamental component can be written as:

$$e_t = \sin(\theta_t + \phi) + n_t, \quad \theta_t = \theta_{t-1} + \frac{2\pi f_{0,t}}{F_\mathrm{s}} \tag{4.6}$$

where $f_{0,t}$ is the sample-level F_0 aligned with the sampling frequency F_s, θ_t is the accumulated phase, ϕ is the initial phase, and n_t is Gaussian noise. NSF extends this formulation to multiple harmonics and then applies a nonlinear neural filter to transform the periodic signal into a speech waveform.

Although NSF is inspired by the source-filter theory, it fundamentally differs from traditional source-filter models in that it employs a nonlinear neural filter rather than a linear vocal-tract filter. From a broader perspective, NSF can therefore be regarded as an F_0-driven neural vocoder, where an analytically generated periodic signal functions as a pitch-dependent template.

Following NSF, several studies have generalized this idea into more flexible neural vocoder architectures. Representative examples include the harmonic-plus-noise neural vocoders (see Section 4.2) such as PeriodNet [34] and HN-PWG [35]. These models inject analytically generated periodic signals into a dedicated periodic subnetwork, enabling a functional decomposition in which periodic and aperiodic components are generated by separate modules.

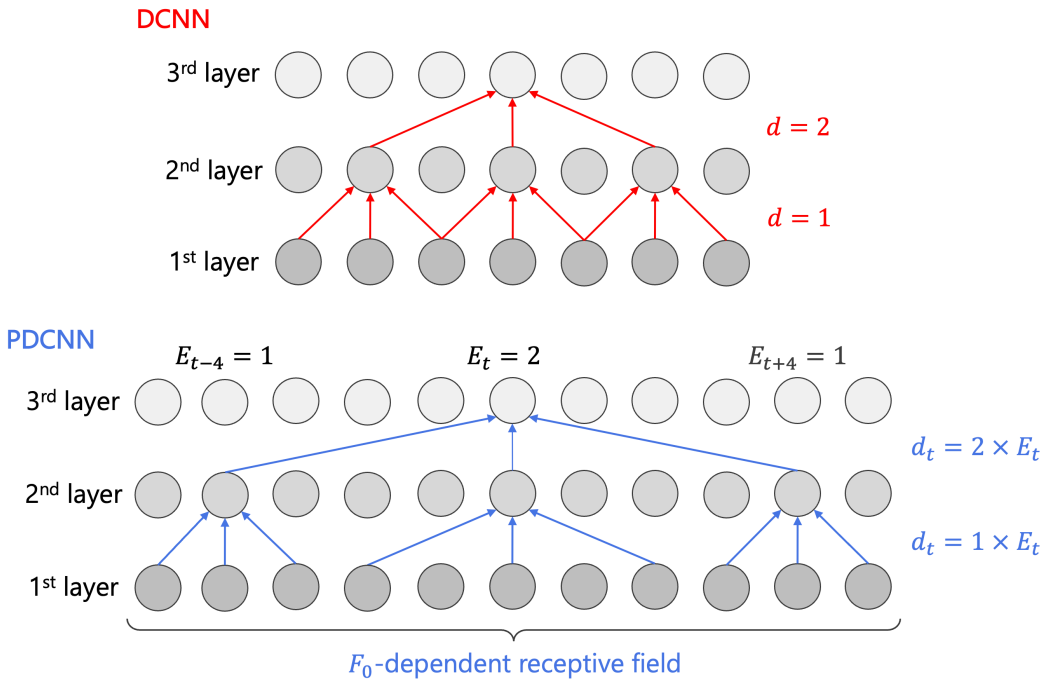

Figure 4.2: *Comparison between a conventional Dilated CNN (DCNN) and a Pitch-Dependent Dilated CNN (PDCNN). In the PDCNN, the dilation factor d_t is dynamically determined according to the fundamental frequency F_0 at each time step.*

A different line of work, including Period-HiFi-GAN [79] and Harmonic-Net [38], introduces a multiscale periodic-signal injection mechanism. In these models, the periodic signal is downsampled to multiple temporal resolutions using learnable convolutions and then injected into intermediate layers of a upsampling generator like HiFi-GAN [67]. This multi-resolution formulation improves F_0 controllability even within upsampling-based architectures.

4.4.3 Pitch-Dependent Dilated Convolution

Another line of research explores CNN architectures whose internal structure dynamically changes in accordance with the given F_0 sequence. This idea was introduced

in Quasi-Periodic WaveNet (QP-Net) [80] and Quasi-Periodic Parallel WaveGAN (QP-PWG) [81], both of which employ pitch-dependent dilated convolutional neural networks (PDCNNs). QP-Net and QP-PWG can therefore be viewed as extensions of WaveNet and Parallel WaveGAN, respectively, improving robustness to unseen pitch.

In PDCNNs, the dilation factor in each convolutional layer is modulated by the instantaneous F_0, allowing the receptive field to expand or contract in synchrony with the target periodicity. This mechanism provides the network with an inductive bias that facilitates the modeling of quasi-periodic patterns in speech waveforms. Formally, let F_s denote the sampling frequency, $f_{0,t}$ the fundamental frequency at time t, and d the base dilation factor. The time-varying dilation factor d_t is defined as:

$$d_t = \begin{cases} \lfloor E_t \rfloor \times d & (E_t > 1), \\ 1 \times d & \text{otherwise}, \end{cases} \tag{4.7}$$

where $E_t = F_\mathrm{s}/(f_{0,t}\, a)$ is proportional to the fundamental period, and a is a tunable coefficient, referred to as the dense factor, which together with the sampling frequency F_s determines the maximum representable fundamental frequency.

Wu et al. [81] reported that placing PDCNNs in the early stages of the generator is more effective for improving F_0 controllability and overall speech quality than inserting them into later layers. This suggests that performing source-related processing, such as periodic structure formation, in early layers while delegating spectral shaping to later layers provides a beneficial inductive bias, implicitly resembling a source-filter theory within the network. Subsequently, a related approach was later adopted in Harmonic-Net [38], which applies PDCNNs at multiple temporal resolutions within a HiFi-GAN-style upsampling generator.

4.4.4 Limitations and Challenges

The introduction of F_0-driven mechanisms has substantially improved the pitch controllability of neural vocoders by enabling explicit manipulation of F_0. This thesis provides further analysis in Chapter 7, that even a single sinusoidal component can provide a strong inductive bias for robust modeling of harmonic structure.

Despite these benefits, several challenges remain. When the target F_0 falls far outside the training distribution, such as at unusually high or low pitches, the generation behavior often becomes unstable, revealing limitations in extrapolative generalization. As discussed in Chapter 7, aliasing inherent to discrete-time neural vocoders is one contributing factor to this degradation under extreme F_0 conditions. Moreover, although these models incorporate F_0-driven mechanisms, their generative processes still do not fully conform to the physical principles of speech production.

Therefore, while F_0-driven mechanisms are effective as auxiliary modules for capturing source-related characteristics, these observations suggest that integrating them within a more comprehensive source-filter framework can lead to greater stability and physical consistency. Motivated by this hypothesis, Chapter 5 explores an integrated source-filter model that incorporates F_0-driven structures.

4.5 Continuous-Discrete Time Gap

Although speech is a continuous-time physical phenomenon, neural vocoders are ultimately discrete-time computational models. DNNs serve as powerful nonlinear function approximators, but their implementations are intrinsically digital. This discrepancy creates a fundamental mismatch between the continuous nature of speech and the discrete nature of DNN-based generation.

In this section, we highlight aliasing as a representative issue emerging from this continuous-discrete time gap, and discuss it from three perspectives: (i) why aliasing constitutes a physical inconsistency, (ii) how it manifests in neural vocoders, and (iii) recent trends aimed at mitigating this issue.

4.5.1 Aliasing Problem

Aliasing refers to the phenomenon in which high-frequency components fold back into the low-frequency range when a signal contains energy beyond the Nyquist frequency (half the sampling frequency) [12, 13]. Because such behavior never occurs in continuous-time speech production, aliasing represents an artificial distortion inherent to discrete-time implementations. In this sense, we can regards aliasing as an indication that the neural vocoder is misaligned with the physical speech production process.

In neural vocoders, aliasing primarily originates from two sources: (1) resampling operations such as upsampling and downsampling, and (2) nonlinear operations such as activation functions. The resulting spectral folding introduces spurious components that are absent in the underlying continuous-time signal, causing irreversible distortion and obscuring the true frequency characteristics. Aliasing often manifests as artificial noise in the synthesized speech and also degrades the shift-invariance properties of CNNs, causing instability in temporal modeling. Prior work [11] has further demonstrated that aliasing can impair a model's extrapolation and generalization performance.

4.5.2 Causes of Aliasing

Nonlinear Processing

Prior work such as StyleGAN3 [10] and BigVGAN [11] have demonstrated that pointwise nonlinear operations within neural networks play a crucial role in generating new frequency components. For example, when the ReLU activation [82] is applied to a sinusoidal input $\sin(\omega t)$, the output contains infinitely many harmonic components. This can be expressed as

$$\mathrm{ReLU}(\sin(\omega t)) = \frac{1}{\pi} + \frac{\sin(\omega t)}{2} - \sum_{k=1}^{\infty} \frac{2\cos(2k\omega t)}{\pi(2k-1)(2k+1)}. \tag{4.8}$$

This expansion illustrates the fundamental mechanism by which nonlinearities broaden a signal's spectrum. A pointwise nonlinearity corresponds to multiplication in the time domain, which becomes convolution in the frequency domain. In the case of ReLU, the operation can be interpreted as applying a dynamic rectangular window to the input signal; the associated window has a sinc-shaped frequency response. Consequently, the input spectrum is convolved with this sinc kernel, spreading an initially finite-band signal across infinitely many harmonic frequencies.

While continuous-time systems can represent these harmonics without restriction, discrete-time systems inevitably fold components above the Nyquist frequency back into the baseband, producing aliasing. These effects have been recognized in classical bandwidth-extension research, where nonlinear processing generates both desirable harmonics and undesirable aliasing artifacts [83]. The same considerations carry over directly to neural vocoders and provide meaningful insight into their behavior.

Resampling Layers

Neural vocoders frequently employ upsampling layers to convert low-time-resolution acoustic features into high-resolution waveform signals. While this strategy enables efficient waveform generation, it is well known that upsampling operations can introduce artifacts [84]. Pons et al. [84] compared several upsampling methods, including transposed convolution, sub-pixel convolution, and linear interpolation. They reported that transposed and sub-pixel convolution-based methods often produce periodic tonal artifacts, perceived as metallic noise, in the generated waveform. In contrast, interpolation-based approaches such as linear interpolation can reduce these artifacts but tend to introduce filtering distortion due to the frequency response of the interpolation kernel (e.g., a triangular filter).

Furthermore, spectral replicas introduced during upsampling can propagate these artifacts through subsequent network layers, leaving consistent artifacts in the final output. Likewise, when pooling or strided convolution operations are applied without appropriate anti-aliasing filters, out-of-band components can fold back into the lower-frequency range, further degrading the generated signal.

4.5.3 Countermeasures

In neural vocoder research, several architectural and training-based strategies have been proposed to mitigate the aliasing-related problems. This section provides an overview of representative approaches.

Temporal Bandwidth Extension

BigVGAN [11] introduces anti-aliased nonlinear operations to structurally suppress aliasing within neural vocoders. This concept follows StyleGAN3 [10], which demonstrated that internal aliasing in image generation networks leads to confusion between high- and low-frequency components, making fine-grained spatial control difficult.

In BigVGAN, the input signal is first temporarily upsampled to increase its effective sampling frequency and widen the available bandwidth before nonlinear processing. Low-pass filters are then applied before and after the nonlinear activations to suppress the high-frequency artifacts generated by the nonlinearities. Finally, the signal is downsampled back to the original sampling frequency, effectively mitigating spectral folding.

As in StyleGAN3, this approach reduces aliasing and improves speech quality and generalizability. However, the temporary oversampling incurs additional computational costs, posing challenges for real-time or resource-constrained applications. The concrete formulation of this method, along with its associated trade-offs, will be discussed in detail in Section 7.2.2.

Shift-Equivalence Promotion

JenGAN [39] addresses aliasing from the perspective of shift equivalence, a property theoretically preserved in continuous-time CNNs composed of convolution operations and pointwise nonlinearities [85, 86]. In the continuous-time domain, such networks maintain strict shift equivalence, meaning that a temporal shift in the input signal yields an equivalent shift in the output. However, in discrete-time implementations, aliasing introduces phase-dependent interference patterns, causing the output to deviate from the expected shifted response. Formally, for the neural net-

work f_θ and the input signal $x(t)$, the presence of aliasing implies that the relation $f_\theta(x(t-\delta)) = f_\theta(x)(t-\delta)$ often does not hold.

To address this issue, JenGAN introduces a training strategy that explicitly promotes shift equivalence, thereby indirectly mitigating aliasing effects. During training, random temporal shifts are applied to the input signal using a sinc filter, and corresponding inverse shifts are applied to the output at appropriate module levels. This encourages the network to produce outputs that respond consistently to temporal shifts in the input, leading to more stable internal representations. Consequently, JenGAN improves the continuity of harmonic structures in the generated speech, leading to enhanced perceptual speech quality.

However, promoting shift equivalence alone does not guarantee complete suppression of aliasing. Shift equivalence enforces consistency with respect to temporal shifts, whereas aliasing stems from spectral folding caused by the finite bandwidth of discrete-time systems. Moreover, JenGAN's method introduces additional filtering operations and random-shift training steps, incurring extra computational overhead.

Training-Based Strategies

Several studies have explored training-based approaches to suppress artificial noises, including aliasing distortion, in neural vocoders. Pons et al. [84] suggested that upsampling-related artifacts can be mitigated through optimization with training. Shang et al. [40] proposed a training framework that detects and suppresses aliasing by quantifying spectral symmetry using the structural similarity index [87]. This method explicitly evaluates and minimizes spectral folding artifacts observed in the training data.

However, despite the theoretical ability of convolutional layers to represent anti-aliasing filters, standard optimization procedures rarely induce such filter character-

istics spontaneously [86, 88]. Furthermore, neural vocoders that operate on long sequential data must learn long convolutional kernels to realize the wide-band filtering responses necessary for effective aliasing suppression, further increasing the difficulty of optimization. Consequently, these training-based methods often struggle to realize sufficiently strong anti-aliasing behavior and to generalize under out-of-distribution conditions.

4.6 Time-Frequency Domain Models

4.6.1 General Formulation

Neural vocoders operating directly in the time-frequency domain differ from conventional waveform-based models in that they predict a time-frequency representation of speech using a neural network. This formulation shortens the effective sequence length that the network must convolve over, thereby improving computational efficiency. Such models are typically built upon the short-time Fourier transform (STFT). In general, the neural network predicts both the amplitude and phase (or equivalently, the real and imaginary components) of the complex spectrogram, and the final waveform is reconstructed using the inverse STFT (iSTFT). The overall generation and reconstruction process can be expressed as:

$$\hat{S}(t, f) = \mathcal{G}(\boldsymbol{z}), \quad \hat{\boldsymbol{x}}(t) = \text{iSTFT}(\hat{S}(t, f)) \tag{4.9}$$

where $\boldsymbol{z}$ denotes the acoustic features, $\mathcal{G}$ represents the neural model, $\hat{S}(t, f)$ is the estimated complex spectrogram, and $\hat{\boldsymbol{x}}$ is the reconstructed waveform. Here, t and f denote the time-frame index and frequency-bin index of the STFT, respectively.

During training, as in time-domain models, a hybrid loss combining adversarial and auxiliary components is commonly used, as described in Eq. 3.18. Consequently, a hybrid architecture, one that performs generation in the time-frequency domain followed by discrimination and evaluation after reconstruction into the time domain, has become a prevailing design paradigm in recent neural vocoder research.

4.6.2 Theoretical Advantage

Although the time-frequency representation is mathematically a coordinate transform of a time-domain signal, it offers several advantages for speech generation. One advantage is that harmonic structures arising from vocal-fold vibrations, as well as spectral envelope variations shaped by vocal-tract resonances, are more clearly represented in the time-frequency domain. For speech and other natural time signals, important information is often more distinctly organized along the frequency dimension, which can allow neural networks to model the intrinsic structure of the signal more effectively. In this sense, the time-frequency domain can be viewed not only as an analytical tool but also as a potentially suitable representational space for speech generation.

Another benefit is computational efficiency: generating signals in the time-frequency domain reduces the effective sequence length, which typically improves inference speed. Recent work has shown that such models can achieve speech quality comparable to, or even surpassing, time-domain neural vocoders while requiring substantially fewer computational costs.

Finally, time-frequency-domain generation carries an important theoretical implication: it enables speech modeling without violating the sampling theorem. In STFT-based generation, the frequency range is explicitly defined and band-limited, allowing

the model to avoid the high-frequency folding that naturally arises in discrete-time models. From this perspective, time-frequency-domain models can be regarded as a more physically consistent formulation. In Chapter 7, this thesis theoretically demonstrates that the time-frequency domain modeling can fundamentally avoid aliasing, and further discusses how the choice of representational space relates to physical consistency.

4.6.3 Existing Methods

Recent neural vocoders synthesize speech by predicting time-frequency representations rather than directly generating waveforms. These spectrogram-based approaches can be broadly categorized according to how they represent and reconstruct the complex spectrogram.

Amplitude Spectrogram Estimation

Early models such as SpecGAN [89] and GANstrument [90] estimate only log-amplitude spectrograms, sometimes using mel-frequency scaling. Estimating the phase component is particularly challenging due to its inherent randomness and the ambiguity introduced by periodicity modulo 2π. Consequently, these methods rely on the Griffin-Lim algorithm (GLA) [91] to reconstruct waveforms by iteratively estimating a phase that is consistent with the predicted amplitude spectrogram. However, because GLA depends solely on amplitude-phase consistency constraints, it often yields unnatural phase structures, leading to audible artifacts and degraded audio quality.

Complex Spectrogram Estimation

To overcome the limitations of amplitude-only prediction, several studies have proposed neural vocoders that directly estimate complex spectrograms and reconstruct waveforms via inverse STFT (iSTFT). iSTFTNet [92, 93] is the first vocoder to follow this approach. It converts low-temporal-resolution mel-spectrograms into high-temporal-resolution, low-frequency-resolution complex spectrograms using multiple transposed convolutional layers and HiFi-GAN-based residual blocks, after which a final iSTFT generates the waveform. Following iSTFTNet, numerous studies have further explored spectrogram-based generative modeling.

APNet [94] extends this direction by removing upsampling layers entirely and directly estimating complex spectrograms. To mitigate the inherent difficulty of phase prediction, stemming from phase ambiguity and instability, APNet trains the model to align multiple components of the target and predicted spectrograms, including log amplitude, real and imaginary parts, instantaneous frequency, group delay, and phase time differences. Although this multi-objective formulation promote explicit phase modeling, relying on explicit distance losses that assume a one-to-one correspondence between complex spectrograms and waveforms can be suboptimal due to redundancy in the complex spectrogram representation.

In contrast, LightVoc [95] and Vocos [96] adopt iSTFT-based synthesis but employ simpler and typical GAN-style training objectives: mel-spectrogram reconstruction loss, adversarial loss, and feature-matching loss. They replace standard CNN backbones with more advanced architectures such as ConvNeXt [97] and Conformer [98] to enhance modeling capacity. Vocos, in particular, achieves significantly faster inference than iSTFTNet. However, simply relying on large model capacity does not guarantee robustness, and Vocos exhibits reduced stability under out-of-distribution conditions,

such as unseen F_0 values, as demonstrated in Chapter 7.

Guided Phase Spectrogram Estimation

Another line of research incorporates parametric excitation signals to explicitly model harmonic structures, following the neural source-filter (NSF). HiNet [99] first estimates a log-amplitude spectrogram and then predicts the phase spectrogram conditioned on an analytic harmonic signal-constructed from F_0 as described in Section 4.4.2. This formulation enforces a coherent phase trajectory and enables more reliable phase estimation through the STFT.

HiFTNet [100] extends this approach by integrating F_0 estimation, a harmonic-plus-noise excitation mechanism, and modern architectural components such as the Snake activation [101] and the truncated relativistic GAN loss [102]. While these models improve phase consistency and harmonic structure representation, they remain susceptible to aliasing introduced by resampling layers and to the limited frequency resolution imposed by discrete STFT bin sizes.

4.7 Summary

This chapter examined neural vocoder designs that pursue consistency with the physical speech production mechanism.

First, the harmonic-plus-noise modeling approach was reviewed. By modeling the periodic and aperiodic components separately, neural vocoders can reproduce the distinct roles, leading to higher naturalness and controllability.

Second, neural vocoders incorporating linear filters were discussed. These models represent vocal-tract resonances through IIR or FIR filters while allowing neural net-

works to model the excitation source or filter coefficients. Such architectures promote efficient, interpretable, and controllable speech synthesis.

Third, F_0-driven mechanisms for achieving robust pitch control were reviewed. Introducing analytic periodic signals and pitch-dependent convolutional layers encourages neural vocoders to maintain consistent behavior across unseen F_0 conditions.

Fourth, the challenges arising from the continuous-discrete time gap in neural vocoding were analyzed. This section brought up the aliasing issue, non-physical artifacts, and the lack of shift invariance introduced by discrete-time operations, and reviewed previous approaches to mitigate these effects.

Finally, models that operate in the time-frequency domain were discussed. Representing and generating signals in this domain improves computational efficiency and inherently avoids aliasing, making the model behavior more consistent with the physics of sound (as demonstrated in Section 7).

5 Unified Source-Filter Modeling

To this end, we propose the Unified Source-Filter GAN (uSFGAN), a neural vocoder to achieve both high speech quality and robust F_0 controllability. This chapter is organized as follows. Section 5.1 presents the motivation and conceptual background of the proposed approach. Section 5.2 details the architecture and learning framework of uSFGAN. Section 5.3 reports experimental evaluations and comparative analyses. Finally, Section 5.4 summarizes the findings and conclusions of this chapter.

5.1 Introduction

Human speech production can be explained as a process in which an excitation signal generated by the vocal folds is shaped by the resonant characteristics of the vocal tract filter [7]. Conventional vocoder research can be broadly categorized into three approaches according to their underlying design principles, as discussed in Chapters 2, 3, and 4. The first is the signal-processing approach, which explicitly models the physical mechanism of speech production. This approach offers high controllability and interpretability but relies on assumptions such as short-term linearity and stationarity. However, the reliance on these assumptions, together with the difficulty of accurate phase modeling, prevents it from fully capturing the complex and dynamic aspects of natural speech, thereby limiting the naturalness. The second is the fully data-driven approach, which learns a nonlinear mapping from acoustic features to speech waveforms

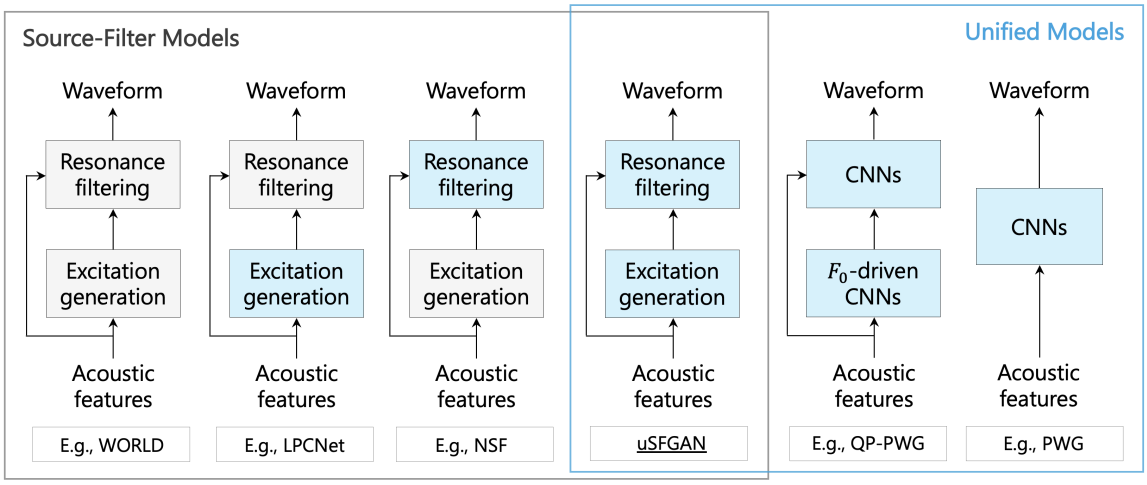

Figure 5.1: *Comparison of source-filter-based and unified vocoder architectures. Gray blocks represent signal-processing (parametric) components, whereas blue blocks denote neural network components. Here, a unified model refers to a vocoder architecture that integrates both the source and filter components within a single network, without requiring an explicit separation between them.*

using large-scale speech datasets and deep neural networks. This approach achieves a level of naturalness that surpasses conventional vocoders. However, because this approach does not consider the physical mechanisms of speech production, it often suffers from limited controllability and generalization ability to unseen conditions.

The third is the hybrid approach, which incorporates analytical excitation models or linear filters into deep neural networks. This approach aims to balance the physical consistency of the speech production mechanism with the flexibility of the data-driven approach. Nevertheless, it still depends on analytically designed parametric modules, making it difficult to fully exploit the expressive potential of deep learning. Deep learning can capture nonlinear and latent relational structures that are difficult to describe explicitly with analytical modeling. To fully leverage this expressive capability, this study adopts a new perspective that encourages the spontaneous formation of structures consistent with the speech production process within a nonlinear model.

Specifically, it embeds a functional perspective based on the source-filter theory [7] within a single neural network, enabling the implicit functional decomposition of excitation generation and resonant filtering. This allows the entire system to be trained in a unified, data-driven manner while maintaining functional consistency with the speech production mechanism. We refer to this framework as Unified Source-Filter Modeling and implement it within the generator of a GAN [9] framework, termed the Unified Source-Filter GAN (uSFGAN).

The proposed uSFGAN builds upon the Quasi-Periodic Parallel WaveGAN (QP-PWG) [37] architecture by guiding the Pitch-Dependent Dilated CNN (PDCNN) [36, 37] to function as a source-network that generates excitation components, while the following Dilated CNN (DCNN) acts as a filter-network modeling vocal-tract resonance. Details of the PDCNN are provided in Section 4.4.3. A spectral regularization loss is applied to the source network to promote spectral flatness, thereby confining timbral information, shaped by the vocal tract, to the filter network and clarifying functional separation. Furthermore, a periodic signal derived from the fundamental frequency (F_0) is provided as an auxiliary input to the source network, enhancing generalization to unseen F_0 regions. Furthermore, inspired by Neural Source-Filte (NSF) [33], a periodic signal derived from the fundamental frequency F_0 is provided as an auxiliary input to the source-network, reinforcing generalization to unseen F_0 regions. Through this design, uSFGAN achieves a unified and interpretable representation of the source-filter mechanism within a deep generative model. Experimental evaluations demonstrate that uSFGAN achieves speech quality comparable to modern data-driven neural vocoders [67] while achieving controllability competitive with the classical source-filter model [3].

5.2 Unified Source-Filter GAN

To develop a high-fidelity and F_0-controllable neural vocoder, we propose uSFGAN, which represents the source-filter architecture with a single neural network based on GAN. The generator network is factorized into a source excitation generation network (source network) and a resonance filtering network (filter network) using a regularization loss to make the source network output reasonable source excitation signals. To further improve the F_0 controllability, we introduce F_0-driven mechanisms designed on the basis of QP-PWG and NSF into the source network. Moreover, inspired by the recent successes of the neural vocoders that adopt harmonic-plus-noise (HN) speech modeling [30, 34, 35, 103], we introduce HN source excitation generation to obtain better speech quality. The overall architecture of uSFGAN is shown in Fig. 5.2, and the generator architectures are shown in Fig. 5.3.

5.2.1 Factorization of Generator Network

In the proposed method, uSFGAN, the single GAN generator is functionally divided into a source network and a filter network. To achieve this, a regularization term is applied to the intermediate output of the generator (i.e., the output of the source network). This regularization is designed so that the intermediate representation possesses appropriate properties as a source signal, and it is incorporated as an additional term into the standard GAN-based optimization objective as described in Section 3.4.2. In this study, two novel regularization losses are introduced for the output of the source network. The fundamental principle shared by both losses is based on the source-filter assumption that a source signal does not contain resonant characteristics introduced by the vocal-tract filter. Therefore, these losses constrain the source-network's output

to prevent it from carrying timbral information.

Spectral Envelope Flattening Regularization Loss

The first regularization term, spectral envelope flattening regularization loss, is designed under the same assumption adopted in traditional source-filter vocoders such as STRAIGHT [2] and WORLD [3]: namely, that the excitation signal exhibits a flat spectral envelope and constant amplitude across all frequency bands and time frames. To enforce this condition, the spectral envelope is extracted from the source network's output signal, and its flatness is constrained via regularization.

For spectral envelope extraction, a simplified version of the CheapTrick [46] algorithm, described in Section 2.2.3, is employed. The original CheapTrick algorithm consists of the following three stages:

1. F_0-adaptive windowing and computation of the log power spectrum,
2. F_0-adaptive spectral smoothing, and
3. F_0-adaptive liftering in the cepstral domain.

In this study, several modifications are introduced to accelerate the estimation process. First, instead of re-estimating F_0 from the output signal, the provided auxiliary F_0 feature is directly used. These F_0 values are rounded to integers, and the corresponding window and liftering functions are pre-generated. Second, stage (2), spectral smoothing, is omitted due to its high computational cost, and spectral smoothing is instead realized by the F_0-adaptive liftering in stage (3). Although these modifications slightly reduce estimation accuracy, precise envelope estimation is not required for regularization purposes, so no practical degradation occurs.

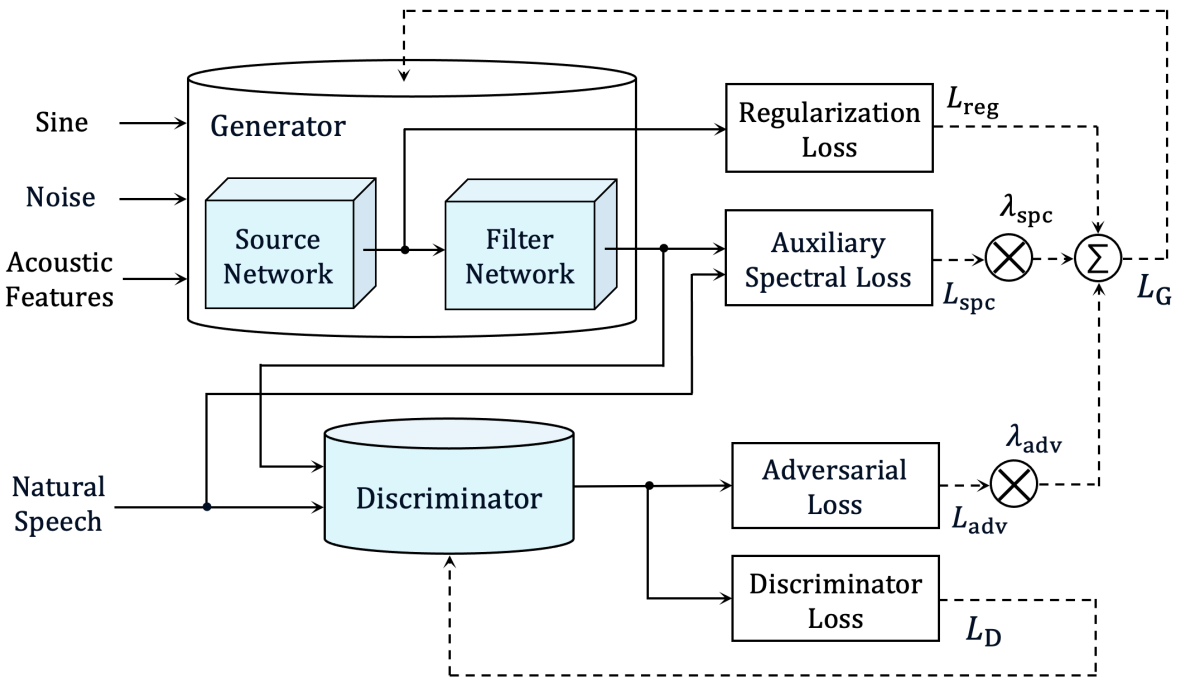

Figure 5.2: *Overall framework of the proposed uSFGAN. The generator takes conditional acoustic features together with a noise signal and produces a source excitation signal as well as a synthesized speech waveform. The discriminator receives natural and generated speech waveforms to perform adversarial training.*

The spectral envelope flattening regularization loss is defined as:

$$\mathcal{L}_{\text{reg}} = \mathbb{E}_{\boldsymbol{z}} \left[\frac{1}{N} || \ \log \hat{E}_{\boldsymbol{z}} \ ||_1 \right], \tag{5.1}$$

where $|| \cdot ||_1$ denotes the L1 norm, $\hat{E}_{\boldsymbol{z}}$ represents the spectral envelope amplitude extracted from the source network output, N is the number of elements, and $\boldsymbol{z}$ denotes the input acoustic features. Note that when this loss approaches zero, the linear-scale amplitude values $\hat{E}$ become one across all frequencies and time frames.

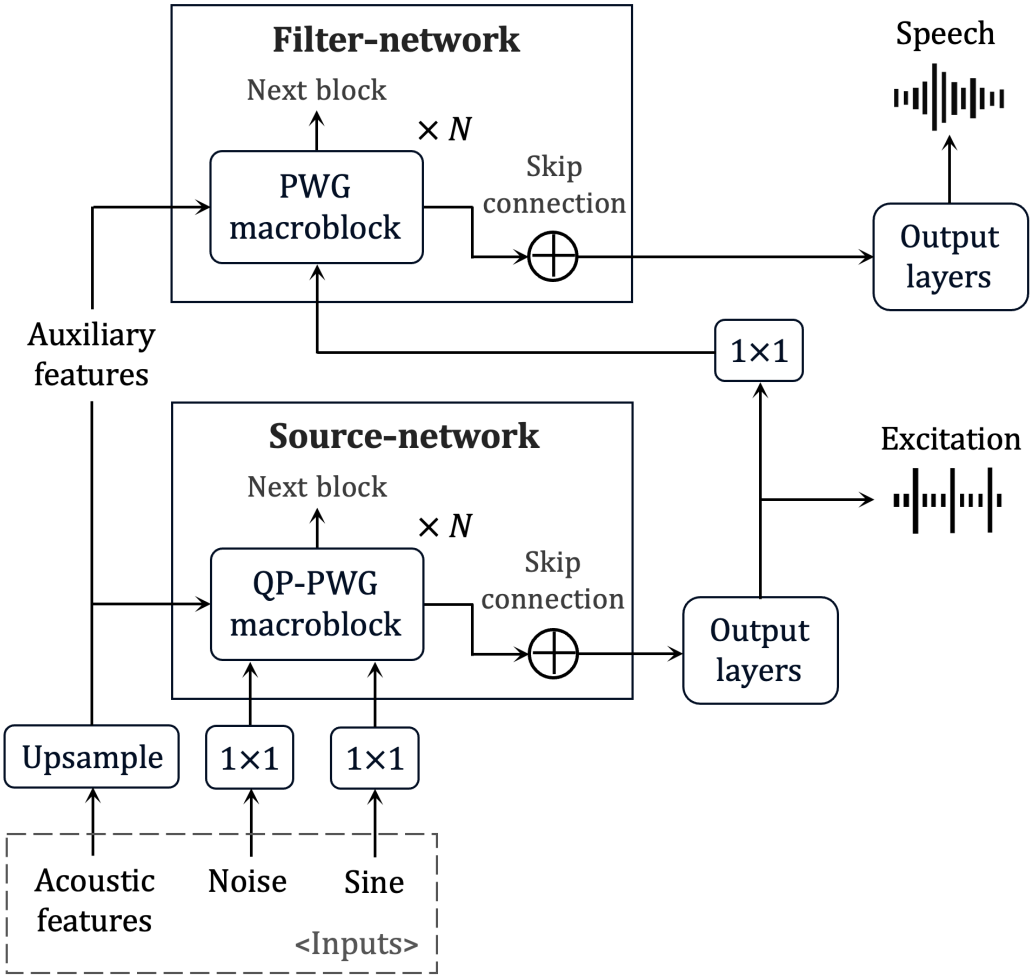

Figure 5.3: *An overview of the uSFGAN generator.*

Residual Spectra Targeting Regularization Loss

The second regularization loss is inspired by the concept of Linear Predictive Coding (LPC) [43,44], and it utilizes the residual spectra computed from natural target speech. Here, the term residual refers to the excitation component obtained by removing the vocal-tract filter characteristics from the speech signal. Unlike previous studies such as GELP [19] and LPCNet [21], which are introduced in Section 4.3 and directly predict the residual waveform itself, the proposed method regularizes the spectrum $\hat{S}_{\boldsymbol{z}}$ of the

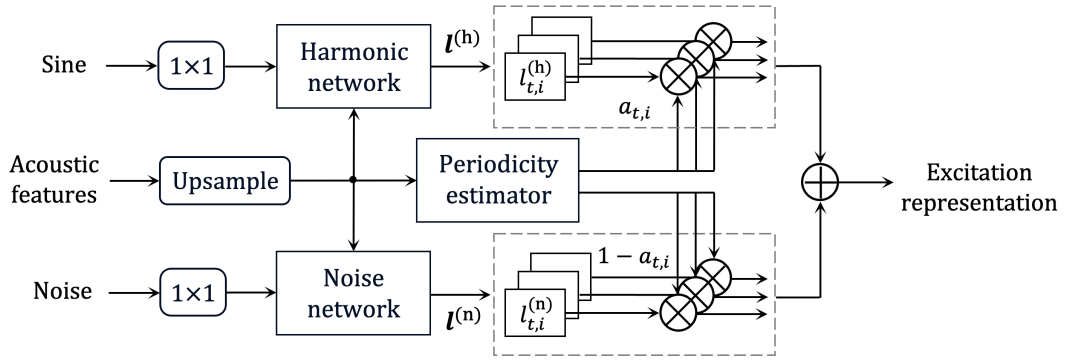

Figure 5.4: *The source-network of the parallel hn-uSFGAN generator. The harmonic- and noise-networks independently generate excitation representations from sinusoidal and noise inputs conditioned on acoustic features. The output excitation representation is used as the input to the filter-network and for computing the source regularization loss.*

excitation signal produced by the source network.

The residual spectra regularization loss is defined as:

$$\mathcal{L}_{\mathrm{reg}} = \mathbb{E}_{\boldsymbol{x},\boldsymbol{z}} \left[\frac{1}{N} || \ \log \psi(S_{\boldsymbol{x}}) - \log \psi(\hat{S}_{\boldsymbol{z}}) \ ||_1 \right], \tag{5.2}$$

where $\boldsymbol{x}$ and $\boldsymbol{z}$ denote the ground-truth waveform and the input acoustic features, $\psi(\cdot)$ represents the mel-spectrogram transform, and N is its dimensionality. The reference residual spectrum $S_{\boldsymbol{x}}$ is obtained by subtracting the estimated spectral envelope of the target speech from its log-amplitude spectrum. This spectral envelope is computed using the simplified CheapTrick algorithm described in the previous section. Furthermore, the power of $S_{\boldsymbol{x}}$ is scaled so that it has the same frame-level average power as the reference waveform (see Fig. 5.8).

Unlike the spectral envelope flattening regularization loss, this regularization encourages the excitation signal to reflect frame-level power variations and frequency-dependent attenuation patterns consistent with the target speech. This formulation is

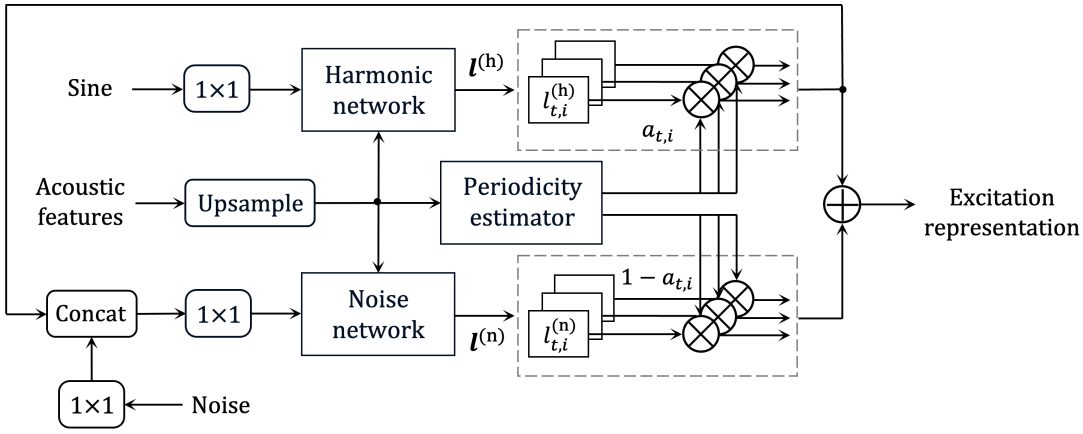

Figure 5.5: *Source network of the cascade hn-uSFGAN generator. Compared with the parallel configuration in Fig. 5.4, the noise-network is additionally conditioned on the output of the harmonic network.*

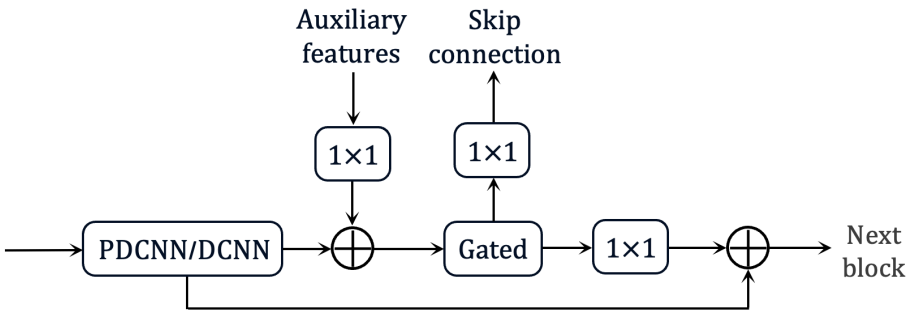

Figure 5.6: *A diagram of the macroblock in QP-PWG [37].*

inspired by the physiological mechanism through which humans control vocal power during phonation, enabling the model to represent both frequency-dependent spectral dynamics and temporal energy variations. To mitigate the effects of estimation errors in F_0 (fundamental frequency) and phase between generated and reference speech, a mel-filter-bank is applied to the amplitude spectra, and the loss is evaluated on the mel-frequency scale. This design allows the model to learn acoustically plausible excitation patterns under the source-filter assumption, while enhancing robustness against fine spectral and pitch mismatches.

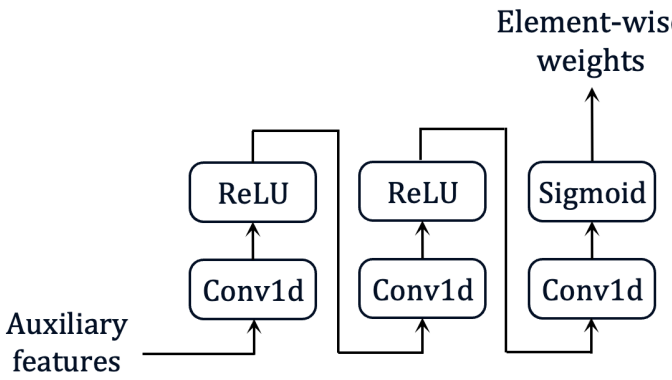

Figure 5.7: *A diagram of the periodicity estimator for hn-uSFGAN.*

5.2.2 F_0-Driven Source Excitation Generation

Source excitation signals have high periodicity owing to their generation process, which is based on vocal fold vibrations. Inspired by NSF [33, 103, 105], we input a sinusoidal-based signal to the generator generated by the same formula as that of NSF. The signal retains the input F_0 as the fundamental frequency, but with an additional random noise signal. Moreover, we apply PDCNNs, which effectively enlarge the receptive fields in accordance with the input F_0 by dynamically changing the DCNN dilation factors. We found that using both the sinusoidal input and PDCNNs significantly improves F_0 controllability. However, the PDCNNs also tend to introduce undesired periodic components to the unvoiced segments. This tendency prevents the proper generation of other aperiodic source components, such as frication, aspiration, and transient sources, which adversely affect speech quality and naturalness.

To improve the source excitation signal modeling, especially for the unvoiced parts, we introduce a harmonic-plus-noise excitation generation mechanism inspired by the current successful works [30, 34, 35, 103] based on [74]. To explicitly model the periodic and aperiodic components, previous works [30, 34, 35, 74, 103] prepared two networks

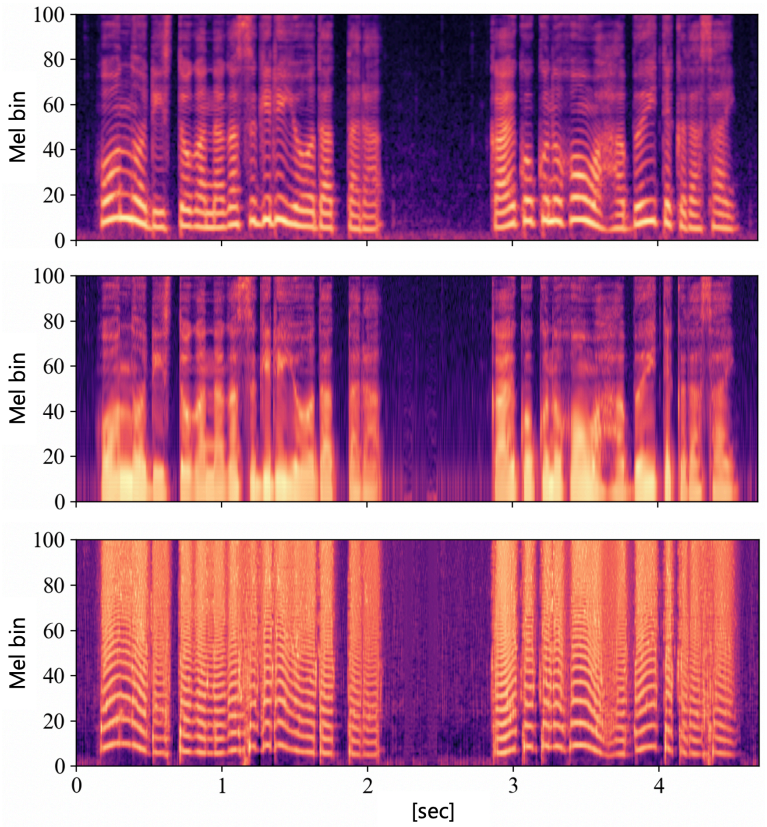

Figure 5.8: *Residual spectrogram on the mel-frequency scale used in the source regularization loss proposed in [104] (bottom). The residual spectrogram is obtained by subtracting the spectral envelope (middle) from the spectrogram of the target speech (top) in the logarithmic amplitude domain. The frame-wise power of the residual is then adjusted to match that of the target speech.*

for generating each component and devised the architecture and input features for each. We adopt two harmonic-plus-noise modeling schemes, the cascade and parallel model structures, referring to PeriodNet [34]. Hono et al. represent the dependence of the periodic and aperiodic speech signals on the model structure. The cascade model structure combines the periodic and aperiodic speech generators in series so that the latter generator can predict the aperiodic component, taking into account the dependence of the periodic component. On the other hand, the parallel model

structure assumes their independence. To ascertain whether the cascade or parallel structure scheme is superior in modeling the source excitation signal, we propose the two approaches following PeriodNet. Moreover, the periodicity estimation is crucial for the naturalness of generated speech. Regarding NHV [30] and HN parallel waveGAN (HN-PWG) [35], we prepare a network to estimate periodicity-related weights from acoustic features and mix periodic and aperiodic source components based on the weights.

The HN source excitation generation module consists of three networks: the harmonic network, noise network, and periodicity estimator, as shown in Fig. 5.4 and Fig. 5.5. The harmonic network outputs latent features $l^{(\mathrm{h})}$ that correspond to the periodic components of the source excitation signal from a sinusoidal signal and auxiliary features. On the other hand, the noise network outputs latent features $l^{(\mathrm{n})}$ that correspond to the aperiodic components of the source excitation signal from a random noise signal and auxiliary features. In the cascade approach (Fig. 5.5), the noise network also receives the output of the harmonic network. We use the QP-PWG macroblock shown in Fig. 5.6 in the harmonic network, while the PWG macroblock is used in the noise network. We adopt the harmonicity estimator of HN-PWG as the periodicity estimator shown in Fig. 5.7. Conditioned on the auxiliary features, the periodicity estimator outputs the channel-wise and sample-wise weights $\boldsymbol{a}$ within $[0, 1]$ corresponding to the speech periodicity. The two generated representations are summed element-wise using the estimated weights. The source excitation latent feature $\boldsymbol{l}$ is formulated as

$$l_{t,i} = a_{t,i} \cdot l^{(\mathrm{h})}_{t,i} + (1 - a_{t,i}) \cdot l^{(\mathrm{n})}_{t,i} \tag{5.3}$$

where the subscripts indicate the i^{th} channel of the t^{th} sample of each latent feature or weight. Since periodicity is estimated from auxiliary features, the input sinusoidal

signal is generated using the continuous F_0 values obtained by interpolating the discontinuous F_0 values.

The cascade approach comprises three steps, as shown in Fig. 5.5. First, the harmonic network outputs the periodic source excitation representation, which is modulated using the channel-wise weights predicted by the periodicity estimator. Second, a random noise signal is mapped to a latent representation and mixed with a periodic source representation using a 1x1 convolution layer and the noise network. Finally, the output latent feature of the noise network is modulated using the weights and summed up with the modulated periodic source excitation representation to output the final source excitation representation. On the other hand, in the parallel approach, the aperiodic source representation is generated without the output periodic source representation of the harmonic network.

5.2.3 Adversarial Training

The training procedure of uSFGAN is common for GAN-based training plus auxiliary regularization losses. The loss function of the generator can be written as the sum of the adversarial loss $\mathcal{L}_{\text{adv}}$, the auxiliary spectral loss $\mathcal{L}_{\text{spec}}$, and the regularization loss $\mathcal{L}_{\text{reg}}$:

$$\mathcal{L}_{\mathcal{G}} = \lambda_{\text{adv}}\mathcal{L}_{\text{adv}} + \lambda_{\text{spec}}\mathcal{L}_{\text{spec}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}, \tag{5.4}$$

where λ_{adv}, λ_{spec}, and λ_{reg} are loss balancing hyperparameters.

We adopt the least-squares GAN [64] loss as described in Eq. (3.19) and (3.20) for adversarial training. For the spectral loss, we compare two representative formulations: (1) the multi-resolution STFT (MR-STFT) loss introduced in Parallel Wave-

GAN (PWG) [8] Eq. (3.24), and (2) the mel-spectrogram loss adopted in HiFi-GAN [67] Eq. (3.28).

While the MR-STFT loss enforces a detailed match in the linear-frequency domain, it is often sensitive to small discrepancies in F_0 and phase, which can hinder convergence when the conditioning features are not perfectly aligned with the ground truth. In contrast, the mel-spectrogram loss provides a perceptually motivated frequency compression that alleviates the effects of pitch and phase mismatches, leading to more stable optimization and perceptually faithful synthesis. Because uSFGAN is conditioned on vocoder features such as F_0, spectral envelopes, and aperiodicity, such mismatches are inevitable; hence, the mel-spectrogram L1 loss serves as a more appropriate spectral constraint and is adopted as $\mathcal{L}_{\text{spec}}$ in our best-performing model. Moreover, with sufficiently strong and well-designed discriminators, adversarial learning itself encourages the generator to produce plausible phase information, reducing the need for the feature matching loss (Eq. (3.23)). Accordingly, unlike HiFi-GAN and MelGAN [66], uSFGAN does not employ the feature matching loss, which is also prone to instability under phase and F_0 mismatches.

Finally, as regularization losses, we consider and compare two complementary objectives: the flatten loss Eq.5.1 and the residual loss Eq.5.2. Their comparative effectiveness is discussed in the following section.

5.3 Experimental Evaluations

5.3.1 Data Preparation

We used the VCTK corpus [106], which contains 109 English speakers. We used only mic2 samples, and p315 was unavailable owing to a technical problem. The sampling

rate was set to 24 kHz using the sox[1] downsampling function. No preprocessing, such as normalization or low-cut filtering, was applied to the audio. We divided the dataset following a specific rule to evaluate robustness against unseen F_0 values. The minimum and maximum F_0 values of the VCTK corpus were respectively found to be about 50 Hz and 400 Hz through careful investigation of each speaker. We limited the F_0 range of the training data from 70 Hz to 340 Hz and excluded two speakers (p271 and p300) from the training data to evaluate the robustness of unseen speakers. Finally, nine speakers were excluded from the training data; therefore, the training data included 99 speakers. Thanks to this limitation, we can evaluate the methods using various conditions of seen or unseen speakers and F_0 ranges.

5.3.2 Model Details

Baseline Models

As baselines, we used the following four models.

- HiFi-GAN: A high-fidelity GAN-based neural vocoder with four multi-period discriminators and four multi-scale discriminators. HiFi-GAN has no clue for controlling F_0, so we used it as the baseline for the evaluation of speech reconstruction. To train the HiFi-GAN model, we adopted the HiFi-GAN V1 [67] configuration and used an unofficial open-source implementation[2] for training the model.

- WORLD: A conventional source-filter model. This model achieves flexible controllability of acoustic features with reasonable speech quality. We used a Python

[1] http://sox.sourceforge.net/
[2] An unofficial code of HiFi-GAN: `https://github.com/kan-bayashi/ParallelWaveGAN`

wrapper[3] of the original WORLD implementation[4].

- HN-NSF: Harmonic-plus-noise neural source-filter with time-variant and trainable sinc filters that predict their cut-off frequency from the input acoustic features. We reimplemented the model on the basis of the official open-source code[5] without changing the model configuration except for increasing the training iterations.

- QP-PWG: A F_0-controllable neural vocoder based on GAN without the source-filter separation. It controls F_0 via the PDCNNs and input auxiliary F_0. We increased the number of residual blocks from the original configuration: PDCNNs $10 \rightarrow 30$ and DCNNs $10 \rightarrow 30$. The capacities of the QP-PWG model and the basic uSFGAN model, detailed below, are the same regarding the number of residual blocks.

We conditioned the HiFi-GAN model by using the mel-spectrogram as the original model with 80 mel-filter-banks, 1024 fast Fourier transform (FFT) points, 1024 points of the Hanning window, and the hop size was set to 120 (5 ms). We trained it for 2500k iterations as the original model with the batch size set to 16, and the batch length set to 18000 (0.75 s), using the original setting of the Adam [107] optimizer. The loss weights followed the original setting. The weights of the adversarial loss, the feature matching loss, and the mel-spectral loss were set to 1.0, 2.0, and 45.0, respectively. HN-NSF was conditioned using discrete F_0, the mel-generalized cepstrum (MGC), and mel-cepstral aperiodicity (MAP). We trained it for 600k steps with the batch size set

[3]A Python wrapper of WORLD vocoder: `https://github.com/JeremyCCHsu/Python-Wrapper-for-World-Vocoder`

[4]WORLD official implementation `https://github.com/mmorise/World`

[5]NSF official Pytorch implementation: `https://github.com/nii-yamagishilab/project-NN-Pytorch-scripts`

Table 5.1: *Number of model parameters and real-time factors (RTF) calculated on a single GPU (Titan RTX 3090) and CPU with four threads (AMD EPYC 7302).*

Model	Parameters	RTF (GPU) ↓	RTF (CPU) ↓
HiFi-GAN	12.9 M	7×10^{-3}	0.83
HN-NSF	0.7 M	15×10^{-3}	1.49
QP-PWG	2.5 M	44×10^{-3}	4.26
uSFGAN	2.4 M	44×10^{-3}	4.38
C-uSFGAN	2.4 M	37×10^{-3}	3.99
P-uSFGAN	2.3 M	36×10^{-3}	4.19

to 1, as the original model, and the batch length was set to 24000 (1.0 s) using the original setting of the Adam optimizer. This model was trained using only the L2 loss on the log power spectrogram. QP-PWG was conditioned using almost the same features as those for HN-NSF, but continuous F_0 and a binary sequence representing voiced or unvoiced (VUV) segments were used instead of the discrete F_0. We trained it for 600k steps with the batch size set to 5 and the batch length set to 18000 (0.75 s) using the original setting of the RAdam [108] optimizer. The loss weights followed the original setting. The weights of the adversarial loss and the multi-resolution STFT loss were set to 4.0 and 1.0, respectively.

We extracted F_0 using the Harvest algorithm [45] with a carefully set F_0 search range for each speaker. Then we extracted the log power spectral envelope using the CheapTrick algorithm [46] and coded it into the corresponding 41-dimensional MGC with the all-pass-constant set to 0.466. Also, we extracted aperiodicity using the D4C algorithm [47] and coded them into the corresponding 21-dimensional MAP. These features were calculated with a shift period set to 5 ms. The mel-spectrogram was calculated using the librosa [109] function with the FFT size and window length set to 1024, and the hop length to 120 (5 ms) with a Hanning window.

Proposed Models

We used the following three uSFGAN-based models in the comparison experiments.

- uSFGAN: This model was based on our method proposed in [110]. The source network comprises 30 PDCNN blocks with six cycles, the filter network comprises 30 DCNN blocks with three cycles, and the PWG discriminator and PWG-based training procedure were used. The modifications are that the regularization loss became the L1 norm, and the input signal became a one-channel sinusoidal-based signal generated by the formula of NSF instead of a two-channel signal (a random noise signal and a sinusoidal-based signal without randomness). The updated loss leads to better performance of the objective metrics, and the input signal was for simplification of the comparison.

- C-uSFGAN (Cascade HN-uSFGAN): The first proposed model with the cascade harmonic-plus-noise excitation generation, the residual spectra targeting loss, the mel-spectral loss, and the HiFi-GAN discriminator. The harmonic network had 20 PDCNN blocks with four cycles, the noise network was composed of five CNN blocks without cycles, and the filter network was the same as that of the basic uSFGAN.

- P-uSFGAN (Parallel HN-uSFGAN): The second proposed model. The network architecture was the same as that of C-uSFGAN except for the parallel or cascade architecture. This model is based on that in [104], but we made several improvements to it. Specifically, continuous F_0 was removed from the auxiliary features, and the two-dimensional BAP was changed to the corresponding 21-dimensional MAP. More details about the feature choices are described in Appendix A. Moreover, empirically better loss weighting hyperparameters were used in this paper.

To enable the model to access the F_0 information only from the input sine waves, the auxiliary features included only MGC and MAP in all models. According to our preliminary experiments, this information restriction mechanism is essential for the proposed models to deal with excessively deviated F_0, such as the $2.0 \times F_0$ of female speakers with higher average F_0. The extractions of these acoustic features followed the same process as the baselines. The batch size and batch length of all the proposed models were set to 5 and 18000 (750 ms), respectively, as in the QP-PWG. The uSFGAN was trained with only the auxiliary losses for the first 100k iterations and with the discriminator in the remaining 500k steps using the RAdam optimizer with the same settings as those in QP-PWG. The loss weights were set based on those of QP-PWG: $\lambda_{\mathrm{adv}} = 4.0, \lambda_{\mathrm{spec}} = 1.0, \lambda_{\mathrm{reg}} = 1.0$. On the other hand, C-uSFGAN and P-uSFGAN followed the HiFi-GAN training procedure of simultaneously training the generator and the discriminators from scratch for 600k iterations using the Adam optimizer with the same setting as that in HiFi-GAN. The loss weights were set based on those of HiFi-GAN: $\lambda_{\mathrm{adv}} = 1.0, \lambda_{\mathrm{spec}} = 45.0, \lambda_{\mathrm{reg}} = 1.0$.

The model sizes of the baselines and proposed models are shown in Table 5.1. Their inference speeds are also detailed with the real-time factor (RTF) in the same table. As shown in the table, the proposed models are much smaller than HiFi-GAN with the V1 configuration, whereas HiFi-GAN achieves a much higher inference speed on a single GPU and CPU than the proposed models. HiFi-GAN adopts a configuration based on upsampling, where the preceding layers have lower temporal resolutions, resulting in higher computational efficiency and enabling fast waveform generation. On the other hand, the other models operate at a fixed temporal resolution consistent with the output waveform from the input. Since the computational complexity is proportional to the temporal resolution, these models tend to have slower speeds than

the upsampling-based approach.

Ablation Models

To investigate the effectiveness of each component in our best-proposed P-uSFGAN described above, we prepared the following four ablation models for the comparison experiments. The input features and the training procedure of the ablation models followed those of P-uSFGAN.

- Reg-Loss: P-uSFGAN trained with the spectral envelope flattening loss instead of the residual spectra targeting loss.
- HN-SN: P-uSFGAN without the parallel harmonic-plus-noise source network but with the generator of the basic uSFGAN (30 layers of PDCNNs).
- HiFi-D: P-uSFGAN without the multi-period or multi-scale discriminator of HiFi-GAN but with the discriminator of PWG. We set $\lambda_{\mathrm{adv}} = 8.0$ to match the reduced number of discriminators.
- Mel-Loss: P-uSFGAN trained with the multi-resolution STFT loss of PWG instead of the mel-spectral L1 loss. We set $\lambda_{\mathrm{spec}} = 20.0$ so that the loss values before and after the change have roughly the same magnitude.

5.3.3 Evaluation of Speech Reconstruction

To evaluate the robustness of the proposed models for unseen acoustic features, both objective and subjective tests were conducted for the speech reconstruction performances. That is, three evaluation sets, including natural acoustic features within, beyond, and below the F_0 training range, were adopted.

Table 5.2: *Results of objective evaluations of speech reconstruction. The best scores are in bold.*

method	RMSE ↓	VUV ↓	MCD ↓
	Within the training F_0 range ($70 - 340$ [Hz])		
WORLD	**0.05**	13	3.23
HiFi-GAN	0.07	10	3.37
HN-NSF	**0.05**	13	4.50
QP-PWG	0.07	11	**2.84**
uSFGAN	0.06	12	3.07
C-uSFGAN	**0.05**	**8**	2.86
P-uSFGAN	**0.05**	**8**	2.88
	Below the training F_0 range (> 50 [Hz])		
WORLD	0.10	20	3.28
HiFi-GAN	0.11	24	3.11
HN-NSF	0.10	27	4.45
QP-PWG	0.12	25	2.92
uSFGAN	0.10	22	2.98
C-uSFGAN	**0.09**	**18**	**2.69**
P-uSFGAN	**0.09**	20	2.85
	Beyond the training F_0 range (< 400 [Hz])		
WORLD	**0.06**	9	3.35
HiFi-GAN	0.08	8	3.60
HN-NSF	0.07	8	4.81
QP-PWG	0.09	9	3.07
uSFGAN	0.07	9	3.17
C-uSFGAN	**0.06**	**7**	2.91
P-uSFGAN	**0.06**	**7**	**2.87**

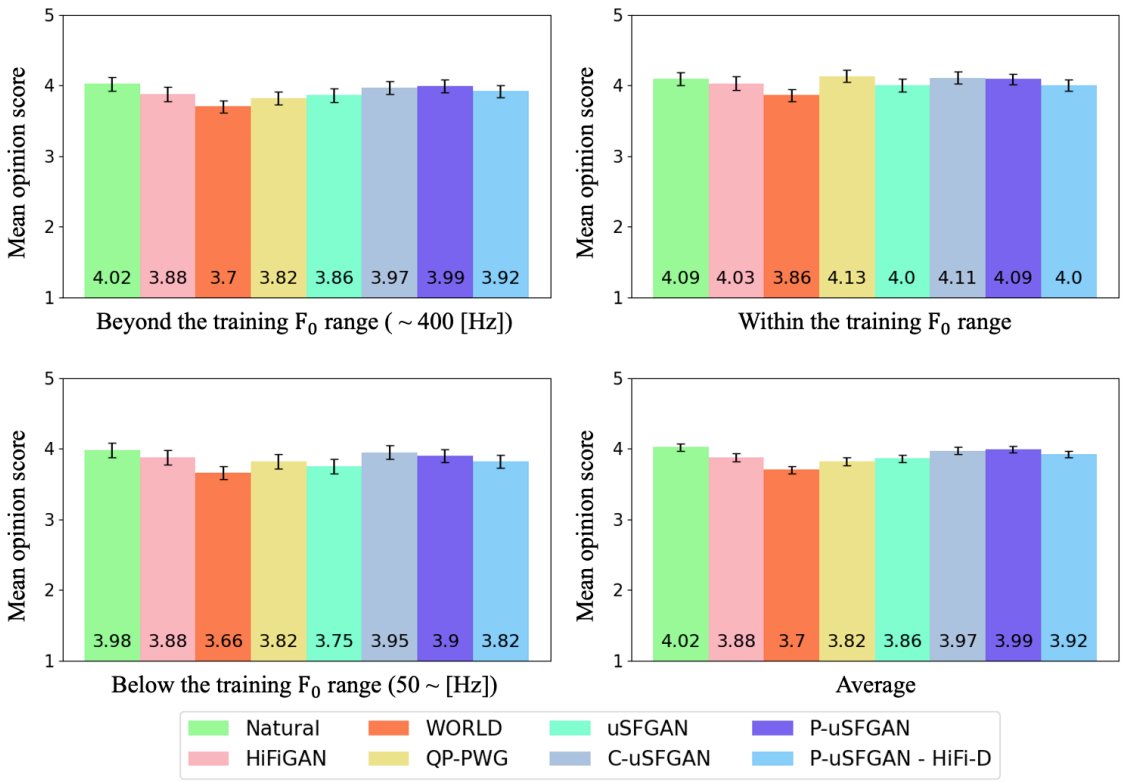

Figure 5.9: *Mean opinion scores (MOS) for different F_0 ranges. The upper-left, upper-right, and lower-left panels show the results for speech samples with fundamental frequencies beyond, within, and below the training F_0 range, respectively. The lower-right panel shows the average MOS values averaged over all F_0 ranges. Error bars indicate 95% confidence intervals.*

Objective evaluation

As the objective evaluation measurements, the root mean square error of log F_0 [Hz] (RMSE), the voiced or unvoiced decision error [%] (VUV), and mel-cepstral distortion [dB] (MCD) were used. The results are shown in Table 5.2, where the results are divided based on F_0 range. Each group included 200 utterances containing equal numbers of utterances by seen and unseen speakers. Since the primary purpose of our experiment was to investigate the F_0 robustness of the neural vocoders, and we confirmed that the proposed method did not cause significant degradation for unknown speakers [104], we

only report the evaluation results for all speakers together.

Conventional parametric vocoders such as WORLD usually achieve higher objective acoustic controllability than neural vocoders [37], and the results of objective evaluation also demonstrate the same tendency. Specifically, baseline neural vocoders suffer from degradation when unseen F_0 values were given, even though they partly outperform WORLD in the case of the seen F_0 range. In particular, QP-PWG shows large degradation in the VUV error rate for the F_0 range below the training range. On the other hand, uSFGAN, whose difference from QP-PWG is the explicit decomposition of the source and filter network and the input sinusoidal-based signal, does not show significant degradation in any case. This implies the benefit provided by the source-filter modeling, that is, an inductive bias for the speech production process leading to robustness to unseen F_0. Note that C-uSFGAN and P-uSFGAN show the best results in VUV error rate, greatly outperforming WORLD, indicating the effectiveness of the harmonic-plus-noise architecture and of updating the loss functions. In conclusion, the proposed methods attain acoustic controllability similar to or better than those of conventional parametric vocoders.

Subjective evaluation

For the subjective evaluation, we conducted an opinion test on speech quality using seven models and natural speech with ten subjects. Each subject evaluated 20 utterances per method. We recruited English-speaking evaluators through Amazon Mechanical Turk and instructed them to listen to the audio in a quiet room with headphones or earphones. Also, we filtered out scores from evaluators with unreasonable answers, such as where almost all scores were the same or the score of natural speech was lower than any system.

The results are shown in Fig. 5.9, where the results are divided on the basis of F_0 range. HN-NSF was clearly inferior to the other models in speech quality, so we excluded it in the subjective evaluation experiment because of the possibility of undesired bias that the other samples would be highly evaluated. HN-NSF is a very basic baseline, and we speculate that the degradation was due to the simplicity of the model architecture and its low capacity to adapt to the large number of speakers in the VCTK corpus. However, we did not conduct any hyperparameter tuning on HN-NSF and note that there is a possibility that its performance can be improved by increasing the number of layers or introducing adversarial training.

We can see that all models except for WORLD achieve comparable scores for natural speech. Interestingly, QP-PWG, which uses the discriminator of PWG, achieves the best score, outperforming HiFi-GAN. The reason for the improvement of QP-PWG from the original model would be the increase in the number of the generator layers ($20 \rightarrow 60$ residual blocks). However, for the unseen F_0 ranges, the proposed C-uSFGAN and P-uSFGAN achieve the best results, whereas QP-PWG is considerably degraded. Moreover, the differences between HiFi-GAN and natural speech become more prominent than in the case within the training F_0 range. On the other hand, there are no significant differences between C-uSFGAN and P-uSFGAN and natural speech in all cases. These results indicate that HiFi-GAN is data-driven and QP-PWG is highly data-driven. However, our proposed C-uSFGAN and P-uSFGAN complement the shortcomings of a data-driven approach.

5.3.4 Evaluation of F_0 Transformation

Next, we evaluated the performances of F_0 transformation with factors within $[2^{-1.0}, 2^{1.0}]$. The magnifications were taken equally on the logarithmic axis with the base at 2. The

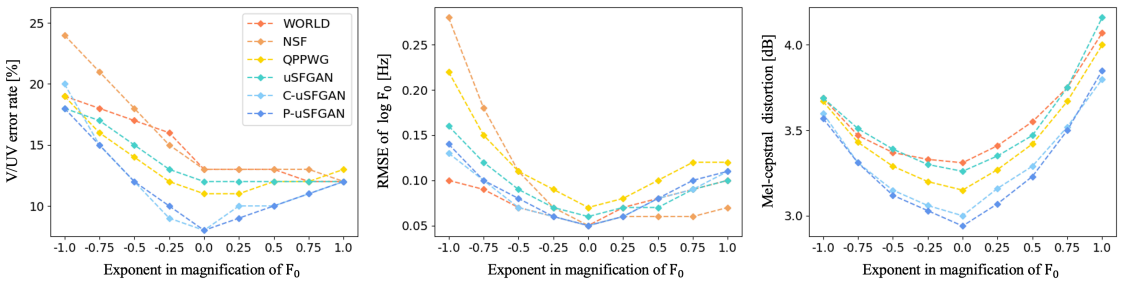

Figure 5.10: *Objective evaluation results of F_0 transformation for the comparison with baseline models. The MCD values of HN-NSF are excluded because they deviate from the range of the y-axis where the results of the other models are gathered.*

ground-truth F_0 was determined by multiplying the F_0 extracted from natural speech by the scale factors, and they were also adopted as the input F_0 of the models.

Objective evaluation settings

We extracted F_0 using the WORLD analyzer by the following procedure. When F_0 was multiplied by a scale factor greater than one, only the upper bound of the F_0 search range was multiplied and transformed; otherwise, only the lower bound of the range was multiplied and transformed. MCD was calculated using the CheapTrick [46] algorithm provided by the WORLD analyzer, and the extracted F_0 was used for the calculation. However, we downsampled audio signals to 16000 [Hz] before estimating spectral envelopes because the CheapTrick algorithm sometimes fails in the estimation when the F_0 adaptive window size is larger than the FFT size. We made the available fixed FFT size sufficiently large by reducing the size of the F_0 adaptive window through downsampling and calculated MCD more accurately. The evaluations were conducted using the evaluation data whose F_0 range was within the training F_0 range (i.e., $70-340$ [Hz]).

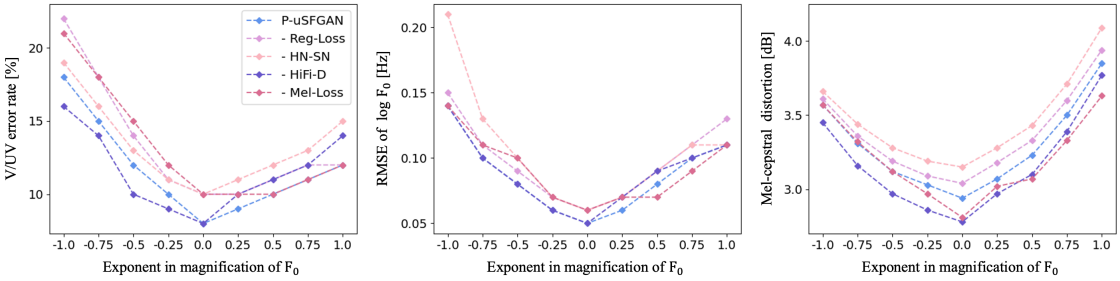

Figure 5.11: *Objective evaluation results of F_0 transformation for the ablation study.*

Objective evaluation

The objective evaluation results of comparison with baseline models are shown in Fig. 5.10. The result of log F_0 RMSE shows that although other models suffer from degradation in extreme cases ($F_0 \times \{2^{-1.0}, 2^{1.0}\}$), the proposed C-uSFGAN and P-uSFGAN models achieve stable values close to that of WORLD. However, the two models achieve much lower VUV error rates than all baseline models, which we found to have more impact on speech quality in our preliminary experiments. Moreover, we can see that all proposed models achieve better MCDs than WORLD. Again, the VUV error rate and the RMSE of log F_0 in QP-PWG degrade as the scale factor increases or decreases, respectively. In contrast, uSFGAN does not significantly degrade for any factor, indicating the benefit of the source-filter decomposition.

Ablation study

The objective evaluation results of the ablation study are shown in Fig. 5.11. From the results for P-uSFGAN and P-uSFGAN - HN-SN, we can see that the harmonic-plus-noise source network is very effective in improving the VUV error rate and RMSE of log F_0. Moreover, the residual spectra targeting loss (P-uSFGAN vs P-uSFGAN - Reg-Loss) and mel-spectral loss (P-uSFGAN vs P-uSFGAN - Mel-Loss) effectively improve

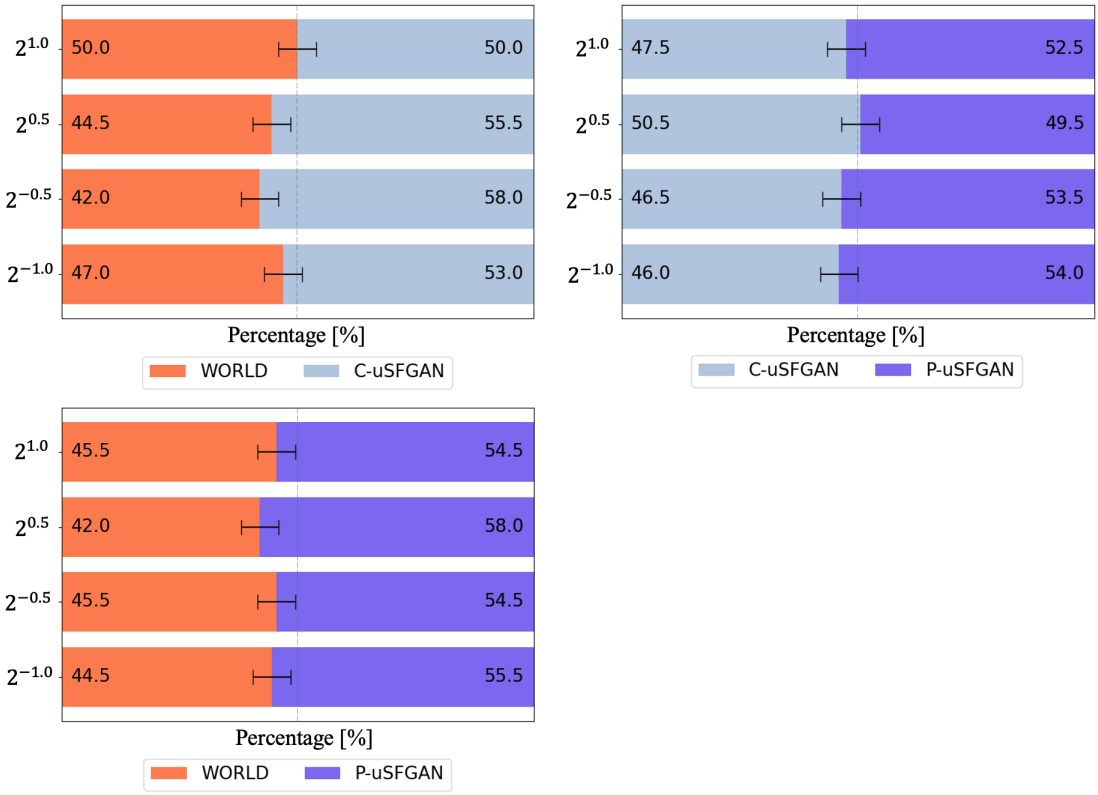

Figure 5.12: *Evaluation results of the preference test for F_0 transformation with the baseline WORLD and proposed C-uSFGAN and P-uSFGAN.*

the VUV error rate. Although P-uSFGAN - HiFi-D exhibits stable performance in the objective metrics, it is inferior to P-uSFGAN in terms of subjective speech quality, as shown in Fig. 5.9. This observation suggests that, in this ablation setting, the discriminator contributes more to perceptual speech quality.

Subjective evaluation

For the subjective evaluation, we conducted preference tests on speech quality using WORLD, C-uSFGAN, and P-uSFGAN for four F_0 scaling factors $\{2^{-1.0}, 2^{-0.5}, 2^{0.5}, 2^{1.0}\}$. Twenty subjects participated, and each subject evaluated ten pairs per F_0 scaling fac-

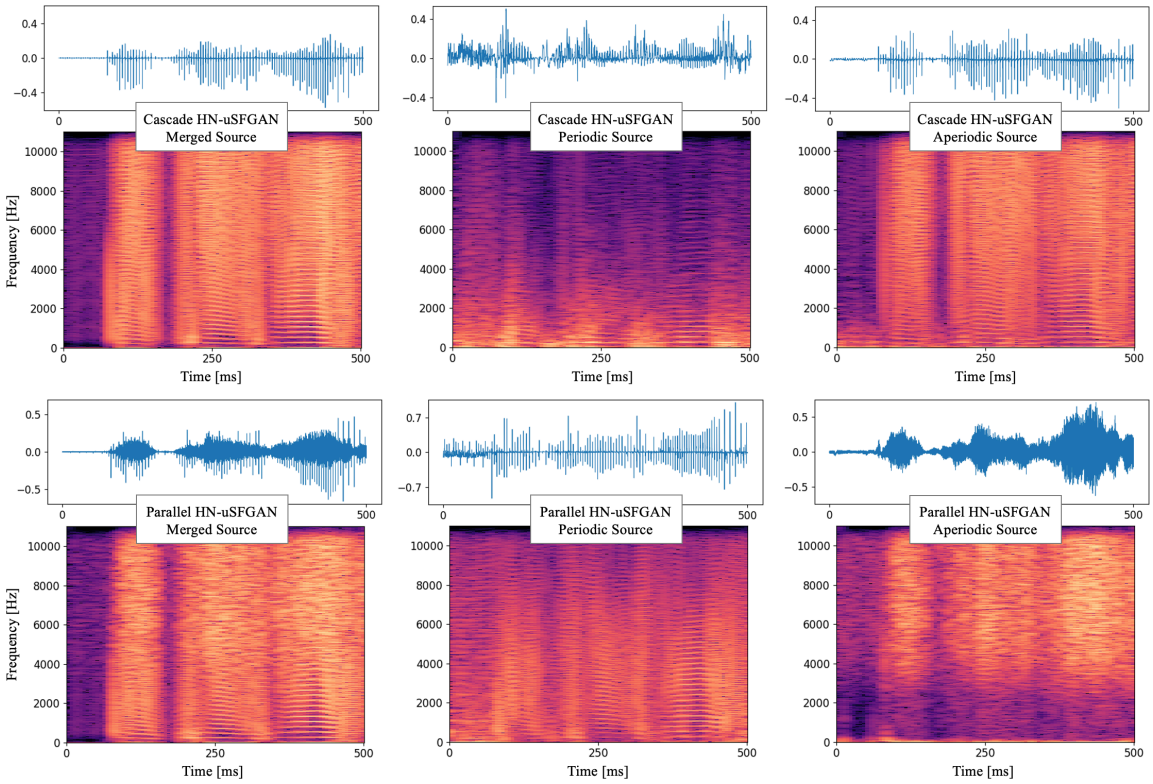

Figure 5.13: *Plots of output source excitation signals and spectrograms of C-uSFGAN (upper row) and P-uSFGAN (lower row) for 500 ms. The left column indicates the final source excitation signal, the middle column indicates the periodic source excitation signal, and the right column indicates the aperiodic source excitation signal.*

tor per method pair. The results are shown in Fig. 5.12. From the figures, both C-uSFGAN and P-uSFGAN outperform WORLD for all given F_0 scale factors, and P-uSFGAN is superior to C-uSFGAN in 3/4 of the items.

5.3.5 Visualization of output source excitation signals

To investigate the behavior of cascade and parallel HN-uSFGAN models (C-uSFGAN and P-uSFGAN), we visualized their output periodic and aperiodic source excitation signals in Fig. 5.13 with the spectrograms. These signals were obtained from the output

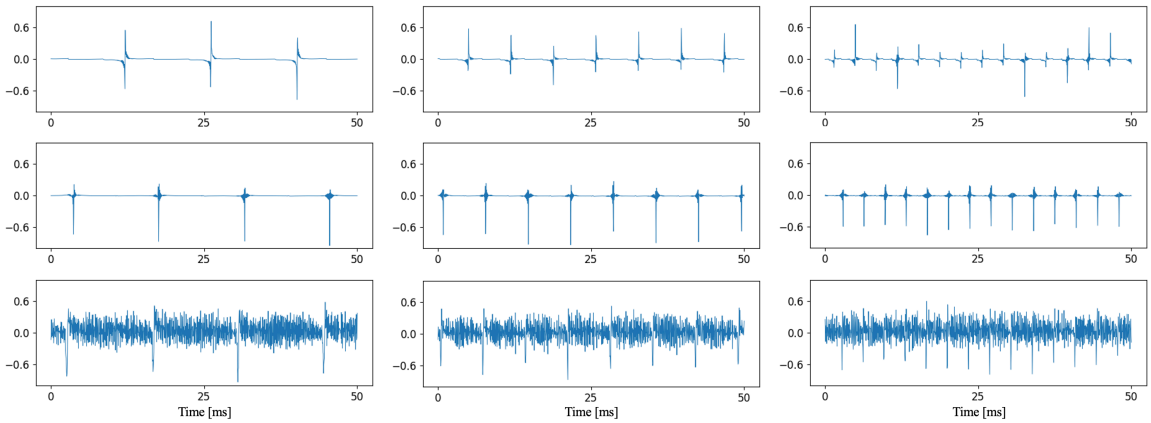

Figure 5.14: *Plots of output source excitation signals and spectrograms of uSFGAN, C-uSFGAN, and P-uSFGAN (from top to bottom row) with three F_0 scaling factors:* 0.5*,* 1.0*, and* 2.0 *(left to right column), for 50 ms. All of them were clipped from the same segment of the same utterance. The original F_0 values in this segment were around 140 [Hz].*

latent representations of $\boldsymbol{l}$, $\boldsymbol{l}^{(\mathrm{h})}$, and $\boldsymbol{l}^{(\mathrm{n})}$ using the output layers of the filter network and normalization of the signal power.

In Fig. 5.13, the output source excitation signals of C-uSFGAN seem to include fewer aperiodic components than in P-uSFGAN. Moreover, whereas P-uSFGAN well models the periodic and aperiodic components by the corresponding networks, C-uSFGAN does not seem to be able to disentangle these components. This indicates that the input aperiodic components are ignored as they pass through some networks. C-uSFGAN and P-uSFGAN achieve almost the same performance in speech reconstruction evaluation, as shown in Section 5.3.3. However, P-uSFGAN significantly outperforms C-uSFGAN in the evaluation of F_0 transformation, as shown in Section 5.3.4. From the results, we can conclude that the disentanglement of periodic and aperiodic components has a good effect on the speech quality in F_0 transformation scenarios. Thus, we choose P-uSFGAN as our best-proposed model in this work.

Furthermore, source excitation signals of uSFGAN, C-uSFGAN, and P-uSFGAN for

several F_0 scaling factors are plotted in Fig. 5.14. The figure shows that all proposed models can generate reasonable source excitation signals in accordance with the input F_0.

5.4 Conclusion

This chapter proposed a novel framework, unified source-filter GAN (uSFGAN), that functionally decomposes a single neural network into a source network and a filter network, using a regularization loss applied to the intermediate output. By jointly optimizing the source excitation and resonance filtering networks while incorporating the harmonic-plus-noise structure, the proposed method maintains speech quality comparable to high-fidelity neural vocoders, while achieving F_0 controllability comparable to the conventional source-filter model, WORLD [3]. The uSFGAN models are built upon PWG [8] as their backbone, taking advantage of the stable GAN-based training framework. However, since PWG is implemented as a convolutional neural network operating directly on long time-domain sequences, it is not suitable for real-time applications under low-resource environments. In the next chapter, we address this limitation by introducing a more efficient upsampling-based generator architecture (HiFi-GAN [67]) as the backbone, and integrating it with the unified source-filter framework.

6 Efficient Unified Source-Filter Modeling

This chapter proposes Source-Filter HiFi-GAN (SiFi-GAN), an extended framework that builds upon uSFGAN presented in the previous chapter. The structure of this chapter is as follows. Section 6.1 reviews the design philosophy and limitations of uSFGAN, clarifying the motivation behind the development of SiFi-GAN. Section 6.2 describes the architectural design of SiFi-GAN in detail. Section 6.3 presents experimental results, focusing primarily on singing-voice synthesis to evaluate the proposed method. Finally, Section 6.5 discusses the limitations of SiFi-GAN, outlines future research directions, and concludes with a summary and prospects of this study.

6.1 Introduction

In the first theme of this study, the Unified Source-Filter GAN (uSFGAN) and its extended model, the harmonic-plus-noise uSFGAN (hn-uSFGAN), were introduced. These models established an integrated framework in which the functional roles of excitation generation and resonance filtering emerge spontaneously within a single neural network through adversarial learning. By incorporating a source regularization loss, uSFGAN further clarified the functional separation between excitation and filtering processes in Quasi-Periodic Parallel WaveGAN (QP-PWG) [37], thereby enhancing

both interpretability and F_0 controllability. However, since QP-PWG and consequently uSFGAN operate at the same temporal resolution as the final waveform, their fixed-resolution design leads to computational inefficiency for long sequences, limiting the suitability for real-time and large-scale applications. Thus, although uSFGAN demonstrated high performance in terms of speech quality and controllability, it remained practically constrained by slow generation speed.

To overcome these limitations, this chapter introduces Source-Filter HiFi-GAN (SiFi-GAN), which integrates the unified source-filter modeling framework of uSFGAN into the hierarchical upsampling architecture of HiFi-GAN [67]. By combining source-filter regularization with a multi-stage upsampling strategy, SiFi-GAN achieves high-quality, controllable, and computationally efficient speech generation. In CNN-based neural vocoders, computational cost typically scales with sequence length. However, by adopting a hierarchical upsampling design, SiFi-GAN progressively expands low-resolution acoustic representations, allowing for large receptive fields while maintaining efficient computational scaling. This architecture effectively balances model capacity and computational efficiency, achieving higher speech quality and faster generation compared to fixed-resolution models such as uSFGAN.

Prior to SiFi-GAN, few extensions of HiFi-GAN have explored improving F_0 controllability by incorporating explicit F_0-driven mechanisms. For example, Period-HiFi-GAN introduces analytically generated sinusoidal signals to facilitate F_0 control, whereas Harmonic-Net [38] combines this approach with Pitch-Dependent Dilated CNNs (PDCNNs) [36,37] to achieve more robust F_0 controllability. Building upon these efforts, SiFi-GAN extends such F_0-driven frameworks by integrating a source-filter structure, analogous to the relationship between uSFGAN and QP-PWG. Figure 6.1 illustrates the conceptual position of SiFi-GAN relative to existing approaches. Struc-

turally, SiFi-GAN connects two subnetworks, the source-network and filter-network, in series within the generator, forming a sequential processing flow that mirrors the classical source-filter relationship in human speech production. To mitigate the computational overhead introduced by this additional subnetwork, redundant components from the base HiFi-GAN generator are pruned, enabling fast waveform generation without compromising speech quality. Experimental evaluations presented in Section 6.3 on singing-voice synthesis demonstrate that SiFi-GAN outperforms both HiFi-GAN and hn-uSFGAN in terms of perceptual quality and inference speed. These results confirm that the principles of unified source-filter modeling remain valid and effective even when extended to the upsampling-based generator architecture. Overall, SiFi-GAN establishes itself as a high-quality, efficient, and highly controllable neural vocoder, capable of real-time speech generation even on a single CPU.

6.2 Source-Filter HiFi-GAN

As illustrated in Fig. 6.2 and Fig. 6.3, two source-filter architectures are examined: the cascade and parallel models. In the cascade configuration (Fig. 6.2, the excitation representations produced by the source-network are first processed by downsampling CNNs and then added to the outputs of the corresponding upsampling layers in the filter-network. In contrast, the parallel configuration (Fig. 6.3 connects the source-network and filter-network at each temporal resolution without additional downsampling. Between these two architectures, the cascade model achieves superior performance, as demonstrated in the experimental results presented in Section 6.3. Therefore, we adopt the cascade model as our proposed method. Unless otherwise noted, we refer to it simply as SiFi-GAN throughout this dissertation.

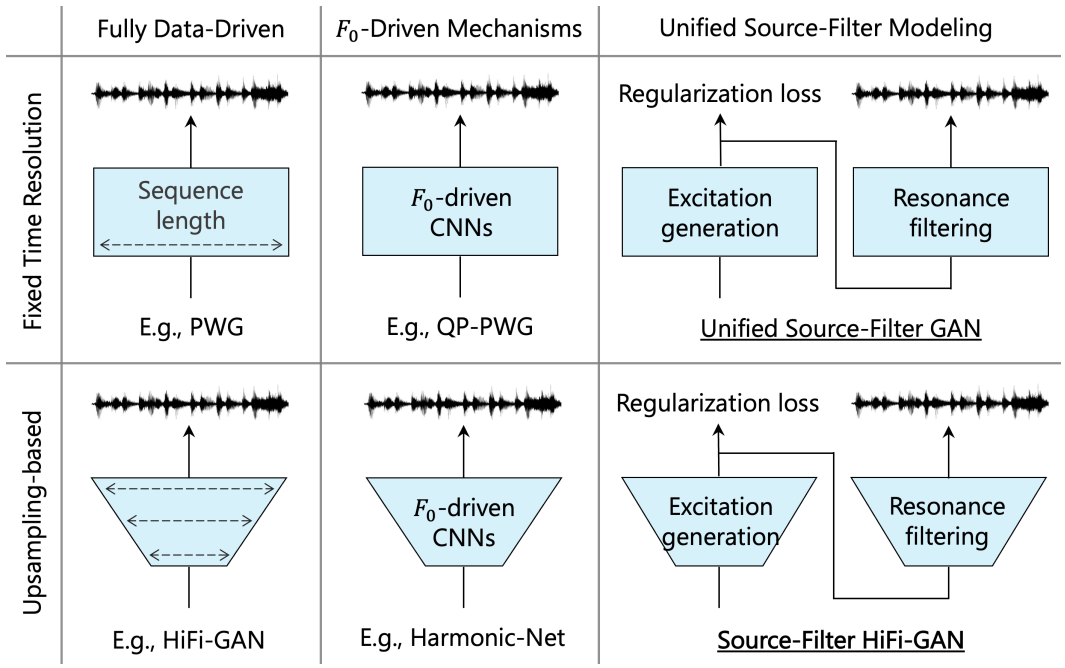

Figure 6.1: *Comparison of several GAN-based [9] neural vocoder architectures from the perspectives of temporal resolution, F_0-driven mechanisms, and source-filter modeling. Quasi-Periodic Parallel WaveGAN (QP-PWG) [81] and unified Source-Filter GAN (uSFGAN) [111], both based on Parallel WaveGAN (PWG) [8], operate at a fixed temporal resolution from input to output signals. In contrast, Harmonic-Net [38] and the proposed SiFi-GAN [77], which are based on HiFi-GAN [67], employ upsampling-based architectures that achieve higher synthesis efficiency relative to model capacity by processing features at lower temporal resolutions.*

6.2.1 Source excitation generation network

The source-network consists of three components: (1) downsampling layers based on one-dimensional (1D) convolutional neural networks (CNNs), (2) upsampling layers based on transposed 1D CNNs, and (3) the proposed quasi-periodic residual blocks (QP-ResBlocks). A sinusoidal waveform is analytically generated from an F_0 sequence following the method in [105], and then passed through the downsampling layers to obtain resolution-aligned periodic representations. These representations are added

to the outputs of the corresponding upsampling layers, following the approach of [38]. The stride settings in the downsampling layers are configured to match the upsampling ratios of the transposed convolutions. Each QP-ResBlock consists of multiple iterations of leaky ReLU [112], pitch-dependent dilated convolutional layers (PDCNNs) [80, 81], and 1D CNNs, with residual connections applied after each iteration. The number of iterations in each block corresponds to the number of customized dilation sizes d assigned to it. For each temporal resolution, the dense factors a [80] are carefully set considering the theoretically producible maximum frequencies of the corresponding PDCNNs. The dilation sizes d and dense factors a used in this study are detailed in Fig. 6.2, and all kernel sizes in the QP-ResBlocks are fixed to three.

The source regularization loss is computed from a single-channel signal obtained by applying a leaky ReLU and a 1D convolutional layer to the output of the final QP-ResBlock. This loss enforces excitation-like behavior in the projected signal, whereas the latent representation before projection can still encode richer information. The latent representation is then passed to the filter-network for final waveform generation. Note that F_0-dependent architectures are employed only in the source-network, encouraging a structural separation between excitation and vocal-tract filtering.

6.2.2 Resonance filtering network

The filter-network consists of three components: (1) upsampling layers based on transposed 1D CNNs, (2) multi-receptive field fusion (MRF) blocks of HiFi-GAN [67], and (3) additional downsampling CNNs. The downsampling CNNs are used only in the cascade structure to process the excitation representations from the final layer of the source-network. These downsampling CNNs are configured in the same manner as those used in the source-network. The final speech waveform is then generated by

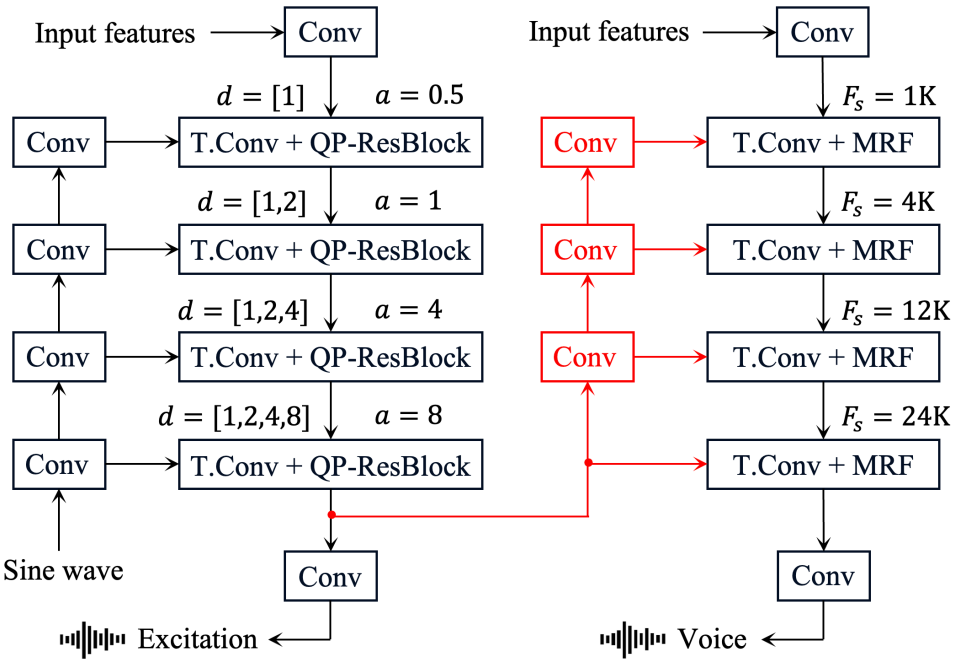

Figure 6.2: *Architecture of the proposed SiFi-GAN Cascade. The source-network and filter-network are connected in a cascaded manner: the output of the source-network is processed by a chain of downsampling layers and then passed to the filter-network, forming a clear cascade structure. Conv, T.Conv, QP-ResBlock, and MRF denote 1D convolution, transposed 1D convolution, quasi-periodic residual block, and multi-receptive fusion, respectively. d, a, and F_s denote the dilation sizes, the dense factor [36], and the sampling rate in Hz, respectively.*

passing the output of the filter-network through the same output layers as in HiFi-GAN.

Since SiFi-GAN employs two upsampling-based networks (i.e., the source- and filter-networks), it inevitably increases both computational cost and parameter count compared with the original HiFi-GAN (V1). To mitigate this increase and maintain generation efficiency, we adjusted the hyperparameters of the MRF modules. Specifically, the kernel sizes $\{3, 7, 11\}$ were reduced to $\{3, 5, 7\}$, which effectively reduces compu-

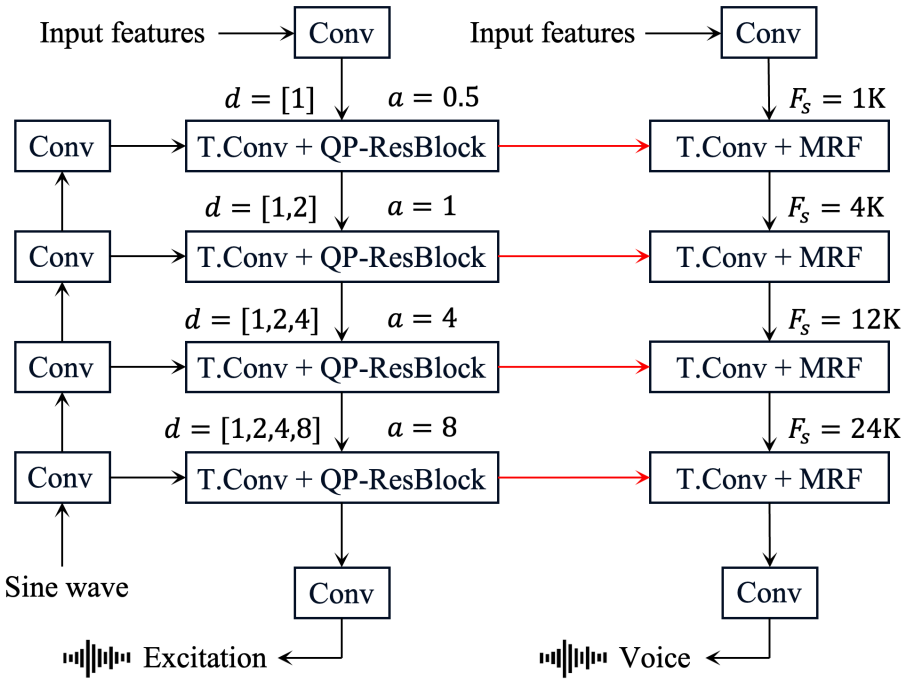

Figure 6.3: *Architecture of SiFi-GAN Parallel. The source-network and filter-network operate in parallel, and their corresponding layers are connected at each temporal resolution via feature-wise addition. That is, the intermediate outputs of the source-network are directly added to the outputs of the corresponding transposed convolution layers in the filter-network.*

tational complexity and the number of model parameters while preserving perceptual quality. We also removed the auxiliary CNNs that followed each dilated convolution layer in the MRF modules.

As described above, the proposed SiFi-GAN (i.e., the cascade model) employs a sequential connection between the source- and filter-networks, where the source output is passed through downsampling CNNs before being fed into the filter-network. Interestingly, this cascade strategy is essential for the high-frequency reproduction in the generated speech. The effectiveness of this design is further analyzed and demonstrated

in our ablation study in Section 6.3.

6.2.3 Training Criteria

The training procedure follows HiFi-GAN [67], except that the original feature matching loss [66, 67] is replaced with the regularization term defined in Eq. (5.2). The discriminator $\mathcal{D}$ and generator $\mathcal{G}$ are optimized using the least-squares GAN [64] described in Eq. (3.19) and (3.20). To ensure perceptual consistency between generated and target speech, the generator employs the mel-spectrogram loss $\mathcal{L}_{\text{spec}}$ defined in Eq. (3.28). The overall generator objective is formulated as

$$\mathcal{L}_{\mathcal{G}} = \mathcal{L}_{\text{adv}} + \lambda_{\text{spec}}\mathcal{L}_{\text{spec}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}, \tag{6.1}$$

where λ_{spec} and λ_{reg} are empirically set to 45.0 and 1.0, respectively.

Although the feature matching loss [66] is widely used for stabilizing GAN-based vocoders, we excluded it due to its adverse effects in this setting. Because neural vocoding is inherently an ill-posed problem, even slight phase mismatches between the generated and reference signals can result in large deviations in the discriminator's feature space [104], leading to over-penalization and mode-missing behavior [70, 113]. Empirically, removing this loss improved training stability and efficiency without degrading perceptual quality.

6.3 Experimental Evaluations

This section presents the performance of the proposed method. Because F_0 controllability is more required in SVS, we evaluated singing voices generated with different

F_0 scaling factors in the analysis-synthesis scenario.

6.3.1 Data Preparation

We used Namine Ritsu Database [114], a Japanese singing voice dataset of 110 songs with a single female singer. The total recording duration is about 4.35 hours. We split the songs using a ratio of 100/5/5 for training, validation, and test sets. Also, each song was split into small clips based on the rest notes in the musical scores. All data were downsampled from 44.1 kHz to 24 kHz and normalized to -26 dB. The input acoustic features were extracted using WORLD [3] with a 5 ms frame shift. Specifically, we used one-dimensional continuous F_0 (cF_0) [115], one-dimensional voiced/unvoiced binary flags (v/uv), 40-dimensional mel-generalized cepstral coefficients (mgc), and three-dimensional band aperiodicity (bap). All sinusoidal signals were deterministically generated from cF_0 with the generation metric of [105].

6.3.2 Model Details

We prepared three baseline models for comparison.

1) **WORLD** is a conventional source-filter vocoder [3] based on signal processing, providing high F_0 controllability with reasonable sound quality.

2) **hn-uSFGAN** denotes the harmonic-plus-noise uSFGAN [104], which achieves high levels of sound quality and F_0 controllability. This model was conditioned on {mgc, bap} and a sinusoidal signal derived from F_0.

3) **HiFi-GAN + Sine** refers to HiFi-GAN [67] conditioned on {cF_0, v/uv, mgc, bap} and the sinusoidal embedding obtained through downsampling CNNs [38], as in SiFi-

GAN. This model was used to examine the effectiveness and limitations of using only a sinusoidal input derived from F_0 for controlling F_0.

4) **HiFi-GAN + Sine + QP** extends HiFi-GAN + Sine by inserting QP-ResBlocks after each transposed CNN layer. These QP-ResBlocks were configured in the same way as those used in the source-network of SiFi-GAN, as illustrated in Fig. 6.3. Unlike SiFi-GAN, which consists of two lightweight HiFi-GAN-based subnetworks (source- and filter-networks) with reduced kernel sizes and fewer layers, HiFi-GAN + Sine + QP retains the full HiFi-GAN capacity while adding F_0-dependent mechanisms on top of it. Consequently, it has more parameters than SiFi-GAN despite containing only a single subnetwork. This model was designed to evaluate the effect of F_0-dependent mechanisms and to compare the performance when the same F_0-driven mechanism is applied within a single network versus distributed across source and filter subnetworks. Note that the configuration of HiFi-GAN + Sine + QP is roughly equivalent to that of [38].

Although including a vanilla HiFi-GAN without any F_0-driven mechanisms would be a natural baseline for comparison, it could not generate reasonable singing voices under F_0 transformation conditions. Therefore, it was omitted from the main experiments, but its generated samples are available on our demo page for reference.

As described in Section 6.2, we propose two variants of SiFi-GAN that differ in how the source-network and filter-network are connected: the cascade and parallel architectures. In **SiFi-GAN Cascade**, the output of the source-network is passed through downsampling layers and then fed into the filter-network, forming a sequential structure. In **SiFi-GAN Parallel**, on the other hand, the source excitation representations from each QP-ResBlock are directly added to the filter-network without any downsampling. Both models are conditioned on {mgc, bap}.

Table 6.1: *Results of objective and subjective evaluations. The MOS of the ground truth samples was* 3.99 ± 0.05 *(*$1.0 \times F_0$*). CPU and GPU rows show real-time factors (RTFs), indicating the synthesis speed relative to real time when running on an AMD EPYC 7542 (CPU) and GeForce RTX 3090 (GPU), respectively. Params denotes the number of model parameters (trainable weights). The RTFs were computed using 108 clips (total duration: 878 s). The rightmost column corresponds to the proposed method, SiFi-GAN Cascade.*

Metrics	WORLD	hn-uSFGAN	HiFi-GAN + Sine	HiFi-GAN + Sine + QP	SiFi-GAN Parallel	**SiFi-GAN Cascade**
Copy Synthesis ($1.0 \times F_0$)						
VUV ↓	4	3	2	2	2	2
RMSE ↓	0.09	0.06	0.06	0.06	0.06	0.06
MOS ↑	2.98 ± 0.07	3.55 ± 0.05	3.86 ± 0.05	3.73 ± 0.05	3.78 ± 0.05	$\mathbf{3.90 \pm 0.05}$
F_0 transformation ($0.5 \times F_0$)						
VUV ↓	5	3	4	4	5	4
RMSE ↓	0.10	0.09	0.08	0.09	0.10	0.08
MOS ↑	2.63 ± 0.08	2.97 ± 0.07	2.95 ± 0.06	3.24 ± 0.06	2.91 ± 0.06	$\mathbf{3.29 \pm 0.06}$
F_0 transformation ($2.0 \times F_0$)						
VUV ↓	7	6	19	26	13	10
RMSE ↓	0.11	0.11	0.22	0.20	0.18	0.13
MOS ↑	2.74 ± 0.08	$\mathbf{3.23 \pm 0.07}$	2.69 ± 0.08	2.61 ± 0.09	2.87 ± 0.07	3.05 ± 0.08
CPU ↓	–	3.97	0.88	1.13	0.73	0.74
GPU ↓	–	1.9×10^{-1}	3.0×10^{-3}	8.6×10^{-3}	6.2×10^{-3}	6.2×10^{-3}
Params ↓	–	2.3×10^{6}	14.4×10^{6}	15.1×10^{6}	9.7×10^{6}	11.3×10^{6}

The upsampling ratios of the transposed CNNs were set to 5, 4, 3, and 2 for HiFi-GAN and SiFi-GAN models. The UnivNet multi-period and multi-resolution discriminators [68] were adopted for all vocoders because of the effectiveness of the high-frequency component generations. We trained all neural vocoders for 400 K steps with the same training settings as HiFi-GAN [67]. The batch size was 16, and the sequence length was 8400. However, since the training of hn-uSFGAN takes longer than HiFi-GAN-based models, hn-uSFGAN was trained with a minibatch size of six.

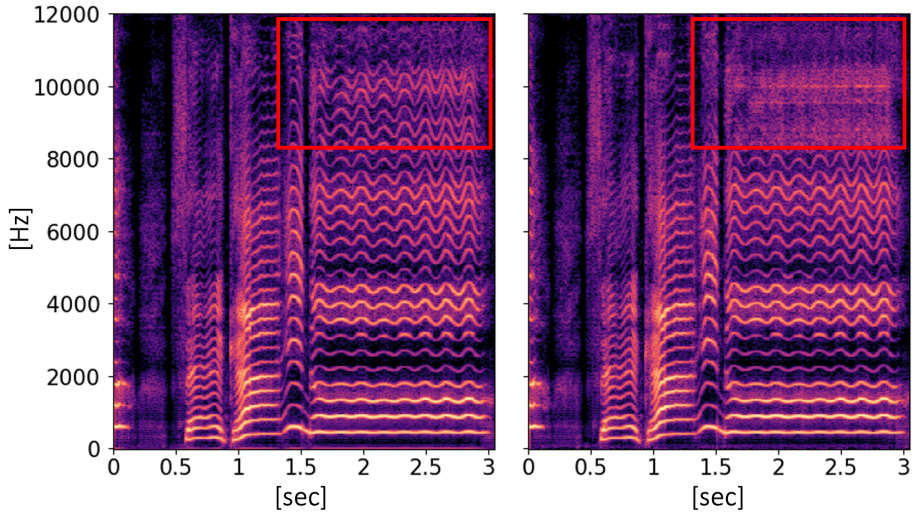

Figure 6.4: *Spectrograms of output singing voices from SiFi-GAN Cascade (left) and SiFi-GAN Parallel (right), respectively. This figure is adapted from [77].*

6.3.3 Evaluation metrics

To assess the performance of the proposed method, both efficiency and F_0 controllability were evaluated using objective measures. Synthesis efficiency was assessed by the real-time factor (RTF) and the number of model parameters. RTF is defined as the ratio of the synthesis time to the length of the generated waveform, where lower values indicate faster synthesis. RTFs on both a single CPU and a single GPU were reported to examine computational efficiency under different hardware conditions. F_0 controllability was evaluated using the root mean square error (RMSE) of log F_0 and the voiced/unvoiced decision error rate (VUV). The RMSE of log F_0 measures the accuracy of F_0 generation by comparing the synthesized and target F_0 trajectories on a logarithmic scale. It was computed only over frames judged as voiced in both the reference and synthesized signals; therefore, large VUV errors can indirectly affect the RMSE values. The VUV error rate represents the frame-wise disagreement in voiced/unvoiced classification between the synthesized and reference signals. The

F_0 and VUV sequences were extracted using the Harvest algorithm [45]. In addition, five-point mean opinion score (MOS) tests were conducted with 20 Japanese listeners to evaluate the perceptual quality of the synthesized singing voices. Each listener evaluated twelve samples for each method and F_0 scaling factor.

6.3.4 Evaluation Results

All results are summarized in Table 6.1. First, when comparing the ablation models (HiFi-GAN and SiFi-GAN variants), the proposed SiFi-GAN Cascade achieves the lowest RMSE and VUV error rates across all F_0 conditions and the highest MOS scores, indicating its superior F_0 controllability and perceptual quality. In particular, when F_0 is scaled by a factor of 2.0, the HiFi-GAN models exhibit substantial degradation in all metrics, revealing the limitation of relying solely on pitch-dependent mechanisms. Meanwhile, SiFi-GAN Parallel also suffers from noticeable degradation when F_0 is scaled to 0.5 or 2.0 times, whereas SiFi-GAN Cascade successfully maintains stable performance across all F_0 conditions. This indicates that explicitly incorporating the sequential source-filter structure into the model contributes to improved F_0 controllability. As shown in Fig. 6.4, even under the $F_0 \times 1.0$ condition, SiFi-GAN Cascade preserves clear high-frequency components, whereas SiFi-GAN Parallel exhibits smeared high-frequency regions. In addition, the SiFi-GAN models achieve faster synthesis with fewer parameters than the HiFi-GAN models. Notably, SiFi-GAN (RTF = 0.73) synthesizes faster on the CPU than the vanilla HiFi-GAN without sinusoidal embeddings or QP-ResBlocks (RTF = 0.84). These results suggest that the improvement in controllability achieved by the source-filter architecture outweighs the potential degradation caused by parameter pruning (see Section 6.2.2) for faster synthesis.

Next, when compared with the baseline models, the proposed SiFi-GAN Cascade

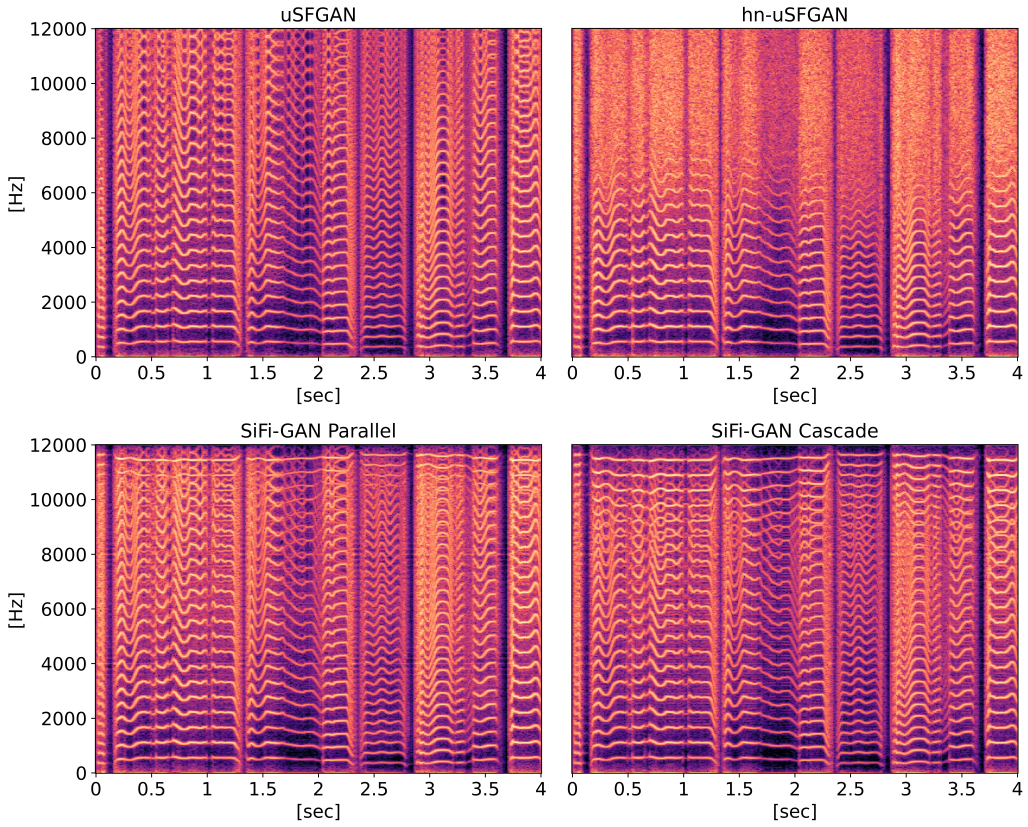

Figure 6.5: *Spectrograms of the source excitation signals generated by uSFGAN, hn-uSFGAN, SiFi-GAN Parallel, and SiFi-GAN Cascade. When converted to the mel-frequency scale, these spectrograms exhibit a similar appearance to the residual spectrogram shown in Fig. 5.8.*

achieves RMSE and VUV scores comparable to those of WORLD and hn-uSFGAN under $F_0\times$ 0.5 and 1.0 transformation conditions. Furthermore, SiFi-GAN obtained higher MOS scores under these conditions, indicating its superior perceptual quality. However, when F_0 is scaled by a factor of 2.0, SiFi-GAN shows inferior scores compared with hn-uSFGAN. The reason for this degradation will be discussed later in this section, where the negative effects of SiFi-GAN's resampling architecture are quantitatively analyzed. In terms of synthesis speed, the upsampling-based generator of SiFi-GAN

Cascade greatly improves efficiency compared with hn-uSFGAN, as evidenced by lower RTFs on both CPU and GPU. Overall, these results confirm that SiFi-GAN successfully combines the benefits of upsampling-based and source-filter architectures, achieving a balance between high audio quality, F_0 controllability, and computational efficiency.

6.3.5 Excitation Signal Analysis

To gain a deeper understanding of the model behavior, we conducted a qualitative analysis by visualizing the spectrograms of the source excitation signals generated by uSFGAN [111], hn-uSFGAN, SiFi-GAN Parallel, and SiFi-GAN Cascade. The resulting spectrograms are shown in Fig. 6.5. Here, uSFGAN is a simpler variant of hn-uSFGAN that does not employ the H+N structure. The vanilla uSFGAN shares similarities with SiFi-GAN in that it uses sinusoidal excitation inputs and PDCNN-based source-networks. However, unlike SiFi-GAN, uSFGAN adopts WaveNet-style [4, 8] residual blocks and does not employ the upsampling strategy like MelGAN [66] and HiFi-GAN. This architectural difference allows for a clearer comparison of the effects introduced by the H+N structure and upsampling operations.

First, we compared the uSFGAN and SiFi-GAN models to investigate the influence of the upsampling layers. The excitation signals generated by these models exhibit characteristics similar to residual signals, which is consistent with the design of the regularization loss (see Eq. (5.2)) based on residual spectrograms. They clearly display harmonic structures extending into the high-frequency region; however, aliasing artifacts appear as mirrored harmonic patterns. This aliasing effect is more pronounced in the SiFi-GAN models, suggesting a potential negative influence introduced by the upsampling layers.

In contrast, the high-frequency region of hn-uSFGAN is dominated by noise compo-

nents, unlike the other two models. This observation suggests that the H+N structure makes the high-frequency region noise-dominant, thereby preventing aliasing artifacts from propagating to the subsequent resonance filtering stage. Building on this observation, incorporating the H+N structure into SiFi-GAN could potentially mitigate aliasing and improve F_0 controllability. However, incorporating the H+N structure introduces additional network components, which inevitably increase computational complexity and model size, and therefore, its adoption should be carefully considered. Developing methods that can exploit the advantages of the H+N structure while maintaining the computational efficiency of SiFi-GAN is a potential direction for further study.

6.4 Limitations

While upsampling-based generator architectures enable efficient waveform generation, they exhibit two major drawbacks: aliasing and artifacts. We examine these issues below in terms of their causes and perceptual impact.

(1) Aliasing Artifacts: One significant drawback of upsampling is aliasing. In the $F_0 \times 2.0$ condition, SiFi-GAN performed worse than hn-uSFGAN, despite its superior performance in other settings. We attribute this degradation to aliasing artifacts introduced by the upsampling process in the generator (Fig. 6.6). When neural network latent representations are viewed as time series, any resampling operation can introduce aliasing distortion. Interpreting latent representations as time-domain signals, upsampling becomes equivalent to resampling, which introduces aliasing unless proper filtering is applied. SiFi-GAN employs transposed convolutions for upsampling, but unless these operations closely approximate an ideal low-pass filter, or are followed by appropriate filtering, aliasing is inevitable. In practice, aliasing can be suppressed

to some extent within the F_0 range covered during training, as the network learns to adapt its convolutional filters. However, the compensation is limited to the F_0 range seen during training and does not generalize well to higher, unseen F_0 values [117]. In such cases, aliasing becomes more perceptible due to the concentration of spectral energy in high-frequency harmonics. According to Parseval's identity, the total energy is preserved across time and frequency domains; thus, when harmonics shift upward with increasing F_0, more energy is allocated to high-frequency components that are subject to aliasing. Since aliasing reflects high-frequency components back into the audible range, this concentration of energy makes artifacts especially prominent for high-F_0 signals, even when the overall amplitude remains constant.

(2) Fixed-Frequency Artifacts: Another persistent issue arising from upsampling is the presence of fixed-frequency artifacts. These manifest as subtle noise components with concentrated energy at specific, unchanging frequencies in the generated waveform. Although it is known that replacing transposed convolutions with alternative upsampling methods, such as nearest-neighbor interpolation or sub-pixel convolution, can mitigate these artifacts [84], they could not eliminate these artifacts in our internal experiments. While often imperceptible in a single generated waveform, these artifacts become problematic in scenarios that involve waveform superposition, such as choral synthesis or the mixing of multiple vocal or instrumental tracks. Due to the linearity of the Fourier transform, fixed-frequency components introduced by each waveform accumulate linearly when summed.

As shown in Fig. 6.7, when multiple generated samples are combined using a singing voice synthesis system [116], prominent noise bands appear at regular 100 Hz intervals across the spectrum. This spacing is likely related to the 100 Hz frame rate of the input acoustic features. In such cases, the accumulated noise can noticeably affect

the spectral characteristics of the result, leading to audible artifacts that hinder the use of neural vocoders, including SiFi-GAN, in waveform superposition scenarios. We also confirmed that these artifacts appear even when ground-truth acoustic features are used instead of predicted ones, reinforcing that they are independent of prediction errors in the acoustic model.

6.5 Conclusion

In the previous chapter, uSFGAN demonstrated excellent performance in both speech quality and controllability; however, its generation speed was insufficient, limiting its applicability to real-time processing and large-scale deployment. To address this limitation, we designed a new framework that integrates unified source-filter modeling into the hierarchical upsampling architecture of HiFi-GAN, thereby achieving a better balance between efficiency and quality. The proposed Source-Filter HiFi-GAN (SiFi-GAN) aims to simultaneously satisfy the three essential requirements of neural vocoders: high speech quality, controllability, and fast generation speed. It achieves high audio fidelity and strong F_0 controllability while maintaining efficient inference, making it more suitable for integration into end-to-end speech generation systems such as TTS and SVS, as well as deployment in low-resource environments. However, many time-domain neural vocoders, including SiFi-GAN, suffer from non-physical distortions such as aliasing and fixed-frequency noise due to their use of nonlinear operations and upsampling. To fundamentally overcome this limitation, the next chapter introduces a new neural vocoder, Wavehax, which structurally avoids such distortions by eliminating nonlinear operations and upsampling in the time domain. Instead, Wavehax performs operations in the time-frequency domain, a redundant representation of the time signal.

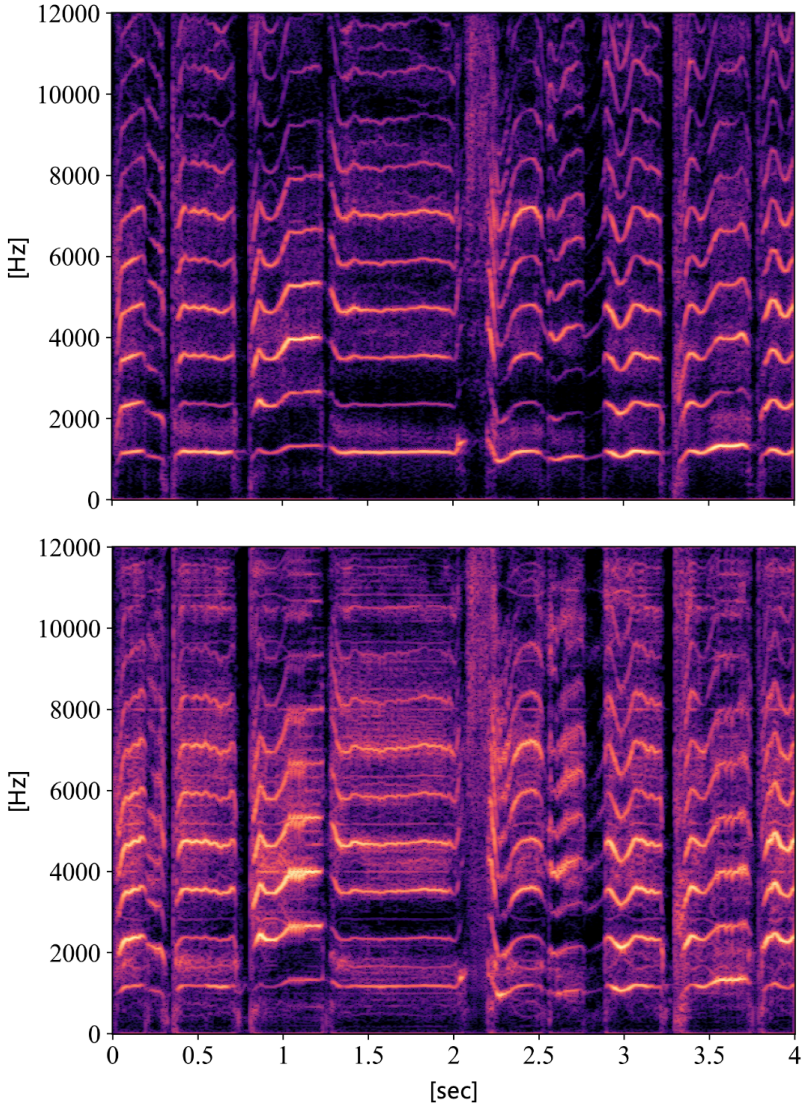

Figure 6.6: *Spectrograms of $F_0 \times 2.0$ speech generated by hn-uSFGAN (top) and SiFi-GAN (bottom). Aliasing artifacts are clearly observed in the SiFi-GAN output, which are likely caused by the upsampling process.*

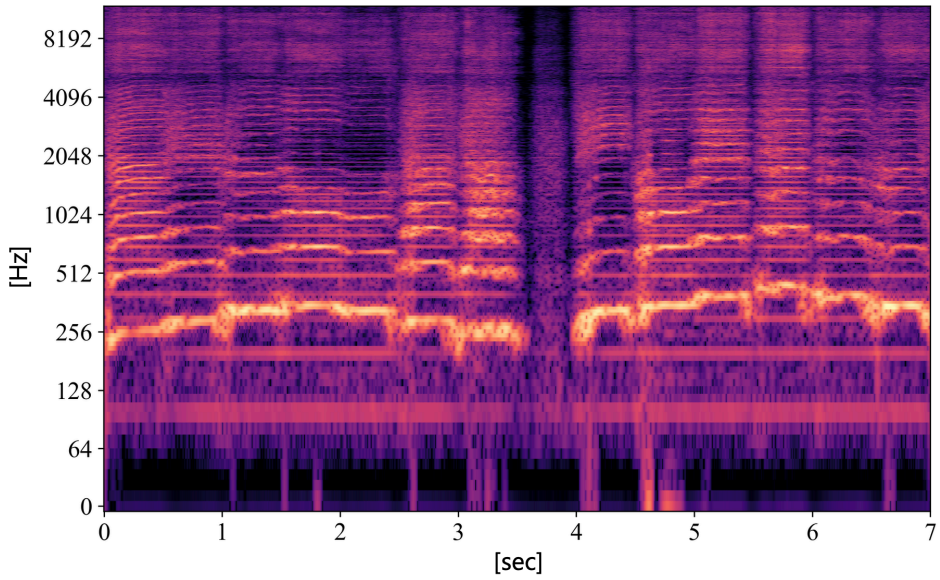

Figure 6.7: *Log-amplitude Mel-spectrogram of 100 superposed waveforms generated using neural network-based singing voice synthesis (NNSVS) [116] with the SiFi-GAN vocoder. Clearly visible noise bands occur at regular frequency intervals, likely due to fixed-frequency artifacts. Each waveform was generated using a different random seed, leading to slight variations across samples, such as in timing, F_0, and timbre.*

7 Aliasing-Free Neural Waveform Synthesis

This chapter focuses on the signal-processing inconsistency that arises between the continuous-time nature inherent to human speech production and the discrete-time processing of neural vocoders. This chapter is organized as follows. Section 7.1 describes the background and motivation of this research theme and its positioning within prior work. Section 7.2 provides a theoretical analysis of aliasing in neural vocoders from the perspective of signal processing. Section 7.3 explains the design philosophy, architecture, and training methodology of the proposed model, Wavehax. Sections 7.4 and 7.5 evaluate the performance of Wavehax through different experimental settings. Finally, Section 7.6 summarizes this chapter.

7.1 Introduction

In the previous chapters, we investigated the functional perspective of the speech production process based on the source-filter theory. Chapter 5 presented the Unified Source-Filter GAN (uSFGAN), which realizes the functional separation of source and filter components within a single neural network by means of a regularization loss and adversarial learning. Chapter 6 further extended this framework into a hierarchical upsampling architecture and introduced the Source-Filter HiFi-GAN (SiFi-GAN), which

achieves an improved balance among speech quality, F_0 controllability, and generation efficiency. This chapter addresses a different issue, namely the mismatch between the continuous-time nature of speech production and the discrete-time computation performed by neural vocoders. Neural vocoders operate on discrete-time signals using nonlinear and resampling operations, which introduce aliasing that is absent in natural speech production. Aliasing causes high-frequency components to fold into the lower-frequency band, producing artificial spectral distortions and breaking the shift equivariance property of convolutional neural networks. These artifacts represent a fundamental form of physical inconsistency arising from the discrepancy between continuous acoustic phenomena and their discrete computational representations.

In particular, nonlinearity is an essential component of deep neural networks, and its removal inevitably limits their representational capacity. Several studies have therefore explored methods to mitigate aliasing artifacts caused by nonlinear operations. For instance, BigVGAN [11] alleviates aliasing through temporal oversampling, which extends the spectral bandwidth during nonlinear processing. JenGAN [39], on the other hand, indirectly suppresses aliasing by randomly shifting input and output signals during training, thereby promoting shift equivariance. Although CNNs tend to learn to suppress aliasing implicitly during training, this effect is often data-dependent and becomes less effective under unseen or out-of-distribution conditions, where the aliasing effect becomes more pronounced. Consequently, establishing a principled framework that can inherently avoid aliasing remains an open challenge. A separate line of research combines parametric source excitation modeling with linear vocal-tract filtering, as discussed in Section 4.3. Although such approaches can theoretically avoid aliasing, the linearity assumption imposed on the vocal-tract filter considerably limits the model's expressive capacity.

In this chapter, we address these challenges by examining the internal mechanisms of neural vocoders from a signal-processing perspective, with the goal of designing architectures that preserve nonlinearity while maintaining consistency between continuous and discrete representations. We begin by analyzing signal representations and reinterpreting CNN operations through the lens of signal processing. This analysis reveals that, when speech is represented in the time-frequency domain, CNN operations do not cause the spectral folding that typically arises when processing the speech waveform directly in the time domain. This insight motivates a framework in which neural speech waveform generation is formulated as the modeling of complex spectrograms, a redundant representation of the time-domain signal, followed by reconstruction into the time domain via the inverse Short-Time Fourier Transform (iSTFT). At the same time, another important observation emerges: CNN-based modeling in the time-frequency domain is generally less stable in constructing the harmonic structure of speech waveform compared to time-domain approaches. Building on these findings, this chapter introduces Wavehax, a neural vocoder that integrates analytical harmonic modeling with nonlinear filtering in the time-frequency domain. Wavehax applies two-dimensional (2D) convolution to latent representations of complex spectrograms while incorporating a harmonic prior, enabling reliable complex spectrogram generation. The harmonic prior, derived from an analytical harmonic signal model, compensates for the lack of harmonic inductive bias in time-frequency domain processing and contributes to robust synthesis even under out-of-distribution conditions.

Importantly, this study differs from previous time-frequency-domain vocoders, reviewed in Section 4.6, which primarily focused on computational efficiency. In contrast, Wavehax is driven by the objective of achieving aliasing-free waveform synthesis and is therefore compatible with the physical speech production process, demonstrating

both theoretical advantages and practical effectiveness. Experimental results show that Wavehax attains speech quality comparable to existing high-fidelity neural vocoders while exhibiting exceptional robustness in high-F_0 extrapolation scenarios, where aliasing effects typically become severe. Moreover, Wavehax requires less than 5% of the multiply-accumulate operations and parameters of HiFi-GAN V1, a widely used high-quality neural vocoder, and achieves more than a fourfold speedup in CPU inference. Taken together, these properties make Wavehax suitable for real-time speech synthesis even in low-resource environments, highlighting the practical benefits of the proposed aliasing-free vocoder design.

7.2 Theoretical Background

This section discusses the fundamental properties of time, frequency, and time-frequency domain operations in neural vocoders from a signal-processing viewpoint, particularly focusing on aliasing.

7.2.1 Time-domain processing

We start by discussing time-domain convolution and nonlinear operations, focusing on why nonlinear operations in the time domain, such as the ReLU activation function [82], cause aliasing. We then examine the anti-aliased nonlinear operation [10, 11] and highlight their limitations.

Time-domain convolution

Let $\boldsymbol{x} \in \mathbb{R}^N$ be a discrete-time signal defined at each time step $n = 0, 1, \ldots, N-1$. We denote the element at time step n as $\boldsymbol{x}[n]$. The frequency representation of $\boldsymbol{x}$ is given by $\mathcal{F}\{\boldsymbol{x}\} \in \mathbb{C}^N$, which is obtained via the discrete-time Fourier transform (DFT) $\mathcal{F}$ as follows:

$$\mathcal{F}\{\boldsymbol{x}\}[k] = \sum_{n=0}^{N-1} \boldsymbol{x}[n] \exp(-j2\pi \frac{k}{N} n), \tag{7.1}$$

where $0 \leq k < N$ represents the frequency indices. Let $\boldsymbol{h} \in \mathbb{R}^M$ be a filter of length M. The time-domain convolution can be expressed as:

$$(\boldsymbol{x} * \boldsymbol{h})[n] = \sum_{m=0}^{M-1} \boldsymbol{x}[\,(n-m) \bmod N\,]\, \boldsymbol{h}[m], \tag{7.2}$$

which can be implemented as a 1D convolutional layer in neural networks. To simplify the equations, we assume periodic padding, although zero padding or reflection padding can also be applied. According to the convolution theorem, this time-domain convolution modifies each frequency component $\mathcal{F}\{\boldsymbol{x}\}[k]$ depending on $\mathcal{F}\{\boldsymbol{h}\}$ in the frequency domain. Thus, it operates as a linear system with a frequency response characterized by the magnitude $|\mathcal{F}\{\tilde{\boldsymbol{h}}\}|$ and phase $\angle\mathcal{F}\{\tilde{\boldsymbol{h}}\}$.

Time-domain nonlinear operation

Let $f : \mathbb{R} \to \mathbb{R}$ be an arbitrary pointwise nonlinear function. From the pointwise property, the resulting signal from applying f to each element of the input signal $\boldsymbol{x}$ can be expressed as $f(\boldsymbol{x})[n] = f(\boldsymbol{x}[n])$. To facilitate the analysis of the anti-aliased nonlinear operation discussed in Section 7.2.2, we provide an equivalent interpretation and implementation of the pointwise nonlinear operation $f(\boldsymbol{x})$ as a vector multiplica-

tion. Specifically, we introduce a coefficient signal $\boldsymbol{a}$, so that the function $f(\boldsymbol{x})$ can be expressed as the Hadamard product of $\boldsymbol{x}$ and $\boldsymbol{a}$, i.e., $f(\boldsymbol{x}) = \boldsymbol{x} \odot \boldsymbol{a}$, where $\boldsymbol{a}$ is defined as:

$$\boldsymbol{a}[n] = \begin{cases} f(\boldsymbol{x}[n]) \ / \ \boldsymbol{x}[n] & \text{if } \boldsymbol{x}[n] \neq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{7.3}$$

Note that this formulation implicitly assumes that $f(0) = 0$, i.e., the activation function belongs to a family for which the output is zero when the input is zero. This assumption holds for most standard activation functions, and thus does not significantly limit generality.

According to the inverse convolution theorem, multiplying two discrete-time signals in the time domain corresponds to the circular convolution of their spectra in the frequency domain. Therefore, the spectrum of $\boldsymbol{x} \odot \boldsymbol{a}$ is given by:

$$\begin{aligned} \mathcal{F}\{\boldsymbol{x} \odot \boldsymbol{a}\}[k] &= (\mathcal{F}\{\boldsymbol{x}\} * \mathcal{F}\{\boldsymbol{a}\})[k] \\ &= \sum_{m=0}^{N-1} \mathcal{F}\{\boldsymbol{x}\}[\,(k+m) \bmod N\,]\ \mathcal{F}\{\boldsymbol{a}\}[m]. \end{aligned} \tag{7.4}$$

This equation indicates that high-frequency components with indices exceeding N fold back into the lower-frequency band due to the periodic nature of the discrete Fourier transform, resulting in overlapping spectra. If the sampling frequency of $\boldsymbol{x}$ is insufficient to represent the bandwidth required for $f(\boldsymbol{x})$, aliasing occurs.

7.2.2 Anti-aliased nonlinear operation

BigVGAN [11] incorporates the anti-aliased nonlinear operation [10] to mitigate aliasing effects. After explaining the original formulation of this operation, we present an

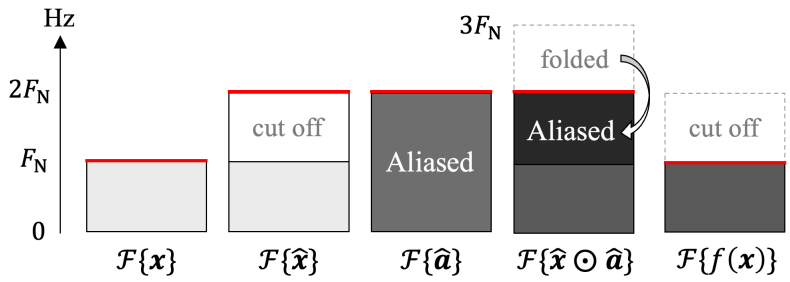

Figure 7.1: *Frequency domain representations of each signal in the anti-aliased nonlinear operation [11] are depicted. The output signal $f(\boldsymbol{x})$, shown on the right, is obtained through the process described in Eq. (7.6). The red lines indicate the Nyquist frequencies.*

equivalent interpretation and implementation based on vector multiplication, as described in Section 7.2.1. Finally, we discuss the limitations and drawbacks of the anti-aliased nonlinear operation.

Original formulation

The anti-aliased nonlinear operation first upsamples the input signal $\boldsymbol{x}$ by a factor of two, applies the nonlinear operation f, and then downsamples the signal back to its original temporal resolution after low-pass filtering. Defining F_N as the Nyquist frequency associated with $\boldsymbol{x}$, the resulting signal $f(\boldsymbol{x})$ is obtained as follows:

$$\hat{\boldsymbol{x}} = \mathrm{lowpass}(\,\mathrm{resample}(\boldsymbol{x}, 2), F_\mathrm{N}), \tag{7.5}$$

$$f(\boldsymbol{x}) = \mathrm{resample}(\,\mathrm{lowpass}(f(\hat{\boldsymbol{x}}), F_\mathrm{N}), 0.5), \tag{7.6}$$

where lowpass($\boldsymbol{v}, c$) represents low-pass filtering that retains only frequencies below c, and resample($\boldsymbol{v}, c$) denotes resampling the input signal $\boldsymbol{v}$ by a factor of c. Note that f is applied to the upsampled signal $\hat{\boldsymbol{x}} \in \mathbb{R}^{2N}$, which has a Nyquist frequency of $2F_{\mathrm{N}}$.

Equivalent interpretation and implementation

Building on the discussion in Section 7.2.1, we can derive an alternative implementation for computing $f(\hat{\boldsymbol{x}})$ in Eq. (7.6) using a coefficient signal $\hat{\boldsymbol{a}} \in \mathbb{R}^{2N}$ as follows:

$$\hat{\boldsymbol{a}}[n] = \begin{cases} f(\hat{\boldsymbol{x}}[n]) \ / \ \hat{\boldsymbol{x}}[n] & \text{if } \hat{\boldsymbol{x}}[n] \neq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{7.7}$$

$$f(\hat{\boldsymbol{x}}) = \hat{\boldsymbol{x}} \odot \hat{\boldsymbol{a}}. \tag{7.8}$$

This reformulation clarifies the advantages of the anti-aliased nonlinear operation. Similar to Eq. (7.4), the spectrum of $\hat{\boldsymbol{x}} \odot \hat{\boldsymbol{a}}$ is given by the circular convolution of the spectra of $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{a}}$:

$$\begin{aligned} \mathcal{F}\{\hat{\boldsymbol{x}} \odot \hat{\boldsymbol{a}}\} &= (\mathcal{F}\{\hat{\boldsymbol{x}}\} * \mathcal{F}\{\hat{\boldsymbol{a}}\})[k] \\ &= \sum_{m=0}^{2N-1} \mathcal{F}\{\hat{\boldsymbol{x}}\}[\,(k+m) \bmod 2N]\ \mathcal{F}\{\hat{\boldsymbol{a}}\}[m]. \end{aligned} \tag{7.9}$$

Since $\hat{\boldsymbol{x}}$ is obtained by $2\times$ oversampling of $\boldsymbol{x}$, its Nyquist frequency extends to $2F_{\mathrm{N}}$, although the actual occupied spectrum remains within the original band $[-F_{\mathrm{N}},\ F_{\mathrm{N}}]$. This oversampling causes spectral overlap in the frequency range from F_{N} to $2F_{\mathrm{N}}$, as shown in Fig. 7.1. As indicated in Eq. (7.6), a low-pass filter removes this overlapping

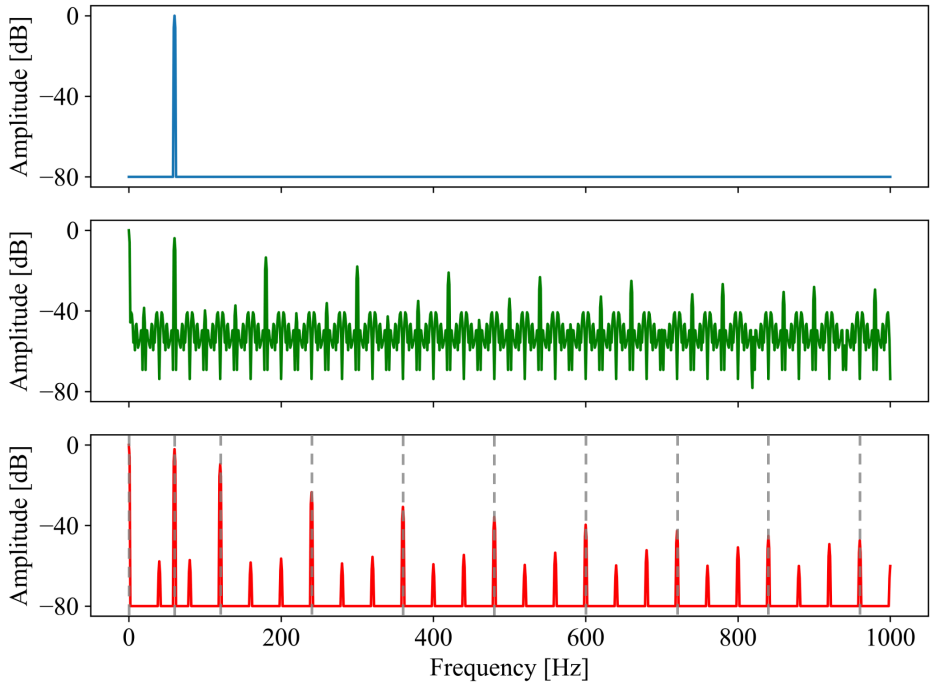

Figure 7.2: *Amplitude spectra in dB of* $\hat{\boldsymbol{x}}$ *(blue),* $\hat{\boldsymbol{a}}$ *(green), and* $\hat{\boldsymbol{x}} \odot \hat{\boldsymbol{a}}$ *(red) are shown for a 60 Hz sinusoidal signal* $\boldsymbol{x}$ *when the ReLU [82] function is applied. The sampling frequency of the original signal* $\boldsymbol{x}$ *is 1 kHz. The dotted lines mark expected harmonic frequencies from Eq. (4.8). Despite the use of an anti-aliasing nonlinear operation [11], aliasing artifacts are still clearly observed in the green and red spectra.*

frequency range, effectively mitigating aliasing in the resulting signal $f(\boldsymbol{x})$.

Limitations and drawbacks

While the anti-aliased nonlinear operation effectively reduces aliasing, it has notable limitations. First, it does not prevent aliasing introduced into the coefficient signal $\hat{\boldsymbol{a}}$ in Eq. (7.7). For instance, consider applying the ReLU activation function using the anti-aliased nonlinear operation. The rectified version of the input signal $\hat{\boldsymbol{x}}$ can be

expressed as the product of $\hat{\boldsymbol{x}}$ and a coefficient signal $\hat{\boldsymbol{a}}$, defined as follows:

$$\hat{\boldsymbol{a}}[n] = \begin{cases} 1 & \text{if } \hat{\boldsymbol{x}}[n] > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{7.10}$$

As illustrated by Eq. 4.8, the ReLU activation function corresponds to multiplying rectangular windows with infinite support in the frequency domain. However, pointwise computation of $\hat{\boldsymbol{a}}$ corresponds to sampling from a continuous-time signal, potentially introducing aliasing into $\hat{\boldsymbol{a}}$. Figure 7.2 shows that aliasing occurs in both $\hat{\boldsymbol{a}}$ and $\hat{\boldsymbol{x}} \odot \hat{\boldsymbol{a}}$ when ReLU is applied to a sinusoidal signal. Additionally, as indicated by Eq. (4.8), higher F_0 values result in stronger harmonic amplitudes, which makes aliasing more pronounced at higher F_0 and can degrade the quality of synthesized high-pitched audio.

Second, there is a fundamental trade-off between the effectiveness of anti-aliasing and computational efficiency. This arises from the need to insert resampling operations before and after each nonlinear operation, where the computational cost and suppression performance depend heavily on design parameters such as filter length and oversampling ratio. A longer filter enables sharper attenuation in the high-frequency range, improving aliasing suppression. A higher oversampling ratio increases temporal resolution and shifts aliasing artifacts away from the frequency band of interest, making them easier to remove. While oversampling ratios of $4\times$ or higher are common in classical signal processing, neural vocoders are typically limited to lower ratios such as 2, due not only to hardware constraints like GPU memory but also to deployment scenarios where high-end hardware is not available.

7.2.3 Frequency-domain processing

We now turn to frequency-domain convolution and nonlinear operations, highlighting how they can effectively avoid aliasing. In this context, frequency-domain processing refers to operations performed on the spectrum $\mathcal{F}\{\boldsymbol{x}\} \in \mathbb{C}^N$, obtained by applying DFT to the time-domain signal $\boldsymbol{x} \in \mathbb{R}^N$.

Frequency-domain convolution

Let $\boldsymbol{l} \in \mathbb{C}^L$ be a convolution kernel of length $L \leq N$. If periodic padding is used, the convolution in the frequency domain is given by:

$$(\mathcal{F}\{\boldsymbol{x}\} * \boldsymbol{l})[k] = \sum_{m=0}^{L-1} \mathcal{F}\{\boldsymbol{x}\}[\,(k+m) \bmod N]\; \boldsymbol{l}[m]. \tag{7.11}$$

Unlike time-domain nonlinear operations, which result in circular convolution in the frequency domain, frequency-domain convolution is a localized process limited by the receptive field size L. If L is sufficiently smaller than the spectrum length N, this operation avoids problematic spectral overlaps, thereby effectively avoiding the aliasing.

Frequency-domain nonlinear operation

Consider an arbitrary pointwise nonlinear operation in the frequency domain, denoted as $g : \mathbb{C} \to \mathbb{C}$. As discussed in Section 7.2.1, the resulting spectrum $g(\mathcal{F}\{\boldsymbol{x}\})[k] = g(\mathcal{F}\{\boldsymbol{x}\}[k])$ can be expressed as the Hadamard product of $\mathcal{F}\{\boldsymbol{x}\}$ and a coefficient vector

$\boldsymbol{b} \in \mathbb{C}^N$. That is, $g(\mathcal{F}\{\boldsymbol{x}\}) = \mathcal{F}\{\boldsymbol{x}\} \odot \boldsymbol{b}$, where $\boldsymbol{b}$ is defined as:

$$\boldsymbol{b}[k] = \begin{cases} g(\mathcal{F}\{\boldsymbol{x}\}[k]) \ / \ \mathcal{F}\{\boldsymbol{x}\}[k] & \text{if } \mathcal{F}\{\boldsymbol{x}\}[k] \neq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{7.12}$$

This operation independently modifies each element of $\mathcal{F}\{\boldsymbol{x}\}$, similar to time-domain convolution (Section 7.2.1). Although this corresponds to circular convolution in the time domain, potentially causing time-domain overlaps, this study focuses on spectral overlaps in $\mathcal{F}\{\boldsymbol{x}\}$, recognizing them as a more critical factor contributing to practical drawbacks. Consequently, frequency-domain nonlinear operations do not introduce problematic spectral overlap in $\boldsymbol{x}$.

7.2.4 Time-frequency-domain processing

Typical audio signals, such as speech or music, are non-stationary and are best represented using time-frequency representations like spectrograms. To extend our discussion of 1D frequency-domain processing in Section 7.2.3 to the 2D time-frequency domain, we consider the spectrogram $\boldsymbol{X} \in \mathbb{C}^{M \times K}$, which is obtained by analyzing the input time signal $\boldsymbol{x} \in \mathbb{R}^N$ via STFT. Here, M is the number of time frames and K is the number of frequency bins, while $\boldsymbol{X}[m, k]$ represents the k-th frequency component in the m-th time frame. Unlike 1D convolution in the frequency domain, 2D convolution in the time-frequency domain influences each spectrum along the time axis. Despite this, the process remains localized and does not cause problematic spectral overlaps. Meanwhile, pointwise nonlinear operations in the time-frequency domain are applied independently to each spectrum $\boldsymbol{X}[m]$, coming down to the discussion in Section 7.2.3. Consequently, time-frequency-domain processing offers advantages over time-domain

processing, as it can effectively avoid aliasing.

From the perspective of image processing, complex spectrograms can be interpreted as 2D images, where nonlinear operations and resampling on them introduce aliasing artifacts, analogous to visual data processing. However, image-domain aliasing typically results in localized distortions [10], whereas aliasing in time-domain processing leads to more severe artifacts such as spectral folding. Therefore, we regard image-domain aliasing as a secondary concern in this study. Still, techniques such as intermediate oversampling [10, 11] and shift-invariance facilitation [39] are potentially beneficial for spectrogram-based generation, as they are domain-agnostic and applicable to both temporal and spatial signals.

7.3 Proposed Method

This chapter introduces Wavehax, an aliasing-free neural vocoder based on the standard GAN [9] vocoder framework, designed to avoid aliasing by leveraging time-frequency domain processing, as discussed in Section 7.2. Additionally, we highlight the importance of integrating 2D CNNs and a harmonic prior for robust, high-fidelity complex spectrogram estimation. Figure 7.3 illustrates an overview of Wavehax.

7.3.1 Harmonic prior

Analogous to prior distributions in Bayesian statistics, we use the term *prior* to refer to signals considered by the model before observing any training examples. Intuitively, the prior signal acts as an explicit bias, guiding the model in generating the output

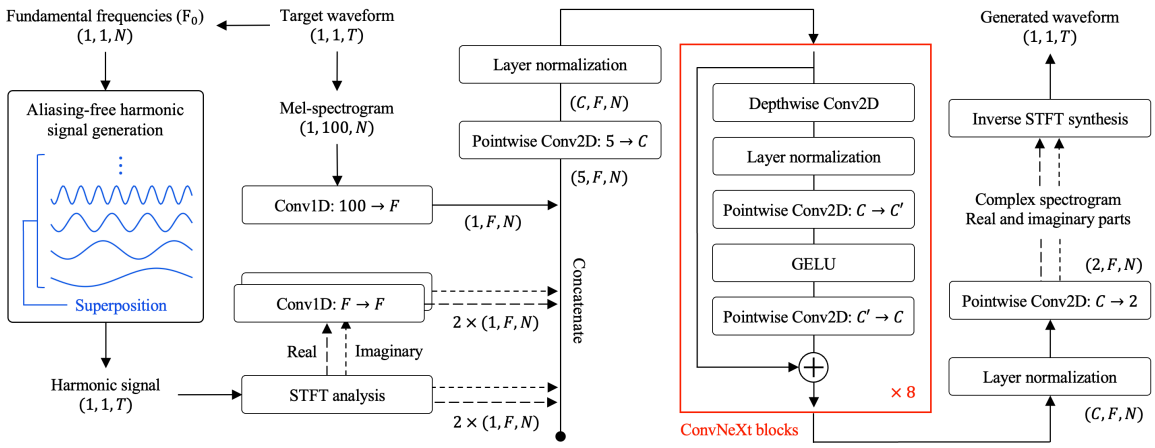

Figure 7.3: *This diagram provides an overview of Wavehax. The kernel width of the 1D convolution is set to 7, while the kernel size of the depthwise convolution is set to 7×7. The number of hidden channels, denoted as C and C', are set to 32 and 64, respectively. The number of frequency bins, F, is set to 241, calculated as half of the discrete Fourier transform points plus one. T and N represent the number of time steps in the waveforms and time frames in the features, respectively.*

waveform[1]. Periodic priors have proven effective for explicit prosodic control in numerous studies, particularly in GAN vocoders [30,32,34,35,77,111,118], as well as in other frameworks [24, 33, 119]. We further analyze and provide a theoretical explanation of how periodic priors enhance neural vocoders in both the time and time-frequency domains.

Periodic prior for time domain models

Our hypothesis is that time-domain neural vocoders can effectively generate harmonic components from periodic priors by modulating them through nonlinear operations. To substantiate this hypothesis, we first examine Maclaurin expansion of a

[1]Note that GAN-based vocoders, such as PWG [8], can be seen as using a non-informative prior, where Gaussian noise serves as input. This absence of prior knowledge forces the models to infer the spectral and temporal structures of output waveforms entirely from the training data.

general pointwise nonlinear function, $f : \mathbb{R} \to \mathbb{R}$, assuming this function satisfies the condition for Maclaurin expansion[2]. Applying the Maclaurin series of f to a time-domain signal $\boldsymbol{x}$ defined at each time step n, we obtain the following expression for the transformed signal $f(\boldsymbol{x})[n] = f(\boldsymbol{x}[n])$:

$$f(\boldsymbol{x})[n] = \sum_{k=0}^{\infty} \frac{f^{(k)}(0)}{k!} \boldsymbol{x}^k[n]. \tag{7.13}$$

This expansion shows that $f(\boldsymbol{x})$ is a weighted sum of powers of $\boldsymbol{x}^k$, and the magnitude of the derivatives $f^{(k)}$ influences the degree of aliasing. According to the inverse convolution theorem, the powers of a signal in the time domain correspond to repeated convolutions in the frequency domain, implying that higher-order terms introduce additional frequency components. Figure 7.4 illustrates the frequency characteristics of each power up to the 6th order when $\boldsymbol{x}$ is a sinusoidal signal.

In particular, when we consider a simple periodic input signal, such as a sinusoidal signal defined as $\boldsymbol{x}[n] = \sin(\omega n)$, where ω is the angular frequency, the harmonics generated by the nonlinear operation can be analytically derived as shown in the following general formula:

$$\sin^k(\omega n) = \sum_{m=0}^{k} \big\{ a_m \sin((m\omega)n) + b_m \cos((m\omega)n) \big\}, \tag{7.14}$$

where a_m and b_m are coefficients dependent on k. This indicates that if the input $\boldsymbol{x}$ is a sinusoidal signal, the k-th power term in Eq. (7.13) becomes a sum of harmonics up to the k-th order, implying that nonlinear operations effectively generate harmonics that align with the input periodic signal. The proof of Eq. (7.14) is provided in Appendix 9.2.

[2]While this analysis strictly holds only when f is infinitely differentiable at 0, which is not true for functions like ReLU, approximate analysis is still possible using piecewise definitions or smooth approximations (e.g., SoftPlus), allowing similar interpretations in a weak or local sense.

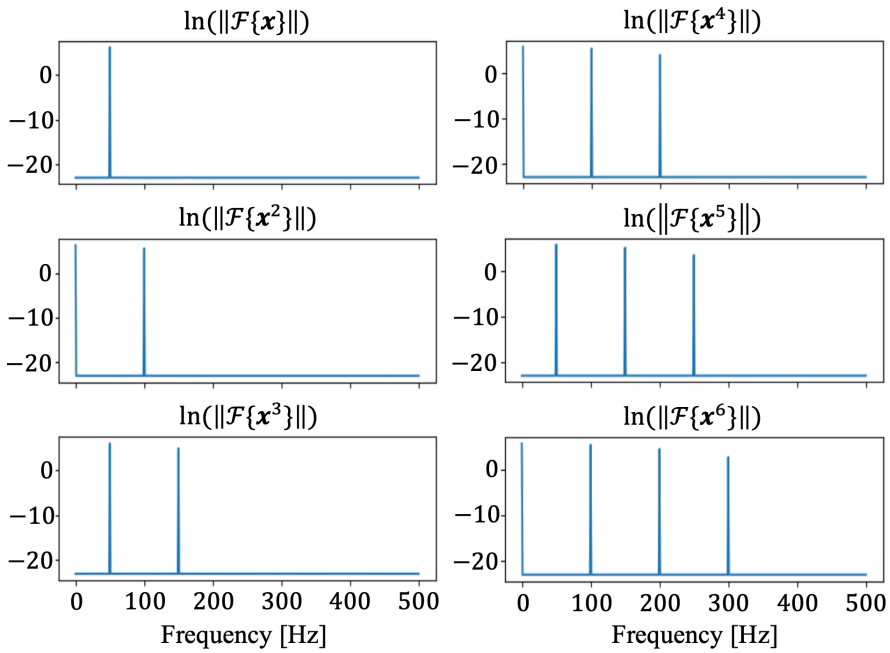

Figure 7.4: *Log amplitude spectra of* $\boldsymbol{x}^k$ *for* k *up to 6, where* $\boldsymbol{x}$ *is a 50 Hz sinusoidal signal, are shown. The sampling frequency is 1 kHz, with the discrete Fourier transform over a duration of 1 second.*

This demonstrates why periodic priors (even for simple sinusoidal signals) offer a strong inductive bias for robust and efficient speech waveform generation when combined with time-domain nonlinear operations.

Periodic prior for time-frequency domain models

Wavehax operates in the time-frequency domain, which presents an additional challenge: the effective harmonic generation mechanism through time-domain nonlinear operations is inherently unavailable. Given the structural differences compared to time-domain vocoders, we argue that directly feeding harmonic information into the neural network is crucial for high-fidelity waveform synthesis. To this end, we utilize a complex spectrogram derived from a harmonic prior, denoted as $\boldsymbol{e}$, via STFT. We

construct $\boldsymbol{e}$ based on F_0 values $f[n]$ on each time step n, ensuring that the process is aliasing-free while maintaining a pseudo-constant power across all time frames. This process is expressed by the following formula:

$$\boldsymbol{e}[n] = \sum_{k=1}^{K_n} \sqrt{\frac{0.02}{K_n}} \sin(2\pi k\phi[n] + k\varphi) + 0.01\boldsymbol{z}[n]. \tag{7.15}$$

Here, F_{N} denotes the Nyquist frequency and $\boldsymbol{n}[n]$ is a random variable sampled from a normal distribution $\mathcal{N}(\mathbf{0}, I)$. The number of harmonics, K_n, is defined as $K_n = \lfloor F_{\mathrm{N}}/f[n] \rfloor$, while the cumulative phase, $\phi[n]$, is given by $\phi[n] = \sum_{m=0}^{n} f[m]/F_{\mathrm{N}}$. The details of this process are provided in Appendix 9.3. This method enables smooth phase modeling of harmonic components without relying on the structural advantages typical of time-domain vocoders. In Section 7.4, we demonstrate that using harmonic priors yields significantly better results than sinusoidal priors.

7.3.2 Model architecture

Complex spectrograms have strong correlations between harmonic components, which are broadly distributed but appear intermittently. Additionally, the frequency elements between harmonics exhibit significant randomness, further complicating the estimation process. To address this complexity, we employ 2D CNNs, which have proven effective for complex spectrogram estimation as demonstrated in [93]. In Wavehax, the input harmonic signal is first transformed into a complex spectrogram using STFT with a Hanning window. A 1D convolutional layer is applied to this complex spectrogram to capture the receptive field comprehensively along the frequency axis. Concurrently, the input acoustic feature (e.g., mel-spectrogram) is converted into a single-channel 2D map via another 1D convolutional layer, matching the dimensions of the complex spec-

trogram. These features are concatenated along the channel axis to form a five-channel 2D map, which is then processed through linear and layer normalization layers [120] to align with the channels in the subsequent ConvNeXt [97] blocks. Each ConvNeXt block comprises a depthwise CNN, layer normalization, and a GELU [121] activation function, sandwiched between two pointwise CNNs, which modulate the input feature map. After passing through these blocks, the latent features are further transformed by additional layer normalization and linear layers, yielding a two-channel 2D map representing the real and imaginary components of the complex spectrogram. Finally, the output audio waveform is reconstructed using iSTFT, followed by overlap-add with a Hanning window function.

7.3.3 Adversarial training

Wavehax employs the standard framework of GAN-based vocoders, incorporating mel-spectrogram loss, adversarial loss, and feature-matching loss. The mel-spectrogram loss is defined in Eq. (3.28), which evaluates the L1 distance between the mel-spectrograms of the natural and the generated waveforms. For adversarial learning, we adopt the hinge GAN objective as described in Eq. (3.21) and (3.22). The feature matching loss, defined in Eq. (3.23), computes the L1 distance between intermediate activations of real and generated samples across all layers of all subdiscriminators. These configurations follow the same setup used in the baseline model Vocos [96], enabling a clearer and more controlled comparison.

The final generator loss is defined as the sum of $\mathcal{L}_{\text{mel}}$, $\mathcal{L}_{\text{adv}}$, and $\mathcal{L}_{\text{fm}}$:

$$\mathcal{L}_{\mathcal{G}} = \lambda_{\text{mel}}\mathcal{L}_{\text{mel}} + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}} + \lambda_{\text{fm}}\mathcal{L}_{\text{fm}}, \tag{7.16}$$

where λ_{mel}, λ_{adv}, and λ_{fm} are balancing weights set to 45.0, 1.0, and 2.0, respectively, based on [67].

Using sophisticated discriminators is essential for high-fidelity speech synthesis. [11, 68] suggested that the multi-resolution discriminator (MRD), introduced by UnivNet [68], is superior to MSD, preventing over-smoothed spectrograms of synthesized speech. Following these studies, we adopt the combination of MPD and MRD to maximize performance.

7.4 Experimental Evaluation on Speech Analysis Synthesis

7.4.1 Overview

We evaluate several neural vocoders through speech analysis and synthesis. As discussed in Section 7.2, aliasing impacts are theoretically more pronounced when generating speech with unknown F_0, particularly at higher F_0 values. To examine this hypothesis, we investigate performance using training data constrained to a limited F_0 range, allowing for explicit evaluation on unknown F_0 values. We begin by presenting an ablation study on our proposed method, followed by a comparison with baseline models based on existing approaches.

7.4.2 Data preparation

We used the Japanese Versatile Speech (JVS) corpus [122], which consists of approximately 15K utterances from 100 male and female speakers, with a total duration of about 30.2 hours. The corpus includes normal, whispered, and falsetto speech record-

ings, all sampled at 24 kHz. We used the raw audio data without volume normalization or preprocessing. First, we carefully examined the F_0 range for each speaking style and speaker. The overall F_0 range in the corpus was 57 to 873 Hz. We calculated the lower 10% and upper 20% boundaries on a log F_0 scale, corresponding to frequencies below 71 Hz and above 499 Hz. We excluded utterances from the training dataset that contained any F_0 outside the training range 71 $\sim$ 499 Hz. Utterances with F_0 values outside the 71-499 Hz range were excluded from the training dataset, resulting in the removal of 26% of the data for values below the lower limit and 5% for values above the upper limit. The evaluation dataset consisted of the following three subsets:

I. 200 utterances with F_0 entirely within the training range.

II. 200 utterances containing F_0 values below 71 Hz.

III. 200 utterances containing F_0 values above 499 Hz.

200 utterances, not used in the training or evaluation sets, were randomly selected for the validation dataset. For conditioning the vocoders, we extracted 100-band log mel-spectrograms using a 2048-point Fast Fourier Transform (FFT), a Hanning window, and a 10 ms frame shift. The mel-filter bank covered a frequency range from 0 to 8 kHz. Additionally, we extracted F_0 using the Harvest algorithm [45], with search ranges tailored for each speaker and speaking style.

7.4.3 Model details

We used three baseline methods, each of which was optionally extended with specific features as follows:

- PWG [8]: This model operates at a fixed time resolution and does not involve up-sampling, which allows us to evaluate aliasing effects that are caused exclusively by

the nonlinear functions. Additionally, we explored the effectiveness of the periodic priors and anti-aliased nonlinear operation [11] in scenarios involving F_0 extrapolation. We used the official implementation of the anti-aliased nonlinear operation[3] for all activation layers.

- HiFi-GAN [67]: This model utilizes progressive upsampling, which introduces aliasing from both upsampling and nonlinear operations. Similar to PWG, we investigated the effectiveness of the periodic priors and anti-aliased nonlinear operation. Periodic waveforms were incorporated using downsampling CNNs, as described in [38].

- Vocos [96]: This model estimates complex spectrograms without upsampling and generates waveforms using iSTFT. The FFT size and window length were set to 960, with a Hanning window function. Unlike Wavehax, Vocos does not incorporate any priors; instead, it uses 1D CNNs to estimate the log amplitude and implicit phase-wrapping [96]. To investigate whether periodic priors are also effective for Vocos, we optionally modified Vocos to convert the priors to log-amplitude and absolute-phase spectrograms, with phase values constrained to the range $(-\pi, \pi]$. The backbone network then receives the sum of these latent features and the projected mel-spectrogram.

- Wavehax: The FFT size and window length were set to 480, with a 50% overlap between frames, compared to Vocos's 75% overlap. Since Wavehax employs 2D CNNs, computational complexity increases with FFT size. Therefore, we selected a smaller FFT size to minimize complexity while maintaining sufficient frequency resolution.

All models were trained for 1M steps with a batch size of 16 and a sequence length

[3]BigVGAN official code: `https://github.com/NVIDIA/BigVGAN`

of 7,680 samples (320 ms). Gradient clipping with a threshold of 10.0 was applied to all models to stabilize training and accelerate convergence. PWG models were trained using the Adam optimizer [107] with beta parameters set to $[0.5, 0.9]$. The initial learning rate was 2.0×10^{-4}, halved every 200K steps. PWG models were trained with the same loss functions as HiFi-GAN to enhance performance [123]. HiFi-GAN was trained with the AdamW optimizer [124], using the original beta parameters of $[0.8, 0.99]$. We found that the original learning rate decay of 0.999 was too small with gradient clipping. Therefore, we adopted the learning rate settings from BigVGAN [11], using an initial rate of 1.0×10^{-4} and a decay factor of 0.9999996. Vocos and Wavehax were trained using the AdamW optimizer with a cosine learning rate scheduler, following the official Vocos code[4]. The initial learning rate was 2.0×10^{-4}, with beta parameters set to $[0.8, 0.9]$. We followed Vocos' hyperparameter settings for the discriminators except for the tuned loss weights for each subdiscriminator.

7.4.4 Evaluation metrics

We used six evaluation metrics, detailed below:

- VUV↓ measures the percentage of incorrect classifications of speech segments as voiced or unvoiced (i.e., produced with or without vocal fold vibration), based on F_0 sequences extracted from recorded and synthesized speech.
- RMSE↓ measures the root mean squared error of log F_0 values between recorded and synthesized speech, assessing F_0 reproduction accuracy.
- STFT↓ measures the L1 distance between multi-resolution log amplitude spectrograms of recorded and synthesized speech [8], capturing fine and coarse spectral

[4]Vocos official code: `https://github.com/charactr-platform/vocos`

Table 7.1: *This table shows the speech reconstruction performances from mel-spectrograms for the ablation study. The 'Prior' column represents the type of input waveform generated using F_0. When no prior is specified, the models use only mel-spectrograms, following their original architectures. The 'Subset' column denotes the evaluation dataset detailed in Section 7.4.2. Amp. and IPW denote log amplitude and implicit phase-wrapping [96] spectrograms. The best scores are highlighted in bold. H-RI-RI is the proposed method.*

	Prior	Input	Output	Subset	VUV↓	RMSE↓	STFT↓	PESQ↑	UTMOS↑
N-RI-RI	Noise	Real Imag.	Real Imag.	I	10	0.111	0.711	3.108	2.038
				II	34	0.212	0.821	2.288	1.465
				III	**4**	0.078	0.763	3.306	1.424
S-RI-RI	Sine	Real Imag.	Real Imag.	I	**6**	0.085	0.701	3.733	2.972
				II	**13**	0.127	0.708	3.589	3.326
				III	**4**	0.072	0.839	3.209	1.524
H-RI-RI	Har.	Real Imag.	Real Imag.	I	**6**	0.081	**0.678**	**3.818**	**3.122**
				II	**13**	**0.114**	**0.683**	**3.702**	**3.585**
				III	**4**	0.071	0.722	3.927	**1.686**
H-LA-LP	Har.	Amp. Phase	Amp. IPW	I	**6**	**0.080**	0.682	3.793	3.071
				II	**13**	0.117	0.687	3.684	3.531
				III	**4**	**0.070**	**0.714**	**3.934**	**1.686**
H-L-LP	Har.	Amp.	Amp. IPW	I	8	0.098	0.683	3.47	2.309
				II	22	0.167	0.712	2.84	1.994
				III	4	0.075	0.720	3.67	1.524

details. It is particularly useful for evaluating aliasing artifacts, which often appear as spectral distortions or blurring. We used FFT sizes of 512, 1024, and 2048, with corresponding window lengths and one-fourth frame shifts.

- PESQ↑ assesses speech quality by comparing the spectral distance between recorded and synthesized speech, correlating with human auditory perception. We used the open-source Python library[5] for calculations.

- UTMOS↑ indicates estimated subjective speech quality by a DNN model trained to predict mean opinion scores (MOS) [71]. Calculations were performed using the official library[6].

- MOS↑ measures subjective speech quality through human listener evaluations. Participants rated the overall naturalness of the speech on a scale from 1 (poor) to 5 (excellent), considering sound quality, clarity, and intelligibility.

All scores were calculated after normalizing each audio sample to -24 dB using the open-source Python library[7]. PESQ and UTMOS were computed at a 16 kHz sampling rate, excluding the frequency band above 8 kHz, where aliasing is more prominent. Subjective evaluations were conducted independently for each of the three evaluation subsets (I-III). For each subset, we recruited 30 native Japanese participants with no requirement for expertise in speech or audio technology. Each participant rated all vocoder models in the assigned subset, resulting in a total of 300 opinion scores collected per model-subset pair.

[5]PyPESQ: `https://github.com/vBaiCai/python-pesq`
[6]UTMOS: `https://github.com/sarulab-speech/UTMOS22`
[7]pyloudnorm: `https://github.com/csteinmetz1/pyloudnorm`

7.4.5 Ablation study

Before comparing with baseline models, we present an ablation study on the proposed method. This study examines the impact of variations in prior spectrogram types and the representation of complex spectrograms, assessed using objective metrics. All models were trained as described in Section 7.4.3, and are outlined as follows:

- N-RI-RI: This model uses a non-informative prior (i.e., Gaussian noise). Both the input and output spectrograms are represented by their real and imaginary components.
- S-RI-RI: This model employs a sinusoidal prior, containing only F_0 components. The input and output spectrograms are also represented by their real and imaginary components.
- H-RI-RI: This model uses the harmonic prior, as detailed in Appendix 9.3. The input and output spectrograms are represented by their real and imaginary components.
- H-LA-LP: This model also uses the harmonic prior. The input spectrograms are represented in the log amplitude and absolute phase format. The output spectrograms are provided as pairs of log amplitude and implicit phase-wrapping, as in Vocos.
- H-L-LP: Unlike H-LA-LP, this model uses only the log amplitude spectrogram of the harmonic prior.

Table 7.1 shows the evaluation results. First, N-RI-RI and S-RI-RI exhibit significantly poorer scores compared to H-RI-RI. This supports the hypothesis in Section 7.3.1 that time-frequency-domain processing lacks the inductive bias needed for harmonic generation, unlike time-domain processing, and that the harmonic prior significantly improves performance. Additionally, H-L-LP, which lacks the phase information of

Table 7.2: *This table summarizes the number of learnable parameters, the number of giga multiply-accumulate operations (GMACs) per second for waveform generation, and the real-time factors (RTFs), measured on a single GPU (GeForce RTX 3090) and a single-threaded CPU (AMD EPYC 7542). Note that operations from modules not supported by torchprofile, such as the STFT, are excluded; therefore, the reported MACs represent a lower bound. Models marked with an asterisk (*) use the antialiased nonlinear operation [11] in all activation functions. 'Har.' indicates the use of the harmonic prior described in Section 9.3. RTFs are averaged over 200 utterances.*

	Prior	Params↓	GMACs↓	GPU↓	CPU↓
PWG	Noise	1.423 M	33.41	0.0076	2.598
PWG*	Noise	1.423 M	36.53	0.0089	5.884
HiFi-GAN	–	13.82 M	28.01	0.0051	0.825
HiFi-GAN*	–	13.82 M	29.69	0.0074	1.967
Vocos	–	13.49 M	1.348	0.0026	0.036
Wavehax	Noise	0.623 M	1.787	0.0035	0.150
Wavehax	Har.	0.623 M	1.787	0.0039	0.196

input harmonic signals, performs worse than H-RI-RI and H-LA-LP, both of which incorporate phase information in their priors. This suggests that time-frequency processing struggles to model smooth and coherent harmonic phases due to a lack of structural support. H-RI-RI outperformed H-LA-LP, and we adopted H-RI-RI as our final proposed model. This approach contrasts with previous vocoders based on complex spectrogram estimation [93, 94, 96, 99, 100], which aim to estimate amplitude and phase pairs. We speculate that the inherent uncertainty of 2π phase rotation complicates phase spectrogram estimation in our method.

7.4.6 Comparison with baselines

Model efficiency

Table 7.2 shows the number of trainable parameters, multiply-accumulate operations (MACs) computed using the profile$_{\mathrm{m}}$acs function in the torchprofile library[8], and waveform generation speeds, expressed as real-time factors (RTF) for both GPU and CPU. Note that the anti-aliasing mechanism involves resampling and low-pass filtering with fixed parameters, without increasing the learnable parameters. Wavehax (i.e., H-RI-RI) retains less than 4.5% of the parameters of HiFi-GAN and around 4.6% of Vocos, achieving the lowest MAC count. While the CPU generation speed of Wavehax with the harmonic prior is about 18% of Vocos', it outperforms HiFi-GAN by over four times. Comparing the Wavehax models with noise or harmonic priors, the computation time for generating the harmonic prior accounts for about 23% of the total generation time. This is because the superposition of sinusoidal waveforms takes a large computational cost. PWG* and HiFi-GAN*, which are equipped with anti-aliased nonlinear operations, exhibit significant reductions in CPU generation speed, highlighting their impracticality in environments without GPU resources.

Reconstruction performances

Table 7.3 presents the reconstruction results. The variation in UTMOS [71] across subsets primarily reflects differences in speaking styles, such as whispered speech (Subset I) and falsetto speech (Subset III), which typically yield lower UTMOS scores than normal speech. This inherent bias in the recorded data also influences the synthesized speech. As shown in the table, Wavehax achieves speech quality comparable to that

[8]torchprofile: `https://github.com/zhijian-liu/torchprofile`

Table 7.3: *This table presents the speech reconstruction performances from mel-spectrograms. The 'Prior' column indicates the type of input signal generated using F_0. The 'Subset' column refers to the evaluation dataset described in Section 7.4.2. The best scores are highlighted in bold, and scores with no statistically significant difference from the best are underlined.*

	Prior	Subset	VUV↓	RMSE↓	STFT↓	PESQ↑	UTMOS↑	MOS↑
Recording	–	I	–	–	–	–	3.349	4.007 ± 0.030
		II	–	–	–	–	3.821	4.127 ± 0.028
		III	–	–	–	–	1.796	4.033 ± 0.034
PWG	Noise	I	**6**	0.094	0.725	3.222	2.418	2.860 ± 0.037
		II	16	0.146	0.731	3.077	2.635	2.037 ± 0.032
		III	5	0.082	0.919	2.503	1.416	1.907 ± 0.029
	Sine	I	**6**	**0.078**	0.714	3.631	3.086	3.803 ± 0.031
		II	**13**	0.111	0.714	3.582	3.533	3.810 ± 0.031
		III	**4**	**0.066**	0.777	3.313	1.586	3.383 ± 0.033
	Har.	I	**6**	0.085	0.709	3.656	3.100	3.843 ± 0.030
		II	14	0.119	0.711	3.586	3.552	3.960 ± 0.029
		III	**4**	0.072	0.774	3.364	1.625	3.420 ± 0.035
PWG*	Noise	I	7	0.096	0.713	3.308	2.536	3.020 ± 0.036
		II	16	0.144	0.720	3.172	2.763	2.333 ± 0.033
		III	5	0.082	0.897	2.642	1.419	1.920 ± 0.031
	Sine	I	7	0.085	0.713	3.529	3.010	3.727 ± 0.033
		II	15	0.125	0.713	3.481	3.462	3.603 ± 0.030
		III	**4**	0.071	0.843	2.971	1.524	2.890 ± 0.036
	Har.	I	7	0.083	0.715	3.605	3.092	3.853 ± 0.030
		II	14	0.121	0.714	3.549	3.537	3.657 ± 0.030
		III	**4**	0.071	0.799	3.229	1.612	3.140 ± 0.035
HiFi-GAN	–	I	7	0.089	0.678	3.649	2.875	3.570 ± 0.034
		II	15	0.136	0.686	3.464	3.107	2.803 ± 0.036
		III	**4**	0.081	0.804	3.124	1.482	2.420 ± 0.034
	Sine	I	**6**	0.080	0.666	3.930	3.195	<u>3.930</u> ± 0.031
		II	14	0.113	0.673	3.807	3.590	3.877 ± 0.031
		III	**4**	0.072	0.749	3.648	1.660	3.203 ± 0.037
	Har.	I	**6**	0.079	0.665	3.934	**3.217**	<u>3.950</u> ± 0.031
		II	14	**0.110**	0.669	3.832	3.647	<u>3.970</u> ± 0.029
		III	**4**	0.073	0.727	3.709	1.691	3.343 ± 0.036
HiFi-GAN*	–	I	**6**	0.087	0.674	3.703	2.912	3.543 ± 0.032
		II	14	0.126	0.683	3.538	3.243	3.160 ± 0.034
		III	**4**	0.082	0.792	3.162	1.510	2.507 ± 0.034
	Sine	I	7	0.080	**0.664**	**3.950**	3.207	<u>3.923</u> ± 0.029
		II	15	0.115	0.700	**3.857**	**3.649**	**4.023** ± 0.030
		III	**4**	0.071	0.769	3.517	1.664	3.027 ± 0.038
	Har.	I	7	0.081	0.668	3.937	3.195	**3.973** ± 0.029
		II	14	0.115	**0.673**	3.826	3.640	**4.023** ± 0.029
		III	**4**	0.072	0.727	3.749	**1.710**	3.430 ± 0.036
Vocos	–	I	7	0.090	0.685	3.656	2.909	3.520 ± 0.033
		II	16	0.137	0.703	3.463	3.092	2.733 ± 0.034
		III	**4**	0.077	0.731	3.456	1.538	2.683 ± 0.035
	Sine	I	7	0.092	0.695	3.633	2.904	3.623 ± 0.034
		II	16	0.140	0.709	3.378	3.049	2.817 ± 0.035
		III	**4**	0.078	0.735	3.420	1.545	2.690 ± 0.037
	Har.	I	7	0.091	0.696	3.576	2.936	3.640 ± 0.032
		II	17	0.136	0.706	3.394	3.205	3.197 ± 0.034
		III	**4**	0.079	0.762	3.230	1.522	2.570 ± 0.037
Wavehax	Har.	I	**6**	0.081	0.678	3.818	3.122	3.907 ± 0.032
		II	**13**	0.114	0.683	3.702	3.585	<u>3.973</u> ± 0.031
		III	**4**	0.071	**0.722**	**3.927**	1.686	**3.860** ± 0.035

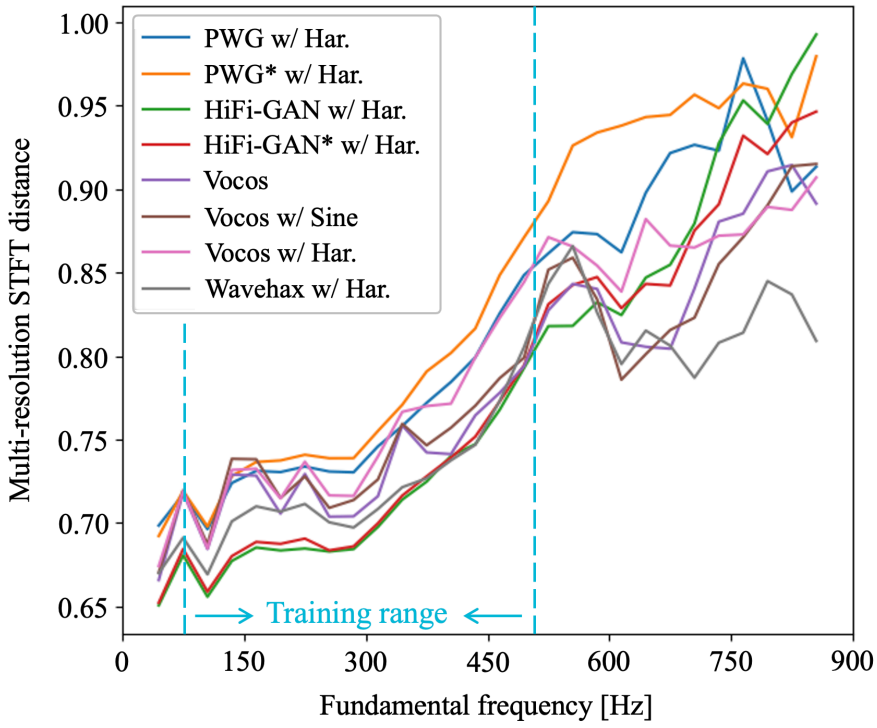

Figure 7.5: *This figure shows the multi-resolution STFT distances averaged for each F_0 bin, grouped by 30 Hz intervals. Models marked with an asterisk (*) are equipped with the anti-aliased nonlinear operation [11]. 'w/ Sine' and 'w/ Har.' indicate models enhanced by the sinusoidal or harmonic priors, respectively. The STFT frame shift is fixed at 10 ms across all resolutions to match the temporal resolution of the F_0 sequences.*

of the top-performing baseline, HiFi-GAN* with harmonic prior, on Subsets I and II, while using significantly fewer parameters and achieving faster synthesis (see Table 7.2). These results indicate that Wavehax effectively avoids the aliasing-related complexities seen in conventional waveform generators, enabling high-fidelity and computationally efficient speech synthesis.

On subset III, all time-domain vocoders show notable degradation, especially in MOS, with performance gaps between subsets I and III much larger than those between I and II. This result supports the hypothesis that aliasing in neural vocoders

compromises robustness in synthesizing high-pitched speech. In contrast, Wavehax achieves significantly higher scores than all baseline models on subset III, showcasing its robustness, due to its aliasing-free design. Figure 7.5 further supports this, illustrating how multi-resolution STFT distances of some models change with respect to F_0. The graph shows that Wavehax, unlike the other models, maintains stable performance without significant degradation at unseen high F_0 values.

Compared to the Vocos models, Wavehax consistently outperforms them across all metrics, particularly in subsets II and III. This discrepancy likely arises from differences in architectural design. Vocos relies on 1D CNNs to achieve high expressiveness and complexity, strongly reflecting the training data through a large number of parameters. In contrast, Wavehax employs 2D CNNs, capturing time-frequency representations in more redundant latent feature spaces with fewer parameters, allowing more effective adaptation to unseen conditions. The translational invariance of 2D CNNs further enhances robustness by preserving spatial coherence in processing harmonic spectrograms.

Focusing on the time-domain baseline models, PWG and HiFi-GAN, we observe that the harmonic prior almost consistently yields the best results. Morrison et al. [78] argued that non-autoregressive GAN-based vocoders struggle to model the relationship between F_0 and harmonic phases, which involves cumulative summation. This challenge can also extend to harmonic components not generated via time-domain nonlinear operations, where certain terms in Eq. (7.13) can disappear due to vanishing derivatives of particular orders. We speculate that models using the sinusoidal prior build harmonic structures less effectively than those with harmonic priors. Additionally, anti-aliased nonlinear operations have proven effective in some cases, such as PWG* and HiFi-GAN* without periodic priors, and HiFi-GAN* with the harmonic

prior. However, the improvements brought by them were inconsistent, implying the limitations discussed in Section 7.2.2.

7.5 Experimental Evaluation via Singing Voice Analysis-Synthesis

7.5.1 Overview

Aliasing artifacts tend to appear more prominently in the high-frequency range due to the nature of signal processing. Because singing voice signals contain strong harmonic components extending into the high frequencies, we conducted analysis-synthesis experiments at a 48 kHz sampling rate using singing voice data to more directly examine the impact of aliasing. In addition, we evaluated the robustness of the models under pitch-shifted conditions, which serve as out-of-distribution inputs not observed during training.

7.5.2 Data preparation

We used the Namine Ritsu Database V2 [125], a Japanese singing voice dataset featuring a single female vocalist. The database comprises 210 songs totaling approximately 9.3 hours of recordings, with an F_0 range (measured from the musical score) spanning from 100 Hz to 1000 Hz. Although the recordings are sampled at 48 kHz, the released audio contains usable bandwidth only up to that corresponding to a 44.1 kHz sampling rate. While this is generally considered suboptimal, it does not substantially affect the objectives of our experiments, which are to investigate high-frequency aliasing and evaluate pitch-shift performance. Each recording was automatically segmented at

musical rests, based on the musical score, to reduce memory usage during training. For evaluation, we selected 101 segments not included in the training set, each ranging from 4.21 to 14.7 seconds in duration. For model inputs, we used WORLD features [3] extracted with a 10 ms frame shift, as their design for flexible speech manipulation makes them well-suited for evaluating F_0 transformation performance. Specifically, F_0, spectral envelope, and aperiodicity were extracted using the Harvest [45], CheapTrick [46], and D4C [47] algorithms, respectively. The spectral envelope and aperiodicity were parameterized as 60- and 40-dimensional mel-generalized cepstral coefficients, respectively. Pitch-shift was carried out by altering the F_0 value used for conditioning the model.

7.5.3 Model details

As a baseline, we employed SiFi-GAN [77], a high-quality vocoder with strong F_0 transformation performance. SiFi-GAN decomposes HiFi-GAN [67] into a source-filter-like architecture while incorporating sinusoidal input as Harmonic-Net [38]. For full-band synthesis, we added an upsampling block, doubled the initial channels to 1024, and set the upsampling rates to $(5, 4, 4, 3, 2)$ to match the 10 ms (480 samples) frame shift. We also prepared SiFi-GAN*, which incorporates the anti-aliasing mechanism of BigVGAN [11], to examine its effectiveness for pitch shifting. For Wavehax, we increased the frequency-direction kernel size from 7 to 15 and set the FFT size/window length to 20 ms with a 10 ms frame shift for full-band synthesis. The discriminators in all models were modified from the configuration in Section 7.5.3. For the MPD [67], we added blocks with periods of 13 and 17. For the MRD [68], we added a sub-discriminator with an FFT size and window length of 4096 and a frame shift of 1024, while removing the sub-discriminator with the shortest configuration. All models were

Table 7.4: *Objective evaluation results for the F_0 conversion experiment on singing voice. All metrics are the same as those described in Section 7.4.4, except that the smallest FFT size and window length used for the STFT metric were set to 4096 and 1024, respectively. Mel-cepstral distortion (MCD) was measured using 25-dimensional mel-cepstral coefficients computed from the spectral envelope extracted by the CheapTrick algorithm [46], to reduce the influence of F_0 conversion. As a reference, scores for analysis-synthesis using the WORLD vocoder [3] are also provided. Boldface indicates the best score.*

	WORLD	SiFi-GAN	SiFi-GAN*	Wavehax
Copy Synthesis (0 octave shift)				
VUV↓	4	**2**	**2**	**2**
RMSE↓	0.083	0.061	**0.054**	0.069
MCD↓	7.338	6.466	6.540	**6.436**
STFT↓	0.953	0.894	0.891	**0.883**
−1 octave shift				
VUV↓	4	3	**2**	3
RMSE↓	0.101	**0.064**	**0.064**	0.089
MCD↓	7.382	6.691	6.671	**6.500**
−0.5 octave shift				
VUV↓	4	**2**	**2**	**2**
RMSE↓	0.093	0.057	**0.052**	0.077
MCD↓	7.344	6.565	6.580	**6.432**
+0.5 octave shift				
VUV↓	4	**2**	**2**	3
RMSE↓	0.088	0.080	**0.074**	0.086
MCD↓	7.724	7.032	7.087	**6.981**
+1 octave shift				
VUV↓	7	**6**	**6**	**6**
RMSE↓	**0.099**	0.105	0.101	0.100
MCD↓	8.823	8.426	8.609	**8.234**

Table 7.5: *This table shows the number of model parameters, GMACs, and RTF on a single GPU and CPU of the full-band models, calculated in the same manner as in Table 7.2.*

	SiFi-GAN	SiFi-GAN*	Wavehax
Parameters↓	35.21 M	35.21 M	2.019 M
GMACs↓	53.14	57.02	5.572
RTF (GPU)↓	0.0107	0.0263	0.0048
RTF (CPU)↓	2.123	5.086	0.694

trained under the same settings as in the first experiment, except that the total training steps were reduced from 1000K to 500K, with a batch size of 8 and batch length of 24K samples, since the simpler single-singer dataset makes training easier.

7.5.4 Results

We evaluated the models at pitch-shift ratios of 0, ±0.5, and ±1 octaves. The results of the objective evaluation are shown in Table 7.4. The SiFi-GAN-based model outperforms Wavehax in terms of VUV and RMSE, which can be attributed to the inductive bias of its time-domain architecture that facilitates harmonic generation (Section 7.3.1) and benefits from the source-filter structure [77]. In contrast, Wavehax achieves lower MCD and STFT distances, indicating more accurate reproduction of the overall spectral shape and higher fidelity in high-frequency reconstruction. We attribute these differences in MCD and STFT to spectral distortion in the high-frequency range, at least partially caused by aliasing. As shown in Fig. 7.7, the SiFi-GAN-based model exhibits noticeable high-frequency distortion, whereas Wavehax more clearly preserves high-frequency harmonic components. These distortions become more prominent in segments with higher F_0 values or in pitch shifts with positive octaves, which is consistent with our hypothesis in Section 7.2.2 that the impact of aliasing becomes more

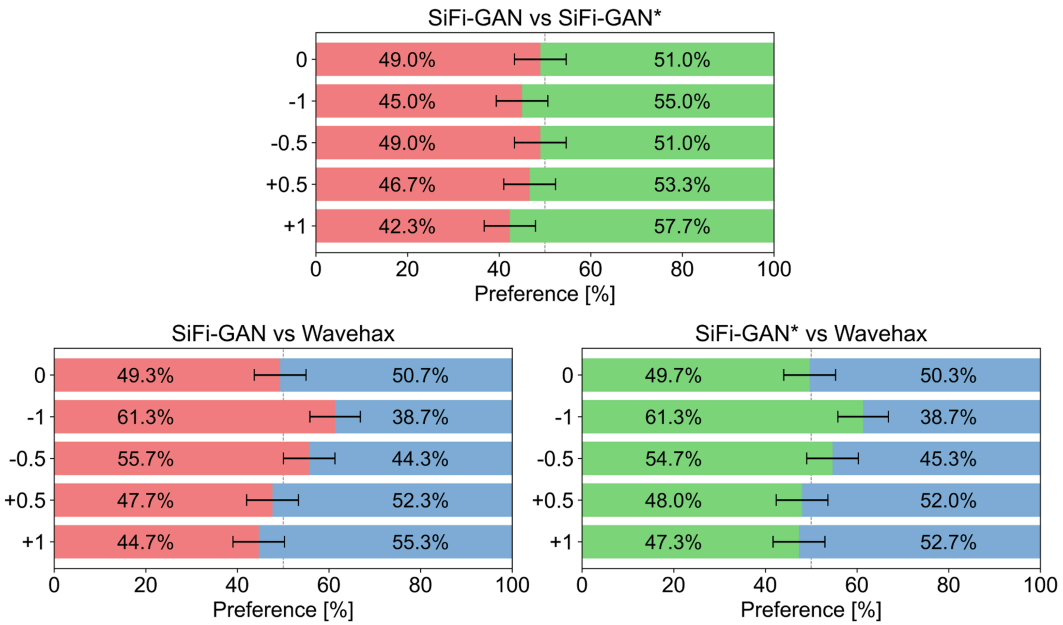

Figure 7.6: *Results of the preference test for octave shifts of* 0, ± 0.5 *and* ± 1. *Error bars indicate the 95% confidence intervals.*

pronounced at higher F_0.

We also conducted a listening preference test on singing voice quality. For each octave-shift condition, 15 native Japanese non-expert listeners participated independently. In each trial, participants were presented with a pair of samples generated under the same condition but by different models, and were asked to choose the one with better perceived quality. Each participant evaluated 20 sample pairs per condition. The results are shown in Fig. 7.6. Despite its much smaller model size and faster synthesis (as shown in Table 7.5), Wavehax achieved voice quality comparable to or better than SiFi-GAN in both the no-shift condition and upward octave shifts. Given the differences observed in high-frequency reproduction, this degradation is likely influenced by aliasing effects, whereas Wavehax mitigates this issue. On the other hand, for downward octave shifts, Wavehax performed worse than SiFi-GAN. However, in the

speech analysis-synthesis experiments with multiple speakers (described in Section 7.4), Wavehax was able to handle even lower unseen F_0 values without exhibiting this issue. This observation suggests the presence of phenomena similar to those described in the following section.

7.5.5 Discussion

We found that lowering F_0 disrupted the harmonic structure in the complex spectrogram output from Wavehax. One hypothesis we considered was that, although no explicit aliasing occurs in the time-domain waveform (i.e., spectral folding), the complex spectrogram can suffer from image-domain aliasing when processed by 2D CNNs, potentially contributing to the observed degradation. To test this hypothesis, we conducted preliminary experiments incorporating two types of 2D anti-aliasing techniques into Wavehax: (1) a 2D extension of the 1D anti-aliasing mechanism used in BigVGAN [11], which is conceptually equivalent to the image-domain strategy proposed in StyleGAN3 [10], and (2) a method inspired by JenGAN [39], which applies random shift and inverse-shift operations to promote shift invariance in convolutional layers and indirectly suppress aliasing. However, neither method led to any improvement in synthesis quality under the low-F_0 condition. These results suggest that image-domain aliasing is unlikely to be the primary cause of the degradation.

To further investigate the possible causes of these artifacts, we analyze the patterns of the complex spectrograms predicted by Wavehax. As shown in Fig. 7.8, the predicted magnitude and phase spectrograms from Wavehax differ from the patterns typically observed in natural speech. This illustrates that the complex spectrogram is an inherently redundant representation, and the model tends to learn structural patterns that are easier to represent. On the other hand, the learned structural patterns possibly are

sensitive to the distribution of the training data. The present results imply that this influence can limit the generalization capability of the model. In particular, at low F_0, where harmonics are densely spaced, inconsistencies in the complex spectrogram can lead to misaligned harmonic components in the waveform, degrading the naturalness of the synthesized speech. One possible way to alleviate this limitation is to increase the diversity and coverage of the training data. Developing more fundamental solutions to improve robustness under such challenging conditions remains an important direction for future work.

7.6 Conclusion

This chapter presented Wavehax, an aliasing-free neural vocoder proposed as the third theme of this dissertation. This work addresses the issue of physical inconsistency that arises between the continuous physical process of speech production and the discrete computations of neural networks. To analyze the aliasing problem inherent in time-domain processing, we re-examined the internal mechanisms of neural vocoders from a signal-processing perspective. By contrasting time- and time-frequency-domain processing, we theoretically demonstrated why the latter can inherently avoid aliasing, and further showed that the combination of 2D CNNs and harmonic spectrograms plays a crucial role in achieving stable and accurate complex-spectrogram estimation. Through this design, Wavehax avoids aliasing in latent representations while overcoming the lack of harmonic-generation bias typically seen in time-frequency-domain processing. As a result, it achieves a substantial improvement in computational efficiency while maintaining speech quality comparable to high-fidelity time-domain vocoders. This study establishes the effectiveness of aliasing-free waveform synthesis and highlights a new direction for achieving numerical and structural stability by engineering

neural networks that emulate continuous-time signal-processing properties.

Nevertheless, several challenges remain. One limitation is that Wavehax tends to learn data-dependent time-frequency representations owing to the redundancy of the complex spectrogram, which may cause degraded generalization under low-F_0 conditions (as discussed in the second experiment). Moreover, across Chapters 5 to 7, some unresolved design issues and methodological constraints persist. These remaining challenges will be comprehensively discussed in Chapter 8.

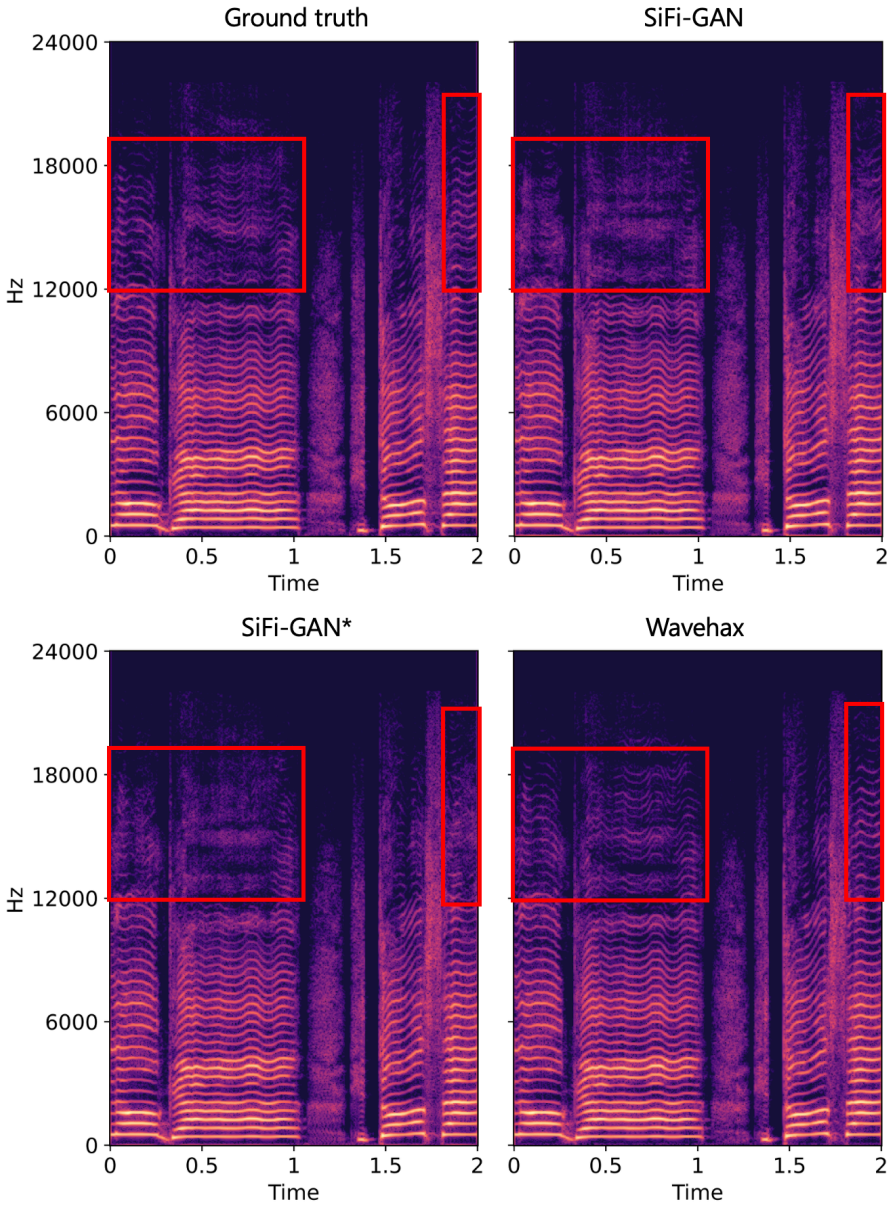

Figure 7.7: *Log-magnitude spectrograms of the ground truth, SiFi-GAN, SiFi-GAN*, and Wavehax (left to right). The red boxes highlight regions where the SiFi-GAN outputs exhibit degraded harmonic structures, likely caused by aliasing.*

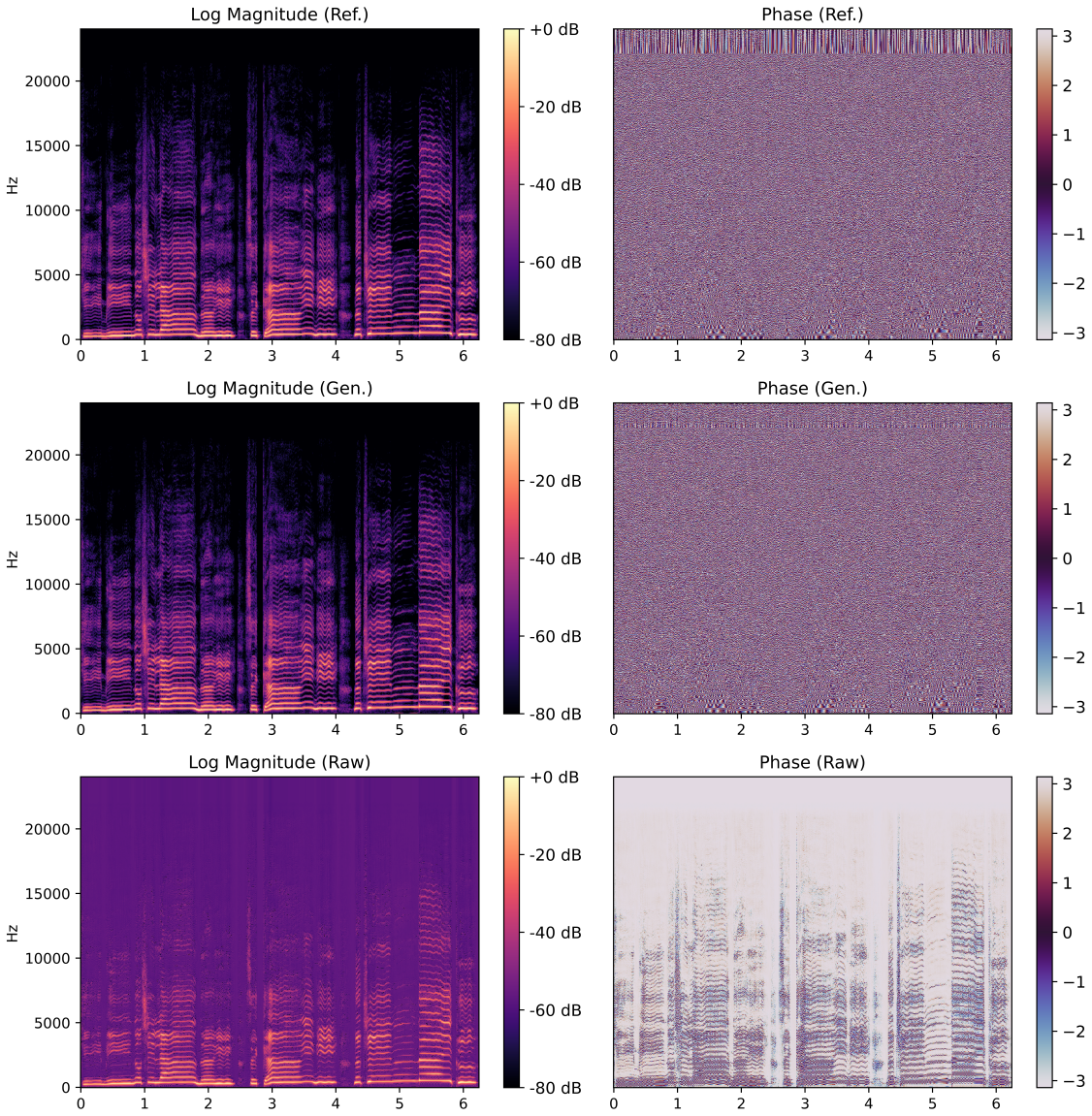

Figure 7.8: *Comparison of complex spectrograms. Each row shows a pair of spectrograms, with the log-magnitude representation on the left and the corresponding phase representation on the right. The top row (Ref.) presents the ground-truth spectrogram analyzed using the same STFT parameters as the model (FFT/window 20 ms, frame shift 10 ms). The middle row (Gen.) shows the STFT analysis of the waveform obtained by applying iSTFT to the complex spectrogram predicted by Wavehax. The bottom row (Raw) directly visualizes the complex spectrogram predicted by Wavehax, where the real and imaginary components are converted into magnitude and phase for display. Note that Wavehax outputs complex spectrograms, which are converted to waveforms via iSTFT; the middle row therefore corresponds to a re-analysis of the generated waveform.*

8 Conclusions

8.1 Thesis Summary

This thesis presents a unified framework for the design of neural vocoders that incorporate the physical process of speech production. The goal is to construct a neural vocoder that maintains physical consistency in speech generation while exploiting the flexible representational power of deep learning. By bridging the conceptual and methodological gap between signal processing and deep learning, this thesis explores a new design principle aimed at improving the overall naturalness, controllability, computational efficiency, and compactness of speech synthesis.

Chapter 2 describes the fundamental concepts, representative implementations, and limitations of source-filter models. Particular attention is given to WORLD [3], a high-quality vocoder based on signal processing, where the principles of speech analysis and synthesis, as well as the formulation of parametric representations, are discussed in detail. The advantages of explicit structure and controllability in source-filter models are clarified, together with the expressive limitations that arise from linear approximations and simplified model assumptions.

Chapter 3 provides a systematic overview of the technical development of neural vocoders based on deep generative modeling. The chapter reviews the genealogy of major generative model families, including autoregressive models, normalizing flows, and generative adversarial networks, and clarifies their probabilistic formulations, learn-

ing objectives, and practical trade-offs. Through this review, it is shown how neural vocoders have enabled high-quality waveform generation beyond the limitations of classical signal-processing approaches, while also revealing fundamental challenges for practical deployment.

Chapter 4 surveys neural vocoder designs that incorporate aspects of the physical speech production mechanism. The chapter reviews representative approaches based on harmonic-plus-noise modeling, linear filter integration, F_0-driven mechanisms, continuous–discrete time considerations, and time–frequency domain modeling. These approaches demonstrate how physically motivated inductive biases can improve the performance, while also highlighting unresolved issues such as aliasing artifacts, generalization to unseen conditions, and the coherent integration of physical models with data-driven learning.

Chapter 5 proposes uSFGAN, a unified source-filter generative adversarial network that jointly optimizes the source and filter components under an adversarial learning framework. This approach maintains functional separability derived from source-filter theory, while achieving simultaneous optimization of both components within a single neural network. The proposed unified source-filter modeling achieved a high-level balance between speech quality and F_0 controllability, surpassing conventional methods.

Chapter 6 introduces SiFi-GAN, which integrates unified source-filter modeling with an upsampling architecture. It demonstrates that functional separation based on a cascade structure derived from source-filter theory remains effective even when the upsampling process is embedded in the network. By employing low temporal resolution processing, the proposed model realizes efficient waveform generation that maintains speech quality while enabling real-time synthesis on a single CPU.

Chapter 7 theoretically analyzes the internal signal processing of neural vocoders and

investigates the influence of aliasing, an artifact inherent to discrete-time processing, on speech quality, both theoretically and experimentally. An aliasing-free neural vocoder, Wavehax, is proposed to fundamentally eliminate aliasing through time-frequency domain processing. Furthermore, it is shown that the combination of two-dimensional convolution and harmonic signal modeling plays a crucial role in achieving high-quality synthesis. By removing artificial distortions that do not occur in the physical process of vocal production, the proposed method demonstrates improvements in naturalness, robustness, and efficiency.

Through these studies, this thesis presents a theoretical framework for neural network design that integrates the physical principles of speech production into the architecture itself. Rather than a simple hybridization of signal processing and deep learning, this work introduces physical consistency as a practical inductive bias in neural vocoder design. The results demonstrate that alignment with the physical process of sound generation contributes directly to improved synthesis performance, providing fundamental insights for the next generation of neural vocoder design. These findings also hold potential applicability to neural audio codecs, which have attracted considerable attention since the emergence of SoundStream [126].

8.2 Future Work

8.2.1 Extending Wavehax with a Learnable Source Model

As a direction for future work, the source generation module of Wavehax could be extended into a trainable structure. In the current implementation, the excitation process in Wavehax is analytically defined, offering simplicity and high stability, but at the cost of limited expressive power in the source model. To alleviate this limitation,

it would be effective to introduce a learnable source module and adopt a framework that allows joint optimization of the source and filter components. Specifically, incorporating an integrated source-filter modeling approach, as in uSFGAN, would enable nonlinear and flexible source modeling. Alternatively, as demonstrated in FIRNet [32], a trainable linear filter for residual signal generation can be integrated as a front-end module within Wavehax. Moreover, applying the harmonic-plus-noise decomposition proven effective in hn-uSFGAN to the source generation stage can further enhance performance. Through such extensions, making the source module learnable and optimizing it jointly with the filter, Wavehax can be developed into a neural vocoder with even higher speech quality and controllability.

8.2.2 Exploration of Alternative Pitch Representations

In this study, the F_0 was employed as the primary control parameter to represent the pitch of speech. Each system analytically computed the phase rotation speed of harmonic components based on F_0, thereby generating smooth periodic excitation signals that served as model inputs. However, this design strongly depends on the accuracy of F_0 estimation, and estimation errors directly affect the quality of synthesized speech. Moreover, F_0-based pitch representation is not well-suited for non-periodic sounds such as percussive noises or for cases involving missing fundamentals, and it is difficult to extend to polyphonic signals containing multiple fundamental frequencies. Although, in principle, multi-F_0 modeling or part-wise source separation could handle such cases, practical applications remain limited due to the lack of reliable analysis models and training data for polyphonic speech.

From the viewpoint of controllability in speech synthesis, it is also desirable to manipulate pitch using higher-level, perceptually meaningful features rather than low-level

physical quantities such as F_0. Against this background, introducing alternative pitch representations independent of F_0 is a promising direction. For example, discrete pitch representations derived from musical scores could enable abstract and musically consistent pitch control.

In practical applications, acoustic features, including F_0, are estimated by an acoustic model from higher-level representations such as text or musical scores. As a result, estimation errors in these features are inevitable and can propagate through the synthesis pipeline, leading to perceptual degradation. Importantly, the impact of such errors on the synthesized speech is affected by the design choices of the acoustic model, the acoustic feature representation, and the neural vocoder. From a system-level perspective, this highlights the importance of exploring pitch representations that improve robustness and perceptual quality across the entire synthesis pipeline, rather than optimizing individual components in isolation.

8.2.3 Analysis Model Considering Speech Production Mechanism

Explicit control parameters such as F_0 inherently face limitations, making it difficult to simultaneously achieve both interpretability and expressive power in speech synthesis. Neural vocoders, on the other hand, learn nonlinear mappings from acoustic features to waveforms in a data-driven manner, capturing high-order dependencies and contextual variations beyond the capability of signal-processing-based models. However, such flexibility often results in overfitting to statistical dependencies in the training data, causing latent representations to be tightly bound to the data distribution. As observed in the experiments of Appendix 1, conflicts between latent F_0 information implicitly contained in mel-spectrograms and explicitly provided F_0 sequences can

degrade synthesis quality, revealing ambiguity in the underlying representation.

Traditional signal processing-based approaches extract parametric features such as F_0 and spectral envelopes via hand-designed algorithms, enabling explicit source-filter separation and high controllability. Nevertheless, these methods rely on assumptions of linearity and stationarity, which prevent them from accurately modeling the complex, nonlinear generative mechanisms of human speech. Moreover, F_0 and spectral envelopes are highly correlated, and independent manipulation often results in noticeable quality degradation. Additionally, when the underlying signal violates the assumptions of the analysis model, estimation errors increase significantly, and robustness under noisy or real-world conditions remains limited.

To overcome these limitations, a data-driven analysis framework that reflects the physical generation principles of speech is required. One promising direction is the joint optimization of an encoder (analyzer) and a decoder (synthesizer), where the decoder, designed based on the physical production process, is given as a known structure, and the encoder learns latent representations consistent with it. Recent studies such as NANSY++ [127] and NaturalSpeech3 [128] represent early examples of this approach, aiming to learn disentangled representations of linguistic content, prosody, and timbre through self-supervised learning. However, the disentanglement in NaturalSpeech3 relies on information bottlenecks and supervised F_0 extractors, and thus does not achieve truly data-driven analysis. Similarly, although NANSY++ learns latent representations via self-supervision, its decoder does not explicitly reflect the physical generative principles of speech, resulting in ambiguous correspondence between latent variables and physical factors, and unstable behavior in pitch control or timbre conversion.

As a future direction, a cooperative learning framework that aligns the encoder and

decoder with the physical speech production process is highly promising. Specifically, by fixing a decoder with an explicit physical structure and guiding the encoder to learn latent representations consistent with that structure, it is expected that the resulting latent space will preserve physical interpretability while retaining sufficient expressive capacity. This approach could yield higher-order separation of generative factors such as source and filter characteristics, providing a physically grounded and robust representation framework for neural vocoders.

8.2.4 Quantitative Evaluation of Extrapolated Speech

In this study, the effectiveness of pitch control was evaluated by testing F_0 extrapolation beyond the natural pitch range of each speaker, that is, by transforming the F_0 toward higher or lower values outside the training data distribution. This evaluation is important for verifying whether the model can maintain functional independence between the source and filter, and thus generalize to unseen pitch conditions while preserving the physical correspondence of the generative process. However, since such extrapolated generation often produces physically unrealistic speech, there are cases in which the generated sound is perceived as natural even though it is not physically plausible, raising questions about the validity of subjective evaluation. In this study, perceptual listening tests were used as the primary evaluation method, but relying solely on subjective judgments has limitations, as it cannot quantify physical consistency or statistical validity.

Therefore, there is a need to establish new quantitative measures for evaluating the naturalness of extrapolated speech in probabilistic and statistical terms. Specifically, probabilistic generative models such as autoregressive models, normalizing flows, and GAN discriminators can be employed to compute the likelihood or distributional

consistency of the generated signals. Such probabilistic metrics could complement subjective assessments and provide an objective basis for evaluating extrapolated or out-of-distribution speech generation. In particular, large-scale generative models trained on diverse audio signals, rather than being restricted to speech alone, are encouraged to internalize statistical constraints that are common to real-world sound generation, instead of relying on surface-level features specific to individual tasks. As a result, even without explicitly incorporating physical models, such models may implicitly reflect distributional structures that are consistent with the physical processes underlying sound generation.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Prof. Tomoki Toda in Nagoya University, for his unwavering guidance and support throughout my studies at the university. His expertise and insightful feedback continually inspired me to refine my thinking and elevate the quality of my work, and he was the one who taught me the true joy of conducting research. He has been my role model both as a researcher and as a person.

I am deeply grateful to the Voice Team at LINE Corporation, and later to the Speech Processing Team at LY Corporation, who supported me from my earliest days in the field through my internship and collaborative research. They played a central role in my development as a speech research engineer over nearly four years, during which I had the privilege of working with them. I would also like to express my sincere appreciation to Mr. Yamamoto, who guided me with exceptional care both as a mentor and as a senior Ph.D. student.

My sincere appreciation also goes to the Speech and Language Team at SB Intuitions Corporation, for welcoming me as an intern, granting me access to large-scale computing resources, and providing an environment that encouraged exploration and growth. I truly appreciate the guidance and support provided by all team members, and I would like to extend particular thanks to Mr. Yui Sudo and Mr. Yusuke Fujita for their dedicated mentorship.

I am indebted to DubGuild Inc., where I had the opportunity to contribute to research and development in speech synthesis technologies as an advisor. I am especially grateful to Mr. Masatoshi Otake, the leader of DubGuild Inc., for offering me this opportunity and for fostering a collaborative environment that brought me into contact with exceptional colleagues.

I would also like to extend my sincere appreciation to the members of the Toda Laboratory for their kindness and support. I am particularly grateful to Ms. Nami Noro, whose dedicated administrative support greatly facilitated my research activities. I am also deeply thankful to Prof. Yusuke Yasuda, Prof. Wen-Chin Huang, Dr. Kazuhiro Kobayashi of TARVO Inc., Dr. Tomoki Hayashi of Human Dataware Lab Co., Ltd., and Dr. Yi-Chiao Wu of FAIR for their invaluable guidance. Prof. Yusuke Yasuda offered generous advice on both research and proposal writing, and I gained valuable insights from him through many academic interactions. Prof. Wen-Chin Huang, who guided me as a senior colleague and continues to support me as an assistant professor, has long been someone I have deeply respected. My part-time work at TARVO further deepened my understanding of the connection between research and its practical social impact, thanks to Dr. Kazuhiro Kobayashi and Dr. Tomoki Hayashi, who taught me its importance through their dedicated mentorship. In the early stages of my research, when I was still unfamiliar with many aspects of the field, I received thoughtful guidance from Dr. Yi-Chiao Wu, who was then a senior Ph.D. student. I am also grateful to my cohort peers, Ms. Yuka Hashizume and Mr. Lester Phillip Violeta, with whom I shared many challenges and mutual encouragement throughout this long journey toward completing our doctoral degrees.

I would like to express my sincere appreciation for the financial support provided by the Japan Society for the Promotion of Science (JSPS) Research Fellowship for

Young Scientists (DC2), the Nagoya University Interdisciplinary Frontier Fellowship, and the TMI WISE Program: Graduate Program for Lifestyle Revolution Based on Transdisciplinary Mobility Innovation. Their support allowed me to devote myself fully to my research.

Finally, I would like to offer my heartfelt gratitude to my family and my wife for their constant encouragement, warmth, and wise counsel. I also thank the many friends I have met around the world, as well as my undergraduate classmates at Nagoya University, with whom I learned and grew through shared effort. You all enriched my life far beyond research, and for that, I am truly grateful.

References

[1] H. W. Dudley, "Remaking speech," *The Journal of the Acoustical Society of America*, vol. 11, no. 2, pp. 169–177, 1939.

[2] H. Kawahara, I. Masuda-Katsuse, and A. De Cheveigne, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, no. 3-4, pp. 187–207, 1999.

[3] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications," *IEICE Trans. Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[4] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," in *Proc. SSW*, 2016, p. 125.

[5] A. Tamamori, T. Hayashi, K. Kobayashi, K. Takeda, and T. Toda, "Speaker-Dependent WaveNet Vocoder," in *Proc. Interspeech*, 2017, pp. 1118–1122.

[6] T. Hayashi, A. Tamamori, K. Kobayashi, K. Takeda, and T. Toda, "An investigation of multi-speaker training for wavenet vocoder," in *Proc. ASRU*, 2017, pp. 712–718.

[7] R. McAulay and T. Quatieri, "Speech Analysis/Synthesis Based on a Sinusoidal Representation," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 34, no. 4, pp. 744–754, 1986.

[8] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel WaveGAN: A Fast Waveform Generation Model Based on Generative Adversarial Networks with Multi-Resolution Spectrogram," in *Proc. ICASSP*, 2020, pp. 6199–6203.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Proc. NeurIPS*, vol. 27, 2014, pp. 2672–2680.

[10] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-Free Generative Adversarial Networks," in *Proc. NeurIPS*, 2021.

[11] S. gil Lee, W. Ping, B. Ginsburg, B. Catanzaro, and S. Yoon, "BigVGAN: A Universal Neural Vocoder with Large-Scale Training," in *Proc. ICLR*, 2023, 20 pages.

[12] H. Nyquist, "Certain Topics in Telegraph Transmission Theory," *Trans. American Institute of Electrical Engineers*, vol. 47, no. 2, pp. 617–644, 1928.

[13] C. Shannon, "Communication in the Presence of Noise," in *Proc. IRE*, vol. 37, no. 1, 1949, pp. 10–21.

[14] L. Juvela, B. Bollepalli, M. Airaksinen, and P. Alku, "High-pitched excitation generation for glottal vocoding in statistical parametric speech synthesis using a deep neural network," in *Proc. ICASSP*, 2016, pp. 5120–5124.

[15] M. Airaksinen, B. Bollepalli, L. Juvela, Z. Wu, S. King, and P. Alku, "GlottDNN — A Full-Band Glottal Vocoder for Statistical Parametric Speech Synthesis," in *Proc. Interspeech*, 2016, pp. 2473–2477.

[16] L. Juvela, B. Bollepalli, V. Tsiaras, and P. Alku, "GlotNet — A Raw Waveform Model for the Glottal Excitation in Statistical Parametric Speech Synthesis," *IEEE/ACM Trans. ASLP*, vol. 27, no. 6, pp. 1019–1030, 2019.

[17] B. Bollepalli, L. Juvela, and P. Alku, "Generative adversarial network-based glottal waveform model for statistical parametric speech synthesis," in *Proc. Interspeech*, 2017, pp. 3394–3398.

[18] L. Juvela, B. Bollepalli, J. Yamagishi, and P. Alku, "Waveform generation for text-to-speech synthesis using pitch-synchronous multi-scale generative adversarial networks," in *Proc. ICASSP*, 2019, pp. 6915–6919.

[19] L. Juvela, B. Bollepalli, J. Yamagishi, and P. Alku, "GELP: GAN-excited linear prediction for speech synthesis from mel-spectrogram," in *Proc. Interspeech*, 2019, pp. 694–698.

[20] M. Hwang, F. Soong, E. Song, X. Wang, H. Kang, and H. Kang, "LP-WaveNet: Linear Prediction-based WaveNet Speech Synthesis," in *Proc. APSIPA*, 2020, pp. 810–814.

[21] J.-M. Valin and J. Skoglund, "LPCNet: Improving Neural Speech Synthesis through Linear Prediction," in *Proc. ICASSP*, 2019, pp. 5891–5895.

[22] M.-J. Hwang, E. Song, R. Yamamoto, F. Soong, and H.-G. Kang, "Improving LPCNET-Based Text-to-Speech with Linear Prediction-Structured Mixture Density Network," in *Proc. ICASSP*, 2020, pp. 7219–7223.

[23] K. Matsubara, T. Okamoto, R. Takashima, T. Takiguchi, T. Toda, Y. Shiga, and H. Kawai, "Full-Band LPCNet: A Real-Time Neural Vocoder for 48 kHz Audio With a CPU," *IEEE Access*, vol. 9, pp. 94 923–94 933, 2021.

[24] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *Proc. ICLR*, 2020, 19 pages.

[25] D.-Y. Wu, W.-Y. Hsiao, F.-R. Yang, O. Friedman, W. Jackson, S. Bruzenak, Y.-W. Liu, and Y.-H. Yang, "DDSP-based Singing Vocoders: A New Subtractive-based Synthesizer and A Comprehensive Evaluation," *Proc. ISMIR*, 2022, 8 pages.

[26] C.-Y. Yu and G. Fazekas, "Singing Voice Synthesis Using Differentiable LPC and Glottal-Flow-Inspired Wavetables," in *Proc. ISMIR*, 2023, 9 pages.

[27] C.-Y. Yu and G. Fazekas, "Differentiable Time-Varying Linear Prediction in the Context of End-to-End Analysis-by-Synthesis," in *Proc. Interspeech 2024*, 2024, pp. 1820–1824.

[28] P. L. Tobing and T. Toda, "High-Fidelity and Low-Latency Universal Neural Vocoder Based on Multiband WaveRNN with Data-Driven Linear Prediction for Discrete Waveform Modeling," in *Proc. Interspeech*, 2021, pp. 2217–2221.

[29] Krishna Subramani and Jean-Marc Valin and Umut Isik and Paris Smaragdis and Arvindh Krishnaswamy, "End-to-end LPCNet: A Neural Vocoder With Fully-Differentiable LPC Estimation," in *Proc. Interspeech*, 2022, pp. 818–822.

[30] Z. Liu, K. Chen, and K. Yu, "Neural Homomorphic Vocoder," in *Proc. Interspeech*, 2020, pp. 240–244.

[31] T. Yoshimura, S. Takaki, K. Nakamura, K. Oura, Y. Hono, K. Hashimoto, Y. Nankaku, and K. Tokuda, "Embedding a Differentiable Mel-Cepstral Synthesis Filter to a Neural Speech Synthesis System," in *Proc. ICASSP*, 2023, pp. 1–5.

[32] Y. Ohtani, T. Okamoto, T. Toda, and H. Kawai, "FIRNet: Fundamental Frequency Controllable Fast Neural Vocoder With Trainable Finite Impulse Response Filter," in *Proc. ICASSP*, 2024, pp. 10 871–10 875.

[33] X. Wang, S. Takaki, and J. Yamagishi, "Neural Source-Filter Waveform Models for Statistical Parametric Speech Synthesis," *IEEE/ACM Trans. ASLP*, vol. 28, pp. 402–415, 2020.

[34] Y. Hono, S. Takaki, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Periodnet: A Non-Autoregressive Waveform Generation Model with a Structure Separating Periodic and Aperiodic Components," in *Proc. ICASSP*, 2021, pp. 6049–6053.

[35] M.-J. Hwang, R. Yamamoto, E. Song, and J.-M. Kim, "High-Fidelity Parallel WaveGAN with Multi-Band Harmonic-Plus-Noise Model," in *Proc. Interspeech*, 2021, pp. 2227–2231.

[36] Y.-C. Wu, T. Hayashi, P. L. Tobing, K. Kobayashi, and T. Toda, "Quasi-Periodic WaveNet: An Autoregressive Raw Waveform Generative Model With Pitch-Dependent Dilated Convolution Neural Network," *IEEE/ACM Trans. ASLP*, vol. 29, pp. 1134–1148, 2021.

[37] Y.-C. Wu, T. Hayashi, T. Okamoto, H. Kawai, and T. Toda, "Quasi-Periodic Parallel WaveGAN: A Non-Autoregressive Raw Waveform Generative Model With

Pitch-Dependent Dilated Convolution Neural Network," *IEEE/ACM Trans. ASLP*, vol. 29, pp. 792–806, 2021.

[38] K. Matsubara, T. Okamoto, R. Takashima, T. Takiguchi, T. Toda, and H. Kawai, "Harmonic-Net: Fundamental Frequency and Speech Rate Controllable Fast Neural Vocoder," *IEEE/ACM Trans. ASLP*, vol. 31, pp. 1902–1915, 2023.

[39] H. Cho, J. Lee, and W. Jung, "JenGAN: Stacked Shifted Filters in GAN-Based Speech Synthesis," in *Proc. Interspeech*, 2024, pp. 3879–3883.

[40] Z. Shang, H. Zhang, P. Zhang, L. Wang, and T. Li, "Analysis and Solution to Aliasing Artifacts in Neural Waveform Generation Models," *Applied Acoustics*, vol. 203, p. 109183, 2023.

[41] D. W. Griffin and J. S. Lim, "Multiband excitation vocoder," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 8, pp. 1223–1235, 1988.

[42] A. McCree and T. Barnwell, "A mixed excitation lpc vocoder model for low bit rate speech coding," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 4, pp. 242–250, 1995.

[43] B. S. Atal and M. R. Schroeder, "Predictive coding of speech signals," in *Proc. Speech Commun. and Processing*, 1967, pp. 360–361.

[44] F. Itakura and S. Saito, "Analysis synthesis telephony based on the maximum likelihood method," in *Proc. ICA*, 1968, pp. C17–C20.

[45] M. Morise, "Harvest: A High-Performance Fundamental Frequency Estimator from Speech Signals," in *Proc. Interspeech*, 2017, pp. 2321–2325.

[46] M. Morise, "Cheaptrick, a spectral envelope estimator for high-quality speech synthesis," *Speech Communication*, vol. 67, pp. 1–7, 2015.

[47] M. Morise, "D4C, a band-aperiodicity estimator for high-quality speech synthesis," *Speech Communication*, vol. 84, pp. 57–65, 2016.

[48] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A Flow-based Generative Network for Speech Synthesis," in *Proc. ICASSP*, 2019, pp. 3617–3621.

[49] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "DiffWave: A Versatile Diffusion Model for Audio Synthesis," in *Proc. ICLR*, 2021, 17 pages.

[50] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient Neural Audio Synthesis," in *Proc. ICML*, 2018, pp. 2415–2424.

[51] Z. Jin, A. Finkelstein, G. J. Mysore, and J. Lu, "FFTNet: A real-time speaker-dependent neural vocoder," in *Proc. ICASSP*, 2018, pp. 2251–2255.

[52] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis, "Parallel WaveNet: Fast High-Fidelity Speech Synthesis," in *Proc. ICML*, 2018, pp. 3918–3926.

[53] W. Ping, K. Peng, and J. Chen, "ClariNet: Parallel wave generation in end-to-end text-to-speech," in *Proc. ICLR*, 2019, 15 pages.

[54] K. Arora, L. El Asri, H. Bahuleyan, and J. Cheung, "Why exposure bias matters: An imitation learning perspective of error accumulation in language generation," in *Proc. ACL*, 2022, pp. 700–710.

[55] S. Kim, S.-G. Lee, J. Song, J. Kim, and S. Yoon, "FloWaveNet : A Generative Flow for Raw Audio," in *Proc. ICML*, 2019, pp. 3370–3378.

[56] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using Real NVP," in *Proc. ICLR*, 2017, 32 pages.

[57] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *Proc. NeurIPS*, vol. 31, 2018, 10 pages.

[58] W. Ping, K. Peng, K. Zhao, and Z. Song, "WaveFlow: A Compact Flow-Based Model for Raw Audio," in *Proc. ICML*, 2020, pp. 7706–7716.

[59] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improved variational inference with inverse autoregressive flow," in *Proc. NeurIPS*, 2016, pp. 4743–4751.

[60] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural Ordinary Differential Equations," *Proc. NeurIPS*, vol. 31, 2018, 13 pages.

[61] H. Kim, H. S. Lee, W. H. Kang, S. J. Cheon, B. J. Choi, and N. S. Kim, "WaveNODE: A Continuous Normalizing Flow for Speech Synthesis," in *Proc. ICML*, 2020, 8 pages.

[62] N.-Q. Wu and Z.-H. Ling, "WaveFFJORD: FFJORD-Based Vocoder for Statistical Parametric Speech Synthesis," in *Proc. ICASSP*, 2020, pp. 7214–7218.

[63] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based Generative Adversarial Networks," in *Proc. ICLR*, 2017, 17 pages.

[64] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least Squares Generative Adversarial Networks," in *Proc. ICCV*, 2017, 9 pages.

[65] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *Proc. ICLR*, 2014, 14 pages.

[66] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis," in *Proc. NeurIPS*, vol. 32, 2019, 14 pages.

[67] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis," in *Proc. NeurIPS*, vol. 33, 2020, pp. 17 022–17 033.

[68] W. Jang, D. Lim, J. Yoon, B. Kim, and J. Kim, "UnivNet: A Neural Vocoder with Multi-Resolution Spectrogram Discriminators for High-Fidelity Waveform Generation," in *Proc. Interspeech*, 2021, pp. 2207–2211.

[69] J. You, D. Kim, G. Nam, G. Hwang, and G. Chae, "GAN Vocoder: Multi-Resolution Discriminator Is All You Need," in *Proc. Interspeech*, 2021, pp. 2177–2181.

[70] Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, and M.-H. Yang, "Mode Seeking Generative Adversarial Networks for Diverse Image Synthesis," in *Proc. CVPR*, 2019, pp. 1429–1437.

[71] T. Saeki, D. Xin, W. Nakata, T. Koriyama, S. Takamichi, and H. Saruwatari, "UTMOS: UTokyo-SaruLab System for VoiceMOS Challenge 2022," in *Proc. Interspeech*, 2022, pp. 4521–4525.

[72] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: a framework for self-supervised learning of speech representations," in *Proc. NeurIPS*, 2020, 12 pages.

[73] Qiantong Xu and Alexei Baevski and Michael Auli, "Simple and Effective Zero-shot Cross-lingual Phoneme Recognition," in *Proc. Interspeech*, 2022, pp. 2113–2117.

[74] Y. Stylianou, "Applying the harmonic plus noise model in concatenative speech synthesis," *IEEE Trans. Speech and Audio Processing*, vol. 9, no. 1, pp. 21–29, 2001.

[75] A. Paavo, "Glottal inverse filtering analysis of human voice production — a review of estimation and parameterization methods of the glottal excitation and their applications," *Sadhana*, vol. 36, no. 5, pp. 623–650, 10 2011.

[76] E. Song, K. Byun, and H.-G. Kang, "ExcitNet Vocoder: A Neural Excitation Model for Parametric Speech Synthesis Systems," in *Proc. EUSIPCO*, 2019, pp. 1–5.

[77] R. Yoneyama, Y.-C. Wu, and T. Toda, "Source-Filter HiFi-GAN: Fast and Pitch Controllable High-Fidelity Neural Vocoder," in *Proc. ICASSP*, 2023, pp. 1–5.

[78] M. Morrison, R. Kumar, K. Kumar, P. Seetharaman, A. Courville, and Y. Bengio, "Chunked Autoregressive GAN for Conditional Waveform Synthesis," in *Proc. ICLR*, 2022, 19 pages.

[79] K. Matsubara, T. Okamoto, R. Takashima, T. Takiguchi, T. Toda, and H. Kawai, "Period-HiFi-GAN: Fast and fundamental frequency controllable neu-

ral vocoder," in *Proceedings of the Acoustical Society of Japan*, Mar. 2022, pp. 901–904.

[80] Y.-C. Wu, T. Hayashi, P. L. Tobing, K. Kobayashi, and T. Toda, "Quasi-periodic WaveNet vocoder: A pitch dependent dilated convolution model for parametric speech generation," in *Proc. Interspeech*, 2019, pp. 196–200.

[81] Y.-C. Wu, T. Hayashi, T. Okamoto, H. Kawai, and T. Toda, "Quasi-Periodic Parallel WaveGAN: A Non-Autoregressive Raw Waveform Generative Model With Pitch-Dependent Dilated Convolution Neural Network," *IEEE/ACM Trans. ASLP*, vol. 29, pp. 792–806, 2021.

[82] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proc. ICML*, 2010, pp. 807–814.

[83] H. Miyamoto, S. Shiota, and H. Kiya, "Non-linear harmonic generation based blind bandwidth extension considering aliasing artifacts," in *Proc. APSIPA ASC*, 2018, pp. 1868–1874.

[84] J. Pons, S. Pascual, G. Cengarle, and J. Serrà, "Upsampling Artifacts in Neural Audio Synthesis," in *Proc. ICASSP*, 2021, pp. 3005–3009.

[85] R. Zhang, "Making Convolutional Networks Shift-Invariant Again," in *Proc. ICML*, 2019, 11 pages.

[86] A. H. Ribeiro and T. B. Schön, "How Convolutional Neural Networks Deal with Aliasing," in *Proc. ICASSP*, 2021, pp. 2755–2759.

[87] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Trans. Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[88] Y. Gong and C. Poellabauer, "Impact of Aliasing on Deep CNN-Based End-to-End Acoustic Models," in *Proc. Interspeech*, 2018, pp. 2698–2702.

[89] C. Donahue, J. McAuley, and M. Puckette, "Adversarial Audio Synthesis," in *Proc. ICLR*, 2019, 18 pages.

[90] G. Narita, J. Shimizu, and T. Akama, "GANStrument: Adversarial Instrument Sound Synthesis with Pitch-Invariant Instance Conditioning," in *Proc. ICASSP*, 2023, pp. 1–5.

[91] D. W. Griffin and J. Lim, "Signal Estimation from Modified Short-Time Fourier Transform," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[92] T. Kaneko, K. Tanaka, H. Kameoka, and S. Seki, "ISTFTNet: Fast and Lightweight Mel-Spectrogram Vocoder Incorporating Inverse Short-Time Fourier Transform," in *Proc. ICASSP*, 2022, pp. 6207–6211.

[93] T. Kaneko, H. Kameoka, K. Tanaka, and S. Seki, "iSTFTNet2: Faster and More Lightweight iSTFT-Based Neural Vocoder Using 1D-2D CNN," in *Proc. Interspeech*, 2023, pp. 4369–4373.

[94] Y. Ai and Z.-H. Ling, "APNet: An All-Frame-Level Neural Vocoder Incorporating Direct Prediction of Amplitude and Phase Spectra," *IEEE/ACM Trans. ASLP*, vol. 31, pp. 2145–2157, 2023.

[95] D. S. Dang, T. L. Nguyen, B. T. Ta, T. T. Nguyen, T. N. A. Nguyen, D. L. Le, N. M. Le, and V. H. Do, "LightVoc: An Upsampling-Free GAN Vocoder Based On Conformer And Inverse Short-time Fourier Transform," in *Proc. Interspeech*, 2023, pp. 3043–3047.

[96] H. Siuzdak, "Vocos: Closing the gap between time-domain and Fourier-based neural vocoders for high-quality audio synthesis," in *Proc. ICLR*, 2024, 15 pages.

[97] Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. CVPR*, 2022, pp. 11 966–11 976.

[98] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[99] Y. Ai and Z.-H. Ling, "A Neural Vocoder with Hierarchical Generation of Amplitude and Phase Spectra for Statistical Parametric Speech Synthesis," *IEEE/ACM Trans. ASLP*, vol. 28, pp. 839–851, 2020.

[100] Y. A. Li, C. Han, X. Jiang, and N. Mesgarani, "HiFTNet: A Fast High-Quality Neural Vocoder with Harmonic-plus-Noise Filter and Inverse Short Time Fourier Transform," *arXiv:2309.09493*, 2023.

[101] L. Ziyin, T. Hartwig, and M. Ueda, "Neural Networks Fail to Learn Periodic Functions and How to Fix It," in *Proc. NeurIPS*, vol. 33, 2020, pp. 1583–1594.

[102] Y. Li and C. Wang, "Improve GAN-based Neural Vocoder using Truncated Pointwise Relativistic Least Square GAN," in *Proc. AISS*, 2023, pp. 1–7.

[103] X. Wang and J. Yamagishi, "Neural Harmonic-plus-Noise Waveform Model with Trainable Maximum Voice Frequency for Text-to-Speech Synthesis," in *Proc. SSW*, 2019, pp. 1–6.

[104] R. Yoneyama, Y.-C. Wu, and T. Toda, "Unified Source-Filter GAN with Harmonic-plus-Noise Source Excitation Generation," in *Proc. Interspeech*, 2022, pp. 848–852.

[105] X. Wang, S. Takaki, and J. Yamagishi, "Neural source-filter-based waveform model for statistical parametric speech synthesis," in *Proc. ICASSP*, 2019, pp. 5916–5920.

[106] J. Yamagishi, C. Veaux, and K. MacDonald, "CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit," 2019. [Online]. Available: https://doi.org/10.7488/ds/2645

[107] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. ICLR*, 2015, 15 pages.

[108] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the Variance of the Adaptive Learning Rate and Beyond," in *Proc. ICLR*, 2020, 13 pages.

[109] B. McFee, A. Metsai, M. McVicar, S. Balke, C. Thomé, C. Raffel, F. Zalkow, A. Malek, Dana, K. Lee, O. Nieto, D. Ellis, J. Mason, E. Battenberg, S. Seyfarth, R. Yamamoto, viktorandreevichmorozov, K. Choi, J. Moore, R. Bittner, S. Hidaka, Z. Wei, nullmightybofo, A. Weiss, D. Hereñú, F.-R. Stöter, L. Nickel, P. Friesch, M. Vollrath, and T. Kim, "librosa/librosa: 0.9.2," 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6759664

[110] R. Yoneyama, Y.-C. Wu, and T. Toda, "Unified Source-Filter GAN: Unified Source-Filter Network Based On Factorization of Quasi-Periodic Parallel WaveGAN," in *Proc. Interspeech*, 2021, pp. 2187–2191.

[111] R. Yoneyama, Y.-C. Wu, and T. Toda, "High-Fidelity and Pitch-Controllable Neural Vocoder Based on Unified Source-Filter Networks," *IEEE/ACM Trans. ASLP*, vol. 31, pp. 3717–3729, 2023.

[112] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier Nonlinearities Improve Neural Network Acoustic Models," in *Proc. ICML*, vol. 30, no. 1, 2013, p. 3.

[113] T. Che, Y. Li, A. Jacob, Y. Bengio, and W. Li, "Mode Regularized Generative Adversarial Networks," in *Proc. ICLR*, 2017, 13 pages.

[114] Canon, "[NamineRitsu] Blue (YOASOBI) [ENUNU model Ver.2, Singing DBVer.2 release]," https://www.youtube.com/watch?v=pKeo9IE_L1I, accessed: 2022.10.06.

[115] K. Yu and S. Young, "Continuous F0 modeling for HMM based statistical parametric speech synthesis," *IEEE/ACM Trans. ASLP*, vol. 19, no. 5, pp. 1071–1079, 2010.

[116] R. Yamamoto, R. Yoneyama, and T. Toda, "NNSVS: A Neural Network-Based Singing Voice Synthesis Toolkit," in *Proc. ICASSP*, 2023, pp. 1–5.

[117] R. Yoneyama, A. Miyashita, R. Yamamoto, and T. Toda, "Wavehax: Aliasing-Free Neural Waveform Synthesis Based on 2D Convolution and Harmonic Prior for Reliable Complex Spectrogram Estimation," *IEEE Trans. ASLP*, vol. 33, pp. 4454–4470, 2025.

[118] K. Oura, K. Nakamura, K. Hashimoto, Y. Nankaku, and K. Tokuda, "Deep neural network based real-time speech vocoder with periodic and aperiodic inputs," in *Proc. SSW*, 2019, pp. 13–18.

[119] Y. Hono, K. Hashimoto, Y. Nankaku, and K. Tokuda, "PeriodGrad: Towards Pitch-Controllable Neural Vocoder Based on a Diffusion Probabilistic Model," in *Proc. ICASSP*, 2024, pp. 12 782–12 786.

[120] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," *arXiv:1607.06450*, 2016.

[121] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," *arXiv:1606.08415*, 2023.

[122] S. Takamichi, K. Mitsui, Y. Saito, T. Koriyama, N. Tanji, and H. Saruwatari, "JVS corpus: free Japanese multi-speaker voice corpus," *arXiv:1908.06248*, 2019.

[123] T. Hayashi, K. Kobayashi, and T. Toda, "An Investigation of Streaming Non-Autoregressive sequence-to-sequence Voice Conversion," in *Proc. ICASSP*, 2022, pp. 6802–6806.

[124] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," in *Proc. ICLR*, 2019, 18 pages.

[125] Canon, "[NamineRitsu] Blue (YOASOBI) [ENUNU model Ver.2, Singing DBVer.2 release]," https://www.youtube.com/watch?v=pKeo9IE_L1I, accessed: 2022.10.06.

[126] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "SoundStream: An End-to-End Neural Audio Codec," *IEEE/ACM Trans. ASLP*, vol. 30, pp. 495–507, 2021.

[127] H.-S. Choi, J. Yang, J. Lee, and H. Kim, "NANSY++: Unified Voice Synthesis with Neural Analysis and Synthesis," in *Proc. ICLR*, 2023, 24 pages.

[128] Z. Ju, Y. Wang, K. Shen, X. Tan, D. Xin, D. Yang, Y. Liu, Y. Leng, K. Song, S. Tang, Z. Wu, T. Qin, X.-Y. Li, W. Ye, S. Zhang, J. Bian, L. He, J. Li, and S. Zhao, "NaturalSpeech 3: zero-shot speech synthesis with factorized codec and diffusion models," in *Proc. ICML*, 2024, 19 pages.

List of Publications

Journal Papers

1. R. Yoneyama, Y.-C. Wu, T. Toda, "High-Fidelity and Pitch-Controllable Neural Vocoder Based on Unified Source-Filter Networks," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 31, pp. 3717-3729, Sep. 2023.

2. R. Yoneyama, R. Yamamoto, A. Miyashita, T. Toda, "Wavehax: Aliasing-Free Neural Waveform Synthesis Based on 2D Convolution and Harmonic Prior for Reliable Complex Spectrogram Estimation," IEEE Transactions on Audio, Speech, and Language Processing, vol. 33, pp. 4454-4470, Oct. 2025.

3. R. Yoneyama, T. Toda, "SiFi-GAN: Combining Source-Filter Modeling and Upsampling-Based High-Fidelity Neural Vocoder for Fast and Pitch-Controllable Speech Synthesis," IEICE Transactions on Information, vol. E109-D, no. 6, 12 pages, Jan. 2026. (Accepted)

International Conferences

1. R. Yoneyama, Y.-C. Wu, T. Toda, "Unified Source-Filter GAN: Unified Source-Filter Network Based on Factorization of Quasi-Periodic Parallel WaveGAN,"

Proc. Interspeech, pp. 2187-2191, 2021.

2. R. Yoneyama, Y.-C. Wu, T. Toda, "Unified Source-Filter GAN with Harmonic-PlusNoise Source Excitation Generation," Proc. Interspeech, pp. 848-852, 2022.

3. R. Yoneyama, Y.-C. Wu, T. Toda, "Source-Filter HiFi-GAN: Fast and Pitch-Controllable High-Fidelity Neural Vocoder," Proc. IEEE ICASSP, 5 pages, 2023.

4. R. Yoneyama, R. Yamamoto, K. Tachibana, "Nonparallel High-Quality Audio Super Resolution with Domain Adaptation and Resampling CycleGANs," Proc. IEEE ICASSP, 5 pages, 2023.

5. R. Yoneyama, M. Kawamura, R. Terashima, R. Yamamoto, T. Toda, "Comparative Analysis of Fast and High-Fidelity Neural Vocoders for Low-Latency Streaming Synthesis in Resource-Constrained Environments," Proc. INTERSPEECH, 2025.

6. R. Yamamoto, R. Yoneyama, T. Toda, "NNSVS: A Neural Network-Based Singing Voice Synthesis Toolkit," Proc. IEEE ICASSP, 5 pages, 2023.

7. R. Yamamoto, R. Yoneyama, L. P. Violeta, W.-C. Huang, T. Toda, "A Comparative Study of Voice Conversion Models with Large-Scale Speech and Singing Data: The T13 Systems for the Singing Voice Conversion Challenge 2023," Proc. ASRU, 6 pages, 2023.

8. K. Ogita, R. Yoneyama, W.-C. Huang, T. Toda, "VAE-SiFiGAN: Source-Filter HiFi-GAN Based on Variational Autoencoder Representations with Enhanced Pitch Controllability," Proc. EUSIPCO, 2025.

Domestic Conferences

1. R. Yoneyama, Y.-C. Wu, T. Toda, "A unified source-filter network for neural vocoder," IEICE Tech. Rep., vol. 120, no. 399, SP2020-34, pp. 57-62, 2021. (in Japanese)

2. R. Yoneyama, Y.-C. Wu, T. Toda, Y. Tsao, H.-M. Wang. "Unified Source-Filter Network with Adversarial Learning," The Acoustical Society of Japan, 2-3-2, pp. 905-906, Autumn 2021. (in Japanese)

3. R. Yoneyama, Y.-C. Wu, T. Toda, "Improvement of Unified Source-Filter Network with Adversarial Learning," The Acoustical Society of Japan, 1-3-10, pp. 907-908, Spring 2022. (in Japanese)

4. R. Yoneyama, Y.-C. Wu, T. Toda, "Source-Filter-Architecture-Based HiFi-GAN," The Acoustical Society of Japan, 2-3-5, pp. 721-722, Spring 2023. (in Japanese)

5. R. Yoneyama, R. Yamamoto, A. Miyashita, T. Toda, "Aliasing-Free Neural Vocoder Based on Complex Spectrogram Estimation with Harmonic Signal Modeling and 2D Convolution," The Acoustical Society of Japan, 1-2-9, Spring 2025. (in Japanese)

6. R. Yamamoto, R. Yoneyama, T. Toda, "NNSVS: A Neural Network-Based Singing Voice Synthesis Toolkit," The Acoustical Society of Japan, 1-9-19, pp. 1057-1060, Autumn 2023. (in Japanese)

7. K. Ogita, R. Yoneyama, W.-C. Huang, T. Toda, "VAE-SiFi-GAN: SiFi-GAN Based on Variational Autoencoder Representations," The Acoustical Society of Japan, 1-R-30, Spring 2025. (in Japanese)

Awards

1. Student Outstanding Presentation Award, the 23rd Autumn Meeting of the Acoustical Society of Japan, Sep. 2021.

2. The Award for Excellence in Q&A at the Midterm Master's Thesis Presentation of the Tokai-area Speech-related Laboratories, Oct. 2022.

3. 17th Student Conference Paper Award, IEEE SPS Japan, Sep. 2023.

Invited Talk

1. R. Yoneyama, "An Overview of Neural Vocoders: From the Perspective of Generative Models and Practicality," the 143rd MUS & 156th SLP Joint Workshop, Tokyo, June 2025.

9 Appendix

9.1 Investigation of Input Acoustic Features

To further investigate the impact of different conditional acoustic features, we evaluated several models with different types of conditioning features with the same model architecture. In our proposed methods used in the experimental evaluations (Section 5.3), we chose the set of MGC, MAP as the best combination for the auxiliary features whose total number of dimensions is 62. Here, we compare P-uSFGAN with the following three models with different auxiliary features.

- MEL: This model adopts a full-band 80-dimensional log mel-spectrogram calculated in the setting described in Section 5.3.2 instead of the vocoder features.
- AuxF_0: This model includes one-dimensional continuous F_0 in the default set of the auxiliary feature. The total number of dimensions of the auxiliary feature is 63.
- BAP: This model adopts the three-dimensional band-aperiodicity extracted using WORLD instead of the 21-dimensional MAP. Coding is performed by one-dimensional interpolation on the frequency axis, which compresses the half-FFT size to three. The total number of dimensions of the auxiliary feature is 44.

All the ablation models were trained in the same setting as that in the P-uSFGAN model, except for their auxiliary features. Note that all subnetworks (i.e., harmonic

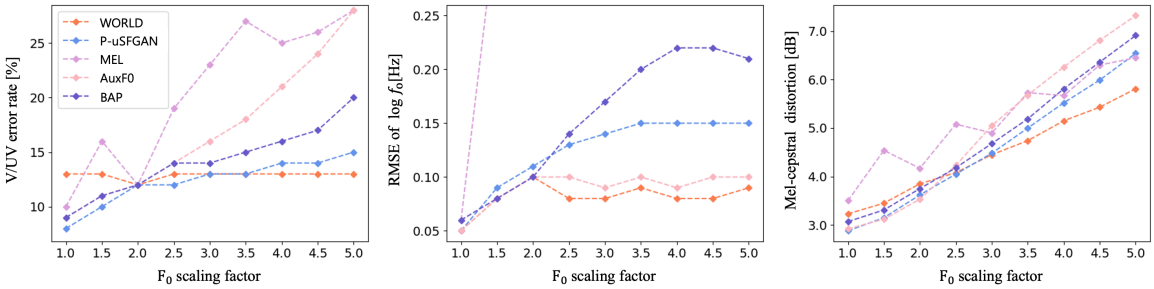

Figure 9.1: *Objective evaluation results of F_0 transformation for the ablation study on auxiliary features.*

network, noise network, filter network, and periodicity estimator) are conditioned using the same auxiliary features.

The objective evaluation results are shown in Fig. 9.1. The WORLD results are provided as references. We found that differences between the models become apparent when F_0 is significantly high, so the study was conducted with F_0 increased by a factor of five. First, we can see that the MEL model degrades even with a small F_0 change. Since the mel-spectrogram already contains the F_0 information, we speculate that it is difficult for the model to manipulate F_0 by merely changing the sinusoidal inputs. The AuxF_0 model shows significant degradation with F_0 increased by a factor of two or more in its VUV error rate, which is more critical for sound quality than the RMSE of log F_0. We confirmed that the generated speech is hardly voiced, resulting in significant degradation. We assume that this tendency is because the inductive bias for speech production provided by the source-filter modeling is not obtained owing to the leakage of F_0 information to the filter network. The total degradation in the MEL model can be considered to have the same cause. From these experiences, we concluded that disentanglement of the input acoustic features and the restriction of F_0 information leakage to the filter network is essential to gaining the benefit from source-filter modeling. The BAP model, which gives periodicity information with fewer

dimensions, shows minimal degradation in both VUV error rate and RMSE of log F_0. We assume that the degradation is because the neural network can ignore a lower-dimensional input feature (i.e., BAP) when the network can reconstruct the target waveform from the other input features in training. Moreover, this result suggests the importance of information about periodicity in neural vocoders based on periodic and aperiodic component decomposition.

9.2 Harmonic Decomposition of Sine Powers

We prove that for any non-negative integer n, $\sin^n(\theta)$ can be expressed as a linear combination of sine and cosine terms, involving harmonics up to and including the n-th harmonic. Specifically, we prove the following proposition by mathematical induction:

$$\sin^n(\theta) = \sum_{m=0}^{n} \big\{ a_m \sin(m\theta) + b_m \cos(m\theta) \big\} \tag{9.1}$$

where a_m and b_m are coefficients dependent on n.

1) For $n = 0$, the proposition holds because

$$\sin^0(\theta) = 1 = \cos(0\,\theta). \tag{9.2}$$

2) Assuming Eq. (9.1) holds for $n = k$ ($k \geq 0$), we prove the case for $n = k + 1$ as

follows.

$$\begin{aligned}
\sin^{k+1}(\theta) &= \sin(\theta)\sum_{m=0}^{k}\big\{a_m\sin(m\theta)+b_m\cos(m\theta)\big\} \\
&= \sum_{m=0}^{k}\big\{a_m\sin(m\theta)\sin(\theta)+b_m\cos(m\theta)\sin(\theta)\big\} \\
&= \sum_{m=0}^{k}\Big\{\frac{a_m}{2}\big(\cos((m-1)\theta)-\cos((m+1)\theta)\big) \\
&\quad+\frac{b_m}{2}\big(\sin((m+1)\theta)-\sin((m-1)\theta)\big)\Big\} \\
&= \sum_{m=0}^{k+1}\big\{a'_m\sin(m\theta)+b'_m\cos(m\theta),\big\}
\end{aligned} \tag{9.3}$$

where a'_m and b'_m are the updated coefficients derived from a_m and b_m. The transformation between the second and third lines is achieved using the following product-to-sum identities:

$$\sin(m\theta)\sin(\theta) = \frac{\cos((m-1)\theta)-\cos((m+1)\theta)}{2} \tag{9.4}$$

$$\cos(m\theta)\sin(\theta) = \frac{\sin((m+1)\theta)-\sin((m-1)\theta)}{2}. \tag{9.5}$$

Consequently, from steps 1) and 2), the proposition of Eq. (9.1) has been proved.

9.3 Aliasing-Free Harmonic Signal Generation

Pulse trains are widely used as excitation signals in vocoders [2,3,30], and are a natural candidate for the harmonic prior. However, pulse trains, constructed via pointwise sampling from continuous-time signals, inherently involve aliasing. To avoid aliasing,

we adopt a superposition of band-limited sinusoidal signals as [30]. We extend this approach by modulating each harmonic's amplitude to maintain constant signal power independent of F_0 as [2, 3]. This helps the network focus on relevant features without being affected by power variations, enhancing its generalizability to unseen F_0. The harmonic signal is defined as follows:

$$\boldsymbol{h}[n] = \sum_{k=1}^{K_n} g_k[n] \sin(2\pi k\phi[n] + \varphi_k) \tag{9.6}$$

$$= \sum_{k=-K_n}^{K_n} \frac{g_k[n]}{2} \exp(j\varphi_k) \exp(j2\pi k\phi[n]), \tag{9.7}$$

where $g_k[n]$ is the amplitude of the k-th harmonic component at time step n, and φ_k is the initial phase of the k-th harmonic component. $f[n]$ and $\phi[n]$ are ones defined in Section 7.3.1. The number of harmonic components K_n is determined by dividing the maximum frequency $F_{\max}$ by $f[n]$, i.e., $K_n = \lfloor F_{\max}/f[n] \rfloor$. $F_{\max}$ sets the upper harmonic frequency limit, balancing frequency coverage and computational cost. We set $F_{\max}$ to the Nyquist frequency F_{N}. Signal power can be controlled by modulating $g_k[n]$ based on Parseval's identity:

$$\frac{1}{L} \sum_{k=-K_n}^{K_n} \left(\frac{g_k[n]}{2} \right)^2 = \sum_{m=n-L/2}^{n+L/2} \boldsymbol{e}[m]^2, \tag{9.8}$$

where L is the window length. Since we assume constant power C^2 for each time frame, the identity becomes:

$$\frac{1}{L^2} \sum_{k=-K_n}^{K_n} \left(\frac{g_k[n]}{2} \right)^2 = \frac{1}{L} \sum_{m=n-L/2}^{n+L/2} \boldsymbol{e}[m]^2 = C^2, \tag{9.9}$$

While alternative spectral envelopes, such as linear decay, can be considered, this paper assumes a flat spectral envelope across each time frame (i.e., $g_k[n] = g_{k+1}[n]$ for all $k \leq K_n - 1$), yielding:

$$\sum_{k=-K_n}^{K_n} \left(\frac{g_k[n]}{2}\right)^2 = 2K_n \left(\frac{g_k[n]}{2}\right)^2 = L^2C^2. \tag{9.10}$$

$$g_k[n] = \begin{cases} LC\sqrt{2/K_n} & \text{if } f[n] \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{9.11}$$

where LC is set to 0.1 in this study. Additionally, we observed better performance when using linear phases for φ_k compared to zero or random phases. Linear initial phases are defined as:

$$\varphi_k = k\varphi, \tag{9.12}$$

where $\varphi \sim \mathcal{U}(-\pi, \pi)$. The harmonic signal $\boldsymbol{h}$ is derived by substituting Eq. (9.11) and Eq. (9.12) into Eq. (9.6). Finally, the sum of $\boldsymbol{e}$ and small Gaussian noise $\boldsymbol{n} \sim \mathcal{N}(\mathbf{0}, \sigma I)$, where $\sigma = 0.01$, is fed into the neural vocoders.