A Fast High-Fidelity Source-Filter Vocoder With Lightweight Neural Modules

Runxuan Yang, Yuyang Peng, and Xiaolin Hu, *Senior Member, IEEE*

***Abstract*—The quality of raw audio waveform generated by a vocoder could affect various audio generative tasks. In recent years, the dominance of source-filter vocoders was greatly challenged by neural vocoders as the latter presents far superior synthesized audio quality. Meanwhile, neural vocoders introduced unprecedented limitations including low runtime efficiency as well as unstable pitch especially in those without explicit periodic excitation input, while these have never been a problem in source-filter vocoders. We present in this article a novel approach that takes the best from both parties. We start by an in-depth examination of every building block in WORLD – one of the best-performing source-filter vocoders based on plain signal processing algorithms, looking for ones that do not work well, and we replace them with small, lightweight and task-specific neural network models. We also rearranged the vocoding pipeline for a smoother collaboration between building blocks. Our objective and subjective evaluations demonstrate that our methods present competitive synthesized audio quality even when compared against neural vocoders at a much lower computational cost, while keeping spectral envelope acoustic feature, high pitch accuracy as in conventional source-filter vocoders.**

***Index Terms*—Neural network, singing voice synthesis, spectral envelope, vocoder.**

I. Introduction

VOCODER is an important building block of many audio-related generative tasks including singing voice synthesis (SVS), text-to-speech (TTS), voice conversion, etc. Taking an SVS or TTS pipeline for example – training an end-to-end mapping from phoneme to audio waveform directly is known to be extremely computationally inefficient as shown in WaveNet [1]. Modern SVS and TTS pipelines often split their generation process into two parts: acoustic model and vocoder, each having its own significance. An acoustic model is in charge of understanding symbolic features extracted from human-readable text and conversion of these features to a time-continuous acoustic representation. A vocoder is directly responsible for the final output audio quality – it not only needs to synthesize audio waveform based on acoustic features, but also needs to be able to extract acoustic features from ground-truth audio recordings for training the acoustic model. Acoustic features act as a bridge connecting these two modules, capturing high-level acoustic information (usually in spectral domain) from plain waveforms, while eliminating as much redundant information as possible. The designed acoustic features should be clear and concise while preserving the integrity of acoustic information.

A vocoder may find its application into other tasks as well. Some of the well-known ones include pitch transposition and time scaling. These tasks often require manual modifications over extracted acoustic features, unlike previously mentioned generative pipelines where acoustic features are kept untouched and used as ground-truth data for training the preceding acoustic model.

All these tasks suggest that a complete vocoder needs to have two distinct functionalities – extraction of acoustic features from an audio waveform, named *analysis stage*, and generation of audio waveform from acoustic features, named *synthesis stage*. There are nowadays two main technical approaches for vocoder development – source-filter vocoder and neural vocoder, based on quite different concepts. Diagram illustrated in Fig. 1 gives a rough overview of the two pipelines.

Conventional source-filter vocoders such as STRAIGHT [2], [3] and WORLD [4] make heavy use of digital signal processing (DSP) knowledge. The overall idea is, during analysis stage, it first extracts a fundamental frequency curve [5], [6], and then it splits harmonic and breathiness components apart, generating a feature representation for each of them. During synthesis stage, it performs spectral filtering over two excitation sources separately for harmonic and breathiness, and then the two parts are added together for the final output. This means that much of the complication is done at analysis stage, making synthesis stage fast and lightweight. Moreover, periodic excitation source is generated using an oscillator that works in strict accordance with the fundamental frequency curve. This ensures pitch accuracy while still allowing easy pitch manipulation such as transposition. On the other hand, the downside of existing source-filter approaches

Manuscript received 23 November 2022; revised 15 June 2023 and 22 August 2023; accepted 14 September 2023. Date of publication 4 October 2023; date of current version 20 October 2023. This work was supported by the National Natural Science Foundation of China under Grants 62061136001 and 61836014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Hema A Murthy. *(Corresponding author: Xiaolin Hu.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Department of Psychology Ethics Committee, Tsinghua University under Application No. 2021-15.

Runxuan Yang is with the Department of Computer Science and Technology, State Key Laboratory of Intelligent Technology and Systems, Institute for AI, THBI, BNRist, Tsinghua University, Beijing 100084, China (e-mail: yangrx20@mails.tsinghua.edu.cn).

Yuyang Peng is with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: pengyc23@mails.tsinghua.edu.cn).

Xiaolin Hu is with the Department of Computer Science and Technology, State Key Laboratory of Intelligent Technology and Systems, Institute for AI, THBI, BNRist, Tsinghua University, Beijing 100084, China, and also with Chinese Institute for Brain Research (CIBR), Beijing 102206, China (e-mail: xlhu@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TASLP.2023.3321191

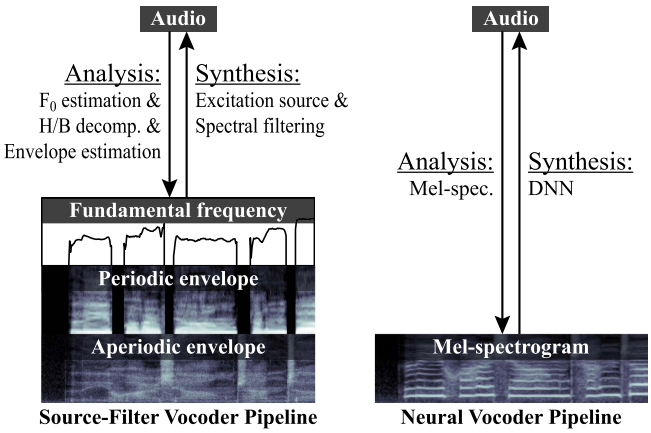


Fig. 1. Diagram for analysis and synthesis pipeline for source-filter vocoder and neural vocoder.

is inherently obvious – recovery of lost phase information is far from perfect, causing flaws in synthesized audio quality.

Neural vocoder arises along with the era of deep learning, challenging source-filter vocoders through a very different technical path. The analysis stage is relatively straightforward as it is simply log power spectrogram under Mel frequency scale, commonly used in many existing neural vocoders [1], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21]. The essence lies in the synthesis part – a full neural network that directly outputs audio waveform based on the input Mel-spectrogram. As the training for audio synthesis is done in an end-to-end manner, we no longer need those DSP complications such as phase recovery in order to achieve an outstanding synthesized audio quality. However, neural vocoders have some major drawbacks. One of them is reduced runtime efficiency especially when running large neural network models. Also, neural vocoders without explicit periodic excitation as input often introduce pitch distortions in synthesized audio due to its data-driven nature, especially when encountering unseen data. This is especially problematic during singing voice synthesis where pitch accuracy is an indispensable requirement.

These two approaches are not mutually exclusive. We believe that, an ideal vocoder should take the best from both parties – one that can achieve synthesized audio quality as in neural vocoders, while keeping pitch accuracy and runtime performance as in a typical source-filter vocoder. Pitch accuracy is especially important when processing singing voice. There have been several recent efforts [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32] that incorporates one idea into another, producing impressive outcomes.

In our work, we perform an in-depth analysis over several important modules of WORLD – an existing state-of-the-art plain-DSP vocoder, keeping those that worked well, and replacing others with mostly lightweight, task-specific, deep learning-based approaches. We name our vocoder system "VogenVoc", as part of a larger SVS ecosystem codenamed Vogen. Our system features a neural network model for smart separation between periodic and aperiodic components instead of the conventional method based on orthogonal phase decomposition, a neural network model for fast and realistic aperiodic excitation generation instead of using plain white noise signal, as well as many other minor adjustments to the overall vocoding pipeline so as to combine source-filter and neural approaches, keeping advantages from both parties. All these modules contribute to reaching our goal as a whole.

Subjective and objective evaluations showed the effectiveness of our method in several aspects:

- Fast runtime speed comparable to a plain-DSP vocoder;
- Competitive synthesized audio fidelity when compared to neural vocoders;
- High pitch accuracy suitable for synthesizing singing voices as from a typical source-filter pipeline.

II. Related Work

A. Plain-DSP Approaches

Attempts at obtaining isolated representation of pitch and pitch-independent spectral features date back to early ages of vocoder research [33]. Various spectral envelope estimation algorithms have been proposed throughout the years, including Cepstrum-based ones [34], [35] and Linear Predictive Coding (LPC)-based ones [36], [37]. With the advancement of hardware resources, more sophisticated vocoders such as STRAIGHT [2], [3] and WORLD [4] have emerged, featuring different ways to handle harmonic and breathiness components. Modern plain-DSP vocoders typically extract the following three types of feature from an audio recording, as illustrated in Fig. 1:

- Fundamental frequency F_0 representing the pitch curve of audio recording, or 0 if unpitched;
- Periodic spectral envelope for the harmonic component of audio recording independent of F_0;
- Aperiodic spectral envelope for the breathiness component of audio recording.

Periodic and aperiodic envelopes often exhibit many similarities in envelope shape as they represent different acoustic parts of the same pronunciation and timbre. Thus, some works [38], [39] choose to use an envelope at full spectral resolution to represent the sum of both components, and another envelope at lower resolution for their ratio, instead of having two separate envelopes both at full resolution. Moreover, we can use the idea of Mel-spectrogram to compress spectral envelopes by reducing resolution at higher frequencies [40]. During synthesis phase, we start with a periodic excitation signal generated with an oscillator according to F_0, then filtered it using periodic envelope, and finally we obtain waveform of the periodic component using inverse STFT. Aperiodic component works similarly, except that we use a noise generator instead of an oscillator. The two components add up to the final output waveform.

B. Neural Vocoders With No Explicit Periodic Excitation

Neural vocoders without explicit periodic excitation modelling typically use Mel-spectrogram as acoustic feature. In the terminology of deep learning, converting raw waveform to Mel-spectrogram can be regarded as a form of dimensionality reduction, while the inverse process is a generative task that

restores information. Various generative methods have been explored deeply for spectrogram inversion in previous research. Early approaches like WaveNet [1], SampleRNN [7] and FFTNet [8] relied on autoregressive waveform generation. Other approaches include WaveRNN [9] based on recurrent neural network (RNN), WaveGlow [10] based on normalizing flow, as well as Parallel WaveNet [11] and ClariNet [12] based on inverse autoregressive flows (IAF) and knowledge distillation as some of the first attempts to avoid autoregressive generation. Later, Generative Adversarial Networks (GAN) became popular and inspired many works such as Parallel WaveGAN [13], MelGAN [14], HiFi-GAN [15], UnivNet [16], iSTFTNet [17], and so on. Moreover, with the development of denoising diffusion probabilistic models, works like WaveGrad [18], DiffWave [19], SpecGrad [20], HPG [21], and others have emerged. Although these approaches have achieved outstanding audio quality compared to conventional source-filter methods based on plain DSP, they suffer from reduced runtime efficiency especially when running large neural network models. Also, due to the fact that all harmonics are represented as bitmap inside Mel-spectrogram, the neural network would need to infer the correct fundamental frequency for oscillation from pixel combinations, meaning a slight change in pixel brightness would largely affect pitch accuracy of the output waveform.

C. Hybrid Approaches

To address the pitch accuracy and runtime efficiency issues in previously mentioned neural vocoders, researchers have explored hybrid approaches that combine the advantages of both methods. QP-Net [22] and QP-PWG [23] introduced pitch-dependent dilated convolution in accordance with the input F_0 curve. LPCNet [24], GELP [25] and GlotGAN [26] investigated various neural methods on periodic and aperiodic (noise) excitation generation. Neural Homomorphic Vocoder (NHV) [27] proposed trainable linear time-varying filters with adversarial loss. Neural Source-filter (NSF) and its variants [28], [29], [30] examined a variety of excitation signals with neural filtering. Unified Source-filter GAN (uSFGAN) [31] proposed separate networks for excitation generation and resonance filtering. Source-filter HiFi-GAN (SiFi-GAN) [32] presented a modified HiFi-GAN network that takes in an excitation source as an extra input. Moreover, DDSP [41] proposed a fully end-to-end differentiable source-filter model for deep learning methods.

Though many existing studies have focused on improving the synthesis part of a vocoder, the analysis part has received less attention. Our proposed methods achieve an improvement through efforts on both parts.

III. Method

In this section, we will first go through the synthesis pipeline in detail, with explanation on our choice of the three acoustic features, and the way how they interact with the overall vocoding pipeline. We will also build a small, GAN-based neural network for aperiodic excitation generation, substituting the conventional white noise signal.

After that, we will go through the analysis pipeline, namely the method for extracting these acoustic features from input audio. We will start by introducing some of the existing limitations we found in WORLD vocoder [4], as well as our proposed methods for tackling them. We will also build a neural network specifically for the task of decomposing harmonic and breathiness components from an input audio.

A. Overall Synthesis Pipeline

In source-filter vocoder, an audio recording waveform x is considered a sum of its harmonic component x_h and breathiness component x_n, each representing the part of x dependent of F_0 and the part independent of it, respectively. The periodic component is obtained by filtering on periodic excitation signal β_h using periodic spectral envelope p_h (aperiodic component works analogously with β_n and p_n):

$$\begin{aligned} x &= x_h + x_n \\ &= \mathrm{conv1d}(p_h, \beta_h) + \mathrm{conv1d}(p_n, \beta_n). \end{aligned} \tag{1}$$

Because convolutional operations are often computationally expensive, signal filtering is commonly carried out in spectral domain using elementwise multiplication (denoted with $\odot$):

$$\begin{aligned} x = \mathcal{F}^{-1}\{X\} &= \mathcal{F}^{-1}\{X_h + X_n\} \\ &= \mathcal{F}^{-1}\{P_h \odot B_h + P_n \odot B_n\}. \end{aligned} \tag{2}$$

Periodic excitation B_h is constructed according to F_0 with an oscillator. F_0, P_h, P_n constitute the acoustic features required by a source-filter vocoder.

An example of our synthesis pipeline is shown in Fig. 2. The same concept is used throughout many source-filter approaches, though many details are different. Specifically, our pipeline is different from WORLD in the following aspects:

- For the aperiodic part – WORLD uses white noise for aperiodic excitation signal β_n. However, we found that white noise exhibits different phase characteristics than human breathiness in singing. We propose to use a generative adversarial network (GAN) to construct this part of the signal. Method details will be described in Section III-B.
- For the periodic part – WORLD calculates every single impulse response separately when synthesizing periodic component x_h, involving more calculation steps such as cepstral filtering and sub-sample time-shift than directly performing spectral domain multiplication over periodic excitation as $x_h = \mathcal{F}^{-1}\{P_h \odot B_h\}$. We choose the latter for a slight performance boost.

B. Synthesis of Aperiodic Excitation

Existing DSP approaches for generating breathiness typically employs white noise signal as aperiodic excitation. However, the phase distribution of white noise is very different from that of real human aspiration, causing breathiness generated with the former to always provide a fake, "plastic" kind of hearing experience. Phase variations are notoriously nasty to handle manually. To the best of our knowledge, no existing work is able to reproduce perfect human breathiness with plain-DSP method.

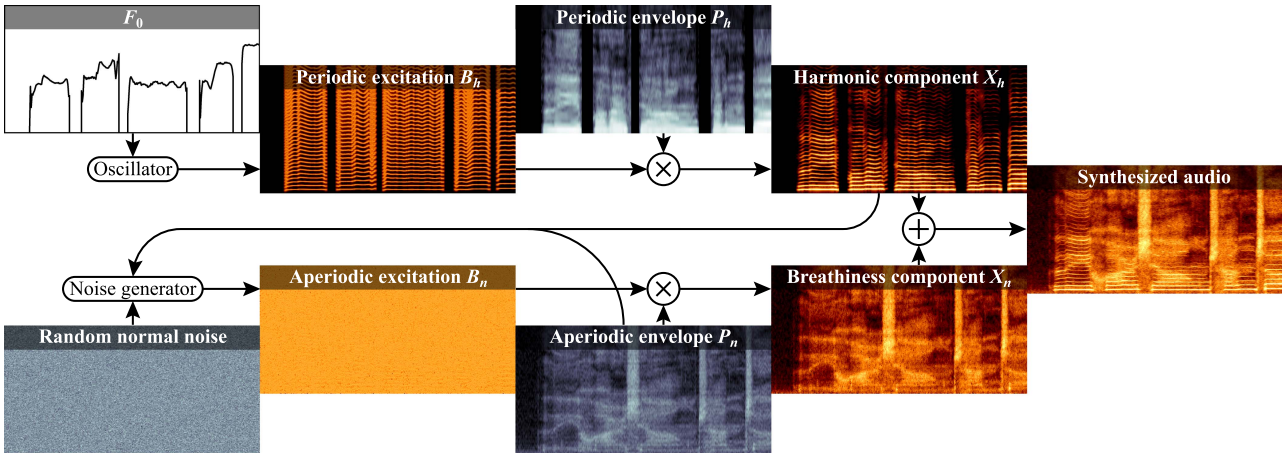


Fig. 2. Source-filter synthesis pipeline used in this work with examples. Images in red-orange are complex-valued, though this figure only displays magnitude spectrograms. Images in blue-gray are real-valued.

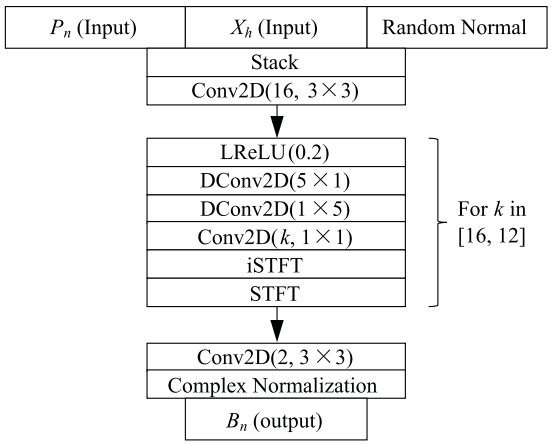


Fig. 3. Neural network architecture for noise (aperiodic excitation) generator model.

On the other hand, neural vocoders perform a much better job at reconstructing human breathiness than DSP ones. Moreover, an end-to-end neural vocoder does much more than just breathiness reconstruction including harmonic oscillation, spectral envelope filtering, etc. – many of which could have been done manually using DSP methods. We can then construct a much simplified version of neural vocoder that focuses on breathiness reconstruction only, keeping the entire generation pipeline fast and lightweight.

Our aperiodic excitation generator model is designed to be small, fast and lightweight. It takes three inputs – generated harmonic component spectrogram $\hat{X}_h$, aperiodic spectral envelope P_n and Gaussian normal noise with the same size as the other inputs. A diagram of the model architecture is shown in Fig. 3. Note that 2-dimensional convolution is often computationally expensive as it involves 3-D matrix multiplication. To address this problem, we split our 2-D convolution in each iteration block into three different convolutional operations, each tackling only one dimension at a time, namely depthwise separable convolutions [42]. We also appended an inverse STFT operation immediately followed by a forward STFT operation to the end of each iteration block, in order to help spectral convergence as inspired by the Griffin-Lim algorithm [43]. Finally, the model outputs aperiodic excitation $\hat{B}_n$, for the use of upcoming steps in the whole synthesis pipeline.

We base our training settings on UnivNet [16], one of the best-performing GAN-based neural vocoders. We replace its generator with our own breathiness generation model, leaving everything else as-is. Note that we do not calculate loss on $\hat{B}_n$ directly; instead, we do it on $\hat{X}_n = \hat{B}_n \odot P_n$, namely after we perform spectral filtering with envelope P_n, as the original losses in UnivNet are designed to work on fully-synthesized audio.

C. Overall Analysis Pipeline

The analysis pipeline aims to extract acoustic features from a piece of given audio recording x. In WORLD, we first need to extract fundamental frequency curve F_0 from input audio x using Harvest [6] or DIO [5]. We then estimate spectral envelope P_x with CheapTrick [38], and the power ratio between periodic and aperiodic components α with D4C [39]. We can also obtain periodic envelope $P_h = \alpha^2 P_x$ and aperiodic envelope $P_n = (1 - \alpha^2)P_x$ separately. A diagram of the described pipeline is shown in the upper part of Fig. 4.

While WORLD already provides an excellent source-filter vocoding solution incorporating deep theoretical foundations and many practical considerations, we still observe some issues from its pipeline:

- D4C's use of group delay is not a sufficient condition for separation between periodic and aperiodic components. This is because with breathiness having much more phase randomness than harmonic, we simply cannot guarantee its first angular derivative is perfectly orthogonal to that of harmonics, causing overestimation of harmonic component in the output aperiodicity ratio. This is also reflected in enhanced harmonic overtones in high-frequency parts of synthesized audio, as shown in bottom-left of Fig. 9.
- The consequence of D4C overestimating harmonic components could be more pronounced at V/UV boundaries, especially when next to a sibilant consonant for example,

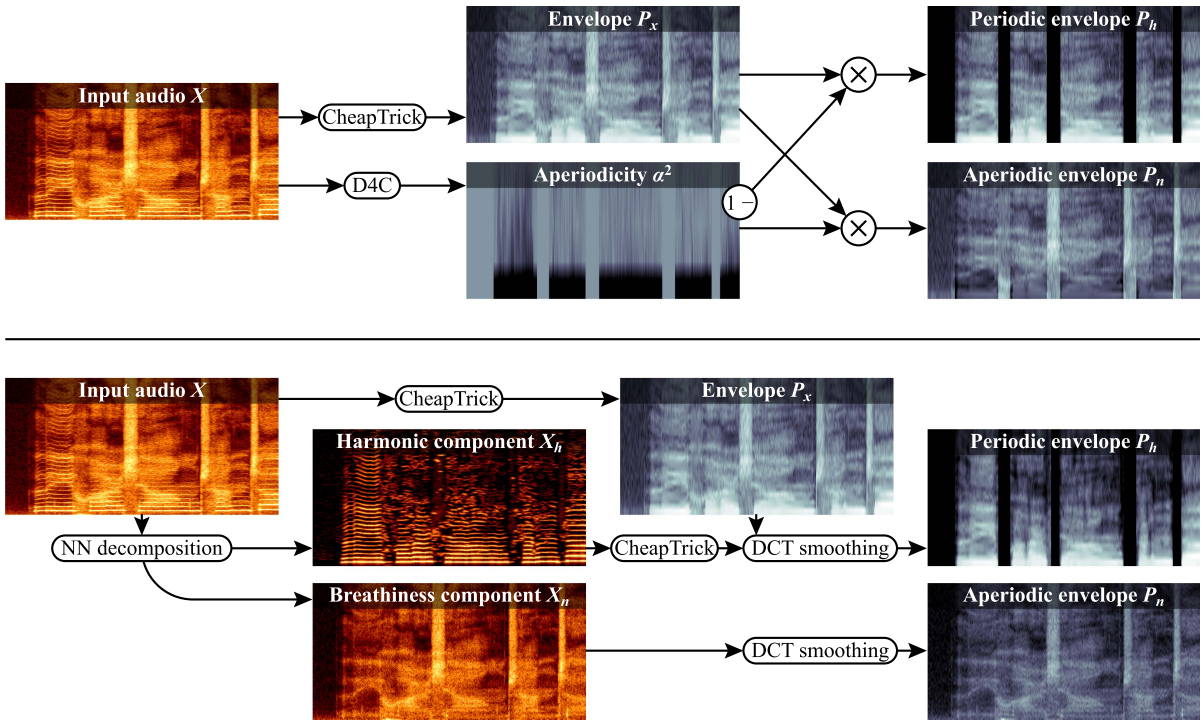


Fig. 4. Source-filter analysis pipeline used in WORLD [4] (upper) and in our work (lower) with examples.

where dense full-band sibilant airflow may become strident pulse train once resynthesized as harmonic components. In order to avoid this issue, D4C embeds another algorithm called LoveTrain that performs an additional stricter V/UV classification over the input audio. Unfortunately, V/UV boundaries are not perfectly time-aligned across all frequencies – voiced frames are often temporally more extended in low frequency than in high frequency. If ever a frame with low-frequency harmonics is wrongly classified as unvoiced, its envelope will be resynthesized as loud low-frequency noise, as shown in bottom-right of Fig. 9.

- The pitch-synchronized analysis method used in CheapTrick sets FFT window size to $3F_s/F_0$, with a fallback value $F_0 = 500$ for unvoiced frames, equivalent to a hard-coded 6-millisecond window size. This leads to quite some loss in estimated spectral envelope resolution.
- Another example of low resolution issue also exists in D4C. D4C sets band aperiodicity to be -60 dB at 0 Hz and 0 dB at $F_s/2$ Hz, and calculates band aperiodicity once only every 3000 Hz, greatly diminishing the distinction between periodic and aperiodic components.

Among these issues, much of the trouble comes from the separation task between harmonic and breathiness. Even though its theoretical foundations may look simple, we are still faced with lots of case-specific fine-tuning adaptation that would be labor-intensive to complete manually. Instead, we decide to train a lightweight and efficient neural network model for this separation task, and we can then estimate periodic envelope P_h and aperiodic envelope P_n separately. A flow chart of the modified envelope estimation pipeline is shown in the lower part of Fig. 4. Comparing to that of WORLD, the most different part is that envelope estimation now depends on separation results, and the estimation methods for the two components are different. Specifically:

- Harmonic/breathiness separation model is a neural network trained to directly output X_h and X_n given complex spectrogram X. More details are given in Section III-D.
- We use CheapTrick to calculate for periodic envelope P_h from X_h, though we only keep results from voiced frames. Due to scatters of small harmonic overtone fragments in separation results, we need to smooth out periodic envelope P_h according its proportion in full envelope P_x using DCT:

$$P_h = P_x / \exp\left(\mathrm{DCTSmoothen}_{s_h}\left(\log \frac{P_x}{P'_h}\right)\right), \tag{3}$$

$$P_x = \mathrm{CheapTrick}(x, F_s), \tag{4}$$

$$P'_h = \mathrm{CheapTrick}(x_h, F_s), \tag{5}$$

$$\mathrm{DCTSmoothen}_s(X) = \mathrm{DCT}_4\{\mathrm{DCT}_1\{X\}|_{1\ldots s}\}. \tag{6}$$

Here, the constant s_h stands for the number of dimensions to keep after DCT. A smaller s_h gives a smoother output envelope. We use $s_h = \lfloor F_s/3000 \rceil$.

- As for the part of estimating aperiodic envelope P_n from X_n, we simply smooth away random perturbations from the spectrogram using DCT:

$$P_h = \exp(\mathrm{DCTSmoothen}_{s_n}(\log \|X_n\|)). \tag{7}$$

Again, the constant s_n stands for the number of dimensions to keep after DCT. We use $s_n = \lfloor F_s/(2F_0) \rfloor$ so that s_n is no larger than the number of harmonic peaks in each spectral frame. For the case of unvoiced frames, any arbitrary s_n

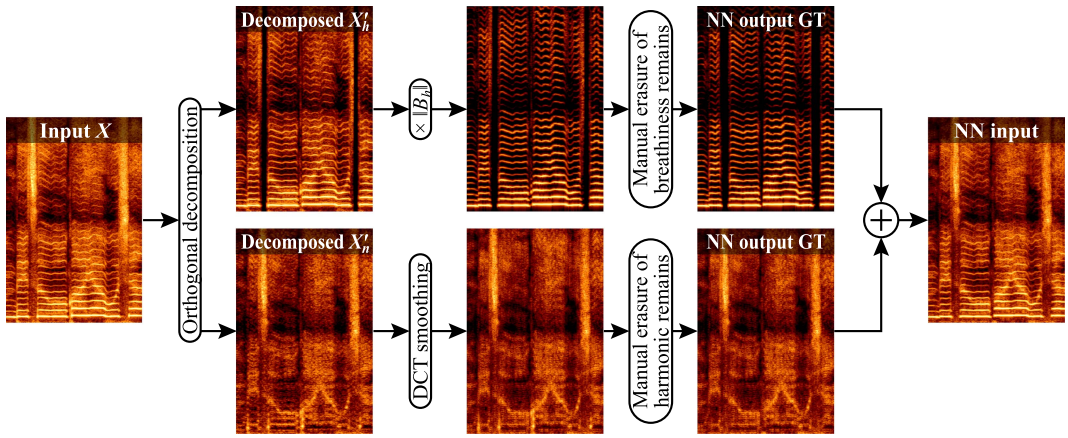


Fig. 5. Corpus preparation procedure for harmonics/breathiness separation model.

with reasonable equivalent F_0 would work. We use $s_n = \lfloor F_s/350 \rfloor$ in our experiments.

D. Harmonic/Breathiness Separation

Our task is to build a neural network model that, when given input complex spectrogram X, outputs complex spectrograms X_h and X_n representing harmonic and breathiness components respectively, and satisfying $X_h + X_n = X$. Existing approach uses orthogonal decomposition of $\frac{\partial}{\partial \tau} \arg X$ into $\frac{\partial}{\partial \tau} \arg B_h$, namely calculating vector projection of angular derivative of X over time onto that of periodic excitation B_h. One may also use angular derivative over frequency ω as in D4C instead of time and obtain similar results. Formally:

$$X'_h = X - \frac{\|X\|}{\|B_h\|} B_h \cos\left(\frac{\partial}{\partial \tau} \arg X - \frac{\partial}{\partial \tau} \arg B_h\right), \quad (8)$$

$$X'_n = X - X'_h. \quad (9)$$

The intuition behind this is that, with B_h consisting of pure sinusoidal waves, its first angular derivative over time is always $\frac{\partial \varphi}{\partial \tau} = 2\pi n F_0$ for $n \in \mathbb{N}$, meaning whenever $\frac{\partial}{\partial \tau} \arg X$ is different than $2\pi n F_0$, there is noise – at least the part of X that is orthogonal to it. We can then isolate a large part of X_n from X following (9). On the other hand, this also means that parts of X_n is still left in X'_h, as breathiness contains so much randomness that is almost never perfectly orthogonal to B_h.

Even though we cannot directly use separation results from orthogonal decomposition outputs X'_h and X'_n, we can still use them as a starting point for training a neural network by building a training set upon them:

- We attenuate leftover breathiness between harmonics in X'_h by multiplying it with $\|B_h\|$. Further remaining breathiness at V/UV boundaries (especially when next to fricative and affricate consonants) are erased manually using any tool capable of editing spectrogram. We adopt a conservative approach and eliminate any spectral components that are doubtful. This results in a refined X_h serving as the ground-truth for training.

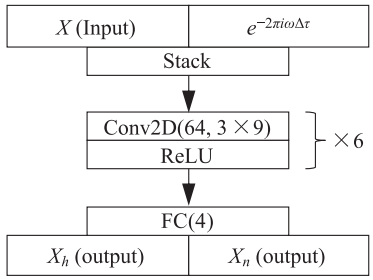


Fig. 6. Neural network architecture for harmonics/breathiness separation model. Blocks indicates with "×6" are repeated for six times.

- We use DCT-based smoothing as in (6) to smoothen $\|X'_n\|$ a little bit, keeping $\arg(X'_n)$ intact. Leftover harmonic fragments at V/UV boundaries are erased manually, following a similar procedure as above. This results in a refined X_n serving as the ground-truth for training.
- We use the sum of edited X_h and X_n as input to the neural network instead of the original X.

An example of the described corpus-building procedure is shown in Fig. 5. This training set does not need to be large. We used only a total of 10 minutes to obtain decent results.

The neural network architecture is designed simple on purpose, consisting of mostly repeated combinations of 2D-convolution and ReLU [44], as we only need to focus on local spectral features for separation. A full diagram of neural network architecture is shown in Fig. 6. We do not need to input B_h along with X as in orthogonal decomposition though, as we can already find good enough hints of phase change through time from the frequency scale ω. Specifically, we input, along with X, angular derivative over time as unit complex values for every frequency ω according to the time-shift property of Fourier transform:

$$\mathcal{F}\{x(t - d\tau)\}(\omega) = e^{-2\pi i \omega d\tau} \mathcal{F}\{x(t)\}(\omega), \quad (10)$$

$$\mathcal{F}\{\delta(t - d\tau)\}(\omega) = e^{-2\pi i \omega d\tau} 1. \quad (11)$$

Here, $\delta(\cdot)$ stands for the Dirac delta function. In practice, we set $d\tau$ to be the time interval between STFT frames. We trained our model using Adam optimizer and complex MAE loss for both

TABLE I
HYPER-PARAMETER SETTINGS FOR NEURAL VOCODERS IN COMPARISON

	SiFi-GAN	HiFi-GAN	UnivNet
Base Model Spec	Default	V1	c32
Max Spec Freq	22,050	20,000	20,000
Frame Shift	441	512	256
Upsampling Strides	[7, 7, 3, 3]	[8, 8, 4, 2]	[8, 8, 4]
Segment Length	370,440	24,576	24,576
Batch Size	16	18	60
Parameter Count	14,198,754	14,007,810	14,865,506
Trained Steps	500,000	500,000	552,123

X_h and X_n:

$$\mathcal{L}_1(X, \hat{X}) = \|X - \hat{X}\|. \tag{12}$$

IV. EXPERIMENTS

To show the effectiveness of our proposed methods, we included WORLD [4], an existing state-of-the-art plain-DSP vocoder, as well as several neural vocoders – SiFi-GAN [32], HiFi-GAN [15] and UnivNet [16] into our comparison. In all evaluation settings, we used a sampling rate of 44.1 kHz and FFT size 2048. The training corpus used is OpenSinger [45], an open-source singing voice dataset featuring 50 hours of vocal recording from 66 different singers, all singing in Chinese. While our method could be used on both speech and singing voice, we chose to train and evaluate solely on singing voice corpus because singing voice imposes higher requirements on audio quality (usually recorded on a professional condenser microphone) and features much larger vocal ranging from below 100 Hz to above 900 Hz.

SiFi-GAN as well as the synthesis part of our method were trained on a single instance of NVIDIA GeForce RTX 2080 Ti. HiFi-GAN and UnivNet were trained on 6 instances of NVIDIA GeForce RTX 2080 Ti. Note that neural vocoders typically only support 22.05 kHz officially. We thus made several adjustments to hyper-parameters so that they would work for our purpose. Moreover, all source-filter vocoders use frame shift 441 (10 ms) while neural vocoders have different frame shift values as part of their neural architecture design. Detailed hyper-parameter settings are shown in Table I.

We performed our evaluation procedure through three aspects:

- Audio fidelity is evaluated subjectively based on the MUSHRA scheme (MUltiple Stimuli with Hidden Reference and Anchor) [46].
- Pitch accuracy is evaluated objectively by calculating the difference between extracted F_0 curves of synthesized audio and that of original audio using a variety of F_0-extraction algorithms [5], [6], [47], [48].
- Runtime performance is evaluated objectively by comparing synthesis time cost between vocoders.

We provide sample audio files at `https://aqtq314.github.io/VogenSVS/VocoderV01/`. Source code and pretrained models are available at `https://github.com/aqtq314/VogenSVS`.

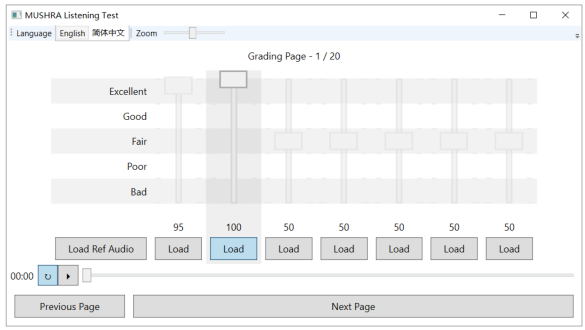


Fig. 7. Graphical user interface used for MUSHRA listening test.

A. Audio Fidelity

We set up a subjective evaluation procedure[1] according to the MUSHRA (MUltiple Stimuli with Hidden Reference and Anchor) specification [46]. MUSHRA is a subjective evaluation scheme designed specifically for audio compression algorithm. It features a reference input audio and several resynthesized ones, asking evaluators to grade each of them on a integer scale between 0 and 100. This allows us to see finer differences between pairs of items rather than a more general purpose scheme such as MOS score. It also allows us to evaluate over fidelity to the original audio instead of plain audio quality. The graphical user interface used for grading is shown in Fig. 7.

The evaluation procedure contained 20 different sets of dry vocal excerpts of 6 to 10 seconds from 20 different experienced Chinese pop singers (9 males and 11 females). All singers are unseen from training corpus. Two of them have an F_0 higher than the vocal range seen in training set (referred-to as unseen vocal range). Each set contained an original sound recording for reference, and several ones for grading, all in shuffled order:

- The original sound unmodified (as hidden reference);
- The original sound downsampled to 8 kHz – equivalent to applying a low-pass filter of 4 kHz (as hidden low-anchor);
- Copy-synthesis from WORLD;
- Copy-synthesis from proposed methods;
- Copy-synthesis from SiFi-GAN;
- Copy-synthesis from HiFi-GAN;
- Copy-synthesis from UnivNet.

The term "copy-synthesis" refers to synthesizing waveform directly from analyzed acoustic features. The purpose of having hidden reference and low-anchor is that, as specified in the MUSHRA specification, to ensure validity of collected responses by checking whether assessors are able to correctly identify items with no modification or those with highly perceivable defects through pure listening. We expect each assessor to score the hidden reference above 90 for at least 18 test sets, and the hidden low-anchor below 90 for all test sets.

We crowdsourced our evaluation results from Internet virtual singer communities. All evaluators had experience with at least one commercial SVS application and were aged between 18 to 30 years old. We collected from our evaluators 10 valid

[1]This experiment has been approved by the Department of Psychology Ethics Committee, Tsinghua University.

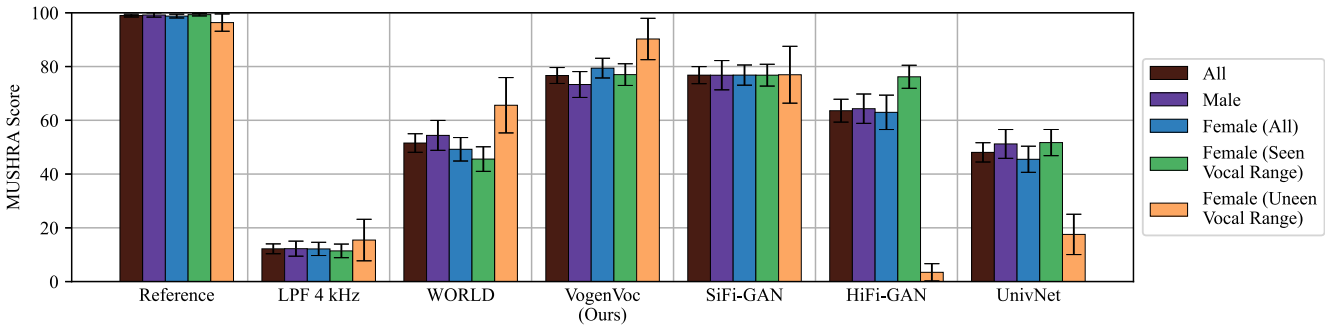


Fig. 8. Statistical visualizations of MUSHRA test results. Vertical axis represents MUSHRA score on audio fidelity to the reference audio. Error bar shows 95% confidence interval. Seen vocal range means F_0 of vocal excerpt does not go beyond vocal range seen in training dataset. Unseen vocal range means otherwise.

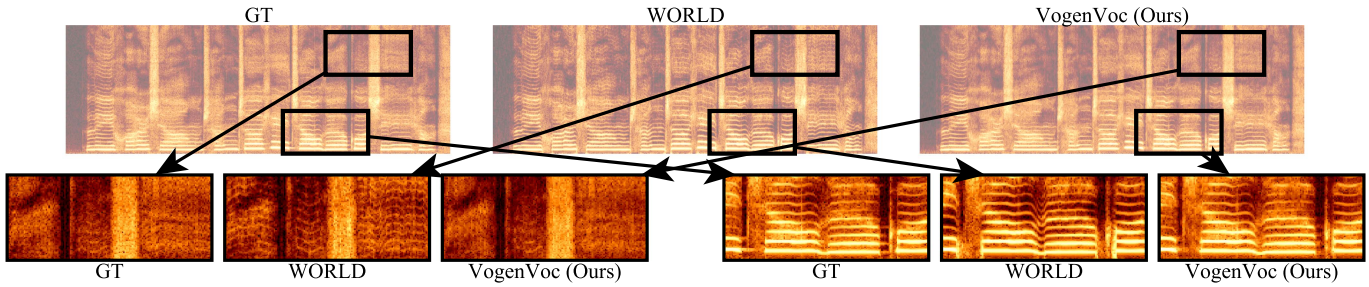


Fig. 9. Comparison of resynthesized audio spectrograms between plain-DSP vocoder WORLD and our method, with emphasis on avoiding enhanced harmonic overtones and noise at V/UV boundaries.

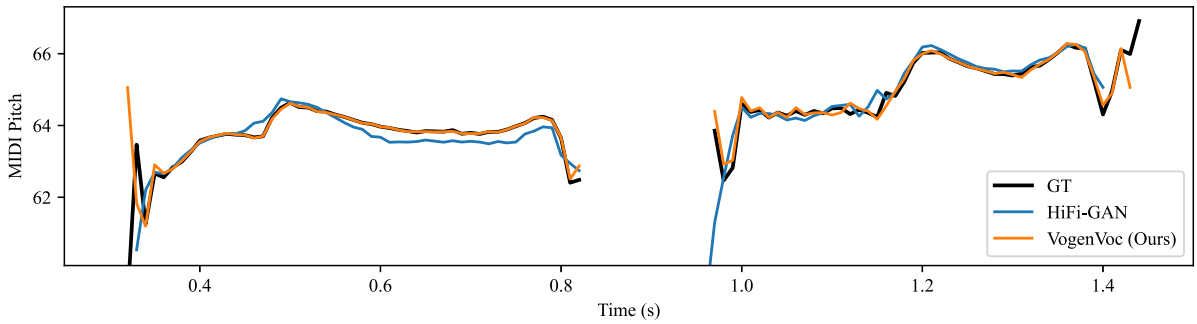
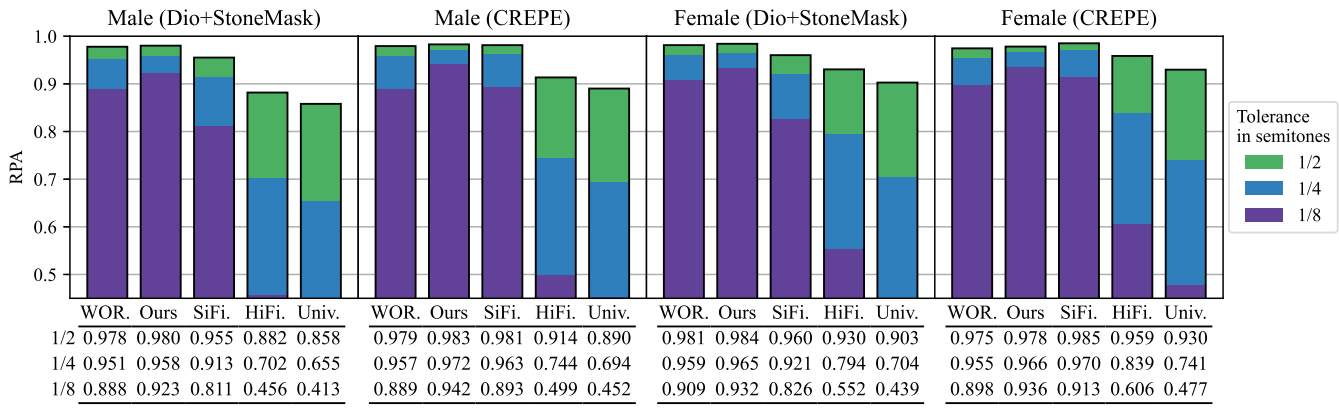


Fig. 10. Comparison of resynthesized audio F_0 curves (extracted using DIO+StoneMask) between neural vocoder HiFi-GAN and our method. Horizontal axis represents time in seconds. Vertical axis represents pitch in MIDI scale.

questionnaire responses, consisting of 200 score samples for each vocoder in comparison. Evaluation results are illustrated in Fig. 8. The overall results show that our method achieved a similar performance comparing to SiFi-GAN while scoring higher than the other two neural vocoders HiFi-GAN and UnivNet. Comparing with existing plain-DSP vocoder WORLD, our method could avoid enhanced harmonic overtones that generate a "metallic"-sounding artifact (bottom-left of Fig. 9), as well as low-frequency noise at V/UV boundaries (bottom-right of Fig. 9) with the use of a neural harmonic/breathiness decomposer. Moreover, our evaluators particularly pointed out that the breathiness part sounds more natural in our method after we replaced white-noise aperiodic excitation with an NN-based noise generator.

Comparison with neural vocoders is a little bit more complicated, as the results changed quite a bit when we tried to classify singers in test by their characteristics. While our method outperformed HiFi-GAN and UnivNet in all cases, it was inferior to SiFi-GAN for male voices and slightly superior for female voices. Notably, our method was the most robust at high pitches exceeding the vocal range seen in training set. Furthermore, source-filter vocoders have a natural advantage over neural vocoders without explicit periodic excitation modelling in terms of pitch fidelity to the input F_0 as well as continuity of harmonics. Fig. 10 shows an example where F_0 curve gets distorted when resynthesized with neural vocoder HiFi-GAN while staying mostly intact in our method. This is also evaluated objectively in more details in Section IV-B.

In the case of male voice, or voice with relatively low F_0 in general, our method often generates audio with overly heavy airflow that makes it sound more breathy than the original audio. This is because harmonic and breathiness components are not



	Male (Dio+StoneMask)					Male (CREPE)					Female (Dio+StoneMask)					Female (CREPE)				
	WOR.	Ours	SiFi.	HiFi.	Univ.	WOR.	Ours	SiFi.	HiFi.	Univ.	WOR.	Ours	SiFi.	HiFi.	Univ.	WOR.	Ours	SiFi.	HiFi.	Univ.
1/2	0.978	0.980	0.955	0.882	0.858	0.979	0.983	0.981	0.914	0.890	0.981	0.984	0.960	0.930	0.903	0.975	0.978	0.985	0.959	0.930
1/4	0.951	0.958	0.913	0.702	0.655	0.957	0.972	0.963	0.744	0.694	0.959	0.965	0.921	0.794	0.704	0.955	0.966	0.970	0.839	0.741
1/8	0.888	0.923	0.811	0.456	0.413	0.889	0.942	0.893	0.499	0.452	0.909	0.932	0.826	0.552	0.439	0.898	0.936	0.913	0.606	0.477

Fig. 11. Comparison of raw pitch accuracy (RPA) of synthesized audio between different genders among vocoders, evaluated using various F_0 estimation algorithms. The vertical axis shows the RPA score.

completely independent of each other in practice, unlike how we treat them in our methods where we separate them in the first step of analysis, and only combine them back in the last step of synthesis. WORLD partially avoids this with amplified harmonics, though is also generates artifacts in high-frequency bands. Full neural vocoders are able to handle this much better with a learned end-to-end pipeline. To tackle this problem, it will be necessary to redesign our pipeline so that finer relationship between harmonic and breathiness components are taken into account.

B. Pitch Accuracy

To further verify pitch accuracy among different vocoders, we performed an objective evaluation on raw pitch accuracy (RPA) as used in Music Information Retrieval Evaluation eXchange (MIREX) [49], defined as the proportion of frames in predicted $\hat{F}_0$ where the pitch stays within $\pm$ 1/2 semitones of the ground-truth F_0:

$$\mathrm{RPA}'_k(\hat{f}, f) = \mu\left(\left|12\log_2\left(\frac{f}{\hat{f}}\right)\right| - k\right), \tag{13}$$

$$\mathrm{RPA}_k(\hat{F}_0, F_0) = \frac{\sum_\tau V[\tau]\,\mathrm{RPA}'_k(\hat{F}_0[\tau], F_0[\tau])}{\sum_\tau V[\tau]}. \tag{14}$$

Here, μ stands for the unit step function. The V/UV mask V is 1 for voiced frames, 0 otherwise. The original definition in MIREX for tolerance threshold k is 0.5, meaning $\pm$ 1/2 semitones. We further added $\pm$ 1/4 and $\pm$ 1/8 semitones for a stricter verification in order to better illustrate their difference.

We resynthesized the entire OpenSinger corpus using different vocoders in test, and calculated RPA of resynthesized audio pieces. In addition to plain copy-synthesis with no pitch transposition, we also carried out synthesis using pitch ratio at 0.5, $\sqrt{0.5}$, $\sqrt{2}$ and 2, meaning the extracted F_0 curve is multiplied with each of those values and resynthesized to waveform audio, keeping other acoustic features unchanged. This also implies that vocoders without using explicit F_0 curves (e.g. HiFi-GAN and UnivNet) are ineligible to this method of evaluation, as pitch transposition directly on Mel-spectrograms is far from a trivial task. Due to reconstruction errors in F_0 estimation, we performed our test using 2 different F_0 estimation algorithms: DIO + StoneMask and CREPE.

Fig. 11 shows RPA evaluation results without pitch transposition. WORLD, ours and SiFi-GAN scored much higher than the other two under all F_0 estimation algorithms. This is expected as the three vocoders use DSP-based oscillators as periodic excitation generator. In addition, Fig. 12 shows RPA evaluation results for WORLD, ours and SiFi-GAN under pitch-transposed scenarios. While the results do demonstrate slightly growing degredation in some cases as pitch ratio deviates further from 1, the RPA scores are still far above HiFi-GAN and UnivNet even though the latter two have an unfair advantage by doing copy synthesis.

Moreover, our method scored slightly higher than WORLD as we perform filtering through direct multiplication of spectral envelope and excitation signal instead of calculating every single impulse response as implemented in WORLD. In practice, this marginal improvement over WORLD is barely perceivable to human. The essential idea is to show that our method have achieved a far superior RPA than neural vocoders without explicit periodic excitation as input. In addition, all vocoders scored slightly higher for female voices than for male voices, though this did not affect the overall comparison between vocoders.

C. Runtime Performance

To verify runtime speed of proposed methods, we measured the average time cost for synthesis among vocoders. Measurements were carried out on several hardware settings including various GPU and CPU. All neural vocoders as well as our method were implemented in PyTorch. In the case of WORLD, we used the version implemented in C++, meaning it did not take part in tests carried out on GPU. All tests were run on a single CPU thread with the help of benchmarking utilities built-in to PyTorch.

Results are shown in Fig. 13. In all hardware settings, our method achieved significant speedup over other neural vocoders, with or without the help of GPU-based parallelism. Also, our method ran even slightly faster than WORLD with the help of PyTorch library optimizations. It is worth noting that, since tests on WORLD was carried out on a different runtime library than PyTorch, its results may fluctuate when tested on a different environment, though unlikely to change drastically. It is still fair to say that our method has achieved similar runtime efficiency to that of a conventional source-filter vocoder.

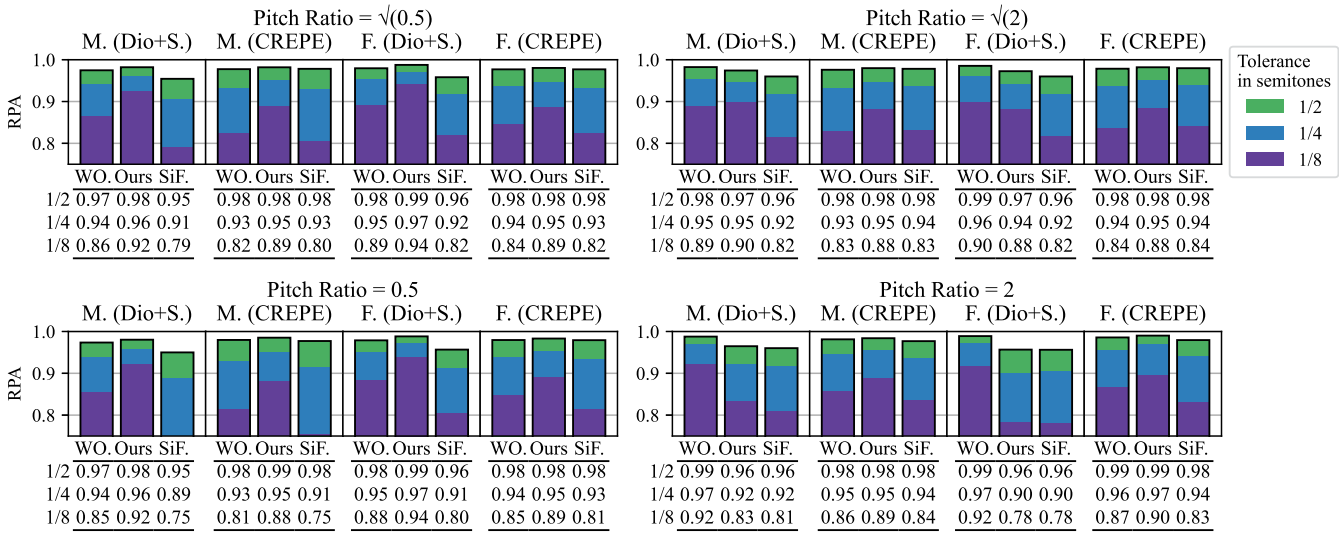


Fig. 12. Comparison of raw pitch accuracy (RPA) of pitch-transposed synthesized audio among vocoders using explicit F_0 curve as part of acoustic features, evaluated for different genders and pitch transposition ratios using various F_0 estimation algorithms. The vertical axis shows the RPA score.

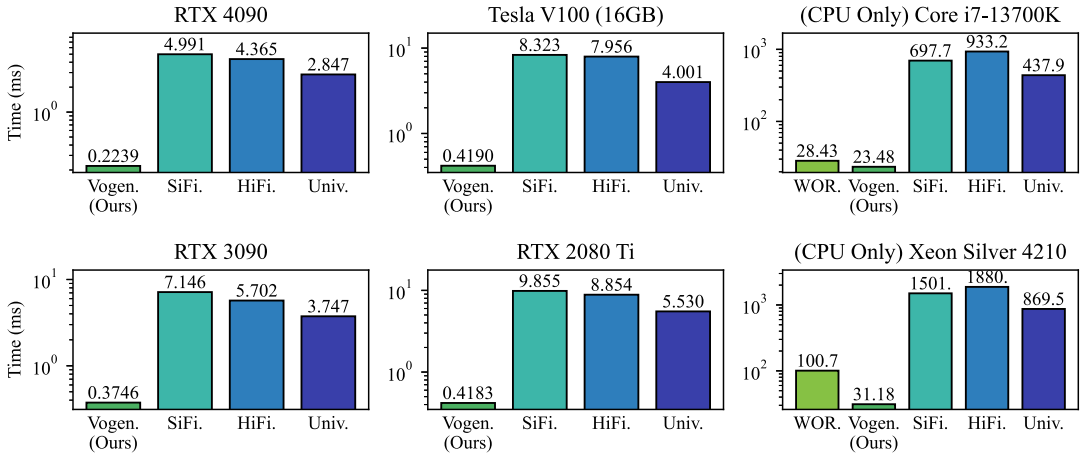


Fig. 13. Runtime performance comparison among vocoders under various hardware settings. The numbers are time cost in milliseconds per synthesis of a 1-second audio. All tests were run on a single CPU thread.

V. Conclusion

In this article, we present a novel source-filter vocoding pipeline equipped with small, lightweight and task-specific neural networks. In particular, we constructed a neural network model for decomposition into harmonic and breathiness components from an input audio, as well as a noise generator neural network model for aperiodic excitation. Our subjective and objective evaluation procedures have shown that our method was able to synthesize audio at a level of fidelity comparable to neural vocoders, while still keeping high runtime efficiency comparable to conventional DSP vocoder and high pitch accuracy as from a typical source-filter vocoder.

References

[1] A. v. d. Oord et al., "WaveNet: A generative model for raw audio," in *Proc. 9th ISCA Workshop Speech Synth. Workshop*, 2016, p. 125.

[2] H. Kawahara, "STRAIGHT, Exploitation of the other aspect of VOCODER: Perceptually isomorphic decomposition of speech sounds," *Acoustical Sci. Technol.*, vol. 27, no. 6, pp. 349–353, 2006.

[3] H. Kawahara, M. Morise, T. Takahashi, R. Nisimura, T. Irino, and H. Banno, "Tandem-STRAIGHT: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, F0, and aperiodicity estimation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2008, pp. 3933–3936.

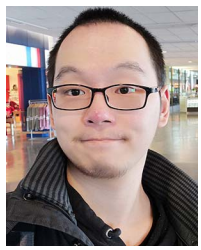
[4] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Trans. Inf. Syst.*, vol. 99-D, pp. 1877–1884, 2016.

[5] M. Morise, H. Kawahara, and H. Katayose, "Fast and reliable F0 estimation method based on the period extraction of vocal fold vibration of singing voice and speech," in *Proc. Audio Eng. Soc. Conf.: 35th Int. Conf.: Audio Games*, 2009. [Online]. Available: https://www.aes.org/e-lib/browse.cfm?elib=15165

[6] M. Morise, "Harvest: A high-performance fundamental frequency estimator from speech signals," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2017, pp. 2321–2325.

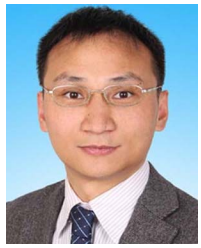
[7] S. Mehri et al., "SampleRNN: An unconditional end-to-end neural audio generation model," in *Proc. Int. Conf. Learn. Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=SkxKPDv5xl

[8] Z. Jin, A. Finkelstein, G. J. Mysore, and J. Lu, "FFTNet: A real-time speaker-dependent neural vocoder," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 2251–2255.
[9] N. Kalchbrenner et al., "Efficient neural audio synthesis," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 2410–2419. [Online]. Available: https://proceedings.mlr.press/v80/kalchbrenner18a.html
[10] R. Prenger, R. Valle, and B. Catanzaro, "WaveGlow: A flow-based generative network for speech synthesis," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 3617–3621.
[11] A. v. d. Oord et al., "Parallel WaveNet: Fast high-fidelity speech synthesis," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 3918–3926. [Online]. Available: https://proceedings.mlr.press/v80/oord18a.html
[12] W. Ping, K. Peng, and J. Chen, "ClariNet: Parallel wave generation in end-to-end text-to-speech," in *Proc. Int. Conf. Learn. Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=HklY120cYm
[13] 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. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 6199–6203.
[14] K. Kumar et al., "MelGAN: Generative adversarial networks for conditional waveform synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, Art. no. 1335. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/6804c9bca0a615bdb9374d00a9fcba59-Paper.pdf
[15] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 17022–17033. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/c5d736809766d46260d816d8dbc9eb44-Paper.pdf
[16] 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. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 2207–2211.
[17] T. Kaneko, K. Tanaka, H. Kameoka, and S. Seki, "ISTFTNet: Fast and lightweight mel-spectrogram vocoder incorporating inverse short-time Fourier transform," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 6207–6211.
[18] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, "WaveGrad: Estimating gradients for waveform generation," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=NsMLjcFaO8O
[19] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "DiffWave: A versatile diffusion model for audio synthesis," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=a-xFK8Ymz5J
[20] Y. Koizumi, H. Zen, K. Yatabe, N. Chen, and M. Bacchiani, "SpecGrad: Diffusion probabilistic model based neural vocoder with adaptive noise spectral shaping," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 803–807.
[21] N. Takahashi, M. Kumar, Singh, and Y. Mitsufuji, "Hierarchical diffusion models for singing voice neural vocoder," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.
[22] 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. Audio, Speech, Lang. Process.*, vol. 29, pp. 1134–1148, 2021.
[23] 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. Audio, Speech, Lang. Process.*, vol. 29, pp. 792–806, 2021.
[24] J.-M. Valin and J. Skoglund, "LPCNET: Improving neural speech synthesis through linear prediction," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 5891–5895.
[25] L. Juvela, B. Bollepalli, J. Yamagishi, and P. Alku, "GELP: GAN-excited linear prediction for speech synthesis from mel-spectrogram," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 694–698.
[26] 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. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 6915–6919.
[27] Z. Liu, K. Chen, and K. Yu, "Neural homomorphic vocoder," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 240–244, doi: 10.21437/Interspeech.2020-3188.
[28] X. Wang, S. Takaki, and J. Yamagishi, "Neural source-filter-based waveform model for statistical parametric speech synthesis," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 5916–5920.
[29] X. Wang, S. Takaki, and J. Yamagishi, "Neural source-filter waveform models for statistical parametric speech synthesis," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 402–415, 2020.
[30] X. Wang and J. Yamagishi, "Using cyclic noise as the source signal for neural source-filter-based speech waveform model," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 1992–1996.
[31] 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. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 2187–2191.
[32] R. Yoneyama, Y.-C. Wu, and T. Toda, "Source-filter HiFi-GAN: Fast and pitch controllable high-fidelity neural vocoder," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.
[33] H. Dudley, "Remaking speech," *J. Acoustical Soc. Amer.*, vol. 11, no. 2, pp. 169–177, 1939.
[34] A. M. Noll, "Short-time spectrum and "cepstrum" techniques for vocal-pitch detection," *J. Acoustical Soc. Amer.*, vol. 36, no. 2, pp. 296–302, 1964.
[35] A. V. Oppenheim, "Speech analysis-synthesis system based on homomorphic filtering," *J. Acoustical Soc. Amer.*, vol. 45, no. 2, pp. 458–465, 1969.
[36] B. S. Atal and S. L. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," *J. Acoustical Soc. Amer.*, vol. 50, no. 2B, pp. 637–655, 1971.
[37] J. Markel and A. Gray, "A linear prediction vocoder simulation based upon the autocorrelation method," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 22, no. 2, pp. 124–134, Apr. 1974.
[38] M. Morise, "CheapTrick, a spectral envelope estimator for high-quality speech synthesis," *Speech Commun.*, vol. 67, pp. 1–7, 2015.
[39] M. Morise, "D4C, A band-aperiodicity estimator for high-quality speech synthesis," *Speech Commun.*, vol. 84, pp. 57–65, 2016.
[40] M. Morise, G. Miyashita, and K. Ozawa, "Low-dimensional representation of spectral envelope without deterioration for full-band speech analysis/synthesis system," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2017, pp. 409–413.
[41] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *Proc. Int. Conf. Learn. Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=B1x1ma4tDr
[42] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1800–1807.
[43] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 2, pp. 236–243, Apr. 1984.
[44] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, 2010, pp. 807–814.
[45] R. Huang, F. Chen, Y. Ren, J. Liu, C. Cui, and Z. Zhao, "Multi-singer: Fast multi-singer singing voice vocoder with a large-scale corpus," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 3945–3954.
[46] *Method for the Subjective Assessment of Intermediate Quality Level of Audio Systems*, ITU-R Rec. BS.1534-2 Int. Telecomm. Union, 2014.
[47] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 161–165.
[48] B. Gfeller, C. Frank, D. Roblek, M. Sharifi, M. Tagliasacchi, and M. Velimirović, "SPICE: Self-supervised pitch estimation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 1118–1128, 2020.
[49] J. Salamon, E. Gómez, D. P. Ellis, and G. Richard, "Melody extraction from polyphonic music signals: Approaches, applications, and challenges," *IEEE Signal Process. Mag.*, vol. 31, no. 2, pp. 118–134, Mar. 2014.



Runxuan Yang received the B.Sc. degree in computer science from McGill University, Montreal, QC, Canada, in 2015, and the M.Sc. degree in computer science and technology in 2019 from Tsinghua University, Beijing, China, where he is currently working toward the Ph.D. degree in computer science and technology. His research interests include of singing voice synthesis and speech recognition.

Yuyang Peng received the B.E. degree in automation from Tsinghua University, Beijing, China, in 2023. He is currently a Graduate Student belonging to the Brain and Cognitive Institute, Department of Automation, Tsinghua University. His research interests include latent diffusion model and conditional video generation.



Xiaolin Hu (Senior Member, IEEE) received the B.E. and M.E. degrees in automotive engineering from the Wuhan University of Technology, Wuhan, China, in 2001 and 2004, respectively, and the Ph.D. degree in automation and computer-aided engineering from The Chinese University of Hong Kong, Hong Kong, in 2007. He is currently an Associate Professor with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His current research interests include deep learning and computational neuroscience. He is currently an Associate Editor for IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING.