frontiers | Frontiers in Signal Processing

TYPE Review
PUBLISHED 11 January 2024
DOI 10.3389/frsip.2023.1284100

OPEN ACCESS

EDITED BY
Cumhur Erkut,
Aalborg University, Denmark

REVIEWED BY
Stefano Fasciani,
University of Oslo, Norway
Chengshi Zheng,
Chinese Academy of Sciences (CAS), China
Shahan Nercessian,
iZotope/Native Instruments, United States

*CORRESPONDENCE
Jordie Shier,
j.m.shier@qmul.ac.uk
Ben Hayes,
b.j.hayes@qmul.ac.uk

RECEIVED 27 August 2023
ACCEPTED 08 December 2023
PUBLISHED 11 January 2024

CITATION
Hayes B, Shier J, Fazekas G, McPherson A and Saitis C (2024), A review of differentiable digital signal processing for music and speech synthesis.
Front. Sig. Proc. 3:1284100.
doi: 10.3389/frsip.2023.1284100

A review of differentiable digital signal processing for music and speech synthesis

Ben Hayes[1]*, Jordie Shier[1]*, György Fazekas[1], Andrew McPherson[2] and Charalampos Saitis[1]

[1]Centre for Digital Music, School of Electronic Engineering and Computer Science, Queen Mary University of London, London, United Kingdom, [2]Dyson School of Design Engineering, Imperial College London, London, United Kingdom

The term "differentiable digital signal processing" describes a family of techniques in which loss function gradients are backpropagated through digital signal processors, facilitating their integration into neural networks. This article surveys the literature on differentiable audio signal processing, focusing on its use in music and speech synthesis. We catalogue applications to tasks including music performance rendering, sound matching, and voice transformation, discussing the motivations for and implications of the use of this methodology. This is accompanied by an overview of digital signal processing operations that have been implemented differentiably, which is further supported by a web book containing practical advice on differentiable synthesiser programming (https://intro2ddsp.github.io/). Finally, we highlight open challenges, including optimisation pathologies, robustness to real-world conditions, and design trade-offs, and discuss directions for future research.

KEYWORDS

digital signal processing, machine learning, audio synthesis, automatic differentiation, neural networks

1 Introduction

Audio synthesis, the artificial production of sound, has been an active field of research for over a century. Early inventions included entirely new categories of musical instruments (Cahill, 1897; Ssergejewitsch, 1928) and the first machines for artificial speech production (Dudley, 1939; Dudley and Tarnoczy, 1950), while the latter half of the 20th century saw a proliferation of research into digital methods for sound synthesis, built on advances in signal processing (Keller, 1994; Smith, 2010) and numerical methods (Bilbao, 2009). Applications of audio synthesis have since come to permeate daily life, from music (Holmes, 2008), through voice assistant technology, to the sound design in films, TV shows, video games, and even the cockpits of cars (Dupre et al., 2021).

In recent years, the field has undergone something of a technological revolution. The publication of WaveNet (van den Oord et al., 2016), an autoregressive neural network which produced a quantised audio signal sample-by-sample, first illustrated that deep learning might be a viable methodology for audio synthesis. Over the following years, new techniques for neural audio synthesis—as these methods came to be known—abounded, from refinements to WaveNet (Oord et al., 2018) to the application of entirely different classes of generative model (Donahue et al., 2019; Kumar et al., 2019; Kong et al., 2020; Chen et al., 2021), with the majority of work focusing on speech (Tan et al., 2021) and music (Huzaifah and Wyse, 2021) synthesis.

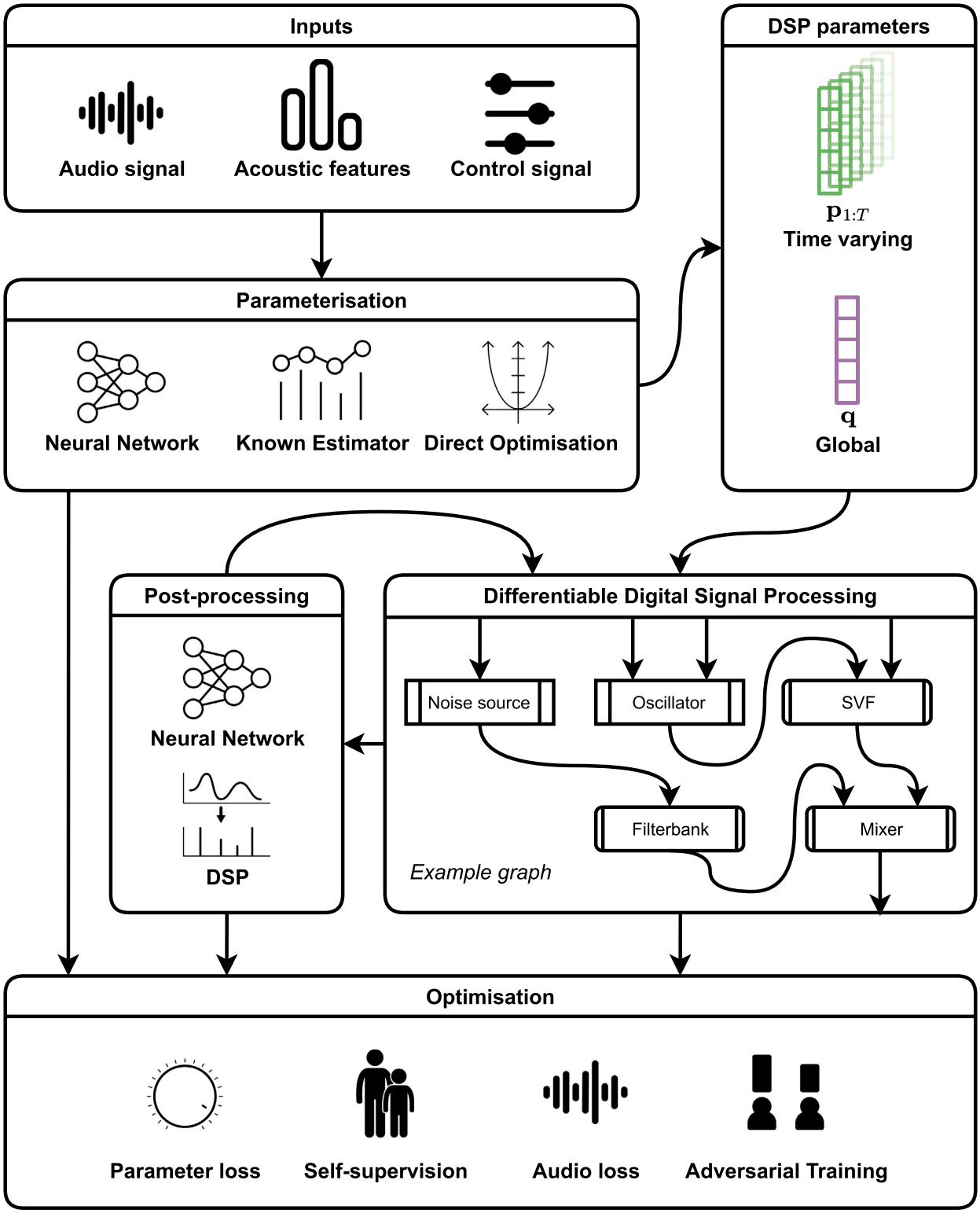

FIGURE 1
A high level overview of the general structure of a typical DDSP synthesis system. Not every depicted component is present in every system, however we find this structure broadly encompasses the work we have surveyed. Graphical symbols are included for illustrative purposes only.

Nonetheless, modelling audio signals remained challenging. Upsampling layers, crucial components of workhorse architectures such as generative adversarial networks (Goodfellow et al., 2014) and autoencoders, were found to cause undesirable signal artifacts (Pons et al., 2021). Similarly, frame-based estimation of audio signals was also found to be more challenging than might

TABLE 1 A summary of DDSP synthesis papers reviewed in compiling this article. Papers are grouped by major application area. Those which are applied to more than one area are grouped with their primary application.

Authors	Year	Contributions
Speech Synthesis		
Valin and Skoglund	2019	LPC integration to WaveRNN
Wang et al.	2019a	Neural source-filter (NSF)
Juvela et al.	2019	Differentiable LPC
Wang and Yamagishi	2019	Differentiable sinc FIR design
Wang et al.	2019b	Further NSF models
Wang and Yamagishi	2020	Cyclic noise source for NSF
Liu et al.	2020	Neural homomorphic vocoder (NHV)
Mv and Ghosh	2020	Fully differentiable source-filter model
Tian et al.	2020	Multi-band LPC
Vipperla et al.	2020	Bunched LPC
Fabbro et al.	2020	Differentiable harmonic-plus-noise for speech
Nercessian	2021	Harmonic-plus-noise for voice conversion
Subramani et al.	2022	Differentiable LPC estimation
Choi et al.	2023a	Hybrid model with self-supervised disentanglement
Kaneko et al.	2022	Differentiable ISTFT-based vocoder
Webber et al.	2023	Differentiable ISTFT-based vocoder
Watts et al.	2023	Differentiable pitch synchronous overlap add
Südholt et al.	2023	Differentiable digital waveguide (Kelly-Lochbaum)
Song et al.	2023	Hybrid model combining DDSP with GAN vocoder
Music Synthesis		
Engel et al.	2020a	DDSP library; differentiable spectral modelling synthesiser
Zhao et al.	2020	NSF applied to musical instrument synthesis
Michelashvili and Wolf	2020	Hierarchical NSF model
Castellon et al.	2020	DDSP-based performance rendering
Jonason et al.	2020	DDSP-based performance rendering
Caillon and Esling	2021	Hybrid real-time audio generative model
Carney et al.	2021	Efficient in-browser DDSP implementation; numerically stable TF.js kernels
Hayes et al.	2021	Differentiable waveshaping synthesiser
Masuda and Saito	2021	Differentiable subtractive synthesiser
Caspe et al.	2022	Differentiable FM synthesiser
Diaz et al.	2023	Differentiable modal synthesiser
Shan et al.	2022	Differentiable wavetable synthesiser
Kawamura et al.	2022	DDSP-based mixture model for synthesis parameter estimation
Renault et al.	2022	Differentiable piano model; explicit inharmonicity modelling
Wu et al.	2022c	DDSP-based performance modelling
Masuda and Saito	2023	Semi-supervised hybrid training; differentiable ADSR
Ye et al.	2023	Neural architecture search over differentiable FM synthesisers
Shier et al.	2023	Hybrid NSF model for percussion synthesis

(Continued on following page)

TABLE 1 (*Continued*) A summary of DDSP synthesis papers reviewed in compiling this article. Papers are grouped by major application area. Those which are applied to more than one area are grouped with their primary application.

Authors	Year	Contributions
Singing Voice Synthesis		
Alonso and Erkut	2021	Experiments on autoencoder from Engel et al. (2020a) for singing voice synthesis
Wu et al.	2022a	Differentiable subtractive singing voice synthesiser
Guo et al.	2022a	Differentiable filtering of sine excitation for adversarial SVC
Yoshimura et al.	2023	Differentiable mel cepstral synthesis filter[21]
Nercessian	2023	Differentiable WORLD vocoder
Yu and Fazekas	2023	Glottal-flow wavetable; efficient all-pole IIR training algorithm and implementation
Other		
Shynk	1989	Gradient-based IIR optimisation
Back and Tsoi	1991	Efficient gradient-based training algorithms for FIR and IIR based neural networks
Campolucci et al.	1995	Approximate online learning algorithms for IIR networks
Bhattacharya et al.	2020	Differentiable IIR filters with instantaneous backpropagation through time
Kuznetsov et al.	2020	Differentiable IIR filters with truncated backpropagation through time
Nercessian	2020	Differentiable IIR filters via frequency sampling
Engel et al.	2020b	Differentiable additive sinusoidal model; self-supervised hybrid training
Turian and Henry	2020	Experiments on differentiable frequency estimation pathologies
Turian et al.	2021	Differentiable modular synthesiser; billion sound dataset
Martinez Ramirez et al.	2021	Stochastic approximation of black-box signal processor gradients
Nercessian et al.	2021	Differentiable hyperconditioned IIR filters; stability preserving activations
Colonel et al.	2022	Random polynomial sampling for differentiable IIR self-supervision
Hagiwara et al.	2022	Experiments on animal vocal sound modelling via DDSP
Lee et al.	2022	Differentiable artificial reverberation
Steinmetz et al.	2022a	Neural proxies for DDSP; evaluation of gradient estimation methods
Barahona-Ríos and Collins	2023	Sound effect synthesis with differentiable multiband noise synthesiser
Carson et al.	2023	Differentiable grey-box phaser model; differentiable LFO estimation
Hayes et al.	2023	Differentiable frequency estimation; surrogate model for sinusoidal oscillator
Schulze-Forster et al.	2023	Unsupervised source separation with differentiable source models

naïvely be assumed, due to the difficulty of ensuring phase coherence between successive frames, where frame lengths are independent of the frequencies contained in a signal (Engel et al., 2019).

Aiming to address such issues, one line of research explored the integration of domain knowledge from speech synthesis and signal processing into neural networks. Whilst some methods combined the outputs of classical techniques with neural networks (Valin and Skoglund, 2019), others integrated them by expressing the signal processing elements differentiably (Wang et al., 2019b; Juvela et al., 2019). This was crystalized in the work of Engel et al. (2020a), who introduced the terminology *differentiable digital signal processing* (DDSP). In particular, Engel et al. suggested that some difficulties in neural audio synthesis could be explained by certain biases induced by the underlying models. The proposed advantage of DDSP was thus to gain a domain-appropriate inductive bias by incorporating a known signal model to the neural network. Implementing the signal model differentiably allowed loss gradients to be backpropagated through its parameters, in a manner similar to differentiable rendering (Kato et al., 2020).

In subsequent years, DDSP was applied to tasks including music performance synthesis (Jonason et al., 2020; Wu et al., 2022c), instrument modelling (Renault et al., 2022), synthesiser sound matching (Masuda and Saito, 2021), speech synthesis and voice transformation (Choi H.-S. et al., 2023), singing voice synthesis and conversion (Nercessian, 2023; Yu and Fazekas, 2023), sound-effect generation (Hagiwara et al., 2022; Barahona-Ríos and Collins, 2023).

The technology has also been deployed in a number of publicly available software instruments and real-time tools.[1] Figure 1 illustrates the general structure of a typical DDSP synthesis system and we list included papers in Table 1.

Differentiable signal processing has also been applied in tasks related to audio engineering, such as audio effect modelling (Kuznetsov et al., 2020; Lee et al., 2022; Carson et al., 2023), automatic mixing and intelligent music production (Martinez Ramirez et al., 2021; Steinmetz et al., 2022a), and filter design (Colonel et al., 2022). Whilst many innovations from this work have found use in synthesis, and *vice versa*, we do not set out to comprehensively review these tasks areas. Instead, we address this work where it is pertinent to our discussion of differentiable audio synthesis, and refer readers to the works of Ramírez et al. (2020), Moffat and Sandler (2019), De Man et al. (2019), for reviews of the relevant background, and to the work of Steinmetz et al. (2022b) for a summary of the state of differentiable signal processing in this field.

In the wake of a proliferation of work applying DDSP to audio synthesis, we make two key observations. Firstly, DDSP is increasingly acknowledged as a promising methodology, and secondly, the application of DDSP presents non-trivial challenges that have only recently begun to be thoroughly addressed in the literature. We argue that this disparity between successful applications and demonstrations of fundamental issues such as optimisation instability leads to a degree of ambiguity, rendering it unclear whether DDSP is appropriate for specific task, or even likely to work at all. Consequently, this article aims to clearly delineate the capabilities and limitations of DDSP-based methods, through a comprehensive treatment of existing research. Additionally, we endeavour to consolidate the wide variety of techniques under the DDSP umbrella, particularly across the music and speech domains, aiming to facilitate future research and prevent the duplication of efforts in these intersecting fields.

The terms *differentiable digital signal processing* and *DDSP* have been ascribed various meanings in the literature. For the sake of clarity, whilst also wishing to acknowledge the contributions of Engel et al. (2020a), we therefore adopt the following disambiguation in this article.

1. We use the general term *differentiable digital signal processing* and the acronym *DDSP* to describe the technique of implementing digital signal processing operations using automatic differentiation software.
2. To refer to Engel, et al.'s Python library, we use the term *the DDSP library*.
3. We refer to the differentiable spectral modelling synthesiser and neural network controller introduced by Engel et al. (2020a), like other work, in terms of their specific contributions, e.g., *Engel, et al.'s differentiable spectral-modelling synthesiser*.

2 Applications and tasks

A high level view of the tasks discussed in this section is given in Figure 3. In this section, we survey the tasks and application areas in which DDSP-based audio synthesis has been used, focusing on two goals. Firstly, we aim to provide sufficient background on historical approaches to contextualize the discussion on applications of differentiable signal processing to audio synthesis. Secondly, we seek to help practitioners in the task areas listed below, and in related fields, answer the question, "what is currently possible with differentiable digital signal processing?" For readers interested in the differing considerations in speech and music synthesis, we refer to the discussion by Schwarz (2007).

2.1 Musical audio synthesis

Synthesisers play an integral role in modern music creation, offering musicians nuanced control over musical timbre Holmes (2008). Applications of audio synthesis are diverse, ranging from faithful digital emulation of acoustic musical instruments to the creation of unique and novel sounds. Schwarz (2007) proposed a division of techniques for musical audio synthesis into parametric–including signal models, such as spectral modelling synthesis (Serra and Smith, 1990), and physical models, such as digital waveguides (Smith, 1992)–and concatenative families–which segment, reassemble, and align samples from corpora of pre-recorded audio (Schwarz, 2006).[2] We propose an updated version of this classification in Figure 2, accommodating developments in neural audio synthesis and DDSP.

Compared to other domains, music has particularly stringent requirements for audio synthesisers. Real-time inference is a necessity for integration of synthesisers into digital musical instruments, where action-sound latencies above 10 ms are likely to be disruptive (Jack et al., 2018). This has previously been challenging to address with generative audio models (Huzaifah and Wyse, 2021), particularly at the high sample rates demanded by musical applications. Expressive control over generation is also necessary in order to provide meaningful interfaces for musicians (Devis et al., 2023). The reliability of this control is also crucial, yet often challenging. Pitch coherence, for example, is a known issue with GAN-based audio generation (Song et al., 2023). Further, the comparative scarcity of high quality musical training data further compounds these issues for generative models.

2.1.1 Musical instrument synthesis

In response to the challenges of neural music synthesis, Engel et al. (2020a) implemented a differentiable *spectral modelling* synthesiser (Serra and Smith, 1990) and effectively replaced its parameter estimation algorithm with a recurrent neural network. Specifically, the oscillator bank was constrained to harmonic frequencies, an inductive bias which enabled monophonic modelling of instruments with predominantly harmonic spectra,

1 These include Google Magenta's DDSP-VST (https://magenta.tensorflow.org/ddsp-vst), Bytedance's Mawf (https://mawf.io/), Neutone Inc.'s Neutone (https://neutone.space/), ACIDS-IRCAM's *ddsP* (https://github.com/acids-ircam/ddsp_pytorch) and Aalborg University's JUCE implementation (https://github.com/SMC704/juce-ddsp). Accessed 21st August 2023.

2 Concatenative methods continue to underpin the dominant professional tools for realistically simulating musical instruments. For example, the EastWest instrument libraries. https://www.soundsonline.com/hollywood-solo-series. Accessed 10th August 2023.

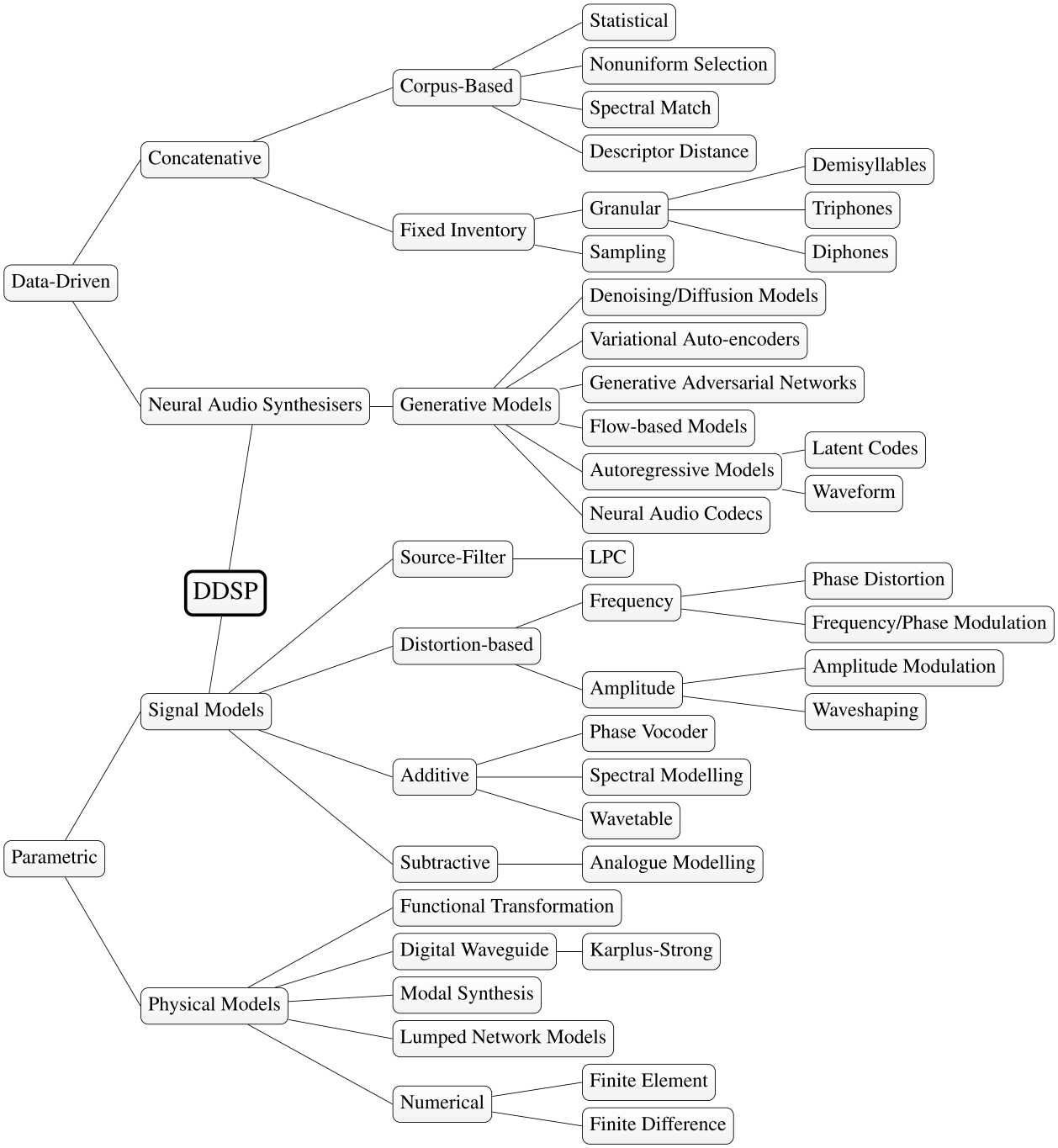

FIGURE 2
A high level taxonomy of popular sound synthesis methods, based on the classifications of Schwarz (2007) and Bilbao (2009). DDSP methods (bold) offer a combination of data-driven and parametric characteristics. This diagram is illustrative of high level relationships, and is not intended to exhaustively catalogue all audio synthesis techniques.

including violin performances from the MusOpen library[3] and instruments from the NSynth dataset (Engel et al., 2017). The resulting model convincingly reproduced certain instrument sounds from as little as 13 min of training data. It also allowed, through its low dimensional control representation, similarly convincing timbre transfer between instruments (Carney et al., 2021).

Building on this success, a number of subsequent works applied various synthesis methods to monophonic and harmonic instrument synthesis including waveshaping synthesis (Hayes et al., 2021), frequency modulation synthesis (Caspe et al., 2022; Ye et al., 2023), and wavetable synthesis (Shan et al., 2022).

3 https://musopen.org/. Accessed 26th August 2023.

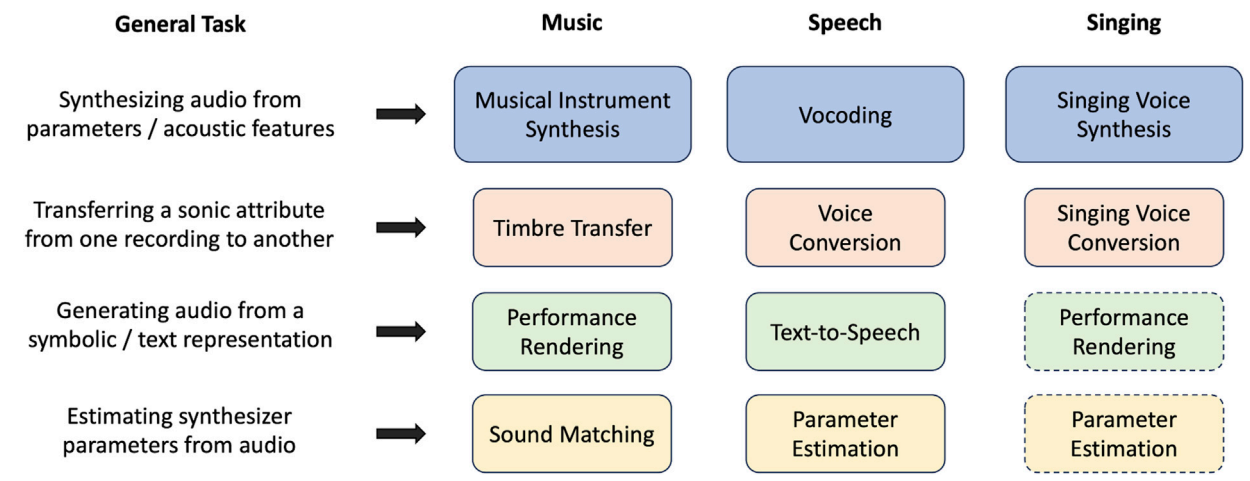

FIGURE 3
A high level view of audio synthesis tasks to which DDSP has been applied. Further discussion on each is presented in Section 2.

These works approach musical audio synthesis as the task of modelling time-varying harmonics and optionally filtered noise, utilizing input loudness and fundamental frequency signals as conditioning. Hayes et al. (2021) and Shan et al. (2022) focus on improving computational efficiency, demonstrating how learned waveshapers and wavetables, respectively, can reduce the inference-time cost. Caspe et al. (2022) and Ye et al. (2023) focus on control and interpretability of the resulting synthesiser. In contrast to the dense parameter space of an additive synthesier, they apply FM synthesis which, despite its complex parameter interrelationships, facilitates user intervention post-training due to its vastly smaller parameter count. Ye et al. (2023) built upon the work of Caspe et al. (2022), further tailoring the FM synthesiser to a target instrument through neural architecture search (Ren et al., 2022) over modulation routing.

Two common characteristics of the works discussed in this section are a reliance on pitch tracking, and the assumption of predominantly harmonic spectra. This, we argue, is due to the difficulty of performing gradient descent over oscillator frequency, discussed further in Section 3.2.2. Consequently, adaptation of such methods to polyphonic, unpitched, and non-harmonic instruments, or to modelling inharmonicity induced by stiffness and nonlinear acoustic properties, is challenging. Nonetheless, domain-specific solutions have been proposed. Renault et al. (2022), for example, proposed a method for polyphonic piano synthesis, introducing an extended pitch to model string resonance after note releases, explicit inharmonicity modeling based on piano tuning, and detuning to replicate partial interactions on piano strings. Diaz et al. (2023) presented a differentiable signal model of inharmonic percussive sounds using a bank of resonant IIR filters, which they trained to match the frequency responses produced by modal decomposition using the finite element method. This formulation was able to converge on the highly inharmonic resonant frequencies produced by excitation of arbitrarily shaped rigid bodies, with varying material parameters.

2.1.2 Timbre transfer

Building on the success of neural style transfer in the image domain (Gatys et al., 2015), timbre transfer emerged as a task in musical audio synthesis. Dai et al. (2018) define timbre transfer as the task of altering "timbre information in a meaningful way while preserving the hidden content of performance control." They highlight that it requires the disentanglement of timbre and performance, giving the example of replicating a trumpet performance such that it sounds like it was played on a flute while maintaining the original musical expression. Specific examples of timbre transfer using generative models include Bitton et al. (2018) and Huang et al. (2019).

Timbre transfer has been explored a number of times using DDSP. Engel et al. (2020a)'s differentiable spectral modelling synthesiser and the associated f_0 and loudness control signals naturally lend themselves to the task, effectively providing a low dimensional representation of a musical performance while the timbre of a particular instrument is encoded in the network's weights. During inference, f_0 and loudness signals from any instrument can be used as inputs, in a many-to-one fashion. A similar task formulation was explored by Michelashvili and Wolf (2020), Carney et al. (2021), Hayes et al. (2021), and Caspe et al. (2022).

2.1.3 Performance rendering

Performance rendering systems seek to map from a symbolic musical representation to audio of that musical piece such that the musical attributes are not only correctly reflected, but expressive elements are also captured. Castellon et al. (2020); Jonason et al. (2020); Wu D.-Y. et al. (2022) augmented Engel et al.'s differentiable spectral-modelling synthesiser with a parameter generation frontend that received MIDI for performance rendering. Castellon et al. (2020) and Jonason et al. (2020) used recurrent neural networks to create mappings from MIDI to time-varying pitch and loudness controls. Wu D.-Y. et al. (2022) presented a hierarchical generative system to map first from MIDI to expressive

performance attributes (articulation, vibrato, timbre, etc.), and then to synthesis controls.

2.1.4 Sound matching

Synthesizer sound matching, also referred to as automatic synthesizer programming, aims to find synthesiser parameters that mostly closely match a target sound. Historical approaches include genetic algorithms (Horner et al., 1993), while deep learning has more recently gained popularity (Yee-King et al., 2018; Barkan et al., 2019).

Masuda and Saito (2021) proposed to approach sound matching with a differentiable audio synthesiser, in contrast to previous deep learning methods which used a parameter loss function. In later work (Masuda and Saito, 2023), they extended their differentiable synthesiser introduced a self-supervised training scheme blending parameter and audio losses.

2.2 Speech synthesis

Artificial generation of human speech has long fascinated researchers, with its inception tracing back to Dudley (1939) at Bell Telephone Laboratories, who introduced *The Vocoder*, a term now widely adopted (Dudley and Tarnoczy, 1950). Subsequent research has led to a substantial body of work on speech synthesis, driven by escalating demands for diverse and high-quality solutions across applications such as smartphone interfaces, translation devices, and screen readers (Tamamori et al., 2017).

Classical DSP-based methods for speech synthesis can be broadly split into three categories: articulatory (Shadle and Damper, 2001; Birkholz, 2013), source-filter/formant (Seeviour et al., 1976), and concatenative (Khan and Chitode, 2016). This aligns with the parametric and concatentivative distinction due to Schwarz (2007), discussed in section 2.1. Specifically, articulatory and source-filter/formant synthesis are parametric methods. Moreover, articulatory synthesis techniques are related to physical modeling approaches, as they strive to directly replicate physical movements in the human vocal tract. Unit-selection techniques (Hunt and Black, 1996), a subset of concatenative synthesis, use a database of audio segmented into speech units (e.g., phonemes) and an algorithm to sequence units to produce speech. Subsequent developments in machine learning gave rise to statistical parametric speech synthesis (SPSS) systems. Unlike unit selection, SPSS systems removed the need to retain speech audio for synthesis, focusing instead on developing models, such as hidden Markov models, to predict parameters for parametric speech synthesizers (Zen et al., 2009).

In this review, two subtasks of speech synthesis are particularly pertinent, namely, text-to-speech (TTS) and voice transformation. TTS involves converting text into speech, a process often synonymous with the term speech synthesis itself (Tan et al., 2021). Voice transformation, in contrast, focuses on altering roperties of an existing speech signal, including as voice identity and mood (Stylianou, 2009). A central component in both tasks is the vocoder, responsible for generating speech waveforms from acoustic features.

As neural audio synthesis became feasible, neural vocoders such as the autoregressive WaveNet van den Oord et al. (2016) quickly became state-of-the-art for audio quality. However, WaveNet's sequential generation was prohibitively costly, motivating the development of more efficient neural vocoders. Techniques included cached dilated convolutions (Ramachandran et al., 2017), optimized recurrent neural networks (Kalchbrenner et al., 2018), enabled parallel generation using flow-based models (Oord et al., 2018; Prenger et al., 2019), denoising diffusion probabilistic models (DDPMs) (Kong et al., 2021), and generative adversarial networks (GANs) (Kumar et al., 2019; Kong et al., 2020). GAN-based vocoders have become something of a workhorse in speech tasks, owing to their high quality and fast inference (Matsubara et al., 2022; Song et al., 2023).

2.2.1 DDSP-based vocoders

Computational efficiency is a central consideration in neural vocoders, as inference time is crucial in many application areas. Before explicitly DDSP-based models, researchers began integrating DSP knowledge into networks. Jin et al. (2018) made the connection between dilated convolutions and wavelet analysis, which involves iterative filtering and downsampling steps, and proposed a network structure based on the Cooley-Tukey Fast Fourier Transform (FFT) (Cooley and Tukey, 1965), capable of real-time synthesis. Similarly sub-band coding through pseudo-mirror quadrature filters (PQMF) was applied to enable greater parallelisation of WaveRNN-based models (Yu et al., 2020).

Later work saw the integration of models for speech production. The source-filter model of voice production has proven particularly fruitful—LPCNet (Valin and Skoglund, 2019) augmented WaveRNN (Kalchbrenner et al., 2018) with explicit linear prediction coefficient (Atal and Hanauer, 1971) calculation. This allowed a reduction in model complexity, enabling real-time inference on a single core of an Apple A8 mobile CPU. Further efficiency gains were made subsequently through multi-band linear prediction (Tian et al., 2020) and sample "bunching" (Vipperla et al., 2020). Subramani et al. (2022) later observed that the direct calculation of LPCs limited LPCNet to acoustic features for which explicit formulas were known. To alleviate this issue, they proposed to backpropagate gradients through LPCs, enabling their estimation by a neural network.

Despite improved efficiency, Juvela et al. (2019) noted that LPCNet's autoregression is nonetheless a bottleneck. To address this, they proposed GAN-Excited Linear Prediction (GELP) which produced the residual signal with a GAN signal, with explicit computation of LPCs from acoustic features, thus limiting autoregression to the synthesis filter and parallelising excitation.

LPCNet and GELP both incorporated DSP-based filters into a source-filter enhanced neural vocoder. Conversely, Wang et al. (2019b) proposed the *neural source filter* model which used DSP-based model for the excitation signal (i.e., harmonics plus noise) and a learned neural network filter. Through ablations, they demonstrated the benefit of using such sinusoidal excitation of the neural filters. Subsequent improvements to their neural source filter (NSF) method included the addition of a differentiable maximum voice frequency crossover filter (Wang and Yamagishi, 2019), and a quasi-periodic cyclic noise excitation (Wang and Yamagishi, 2020).

Combining these approaches, Mv and Ghosh (2020) and Liu et al. (2020) proposed to use differentiable implementations of DSP-based excitation and filtering, and parameterise these with a neural network. This allowed audio sample rate operations to be offloaded to efficient DSP implementations, while the parameter estimation networks could operate at frame level.

Eschewing the source-filter approach entirely, Kaneko et al. (2022) demonstrated that the last several layers in a neural vocoder such as HiFi-GAN (Kong et al., 2020) could be replaced with an inverse short-time Fourier transform (ISTFT). The number of replaced layers can be tuned to balance efficiency and generation quality. Webber et al. (2023) also used a differentiable ISTFT, reporting a real-time factor of over 100× for speech at 22.05 kHz on a high-end CPU. In contrast to a GAN, Webber et al. learned a compressed latent representation and by training a denoising autoencoder. Nonetheless, Watts et al. (2023) noted that neural vocoders which run in real-time on a CPU often rely on powerful CPUs, limiting use in low-resource environments. In particular, they highlight Alternative and Augmentative Communication (AAC) devices, which are used by people with speech and communication disabilities (Murray and Goldbart, 2009). Watts et al. proposed a method using the ISTFT and pitch-synchronous overlap add (PSOLA), with lightweight neural networks operating at rates below the audio sample rate.

Song et al. (2023) highlight that GAN vocoders struggle with periodicity, especially during prolonged vocalisations, and attribute this to unstable parallel generation. This is compounded by over-smoothing of acoustic features output by TTS or voice transformation models. They note that DSP vocoders can be more robust and less prone to pitch and phase errors, and thus propose to use a pre-trained neural homomorphic vocoder (Liu et al., 2020) to generate mel-spectrograms for a GAN vocoder, which they confirmed experimentally to improve generalisation to unseen speakers. Even in DSP-based models, however, design choices can influence robustness. Wang and Yamagishi (2020) observed that the choice of source signal interacts with the speaker's gender: sinusoidal excitation performs better for female voices than for male voices, while a cyclic noise signal improves performance on male voices. Additionally, they note the sine-based source signals may lead to artefacts during less periodic, expressive vocalisations such as creaky or breathy voices.

2.2.1.1 Control

DDSP-based methods can also facilitate control over speech synthesis. Fabbro et al. (2020) distinguish between two categories of method: those that necessitate control and those that offer optional control. The authors advocate for the latter, arguing that disentangling inputs into components—namely, pitch, loudness, rhythm, and timbre—provides greater flexibility, proposing a method that enables this. These disentangled factors are then utilized to drive a differentiable harmonic plus noise synthesizer, although the authors note that there is room for improvement in the quality of the synthesis.

Choi H.-S. et al. (2023) also decompose the voice into four aspects: pitch, amplitude, linguistic features, and timbre. They identify a that control parameters are often entangled in a mid-level representation or latent space, in existing neural vocoders, restricting control and limiting models' potential as co-creation tools. To address this, in contrast to the single network of Fabbro et al. (2020), they combine dedicated modules for disentangling controls. They evaluate their methodology through a range of downstream tasks, using a modified parallel WaveGAN model (Yamamoto et al., 2020) with sinusoidal and noise conditioning, along with timbre and linguistic embeddings. Reconstruction was found to be nearly identical in a copy synthesis task.

2.2.2 Text-to-speech synthesis

Text-to-speech (TTS) is the task of synthesising intelligible speech from text, and has received considerable attention due to its numerous commercial applications. Tan et al. (2021) provide a comprehensive review of the topic, including references to reviews on classical methods and historical perspectives. While most research on DDSP audio synthesis focuses on vocoding, some studies have also assessed its application in TTS systems (Juvela et al., 2019; Wang and Yamagishi, 2019; Liu et al., 2020; Choi H.-S. et al., 2023; Song et al., 2023).

2.2.3 Voice transformation

An application of speech synthesis systems is the ability to modify and transform the voice. Voice transformation is an umbrella term used to refer to a modification that is made to a speech signal that alters one or more aspects of the voice while keeping the linguistic content intact Stylianou (2009). Voice conversion (VC) is a subtask of voice transformation that seeks to modify a speech signal such that a utterance from a source speaker sounds like it was spoken by a target speaker. Voice conversion is a longstanding research task (Childers et al., 1985) that has continued to receive significant attention in recent years, demonstrated by the biannual voice conversion challenges operating in 2016, 2018, and 2020. An overview of the field is provided by (Mohammadi and Kain, 2017) and more recent applications of deep learning towards VC is reviewed by (Sisman et al., 2021).

Nercessian (2021) incorporated a differentiable harmonic-plus-noise synthesiser (Engel et al., 2020a) to a end-to-end VC model, augmenting it with convolutional pre- and post-nets to further shape the generated signal. This formulation allowed end-to-end training with perceptually informed loss functions, as opposed to requiring autoregression. Nercessian also argued that such "oscillator driven networks" are better equipped to produce coherent phase and follow pitch conditioning.

Choi H.-S. et al. (2023) explored zero-shot voice conversion with their NANSY++ model, which was facilitated by the distentangled intermediate representations. Their approach was to replace the timbre embedding with that of a target speaker, while also transforming pitch conditioning for the sinusoidal generator to match the target.

In voice designing or speaker generation (Stanton et al., 2022), the goal is to provide a method to modify certain characteristics of a speaker and generate a completely unique voice. Creation of new voice identifies has application for a number of downstream applications including audiobooks and speech-based assistants. Choi H.-S. et al. (2023) fit normalising flow models, conditioned on age and gender attributes, to generate synthesis control parameters for this purpose.

TABLE 2 Differentiable implementations of discrete time IIR filters with trainable parameters. Specific parameterisations, training algorithms, and stability constrains are presented. Recursive filter structure is also indicated where appropriate and when clear from the original manuscript.

Type	References	Representation	Parameters	Training algorithm	Stability constraints	Time varying
First Order	Kuznetsov et al. (2020)	Direct	b_0, b_1, a_1	TBPTT	$\lvert a_1\rvert < 1$	✗
	Bhattacharya et al. (2020)	Shelving filter	Freq. (f_c), gain (G)	IBPTT	—	✗
Second Order	Kuznetsov et al. (2020)	Direct	b_0, b_1, b_2, a_1, a_2	TBPTT (TDF-II)	$\lvert a_1\rvert \leq 0.5$ $\lvert a_2\rvert < 0.5$	✗
	Kuznetsov et al. (2020)	State-variable filter	Cutoff (g), damping (R), band gains (c_{LP}, c_{HP}, c_{BP})	TBPTT	$g = 1$ $R = \frac{1}{\sqrt{2}}$	✗
	Bhattacharya et al. (2020)	Peak	Freq. (f_c), bandwidth (f_b) gain (G)	IBPTT (DF-II)	—	✗
	Nercessian (2020)	Low/high shelf, peak	Freq. (ω_0), gain (A) Q-factor (q)	Freq. Sampling	—	✗
	Nercessian et al. (2021)	Direct	b_0, b_1, b_2, a_1, a_2	Freq. Sampling	$a_1 \leftarrow 2\tanh a_1$ $a_2 \leftarrow \frac{1}{2}(\lvert a_1\rvert + (2 - \lvert a_1\rvert)\tanh a_2)$	Via conditioning
		Pole/zero	$p, q \in \mathbb{C}$	Freq. Sampling	$p \leftarrow p \cdot \frac{\tanh\lvert p\rvert}{\lvert p\rvert}$	Via conditioning
		Low/high shelf, peak	Freq. (ω_0), gain (A) Q-factor (q)	Freq. Sampling	$Q \leftarrow \frac{Q_{\max}}{1+e^{Q}}$	Via conditioning
	Yu and Fazekas (2023)	Direct (all-pole)	a_1, a_2	TBPTT (DF-I)	$a_1 \leftarrow 2\tanh a_1$ $a_2 \leftarrow \frac{1}{2}(\lvert a_1\rvert + (2 - \lvert a_1\rvert)\tanh a_2)$	Framewise
	Carson et al. (2023)	All-pass	Break freq. (ω_b), thru gain (g_1), feedback gain (g_2)	Freq. Sampling	—	Framewise
Arbitrary order	Kuznetsov et al. (2020)	Linear state-space	Transition matrices ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$)	TBPTT	$\mathbf{A} \sim \mathcal{U}_{n\times n}(-\frac{1}{n}, \frac{1}{n})$	✗
	Mv and Ghosh (2020)	LPC	Reflection Coefficients (k_m)	TBPTT	$k_m \in (-1, 1)$	Framewise
	Subramani et al. (2022)	LPC	Reflection Coefficients (k_m)	Autoregressive	$k_m \in (-1, 1)$	Framewise

2.3 Singing voice synthesis

Singing voice synthesis (SVS) aims to generate realistic singing audio from a symbolic music representation and lyrics, a task that inherits challenges from both speech and musical instrument synthesis. The musical context demands an emphasis on pitch and timing accuracy (Saino et al., 2006), as well as ornamentation through dynamic pitch and loudness contours. Audio is typically expected at the higher resolutions, typical of musical recordings (i.e., 44.1 kHz CD quality vs. 16 kHz or 24 kHz as often used for speech), incurring additional computational complexity (Chen et al., 2020). Applications of SVS include performance rendering from scores, modifying or correcting existing performances, and recreating performances in the likeness of singers (Rodet, 2002).

SVS methods originated in the 1960s, evolving from speech synthesis systems, and early methods can be similarly coarsely categorised into *waveform* and *concatenative* techniques (Rodet, 2002). A historical perspective is provided by Cook (1996), and statistical methods were later introduced by Saino et al. (2006).

Early deep learning approaches to SVS included simple feed-forward networks (Nishimura et al., 2016) and a WaveNet-based autoregressive model Blaauw and Bonada (2017), which showed improvements over then state-of-the-art concatenative synthesisers. Deep learning-based SVS systems rely on large, annotated datasets for training, necessitated by the diverse vocal expressions in musical singing (Yu and Fazekas, 2023). The scarcity of such singing datasets was noted by Gómez et al. (2018) and Cho et al. (2021), contrasting with advances in TTS research predicated on open datasets like Hi-Fi TTS (Bakhturina et al., 2021) and LJSpeech (Ito and Johnson, 2017). The need for such data, including for specific vocal techniques like growls and rough voice (Gómez et al., 2018), motivated self-supervised systems incorporating DDSP like NANSY++ (Choi H.-S. et al., 2023), which supported high-quality resynthesis with a fraction of the OpenCPOP dataset (Wang et al., 2022). Wu D.-Y. et al. (2022) also found that the differentiable vocoder SawSing performed well in resource-limited training.

Gómez et al. (2018) and Cho et al. (2021) note that the black-box nature of deep learning limits analytical understanding of learned mappings and the ability to gain domain knowledge from trained

SVS systems. Gomez et al. acknowledged that this weakens the link between acoustics research and engineering, and foreshadowed DDSP-like innovation, hypothesising that "transparent" algorithms might restore it. Indeed, exploitability has motivated numerous DDSP-based SVS systems Yu and Fazekas (2023); Alonso and Erkut (2021); Nercessian (2023). Yu and Fazekas highlight the potential for their differentiable LPC method to be used for voice decomposition and analysis. Nercessian (2023) notes that the differentiable harmonic-plus-noise synthesiser of Engel et al. (2020a) has limited exploitability, and proposed as an alternative a differentiable implementation of the non-parametric WORLD feature analysis and synthesis model.

A further impetus for applying DDSP to SVS is audio quality, with two major challenges in neural vocoders being phase discontinuities and accurate pitch reconstruction. Reconstructing phase information from mel-spectrograms is difficult, and phase discontinuities can cause unnatural sound glitches and voice tremors (Wu D.-Y. et al., 2022). Differentiable oscillators, like the sawtooth oscillator proposed by Wu et al., address this by enforcing phase continuity.

Accurate pitch control and reconstruction are known challenges for neural vocoders (Hono et al., 2021). Yoshimura et al. (2023) argue that non-linear filtering operations in neural vocoders obscure the relationship between acoustic features and output, complicating accurate pitch control. They propose differentiable linear filters in a source-filter model to address this. Nercessian (2023) stress the importance of accurate pitch control and reconstruction for musical applications, a feature inherent to their differentiable WORLD synthesiser.

Computational efficiency is less emphasized in SVS literature compared to speech synthesis; however, GOLF Yu and Fazekas (2023) required less than 40% GPU memory during training and provided nearly 10x faster inference speed than other DDSP SVS methods (which already supported real-time operation). This fast inference speed can facilitate downstream applications, including real-time musicking contexts or functioning in low-resource embedded devices.

2.3.1 Singing voice conversion

The task of singing voice conversion (SVC) aims to transform a recording of a source singer such that it sounds like it was sung by a given target singer. This task, related to voice conversion, introduces further challenges (Huang et al., 2023). Specifically, there are a wider range of attributes to model, such as pitch contours, singing styles, and expressive variations. Further, perceived pitch and timing must adhere to the source material while incorporating stylistic elements from the target. Variations of this task include in-domain transfer, where target singing examples are available, and the more complex cross-domain transfer, where the target must be learned from speech samples (Huang et al., 2023) or from examples in a different language (Polyak et al., 2020).

The 2023 SVC Challenge (Huang et al., 2023) illustrated the applicability of DDSP methods for SVC. According to a subjective evaluation of naturalness, the top two performing models were both based on DSPGan (Song et al., 2023), which uses a pre-trained Neural Homomorphic Vocoder (Liu et al., 2020) to generate mel-spectrograms for resynthesis using a differentiable source-filter based model. However, the organisers noted that it would be premature to conclude that DSPGan is unilaterally the best SVC model, given the small sample size. Nonetheless, five other teams incorporated neural source-filter (Wang et al., 2019b) based components into HiFi-GAN (Kong et al., 2020) to improve generalisation, implying that incorporation of domain knowledge via DDSP offers some benefit.

Nercessian (2023) implemented a differentiable WORLD vocoder, which was also applied to SVC. They argue that this implementation, paired with a deterministic WORLD feature encoder and a learned decoder, offers increased control over the pitch contour while ensuring phase coherence–both of which are challenging for neural vocoders. Further, the interpretable feature representation and extraction procedure allows for direct manipulation of audio attributes, as well as pitch and loudness conditioned timbre transfer, as described by Engel et al. (2020a).

3 Differentiable digital signal processing

In this section we survey differentiable formulations of signal processing operations for audio synthesis. Whilst many were first introduced for other tasks, their relevance to audio synthesis is often clear as many correspond to components of classical synthesis algorithms. For this reason, we choose to include them here.

Our aim in this section is to provide an overview of the technical contributions that underpin DDSP in order to make clearer the connections between methods, as well as facilitate the identification of open directions for future research. We also hope that this section will act as a technical entry point for those wishing to work with DDSP. We thus do not intend to catalogue every *application* of a given method, but instead endeavour to acknowledge technical contributions and any prominent variations. Readers interested in further practical advice on implementing DDSP techniques for audio synthesis should refer to the accompanying web book, available at https://intro2ddsp.github.io/.

3.1 Filters

3.1.1 Infinite impulse response

A causal linear time-invariant (LTI) filter with impulse response $h(t)$ is said to have an infinite impulse response (IIR) if there does not exist a time T such that $h(t) = 0$ for all $t > T$. In the case of digital filters, this property arises when the filter's difference equation includes a nonzero coefficient for a previous output.

We present the dominant methods for differentiable IIR filtering in Table 2.

3.1.1.1 Recursive methods

Optimising IIR filter coefficients by gradient descent is not a new topic. Several algorithms for adaptive IIR filtering and online system identification rely on the computation of exact or approximate gradients (Shynk, 1989). Moreover, to facilitate the training of locally recurrent IIR multilayer perceptrons (IIR-MLP) (Back and Tsoi, 1991), approximations to backpropagation-through-time (BPTT) have been proposed (Campolucci et al., 1995). However, prior to widely available automatic differentiation, such methods required cumbersome manual gradient derivations, restricting the exploration of arbitrary filter parametrisations or topologies.

One such training algorithm, known as instantaneous backpropagation through time (IBPTT) Back and Tsoi (1991),[4] was applied by Bhattacharya et al. (2020) to a constrained parameterisation of IIR filters, namely, peak and shelving filters such as those commonly found in audio equalisers. This method was tested on a cascade of such filters and used to match the response of target head-related transfer functions (HRTFs). However, the formulation of IBPTT precludes the use of most modern audio loss functions, somewhat hindering the applicability of this method.

Kuznetsov et al. (2020) identified the close relationship between IIR filters and recurrent neural networks (RNNs). In particular, they illustrated that in the case of a simple RNN known as the *Elman network* (Elman, 1990), the two are equivalent when the activations of the Elman network are element-wise linear maps, and the bias vectors are zero. This is illustrated below:

$$\begin{aligned}\mathbf{h}[n] &= \sigma_h\left(\mathbf{W}_h\mathbf{h}[n-1] + \mathbf{U}_h\mathbf{x}[n] + b_h\right),\\ \mathbf{y}[n] &= \sigma_y\left(\mathbf{W}_y\mathbf{h}[n] + b_y\right).\end{aligned} \quad \begin{aligned}&\text{(Elman network)}\end{aligned} \qquad \begin{aligned}\mathbf{h}[n] &= \mathbf{W}_h\mathbf{h}[n-1] + \mathbf{U}_h\mathbf{x}[n],\\ \mathbf{y}[n] &= \mathbf{W}_y\mathbf{h}[n].\end{aligned} \quad \begin{aligned}&\text{(All}-\text{pole IIR filter)}\end{aligned} \tag{1}$$

Such RNNs are typically trained with backpropagation through time (BPTT), and more commonly its truncated variant (TBPTT) in which training sequences are split into shorter subsequences.[5] Based on this equivalence, Kuznetsov et al. directly applied TBPTT to IIR filters, effectively training various filter structures as linear recurrent networks. To ensure filter stability, they propose simple constraints on parameter initialisation.

A challenge with recursive optimisation is the necessity of memory allocations for each "unrolled" time step. For long sequences, this can result in poor performance and high memory cost. To address this and allow optimisation over high order IIR filters, Yu and Fazekas (2023) provided an efficient algorithm for applying BPTT to all-pole filters. Specifically, for an all-pole filter with coefficients $\mathbf{a} \in \mathbb{R}^M$, they showed that the partial derivatives $\frac{\partial y[n]}{\partial a_i}$ and $\frac{\partial \mathcal{L}}{\partial x[n]}$ can be expressed as applications of the same all-pole filter to $-y\,[n-i]$, and a filter with time-reversed coefficients to $\frac{\partial \mathcal{L}}{\partial y[M-n]}$, respectively. That is.

$$\begin{aligned}\frac{\partial y[n]}{\partial a_i} &= -y[n-i] - \sum_{k=1}^{M} a_k \frac{\partial y[n-k]}{\partial a_i} \;\Rightarrow\; \mathcal{Z}\left\{\frac{\partial y[n]}{\partial a_i}\right\}\\ &= \frac{1}{1+\sum_{k=1}^{M} a_k z^{-k}} \cdot Y(z) \cdot z^{-i},\end{aligned} \tag{2}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial x[n]} &= \frac{\partial \mathcal{L}}{\partial y[n]} - \sum_{k=1}^{M} a_k \frac{\partial \mathcal{L}}{\partial x[n+k]} \;\Rightarrow\; \mathcal{Z}\left\{\frac{\partial \mathcal{L}}{\partial x[n]}\right\}\\ &= \frac{1}{1+\sum_{k=0}^{M-1} a_{M-k} z^{-k}} \cdot \mathcal{Z}\left\{\frac{\partial \mathcal{L}}{\partial y[n]}\right\},\end{aligned} \tag{3}$$

where $\mathcal{Z}\{\cdot\}$ denotes the z-transform operator. It is clear from Eqs 2, 3 that these derivatives can be evaluated without the need for a computation graph of unrolled filter timesteps, enabling the use of efficient, recursive IIR implementations.[6].

3.1.1.2 Frequency sampling methods

It is common, when working with higher order filters, to factorise the transfer function into a cascade of second order sections in order to ensure numerical stability. It has been reported (Nercessian et al., 2021) that optimising differentiable cascades using BPTT/TBPTT limits the number of filters that can be practically learned in series, motivating an alternate algorithm for optimising the parameters of cascaded IIR filters.

One such approach, proposed by Nercessian (2020), circumvented the need for BPTT by defining a loss function in the spectral domain, with the desired filter magnitude response as the target. This method was used to train a neural network to match a target magnitude response using a cascade of differentiable parametric shelving and peak filters. To compute the frequency domain loss function, the underlying response of the filter must be sampled at some discrete set of complex frequencies, typically selected to be the Kth roots of unity $e^{j\frac{k}{K}}, \quad k = 0, 1, \ldots, K-1$.

This procedure is equivalent to the frequency sampling method for finite-impulse response filter design. That is, by sampling the filter's frequency response we are effectively optimising an FIR filter approximation to the underlying frequency response. Naturally, this sampling operation results in time-domain aliasing of the filter impulse response, and the choice of K thus represents a trade-off between accuracy and computational expense.

In subsequent work, Nercessian et al. (2021) extended this frequency sampling approach to *hyperconditioned* IIR filter cascades, to model an audio distortion effect. Hyperconditioning refers to a hypernetwork-like (Ha et al., 2017) structure in which the hypernetwork introduces conditioning information to the main model by generating its parameters. In this case, the hypernetwork's inputs are the user-facing controls of an audio effect, and the main model is a cascade of biquadratic IIR filters.

Whilst filter stability is less of a concern when training a model to produce fixed sets of filter coefficients, the hyperconditioned setting carries a greater risk. Both user error and erroneous model predictions may lead to a diverging impulse response at either inference or training time. For this reason, Nercessian et al. tested three parameterisations of cascaded biquads (coefficient, conjugate pole-zero, and parametric EQ representations), and proposed stability enforcing activations for each. Rather than directly optimising filter magnitude responses, the model was instead optimised using an audio reconstruction loss by applying the filters to an input signal. During optimisation, this was approximated by complex multiplication of the frequency-sampled filter response in the discrete Fourier domain.

Subsequently, this constrained frequency sampling method was applied to the more general task of IIR filter design (Colonel et al., 2022). Here, a neural network was trained to parameterise a differentiable biquad cascade to match a given input magnitude response, using synthetic data sampled from the coefficient space of

4 Note that the algorithm proposed by Back and Tsoi (1991) was not referred to as IBPTT in the original work. This name was given later by Campolucci et al. (1995).

5 Note that the truncation in TBPTT, which is applied to input sequences, is distinct from that in CBPTT, which is applied to intermediate backpropagated errors.

6 This algorithm has been implemented by Yu and Fazekas (2023) and is available in the open source TorchAudio package. https://github.com/pytorch/audio. Accessed 21st July 2023.

the differentiable filter. However, the authors do note that training a model with higher order filters ($N \geq 64$) tended to lead to instability, suggesting that even the frequency sampling approach may be insufficient to solve the aforementioned challenges with cascade depth (Nercessian et al., 2021).

Diaz et al. (2023) also applied the constrained frequency sampling method to hybrid parallel-cascade filter network structures, for the purpose of generating resonant filterbanks which match the modal frequencies of rigid objects. In comparing filter structures, authors found that the best performance was achieved with a greater number of shallower cascades. Paired with the insight from Colonel et al. (2022), this suggests that further research is necessary into the stability and convergence of the frequency sampling method under different filter network structures, as well as the effect of saturating nonlinearities on filter coefficients.

Some applications call for all-pass filters–i.e., filters which leave frequency magnitudes unchanged, but which do alter their phases. Carson et al. (2023) optimised all-pass filter cascades directly using an audio reconstruction loss. Due to the difficulty of optimising time-varying filter coefficients, the time-varying phaser filter is piecewise approximated by M framewise time-invariant transfer functions $H^{(m)}(z)$, which are predicted and applied to windowed segments of the input signal. Overlapping windows are then summed such that the constant overlap-add property (COLA) is satisfied.

In the case where an exact sinusoidal decomposition of the filter's input signal is known, such as in a synthesiser with additive oscillators, the filter's transfer function can be sampled at exactly the component frequencies of the input. This is applied by Masuda and Saito (2021) to the task of sound-matching with a differentiable subtractive synthesiser.

3.1.1.3 Source-filter models and linear predictive coding

The source-filter model of speech production describes speech in terms of a source signal, produced by the glottis, which is filtered by the vocal tract and lip radiation. This is frequently approximated as the product of three LTI systems, i.e., $Y(z) = G(z)H(z)L(z)$ where $G(z)$ describes the glottal source, $H(z)$ describes the response of the vocal tract, and $L(z)$ is the lip radiation. Often, $L(z)$ is assumed to be a first order differentiator, i.e., $L(z) = 1 - z^{-1}$, and the glottal flow derivative $G'(z) = G(z)L(z)$ is directly modelled.

Frequently, a local approximation to the time-varying filter $H(z)$ is obtained via linear prediction over a finite time window of length M:

$$y[n] = e[n] + \sum_{m=1}^{M} a_m y[n-m], \tag{4}$$

where $e[n]$ describes the excitation signal, which is equivalent to the linear prediction residual. The coefficients a_m are exactly the coefficients of an Mth order all-pole filter. This representation is known as linear predictive coding (LPC).

To the best of our knowledge, Juvela et al. (2019) were the first to incorporate a differentiable synthesis filter into a neural network training pipeline. In their method, a GAN was trained to generate excitation signals $e[n]$, while the synthesis filter coefficients were directly estimated from the acoustic feature input, i.e., non-differentiably. Specifically, given a mel-spectrum input, $\mathbf{m} = \log(\mathbf{M}\hat{\mathbf{x}})$, where $\mathbf{M}$ is the mel-frequency filterbank and $\hat{\mathbf{x}}$ is the discrete Fourier transform of a signal window, synthesis filter coefficients a_m are estimated by approximating the discrete Fourier spectrum, $\hat{\mathbf{x}} \approx \max(\mathbf{M}^{\dagger}\exp(\mathbf{m}), \epsilon)$, from which the autocorrelation function is computed and the normal equations are solved for a_m. In order to backpropagate error gradients through the synthesis filter to the excitation generator, the filter is approximated for training by truncating its impulse response, which is equivalent to the frequency sampling method discussed in Section 3.1.1.2.

LPC has also been used to augment autoregressive neural audio synthesis models. In LPCNet (Valin and Skoglund, 2019), linear prediction coefficients were non-differentiably computed from the input acoustic features, while the sample-level neural network predicted the excitation. However, in subsequent work (Mv and Ghosh, 2020; Subramani et al., 2022) this estimation procedure was made differentiable, allowing synthesis filter coefficients to also be directly predicted by a neural network. As LPC coefficients can easily be unstable, both Subramani et al. (2022) and Mv and Ghosh (2020) opted not to directly predict the LPC coefficients a_m, but instead described the system in terms of its *reflection coefficients* k_i. The fully differentiable Levinson recursion then allowed recovery of the coefficients a_m. Specifically, where $a_m^{(i)}$ denotes the mth LPC coefficient in a filter of order i (i.e., $M = i$), the recursion is defined:

$$a_m^{(i)} = \begin{cases} k_m & \text{if } m = i \\ a_m^{(i-1)} + k_i a_{i-m}^{(i-1)} & \text{otherwise.} \end{cases} \tag{5}$$

When $k_i \in (-1, 1)$ filter stability is thus guaranteed. This is applied by Subramani et al. (2022) in an end-to-end differentiable extension of the LPCNet architecture, again with autoregressive prediction of the excitation signal. Conversely, Mv and Ghosh (2020) employ a notably simpler source model, consisting of a mixture of an impulse train and filtered noise signal. Synthesis filter coefficients were also estimated differentiably by Yoshimura et al. (2023), who used the truncated Maclaurin series of a mel-cepstral synthesis filter to enable FIR approximation.

An alternative approach to modelling the glottal source is offered by Yu and Fazekas (2023), who use a one-parameter (R_d) formulation of the Liljencrants-Fant (LF) model of the glottal flow derivative $G'(z)$. The continuous time glottal source is sampled in both time and the R_d dimension to create a two-dimensional wavetable. To implement differentiable time-varying LPC filtering, the authors opt to use a locally stationary approximation and produce the full signal by overlap-add resynthesis. Rather than indirect paramaterisation of the filter via reflection coefficients, the synthesis filter is factored to second order sections. The coefficient representation and accompanying constraint to the biquad triangle proposed by Nercessian et al. (2021) is then used to enforce stability.

Südholt et al. (2023) apply a differentiable source-filter based model of speech production to the inverse task of recovering articulatory features from a reference recording. To estimate articulatory parameters of speech production, Südholt et al. use a differentiable implementation of the Pink Trombone,[7] which

7 https://dood.al/pinktrombone. Accessed 3rd August 2023.

approximates the geometry of the vocal tract as a series of cylindrical segments of varying cross-sectional area—i.e., the Kelly-Lochbaum vocal tract model. Instead of independently defining segment areas, these are parameterised by tongue position t_p, tongue diameter t_d, and some number of optional constrictions of the vocal tract. To estimate these parameters, Südholt et al. derive the waveguide transfer function and perform gradient descent using the mean squared error between the log-scaled magnitude response and the vocal tract response estimated by inverse filtering.

3.1.2 Finite impulse response

A LTI filter is said to have a finite impulse response (FIR), if given impulse response $h(t)$ there exists a value of T such that $h(t) = 0$ for all $t > T$. In discrete time, this is equivalent to a filter's difference equation being expressible as a sum of past inputs. Due to the lack of recursion, FIR filters guarantee stability and are less susceptible to issues caused by the accumulation of numerical error. However, this typically comes at the expense of ripple artifacts in the frequency response, or an increase in computational cost to reduce these artifacts.

As with IIR filters, the discrete time FIR filter is equivalent to a common building block of deep neural networks, namely, the *convolutional layer*[8]. Again, linear activations and zero bias yield the exact filter formulation:

$$\underbrace{\mathbf{y}[n] = \sigma((\mathbf{W}^{\star}\mathbf{h})[n] + \mathbf{b})}_{\text{(Convolutional layer)}}, \qquad \underbrace{\mathbf{y}[n] = (\mathbf{W}^{\star}\mathbf{h})[n]}_{\text{(FIR filter)}}. \tag{6}$$

In practice, we frequently wish to produce a time-varying frequency response in order to model the temporal dynamics of a signal. The stability guarantees and ease of parallel evaluation offered by FIR filters mean they are an appropriate choice for meeting these constraints in a deep learning context, but care must be taken to compensate for issues such as spectral leakage and phase distortion using filter design techniques. Moreover, such compensation must be achieved differentiably.

To the best of our knowledge, the first such example of differentiable FIR filter design was proposed by Wang and Yamagishi (2019). This work applied a pair of differentiable high- and low-pass FIR filters, with a time-varying cutoff frequency $f_c\,[m]$ predicted by a neural network conditioning module.[9] The filters are then implemented as windowed sinc filters, with frame-wise impulse responses $\tilde{h}_{\mathrm{LP}}[m,n]$ and $\tilde{h}_{\mathrm{HP}}[m,n]$.

Whilst Wang et al. manually derive an the loss gradient with respect to $f_c\,[m]$, their implementation relies on automatic differentiation.

While closed form parameterisation of FIR filter families allows for relatively straightforward differentiable filter implementations, the complexity of the resulting frequency responses is limited. However, filter design methods such as frequency sampling and least squares allow for FIR approximations to arbitrary responses to be created. The first differentiable implementation of such a method was proposed by Engel et al. (2020a), whose differentiable spectral-modelling synthesiser contained a framewise time-varying FIR filter applied to a white noise signal. To avoid phase distortion and suppress frequency response ripple, Engel et al. proposed a differentiable implementation of the window design method for linear phase filter design.

Specifically, a frequency sampled framewise magnitude response $\hat{\mathbf{h}}[m] \in \mathbb{R}^N$ is output by the decoder, where m denotes the frame index and N is the length of the impulse response.[10] To recover the framewise symmetric impulse responses $\mathbf{h}[m] \in \mathbb{R}^N$, they take the inverse discrete Fourier transform, $\mathbf{h}[m] = \mathbf{W}_N^H \hat{\mathbf{h}}[m]$), for N-point DFT matrix $\mathbf{W}_N$. The symmetry of the impulse responses is a sufficient condition for a linear phase response as a function of frequency. To mitigate spectral leakage, a window function $\mathbf{w} \in \mathbb{R}^N$ is applied to each symmetric impulse response. Finally, the filter is shifted to causal form such that it is centred at $\frac{N}{2}$ for a window length of N samples. The filters are then applied to the signal by circular convolution, achieved via complex multiplication in the frequency domain.

An alternative differentiable FIR parameterisation was suggested by Liu et al. (2020), who adopted complex cepstra as an internal representation. This representation jointly describes the filter's magnitude response and group delay, resulting in a mixed phase filter response. The authors note group delay exerts an influence over speech timbre, motivating such a design. However, the performance of this method is not directly compared to the linear phase method described above. The approximate framewise impulse response $\mathbf{h}[m]$ is recovered from the complex cepstrum $\hat{\mathbf{h}}[m]$ as follows:

$$\mathbf{h}[m] = \mathbf{W}_N^H \exp\left(\mathbf{W}_N \hat{\mathbf{h}}[m]\right) \tag{7}$$

where the exponential function is applied element-wise.

Barahona-Ríos and Collins (2023) applied an FIR filterbank to a white noise source for sound effect synthesis, but circumvented the need to implement the filters differentiably by pre-computing the filtered noise signal and predicting band gains.

3.2 Additive synthesis

Many audio signals contain prominent oscillatory components as a direct consequence of the physical tendency of objects to produce periodic vibrations (Smith, 2010). A natural choice for modelling such signals, additive synthesis, thus also encodes such a preference for oscillation. Motivated by the signal processing interpretation of Fourier's theorem, i.e., that a signal can be decomposed into sinusoidal components, additive synthesis describes a signal as a finite sum of sinusoidal components. Unlike representation in the discrete Fourier basis, however, the

8 Note that the convolution operation in neural networks is usually implemented as a cross-correlation operation. This is equivalent to convolution with a kernel reversed along the dimension(s) of convolution. Hence, we use these terms interchangeably to aid legibility.

9 The specific parameterisations of f_c proposed by Wang and Yamagishi (2019) combine neural network predictions with domain knowledge about the behaviour of the maximum voice frequency during voiced and unvoiced signal regions. We omit these details here to focus on the differentiable filter implementation.

10 Note that here we adopt vector notation for the framewise impulse response (i.e., $\hat{\mathbf{h}}[m]_n = \hat{h}[m,n]$) to allow simplify the representation of operations.

frequency axis is not necessarily discretised, allowing for direct specification of component frequencies. The general form for such a model in discrete time is thus:

$$y[n] = \sum_{k=1}^{K} \alpha_k[n] \sin\left(\phi_k + \sum_{m=0}^{n} \omega_k[m]\right), \tag{8}$$

where K is the number of sinusoidal components, $\alpha[n] \in \mathbb{R}^K$ is a time series of component amplitudes, $\phi \in \mathbb{R}^K$ is the component-wise initial phase, and $\omega[n]$ is the time series of instantaneous component frequencies. Often, parameters are somehow constrained or jointly parameterised, such as in the harmonic model where $\omega_k[n] = k\omega_0[n]$ for some fundamental frequency $\omega_0[n]$.

A prominent extension of additive synthesis is *spectral modelling synthesis* (Serra and Smith, 1990). In this approach, the *residual* signal (i.e., the portion of the signal remaining after estimating sinusoidal model coefficients) is modelled stochastically, as a noise source processed by a time varying filter. Such a model was implemented differentiably by Engel et al. (2020a), using a bank of harmonic oscillators combined with a LTV-FIR filter applied to a noise source. Specifically, the oscillator bank was parameterised as follows:

$$y[n] = A[n] \sum_{k=1}^{K} \hat{\alpha}_k[n] \sin\left(\sum_{m=0}^{n} k\omega_0[m]\right), \tag{9}$$

where $A[n]$ is a global amplitude parameter, and $\hat{\alpha}[n]$ is a normalised distribution over component amplitudes (i.e., $\sum_k \hat{\alpha}_k[n] = 1$ and $\hat{\alpha}_k[n] \geq 0$). In practice, $A[n]$ and $\hat{\alpha}[n]$ are predicted at lower sample rates, and interpolated before evaluating the final signal. The fundamental frequency $\omega_0[n]$ is obtained by means of a pitch estimation algorithm–in the paper, CREPE (Kim et al., 2018) is used–rather than by direct optimisation. It is thus interesting to note that Eq. (9) is linear with respect to parameters $A[n]$ and $\hat{\alpha}[n]$, and thus admits a convex optimisation problem for an appropriate choice of loss function. This is, however, not the case with respect to $\omega_0[n]$, which yields oscillatory gradients.

3.2.1 Wavetable synthesis

Historically, a major obstacle to the adoption of additive synthesis was simply the computational cost of evaluating potentially hundreds of sinusoidal components at every time step. A practical solution was to pre-compute the values of a sinusoid at a finite number of time-steps and store them in a memory buffer. This buffer can then be read periodically at varying "frequencies" by fractionally incrementing a circular read pointer and applying interpolation. The buffer, referred to as a *wavetable*, need not contain only samples from a sinusoidal function, however. Instead, it can contain any values, allowing for the specification of arbitrary harmonic spectra (excluding the effect of interpolation error) when the wavetable is read periodically. Wavetables can thus allow for efficient synthesis of a finite number of predetermined spectra, or even continuous interpolation between spectra through interpolation both between and within wavetables.

Shan et al. (2022) introduced a differentiable implementation of wavetable synthesis, drawing comparison to a dictionary learning task. In particular, their proposed model learns a finite collection of wavetables $D = \{w_k[n]\}_{k\in\{1\ldots K\}}$. The fractional read position of the wavetables is determined by integrating the ω_0 parameter which, as in the harmonic oscillator bank of Engel et al. (2020a), is provided by a separate model. To produce a particular timbre, the model predicts coefficients c_k for a weighted sum over the wavetables, such that $\sum_{k=1}^{K} c_k = 1$ and $c_k \geq 0$.

Notably, Shan et al. found that this approach outperformed the DDSP additive model in terms of reconstruction error when using 20 wavetables, and performed almost equivalently with only 10 wavetables. Further, this method provided a roughly 12× improvement in inference speed, although as the authors note this is likely to be related to the 10-fold decrease in the number of parameter sequences that require interpolation.

3.2.2 Unconstrained additive models

The implementations discussed thus far are all, in a sense, *constrained* additive models. This is because a harmonic relationship between sinusoidal component frequencies is enforced. By contrast, the general model form illustrated in Eq. 8 is *unconstrained*, which is to say that its frequency parameters are independently specified. In some circumstances, the greater freedom offered by such a model may be advantageous–for example, many natural signals contain a degree of inharmonicity due to the geometry or stiffness of the resonating object. However, vastly fewer examples of differentiable unconstrained models exist in the literature than of their constrained counterparts.

Nonetheless, Engel et al. (2020b) introduced a differentiable unconstrained additive synthesiser, which they applied to the task of monophonic pitch estimation in an analysis-by-synthesis framework. The differentiable unconstrained model follows the form in Eq. 8, with the omission of the initial phase parameter. Thus, unlike their differentiable harmonic model, there is no factorised global amplitude parameter–instead each sinusoidal component is individually described by its amplitude envelope $\alpha_k[n]$ and frequency envelope $\omega_k[n]$.

Optimisation of this model, however, is not as straightforward as the constrained harmonic case where an estimate of $\omega_0[n]$ is provided *a priori*. The non-convexity of the sinusoidal oscillators with respect to their frequency parameters leads to a challenging optimisation problem, which does not appear to be directly solvable by straightforward gradient descent over an audio loss function. This was experimentally explored by Turian and Henry (2020) who found that, even in the single sinusoidal case, gradients for most loss functions were uninformative with respect to the ground truth frequency. Thus, Engel et al. incorporated a self-supervised pre-training stage into their optimisation procedure. Specifically, a dataset of synthetic signals from a harmonic-plus-noise synthesiser was generated, for which exact sinusoidal model parameters were know. This was used to pretrain the sinusoidal parameter encoder network with a parameter regression loss (discussed further in Section 4.2), which circumvents the non-convexity issue.

Subsequently, Hayes et al. (2023) proposed an alternate formulation of the unconstrained differentiable sinusoidal model, which aimed to circumvent the issue of non-convexity with respect to the frequency parameters. Specifically, they replaced the sinusoidal function with a complex surrogate, which produced a damped sinusoid:

$$y[n] = \sum_{k=1}^{K} \mathfrak{Re}\{z_k^n\} = \sum_{k=1}^{K} |z_k|^n \cos n \angle z_k, \tag{10}$$

where $\mathfrak{Re}$ denotes the real part of a complex variable, $|z|$ denotes the complex modulus, $\angle z$ denotes the complex phase, and z_k are the complex parameters of the model which jointly encode both frequency and damping. Note that as the sample index n becomes the exponential parameter, this model does not directly accommodate time-varying frequency parameters. However, at the time of writing, no published work exists applying this surrogate to enable differentiable unconstrained additive synthesis.

3.3 Subtractive synthesis

In contrast with additive methods, subtractive synthesis describes a family of methods in which a source signal, rich in frequency components, is shaped into a desired frequency response by the removal or attenuation of certain frequencies (Bilbao, 2009). This approach typically employs filters to shape the sound by attenuating or emphasizing specific frequency components. The task of generating a sound is then reduced to designing appropriate filters. The reader might note that this approach bears a striking similarity to the source-filter model of speech production, in which a spectrally rich signal is shaped by a series of filters. However, the source-filter models discussed so far are physically motivated by a tube model of the vocal tract, whereas subtractive synthesis refers to the more general class of methods that involve spectral attenuation of a source signal.

Sawtooth and square waveforms are commonly used as source signals in subtractive synthesis due to their dense harmonic spectra, with the former containing energy at all harmonic frequencies, and the latter only at odd harmonic frequencies. The true waveforms contain discontinuities which can lead to aliasing, and also pose a challenge for automatic differentiation. For this reason, bandlimited waveforms were produced by Masuda and Saito (2021) using a constrained version of the differentiable harmonic oscillator bank for the purpose of subtractive synthesiser sound matching. Specifically, they produced anti-aliased approximations of square and sawtooth waveforms by summing harmonics at pre-determined amplitudes.

$$y_{\text{saw}}[n] = \sum_{k=1}^{K} \frac{2}{\pi k} \sin\left(2\pi k \sum_{m=0}^{n} \omega_0[m]\right) \tag{11}$$

$$y_{\text{square}}[n] = \sum_{k=1}^{K} \frac{4}{\pi(2k-1)} \sin\left(2\pi(2k-1) \sum_{m=0}^{n} \omega_0[m]\right). \tag{12}$$

Masuda et al. also introduced a waveform interpolation parameter p, such that $y[n] = py_{\text{saw}}[n] + (1-p)y_{\text{square}}[n]$, allowing for differentiable transformation between these fixed spectra. The resulting waveform mixture was then filtered by the method described in Section 3.1.1.2.

A similar bandlimited sawtooth signal was used as a source in SawSing (Wu D.-Y. et al., 2022), a differentiable subtractive singing voice vocoder. This signal was shaped, however, by a linear time-varying FIR filter, similar to the one used with white noise by Engel et al. (2020a).

The differentiable WORLD vocoder (Nercessian, 2023) uses a bandlimited pulse-train as its source signal, implemented similarly to the aforementioned square and sawtooth oscillators. This is also filtered framewise by frequency domain multiplication with a filter response directly derived from the WORLD feature representation, consisting of an aperiodicity ratio and a spectral envelope.

3.4 Non-linear transformations

The techniques discussed to this point are predominantly linear with respect to the parameters through which gradients are backpropagated. Through the 60s, 70s and 80s a number of digital synthesis techniques emerged which involved non-linear transformations of audio signals and/or synthesiser parameters. This approach typically results in the introduction of further frequency components, with frequencies and magnitudes determined by the specifics of the method. As such, these methods are commonly collectively known as *distortion* or *non-linear* synthesis, and they became popular as alternatives to additive and subtractive approaches due to the comparative efficiency with which they could produce varied and complex spectra (Bilbao, 2009).

3.4.1 Waveshaping synthesis

Digital waveshaping synthesis (Le Brun, 1979) introduces frequency components through amplitude distortion of an audio signal. Specifically, for a continuous time signal that can be exactly expressed as a sum of stationary sinusoids, i.e., $x(t) = \sum_{k=1}^{K} \alpha_k \sin f_k t$, the application of a nonlinear function σ produces a signal that can be expressed as a potentially infinite sum of sinusoids at linear combinations of the input frequencies.

In the original formulation of waveshaping synthesis, proposed by Le Brun (1979), the input signal is a single sinusoid. The nonlinear function σ, also referred to as the shaping function, is specified as a sum of Chebyshev polynomials of the first kind T_k, allowing. These functions are defined such that the kth polynomial transforms a sinusoid to its kth harmonic: $T_k(\cos\theta) = \cos k\theta$. In this way, a shaping function can be easily designed that produces any desired combination of harmonics. Further efficiency gains can be achieved by storing this function in a memory buffer and applying it to a sinusoidal input by interpolated fractional lookups, in a manner similar to wavetable synthesis. Timbral variations can be produced by altering the amplitude of the incoming signal and applying a compensatory normalisation after the shaping function.

A differentiable waveshaping synthesiser was proposed by Hayes et al. (2021). This approach replaced the Chebyshev polynomial method for shaping function design with a small multilayer perceptron σ_θ. To allow time-varing control over timbral content, a separate network predicted affine transform parameters α_N, β_N, $\alpha_a\beta_a$, applied to the signal before and after the shaper network, giving the formulation:

$$y[n] = \alpha_N \sigma_\theta(\alpha_a x[n] + \beta_a) + \beta_N. \tag{13}$$

In practice, the network is trained with a bank of multiple such waveshapers.

3.4.1.1 Neural source-filter models

As noted by Engel et al. (2020a), the neural source-filter model introduced by Wang et al. (2019b) can be considered a form of waveshaping synthesiser. This family of models, based on the classical source-filter model, replaces the linear synthesis filter $H(z)$ with a neural network f_θ, which takes a source signal $x\,[n]$ as input, and also accepts a conditioning signal $\mathbf{z}\,[n]$ which alters its response. Due to the nonlinearity of f_θ, we can interpret its behaviour through the lens of waveshaping–that is, it is able to introduce new frequency components to the signal. Thus, for a purely harmonic source signal, a harmonic output will be generated, excluding the effects of aliasing.

The proposed neural filter block follows a similar architecture to a non-autoregressive WaveNet (van den Oord et al., 2016), featuring dilated convolutions, gated activations, and residual connections. Wang et al. experimented with both a mixed sinusoid-plus-noise source signal processed by a single neural filter pathway, and separate harmonic sinusoidal and noise signals processed by individual neural filter pathways. Later extensions of the technique included a hierarchical neural source-filter model, operating at increasing resolutions, for musical timer synthesis Michelashvili and Wolf (2020), and the introduction of quasi-periodic cyclic noise as an excitation source, allowing for more realistic voiced-unvoiced transitions (Wang and Yamagishi, 2020).

3.4.2 Frequency modulation

Frequency modulation (FM) synthesis (Chowning, 1973) produces rich spectra with complex timbral evolutions from a small number of simple oscillators. Its parameters allow continuous transformation between entirely harmonic and discordantly inharmonic spectra. It was applied in numerous commercially successful synthesisers in the 1980s, resulting in its widespread use in popular and electronic music.

A simple stationary formulation, consisting of one *carrier* oscillator (α_c, ω_c, ϕ_c) and one *modulator* (α_m, ω_m, ϕ_m) is given by:

$$y[n] = \alpha_c \sin\left(\omega_c n + \phi_c + \alpha_m \sin\left(\omega_m n + \phi_m\right)\right). \tag{14}$$

For a sinusoidal modulator, FM synthesis is equivalent to *phase modulation* up to a phase shift. Phase modulation is often preferred in practice, as it does not require integration of the modulation signal to produce an instantaneous phase value.

More complex synthesisers can be constructed by connecting multiple modulators and carriers, forming a modulation graph.[11] This graph may contain cycles, corresponding to oscillator feedback, which can be implemented in discrete time with single sample delays. Caspe et al. (2022) published the first differentiable FM synthesiser implementation, which was able to learn to control the modulation indices of modulation graphs taken from the Yamaha DX7, an influential FM synthesiser. To increase the flexibility of the DDX7 approach, Ye et al. (2023) applied a neural architecture search algorithm over the modulation graph, with the intention of allowing optimal FM synthesiser structures to be inferred from target audio.

As the gradients of an FM synthesiser's output with respect to the majority of its parameters are oscillatory, optimisation of a differentiable implementation is challenging, due to the aforementioned issues with sinusoidal parameter gradients (Turian and Henry, 2020; Hayes et al., 2023). For this reason, the differentiable DX7 of Caspe et al. (2022) relied on fixed oscillator tuning ratios, specified *a priori*, enforcing a harmonic spectral structure. In this sense, FM synthesis continues to be an open challenge, representing a particularly difficult manifestation of the previously discussed issues with non-convex DDSP.

3.5 Modulation signals

Automatic modulation of parameters over time is a crucial component of many modern audio synthesisers, especially those used for music and sound design. Control signals for modulation are commonly realised through *envelopes*, which typically take the value of a parametric function of time in response to a trigger event, and *low frequency oscillators* (LFOs), which oscillate at sub-audible (<20Hz) frequencies. Estimating the parameters of envelopes and LFOs is thus valuable for sound matching tasks. This was addressed non-differentiably by Mitcheltree et al. (2023), who estimated LFO shapes from mel spectrograms.

In the context of modelling an analog phaser pedal, Carson et al. (2023) produced a differentiable LFO model using the complex sinusoidal surrogate method of Hayes et al. (2023) in combination with a small waveshaper neural network. Using this technique, they were able to directly approximate the shape of the LFO acting on the all-pass filter break frequency by gradient descent.

Masuda and Saito (2023) introduced a differentiable attack-decay-sustain-release (ADSR) envelope[12] for use in a sound matching task. Specifically, they defined the envelope as follows:

$$u(t) = \left| t \cdot \frac{v_{\max}}{t_{\text{atk}}} \right|_{[0, v_{\max}]} + \left| (t - t_{\text{atk}}) \cdot \frac{v_{\text{sus}} - v_{\max}}{t_{\text{dec}}} \right|_{[v_{\text{sus}} - v_{\max}, 0]} + \left| -(t - t_{\text{off}}) \cdot \frac{v_{\text{sus}}}{t_{\text{rel}}} \right|_{[-v_{\text{sus}}, 0]}, \tag{15}$$

where t_{atk}, t_{dec}, t_{rel}, t_{off} represent the attack, decay, release, and note off times, respectively; v_{sus} is the sustain amplitude and $v_{\max}$ is the peak amplitude; and $|x|_{[a,b]} \triangleq \min(\max(x, a), b)$. Through sound matching experiments and gradient visualisations, they demonstrate that their model is capable of learning to predict parameters for the differentiable envelope, though note that this constraint means that subtle variations in real sounds can not be entirely captured.

4 Loss functions and training objectives

A central benefit of DDSP is that the training loss function can be defined directly in the audio domain, allowing the design of losses which emphasise certain desirable signal characters, for example, through phase invariance or perceptually-informed weighting of

11 This graph is sometimes referred to in commercial synthesisers as an "algorithm".

12 Far from being an arbitrary choice, this is perhaps the most commonly used parametric envelope generator in synthesisers.

frequency bands. Nonetheless, many works we reviewed combined audio losses with other forms of training objective, including parameter regression and adversarial training. In this section, we review the most commonly used such loss functions.

We note that deep feature loss, sometimes referred to as "perceptual loss" — that is, distances between intermediate activations of pretrained neural networks—have been explored, including the use of CREPE (Kim et al., 2018) embeddings (Engel et al., 2020a; Michelashvili and Wolf, 2020). Engel et al. note that this loss during unsupervised learning of pitch. Additionally, Turian et al. (2021) evaluated a number of audio representations (DSP-based and learned representations), comparing them on distance-based ranking tasks using synthesised sounds, and found that OpenL3 (Cramer et al., 2019) performed well. However, Masuda and Saito (2023) remarked that preliminary results using OpenL3 for sound matching worked poorly. Since this initial work on the topic, relatively little attention has been devoted to exploring such distances for DDSP training; however, they may be a promising direction for future work.

4.1 Audio loss functions

Audio loss functions compare a predicted audio signal $\hat{y}[n]$ to a ground truth signal $y[n]$. The simplest such loss function is thus a direct distance between audio samples in the time domain:

$$\mathcal{L}_{\text{wav}}(\mathbf{y},\hat{\mathbf{y}}) = \sum_{n} \left\| y[n] - \hat{y}[n] \right\|_p \tag{16}$$

where $\|\cdot\|_p$ is the L^p norm. Engel et al. (2020a) note that this loss is typically not ideal due to the weak correspondence between individual time-domain audio samples and auditory perception. For example, a time-domain loss penalises imperceptible shifts in oscillator phase, which may not be desirable, depending on the behaviour of a particular synthesiser and target application (Engel et al., 2020a; Liu et al., 2020; Mv and Ghosh, 2020). Wang and Yamagishi (2019) applied a phase difference loss, but noted that despite the loss values falling during training, speech quality was not improved over a randomly initialized phase spectrum. Conversely, Webber et al. (2023) explicitly modelled phase in the frequency domain, finding that an L^2 time-domain loss helped reduce audible artifacts.

The predominant approach to formulating an audio loss for DDSP tasks, however, is based on magnitude spectrograms. These approaches are often referred to as *spectral loss*. While numerous variations on this approach, we found three to recur commonly in the literature.

1. Spectral convergence loss (Arık et al., 2019)

$$\mathcal{L}_{\text{sc}}(\mathbf{y},\hat{\mathbf{y}}) = \frac{\left\| |STFT(\mathbf{y})| - |STFT(\hat{\mathbf{y}})| \right\|_F}{\left\| |STFT(\mathbf{y})| \right\|_F} \tag{17}$$

2. Log magnitude spectral loss (Arık et al., 2019)

$$\mathcal{L}_{\log}(\mathbf{y},\hat{\mathbf{y}}) = \left\| \log|STFT(\mathbf{y})| - \log|STFT(\hat{\mathbf{y}})| \right\|_1 \tag{18}$$

3. Linear magnitude spectral loss

$$\mathcal{L}_{\text{lin}}(\mathbf{y},\hat{\mathbf{y}}) = \left\| |STFT(\mathbf{y})| - |STFT(\hat{\mathbf{y}})| \right\|_1 \tag{19}$$

where $\|\cdot\|_F$ and $\|\cdot\|_1$ denote the Frobenius and L_1 norms, respectively, and $|STFT(\cdot)|$ is the magnitude spectrogram from the short-time Fourier transform. A weighting of $\frac{1}{N}$ is sometimes applied to $L_{\log}$ and L_{lin}, where N is the number of STFT bins (Yamamoto et al., 2020). Perceptually motivated frequency scales, like the mel scale, can also be applied to spectrograms. Arık et al. (2019) note that the spectral convergence loss "emphasises highly on large spectral components, which helps in early phases of training," whereas log magnitude spectral loss tend to help fit small amplitude components, which are to be more important later in training.

Wang and Yamagishi (2019) proposed computing a spectral loss using multiple STFT window sizes and hop lengths, aggregating the outputs into a single loss value.[13] This technique has since come to be known as the *multi-resolution STFT* (MRSTFT) loss (Yamamoto et al., 2020) or *multi-scale spectral loss* (MSS) loss (Engel et al., 2020a). The motivation behind this formulation is to compensate for the time-frequency resolution tradeoff inherent to the STFT.

A general form for MRSTFT losses is thus given by a weighted sum of the different spectral loss formulations at different resolutions:

$$\mathcal{L}_{\text{MRSTFT}} = \sum_{k\in K} \alpha_{\text{sc}}\mathcal{L}_{\text{sc},k} + \alpha_{\log}\mathcal{L}_{\log,k} + \alpha_{\text{lin}}\mathcal{L}_{\text{lin},k} \tag{20}$$

where K is the set of STFT configurations, $\mathcal{L}_{\cdot,k}$ is a spectral loss computed with a particular configuration, and α is the weighting for a loss term.[14] A consensus on the best spectral loss configuration has not emerged, suggesting that tuning of such losses is highly task-dependent.

Mel scaled spectral losses have also been used in a number of works (Mv and Ghosh, 2020; Kaneko et al., 2022; Choi H.-S. et al., 2023; Diaz et al., 2023; Song et al., 2023; Watts et al., 2023), after Fabbro et al. (2020) introduced the multi-resolution formulation and Kong et al. (2021) demonstrated their suitability for using in conjunction with adversarial objectives.

A large number of different multi-resolution configurations have been explored. Wu D.-Y. et al. (2022) suggested just four resolutions was sufficient for satisfactory results. Liu et al. (2020), on the other hand, used 12 different configurations noting that increasing this number resulted in fewer artefacts. (Barahona-Ríos and Collins, 2023). showed that a very small hop-size of 8 samples, with a window length of 32 samples, allowed good transient reconstruction. This is congruent with the findings of Kumar et al. (2023), who noted a small hop-size improved transient reconstruction in their neural audio codec.

Martinez Ramírez et al. (2020) noted that filtering can introduce frequency-dependent delays and phase inversions, which can cause problems for auditory loss functions. They proposed a delay

13 Wang et al. used a slightly different formulation of spectral loss they called *spectral amplitude distance*.

14 We point readers to the *auroloss* python package (Steinmetz and Reiss, 2020) for implementations of these methods and more auditory loss functions not mentioned here. Available on GitHub https://github.com/csteinmetz1/auraloss. Accessed 25th August 2023.

invariant loss to address this, which computes an optimal delay between $y[n]$ and $\hat{y}[n]$ using a cross-correlation function, and evaluates loss functions on time-aligned waveforms.

Wang and Yamagishi (2020) observed issues learning a stable pitch with the introduction of cyclic noise and proposed a masked spectral loss as a solution, which evaluates loss only in frequency bins containing harmonics of the known fundamental frequency. This is intended to penalise only harmonic mismatch, instead of accounting for the full spectral envelope. Wu D.-Y. et al. (2022) trained their parameter estimation model to predict f_0 from a mel spectrogram and used an explicit f_0 regression loss where both the ground truth and target f_0 were extracted using the WORLD vocoder (Morise et al., 2016), noting that MRSTFT alone was not sufficient to learn to reconstruct singing voices in their case while jointly learning f_0.

4.2 Parameter loss and self-supervision

Historically, parameter loss was commonly used in sound matching tasks involving black box or non-differentiable synthesisers (Yee-King et al., 2018; Barkan et al., 2019). This was, however, identified as a sub-optimal training objective (Esling et al., 2020), as synthesisers are ill-conditioned with respect to their parameters—that is, small changes in parameter space may yield large changes in the output.

However, with a differentiable synthesiser, parameter loss can be combined with auditory loss functions as a form of self-supervision, seemingly helping to avoid convergence on bad minima during training (Engel et al., 2020b; Masuda and Saito, 2023). For most parameters, loss is computed directly between estimated and ground truth parameter values, where ground truth parameters are randomly sampled to form a synthetic dataset of audio-parameter pairs.

Engel et al. (2020b) used a parameter regression pretraining phase over a large dataset of synthetic audio signals with complete parameter annotations. This enabled them to the fine-tune their network with an unconstrained differentiable sinusoidal model in conjunction with several other DDSP components for self-supervised pitch estimation. They additionally introduced a *sinusoidal consistency loss*, which is a permutation invariant parameter loss inspired by the two-way mismatch algorithm, to measure the error between sets of parameters for sinusoids representing partials of a target sound.

In a sound matching task with a differentiable subtractive synthesiser, Masuda and Saito (2021) observed that training only with spectral loss was ineffective, speculating that there was not a clear relationship between the loss and subtractive synthesis parameters. In subsequent work (Masuda and Saito, 2023), they used a combination of parameter loss, with a synthetic dataset, and various methods for introducing *out-of-domain* audio during training with a spectral loss. Through this procedure, they noted that certain parameters, such as the frequency of oscillators and chorus delay, were poorly optimised by a spectral loss.

Despite their tenuous perceptual correspondence, an advantage of parameter losses is their relative efficiency, particularly when parameters are predicted globally, or below audio sample rate. Han et al. (2023) proposed a method for reweighting the contributions of individual parameters to provide the best quadratic approximation of a given "perceptual" loss—i.e., a differentiable audio loss with desirable perceptual qualities, such as MRSTFT or joint time-frequency scattering (Muradeli et al., 2022). The proposed technique requires that loss gradients with respect to ground truth to parameters be evaluated once before training, limiting the technique to synthetic datasets, but the advantage is that online backpropagation through the differentiable synthesiser can be effectively avoided.

4.3 Adversarial training

Generative adversarial networks (Goodfellow et al., 2014) consist of two components: a generator, which produces synthetic examples, and a discriminator, which attempts to classify generated examples from real ones. These components are trained to optimise a *minimax* game. In later work, this adversarial training formulation was combined with a reconstruction loss (Isola et al., 2017) for image generation, a technique which has since become popular in audio generation (Kong et al., 2020).

From the perspective of reconstruction, the main motivation for adversarial training is that it tends to improve the naturalness and perceived quality of results (Michelashvili and Wolf, 2020; Choi H.-S. et al., 2023) and enables learning fine temporal structures (Liu et al., 2020), particularly when a multi-resolution discriminator is used (You et al., 2021). Further, despite using a phase invariant reconstruction loss, both Liu et al. (2020) and Watts et al. (2023) observed that adversarial training improved phase reconstruction and reduce phase-related audio artefacts. However, these benefits come at the expense of a more complex training setup.

Several variations on adversarial training for DDSP synthesis have been explored. The HiFi-GAN methodology (Kong et al., 2020) has been particularly popular (Kaneko et al., 2022; Choi H.-S. et al., 2023; Watts et al., 2023; Webber et al., 2023). This involves multiple discriminators operating at different periods and scales in a least-squares GAN setup, and includes a *feature matching loss* Kumar et al. (2019), which involves using distances between discriminator activations as an auxiliary loss. Others have used a hinge loss (Liu et al., 2020; Caillon and Esling, 2021; Nercessian, 2023) and a Wasserstein GAN (Juvela et al., 2019). Caillon and Esling (2021) and Watts et al. (2023) both train in two stages, first optimising for reconstruction, then introducing adversarial training to fine-tune the model.

5 Evaluation

Whilst loss functions may be selected to act as a proxy for certain signal characteristics, perceptual or otherwise, loss values typically can not be assumed to holistically describe the performance of a given method. Other methods must thus be used to evaluate DDSP-based synthesisers. In this section, we survey these methods and highlight trends within the reviewed literature.

Evaluation methods can be broadly subdivided into subjective and objective methods. In both speech and musical audio synthesis, subjective evaluations in the form of listening tests have been argued

TABLE 3 Methods for evaluating DDSP-based synthesis systems, listed with the number of times each method was used in the literature we surveyed (see Table 1). Methods with only one usage are grouped under the "Other" heading.

Evaluation type	Evaluation method	Music	Speech	Singing
Objective	Multi-scale spectral distance	6	1	2
	Extracted control (e.g., f0, loudness, etc.) distance	3	3	2
	Fréchet audio distance	4	1	1
	Log spectral distance	4	1	1
	MFCC distance	2	1	0
	Estimated control distance	2	0	1
	Parameter distance	1	1	0
	Other	0	4	1
	No objective evaluation	4	5	2
Subjective	Mean opinion score (MOS)	4	12	4
	MUSHRA or other multi-stimulus	1	7	0
	Preference ranking	4	0	0
	Informal evaluation	2	0	1
	Other	2	0	0
	No subjective evaluation	5	0	1
Complexity	Real-time factor	1	5	2
	Computation time	1	4	0
	FLOPS	0	4	0
	Other	0	1	0
	No complexity evaluation	14	9	4

to provide a more meaningful assessment of results Wagner et al. (2019); Yang and Lerch (2020), given the inherent subjectivity of the notion of audio quality. This preference was reflected in the DDSP-based speech synthesis literature we reviewed, in which every paper conducted a listening test. Conversely, only slightly over half of the work reviewed on music and singing voice synthesis conducted a subjective evaluation of results. This discrepancy may be partially explained by the existence of standards for subjective evaluation in speech research, in contrast to music. That being said, it became clear through this review that there exists no single unified method for evaluating DDSP synthesis results.

In the following subsections we detail the more widely used evaluation methods, and discuss how these have been used in work to date. We also tally the number of uses of these approaches in Table 3.

5.1 Objective evaluation

An objective evaluation typically involves the calculation of some number of metrics from the outputs of a given model, or from other characteristics of the model such as the time taken to perform a given operation. While such metrics may not directly correspond to perceptual attributes of synthesised signals (Manocha et al., 2022; Vinay and Lerch, 2022), they may be attractive as an alternative to a listening test (Yang and Lerch, 2020) due to their lack of dependency on recruiting participants and their reproducibility. They are also frequently employed alongside listening tests, in which case they may be used to probe specific attributes, or to facilitate comparison between different experiments.

5.1.1 Audio similarity metrics

Perhaps the most obvious target for a metric of audio quality is, when appropriate to the task, a model's ability to reconstruct a given piece of audio. Such evaluations necessarily require ground truth audio, which is typically available in resynthesis tasks such as copy synthesis and musical instrument modelling. Objective evaluation metrics that operate on synthesized and ground truth audio are referred to as audio similarity metrics (also known as *full-reference* or *intrusive*). This is in contrast to *no-reference* metrics (also known as *reference-free* or *non-intrusive*) (Manocha et al., 2022), which do not require a ground truth audio.

Perhaps the simplest audio similarity metric is a waveform distance, taken directly between time-domain samples. These are used infrequently for evaluation of DDSP-based synthesisers, likely because of the emphasis they place on phase differences, which are not necessarily perceived, and are often not explicitly modelled. Yu and Fazekas (2023) are an exception—they used an L_2 waveform loss to highlight the ability of their differentiable source-filter approach to reconstruct phase.

Most commonly reported are distances computed between time-frequency representations of audio signals, sometimes referred to as spectral error. Multi-scale spectrogram error, which is often also used as a loss function, was reported in a number of music (Masuda and Saito, 2021; Wu D.-Y. et al., 2022; Renault et al., 2022; Shan et al., 2022), singing (Yu and Fazekas, 2023), and speech evaluations (Nercessian, 2021). Other spectral distances reported included log-spectral distance Masuda and Saito (2021); Subramani et al. (2022), log mel-spectral distance Nercessian (2023), and mel-cepstrum distortion (MCD) (Masuda and Saito, 2023).

Similarity metrics motivated by perception, such as the Perceptual Evaluation of Speech Quality (PESQ) (International Telecommunication Union, 2001) were developed as alternatives to costly subjective evaluations. However, we observed that these saw limited use in the evaluation of DDSP vocoders Mv and Ghosh (2020).

Similarity metrics have also been designed around the extraction of higher level signal features. For example, the ability of a model to accurately reconstruct fundamental frequency has been evaluated by measuring the MAE error between f_0 extracted from ground truth and synthesized results. Results are typically compared to neural audio synthesis baselines (e.g., WaveRNN) and support the claim that DDSP models are better at preserving pitch (Engel et al., 2020a; Wu D.-Y. et al., 2022). A similar evaluation has also been applied for loudness (Engel et al., 2020a; Kawamura et al., 2022).

5.1.2 Reference-free audio metrics

In contrast to audio similarity metrics, reference free metrics provide an indication of audio quality without the need for a ground truth audio signal. The Fréchet Audio Distance (FAD) (Kilgour et al., 2019) is an example of such a metric, and was originally developed for music enhancement evaluation. The FAD is computed by fitting multivariate Gaussian distributions to two sets of neural embeddings: one computed over "clean" audio (the *background set*) and another computed over test audio. The Fréchet distance is then computed between these distributions. Typically, a pre-trained VGGish model (Hershey et al., 2017) is used to compute embeddings, a formulation that we found to recur in evaluation of both music (Hayes et al., 2021; Caspe et al., 2022; Ye et al., 2023) and singing voice (Wu D.-Y. et al., 2022; Yu and Fazekas, 2023) papers we reviewed. In the speech domain, (Kaneko et al., 2022), used Fréchet wav2vec distance (cFW2VD), which replaced the VGGish model with wav2vec.

Some of the work we reviewed observed a correlation between the FAD and their subjective evaluations (Hayes et al., 2021; Kaneko et al., 2022), which is consistent with the findings of Manocha et al. (2022) who also noted that reference-free metrics aligned better with human judgements than audio similarity metrics. However, these results are not universal and there exist multiple counter-examples in the literature (Vinay and Lerch, 2022; Choi K. et al., 2023). These discrepancies may be explained by sample-size bias or the fact that VGGish embeddings are suboptimal for FAD (Gui et al., 2023). In general, the development of a reliable proxy for perceived quality of audio synthesis algorithms remains an open research question.

5.1.3 Parameter reconstruction

Parameter reconstruction metrics compare synthesizer parameters or latent parameters to ground truth values. These metrics are more common in tasks such as sound matching and performance rendering, where ground truth parameters are either known *a priori* or can be straightforwardly extracted. Absolute (Masuda and Saito, 2023; Südholt et al., 2023) and squared Wu et al. (2022c) distances have both seen use for this purpose. Parameter distances can provide insight into how well a method is able to reconstruct synthesis parameters, although it has been shown in previous work that parameter error and audio similarity are not well-aligned (Esling et al., 2020), at least in the case of music production-focused musical synthesizers. Parameter error for physical models or spectral modeling synthesizers may thus correlate better with perceived quality.

5.1.4 Computational complexity

A repeatedly cited motivation for the use of DDSP is an improvement in computational efficiency and reduction in inference speed, particularly when compared to other neural vocoders or synthesisers. This is corroborated by the substantial number of reviewed papers which included explicit evaluation of computational efficiency. Performing a real-time factor (RTF) test is the most common method for measuring inference speed. The metric is defined as

$$\mathrm{RTF} = \frac{t_p}{t_i} \tag{21}$$

where t_i is the input target duration (e.g., 1 s) and t_p is the time taken to synthesise that much audio. Any value RTF <1.0 indicates a real-time operation. Variations on RTF include reporting the number of samples generated per second Caillon and Esling (2021); Wang et al. (2019a) and times faster than real-time (Kaneko et al., 2022), which is the inverse of Eq. 21. A number of factors contribute to RTF tests including hardware, programming language, audio sampling rate, duration of test samples, and model size[15].

The selection of duration t_i has a bearing on RTF results and has interesting implications on the end application. In speech, utterance-by-utterance processing is a typical use-case for on-device synthesis Webber et al. (2023). Interactive music applications, on the other hand, typically use buffer-by-buffer processing and require a buffer size corresponding to 10 ms or less to fulfill the low-latency constraint for real-time interaction. Maintaining an RTF <1.0 with such small buffer sizes is a challenging constraint, as RTF values can be inversely correlated with buffer size Hayes et al. (2021).

Computational complexity can also be measured by counting the number of floating-point operations required to synthesize one second of audio (FLOPS) (Liu et al., 2020; Tian et al., 2020; Watts et al., 2023). In theory, this provides a hardware agnostic measurement of computational complexity and provides fair method for comparing algorithms (Schwartz et al., 2020). However, running time does not always perfectly correlate with

15 RTF tests may be performed in an end-to-end fashion and include neural networks used to predict synthesis parameters or separately on only the vocoder/synthesizer component.

FLOPS; for instance, the parallelism provided by GPUs offers speed-ups for convolutional layers that are not possible on CPUs (Asperti et al., 2022).

5.2 Subjective evaluation

Evaluation of audio quality using listening tests is considered the "gold standard" Manocha et al. (2022) in speech research. A number of different listening test variations exist, however the most commonly used for DDSP evaluations is the absolute category rating (ACR) test from the ITU-T recommendation P.800 International Telecommunication Union (1996), also known as "mean opinion score" (MOS). When assessing synthesis quality, participants are provided with a set of audio samples and asked to rate each on a five-point Likert scale according to a prompt, usually related to audio quality. The majority of subjective evaluations we reviewed used MOS evaluations and a number reported using crowd-sourcing platforms to recruit participants (Nercessian, 2021; Subramani et al., 2022). MOS was also used to evaluate the quality of TTS applications of DDSP synthesisers (Juvela et al., 2019; Wang and Yamagishi, 2019; Liu et al., 2020; Choi H.-S. et al., 2023; Song et al., 2023).

Another listening test format, originally developed for evaluating audio codecs[16], is the MUlti Stimulus test with Hidden Reference and Anchor (MUSHRA) evaluation (International Telecommunication Union, 2015). Similar to MOS, MUSHRA tests ask participants to rate the quality of an audio recording. However, these use continuous sliders as opposed to a Likert scale and, importantly, they involve comparing a number of stimuli side-by-side to a piece of reference audio. Amongst these stimuli are the hidden reference and anchor, which provide a built-in mechanism to identify listeners that may need excluding from results due to suboptimal listening conditions, poor compliance, or unreported hearing differences—considerations that are especially important when conducting online listening tests. Screening criteria have also been developed for online MOS evaluations Ribeiro et al. (2011).

It is less clear how subjective evaluation of tasks like timbre transfer and voice conversion should be performed. In voice conversion tasks a MOS test can be conducted and participants asked to rate the *speaker similarity* and *naturalness*/*quality* of an utterance (Nercessian, 2021; Guo H. et al., 2022; Choi H.-S. et al., 2023). A challenge with rating speaker similarity, as reported by Wester et al. (2016), is that it is not necessarily a familiar task to listeners—humans are more adept at recognizing individual speakers as opposed to evaluating how close one speaker is to another. Wester et al. instead propose a pairwise evaluation that asks participants to report their confidence in the identity of voice. Nonetheless, the DDSP papers we reviewed used the more conventional MOS scale. Timbre transfer has similarly been evaluated using an *instrument similarity* score and a *melody similarity* score using a MOS test Michelashvili and Wolf (2020).

Performance rendering applications are additionally tasked with modelling of musical expression, itself a challenging task and under-specified problem, with results that are open to subjective interpretation. The evaluation of these systems has similarities with the evaluation of generative music models (Yang and Lerch, 2020). A commonly used method in generative modelling, conceptually related to a Turing test, is to ask participants whether a result sounds like it was produced by a human. In the context of DDSP-based performance rendering, a variant of this approach was used by Castellon et al. (2020), who asked participants to select the recording that sounds "more like a real human playing a real instrument". Wu et al. (2022c) used a similar prompt for violin recordings.

We note that while a majority of the reviewed papers that performed a subjective evaluation reported confidence intervals around their results, not all papers performed statistical hypothesis testing.

6 Discussion

In this section we discuss both the strengths and limitations of DDSP for audio synthesis. For those new to the topic, we hope to assist in evaluating the suitability of DDSP for applications and research directions of interest. With more experienced practitioners in mind, we seek to highlight promising directions for future work and present open research questions that we argue hinder wider applicability and adoption of DDSP for audio synthesis.

6.1 DDSP and inductive bias

Underpinning DDSP, and the related field of differentiable rendering Kato et al. (2020), is the notion of incorporating a domain-appropriate *inductive bias*. Imposing strong assumptions on the form a model's outputs should take—for example, producing a signal using only an additive synthesiser—limits model flexibility, but in return can improve the appropriateness of outputs. When such a bias is appropriate to the task and data, this can be highly beneficial. However, this bias also limits the broader applicability of the model, and may cause issues with generalisation.

In reviewing the literature, we found that authors most commonly referred to the following strengths of DDSP methods.

1. Audio quality: differentiable oscillators have helped reduce artefacts caused by phase discontinuities (Engel et al., 2020a) and pitch errors (Nercessian, 2023), and have enabled SOTA results when incorporated into hybrid models (Choi H.-S. et al., 2023; Song et al., 2023);
2. Data efficiency: an appropriately specified differentiable signal processor seems to reduce the data burden, with good results achievable using only minutes of audio (Engel et al., 2020a; Michelashvili and Wolf, 2020);
3. Computational efficiency: offloading signal generation to efficient synthesis algorithms carries the further benefit of faster inference (Carney et al., 2021; Hayes et al., 2021; Shan et al., 2022).
4. Interpretability: framing model outputs in terms of signal processor parameters allows for *post hoc* interpretation.

16 Synthesizers and vocoders are in a sense narrowly specified audio codecs (Hayes et al., 2021).

Differentiable articulatory models can provide insights into vocal production (Südholt et al., 2023), while common audio synthesiser designs can be used to enable interpretable decomposition of target sounds (Caspe et al., 2022; Masuda and Saito, 2023);

5. Control/creative affordances: explicit controls based on perceptual attributes such as pitch and loudness have enabled creative applications such as real-time timbre transfer (Carney et al., 2021), expressive musical performance rendering (Wu et al., 2022c), and voice designing (Choi H.-S. et al., 2023).

Furthermore, differentiable audio synthesis has enabled new techniques in tasks beyond the realm of speech and music synthesis. For example, Schulze-Forster et al. (2023) applied differentiable source-filter models to perform unsupervised source separation, while Engel et al. (2020b) used an analysis-by-synthesis framework to predict pitch without explicit supervision from ground truth labels. Further, DDSP-based synthesisers and audio effects have been used for tasks such as data amplification (Wu et al., 2022b) and data augmentation (Guo Z. et al., 2022).

However, the majority of these benefits have been realised within the context of synthesising monophonic audio with a predominantly harmonic spectral structure, where it is possible to explicitly provide a fundamental frequency annotation. This includes solo monophonic instruments, singing, and speech. This limitation is necessitated by the choices of synthesis model that have made up the majority of DDSP synthesis research—typically, these encode a strong bias towards the generation of harmonic signals, while the reliance on accurate f_0 estimates renders polyphony significantly more challenging. In this sense, the trade-off is a lack of straightforward generalisation to other classes of sound—producing drum sounds with a differentiable harmonic-plus-noise synthesiser conditioned on loudness and f_0 is unlikely to produce useable results, for example,. This is not an inherently negative characteristic of the methodology—excluding possibilities from the solution space (e.g., non-harmonic sounds or phase discontinuities) has been instrumental in realising the above benefits such as data-efficient training and improved audio quality.

The application of DDSP to a broader range of sounds has consequently been slow. Many synthesis techniques exist which are capable of generating polyphonic, inharmonic, or transient-dense audio, but to date there has been limited exploration of these in the literature. This may, in part, be due to the difficulty inherent in optimising their parameters by gradient descent, as noted by Turian and Henry (2020) and Hayes et al. (2023). Work on differentiable FM synthesis (Caspe et al., 2022; Ye et al., 2023), for example, may eventually lead to the modelling of more varied sound sources due to its ability to produce complex inharmonic spectra, but as noted by Caspe et al. (2022), optimisation of carrier and modulator frequencies is currently not possible due to loss surface ripple.

In summary, the inductive bias inherent to DDSP represents a trade-off. It has enabled in certain tasks, through constraint of solution spaces, improved audio quality, data efficiency, computational efficiency, interpretability, and control affordances. In exchange, these strong assumptions narrowly constrain these models to their respective domains of application, and limit their generalisation to many real world scenarios where perfect harmonicity or isolation of monophic sources cannot be guaranteed. Hence, we argue that a deeper understanding of the trade-offs induced by specific differentiable synthesisers would be a valuable future research direction for the audio synthesis community.

6.2 DDSP in practice

Producing a differentiable implementation of a signal processor or model is now relatively straightforward due to the wide availability of automatic differentiation frameworks.[17] Such libraries expose APIs containing mathematical functions and numerical algorithms with corresponding CUDA kernels and explicit gradient implementations. This allows DSP algorithms to be expressed directly using these primitives, and the resulting composition of their gradients to be calculated by backpropagation. Additionally, with the growing interest in audio machine learning research and DDSP, several specialized packages have been created.[18]

Despite the simplicity of implementation, however, certain techniques may not allow for straightforward optimisation without workarounds or specialised algorithms. These include ADSR envelopes (Masuda and Saito, 2023), quantisation (Subramani et al., 2022), IIR filters (Kuznetsov et al., 2020; Nercessian et al., 2021), and sinusoidal models (Hayes et al., 2023).

Further, the process of implementing a differentiable digital signal processor can itself be time consuming, introducing an additional burden to the research pipeline. Furthermore, the programming language of a particular library–most commonly Python–may not be the same language used in an end application. However, recent efforts have sought to support the translation of DSP code into differentiable implementations[19] and support the deployment of audio code written in machine learning libraries into audio plugins[20]. Future efforts could explore how fast inference libraries focused on real-time audio applications could be integrated with DDSP audio synthesis (Chowdhury, 2021).

6.2.1 Alternatives to automatic differentiation

In certain situations, manual implementation of DSP algorithms in automatic differentiation software is not possible (e.g., when using a black-box like a VST audio plugin or physical hardware audio processor) or is undesirable given the additional challenge and engineering overhead. Three alternative methods to manually

17 PyTorch https://pytorch.org/and TensorFlow https://www.tensorflow.org/are two examples that are well supported and that have been used extensively in previous DDSP work. URLs accessed 27 August 2023.

18 These include the original DDSP library https://github.com/magenta/ddsp introduced by (Engel et al., 2020a), a PyTorch port of the DDSP library https://github.com/acids-ircam/ddsp_pytorch, TorchAudio https://pytorch.org/audio/stable/index.html, and torchsynth https://github.com/torchsynth/torchsynth introduced by (Turian et al., 2021), accessed 27 August 2023.

19 (Braun, 2021) recently introduced the ability to transpile DSP code written in Faushttps://faust.grame.fr/to Jax code (Bradbury et al., 2018) into the DawDreamer library. Released on GitHub, v0.6.14 https://github.com/DBraun/DawDreamer/releases/tag/v0.6.14. URLs accessed 27th August 2023.

20 https://neutone.space/accessed 27 August 2023.

implementing DSP operations differentiably have been explored in previous audio research, although have not been extensively applied to synthesis at the time of writing.

Neural proxies seek to train a deep learning network to mimic the behaviour of an audio processor, including the effect of parameter changes. *Hybrid neural proxies* use a neural proxy only during training, replacing the audio processing component of the proxy with the original DSP during inference. Steinmetz et al. (2022a) first applied hybrid neural proxies to audio effect modelling, further distinguishing between half hybrid approaches which use a neural proxy for both forward and backward optimization passes and full hybrid approaches that only use a proxy for the backward pass. *Numerical gradient approximation* methods, on the other hand, do not require any component of the audio processing chain to be explicitly differentiable and instead estimate black-box gradients using a numerical method such as simultaneous permutation stochastic approximation (SPSA) (Spall, 1998; Martinez Ramirez et al., 2021) successfully applied this to approximate the gradients of audio effect plugins.

6.3 Looking ahead

We wish, finally, to discuss the opportunities, risks, and open challenges in future research into DDSP for audio synthesis.

6.3.1 Hybrid approaches

In this review, we undertook a cross-domain survey of DDSP-based audio synthesis encompassing both speech and music. In doing so, we note that the progression of DDSP methods in speech synthesis commenced with mixed approaches, integrating DSP components to neural audio synthesisers and benefiting from the strengths of both. Early examples included the incorporation of LPC synthesis filters (Juvela et al., 2019; Valin and Skoglund, 2019), effectively "offloading" part of the synthesis task. Conversely, the dominant applications of DDSP to music tend to offload *all* signal generation to differentiable DSP components. We note that there may be opportunities for both fields to benefit from one another's findings here, exploring further intermediate *hybrid* approaches.

Hybrid methods, in general, combine the aforementioned strengths of DDSP with more general deep learning models. This is visible in the use of pre- and post-nets in speech and singing synthesis, for example, (Nercessian, 2021; Choi H.-S. et al., 2023; Nercessian, 2023); in the integration of a differentiable filtered noise synthesiser to RAVE (Caillon and Esling, 2021); or in the results of the recent SVC challenge, in which the top two results used a pre-trained DDSP synthesiser to condition a GAN (Huang et al., 2023). We thus expect hybrid methods to be a fruitful future research direction, where DSP domain knowledge can help guide and constrain more general neural audio synthesisers, and neural networks can help generalise narrow DDSP solutions.

6.3.2 Implementations, efficiency, and stability

A major challenge in DDSP research is ensuring that it is computationally and numerically feasible to even perform optimisation. While many DSP operations are deeply connected to neural network components, the transition into gradient descent over DDSP models is not inherently straightforward. IIR filters operating on audio, for instance, are likely to be applied over many more time steps than a conventional RNN, leading to specific challenges such as the memory cost of unrolled operations, and filter stability. As a result, efficient training algorithms have received considerable attention. Further, recent work by Yu and Fazekas (2023) presented an efficient GPU implementation of all-pole IIR filters, evaluated exactly and recursively, with efficient backpropagation based on a simplified algorithm for evaluating the gradient. Similarly, accompanying their original contribution, Engel et al. (2020a) included an "angular cumsum" CUDA kernel, enabling phase accumulation without numerical issues. The development of open-source and efficient implementations of differentiable signal processing operations or their constituent parts is of clear benefit to future work, and thus we argue that this is a valuable area in which to direct future work.

6.3.3 Evaluation and comparison of approaches

As we note in Section 5, there does not exist a unified framework for evaluating DDSP-based audio synthesisers, or in fact for synthesisers in general. This renders comparison between methods challenging, requiring frequent re-training and re-implementation of baselines under new conditions.

However, it is not clear whether such a unified evaluation is possible, or even desirable. As a broad family of methods, DDSP has found a diverse range of applications, and many of these involve highly specialised implementations. For example, an acoustically informed model of piano detuning (Renault et al., 2022) was paired with a differentiable synthesiser. Evaluating the success of such an approach would arguably require a specialised experimental design.

Nonetheless, the consistency with which listening tests, and in particular MOS tests, are performed with speech synthesisers does allow for a reader to make coarse judgements about the relative quality of the tested methods, even if these do not generalise beyond the particular study. With this in mind, we pose the question: would applications of DDSP to music benefit from wider use of a standardised listening test format?

Subjective listening tests are time consuming and are not always a viable option. This is particularly true when a large number of models or stimuli require comparison. For this reason, quantitative metrics which correlate well with perception are potentially of great use to researchers. We note that while PESQ (International Telecommunication Union, 2001) has seen some limited use in DDSP-based speech research, and FAD (Kilgour et al., 2019) has been used in a handful of works, there does not appear to be a commonly agreed upon quantitate proxy for perceptual evaluation. This is likely due in part to the difficulty of designing such a tool, although we note promising progress in this direction using reference-free speech metrics (Manocha et al., 2022) and improving FAD for generative music evaluation (Gui et al., 2023).

Domain and task-specific challenges also represent a valuable approach to evaluation. The speech and singing voice communities have already made particularly constructive use of this approach. The voice conversion challenge (Wester et al., 2016) and singing voice conversion challenge Huang et al. (2023) are clear examples of the benefits of such task-specific evaluations, and have, in fact, already started to highlight the value of DDSP, while not being specifically focused on DDSP methods. Open-challenges that target

specific applications of DDSP will be a valuable way to both encourage further research in this area and support a deeper understanding of the strengths of various DDSP techniques.

6.3.4 Open questions

Through our review we noted a small number of recurring themes relating to specific challenges in working with DDSP, often mentioned as a corollary to the main results, or to motivate an apparent workaround. We also observed that certain methods have received attention in one application domain but not yet been applied to another, or have simply received only a small amount of attention. In this section, we briefly compile these observations in the hopes that they might help direct future research. These open questions include.

1. The difficulty estimating oscillator frequencies by gradient descent, discussed in detail by Turian and Henry (2020) and Hayes et al. (2023) (see Section 3.2.2)
   - A related issue is the tuning of modulator frequencies in FM synthesis (Caspe et al., 2022) (see Section 3.4.2)
2. The invariance of some signal processors under permutations of their parameters appears to be relevant to optimisation, as noted by Nercessian (2020), Engel et al. (2020b), and Masuda and Saito (2023), but there has been no specific investigation as to how this impacts training.
3. Estimating global parameters like ADSR segment times appears to be challenging, especially when these lead to complex interactions (Masuda and Saito, 2023). This is important for modelling commercial synthesisers differentiably.
4. Estimating delay parameters and compensating for delays poses specific challenges, which will most likely require specialised loss functions. (Martinez Ramirez et al., 2021; Masuda and Saito, 2023).
5. Neural proxy (Steinmetz et al., 2022a) and gradient approximation (Martinez Ramirez et al., 2021) techniques have been applied in automatic mixing and intelligent audio production, but not yet explored for audio synthesis. Could these allow black-box software instruments to be used pseudo-differentiably?
6. Directed acyclic audio signal processing graphs have been estimated blindly (Lee et al., 2023), while FM routing has been optimised through neural architecture search (Ye et al., 2023). Can these methods be generalised to allow estimation of arbitrary synthesiser topologies?

7 Conclusion

In this article, we have surveyed the literature on differentiable digital signal processing (DDSP) across the domains of speech, music, and singing voice synthesis. We provided a detailed overview of the major tasks and application areas where DDSP has been used, and in this process identified a handful of recurring motivations for its adoption. In particular, it became clear that DDSP is most frequently deployed where it confers a benefit in audio quality, data efficiency, computational efficiency, interpretability, or control.

However, simply implementing a known signal processor differentiably is frequently insufficient to realise these benefits. Through our review of the major technical contributions to the field, we observed that there remain several open problems, such as frequency estimation by gradient descent, which may hinder the general applicability of DDSP methods. Further, we identify that selecting a training objective and designing an appropriate evaluation is non-trivial, with many different approaches appearing in the literature. In this sense, we would argue that the fields of music and speech synthesis may benefit from a sharing of expertise—in particular, the prevalence of standardised listening tests in speech may help those working on music-related synthesis tasks better assess their progress.

In our discussion, we noted that the purported advantages of DDSP techniques are typically gained at the expense of broad applicability and generalisation. Finally, we identified promising avenues for future research, such as the development of hybrid models, which incorporate DDSP components into more general models, and concluded our review by highlighting several knowledge gaps that warrant attention in future work.

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Author contributions

BH: Conceptualization, Investigation, Methodology, Writing–original draft, Writing–review and editing. JS: Investigation, Methodology, Writing–original draft, Writing–review and editing. GF: Supervision, Writing–review and editing. AM: Supervision, Writing–review and editing. CS: Supervision, Writing–review and editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was supported by United Kingdom Research and Innovation (grant number EP/S022694/1).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Alonso, J., and Erkut, C. (2021). Latent space explorations of singing voice synthesis using DDSP. arXiv [Preprint]. Available at: https://arxiv.org/abs/2103.07197 (Accessed June 03, 2023).

Arık, S. Ö., Jun, H., and Diamos, G. (2019). Fast spectrogram inversion using multi-head convolutional neural networks. *IEEE Signal Process. Lett.* 26, 94–98. doi:10.1109/LSP.2018.2880284

Asperti, A., Evangelista, D., and Marzolla, M. (2022). "Dissecting FLOPs along input dimensions for GreenAI cost estimations," in *Machine learning, optimization, and data science* (Cham: Springer International Publishing), 86–100. doi:10.1007/978-3-030-95470-3_7

Atal, B. S., and Hanauer, S. L. (1971). Speech analysis and synthesis by linear prediction of the speech wave. *J. Acoust. Soc. Am.* 50, 637–655. doi:10.1121/1.1912679

Back, A. D., and Tsoi, A. C. (1991). FIR and IIR synapses, a new neural network architecture for time series modeling. *Neural Comput.* 3, 375–385. doi:10.1162/neco.1991.3.3.375

Bakhturina, E., Lavrukhin, V., Ginsburg, B., and Zhang, Y. (2021). Hi-fi multi-speaker English TTS dataset. arXiv. [Preprint]. Available at https://arxiv.org/abs/2104.01497 (Accessed July 20, 2023).

Barahona-Ríos, A., and Collins, T. (2023). NoiseBandNet: controllable time-varying neural synthesis of sound effects using filterbanks. arXiv [Preprint]. Available at https://arxiv.org/abs/2307.08007 (Accessed August 02, 2023).

Barkan, O., Tsiris, D., Katz, O., and Koenigstein, N. (2019). InverSynth: deep estimation of synthesizer parameter configurations from audio signals. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* 27, 2385–2396. doi:10.1109/TASLP.2019.2944568

Bhattacharya, P., Nowak, P., and Zölzer, U. (2020). "Optimization of cascaded parametric peak and shelving filters with backpropagation algorithm," in Proceedings of the 23rd International Conference on Digital Audio Effects, 101–108.

Bilbao, S. (2009). *Numerical sound synthesis: finite difference schemes and simulation in musical acoustics*. John Wiley and Sons.

Birkholz, P. (2013). Modeling consonant-vowel coarticulation for articulatory speech synthesis. *PLOS ONE* 8, e60603. doi:10.1371/journal.pone.0060603

Bitton, A., Esling, P., and Chemla-Romeu-Santos, A. (2018). Modulated Variational auto-Encoders for many-to-many musical timbre transfer. arXiv [Preprint] Available at: https://www.researchgate.net/publication/328016649_Modulated_Variational_auto-Encoders_for_many-to-many_musical_timbre_transfer (Accessed July 06, 2023).

Blaauw, M., and Bonada, J. (2017). A neural parametric singing synthesizer. *Proc. Interspeech*, 4001–4005. doi:10.21437/Interspeech.2017-1420

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., et al. (2018). JAX: composable transformations of Python+NumPy programs. [Software] Available at http://github.com/google/jax (Accessed October 25, 2023).

Braun, D. (2021). "DawDreamer: bridging the gap between digital audio workstations and Python interfaces," in Late-Breaking Demo Session of the 22nd International Society for Music Information Retrieval Conference.

Cahill, T. (1897). *Art of and apparatus for generating and distributing music electrically*. US Patent US580035A.

Caillon, A., and Esling, P. (2021). *RAVE: a variational autoencoder for fast and high-quality neural audio synthesis*. doi:10.48550/arXiv.2111.05011

Campolucci, P., Piazza, F., and Uncini, A. (1995). "On-line learning algorithms for neural networks with IIR synapses," in Proceedings of ICNN95 - International Conference on Neural Networks, Perth, WA, Australia (IEEE), 865–870. doi:10.1109/ICNN.1995.4875322

Carney, M., Li, C., Toh, E., Yu, P., and Engel, J. (2021). "Tone transfer: in-browser interactive neural audio synthesis," in Joint Proceedings of the ACM IUI 2021 Workshops.

Carson, A., Valentini-Botinhao, C., King, S., and Bilbao, S. (2023). "Differentiable grey-box modelling of phaser effects using frame-based spectral processing," in Proceedings of the 26th International Conference on Digital Audio Effects.

Caspe, F., McPherson, A., and Sandler, M. (2022). "DDX7: differentiable FM synthesis of musical instrument sounds," in Proceedings of the 23rd International Society for Music Information Retrieval Conference.

Castellon, R., Donahue, C., and Liang, P. (2020). "Towards realistic MIDI instrument synthesizers," in NeurIPS Workshop on Machine Learning for Creativity and Design.

Chen, J., Tan, X., Luan, J., Qin, T., and Liu, T. Y. (2020). HiFiSinger: towards high-fidelity neural singing voice synthesis. arXiv [Preprint]. Available at: https://arxiv.org/abs/2009.01776 (Accessed July 24, 2023).

Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., and Chan, W. (2021). "Wavegrad: estimating gradients for waveform generation," in International Conference on Learning Representations.

Childers, D., Yegnanarayana, B., and Wu, K. (1985). "Voice conversion: factors responsible for quality," in ICASSP '85. IEEE International Conference on Acoustics, Speech, and Signal Processing, 748–751. doi:10.1109/ICASSP.1985.116847910

Cho, Y.-P., Yang, F.-R., Chang, Y.-C., Cheng, C.-T., Wang, X.-H., and Liu, Y.-W. (2021). "A survey on recent deep learning-driven singing voice synthesis systems," in 2021 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR), 319–323. doi:10.1109/AIVR52153.2021.00067

Choi, H.-S., Yang, J., Lee, J., and Kim, H. (2023a). "NANSY++: unified voice synthesis with neural analysis and synthesis," in International Conference on Learning Representations.

Choi, K., Im, J., Heller, L., McFee, B., Imoto, K., Okamoto, Y., et al. (2023b). Foley sound synthesis at the DCASE 2023 challenge. arXiv [Preprint]. Available at: https://arxiv.org/abs/2304.12521 (Accessed August 02, 2023).

Chowdhury, J. (2021). RTNeural: fast neural inferencing for real-time systems. arXiv [Preprint] Available at : https://arxiv.org/abs/2106.03037 (Accessed August 27, 2023).

Chowning, J. M. (1973). The synthesis of complex audio spectra by means of frequency modulation. *J. Audio Eng. Soc.* 21, 526–534.

Colonel, J. T., Steinmetz, C. J., Michelen, M., and Reiss, J. D. (2022). "Direct design of biquad filter cascades with deep learning by sampling random polynomials," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 3104–3108. doi:10.1109/ICASSP43922.2022.9747660

Cook, P. R. (1996). Singing voice synthesis: history, current work, and future directions. *Comput. Music J.* 20, 38–46. doi:10.2307/3680822

Cooley, J. W., and Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. *Math. Comput.* 19, 297–301. doi:10.1090/S0025-5718-1965-0178586-1

Cramer, A. L., Wu, H.-H., Salamon, J., and Bello, J. P. (2019). "Look, listen, and learn more: design choices for deep audio embeddings," in IEEE International Conference on Acoustics, Speech and Signal Processing, Brighton, UK (ICASSP), 3852–3856. doi:10.1109/ICASSP.2019.8682475

Dai, S., Zhang, Z., and Xia, G. G. (2018). Music style transfer: a position paper. In Proceeding of International Workshop on Musical Metacreation (MUME)

De Man, B., Stables, R., and Reiss, J. D. (2019). *Intelligent music production* (Waltham, Massachusetts: Focal Press).

Devis, N., Demerlé, N., Nabi, S., Genova, D., and Esling, P. (2023). "Continuous descriptor-based control for deep audio synthesis," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (IEEE), 1–5. doi:10.1109/ICASSP49357.2023.10096670

Diaz, R., Hayes, B., Saitis, C., Fazekas, G., and Sandler, M. (2023). "Rigid-body sound synthesis with differentiable modal resonators," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (IEEE).

Donahue, C., McAuley, J., and Puckette, M. (2019). "Adversarial audio synthesis," in International Conference on Learning Representations.

Dudley, H., and Tarnoczy, T. H. (1950). The speaking machine of wolfgang von Kempelen. *J. Acoust. Soc. Am.* 22, 151–166. doi:10.1121/1.1906583

Dudley, W. H. (1939). The vocoder. *Bell Labs Rec.* 18, 122–126.

Dupre, T., Denjean, S., Aramaki, M., and Kronland-Martinet, R. (2021). "Spatial sound design in a car cockpit: challenges and perspectives," in *2021 immersive and 3D audio: from architecture to automotive (I3DA)* (Bologna, Italy: IEEE). doi:10.1109/I3DA48870.2021.9610910

Elman, J. L. (1990). Finding structure in time. *Cognitive Sci.* 14, 179–211. doi:10.1207/s15516709cog1402_1

Engel, J., Agrawal, K. K., Chen, S., Gulrajani, I., Donahue, C., and Roberts, A. (2019). "GANSynth: adversarial neural audio synthesis," in 7th International Conference on Learning Representations, New Orleans, LA, USA.

Engel, J., Hantrakul, L. H., Gu, C., and Roberts, A. (2020a). DDSP: differentiable digital signal processing. Available at: https://arxiv.org/abs/2001.04643.

Engel, J., Resnick, C., Roberts, A., Dieleman, S., Norouzi, M., Eck, D., et al. (2017). "Neural audio synthesis of musical notes with WaveNet autoencoders," in Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 1068–1077.70

Engel, J., Swavely, R., and Roberts, A. (2020b). "Self-supervised pitch detection by inverse audio synthesis," in Proceedings of the International Conference on Machine Learning.

Esling, P., Masuda, N., Bardet, A., Despres, R., and Chemla-Romeu-Santos, A. (2020). Flow synthesizer: universal audio synthesizer control with normalizing flows. *Appl. Sci.* 10, 302. doi:10.3390/app10010302

Fabbro, G., Golkov, V., Kemp, T., and Cremers, D. (2020). Speech synthesis and control using differentiable DSP. arXiv [Preprint]. Available at: https://arxiv.org/abs/2010.15084 (Accessed June 03, 2023).

Gatys, L. A., Ecker, A. S., and Bethge, M. (2015). A neural algorithm of artistic style. *J. Vis.* 16, 326. doi:10.1167/16.12.326

Gómez, E., Blaauw, M., Bonada, J., Chandna, P., and Cuesta, H. (2018). Deep learning for singing processing: achievements, challenges and impact on singers and listeners.

arXiv [Preprint]. Available at: https://arxiv.org/abs/1807.03046 (Accessed July 21, 2023).

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). "Generative adversarial nets," in Advances in Neural Information Processing Systems, Montreal, Canada (Montreal, Canada: Curran Associates, Inc.), 2672–2680. *NIPS'14*,.27

Gui, A., Gamper, H., Braun, S., and Emmanouilidou, D. (2023). Adapting frechet audio distance for generative music evaluation. arXiv [Preprint]. Available at http://arxiv.org/abs/2311.01616 (Accessed November 12, 2023).

Guo, H., Zhou, Z., Meng, F., and Liu, K. (2022a). "Improving adversarial waveform generation based singing voice conversion with harmonic signals," in ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (Singapore: ICASSP), 6657–6661. doi:10.1109/ICASSP43922.2022.9746709

Guo, Z., Chen, C., and Chng, E. S. (2022b). DENT-DDSP: data-efficient noisy speech generator using differentiable digital signal processors for explicit distortion modelling and noise-robust speech recognition. arXiv [Preprint]. Available at : https://arxiv.org/abs/2208.00987 (Accessed June 21, 2023).

Ha, D., Dai, A. M., and Le, Q. V. (2017). "HyperNetworks," in International Conference on Learning Representations.

Hagiwara, M., Cusimano, M., and Liu, J.-Y. (2022). Modeling animal vocalizations through synthesizers. arXiv [Preprint]. Available at https://arxiv.org/abs/.2210.10857 (Accessed August 02, 2023).

Han, H., Lostanlen, V., and Lagrange, M. (2023). "Perceptual–neural–physical sound matching," in ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (IEEE), 1–5.

Hayes, B., Saitis, C., and Fazekas, G. (2021). "Neural waveshaping synthesis," in Proceedings of the 22nd International Society for Music Information Retrieval Conference (Online).

Hayes, B., Saitis, C., and Fazekas, G. (2023). "Sinusoidal frequency estimation by gradient descent," in ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (Rhodes, Greece: ICASSP), 1–5. doi:10.1109/ICASSP49357.2023.10095188

Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, R. C., et al. (2017). "CNN architectures for large-scale audio classification," in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA (IEEE), 131–135. doi:10.1109/ICASSP.2017.7952132

Holmes, T. (2008). *Electronic and experimental music: technology, music, and culture*. 3rd edn. New York: Routledge.

Hono, Y., Takaki, S., Hashimoto, K., Oura, K., Nankaku, Y., and Tokuda, K. (2021). "Periodnet: a non-autoregressive waveform generation model with a structure separating periodic and aperiodic components," in ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada (IEEE), 6049–6053. doi:10.1109/ICASSP39728.2021.9414401

Horner, A., Beauchamp, J., and Haken, L. (1993). Machine tongues XVI. Genetic algorithms and their application to FM matching synthesis. *Comput. Music J.* 17, 17–29. doi:10.2307/3680541

Huang, S., Li, Q., Anil, C., Bao, X., Oore, S., and Grosse, R. B. (2019). "Timbretron: a wavenet(cycleGAN(CQT(audio))) pipeline for musical timbre transfer," in International Conference on Learning Representations.

Huang, W.-C., Violeta, L. P., Liu, S., Shi, J., and Toda, T. (2023). The singing voice conversion challenge 2023. arXiv [Prerint]. Available at https://arxiv.org/abs/2306.14422 (Accessed July 25, 2023).

Hunt, A., and Black, A. (1996). "Unit selection in a concatenative speech synthesis system using a large speech database," in 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings, 373–376. doi:10.1109/ICASSP.1996.5411101

Huzaifah, M., and Wyse, L. (2021). "Deep generative models for musical audio synthesis," in *Handbook of artificial intelligence for music: foundations, advanced approaches, and developments for creativity* (Springer), 639–678.

International Telecommunication Union (1996). *Methods for subjective determination of transmission quality*, (Geneva, Switzerland: ITU-T Recommendation).

International Telecommunication Union (2001). *Perceptual evaluation of speech quality (PESQ): an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs* (Geneva, Switzerland: ITU-R Recommendation).

International Telecommunication Union (2015). *Method for the subjective assessment of intermediate quality levels of coding systems* (Geneva, Switzerland: Recommendation ITU-R BS).

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). "Image-to-Image translation with conditional adversarial networks," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI (IEEE), 5967–5976. doi:10.1109/CVPR.2017.632

Ito, K., and Johnson, L. (2017). The LJ speech dataset. Available at https://keithito.com/LJ-Speech-Dataset/(Accessed October 25, 2023).

Jack, R. H., Mehrabi, A., Stockman, T., and McPherson, A. (2018). Action-sound latency and the perceived quality of digital musical instruments. *Music Percept.* 36, 109–128. doi:10.1525/mp.2018.36.1.109

Jin, Z., Finkelstein, A., Mysore, G. J., and Lu, J. (2018). "Fftnet: a real-time speaker-dependent neural vocoder," in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2251–2255. doi:10.1109/ICASSP.2018.8462431

Jonason, N., Sturm, B. L. T., and Thome, C. (2020). "The control-synthesis approach for making expressive and controllable neural music synthesizers," in Proceedings of the 2020 AI Music Creativity Conference.9

Juvela, L., Bollepalli, B., Yamagishi, J., and Alku, P. (2019). "GELP: GAN-excited linear prediction for speech synthesis from mel-spectrogram," in *Interspeech 2019* (Graz, Austria: ISCA), 694–698. doi:10.21437/Interspeech.2019-2008

Kalchbrenner, N., Elsen, E., Simonyan, K., Noury, S., Casagrande, N., Lockhart, E., et al. (2018). "Efficient neural audio synthesis," in Proceedings of the 35th International Conference on Machine Learning, (PMLR), 2410–2419.80

Kaneko, T., Tanaka, K., Kameoka, H., and Seki, S. (2022). "ISTFTNET: fast and lightweight mel-spectrogram vocoder incorporating inverse short-time fourier transform," in ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6207–6211. doi:10.1109/ICASSP43922.2022.9746713

Kato, H., Beker, D., Morariu, M., Ando, T., Matsuoka, T., Kehl, W., et al. (2020). Differentiable rendering: a survey. arXiv [Preprint]. Available at https://arxiv.org/abs/2006.12057 (Accessed August 21, 2023).

Kawamura, M., Nakamura, T., Kitamura, D., Saruwatari, H., Takahashi, Y., and Kondo, K. (2022). "Differentiable digital signal processing mixture model for synthesis parameter extraction from mixture of harmonic sounds," in ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, Singapore (IEEE), 941–945. doi:10.1109/ICASSP43922.2022.9746399

Keller, E. (1994). *Fundamentals of speech synthesis and speech recognition: basic concepts, state of the art, and future challenges*. Chichester [England]; New York: Wiley.

Khan, R. A., and Chitode, J. S. (2016). Concatenative speech synthesis: a review. *Int. J. Comput. Appl.* 136, 1–6. doi:10.5120/ijca2016907992

Kilgour, K., Zuluaga, M., Roblek, D., and Sharifi, M. (2019). "Fréchet audio distance: a reference-free metric for evaluating music enhancement algorithms," in *Interspeech 2019* (Graz, Austria: ISCA), 2350–2354. doi:10.21437/Interspeech.2019-2219

Kim, J. W., Salamon, J., Li, P., and Bello, J. P. (2018). "Crepe: a convolutional representation for pitch estimation," in 2018 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2018 - Proceedings (Calgary, Alberta, Canada: Institute of Electrical and Electronics Engineers Inc.), 161–165. doi:10.1109/ICASSP.2018.8461329

Kong, J., Kim, J., and Bae, J. (2020). "HiFi-GAN: generative adversarial networks for efficient and high fidelity speech synthesis," in *Advances in neural information processing systems*. Editors H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Red Hook, NY, United States: Curran Associates, Inc.), 33, 17022–17033.

Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. (2021). Diffwave: a versatile diffusion model for audio synthesis. In International Conference on Learning Representations

Kumar, K., Kumar, R., de Boissiere, T., Gestin, L., Teoh, W. Z., Sotelo, J., et al. (2019). *MelGAN: generative adversarial networks for conditional waveform synthesis*. Advances in Neural Information Processing Systems.

Kumar, R., Seetharaman, P., Luebs, A., Kumar, I., and Kumar, K. (2023). High-fidelity audio compression with improved RVQGAN. arXiv [Preprint]. Available at : https://arxiv.org/abs/2306.06546 (Accessed August 24, 2023).

Kuznetsov, B., Parker, J. D., and Esqueda, F. (2020). "Differentiable IIR filters for machine learning applications," in Proceedings of the 23rd International Conference on Digital Audio Effects.

Le Brun, M. (1979). Digital waveshaping synthesis. *J. Audio Eng. Soc.* 27, 250–266.

Lee, S., Choi, H.-S., and Lee, K. (2022). Differentiable artificial reverberation. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* 30, 2541–2556. doi:10.1109/TASLP.2022.3193298

Lee, S., Park, J., Paik, S., and Lee, K. (2023). "Blind estimation of audio processing graph," in ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece (IEEE), 1–5. doi:10.1109/ICASSP49357.2023.10096581

Liu, Z., Chen, K., and Yu, K. (2020). "Neural homomorphic vocoder," in *Interspeech 2020* (ISCA), 240–244. doi:10.21437/Interspeech.2020-3188

Manocha, P., Jin, Z., and Finkelstein, A. (2022). Audio similarity is unreliable as a proxy for audio quality. *Proc. Interspeech* 2022, 3553–3557. doi:10.21437/Interspeech.2022-405

Martinez Ramirez, M. A., Wang, O., Smaragdis, P., and Bryan, N. J. (2021). "Differentiable signal processing with black-box audio effects," in ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada (IEEE), 66–70. doi:10.1109/ICASSP39728.2021.9415103

Masuda, N., and Saito, D. (2021). "Synthesizer sound matching with differentiable DSP," in Proceedings of the 22nd International Society for Music Information Retrieval Conference (Online).

Masuda, N., and Saito, D. (2023). Improving semi-supervised differentiable synthesizer sound matching for practical applications. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* 31, 863–875. doi:10.1109/TASLP.2023.3237161

Matsubara, K., Okamoto, T., Takashima, R., Takiguchi, T., Toda, T., and Kawai, H. (2022). Comparison of real-time multi-speaker neural vocoders on CPUs. *Acoust. Sci. Technol.* 43, 121–124. doi:10.1250/ast.43.121

Michelashvili, M. M., and Wolf, L. (2020). "Hierarchical timbre-painting and articulation generation," in Proceedings of the 21th International Society for Music Information Retrieval Conference.

Mitcheltree, C., Steinmetz, C. J., Comunità, M., and Reiss, J. D. (2023). "Modulation extraction for LFO-driven audio effects," in Proceedings of the 26th International Conference on Digital Audio Effects, 94–101.

Moffat, D., and Sandler, M. B. (2019). Approaches in intelligent music production. *Arts* 1–13, 125. doi:10.3390/arts8040125

Mohammadi, S. H., and Kain, A. (2017). An overview of voice conversion systems. *Speech Commun.* 88, 65–82. doi:10.1016/j.specom.2017.01.008

Morise, M., Yokomori, F., and Ozawa, K. (2016). WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Trans. Inf. Syst.* E99, 1877–1884. doi:10.1587/transinf.2015EDP7457

Muradeli, J., Vahidi, C., Wang, C., Han, H., Lostanlen, V., Lagrange, M., et al. (2022). "Differentiable time–frequency scattering on GPU," in Proceedings of the 25th International Conference on Digital Audio Effects.

Murray, J., and Goldbart, J. (2009). Augmentative and alternative communication: a review of current issues. *Paediatr. Child Health* 19, 464–468. doi:10.1016/j.paed.2009.05.003

Mv, A. R., and Ghosh, P. K. (2020). SFNet: a computationally efficient source filter model based neural speech synthesis. *IEEE Signal Process. Lett.* 27, 1170–1174. doi:10.1109/LSP.2020.3005031

Nercessian, S. (2020). "Neural parametric equalizer matching using differentiable biquads," in Proceedings of the 23rd International Conference on Digital Audio Effects, Vienna, Austria, 8.

Nercessian, S. (2021). "End-to-End zero-shot voice conversion using a DDSP vocoder," in 2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA). 1–5. doi:10.1109/WASPAA52581.2021.9632754

Nercessian, S. (2023). *Differentiable WORLD synthesizer-based neural vocoder with application to end-to-end audio style transfer*. Audio Engineering Society Convention 154.

Nercessian, S., Sarroff, A., and Werner, K. J. (2021). "Lightweight and interpretable neural modeling of an audio distortion effect using hyperconditioned differentiable biquads," in ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON (Canada: IEEE), 890–894. doi:10.1109/ICASSP39728.2021.9413996

Nishimura, M., Hashimoto, K., Oura, K., Nankaku, Y., and Tokuda, K. (2016). "Singing voice synthesis based on deep neural networks," in *Interspeech 2016* (San Francisco, CA, USA: ISCA), 2478–2482. doi:10.21437/Interspeech.2016-1027

Oord, A., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., et al. (2018). "Parallel WaveNet: fast high-fidelity speech synthesis," in Proceedings of the 35th International Conference on Machine Learning (Stockholm, Sweden: PMLR), 3918–3926.

Polyak, A., Wolf, L., Adi, Y., and Taigman, Y. (2020). Unsupervised cross-domain singing voice conversion. *Proc. Interspeech*, 801–805. doi:10.21437/Interspeech.2020-1862

Pons, J., Pascual, S., Cengarle, G., and Serra, J. (2021). "Upsampling artifacts in neural audio synthesis," in ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada (IEEE), 3005–3009. doi:10.1109/ICASSP39728.2021.9414913

Prenger, R., Valle, R., and Catanzaro, B. (2019). "Waveglow: a flow-based generative network for speech synthesis," in ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, United Kingdom (IEEE), 3617–3621. doi:10.1109/ICASSP.2019.8683143

Ramachandran, P., Paine, T. L., Khorrami, P., Babaeizadeh, M., Chang, S., Zhang, Y., et al. (2017). "Fast generation for convolutional autoregressive models," in International Conference on Learning Representations (Workshop Track).

Ramírez, M. A., Benetos, E., and Reiss, J. D. (2020). Deep learning for black-box modeling of audio effects. *Appl. Sci. Switz.* 10, 638. doi:10.3390/app10020638

Ren, P., Xiao, Y., Chang, X., Huang, P.-y., Li, Z., Chen, X., et al. (2022). A comprehensive survey of neural architecture search: challenges and solutions. *ACM Comput. Surv.* 54, 1–34. doi:10.1145/3447582

Renault, L., Mignot, R., and Roebel, A. (2022). "Differentiable piano model for midi-to-audio performance synthesis," in Proceedings of the 25th International Conference on Digital Audio Effects, Vienna, Austria, 8.

Ribeiro, F., Florencio, D., Zhang, C., and Seltzer, M. (2011). "CROWDMOS: an approach for crowdsourcing mean opinion score studies," in 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republi: IEEE, 2416–2419. doi:10.1109/ICASSP.2011.5946971

Rodet, X. (2002). "Synthesis and processing of the singing voice," in Proc. 1st IEEE Benelux Workshop on Model Based Processing and Coding of Audio (MPCA-2002), 15–31.

Saino, K., Zen, H., Nankaku, Y., Lee, A., and Tokuda, K. (2006). "An HMM-based singing voice synthesis system," in Ninth International Conference on Spoken Language Processing. doi:10.21437/Interspeech.2006-584

Schulze-Forster, K., Richard, G., Kelley, L., Doire, C. S. J., and Badeau, R. (2023). Unsupervised music source separation using differentiable parametric source models. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* 31, 1276–1289. doi:10.1109/TASLP.2023.3252272

Schwartz, R., Dodge, J., Smith, N. A., and Etzioni, O. (2020). Green ai. *Commun. ACM* 63, 54–63. doi:10.1145/3381831

Schwarz, D. (2006). Concatenative sound synthesis: the early years. *J. New Music Res.* 35, 3–22. doi:10.1080/09298210600696857

Schwarz, D. (2007). "Corpus-based concatenative synthesis," in Conference Name: IEEE Signal Processing Magazine, 92–104. doi:10.1109/MSP.2007.32327424

Seeviour, P., Holmes, J., and Judd, M. (1976). "Automatic generation of control signals for a parallel formant speech synthesizer," in ICASSP '76. IEEE International Conference on Acoustics, Speech, and Signal Processing, 690–693. doi:10.1109/ICASSP.1976.11699871

Serra, X., and Smith, J. (1990). Spectral modeling synthesis: a sound analysis/synthesis system based on a deterministic plus stochastic decomposition. *Comput. Music J.* 14, 12–24. doi:10.2307/3680788

Shadle, C. H., and Damper, R. I. (2001). "Prospects for articulatory synthesis: a position paper," in 4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis.

Shan, S., Hantrakul, L., Chen, J., Avent, M., and Trevelyan, D. (2022). "Differentiable wavetable synthesis," in ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (Singapore: ICASSP), 4598–4602. doi:10.1109/ICASSP43922.2022.9746940

Shier, J., Caspe, F., Robertson, A., Sandler, M., Saitis, C., and McPherson, A. (2023). "Differentiable modelling of percussive audio with transient and spectral synthesis," in Proceedings of the 10th Convention of the European Acoustics Association Forum Acusticum 2023.

Shynk, J. (1989). Adaptive IIR filtering. *IEEE ASSP Mag.* 6, 4–21. doi:10.1109/53.29644

Sisman, B., Yamagishi, J., King, S., and Li, H. (2021). An overview of voice conversion and its challenges: from statistical modeling to deep learning. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* 29, 132–157. doi:10.1109/TASLP.2020.3038524

Smith, J. O. (1992). Physical modeling using digital waveguides. *Comput. Music J.* 16, 74. doi:10.2307/3680470

Smith, J. O. (2010). *Physical audio signal processing: for virtual musical instruments and audio effects*. Stanford, Calif: Stanford University, CCRMA.

Song, K., Zhang, Y., Lei, Y., Cong, J., Li, H., Xie, L., et al. (2023). "DSPGAN: a Gan-based universal vocoder for high-fidelity TTS by time-frequency domain supervision from DSP," in ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 1–5. doi:10.1109/ICASSP49357.2023.10095105

Spall, J. C. (1998). An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins Apl. Tech. Dig.* 19, 482–492.

Ssergejewitsch, T. L. (1928). *Method of and apparatus for the generation of sounds*. U.S. Patent No. US1661058A.

Stanton, D., Shannon, M., Mariooryad, S., Skerry-Ryan, R. J., Battenberg, E., Bagby, T., et al. (2022). "Speaker generation," in ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 7897–7901. doi:10.1109/ICASSP43922.2022.9747345

Steinmetz, C. J., Bryan, N. J., and Reiss, J. D. (2022a). Style transfer of audio effects with differentiable signal processing. *J. Audio Eng. Soc.* 70, 708–721. doi:10.17743/jaes.2022.0025

Steinmetz, C. J., and Reiss, J. D. (2020). "auraloss: audio focused loss functions in PyTorch," in Digital music research network one-day workshop (DMRN+15).

Steinmetz, C. J., Vanka, S. S., Martínez Ramírez, M. A., and Bromham, G. (2022b). Deep learning for automatic mixing (ISMIR). Available at https://dl4am.github.io/tutorial (Accessed October 25, 2023).

Stylianou, Y. (2009). "Voice transformation: a survey," in 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, 3585–3588. doi:10.1109/ICASSP.2009.4960401

Subramani, K., Valin, J.-M., Isik, U., Smaragdis, P., and Krishnaswamy, A. (2022). "End-to-end LPCNet: a neural vocoder with fully-differentiable LPC estimation," in Interspeech 2022 (ISCA), 818–822. doi:10.21437/Interspeech.2022-912

Südholt, D., Cámara, M., Xu, Z., and Reiss, J. D. (2023). "Vocal tract area estimation by gradient descent," in Poceedings of the 26th International Conference on Digital Audio Effects.

Tamamori, A., Hayashi, T., Kobayashi, K., Takeda, K., and Toda, T. (2017). "Speaker-dependent wavenet vocoder," in Interspeech 2017, 1118–1122. doi:10.21437/Interspeech.2017-314

Tan, X., Qin, T., Soong, F., and Liu, T. Y. (2021). A survey on neural speech synthesis. arXiv [Preprint]. Available at: https://arxiv.org/abs/2106.15561 (Accessed July 07, 2023).

Tian, Q., Zhang, Z., Lu, H., Chen, L.-H., and Liu, S. (2020). "FeatherWave: an efficient high-fidelity neural vocoder with multi-band linear prediction," in Proceedings of Interspeech 2020, 195–199. doi:10.21437/Interspeech.2020-1156

Turian, J., and Henry, M. (2020). *I'm sorry for your loss: spectrally-based audio distances are bad at pitch*. I Can't Believe It's Not Better!" NeurIPS 2020 workshop.

Turian, J., Shier, J., Tzanetakis, G., McNally, K., and Henry, M. (2021). One billion audio sounds from GPU-enabled modular synthesis. In Proceedings of the 23rd International Conference on Digital Audio Effects

Valin, J.-M., and Skoglund, J. (2019). "LPCNET: improving neural speech synthesis through linear prediction," in ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (Brighton, UK: ICASSP), 5891–5895. doi:10.1109/ICASSP.2019.8682804

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., et al. (2016). WaveNet: a generative model for raw audio. arXiv [Preprint]. Available at: https://arxiv.org/abs/1609.03499 (Accessed August 08, 2023).

Vinay, A., and Lerch, A. (2022). "Evaluating generative audio systems and their metrics," in Proceedings of the 23rd International Society for Music Information Retrieval Conference.

Vipperla, R., Park, S., Choo, K., Ishtiaq, S., Min, K., Bhattacharya, S., et al. (2020). "Bunched LPCNet: vocoder for low-cost neural text-to-speech systems," in Proceedings of Interspeech 2020, 3565–3569. doi:10.21437/Interspeech.2020-2041

Wagner, P., Beskow, J., Betz, S., Edlund, J., Gustafson, J., Eje Henter, G., et al. (2019). "Speech synthesis evaluation — state-of-the-art assessment and suggestion for a novel research program," in 10th ISCA Workshop on Speech Synthesis (SSW 10) (ISCA), 105–110. doi:10.21437/SSW.2019-19

Wang, X., Takaki, S., and Yamagishi, J. (2019a). "Neural source-filter-based waveform model for statistical parametric speech synthesis," in ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, United Kingdom (IEEE), 5916–5920. doi:10.1109/ICASSP.2019.8682298

Wang, X., Takaki, S., and Yamagishi, J. (2019b). Neural source-filter waveform models for statistical parametric speech synthesis. *IEEE/ACM Trans. Audio, Speech, Lang. Process.* 28, 402–415. doi:10.1109/TASLP.2019.2956145

Wang, X., and Yamagishi, J. (2019). "Neural harmonic-plus-noise waveform model with trainable maximum voice frequency for text-to-speech synthesis," in 10th ISCA Workshop on Speech Synthesis (SSW 10) (Brighton, United Kingdom: ISCA), 1–6. doi:10.21437/SSW.2019-128

Wang, X., and Yamagishi, J. (2020). "Using cyclic noise as the source signal for neural source-filter-based speech waveform model," in Proceedings of Interspeech 2020, 1992–1996. doi:10.21437/Interspeech.2020-1018

Wang, Y., Wang, X., Zhu, P., Wu, J., Li, H., Xue, H., et al. (2022). Opencpop: a high-quality open source Chinese popular Song corpus for singing voice synthesis. arXiv [Preprint]. Available at: https://arxiv.org/abs/2201.07429 (Accessed July 24, 2023).

Watts, O., Wihlborg, L., and Valentini-Botinhao, C. (2023). "PUFFIN: pitch-synchronous neural waveform generation for fullband speech on modest devices," in ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (Rhodes Island, Greece (IEEE), 1–5. doi:10.1109/ICASSP49357.2023.10094729

Webber, J. J., Valentini-Botinhao, C., Williams, E., Henter, G. E., and King, S. (2023). "Autovocoder: fast waveform generation from a learned speech representation using differentiable digital signal processing," in ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1–5. doi:10.1109/ICASSP49357.2023.10095729

Wester, M., Wu, Z., and Yamagishi, J. (2016). "Analysis of the voice conversion challenge 2016 evaluation results," in Interspeech 2016 (ISCA), 1637–1641. doi:10.21437/Interspeech.2016-1331

Wu, D.-Y., Hsiao, W.-Y., Yang, F.-R., Friedman, O., Jackson, W., Bruzenak, S., et al. (2022a). "DDSP-based singing vocoders: a new subtractive-based synthesizer and A comprehensive evaluation," in Proceedings of the 23rd International Society for Music Information Retrieval Conference, 76–83.

Wu, Y., Gardner, J., Manilow, E., Simon, I., Hawthorne, C., and Engel, J. (2022b). "Generating detailed music datasets with neural audio synthesis," in Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA.162

Wu, Y., Manilow, E., Deng, Y., Swavely, R., Kastner, K., Cooijmans, T., et al. (2022c). "MIDI-DDSP: detailed control of musical performance via hierarchical modeling," in International Conference on Learning Representations.

Yamamoto, R., Song, E., and Kim, J.-M. (2020). "Parallel wavegan: a fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain (IEEE), 6199–6203. doi:10.1109/ICASSP40776.2020.9053795

Yang, L.-C., and Lerch, A. (2020). On the evaluation of generative models in music. *Neural Comput. Appl.* 32, 4773–4784. doi:10.1007/s00521-018-3849-7

Ye, Z., Xue, W., Tan, X., Liu, Q., and Guo, Y. (2023). "NAS-FM: neural architecture search for tunable and interpretable sound synthesis based on frequency modulation," in Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, 5869–5877. doi:10.24963/ijcai.2023/651

Yee-King, M. J., Fedden, L., and d'Inverno, M. (2018). Automatic programming of VST sound synthesizers using deep networks and other techniques. *IEEE Trans. Emerg. Top. Comput. Intell.* 2, 150–159. doi:10.1109/TETCI.2017.2783885

Yoshimura, T., Takaki, S., Nakamura, K., Oura, K., Hono, Y., Hashimoto, K., et al. (2023). "Embedding a differentiable mel-cepstral synthesis filter to a neural speech synthesis system," in ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 1–5. doi:10.1109/ICASSP49357.2023.10094872

You, J., Kim, D., Nam, G., Hwang, G., and Chae, G. (2021). "GAN vocoder: multi-resolution discriminator is all you need," in Proceedings of Interspeech 2021, 2177–2181. doi:10.21437/Interspeech.2021-41

Yu, C., Lu, H., Hu, N., Yu, M., Weng, C., Xu, K., et al. (2020). "DurIAN: duration informed attention network for speech synthesis," in Proceedings of Interspeech 2020, 2027–2031. doi:10.21437/Interspeech.2020-2968

Yu, C.-Y., and Fazekas, G. (2023). Singing voice synthesis using differentiable LPC and glottal-flow-inspired wavetables. arXiv [Preprint]. Available at https://arxiv.org/abs/2306.17252 (Accessed July 05, 2023).

Zen, H., Tokuda, K., and Black, A. W. (2009). Statistical parametric speech synthesis. *Speech Commun.* 51, 1039–1064. doi:10.1016/j.specom.2009.04.004

Zhao, Y., Wang, X., Juvela, L., and Yamagishi, J. (2020). "Transferring neural speech waveform synthesizers to musical instrument sounds generation," in ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain (IEEE), 6269–6273. doi:10.1109/ICASSP40776.2020.9053047