THE UNIVERSITY of EDINBURGH

Edinburgh Research Explorer

Autovocoder: fast waveform generation from a learned speech representation using differentiable digital signal processing

OPEN ACCESS

AUTOVOCODER: FAST WAVEFORM GENERATION FROM A LEARNED SPEECH REPRESENTATION USING DIFFERENTIABLE DIGITAL SIGNAL PROCESSING

Jacob J Webber[1], Cassia Valentini-Botinhao[1], Evelyn Williams[1], Gustav Eje Henter[2], Simon King[1]

[1]The Centre for Speech Technology Research, University of Edinburgh, United Kingdom
[2]Division of Speech, Music, and Hearing, KTH Royal Institute of Technology, Stockholm, Sweden

ABSTRACT

Most state-of-the-art Text-to-Speech systems use the mel-spectrogram as an intermediate representation, to decompose the task into acoustic modelling and waveform generation.

A mel-spectrogram is extracted from the waveform by a simple, fast DSP operation, but generating a high-quality waveform from a mel-spectrogram requires computationally expensive machine learning: a neural vocoder. Our proposed "autovocoder" reverses this arrangement. We use machine learning to obtain a representation that replaces the mel-spectrogram, and that can be inverted back to a waveform using simple, fast operations including a differentiable implementation of the inverse STFT.

The autovocoder generates a waveform 5 times faster than the DSP-based Griffin-Lim algorithm, and 14 times faster than the neural vocoder HiFi-GAN. We provide perceptual listening test results to confirm that the speech is of comparable quality to HiFi-GAN in the copy synthesis task.

Index Terms: speech synthesis, neural vocoder, differentiable DSP, representation learning

1. INTRODUCTION

Generating a natural-sounding speech waveform is a challenging task. For this reason, text-to-speech (TTS) is usually divided into *acoustic modelling* to map input text to an intermediate acoustic representation, which is then input to a *waveform generator*. The acoustic representation is, – in almost all state-of-the-art systems – a mel-spectrogram. The frame rate of this spectrogram is much lower than the waveform sampling rate, which simplifies the task for the acoustic model but creates a challenge for the waveform generator. Other tasks, such as Voice Conversion (VC), also involve the generation of a waveform from a spectrogram.

Beginning with WaveNet [1], deep learning has achieved extremely natural-sounding waveform generation, but many

---

Table 1. Comparing encoding and decoding of the proposed system with the current state-of-the-art (SOTA).

System	Acoustic feature extraction (encoder)	Waveform generation (decoder)
SOTA	fast DSP	slow neural network
Autovocoder	fast DDSP + simple neural network	simple neural network + fast DDSP

such *neural vocoders* are autoregressive (to model the auto-correlated nature of speech waveforms) which is computationally expensive, not amenable to parallelisation, and precludes on-device synthesis. More recent neural vocoders reduce computational cost either by leveraging knowledge of speech production, for example with the use of traditional signal-processing ideas in Neural Source-Filter (NSF) models [2] and LPCNet [3], or by making use of generative modelling to remove the autoregression, as in HiFi-GAN [4].

Our proposed system, *autovocoder*, generates high-quality waveforms very efficiently from a frame-based representation of similar size to the mel-spectrogram used in a typical TTS or VC system. The system uses machine learning to define that acoustic representation, then leverages fast, differentiable DSP (DDSP) operations for decoding. This reverses the conventional arrangement, where fast signal processing is used for the one-off task of encoding waveforms into acoustic representations (mel-spectrograms), forcing deployed systems to use slow, difficult to parallelise, and computationally expensive models to generate from these features every time a waveform is synthesised: Table 1.

Autovocoder combines the respective strengths of DSP and deep learning. DSP is fast and efficient, while machine learning can yield rich and maximally informative representations (for a given dimensionality). The mel-spectrogram only represents spectral magnitudes and discards phase, even though this is known to be beneficial for both speech recognition [5] and speech synthesis [6]. The learned representation of *autovocoder* is not required to discard phase. For signal generation, we use the inverse discrete short-time Fourier transform (iSTFT) and overlap-add, which can be executed extremely efficiently on modern hardware. Experimental results show that the proposed approach offers high-quality speech waveform generation whilst also being extremely fast.

2. BACKGROUND & RELATED WORK

2.1. Parallelism and autoregression

The rise of machine learning for image, speech, and language processing has been enabled by rapid advances in parallel computing performance. Most modern machine learning is highly parallelisable, but algorithms parallelise to different degrees. Autoregressive sequence models require values from prior steps in the computation: output must be computed sequentially and not in parallel. Conversely, *embarrassingly* parallel algorithms require no communication between decomposable computations. Many recent TTS acoustic models are not autoregressive [7], [8]; they are fast because a sequence can be processed in parallel without depending on prior steps. Non-autoregressive waveform synthesis has proved more elusive. Autoregression is a straightforward simple approach for modelling between-sample dependencies, and was used by the first machine-learned waveform generator, WaveNet [9], [10]. WaveNet suffered very significant performance issues, which were to some extent addressed by Parallel WaveNet [11]. More recent systems like WaveRNN [12] have achieved better computational performance through more strategic use of autoregressive elements and careful engineering. Even more recently, sophisticated generative models such as GANs have enabled parallel waveform synthesis [4]. However, these systems still rely on learned methods to generate highly correlated signal samples, which is inefficient with respect to both training data and compute.

2.2. Vocoding based on explicit harmonic synthesis

Some recent systems have considered generating waveforms with appropriate correlations by explicitly generating the periodic content. The Neural Source-Filter (NSF) model [2], employs a series of sinewave generators that generate harmonics. This requires a known input f_0. Artefacts for speech with less harmonicity in the source (e.g., breathy voice) were addressed in [13] by using cyclic noise rather than sinewaves for the source. Another important example of DSP-informed waveform synthesis for TTS is LPCNet [3], which employs an RNN to drive a waveform generator based on Linear Predictive Coding. The speech signal is encoded into 20 DSP-derived parameters at a frame rate of $10\,\mathrm{ms}$. Both NSF and LPCNet differ from the work in this paper by continuing to depend on DSP-derived features.

While iSTFTNet [14], developed in parallel with our work, also uses the iSTFT for synthesis, it has key differences in that it employs a much smaller FFT and hop size (16 vs. 1024 and 4 vs. 256 respectively), which is likely to limit computational performance. These settings may be necessary because it still depends on the mel-spectrogram. Puffin [15], another vocoder developed in parallel to ours, relies on a pitch synchronous iSTFT operation to support low-complexity higher sampling-rate generation.

Recent work, such as wav2vec 2.0 [16], has used self-supervised learned audio representations. These have been used in TTS [17], [18], but they necessitate a *more* complex waveform decoder, plus f_0 input. Our representations are designed to be efficiently invertible.

Autovocoder is trained as an autoencoder. Other systems have used autoencoders for unsupervised speech feature learning [19]; however, they emphasise the use of the features for other tasks, and are not designed to perform very fast decoding back to a waveform. Unlike neural speech codecs like [20], we do not target extremely low bit rates, but rather a frame-based representation that could directly replace the mel-spectrogram in applications including TTS.

3. THE PROPOSED SYSTEM

Autovocoder is an autoencoder trained on speech waveforms. The aim is to learn a representation that can replace the conventional, signal processing-based mel-spectrogram. In the current work, we use a dimensionality and frame rate that is comparable to a typical mel-spectrogram, for fair comparison in our experiments. Our focus in this paper is on an architecture whose learned representation can be decoded back to a waveform with very low computational cost.

3.1. Encoder and decoder

The encoder converts from time domain to frequency domain using a differentiable implementation of the STFT. The resulting complex spectrum is then used to derive four spectral components: magnitude, phase, real and imaginary. This use of redundant components is discussed in Section 3.2. These four spectral components are stacked, with each treated as a channel, and fed into a purely convolutional residual network made up of what we term *basic blocks*. Each such block consists of two 2D convolutional layers of width 3, followed by a 2D batch norm and a ReLU nonlinearity. This basic block also applies a residual, by summing the input with the output, but only if the number of input channels is the same as the number of output channels.

The residual net used in the current *autovocoder* architecture consists of 11 basic blocks, the first five having 4 input and output channels, the middle one having 4 input channels but 1 output channel, with the remaining 5 blocks having 1 channel in and out. That single channel output is fed into a single linear layer that reduces the dimensionality per timestep from $(\mathrm{windowsize})/2 + 1$ to our *representation size*. This is the dimensionality of a single frame of the learned representation, which we chose to be similar to the frequency dimension of a typical mel-spectrogram in the experiments.

Our decoder architecture is a mirror of the encoder, with the components applied in reverse order and with similar hyperparameters. The full architecture is shown in Fig. 1. Note that the system has no autoregression at any level. All frames
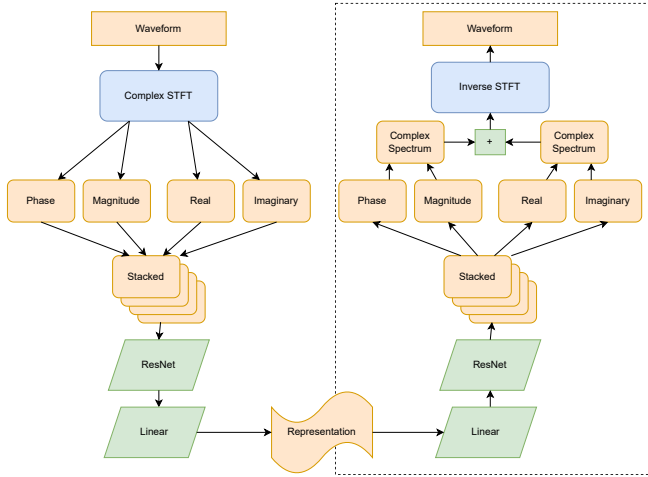
Fig. 1. Autovocoder architecture. Dashed box shows decoder

are processed at once by the network, and the subsequent overlap-add to assemble the complete waveform is performed by the differentiable iSTFT function of PyTorch [21].

3.2. Training regime

The model is trained as a denoising autoencoder [22]. Dropout is used on the embedding during training to increase decoder robustness, with a view to subsequent applications where these features would be generated by a TTS acoustic model. *Autovocoder* training adopts the losses from HiFi-GAN [4]: a mel-spectrogram loss and two adversarial losses from multi-scale and multi-period discriminators. In addition, a time-domain loss (per-sample squared error) is added, to improve phase reconstruction. If the weight of this term is too low, highly audible phase artefacts result.

3.3. Redundant representations of complex numbers

Providing redundant representations of the complex spectrogram to the encoder allows the model to learn how best to represent magnitude and phase. For example, *autovocoder* could potentially learn that, given a magnitude spectrogram, phase can be very efficiently represented [23].

For the decoder, three representations of the complex spectrogram were considered. For Cartesian and polar, the network generates two output channels: real and imaginary, or magnitude and phase, respectively. In a third method the network generates all four output channels, then the Cartesian mean of both complex forms is taken. Inspecting the ratio between the magnitudes of the polar and Cartesian outputs during autoencoding suggested that phase is more easily modelled in Cartesian form. Training with only polar output yielded poorer sound quality, as did the third method using all four output channels.

4. EVALUATION

We compared the performance of *autovocoder* with two other waveform generators – Griffin-Lim and HiFi-GAN – in output quality and computational performance.

Griffin-Lim [24] is an iterative algorithm for phase recovery that converts magnitude spectrograms into phase-coherent time-domain signals. We used the librosa [25] implementation of Griffin-Lim, which uses 32 iterations. This was used with the same frame rate (i.e. hop size) and window duration as the STFT used by *autovocoder*. Griffin-Lim uses full resolution ground truth magnitude spectrograms (513 bins), without mel-scale dimensionality reduction.

An open source implementation of HiFi-GAN[1] was used with a pretrained checkpoint that was trained on the same dataset as *autovocoder* using the the *V1* and *V3* generators. The *V1* generator is designed to achieve maximum output quality, and the *V3* generator is designed to run much faster, at the expense of some output quality.

The configuration of *autovocoder* evaluated here was determined using a combination of informal listening and the loss on a validation set. The STFT and iSTFT hop size was 256 samples (12 ms) with a window size of 1024 samples (46 ms).

Unlike a neural speech codec such as [20], there is no intrinsic need for a small representation size in *autovocoder*, since we are not aiming for a low bitrate. Nevertheless, here we use modest representation dimensionalities of 128, 192 and 256, chosen to be comparable to the internal representations in current TTS models [7], [26], with a view to a future TTS application of *autovocoder*.

Autovocoder and HiFi-GAN were trained on the single-speaker LJ Speech [27] dataset with waveforms at a sample rate of 22.05 kHz. We used a pretrained model for HiFi-GAN that had been trained to 2.3 million steps. *Autovocoder* was trained with a dropout factor of 10% and the Adam optimiser for approximately 1 million steps (limited by available compute). Evaluations used a test section of the dataset, unseen during training, and identical for HiFi-GAN and *autovocoder*.

In this paper, we evaluate copy synthesis. In the case of *autovocoder* this means a complete pass through both the encoder and decoder parts of the system. For the comparison systems, it means synthesising from ground truth spectral features extracted from waveforms using DSP.

4.1. Listening test

The system was compared to relevant baselines using a MUSHRA [28] style evaluation. Each MUSHRA screen presented 7 stimuli to the listener for evaluation. These were *autovocoder* (*AV 128*, *AV 192*, *AV 256*), HiFi-GAN (*HG V1*, *HG V3*), Griffin-Lim (*GL*) and the original ground-truth (*GT*) waveform.

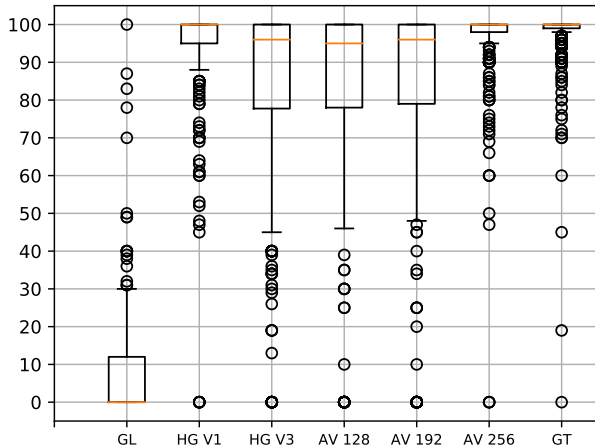[1]https://github.com/jik876/hifi-gan

Fig. 2. Results of the MUSHRA evaluation.

45 listeners were recruited using Prolific[2]. A Qualtrics survey was generated using the *Qualtreats* tool[3] which comprise 60 MUSHRA screens, split evenly into 3 tests, with each test assigned 15 listeners. Each screen presented the reference. Participants were then instructed to find the reference within those seven samples and assign it a score of 100, whilst rating all samples.

In line with the MUSHRA specification, participants who rated GT less than 90 in more than 15% screens were excluded. This left 17 respondents and 340 MUSHRA screens for analysis. This large number of exclusions was a result of the high difficulty of distinguishing between GT, AV 256 and HG V1. Analysis without these exclusions yielded broadly similar results. Figure 2 shows the results. A pairwise t-test was used to evaluate statistical significance. Analysis yielded three sets of systems, {GL}, {HG V3, AV 128, AV 192} and {HG V1, AV 256, GT}. With $p > 0.01$ all systems were significantly different from those not in their own set, and there were no significant differences within sets.

Samples from all systems are available online[4].

4.2. Computational cost

The computational performance of the *autovocoder* decoder was compared with several waveform generation systems. Each system was timed generating individual utterances (with no batching) on a CPU. Such utterance-by-utterance processing represents many typical use cases, such as on-device synthesis. The systems chosen were Griffin-Lim and HiFi-GAN as used in the listening test, plus LPCNet and WaveRNN[5]. A C implementation of LPCNet was used[6], which

---

[2] https://www.prolific.co/
[3] https://github.com/CSTR-Edinburgh/qualtreats
[4] https://jacobjwebber.github.io/autovocoder
[5] https://github.com/fatchord/WaveRNN
[6] https://github.com/xiph/LPCNet

Table 2. Speed of each system. Higher values indicate faster generation; a value below 1 indicates slower than real time.

System	Real-time factor
Griffin-Lim	18.43
Autovocoder 256,192,128	102.01, 101.34, 101.08
HiFi-GAN *V1, V3*	6.76, 42.18
WaveRNN	0.47
LPCNet	1.50

may give it a performance advantage over the Python/PyTorch implementation of *autovocoder*.

Timings were measured on a high-end 10 core desktop workstation processor (Intel i9-10850K). The value given in Table 2 was calculated by dividing the total duration of the generated audio by the total time taken to generate it. The measurement was for generating the test set of ten samples ten times. An average was taken of three runs. The test set contained a range of waveform duration from $2.0\,\text{s}$ to $9.7\,\text{s}$.

LPCNet was not able to exploit the parallelism of the processor, and is therefore likely to perform relatively better on a system with fewer available cores. Time spent loading models from the file system was not found to be significant for any of the systems under test.

As shown in Table 2, *autovocoder* generates a waveform many times faster than the other waveform generators. Performance improvements of *autovocoder* over autoregressive systems should be even larger on hyper-parallel architectures such as GPUs.

5. FUTURE APPLICATION TO TTS

There are two primary ways that *autovocoder* could be incorporated into a text-to-speech system. The first is by using the decoder architecture as the waveform generator within an end-to-end system. The second is by using *autovocoder* to extract representations from waveforms, and using a TTS acoustic model to generate such representations instead of mel-spectrograms. The waveform could then be generated very quickly by the *autovocoder* decoder.

6. CONCLUSION

We propose an alternative to typical neural vocoders. *Autovocoder* generates high-quality audio very quickly: it can reach the same quality as HiFi-GAN in fewer training updates, and is up to 16 times faster during generation. We intend that this system be used for the typical tasks of a neural vocoder, such as speech coding and speech synthesis. Verifying that speech synthesizers can accurately generate *autovocoder* features remains as important future work.

7. REFERENCES

[1] J. Shen, R. Pang, *et al.*, "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in *Proc. ICASSP*, 2018, pp. 4779–4783.

[2] X. Wang, S. Takaki, and J. Yamagishi, "Neural source-filter waveform models for statistical parametric speech synthesis," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 28, pp. 402–415, 2020.

[3] J.-M. Valin and J. Skoglund, "LPCNet: Improving neural speech synthesis through linear prediction," in *Proc. ICASSP*, 2019, pp. 5891–5895.

[4] J. Kong, J. Kim, and J. Bae, "HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Proc. NeurIPS*, vol. 33, 2020, pp. 17 022–17 033.

[5] E. Loweimi, Z. Cvetkovic, *et al.*, "Speech acoustic modelling from raw phase spectrum," in *Proc. ICASSP*, 2021, pp. 6738–6742.

[6] F. Espic, C. V. Botinhao, and S. King, "Direct modelling of magnitude and phase spectra for statistical parametric speech synthesis," in *Proc. Interspeech*, 2017, pp. 1383–1387.

[7] A. Łańcucki, "FastPitch: Parallel text-to-speech with pitch prediction," in *Proc. ICASSP*, 2021, pp. 6588–6592.

[8] Y. Ren, Y. Ruan, *et al.*, "FastSpeech: Fast, robust and controllable text to speech," in *Proc. NeurIPS*, vol. 32, 2019.

[9] A. van den Oord, S. Dieleman, *et al.*, "WaveNet: A generative model for raw audio," *arXiv:1609.03499*, 2016.

[10] A. Tamamori, T. Hayashi, *et al.*, "Speaker-dependent WaveNet vocoder," in *Proc. Interspeech*, 2017, pp. 1118–1122.

[11] A. van den Oord, Y. Li, *et al.*, "Parallel WaveNet: Fast high-fidelity speech synthesis," in *Proc. ICML*, vol. 80, 2018, pp. 3918–3926.

[12] N. Kalchbrenner, E. Elsen, *et al.*, "Efficient neural audio synthesis," in *Proc. ICML*, vol. 80, 2018, pp. 2410–2419.

[13] X. Wang and J. Yamagishi, "Using cyclic noise as the source signal for neural source-filter-based speech waveform model," in *Proc. Interspeech*, 2020, pp. 1992–1996.

[14] T. Kaneko, K. Tanaka, *et al.*, "iSTFTNet: Fast and lightweight mel-spectrogram vocoder incorporating inverse short-time Fourier transform," in *Proc. ICASSP*, 2022, pp. 6207–6211.

[15] O. Watts, L. Wihlborg, and C. Valentini-Botinhao, "Puffin: Pitch-synchronous neural waveform generation for fullband speech on modest devices," in *Proc. ICASSP*, 2023.

[16] A. Baevski, H. Zhou, *et al.*, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. NeurIPS*, 2020.

[17] C. Du, Y. Guo, *et al.*, "VQTTS: High-fidelity text-to-speech synthesis with self-supervised VQ acoustic feature," in *Proc. Interspeech*, 2022, pp. 1596–1600.

[18] H. Siuzdak, P. Dura, *et al.*, "WavThruVec: Latent speech representation as intermediate features for neural speech synthesis," in *Proc. Interspeech*, 2022, pp. 833–837.

[19] J. Chorowski, R. J. Weiss, *et al.*, "Unsupervised speech representation learning using WaveNet autoencoders," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 27, no. 12, 2041–2053, 2019.

[20] N. Zeghidour, A. Luebs, *et al.*, "SoundStream: An end-to-end neural audio codec," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 30, 495–507, 2022.

[21] A. Paszke, S. Gross, *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. NeurIPS*, 2019, pp. 8024–8035.

[22] P. Vincent, H. Larochelle, *et al.*, "Extracting and composing robust features with denoising autoencoders," in *Proc. ICML*, 2008, pp. 1096–1103.

[23] P. Van Hove, M. Hayes, *et al.*, "Signal reconstruction from signed Fourier transform magnitude," *IEEE Trans. Acoustics, Speech, and Sig. Proc.*, vol. 31, no. 5, pp. 1286–1293, 1983.

[24] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Trans. Acoustics, Speech, and Sig. Proc.*, vol. 32, no. 2, pp. 236–243, 1984.

[25] B. McFee, C. Raffel, *et al.*, "librosa: Audio and music signal analysis in Python," in *Proc. Python Sci. Conf.*, vol. 8, 2015.

[26] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *Proc. ICML*, 2021, pp. 5530–5540.

[27] K. Ito and L. Johnson, *The LJ Speech dataset*, `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[28] ITU-R, *Method for the subjective assessment of intermediate quality levels of coding systems*, ITU Recommendation ITU-R BS.1534-3, 2015.