10th International Conference on Speech Prosody 2020
25-28 May 2020, Tokyo, Japan

Modelling Intonation in Spectrograms for Neural Vocoder based Text-to-Speech

Vincent Wan, Jonathan Shen, Hanna Silen, Rob Clark

Google

{vwan,jonathanasdf,silen,rajclark}@google.com

Abstract

Intonation is characterized by rises and falls in pitch and energy. In previous work, we explicitly modelled these prosodic features using Clockwork Hierarchical Variational Autoencoders (CHiVE) to show we can generate multiple intonation contours for any text. However, recent advances in text-to-speech synthesis produce spectrograms which are inverted by neural vocoders to produce waveforms. Spectrograms encode intonation in a complex way; there is no simple, explicit representation analogous to pitch (fundamental frequency) and energy. In this paper, we extend CHiVE to model intonation within a spectrogram. Compared to the original model, the spectrogram extension gives better mean opinion scores in subjective listening tests. We show that the intonation in the generated spectrograms match the intonation represented by the generated pitch curves.
Index Terms: text-to-speech synthesis, prosody, WaveNet, neural vocoder

1. Introduction

Of the many different aspects to prosody, including emotion, style and intonation, this paper focuses on modelling intonation and specifically on the ability to generate pitch, timing and energy to realize variation in semantics and intent. This contrasts other recent approaches to modelling prosody in text-to-speech (TTS) which focus more on aspects of prosody that can be considered speaking style [1, 2] or emotion [3].

This difference is partly achieved by the choice of training data, in that we use a consistent single style or, where there are multiply styles present, we condition explicitly on those styles in our model, leaving only the potential differences in intonation to be modelled by the latent part of our model. The model structure itself reinforces the ability to capture this variability by directly using the linguistic structure of a given utterance along with shallow syntactic information.

This paper extends the Clockwork Hierarchical Variational Autoencoder (CHiVE) prosody model [4] (Section 2) by adding a Mel-spectrogram generating component that may be used to directly drive a neural vocoder. Similar to Tacotron 2 [5], this allows us to generate high quality speech without the difficulties and computation cost of training and running a full WaveNet [6] style model. It simultaneously allows us to keep control of the prosody in ways that 'end-to-end' approaches used by Tacotron and its derivatives do not allow.

We do not subscribe to any specific prosodic theory in this work. The only assumptions we make are that the prosody is realized by changes to individual syllables (think of these as local pitch events), and that there is an overall controlling representation at the utterance level (think of this as intonational tune). Additionally we do not use any specific representation, symbolic (e.g. ToBI [7] or PoLaR [8]) or parametric (e.g. PENTA [9]) to represent prosody. We model pitch duration and energy directly, with an intention to later add conditioning via higher level labels that represent semantics or pragmatics.

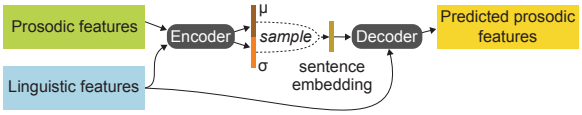

Figure 1: *A high level overview of CHiVE model.*

2. CHiVE Prosody model

Here we describe the CHiVE prosody model [4]; Section 2.1 describes the extensions to the model.

Figure 1 shows an overview of the model topology. It is a conditional variational autoencoder neural network that jointly models the three main features of prosody: pitch ($\log F_0$), energy (approximated by c_0, the zero-th order cepstral coefficient) and duration (at the phone level). CHiVE consists of two parts: an encoder part that converts the prosodic features to a fixed length vector representation (the *sentence embedding*); a decoder part that converts the *sentence embedding* back into prosodic features.

For a given sentence there are multiple valid prosodic renditions. The sentence embedding is a vector in a learned latent space that captures information not present in the text. This embedding is the key component that enables the decoder to produce a range of different prosodic renditions for any sentence. A *variational* autoencoder [10] additionally imposes a distribution upon the embeddings so that it becomes possible to sample from the latent space. In this paper we impose a unit Gaussian distribution (we return to this in Section 2.1). Other options are possible, e.g. [11], but these are beyond the scope of this paper.

Statistical parametric TTS systems necessarily model speech parameters at short time intervals (typically 5ms but even smaller for recent end-to-end approaches such as WaveNet [12] and variants of Tacotron [5, 13] that produce waveform samples directly). However, it has been long understood that prosody is better modelled at longer time scales and studies suggest that it is best modelled at the syllable level [14]. We incorporate this knowledge into the topology of the CHiVE model by using layers that explicitly represent syllables within both the encoder and decoder networks. In addition to the underlying speech parameterization that changes at every frame, a sentence also contains information that changes per word or per phone. This motivates the clockwork aspect of the model where different layers of the network are clocked at intervals corresponding to transitions between words, syllables, phones and frames; the temporal unrolling of the recurrent neural network is dictated by the linguistic hierarchy of the given sentence.

Figure 2 illustrates the encoder model: a frame-rate recurrent neural network (RNN) is fed frame-level acoustic features

10.21437/SpeechProsody.2020-193

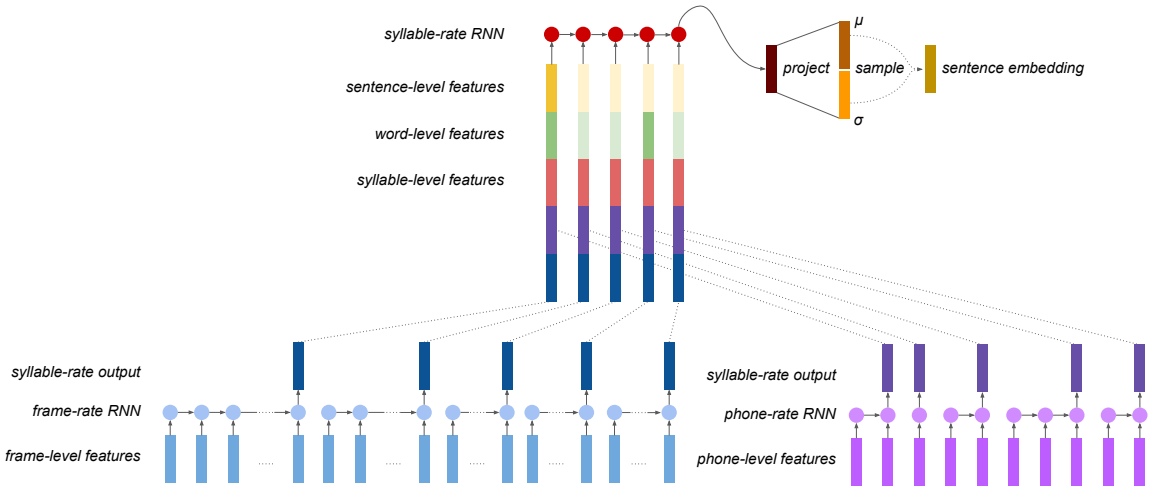

Figure 2: *CHiVE encoder and variational layer. Circles represent blocks of RNN cells, rectangles represent vectors. Broadcasting (vector duplication) is indicated by displaying vectors in a lighter shade of the same colour. (Best viewed in colour).*

(log F_0 and c_0), while phone-level linguistic features and phone durations are fed to a phone-rate RNN. The output of these RNNs are only read at the end of each syllable at which point their states are reset for the next syllable. These outputs are concatenated with the corresponding syllable-, word- and sentence-level linguistic features and then fed to a syllable-rate RNN. The output of the syllable-rate RNN is taken at the end of the sentence and then fed to the variational layer which yields the *sentence embedding*.

Figure 3 shows the decoder model. The sentence embedding is combined with sentence-, word- and syllable-level linguistic features which are fed to a syllable-rate RNN. Each output of that layer is further combined with phone-level linguistic features before being fed into a phone-rate RNN. Phone durations are predicted from this phone-rate RNN; these duration predictions dictate the number of time steps to run the frame-level RNNs. Since energy is more strongly correlated with phones than syllables, a frame-level RNN reads the phone-level output to generate the c_0 sequence for that phone. The pitch is modelled at the syllable level as follows: the output of the phone-rate RNN at the end of each syllable is concatenated with the syllable-rate RNN output and fed to a second frame-level RNN that predicts the log F_0 sequence for the syllable; in this way phone-level linguistic information is accounted at the syllable level.

2.1. Predicting spectrograms

Figure 4 shows the dependency graphs of the final proposed model. Like c_0, spectrogram information is more consistently modelled at the phone-level. Thus, the decoder's c_0 prediction is expanded to additionally generate the spectrogram. There is no explicit loss that constrains the prosody in the spectrogram to be the same as that predicted by the log F_0 generation branch of the network. We rely on the general spectral loss being able to penalize differences in the prosody implicitly present in the spectrogram representation.

The encoder model is unchanged and assuming that intonation is adequately specified by the prosodic features, it should be redundant to also feed the spectrogram to the encoder.

Furthermore, this is useful for performing prosody transfer where we generate a sentence embedding using the encoder for one sentence and then apply the sentence embedding to another sentence. Here we deliberately change the text at the decoder so that the prosody from the encoded sentence gets transferred to the decoded sentence. The transfer is not restricted to the same speaker and can be considered independent of the actual text. For more details see [4]. It is simpler to provide (and manipulate) prosodic features alone than it is to manipulate a full spectrogram; F_0 curves are more easily interpreted by people; and since we are not interested in encoding non intonation-related speaker-specific traits. It is therefore desirable to omit the spectrogram from the encoder.

For arbitrary text, we presently do not have good way to choose a sentence embedding to use at inference time; it depends on the context of the sentence as well as pragmatics. However, the aforementioned unit Gaussian prior distribution imposed on the sentence embeddings enables us to sample randomly within the latent space to obtain different prosodic renditions. Using the mean of the prior, a vector of zeros, typically yields an 'average' prosody contour that is dependent on the data distribution of the underlying training data. This can be thought of as a broad focus rendition of the utterance.

In the baseline CHiVE, the generated prosodic features drive a WaveNet [12] model that is trained to map from the linguistic features plus the predicted c_0 and log F_0 sequences to a speech waveform. In the updated CHiVE model, the generated spectrogram output is fed to a neural vocoder, described in [15].

3. Experiments

The training data consists of speech from 70 speakers of American English recorded in studio conditions totaling 160K sentences. The number of sentences per speaker ranges from a couple of hundred to tens of thousands; gender is roughly balanced. The test set has 100 held-out sentences for each speaker. 24kHz audio is parameterised to spectrograms using 128 Mel-scaled filters using 50ms windows with 12.5ms frame shifts. The zero-th order Mel cepstral coefficient is used as an approximation of energy. F_0 extraction is performed using one of several pitch trackers (including Talkin's RAPT (a.k.a. ESPS), Yin and Yin with high pitch or low pitch specific setting) depending upon how well each algorithm performs on each of the speak-

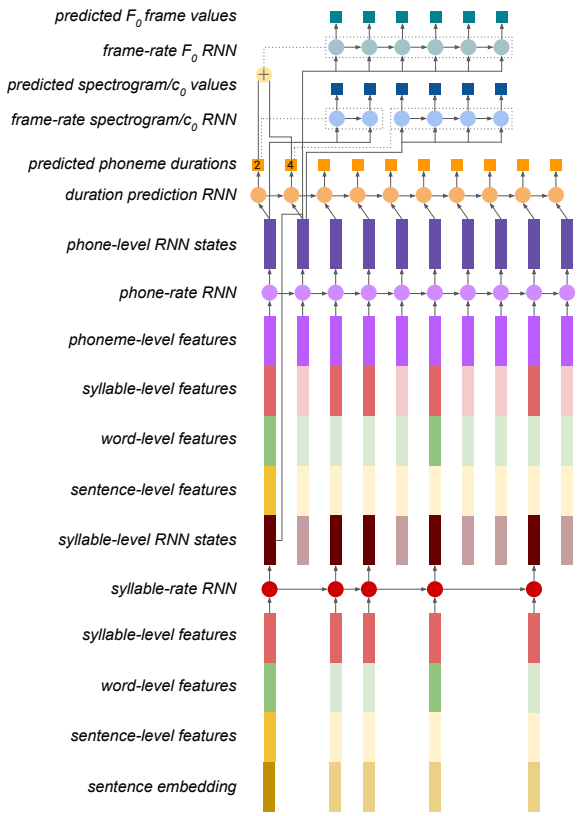

Figure 3: *CHiVE decoder. Circles represent blocks of RNN cells, rectangles represent vectors. Broadcasting (vector duplication) is indicated by displaying vectors in a lighter shade of the same colour. (Best viewed in colour).*

ers. Phone durations are obtained by training a flat start, single Gaussian, monophone hidden Markov model based speech recognition model and running Viterbi forced alignment.

In the encoder network, the frame-rate RNN and phone-rate RNN both consist of two LSTM layers with 64 units each. The syllable-rate RNN consists of two LSTM layers each with 256 units. The variational layer has 256 dimensions. In the decoder, the syllable-rate network consists of two LSTM layers with 256 units. The frame-rate F_0 RNN is two LSTM layers each with 64 units plus a single output linear recurrent layer. The phone-rate RNN has 2 LSTM layers with 32 units. Finally the frame-rate spectrogram/c_0 RNN has 2 LSTM layers with 1024 cells plus a 129 unit linear recurrent layer.

The quality of the synthetic speech is evaluated using mean opinion score (MOS) subjective listening tests. Listeners rate the naturalness of the speech on a scale of 1 to 5. They are native speakers of American English and listen to the samples in quiet conditions using headphones. Each rater provides no more than six judgments.

3.1. Comparison with CHiVE baseline

Table 1 shows MOS results for 1000 unseen sentences generated using a zero-vector sentence embedding for one male and one female speaker. The baseline CHiVE + parallel WaveNet is the one from [4]. We see that predicting the spectrogram directly and using a neural vocoder yields better MOS. It is clearly beneficial to use a neural vocoder.

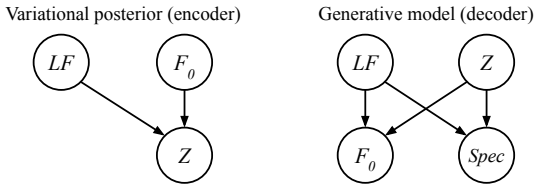

Figure 4: *Dependency graphs for the encoder and decoder. LF denotes the linguistic features, Z the sentence embedding, Spec the spectrogram and F_0 the prosodic features (c_0 and duration have been omitted for brevity).*

Speaker	Baseline CHiVE	Synthesized spectrogram
Female	4.21 ± 0.05	4.27 ± 0.04
Male	4.14 ± 0.05	4.27 ± 0.04

Table 1: *Results of subjective tests comparing the baseline F0+WaveNet system with the proposed spectrogram+neural vocoder system. MOS and 95% confidence intervals.*

3.2. Removing spectrogram from the encoder

To determine the effect of omitting the spectrogram from the encoder input, we train two models where the only difference is whether the encoder is conditioned on the spectrogram.

Table 2 shows the results of MOS tests on a subset of 800 sentences from the held-out test set. Each sentence was rated exactly once. Copy synthesis uses the ground truth spectrograms passed to the neural vocoder to resynthsize the speech. When encoding then decoding the spectrogram with the CHiVE model there is a 0.04 drop in the MOS when compared to copy synthesis. When the spectrogram is removed from the encoder, the MOS drops by a further 0.04, which is significant at the 95% confidence interval.

In the most common use case, the sentence embedding is a vector of zeros: there is no additional information to guide the decoder to predict a specific prosody, resulting in an 'average' intonation. In this scenario, irrespective of whether or not the encoder was trained with the spectrogram, the MOS is 0.1 lower than copy synthesis. This result may be explained by the fact that when no information is provided about the prosody both models produce the 'average' intonation resulting in the same broad focus and, hence, the same MOS.

3.3. Prosodic variety

Objective distortion metrics are not appropriate for measuring differences in prosody encoded within a spectrogram. Distortions could be a result of many things that are unrelated to prosody. A more straight-forward way to show that modify-

Encoder	Spectrogram from:	MOS
Copy synthesis	ground truth (GT)	4.30 ± 0.05
with spectrogram	GT encoded embedding	4.26 ± 0.05
	Zero sentence embedding	4.20 ± 0.05
without spectrogram	GT encoded embedding	4.22 ± 0.05
	Zero sentence embedding	4.20 ± 0.05

Table 2: *Results of subjective tests to determine the effect of not conditioning the encoder on the spectrogram. MOS and 95% confidence intervals.*

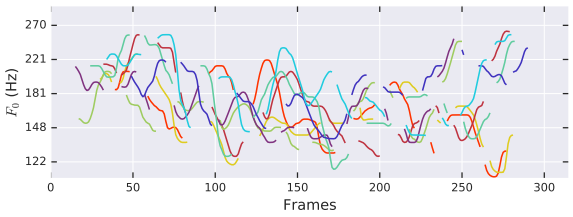

Figure 5: *Eight F_0 curves extracted from generated spectrograms of the same text with different sentence embeddings show that the model is able to generate a variety of different intonations.*

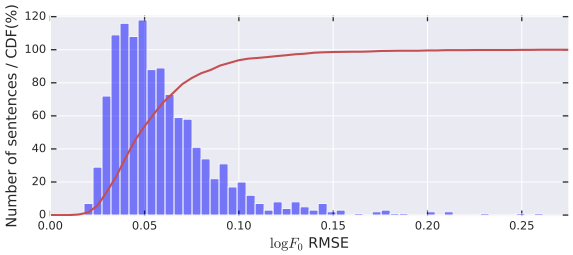

Figure 6: *Histogram showing the distribution of per sentence* $\log F_0$ *RMSE between between the predicted F_0 and the F_0 extracted from the predicted spectrogram (blue) and the corresponding cumulative distribution function (CDF, red).*

ing the sentence embedding does indeed generate different intonation patterns is via audio samples. With the text fixed, we generate multiple versions of the speech using sentence embeddings sampled at random from the prior. The F_0 curves of the spectrogram-synthesized waveforms are extracted using the aforementioned pitch extraction algorithms. A sample of F_0 curves for one sentence is shown in Figure 5. A selection of audio samples is available at `https://google.github.io/chive-prosody/sp2020/`.

3.4. Correspondence between the predicted logF_0 and spectrogram prosody

As mentioned in Section 2.1, there is no explicit loss that constrains the spectrogram to have the same prosody as the predicted $\log F_0$ curve. To check this, we synthesize 140 sentences, 8 times each, all with different sentence embeddings sampled randomly from the prior. The F_0 curves extracted from the spectrogram-synthesized waveforms are compared with the F_0 curves predicted by the network. There is a strong correlation between the $\log F_0$ of the two branches of the model (Pearson correlation coefficient is 0.973). The $\log F_0$ root mean square error (RMSE) computed over the voiced segments of each sentence has an average of 0.067 and a maximum of 0.261. The histogram and cumulative distribution function of the $\log F_0$ RMSE for all of the sentences is shown in Figure 6.

Figure 7 compares the F_0 curves of sentences from different parts of the histogram. Starting with the sentence with the highest RMSE (Figure 7a), we see that most of the error comes from the difference at the end of the sentence. In this particular instance the discrepancy is partially due to pitch tracking errors on the generated audio as there is no audible rising intonation at the end of the sentence; at the same time, the predicted F_0 is unusual because the ending F_0 is too low. In Figure 7b

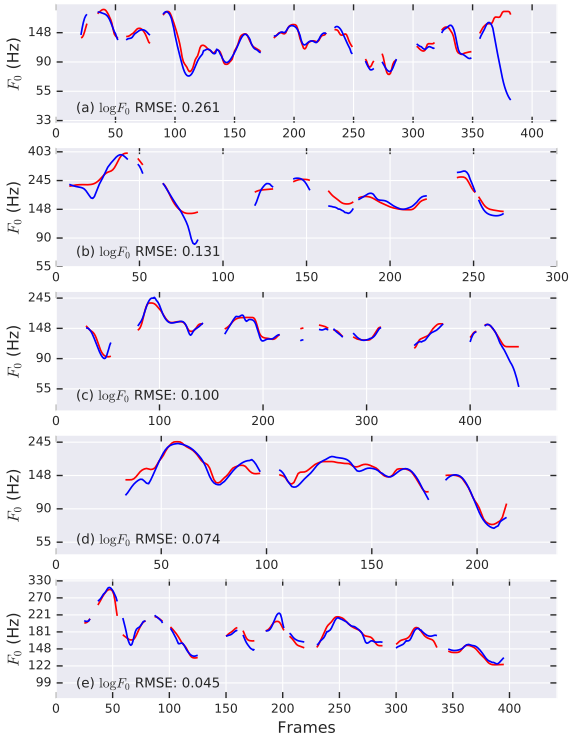

Figure 7: *Comparing the prosody from the two branches of the model when using random sentence embeddings: blue is the directly predicted F_0; red is the F_0 extracted from the synthesized spectrogram.*

we once again see the low F_0 prediction. It is likely that these values are arising because of the randomly chosen sentence embedding: synthesis of the same sentence using a different embedding exhibits different behaviour. Figures 7c to 7e compare the F_0 curves for sentences with smaller error. In these samples, differences in intonation between the two branches of the network are barely noticeable.

4. Conclusion

We have shown that we can use the latent space of the CHiVE model to successfully capture the variation in prosody found in our underlying data while controlling the actual sounds to be spoken with that prosody by the linguistic feature conditioning. Furthermore, this integrated approach plus a neural vocoder, provides quality improvements over supplying the pitch and duration information to a standard WaveNet model.

The work here has been concerned specifically with modelling intonation, rather than other aspects of prosody, such as style and emotion; we have done this primarily by controlling the training data. We would expect the model to capture style variation if it were trained on data displaying more varied prosodic style. However, supervised style labels or more structured latent space representations, would be needed to condition the model, if we were to control that variation.

5. Acknowledgements

We would like to acknowledge contributions from the wider TTS research community at Google and DeepMind and specific contributions from Nicolas Serrano and Tom Kenter.

6. References

[1] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," in *International Conference on Machine Learning*, 2018, pp. 4700–4709.

[2] D. Stanton, Y. Wang, and R. Skerry-Ryan, "Predicting expressive speaking style from text in end-to-end speech synthesis," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 595–602.

[3] J. Lorenzo-Trueba, G. E. Henter, S. Takaki, J. Yamagishi, Y. Morino, and Y. Ochiai, "Investigating different representations for modeling and controlling multiple emotions in dnn-based speech synthesis," *Speech Communication*, vol. 99, pp. 135–143, 2018.

[4] V. Wan, C. Chan, T. Kenter, J. Vit, and R. Clark, "CHiVE: Varying prosody in speech synthesis with a linguistically driven dynamic hierarchical conditional variational network," in *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, 2019, pp. 3331–3340.

[5] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on Mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 4779–4783.

[6] A. v. d. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. v. d. Driessche, E. Lockhart, L. C. Cobo, F. Stimberg *et al.*, "Parallel WaveNet: Fast high-fidelity speech synthesis," *arXiv preprint*, vol. abs/1711.10433, 2017.

[7] K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirschberg, "ToBI: A standard for labeling English prosody," in *Second international conference on spoken language processing*, 1992.

[8] B. Ahn, N. Veilleux, and S. Shattuck-Hufnagel, "Annotating prosody with PoLaR: Conventions for a decompositional annotation system," in *ICPhS*, 2019.

[9] S. Prom-On, Y. Xu, and B. Thipakorn, "Modeling tone and intonation in Mandarin and English as a process of target approximation," *The Journal of the Acoustical Society of America*, vol. 125, no. 1, pp. 405–424, 2009.

[10] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

[11] N. Dilokthanakul, P. A. Mediano, M. Garnelo, M. C. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan, "Deep unsupervised clustering with Gaussian mixture variational autoencoders," *arXiv preprint arXiv:1611.02648*, 2016.

[12] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv preprint*, vol. abs/1609.03499, 2016.

[13] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: A fully end-to-end text-to-speech synthesis model," *arXiv preprint*, vol. abs/1703.10135, 2017.

[14] N. Chomsky and M. Halle, *The Sound Pattern of English.* Harper & Row., 1968.

[15] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. v. d. Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *Proceedings of the 36th International Conference on Machine Learning (ICML 2018)*, 2019.