*INTERSPEECH 2020*
October 25–29, 2020, Shanghai, China

# Learning Syllable-Level Discrete Prosodic Representation for Expressive Speech Generation

*Guangyan Zhang, Ying Qin, Tan Lee*

Department of Electronic Engineering, The Chinese University of Hong Kong
gyzhang@link.cuhk.edu.hk, yingqin@link.cuhk.edu.hk, tanlee@cuhk.edu.hk

## Abstract

This paper presents an extension of the Tacotron 2 end-to-end speech synthesis architecture, which aims to learn syllable-level discrete prosodic representations from speech data. The learned representations can be used for transferring or controlling prosody in expressive speech generation. The proposed design starts with a syllable-level text encoder that encodes input text at syllable level instead of phoneme level. The continuous prosodic representation for each syllable is then extracted. A Vector-Quantised Variational Auto-Encoder (VQ-VAE) is used to discretize the learned continuous prosodic representations. The discrete representations are finally concatenated with text encoder output to achieve prosody transfer or control. Subjective evaluation is carried out on the syllable-level TTS system, and the effectiveness of prosody transfer. The results show that the proposed Syllable-level neural TTS system produce more natural speech than conventional phoneme-level TTS system. It is also shown that prosody transfer could be achieved and the latent prosody codes are explainable with relation to specific prosody variation.

**Index Terms**: text-to-speech, VQ-VAE, prosody control, syllable

## 1. Introduction

Storytelling is a challenging application of the text-to-speech (TTS) technology [1, 2]. It requires to produce a wide range of sound effects and to convey and invoke desired emotions via variation of speech prosody, in order to create an engaging listening experience to the listeners. Prosody is an essential component of human speech that accounts for a variety of paralinguistic functions, which encompass the speaking style, attitude and emotional status[3]. Conventional approaches to TTS, e.g., statistical parametric speech synthesis[4, 5], are able to produce fluent and intelligible speech with flat prosody and neutral affect. In recent years, the end-to-end neural TTS (NTTS) technology has demonstrated major successes in generating high-quality speech with naturalness and expressiveness comparable to human speech [6, 7, 8]. Nevertheless, the ability to control speech prosody to achieve desired expression styles is not found in existing neural TTS systems.

There are generally two strategies to achieve proper prosody control in TTS, namely supervised control and unsupervised control. Supervised control requires manual annotation of prosody, which is a time-consuming task to be done by experts [9]. Unsupervised control relies on a process to automatically learn certain kinds of representation of prosody from a given speech database and apply the representation to achieve prosody control and/or transfer. Unsupervised learning of discrete categorizations of prosodic phenomena or latent prosodic representations from speech has become a prevailing research direction in natural speech generation. Eyben et al. [10] proposed to cluster training data into different expression categories based on prosodic features, and to incorporate the expression cluster information for speech generation by an HMM based system. Watts et al. [11] developed a method of learning low-dimensional continuous-valued prosodic vector with a DNN based TTS system. The prosodic vectors are used to represent utterance-level paralinguistic variation, which could not be derived readily from the text input. Empowered by large-scale speech databases and high-capacity neural network models, there have been numerous studies on deriving prosodic representations from reference speech [12, 13, 14]. Typically a global prosodic representation is distilled from the whole utterance. It does not support the control of prosody on smaller linguistic units within the utterance. Lee et al. [15] attempted a fine-grained representation and control of prosody. The prosodic representation is encoded phoneme by phoneme using a reference attention module with reference speech and text. Klimkov et al. [16] derived a prosodic representation for each phoneme by aggregating prosodic features within the phoneme segment. In recent work[16], the Variational Auto-Encoder(VAE) [17] was used to enhance interpolation ability of the latent prosody space.

Syllables are considered as the primary carrier of important prosodic events like tone and stress [9, 18]. Prosody labeling and modeling are investigated commonly with syllables as the most relevant linguistic units [9, 19, 20, 21]. In the present study, we propose to extract latent prosodic representations from syllable units. Word-level and phrase-level representations can be derived based on the syllable-level one, allowing the coverage of different levels of granularity. On the other hand, previous studies assumed a continuous embedding space for latent prosody modeling [13, 15, 16, 22]. This assumption leads to two limitations. First, it is difficult to establish clear explainable relation of continuous-valued embeddings with perceived prosody, and with acoustic features (e.g., F0) [23]. Second, continuous representation is difficult to be manipulated and applied to achieve specific targets in TTS. Discrete representation of prosody is also found to be more suitable for describing human perception [9, 19].

In this paper, a syllable-level neural speech synthesis system is presented. The system comprises a text encoder that produces syllable-level embeddings [7, 8], in contrast to phone-level embedding in conventional systems. We propose to extract discrete prosodic representations for individual syllables using a Vector-Quantised Variational Auto-Encoder (VQ-VAE) model [24]. The relation between the discrete representations and prosodic features is analyzed to make the unsupervisedly learned information more understandable and usable.



http://dx.doi.org/10.21437/Interspeech.2020-2228

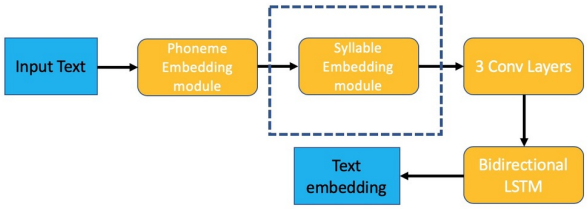


Figure 1: *syllable-level text encoder module*

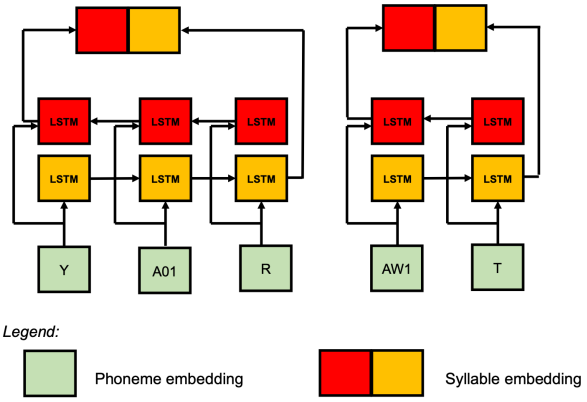


Figure 2: *The English example text "you're out" consists of two syllables,* ***Y-AO1-R*** *and* ***AW1-T****, and each syllable is composed of a sequence of phonemes. The phoneme embeddings (green squares on the bottom) of each syllable is processed by the Bi-LSTM yielding the syllable embeddings (red-yellow square on the top)*

## 2. Syllable-Level Text Encoder

The proposed model follows the standard Tacotron 2 network structure, which consists of a text encoder and an attention-based autoregressive decoder. The text encoder serves to convert the input text into a sequence of text embeddings. To facilitate extraction of syllable-level representations of prosody, we propose a modified design of text encoder as illustrated in Figure 1. The input text is first converted into a sequence of phoneme embeddings by the phoneme embedding module. The phoneme embeddings are grouped on syllable basis to be processed by the proposed syllable embedding module, which is illustrated as in Figure 2. Subsequently the sequence of syllable embeddings go through a stack of 3 convolution layers and a single bidirectional recurrent neural network with LSTM cells (Bi-LSTM) layer to generate the text embedding output as in the standard Tacotron 2.

There may be thousands of distinct syllables in a given language. If each syllable is assigned an embedding vector, the TTS system would have a huge model size. It is also difficult to derive a reliable and meaningful vector representation for an infrequent syllable [25]. In this study, phoneme and syllable embeddings are learned jointly.

Consider a text input that is transcribed into $N$ syllables in the spoken form, i.e., $\{s_1, ..., s_n, ..., s_N\}$. The $n^{th}$ syllable $s_n$ is further decomposed into constituent phonemes as $s_n = \{p_{n,1}, ..., p_{n,i}..., p_{n,|s_n|}\}$. $|s_n|$ denotes the number of phonemes in $s_n$. The phoneme $p_{n,i}$ is represented by a one-hot vector $1(p_{n,i})$. The phoneme embedding module is trained to transform the one-hot vector into a continuous-valued embedding vector as

$$p'_{n,i} = W_p \times 1(p_{n,i}), \tag{1}$$

where $W_p \in \mathbb{R}^{d_p \times |V|}$ is a matrix of learnable parameters, and $|V|$ denotes the number of phonemes (typically in the range of 20 to 50).

As shown in Figure 2, the syllable embedding module takes the phoneme embeddings as input (the green blocks) and uses a Bi-LSTM layer to generate the syllable embedding. The Bi-LSTM has a forward component and a backward component, which are denoted as forward-LSTM and backward-LSTM respectively. Let $\overrightarrow{h}$ and $\overleftarrow{h}$ be the internal states of forward-LSTM and backward-LSTM returned after the entire input sequence has been processed. The syllable embedding $s'_n$ is given by the concatenation of $\overrightarrow{h}$ and $\overleftarrow{h}$, i.e.,

$$s'_n = [\overrightarrow{h}, \overleftarrow{h}], \tag{2}$$

$$\overrightarrow{h} = \text{forward-LSTM}(p'_{n,1}, ..., p'_{n,i}..., p'_{n,|s_n|}), \tag{3}$$

$$\overleftarrow{h} = \text{backward-LSTM}(p'_{n,1}, ..., p'_{n,i}..., p'_{n,|s_n|}), \tag{4}$$

## 3. Generation of Syllable-Level Discrete Prosodic Representation

### 3.1. Extraction of continuous prosodic representation

In [15], it was proposed to extract prosodic representation at phoneme level through reference attention. In our preliminary investigation with a single-speaker expressive speech dataset, it was found that the method in [15] suffers from instabilities of reference attention or may fail to learn the reference attention alignment during training. This problem was also noted in [16]. In the present study, the acoustic features related to speech prosody [3], i.e., F0, intensity, and duration [16] are the input for extracting continuous prosodic representations at syllable level. The design of this module is shown as in Figure 3. For each syllable, the sequence of frame-level F0 and intensity features are processed by the syllable F0/INT embedding module to generate a syllable-level representation named as Representation 1. Representation 1 appended with the syllable duration go through to a feed-forward network to generate Representation 2, which is denoted by $sp'_1, ..., sp'_n, ..., sp'_N$. Representation 2 is taken as the output of the entire module of continuous representation extraction.

The syllable F0/INT embedding module is detailed as in Figure 4. It has a similar structure to the syllable embedding module in Figure 2. By applying this structure, the information about variation of prosodic features within a syllable [9, 26] can be captured.

### 3.2. Discretization of prosodic representation

The VQ-VAE [24, 27] model is a generative model to learn discrete latent representation without supervision. The encoder network of VQ-VAE outputs the discrete latent code, in contrast to the continuous representation as in VAE; and the prior of latent code is learnt in VQ-VAE rather than being a fixed Gaussian distribution in VAE. In our study, the encoder network of VQ-VAE is utilized to discretize the learned prosodic representations. Let $\mathbf{e} \in \mathbb{R}^{K \times D}$ be the latent embedding space (also known as codebook) in the VQ-VAE, where $K$ is the number of latent embedding categories and $D$ is the embedding dimension. The $k^{th}$ element in the codebook is noted by $e_k$. The

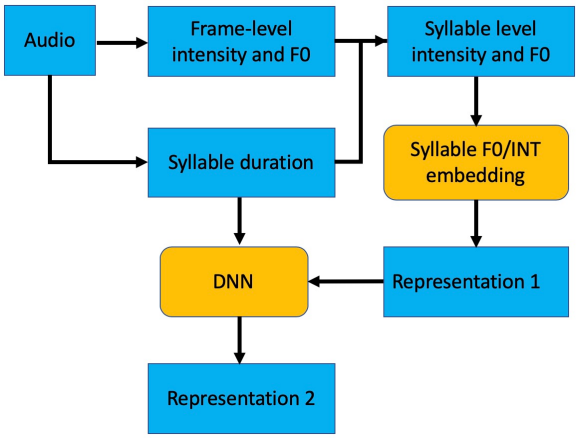


Figure 3: *continuous prosodic representation extraction module*

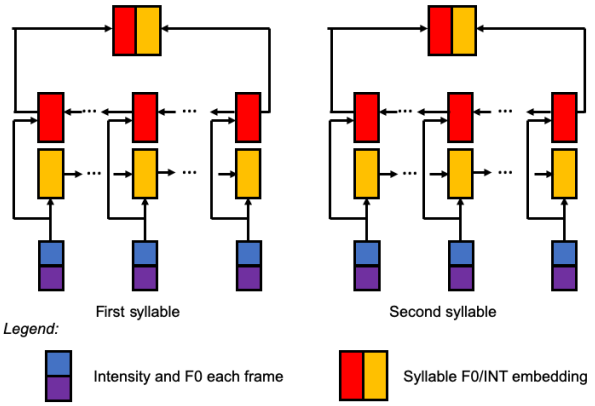


Figure 4: *syllable F0/INT embedding module*

prosodic representation $sp'_n$ generated as described in the previous section is matched to the closest elements in the codebook. In this way, the continuous prosodic representation $sp'_n$ is quantized as $sp''_n$,

$$sp''_n = \text{Quantize}(sp'_n) = e_k \tag{5}$$

where $k = \arg\min||sp'_n - e_k||_2$.

Since $\arg\min()$ is non-differentiable, the gradients $\bigtriangledown_{sp''_n}\text{Loss}$ from output $sp''_n$ is copied to the input $sp'_n$ during the stochastic backpropagation. In order to encourage $sp'_n$ to stay close to the embedding space and prevent it from fluctuating too frequently from one code vector to another, a commit loss is added to final loss:

$$L_{\text{commit}} = \sum_{i}^{N} ||\text{sg}[sp'_n] - sp''_n||_2^2, \tag{6}$$

where $\text{sg}[.]$ is the stop gradient operator. The Exponential Moving Average (EMA) strategy [28] is used to update the embedding vectors for accelerating convergence of model.

## 4. Prosody Transfer and Prosody Control

Prosody transfer is defined as transplanting prosody from reference speech to generated speech, while prosody control defined as generating speech with a desired prosody.

Prosody transfer can be achieved by passing the extracted acoustic features from reference speech through the modules for prosodic representation extraction, and obtaining a sequence of discrete prosody codes at syllable level. The learned codebook is queried with the codes to retrieve syllable-level prosodic representations. The prosodic representations are then concatenated with the output of syllable-level text encoder to generating the mel-spectrogram of output speech.

Since there are limited number of prosody codes, to investigate which code will result in specific prosody variation is possible. For the purpose of prosody control at syllable level, the target syllables will be endowed with specific prosody codes to generate speech with desired prosody.

## 5. Performance Evaluation

### 5.1. Experimental set-up

A series of experiments on speech generation with prosody transfer and prosody control are carried out with the Blizzard 2013 English dataset of audiobook recordings. The database contains speech from a female professional narrator reading the text of a collection of classic novels. The training data consist of approximately 19 hours of recordings covering 2 non-fiction audiobooks, in an expressive reading style.

Praat [29] was used to extract F0 and intensity every 10 ms. Phoneme-level time alignment and separation of a word into syllables were performed using the Festival software tools [30]. The duration of a syllable was obtained by adding up the duration of all phonemes constituting the syllable. A blank symbol was inserted to mark word boundary.

The following three systems were trained and evaluated:

- Baseline neural TTS: The baseline model is a neural TTS system following the general architecture of Tacotron 2, in which the text encoder generates phoneme-level embeddings. The decoder predicts mel-spectrograms from text embeddings frame by frame in an autoregressive manner [6]. The WaveGlow vocoder [31] is used to generate a speech waveform from the predicted mel-spectrograms;
- Syllable-level neural TTS: The model design is the same as the baseline model, except that the text encoder is modified to produce syllable-level embeddings as described in section 2;
- Prosody Transfer TTS: The Syllable-level neural TTS is modified to transfer prosody from reference speech to generated speech as described in section 4.

Subjective evaluation of these TTS systems was carried out by a discriminative listening test implemented via the Amazon Mechanical Turk platform [32]. 20 native English speakers participated in the perceptual evaluation. One of the listening tasks was designed to evaluate the performance of the Syllable-level neural TTS as compared with the phoneme-level neural TTS (the baseline). The other task aimed to evaluate the effectiveness of prosody transfer. Pairs of synthesized sentences were presented to the participants. Each pair of sentences, e.g., A and B, was counter-balanced and was played with a random order, i.e., could be AB or BA.

### 5.2. Evaluation of Syllable-level neural TTS

Each participating listener was presented 10 pairs of sentences generated by the Baseline neural TTS and the Syllable-level neural TTS systems. The listeners were asked to indicate their

Table 1: *Mean and STD values of naturalness preference scores obtained with the Baseline neural TTS and Syllable-level neural TTS systems.*

| | Mean of preference score | STD |
|---|---|---|
| Baseline NTTS | 3.1 | 1.33 |
| Syllable-level NTTS | 5.65 | 1.38 |
| equal naturalness | 1.3 | 0.97 |

Table 2: *Mean and STD values of preference scores obtained from the Syllable-level neural TTS and the Prosody Transfer TTS system on the effectiveness of prosody transfer.*

| | Mean of preference score | STD |
|---|---|---|
| Syllable-level NTTS | 2.15 | 0.93 |
| Prosody Transfer TTS | 7.85 | 0.93 |

preference on "Audio sample A is more natural", "Audio sample B is more natural" or "The two samples are equally natural".

Table 1 summarizes the mean and standard deviation (STD) values of the preference scores (in terms of the number of times that each option is selected). Since there are 10 test pairs, the summation of preference scores from each listener is equal to 10. Compared with the baseline system, the Syllable-level neural TTS system shows significantly higher preference score. With respect to each of the two TTS systems, there were 20 pairs preference scores given by 20 listeners. The paired t-test was performed on these listener-dependent preference scores. The statistical analysis confirms that the advantage of the Syllable-level neural TTS system over the baseline system is statistically significant (p = .002).

### 5.3. Evaluation on Prosody Transfer TTS

Performance evaluation of prosody transfer was carried out by a listening task of pairwise comparison. The listener was presented with 3 speech utterances. One was the reference utterance (natural speech) while the other two were generated by the Syllable-level neural TTS and the Prosody Transfer TTS systems. The listener was asked to choose which of the two synthesized utterances is closer to the reference utterance in terms of intonation and rhythm. The mean and STD values of the preference scores are shown in Table 2. The results show that the Prosody Transfer TTS system is preferred by more listeners, and the difference is statistically significant (p = .007). This indicates that the proposed approach to prosodic representation learning is effective for prosody transfer from reference utterance to generated speech.

### 5.4. Analysis of prosody control

The target syllable's prosody code was manipulated to examine the effectiveness of prosody control. It was found that some codes were related to particular prosody or acoustic variations. For example, code 1 (i.e., the first element of codebook) was observed to be correlated with word boundary. It could be treated to be related to prosodic phrasing phenomenon. Code 3 could contribute to lengthening duration and increasing pitch of a target syllable, while code 6 played an opposite role. We recommend that readers listen to the examples on our demo page[1].

[1]https://patrick-g-zhang.github.io

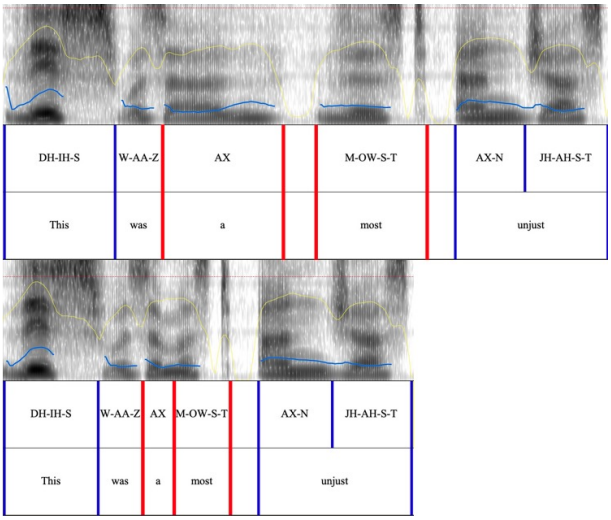


Figure 5: *An example of prosody control at syllable level. Blue curve indicates pitch contour.*

Figure 5 shows an example of fine-grained prosody control at the syllable level. Given the text and audio of a reference utterance (" This was a most unjust "), its prosody codes sequence was extracted. For the generation of the first utterance in Figure 5, the prosody code of the third syllable, AX, and fourth syllable, M-OW-S-T, were both changed to code 3 and other codes remains unchanged. The second synthesized utterance in Figure 5 was obtained by using the unchanged prosody codes. It was shown that the durations of the third and fourth syllables became longer due to the change of code 3, while those of other syllables stayed the same between the first and second utterances. In terms of pitch, code 3 were found to increase the pitch values. This suggests that code 3 is able to consider the prominence phenomenon of the syllable.

## 6. Conclusions

A novel Syllable-level neural TTS system was proposed and investigated. It has been shown to produce more natural speech than the conventional system with phoneme-level text embedding. The discrete prosody embedding representations extracted at syllable level are able to transfer prosody from reference speech to generate synthesized speech. The feasibility of fine-grain prosody control at syllable level has been demonstrated. The experimental results also suggest that some prosody codes contribute to specific prosody variations for syllables. In our future work, all of the prosody codes and their related prosody variation will be investigated. We will further train a language model to help the generation of prosody code sequence, which is expected to provide additional benefits to the synthesis of storytelling speech.

## 7. Acknowledgement

This research is partially supported by a Tier 3 funding from the Innovation and Technology Support Programme (Ref: ITS/309/18) of the Hong Kong SAR Government, a Knowledge Transfer Project Fund (Ref: KPF20QEP26) and a direct research grant from the Chinese University of Hong Kong.

# 8. References


[1] M. Theune, K. Meijs, D. Heylen, and R. Ordelman, "Generating expressive speech for storytelling applications," *IEEE/ACM Trans. Audio, Speech, Language Process*, vol. 14, no. 4, pp. 1137–1144, 2006.

[2] R. Montaño, F. Alías, and J. Ferrer, "Prosodic analysis of storytelling discourse modes and narrative situations oriented to text-to-speech synthesis," in *Proc. Eighth ISCA Workshop on Speech Synthesis*, 2013, pp. 171–176.

[3] B. Schuller and A. Batliner, *Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing*. John Wiley & Sons, 2013.

[4] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[5] H. Ze, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, 2013, pp. 7962–7966.

[6] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," in *Proc. ICLR*, 2017.

[7] 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: Towards end-to-end speech synthesis," in *Proc. Interspeech*, Aug. 2017, pp. 4006–4010.

[8] 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 *Proc. ICASSP*, 2018, pp. 4779–4783.

[9] P. A. Taylor, *Text-to-speech synthesis*. Cambridge University Press, 2009.

[10] F. Eyben, S. Buchholz, N. Braunschweiler, J. Latorre, and K. Knill, "Unsupervised clustering of emotion and voice styles for expressive tts," in *Proc. ICASSP*, 2012, pp. 4009–4012.

[11] O. Watts, Z. Wu, and S. King, "Sentence-level control vectors for deep neural network speech synthesis," in *Proc. Interspeech*, 2015, pp. 2217–2221.

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

[13] Y. Wang, D. Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *Proc. ICML*, 2018, pp. 5167–5176.

[14] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen *et al.*, "Hierarchical generative modeling for controllable speech synthesis," in *Proc. ICLR*, 2019.

[15] Y. Lee and T. Kim, "Robust and fine-grained prosody control of end-to-end speech synthesis," in *Proc. ICASSP*, 2019, pp. 5911–5915.

[16] V. Klimkov, S. Ronanki, J. Rohnke, and T. Drugman, "Fine-grained robust prosody transfer for single-speaker neural text-to-speech," in *Proc. Interspeech*, 2019, pp. 4440–4444.

[17] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. ICLR*, 2014.

[18] S. Nooteboom, "The prosody of speech: melody and rhythm," *The handbook of phonetic sciences*, vol. 5, pp. 640–673, 1997.

[19] J. B. Pierrehumbert, "The phonology and phonetics of english intonation," Ph.D. dissertation, Massachusetts Institute of Technology, 1980.

[20] K. E. Dusterhoff and A. W. Black, "Generating f0 contours for speech synthesis using the tilt intonation theory." in *Intonation: Theory, Models, and Applications*, 1997, pp. 107–110.

[21] K. Hirose, H. Fujisaki, and M. Yamaguchi, "Synthesis by rule of voice fundamental frequency contours of spoken japanese from linguistic information," in *Proc. ICASSP*, 1984, pp. 597–600.

[22] Y.-J. Zhang, S. Pan, L. He, and Z.-H. Ling, "Learning latent representations for style control and transfer in end-to-end speech synthesis," in *Proc. ICASSP*, 2019, pp. 6945–6949.

[23] N. Tits, F. Wang, K. E. Haddad, V. Pagel, and T. Dutoit, "Visualization and interpretation of latent spaces for controlling expressive speech synthesis through audio analysis," in *Proc. Interspeech*, 2019, pp. 4475–4479.

[24] A. van den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," in *Proc. NIPS*, 2017, pp. 6306–6315.

[25] G. Marra, A. Zugarini, S. Melacci, and M. Maggini, "An unsupervised character-aware neural approach to word and context representation learning," in *Proc. ICANN*, 2018, pp. 126–136.

[26] N. Obin, J. Beliao, C. Veaux, and A. Lacheret, "Slam: Automatic stylization and labelling of speech melody," in *Speech Prosody*, 2014, p. 246.

[27] X. Wang, S. Takaki, J. Yamagishi, S. King, and K. Tokuda, "A vector quantized variational autoencoder (vq-vae) autoregressive neural $f\_0$ model for statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 157–170, 2019.

[28] L. Weng, "From autoencoder to beta-vae," *lilianweng.github.io/lil-log*, 2018. [Online]. Available: http://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

[29] P. Boersma, "Praat, a system for doing phonetics by computer," *Glot. Int.*, vol. 5, no. 9, pp. 341–345, 2001.

[30] P. Taylor, A. W. Black, and R. Caley, "The architecture of the festival speech synthesis system," in *Proc. The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*, 1998.

[31] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *Proc. ICASSP*, 2019, pp. 3617–3621.

[32] G. Paolacci, J. Chandler, and P. G. Ipeirotis, "Running experiments on amazon mechanical turk," *Judgment and Decision Making*, vol. 5, no. 5, pp. 411–419, 2010.