Speech Prosody 2024
2-5 July 2024, Leiden, The Netherlands

Is Pitch Contour Sufficient to Encode Prosody in Neural Text-to-Speech?

Alex Peiró-Lilja[1,2], *Mireia Farrús*[1,2]

[1]Centre de Llenguatge i Computació (CLiC), Universitat de Barcelona
[2]Institut de Recerca en Sistemes Complexos (UBICS), Universitat de Barcelona

alex.peiro.lilja@ub.edu, mfarrus@ub.edu

Abstract

Nowadays speech synthesis has reached levels of voice quality and naturalness close to human. This has been achieved thanks to the rapid evolution of generative architectures deployed for neural text-to-speech (TTS). Many approaches have been proposed to encode speech style —i.e. prosody attributes— leveraging these models in order to transfer it to the generated speech. The most common acoustic features for this purpose are the spectrograms. However, is the whole frequency representation really necessary to learn speech attributes? To answer this question, in this work we propose the sparse pitch matrix (SPM), an sparse and binary representation of the pitch sub band. We assumed that pitch is sufficient to make the model extrapolate the rest of the prosody aspects. To study its impact, we performed an experiment built upon the unsupervised global style tokens conditioning the Tacotron2 decoding. The tokens were fed with the encoded SPMs during training, similarly to the original approach. From the posterior analysis we found that: 1) there are significant differences in many prosody attributes between tokens, and 2) all tokens, in isolation, provide acceptable levels of quality, intelligibility and naturalness, according to human evaluators.

Index Terms: prosody attributes, speaker style, text-to-speech

1. Introduction

Speech synthesis has evolved dramatically thanks to the latest neural architectures and generative approaches. Not only by achieving quality and naturalness levels close to the human voice, but also by learning speaking styles and transferring them to the generated voices. Nowadays, artificial speech is produced by Text-to-Speech (TTS) systems based on deep neural networks, which are typically made up by an encoder-decoder duo [1, 2, 3] that predict acoustic feature representations (i.e. spectrograms) plus a neural vocoder [4, 5, 6], which generates audio waveform from the predicted features. Although these two modules were used to be trained separately, recent works have released powerful and lightweight fully end-to-end (E2E) models capable of returning audio waveform directly from text [7, 8, 9, 10]. On the other hand, there has been a big push to improve expressiveness in generative speech to adapt the *style* of the voice according to the content of the speech. Speech style is closely related to prosodic characteristics, such as the intonation, rhythm and stress [11]. Hence, being able to adapt the style in synthesis is essential for many applications such as automatic dialogue systems or storytelling. Historically, prosody has been categorized by either manual and automatic annotation methodologies [12, 13, 14, 15]. However, due to the increasingly amount of available data and the arrival of the latest deep learning training strategies, the need to explicitly label prosody is becoming less necessary. are being put today into model and align prosody along with the text —or its linguistic features— to predict style and prosodic attributes accordingly [16, 17, 18, 19, 20]. Mel spectrograms are the most widely used acoustic features to encode prosodic information. But in this representation more information is implicitly included (i.e. speaker's information) and the model will need to disentangle. Alternative speech representations have not been yet fully explored for this purpose, although some works explored features obtained from self-supervised learning (SSL) models such as HuBERT [21] in exchange of creating large systems.

In this work we explore modeling speech style with global style tokens (GST) by using only a specific representation of mel spectrograms: the pitch subband through time. Consequently, we simplified the reference encoder architecture, thus making the model lighter. The premise was taken from the fact that basic prosody attributes, like pitch accent and energy, are usually correlated [22]. Besides, harmonics are totally dependant of the pitch itself. And also formants, which describe vocal pronunciation of the speaker, could be expected to be learned by the decoder as usual.

The paper is structured in the following way: in Section 2, we present a literature review of recent works on modeling prosody attributes and speaker style. In Section 3, a detailed description of our experiment setup is provided, including the TTS system implemented, the unsupervised approach to learn speech style and our binary sparse matrix as the prosody reference for the model. Section 4 describes the training and inference procedures, and evaluation approach and results from our quantitative and subjective tests are explained in section 5. Finally, in section 6 conclusions and future work are commented.

2. Prosody Encoding

Early work on exploring prosody encoding in TTS neural architectures was presented in [16, 17]. In both, authors used a reference encoder to project mel-scale spectrogram representations into hidden vectors creating thus a latent space with relevant style information. In the former approach, each output frame of the seq2seq Tacotron [1] decoder was predicted conditioned on the latent vector prediction. And the latter approach —the chosen for this work and detailed in section 3.2—, based on a parallel encoder followed by a multi-head attention block connected to a bank of embeddings were trained together with Tacotron2. Besides, other recent works that implemented state-of-the-art deep learning architectures for style transfer in TTS also used the entire representation of melspectrograms, either by outperforming GSTs using variational autoencoders (VAE) to disentangle hidden prosody attributes [23] or later on hierarchical and multi-scale VAEs [18], proposed to cover speech diversity. Quantized latent vectors were also explored [24, 19]. Prosospeech uses the low-frequency band of the mel-spectrogram to be encoded in a latent prosody vector. StyleTTS [25] is a very recent model capable to successfully

10.21437/SpeechProsody.2024-224

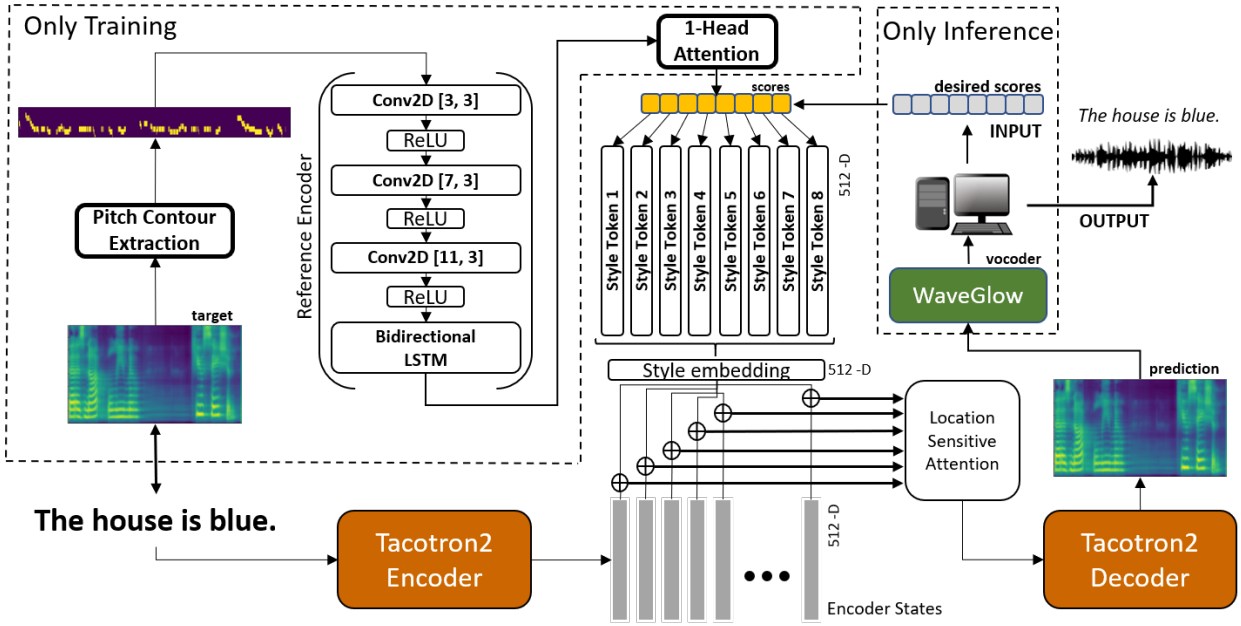

Figure 1: *Overall scheme of the implemented model.*

adapt style and emotion of a reference speaker. It also leverages mel-spectrogram information to train a prosody encoder, although pitch contour is also extracted to improve the prosody prediction. In [19] latent variable prosody is proposed assuming that pitch, duration and energy are not independent attributes, which at some point assumes the same fact that in our work. A newer prosody prediction based on diffusion is presented in [20]. On the other hand, alternative TTS systems have been developed in order to control prosody attributes in inference. Concretely, pitch is extracted from training samples, along with duration and energy to fed specific modules for a better estimation [26, 27].

3. Experiment Setup

3.1. Description

Our main objective was to find evidence that speech style can be modeled using a smaller and, above all, more concise representation of prosody. To this purpose, we designed an input reference matrix containing only the pitch information (explained in 3.3). To test it, we performed an experiment built upon original global style tokens (GST) [17] approach. Assuming less complexity in the reference input, the encoder and attention modules were re-designed to make them lighter (see details in 3.4 and 3.5).

3.2. End-to-End TTS Architecture

Figure 1 shows the overall architecture of the experiment. Our TTS was a combination of Tacotron2 [2] for the acoustic features prediction followed by WaveGlow [28] to generate the waveform. Following the structure presented in [17], we add to the encoder's output a resulting style embedding from a bank of GSTs that were connected to a multi-head attention layer fed by the output of a reference encoder. The scores predicted by the attention module are used for the weighted sum of GSTs that returns the style embedding. It was expected that GSTs capture prosodic attributes of the speaker style during the training process. In the original publication, authors used 10 style tokens. But after testing, we set 8 because we worked with a less-expressive dataset. During training, attention weights (named as *scores*) are learned together with reference encoder and GST with no supervision. And in inference, these values could be modified arbitrarily.

3.3. Sparse Pitch Matrix

For each training sample we created a binary matrix that we called sparse pitch matrix (SPM). It is equivalent to a low-band frequency mask. Rows correspond to the number of 80-channel log-mel spectrogram bins that cover the frequency range where human pitch oscillates. We set this band from 30 to 300 Hz approximately, so the entire range from the highest feminine and the lowest masculine tones was completely covered [29]. Columns are the time frames. Pitch contours –frequency position– of training samples were extracted with Praat [30]. Then, we located them in their corresponding bin and frame of the SPM. Every position in the SPM is set to 0 except pitch locations, which are valued as 1. The resulting matrix is a log mel-scale SPM of 11 bins by the total number of time frames (left side in Figure 1).

3.4. Reference Encoder

Despite their reduced size and sparsity, we treated SPMs as images, like with mel spectrograms. Our reference encoder (left side in Figure 1) follows essentially the same structure as the original presented in [17]. However, we simplified its structure, as SPMs are much less complex representations than spectrograms, and also because we were looking to decrease model size during training. We built a stack of only 3 2D convolutional layers with kernel sizes 3x3, 7x3, 11x3, and with 8, 16 and 16 channels, respectively. Batch normalization, ReLU activation and then dropout at 50% were applied to the output of each convolutional layer. We replaced the last GRU layer by

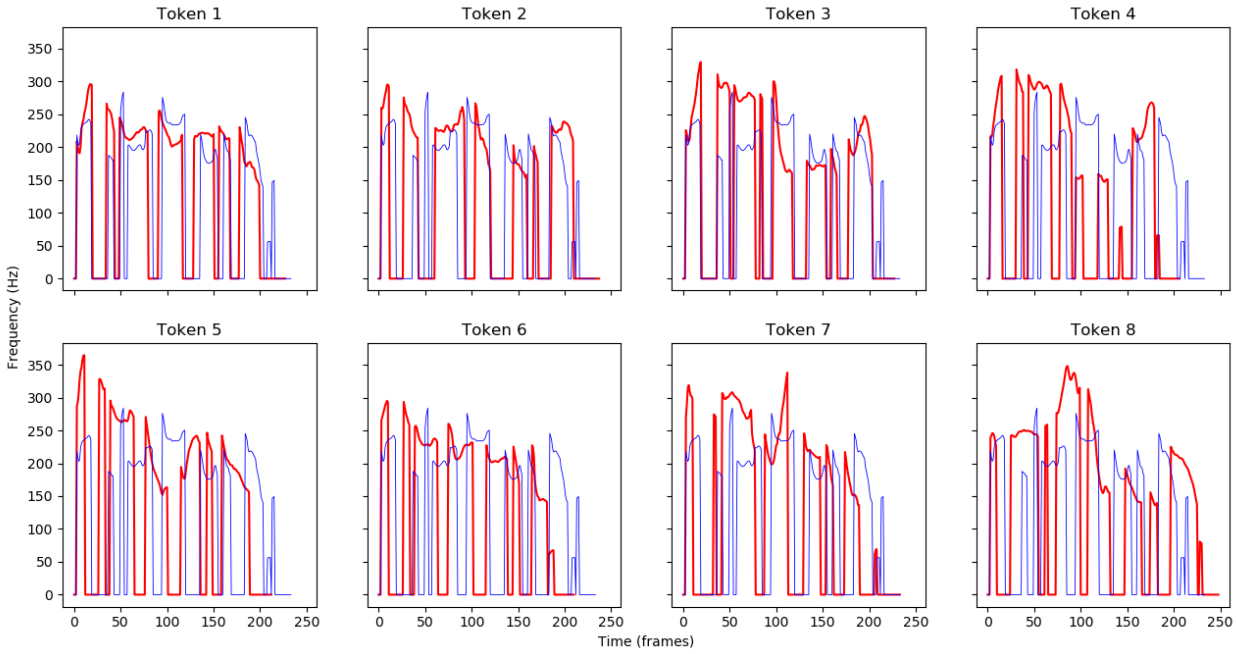

Figure 2: *Pitch curve comparisons of each emphasized style token. A minimum score is set to the rest of tokens to ensure stable synthesis: original (thin and blue) and synthetic (thick and red).*

a bidirectional LSTM, similar to the Tacotron2's text encoder. Finally, the module returned an embedding of 512-D.

3.5. Single-Head Attention

According to [17], different attention-based approaches were explored after choosing multi-head attention from transformers [31] so we decided to keep the same module. Its objective is to predict attention weights –*scores*– that graduate the relevance of each token depending on the reference. Multi-head attention is able to learn different hidden features in parallel. In the case of original GSTs, depending on the number of heads (h), the content of style tokens was a concatenation of h sub-parts. We found multi-head to be problematic to analyze the impact of GSTs, because the number of style weights per token is h. So arbitrary exploration of scores would be too expensive. For this reason, and also because SPM representation is simpler, we decided to implement a 1-head attention module. Then, only 8 style scores, one per style token, were trained and set to be manipulated during evaluation.

4. Implementation

4.1. Training

We used the public domain LJ Speech Dataset from LibriVox project, which contains 13,000 short audio clips of a single female speaker reading passages from 7 non-fiction books, including their transcriptions. [1] Clips vary in length from 1 to 10 seconds and have a total length of approximately 24 hours. We used 12,500 for training and 100 for validation, applying zero-padding to fix sequence lengths. Same dropout and batch normalization as in the original Tacotron 2 model were set. The training was performed with an NVIDIA GPU Titan Xp, achieving an average iteration time of about 4 seconds with a batch size of 32. We stopped training after 160 epochs, when the alignment matrix was as close as identity. Validation was performed every 1,000 training steps.

4.2. Inference

In the inference, reference encoder and 1-head attention module were removed, so the scores previously predicted by the letter could be set manually. This allowed us to decide the relevance of each token of the GSTs bank to produce the final style embedding. After empirical testing, we established two thumb-up rules when choosing score values: 1) the total sum of the eight scores should be around 1.0, and 2) all scores had to be non-zero values. We observed that following both rules a stable synthesis was ensured. Samples can be listened to in our repository [2]. Also, our HuggingFace demo of Tacotron2 trained with GST and single-head attention is released [3], in which users can tune attention weights to generate varied prosody. Note that this demo version has been trained with the whole mel spectrogram representation and only with 3 style tokens.

5. Evaluation

To observe what type of style information was stored in each style token, we configured the score values emphasizing a specific one, keeping the commented thumb up rules. For instance, to evaluate token 5, a possible score configuration to the GST bank would be: [0.05, 0.05, 0.05, 0.05, **0.6**, 0.05, 0.05, 0.05]. Figure 2 depicts a comparison between ground-truth and generated pitch contours when emphasizing specific token scores. Moreover, in order to prove that each token did store relevant prosody information, generated speech was anal-

[1] https://keithito.com/LJ-Speech-Dataset/

[2] https://github.com/AlexK-PL/Tacotron2_GST_SPM

[3] https://huggingface.co/spaces/CLiC-UB/tacotron2-gst-en

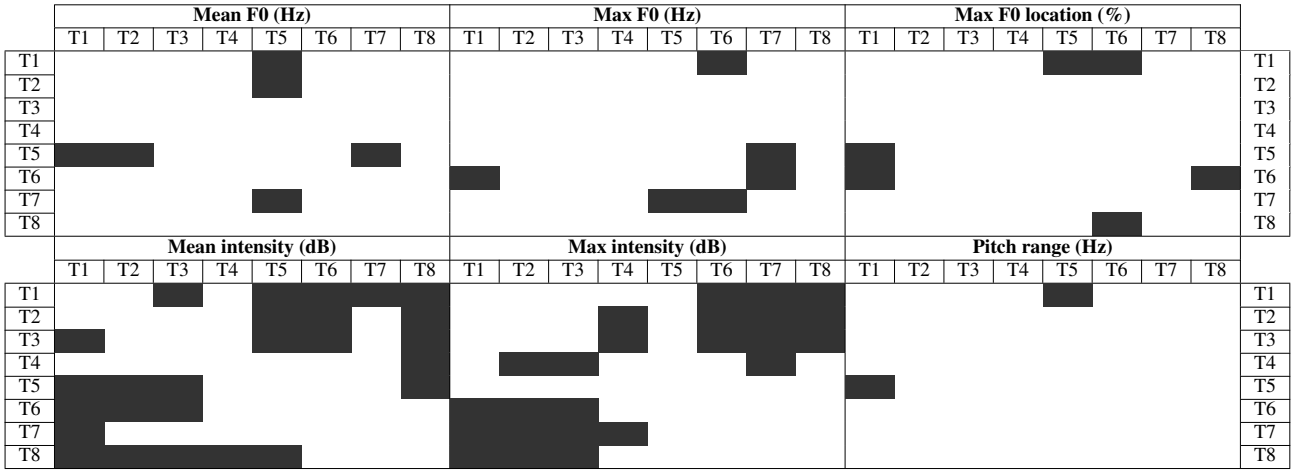

Table 1: *Significant differences between pairs of tokens for different prosody attributes.*

	Averaged MOS scores									
	T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T Avg (Std)**	**Human**
Intelligibility	3.87	3.70	4.10	4.27	**4.50**	3.90	4.43	3.70	4.06 ± 0.32	4.60
Quality	3.24	3.14	3.02	3.47	**3.80**	3.24	3.24	3.53	3.34 ± 0.25	4,67
Naturalness	3.10	3.00	3.44	3.54	**3.87**	3.17	3.34	2.57	3.25 ± 0.39	3.30

Table 2: *MOS scores.*

ysed. We synthesized 9 sentences of 7-15 syllables length from test data 8 times, one for each emphasized token, applying also our mentioned thumb up rule to the score values. From generated clips, we extracted prosody features, such as mean, range and maximum values of F0 and intensity, and also the speech rate. Then, we performed a t-test on every pair of tokens in order to validate whether each of them had statistical significance over each prosodic feature. Tests were performed assuming non-equal variability and rejected null hypothesis when p-values were below 0.05 (black spots in Table 1). Interestingly, more significance between tokens is presented on intensity features, although these were not included in the SPM representation. Pitch-related features presented less significance between tokens, and pitch range only between token 1 and 5. The last observation would make sense since we trained the model with a single speaker. Some features, like the speech rate did not show significant difference although we were able to perceive differences with respect to the target. This may happened due to the nature of the dataset which is a non-fiction read-speech, so the speaker tends to keep similar rhythm and pitch.

Also a MOS test over 15 subjects was carried out. Users had to listen and score a total of 18 audio samples, specifically, 9 versions of two test samples: human version and a synthesis version of each emphasized token. Subjects were asked to score in a 1-5 Likert scale the following aspects: (1) *naturalness* —how close to a natural human voice was the speech—, (2) *quality of audio* —degree of noise or distortion they perceived—, and (3) *intelligibility* —how understandable was the message—. Results show that best token in all aspects is the number 5 (Table 2), while some others like token 1 or 2 did not perform that well in terms of intelligibility and naturalness. Probably because the GSTs were trained with an unsupervised approach, some tokens collected more information than others, so different tokens may depend on each other for specific attributes. What perceptually looks more affected was the quality of audio when style weights are focused on a single token.

6. Conclusions and Future Work

An alternative prosody representation to the melspectrogram has been proposed. We observed that training a bank of speech style tokens with only the pitch contour information is sufficient to encode and transfer prosodic attributes to the generated speech. Moreover, having a simpler prosody representation allowed us to implement lighter architectures than previously explored, thus reducing memory during the training stage. These results lead us to continue exploring in depth alternatives for prosody encoding. We plan to implement SPM on the latest approaches, and also including multi-speaker and expressive dataset in the model training.

7. Acknowledgements

This research was partially funded by the Generalitat de Catalunya under the CLIC-UB 2021 SGR 00313 grant, and by MCIN/AEI/10.13039/501100011033 and the European Union under grant PID2021-124361OB-C33 (FairTransNLP-Language project).

8. References

[1] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, and Q. Le, "Tacotron: Towards end-to-end speech synthesis," in *Proceedings of the Interspeech*, 2017, pp. 4006–4010.

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, J. Navdeep, Y. Zongheng, Z. Chen, Y. Zhang, Y. Wang, R. J. 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)*, 2018, pp. 4779–4783.

[3] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep Voice 3: 2000-Speaker Neural Text-to-Speech," in *Proceedings of ICLR*, 2018, pp. 1–15.

[4] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016.

[5] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 14 910–14 921.

[6] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," *CoRR*, vol. abs/2010.05646, 2020. [Online]. Available: https://arxiv.org/abs/2010.05646

[7] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," *CoRR*, vol. abs/2106.06103, 2021. [Online]. Available: https://arxiv.org/abs/2106.06103

[8] E. Casanova, J. Weber, C. Shulby, A. C. Júnior, E. Gölge, and M. A. Ponti, "Yourtts: Towards zero-shot multi-speaker TTS and zero-shot voice conversion for everyone," *CoRR*, vol. abs/2112.02418, 2021. [Online]. Available: https://arxiv.org/abs/2112.02418

[9] J. Betker, "Better speech synthesis through scaling," 2023.

[10] Y. Gao, N. Morioka, Y. Zhang, and N. Chen, "E3 tts: Easy end-to-end diffusion-based text to speech," 2023.

[11] P. Taylor, "Text-to-speech synthesis," *Text-to-Speech Synthesis*, pp. 1–597, 01 2009.

[12] K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. W. Wightman, P. Price, J. B. Pierrehumbert, and J. Hirschberg, "ToBI: A standard for labeling English prosody," in *Proceedings of ICSLP*, 1992.

[13] P. Taylor, "Analysis and synthesis of intonation using the Tilt model," *Journal of the Acoustical Society of America*, vol. 107, pp. 1697–1714, 1998.

[14] A. Rosenberg, "AuToBI - A tool for automatic ToBI annotation," in *Proceedings of the Interspeech*, 2010, pp. 146–149.

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

[16] R. J. 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," *CoRR*, vol. abs/1803.09047, 2018.

[17] Y. Wang, D. Stanton, Y. Zhang, R. J. 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," *CoRR*, vol. abs/1803.09017, 2018.

[18] J.-S. Bae, J. Yang, T.-J. Bak, and Y.-S. Joo, "Hierarchical and multi-scale variational autoencoder for diverse and natural non-autoregressive text-to-speech," 2022.

[19] Y. Ren, M. Lei, Z. Huang, S. Zhang, Q. Chen, Z. Yan, and Z. Zhao, "Prosospeech: Enhancing prosody with quantized vector pre-training in text-to-speech," 2022.

[20] H.-S. Oh, S.-H. Lee, and S.-W. Lee, "Diffprosody: Diffusion-based latent prosody generation for expressive speech synthesis with prosody conditional adversarial training," 2023.

[21] H. Lim, K. Byun, S. Moon, and E. Visser, "Stylebook: Content-dependent speaking style modeling for any-to-any voice conversion using only speech data," 2023.

[22] A. Rosenberg and J. Hirschberg, "On the correlation between energy and pitch accent in read English speech," in *Proc. Interspeech 2006*, 2006, pp. paper 1294–Mon2A3O.2.

[23] Y. Zhang, S. Pan, L. He, and Z. Ling, "Learning latent representations for style control and transfer in end-to-end speech synthesis," *CoRR*, vol. abs/1812.04342, 2018. [Online]. Available: http://arxiv.org/abs/1812.04342

[24] Y. Zhao, H. Li, C.-I. Lai, J. Williams, E. Cooper, and J. Yamagishi, "Improved prosody from learned f0 codebook representations for vq-vae speech waveform reconstruction," 2020.

[25] Y. A. Li, C. Han, and N. Mesgarani, "Styletts: A style-based generative model for natural and diverse text-to-speech synthesis," 2023.

[26] G. Pamisetty and K. S. R. Murty, "Prosody-tts: An end-to-end speech synthesis system with prosody control," 2021.

[27] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," 2022.

[28] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 3617–3621.

[29] I. R. Titze, *Principles of Voice Production*. Englewood Cliffs: Prentice Hall, 1994.

[30] P. Boersma and D. Weenink, "Praat, a system for doing phonetics by computer," *Glot International*, vol. 5, no. 9/10, pp. 341–345, 2001.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008.