UNSUPERVISED WORD-LEVEL PROSODY TAGGING FOR CONTROLLABLE SPEECH SYNTHESIS

Yiwei Guo, Chenpeng Du, Kai Yu[*]

MoE Key Lab of Artificial Intelligence, AI Institute
X-LANCE Lab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{cantabile_kwok, duchenpeng, kai.yu}@sjtu.edu.cn

ABSTRACT

Although word-level prosody modeling in neural text-to-speech (TTS) has been investigated in recent research for diverse speech synthesis, it is still challenging to control speech synthesis manually without a specific reference. This is largely due to lack of word-level prosody tags. In this work, we propose a novel approach for unsupervised word-level prosody tagging with two stages, where we first group the words into different types with a decision tree according to their phonetic content and then cluster the prosodies using GMM within each type of words separately. This design is based on the assumption that the prosodies of different type of words, such as long or short words, should be tagged with different label sets. Furthermore, a TTS system with the derived word-level prosody tags is trained for controllable speech synthesis. Experiments on LJSpeech show that the TTS model trained with word-level prosody tags not only achieves better naturalness than a typical FastSpeech2 model, but also gains the ability to manipulate word-level prosody.

Index Terms— Prosody control, prosody tagging, word-level prosody, speech synthesis

1. INTRODUCTION

Prosody modeling in neural speech synthesis has been extensively explored in recent research, aiming for natural, diverse, and controllable synthesis. The naturalness of synthetic speech is improved with prosody modeling taken into account [1–3]. Recently, more attention has been attracted by rich prosody modeling and control.

Explicit prosodic features, which have clear linguistic or phonological interpretation, are first investigated. [4, 5] both provide solutions to control specific acoustic aspects of phone-level speech. [4] introduces temporal structures in the embedding networks that can control pitch and amplitude either on speech side or text side. [5] proposes a generative model that controls affect and speaking rate with semi-supervised latent variables. [6] effectively controls word-level pitch accent by multiplying optional bias to pitch encoder's output. [7, 8] presents F0, duration and energy control with variational auto-encoders (VAE). They disentangle these prosody features and provide more independent control. [9, 10] model these features with clustering, which is a purely data-driven method that have more interpretability. In contrast to explicit representation, implicit prosody representation is more complete and richer when modelling prosody diversity, yet uninterpretable. Prosody embeddings sampled from prior distribution with VAE are widely investigated in many linguistic levels. [11] models the global characteristics for an utterance. [12] improves the performance by incorporating GMM prior in VAE. [13] enhances phone-level prosody latent representations by VAE in prosody transfer. [14] uses vector quantization and trains an autoregressive prior model to generate synthetic speech with better sound quality. [15–17] models prosody hierarchically, by conditioning phone and word-level latent variables on coarser ones. These works incorporate more semantic information, thus improve the naturalness of synthetic speech to a great extent. Recently, unsupervised prosody clustering with mixture density network is also proposed in [18, 19], enabling richer prosody diversity.

However, all the prior works control the prosodies manually by providing a reference speech or specifying the values of explicit prosodic features, such as pitch, which is hard to be practically applied. For example, it is expensive to collect reference speech with the prosodies that one needs. Also, hand-written values of explicit features may not correspond to a natural speech, and these explicit features do not represent the entire prosody space. As for implicit prosody representations, there are few known methods that can control prosody in inference stage. This is mainly because of the continuous prosody distributions they use. Therefore, few of the existing works achieve good and interpretable controllability with diverse prosody in natural speech.

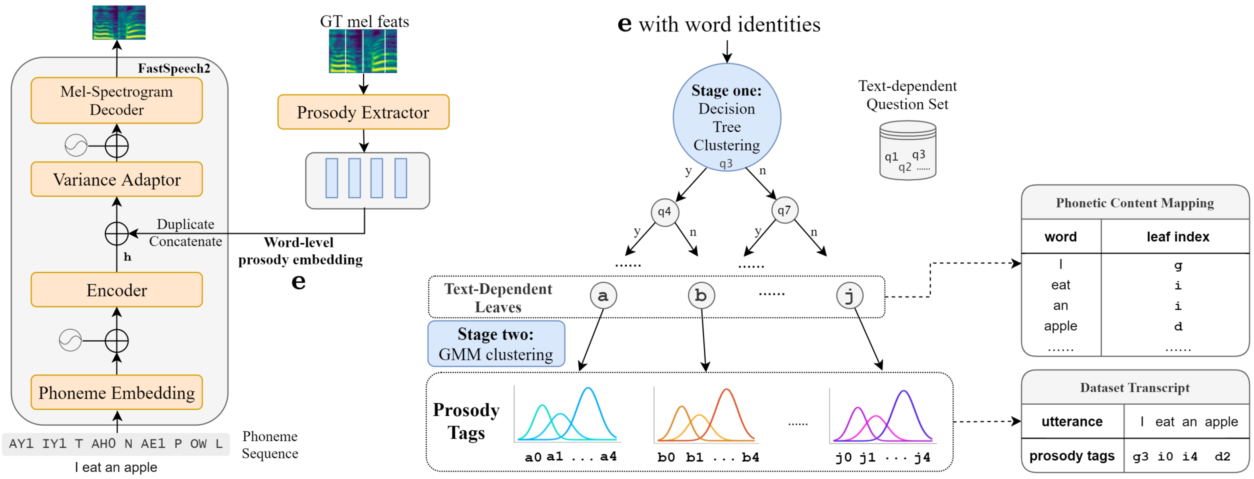
In this work, we propose an unsupervised word-level prosody tagging system that can be directly used for prosody control. We extract prosody embeddings from the mel-spectrogram of reference speech. Then, we obtain the word-level prosody tags in two stages. First, we construct a decision tree that recursively clusters all the words into different text-dependent sets, with a set of questions regarding their phonetic contents. Then, for each text-dependent leaf node, we cluster the prosody embeddings using Gaussian mixture models. The obtained prosody tags represent word-level prosody types and are further embedded to train a TTS system with a prosody tag predictor. The prosody tag predictor is capable of controlling the prosody of synthetic speech by manually specifying the prosody tag of any word.

Our approach has several advantages besides the improved naturalness and controllability. First, the prosody tags are obtained in an unsupervised manner, without the need for expensive manual annotations like emotional labels. Second, the decision tree design makes it easy and robust to generalize to unseen words in inference, in terms of identifying a word into its phonetic cluster. Furthermore, as most of the questions in decision tree are language-agnostic, this design can be easily extended to different languages. By selecting the questions, the tree can also be used for multiple tasks.

The rest of the paper is organized as follows. Section 2 illustrates the overall system. Experiments and results analysis are given in Section 3, and Section 4 draws a conclusion.

[*] Corresponding author

978-1-6654-0540-9/22/$31.00 ©2022 IEEE



(a) Word-level prosody extracting

(b) Two-stage prosody tagging

Fig. 1: Prosody extracting and tagging system architecture

2. WORD-LEVEL PROSODY TAGGING AND CONTROL

Our system is built in three steps: word-level prosody embedding extracting, two-stage word-level prosody tagging, and TTS training with the prosody tags. Note that the TTS models in our system are based on FastSpeech2 [20].

2.1. Word-level prosody extracting

In order to obtain word-level prosody embeddings, we first build a typical FastSpeech2-based TTS model together with a prosody extractor following [19]. As is shown in Fig.1(a), the prosody extractor generates a hidden vector (named as prosody embedding $\mathbf{e}$) for each word from the corresponding mel-spectrogram segment. The generated prosody embeddings are then aligned with the phoneme sequence and concatenated to the encoder output. Accordingly, the extractor is optimized to extract useful information for better reconstructing the output speech, including both prosody information and phonetic contents of the words.

2.2. Prosody tagging with two stages

It is an intuitive idea that words with greatly different phonetic contents, such as the long word ’congratulation’ and the short word ’cat’, are uttered in a completely different ways and consequently should not be tagged with the same set of prosody tags. Therefore, in this work, we design a two-stage prosody tagging strategy, where we first group the words into different types with a decision tree according to their phonetic contents and then cluster the prosodies using GMM within each type of words separately.

2.2.1. *Stage one: decision tree clustering*

Following the HMM state-tying in ASR [21], we construct a binary decision tree for word clustering with a set of questions Q on its phonetic contents, where all the words in the root are clustered into l leaves. These questions are designed based on our expert knowledge, such as ”Whether the phonemes of the word are more than 4 or not?” and ”Whether the word ends with a closed syllable?”. We reference the phonetic questions in HTS[22], which is a direct product of [21].

Each node in the decision tree contains a set of words whose prosody embeddings can be modeled with a Gaussian distribution and the log-likelihood can be formulated as

$$LL^{(i)} = \sum_{\mathbf{e}\in\mathcal{E}^{(i)}} \log \mathcal{N}\left(\mathbf{e} \mid \boldsymbol{\mu}^{(i)}, \boldsymbol{\Sigma}^{(i)}\right) \tag{1}$$

where i is the node index and $\mathcal{E}^{(i)}$ is the set of all prosody embeddings corresponding to the words in the node i. Each non-leaf node i is related to a question q that partitions the words in the node into its left or right child, leading to an increase in log-likelihood of the prosody embeddings

$$\Delta_q LL^{(i)} = LL^{(i\text{'s left child under } q)} + LL^{(i\text{'s right child under } q)} - LL^{(i)}. \tag{2}$$

The initial tree contains only a root node, which is also a leaf node. Then we recursively perform the following step: find the question that maximizes the increase in log-likelihood for all the leaf nodes, and select a leaf node j whose increase is the maximum over all the leaf nodes, which is

$$j = \arg \max_{i\in\text{leaf nodes}} \left(\max_{q\in Q} \Delta_q LL^{(i)}\right), \tag{3}$$

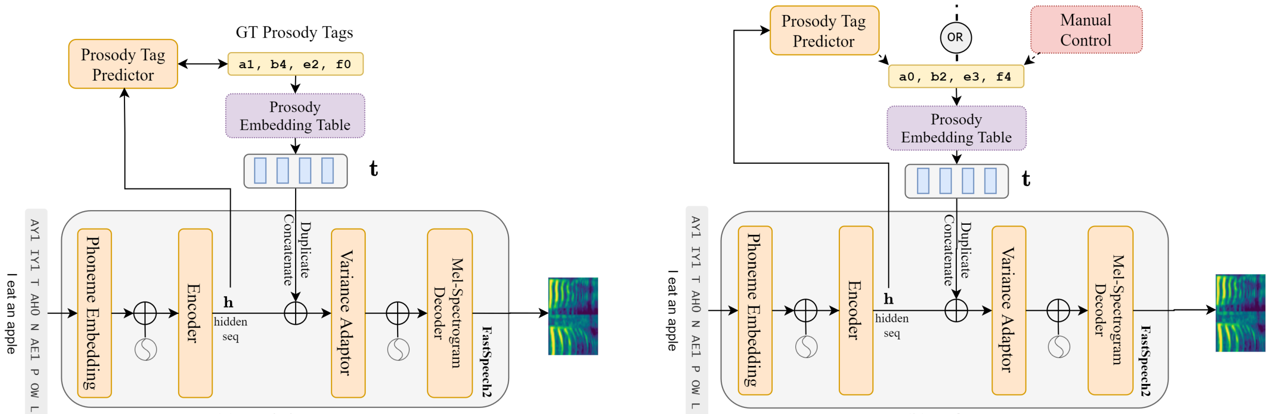
and split the selected node with the corresponding question. This process continues until the increase in log-likelihood is smaller than a threshold. Consequently, the topology of the decision tree is obtained. In this work, the number of leaves l is 10 as shown in Fig. 1(b), whose indices are denoted as letters from `a` to `j`.

2.2.2. *Stage two: Gausian mixture clustering*

The word-level prosody embeddings $\mathbf{e}$ extracted by neural networks contain both prosody information and phonetic content of the words. However, the decision tree clusters the words into l leaves according to the questions only on their phonetic contents, so we assume that the prosody embeddings of the words in a leaf node differ only in prosodies and are similar in phonetic contents. Therefore, clustering within a leaf node is dominated by the prosodies instead of phonetic contents.

We perform GMM-based clustering for the prosody embeddings within each leaf node i separately, which is

$$\mathbf{e}^{(i)} \sim \sum_{k=1}^{m} \omega_k^{(i)} \mathcal{N}\left(\mathbf{e}^{(i)} | \boldsymbol{\mu}_k^{(i)}, \boldsymbol{\Sigma}_k^{(i)}\right) \tag{4}$$



(a) Training

(b) Inference

Fig. 2: Prosody control model architecture in training and inference stage

where k is the Gaussian component index and m is the number of components. The prosody of each word is tagged with the index of the Gaussian component that maximizes the posterior probability of its prosody embedding $\mathbf{e}$

$$t = \arg\max_{k} \left\{ \log \mathcal{N}\left(\mathbf{e} \mid \boldsymbol{\mu}_k^{(i)}, \boldsymbol{\Sigma}_k^{(i)}\right) + \log \omega_k^{(i)} \right\}. \tag{5}$$

In this work, m is set to 5, so the Gaussian component ids range from 0 to 4. Accordingly, all the words in the training set are labelled with the $m \times l = 5 \times 10 = 50$ prosody tags, which is the combination of 10 leaf ids and 5 Gaussian component ids. As shown in Fig. 1(b), the prosody tags are from `a0` to `j4`.

Note that our prosody extracting and tagging system is fully unsupervised in which only audio information is utilized. Also, the tagging system is driven by both data and knowledge.

2.3. Prosody control with prosody tags

Finally, we train a TTS model with the derived word-level prosody tags as shown in Fig.2. In the training stage, the TTS model is guided by prosody embeddings retrieved from a trainable embedding table given the ground-truth prosody tags. In the inference stage, the prosody tags can be either predicted from input text by a prosody predictor or be manually specified.

The prosody predictor in this work is similar to [19]. It predicts the prosody tag for each word given its corresponding phoneme hidden states, i.e. the encoder output sequence $\mathbf{h}$. The prosody predictor contains a bi-GRU that transforms the phoneme hidden states to a vector for each word, two convolutional blocks and a softmax layer. The convolutional blocks here consist of a 1D convolutional layer followed by a ReLU activation layer, layer normalization, and a dropout layer. The predictor is optimized by the cross-entropy loss $\mathcal{L}_{\texttt{PP}}$ with the ground-truth prosody tags. Hence, the overall loss for the model training is defined as

$$\mathcal{L} = \alpha \mathcal{L}_{\texttt{PP}} + \mathcal{L}_{\texttt{FastSpeech2}}, \tag{6}$$

where $\mathcal{L}_{\texttt{FastSpeech2}}$ is the loss of FastSpeech2[20] and α is the relative weight between the two terms.

3. EXPERIMENTS AND RESULTS

3.1. Experimental setup

We use LJSpeech [23], a single-speaker dataset containing about 24 hours of recordings for our experiments. 242 utterances are left out as a test set. All utterances are down-sampled to 16kHz. We use 800-point window length, 200-point hop size, 1024 FFT points, and 320 mel-bins for feature extraction. The phoneme alignment is obtained from an HMM-GMM ASR model trained on Librispeech [24]. The vocoder used in this work is MelGAN [25]. The coefficient α in Eq.(6) is set to 1.0. The prosody embedding $\mathbf{e}$ is 128 dimensional.

3.2. The performance of decision tree in prosody tagging

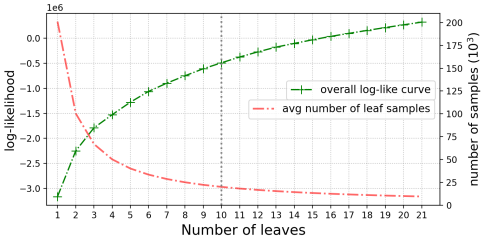


Fig. 3: Curve of overall log-likelihood of leaves and average number of leaf samples

We demonstrate the curve of the average number of prosody embeddings in each leaf node and the overall log-likelihood of prosody embeddings over all leaf nodes $\sum_{i \in \text{leaf nodes}} LL^{(i)}$ in Fig.3 when the tree grows. With the increase of the number of leaf nodes, the average number of prosody embeddings in each leaf node decreases whilst the overall log-likelihood of prosody embeddings increases. We stop the growth of the tree when the number of leaves reaches 10, in consideration of both the performance and the complexity.

3.3. Naturalness of predicted prosodies

The TTS model with a prosody predictor is trained with the derived word-level prosody tags. In the inference stage, the word-level prosodies can be either predicted from the input text by the prosody predictor or be manually specified. In this section, we synthesize the test set whose prosodies are predicted and sampled. Then we evaluate the naturalness with a MUSHRA test in which 30 listeners are asked to rate each utterance in a range from 0 to 100. We compare our model with two baselines: the typical FastSpeech2 model [20] Raw_FSP and a TTS model in which phone-level prosodies are modeled with a mixture density network [19] PLP_MDN. Also, the ground-truth mel-spectrograms of the recordings are reconstructed by the vocoder and then provided as GT in the listening test. The

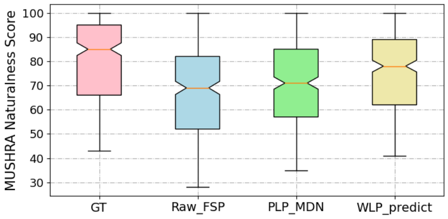


Fig. 4: Subjective evaluation of naturalness

results are reported in Fig.4. It can be observed that our proposed word-level prosody prediction system with predicted prosody tags (WLP_predict) outperforms both other models in terms of naturalness, due to our word-level prosody modelling, although it is still slightly worse than GT.

3.4. Prosody controllability

In order to evaluate the word-level prosody controllability of our TTS model, we first label the ground-truth word prosodies for the test set with the proposed prosody tagging system. Then we synthesize the test set 5 times where the prosody tags of the words in leaf `d` are manually specified as `d0` to `d4` respectively while the prosody tags of other words are predicted and sampled. [1]

Fig. 5 shows an example in which the word “responsibilities” between the yellow dash lines are manually controlled with `d0` to `d4` respectively. It can be observed that all the 5 prosodies of the word are different, showing the controllability of the prosody tags.

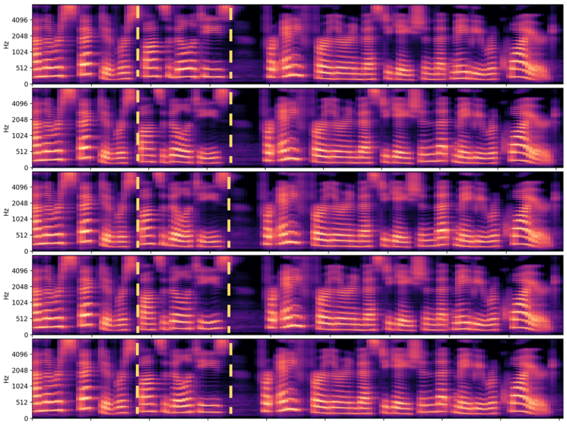


Fig. 5: An example of synthetic speech with manually specified prosodies. The word between the yellow dash lines is “responsibilities” whose prosody tags are specified as `d0` to `d4` respectively.

In addition, we need to confirm that same prosody tags lead to similar prosodies. Therefore, we evaluate the prosody similarity between the recordings and the synthetic speech with different specified prosody tags for all the words in the leaf `d` in the test set. Theoretically, when the specified prosody tag is equal to the ground-truth prosody tag, the word prosody in the synthetic speech should be most similar to the recordings.

We perform the evaluation of prosody similarity in objective and subjective ways respectively. We first compute the average Mel cepstral distortion (MCD) over all the words with ground-truth prosody tag `d`t where t ranges from 0 to 4 between the recordings and the synthetic speech with a certain specified prosody tag. The results are reported in Table 1. As expected, we can find that all the diagonal values are the lowest among the values on their columns, showing that same prosody tags lead to similar prosodies in synthetic speech.

GT Tag Ctrl Tag	0	1	2	3	4
0	**5.389**	5.434	5.371	5.490	5.420
1	5.410	**5.348**	5.379	5.796	5.420
2	5.612	5.670	**5.356**	5.548	5.517
3	5.828	6.023	5.578	**5.442**	5.714
4	5.507	5.507	5.362	5.562	**5.309**

Table 1: Mel cepstral distortion between the recordings and the synthetic speech with different specified prosody tags for all the words in the leaf `d` in the test set.

Also, we evaluate the prosody similarity with a subjective listening test where 30 listeners are provided with the recording and 5 synthetic speech with different prosody tags for each group and are asked to select the synthetic speech whose prosody of the corresponding word is the most similar to the recording. The proportions of the selections are depicted as a confusion matrix in Fig. 6. Similar to the results of objective evaluation, the proportion of the synthetic speech with the same prosody tags to the ground-truth ones, i.e. the diagonal values, achieves the highest among their columns, which further confirms the controllability of prosody tags.

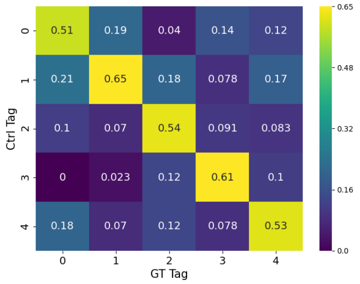


Fig. 6: Subjective evaluation of controllability

4. CONCLUSION

In this work, we propose a novel approach for unsupervised word-level prosody tagging with two stages, where we first group the words into different types with a decision tree according to their phonetic content and then cluster the prosodies using GMM within each type of words separately. Furthermore, a TTS system with the derived word-level prosody tags is trained for controllable speech synthesis, where the prosody can be either predicted from input text or manually specified. Experiments on LJSpeech show that our model achieves better naturalness than a typical FastSpeech2 model with the predicted prosodies. In addition, the objective and subjective evaluations for prosody controllability show that the prosodies can be efficiently controlled by specifying the word-level prosody tags.

5. ACKNOWLEDGEMENTS

This study was supported by State Key Laboratory of Media Convergence Production Technology and Systems Project (No. SKLMCPTS2020003) and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102).

[1] The audio examples are available at `https://cantabile-kwok.github.io/word-level-prosody-tagging-control/`

6. REFERENCES

[1] KC Rajeswari and MP Uma, "Prosody modeling techniques for text-to-speech synthesis systems–a survey," *International Journal of Computer Applications*, vol. 39, no. 16, pp. 8–11, 2012.

[2] Chung-Yao Tsai, Chin-Kuan Kuo, Yih-Ru Wang, Sin-Horng Chen, I-Bin Liao, and Chen-Yu Chiang, "Hierarchical prosody modeling of english speech and its application to tts," in *2014 17th Oriental Chapter of the International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques (COCOSDA)*. IEEE, 2014, pp. 1–6.

[3] V Ramu Reddy and K Sreenivasa Rao, "Prosody modeling for syllable based text-to-speech synthesis using feedforward neural networks," *Neurocomputing*, vol. 171, pp. 1323–1334, 2016.

[4] Younggun Lee and Taesu Kim, "Robust and fine-grained prosody control of end-to-end speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pp. 5911–5915, IEEE.

[5] Raza Habib, Soroosh Mariooryad, M. Shannon, Eric Battenberg, R. Skerry-Ryan, Daisy Stanton, David Kao, and Tom Bagby, "Semi-supervised generative modeling for controllable speech synthesis," .

[6] Cheng Gong, Longbiao Wang, Zhenhua Ling, Shaotong Guo, Ju Zhang, and Jianwu Dang, "Improving naturalness and controllability of sequence-to-sequence speech synthesis by learning local prosody representations," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pp. 5724–5728, IEEE.

[7] D. Mohan, Qinmin Hu, Tian Huey Teh, Alexandra Torresquintero, C. Wallis, Marlene Staib, Lorenzo Foglianti, Jiameng Gao, and S. King, "Ctrl-p: Temporal control of prosodic variation for speech synthesis," *ArXiv*, vol. abs/2106.08352, 2021.

[8] Xiaochun An, Frank K. Soong, Shan Yang, and Lei Xie, "Effective and direct control of neural tts prosody by removing interactions between different attributes," *Neural Networks*, vol. 143, pp. 250–260, 2021.

[9] András Beke and György Szaszák, "Unsupervised clustering of prosodic patterns in spontaneous speech," in *Text, Speech and Dialogue*, Petr Sojka, Aleš Horák, Ivan Kopeček, and Karel Pala, Eds. pp. 648–655, Springer Berlin Heidelberg.

[10] Alexandra Vioni, Myrsini Christidou, Nikolaos Ellinas, Georgios Vamvoukakis, Panos Kakoulidis, Taehoon Kim, June Sig Sung, Hyoungmin Park, Aimilios Chalamandaris, and Pirros Tsiakoulis, "Prosodic clustering for phoneme-level prosody control in end-to-end speech synthesis," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. pp. 5719–5723, IEEE.

[11] Kei Akuzawa, Yusuke Iwasawa, and Yutaka Matsuo, "Expressive speech synthesis via modeling expressions with variational autoencoder," *arXiv preprint arXiv:1804.02135*, 2018.

[12] Wei-Ning Hsu, Yu Zhang, Ron J Weiss, Heiga Zen, Yonghui Wu, Yuxuan Wang, Yuan Cao, Ye Jia, Zhifeng Chen, Jonathan Shen, et al., "Hierarchical generative modeling for controllable speech synthesis," *arXiv preprint arXiv:1810.07217*, 2018.

[13] Viacheslav Klimkov, Srikanth Ronanki, Jonas Rohnke, and Thomas Drugman, "Fine-grained robust prosody transfer for single-speaker neural text-to-speech," 2019.

[14] Guangzhi Sun, Yu Zhang, Ron J Weiss, Yuan Cao, Heiga Zen, Andrew Rosenberg, Bhuvana Ramabhadran, and Yonghui Wu, "Generating diverse and natural text-to-speech samples using a quantized fine-grained vae and autoregressive prosody prior," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6699–6703.

[15] Chung-Ming Chien and Hung-yi Lee, "Hierarchical prosody modeling for non-autoregressive speech synthesis," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. pp. 446–453, IEEE.

[16] Yukiya Hono, Kazuna Tsuboi, Kei Sawada, Kei Hashimoto, Keiichiro Oura, Yoshihiko Nankaku, and Keiichi Tokuda, "Hierarchical multi-grained generative model for expressive speech synthesis," *arXiv preprint arXiv:2009.08474*, 2020.

[17] Guangzhi Sun, Yu Zhang, Ron J Weiss, Yuan Cao, Heiga Zen, and Yonghui Wu, "Fully-hierarchical fine-grained prosody modeling for interpretable speech synthesis," in *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. pp. 6264–6268, IEEE.

[18] Chenpeng Du and Kai Yu, "Phone-level prosody modelling with gmm-based mdn for diverse and controllable speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 190–201, 2022.

[19] Chenpeng Du and Kai Yu, "Rich Prosody Diversity Modelling with Phone-Level Mixture Density Network," in *Proc. Interspeech 2021*, 2021, pp. 3136–3140.

[20] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," *arXiv preprint arXiv:2006.04558*, 2020.

[21] Steve J Young, Julian J Odell, and Phil C Woodland, "Tree-based state tying for high accuracy modelling," in *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

[22] Heiga Zen, Takashi Nose, Junichi Yamagishi, Shinji Sako, Takashi Masuko, Alan W Black, and Keiichi Tokuda, "The hmm-based speech synthesis system (hts) version 2.0.," in *SSW*. Citeseer, 2007, pp. 294–299.

[23] Keith Ito and Linda Johnson, "The lj speech dataset," `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[24] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[25] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brebisson, Yoshua Bengio, and Aaron Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," 2019.