HIERARCHICAL PROSODY MODELING FOR NON-AUTOREGRESSIVE SPEECH SYNTHESIS

Chung-Ming Chien *Hung-yi Lee*

College of Electrical Engineering and Computer Science, National Taiwan University, Taiwan

ABSTRACT

Prosody modeling is an essential component in modern text-to-speech (TTS) frameworks. By explicitly providing prosody features to the TTS model, the style of synthesized utterances can thus be controlled. However, predicting natural and reasonable prosody at inference time is challenging. In this work, we analyzed the behavior of non-autoregressive TTS models under different prosody-modeling settings and proposed a hierarchical architecture, in which the prediction of phoneme-level prosody features are conditioned on the word-level prosody features. The proposed method outperforms other competitors in terms of audio quality and prosody naturalness in our objective and subjective evaluation.

Index Terms— hierarchical prosody, prosody prediction, text-to-speech

1. INTRODUCTION

In order to synthesize human-like speech utterances by Text-to-speech (TTS), it is important to model the variation in speech signals, including rhythm, intonation, and stress, etc. These factors, collectively referred to as *prosody*, are not contained in the text transcripts, but are very important for conveying information that is not specified by the texts. Providing additional prosody information to the TTS model is referred to as *prosody modeling*, which enables expressive and controllable speech synthesis [1, 2, 3, 4, 5].

In this work, we focus on fine-grained prosody modeling [5], where the prosody of an utterance is represented as a sequence of prosody features instead of a single sentence-level prosody feature. Each fine-grained prosody feature encodes the prosody associated with a speech segment, such as a phoneme or a word. Aside from enabling local prosody control in speech synthesis, fine-grained prosody modeling also further reduces the complexity of the TTS task itself, since the information contained in each fine-grained prosody feature is explicitly assigned to a speech segment. This makes prosody modeling especially an important component in non-autoregressive TTS framework [6, 7]. Compared with autoregressive models that suffer from the false alignment problem, non-autoregressive TTS systems are faster and more robust [8], but the training is much more difficult since the model has to predict the entire mel-spectrogram simultaneously. The prosody features provide additional information to the TTS model and effectively simplify the mapping between the text input and the speech output.

Many handcrafted features, such as fundamental frequency (F0) contour and energy [6, 7], and even neural-based features [1, 2, 3, 4, 5, 9], can be used to model the prosody variation within an utterance. The prosody features can be extracted from ground-truth speech signals at training time, while *how to generate natural prosody features at inference time* remains an open problem. Some proposed to infer prosody features from phoneme-level features [6, 7, 9, 10]. However, we consider that the attributes that affect the prosody of an utterance, such as the meaning of the sentence, and the speaker's intention and sentiment, can be better realized with word-level features rather than the phoneme-level ones. The effect of different granularity in fine-grained prosody modeling is studied in this work, and the experimental results verify the above hypothesis.

There are three major contributions of this paper. First, we figure out that there is a trade-off between the quality of synthesized audio samples and the accuracy of prosody prediction, with respect to the granularity of fine-grained prosody modeling. Second, we compare different approaches to extracting prosody features in terms of audio quality and prosody naturalness. Last, we propose a hierarchical prosody modeling framework, where phoneme-level prosody prediction is conditioned on word-level prosody prediction, to combine the advantage of phoneme-level and word-level prosody modeling. With both objective and subjective evaluation, we verify that the proposed hierarchical model outperforms any other prosody modeling frameworks of interest. The readers are encouraged to listen to the audio samples attached here [1].

2. BACKGROUND

A typical TTS model takes a phoneme sequence as inputs to predict mel-spectrogram or raw waveform. As shown in Fig. 1, prosody features can serve as extra inputs of the TTS model, which simplifies the one-to-many mapping from phoneme sequence to speech signals. At training time, the prosody features are extracted from the ground-truth, en-

[1] Audio sample: `https://ming024.github.io/hierarchical_prosody_modeling`

abling the TTS model to generate speech signals with prosody similar to the ground-truth utterance. The prosody features can be obtained with the following methods:

ⓐ Rule-based prosody features: F0 and energy can be computed with many off-the-shelf packages, and are often used to model prosody variation [6, 7].

ⓑ Neural-based prosody features: a reference encoder is used to drain information from the ground-truth utterance [1, 2, 3, 4, 5, 9]. Suppose the size of the representation is small enough. In that case, it is be expected that the information extracted from the ground-truth is mostly about prosody and other information that is not contained in the phoneme features, since the TTS model can obtain phonetic information from the input phoneme sequence [1].

At inference time, because we do not know the prosody of the generated audio, prosody features are generated by the approaches different from the training phase. There are a variety of possible methods to generate prosody features. We summarize them into four categories, as shown in Fig. 1:

① Predicting prosody from phoneme feature sequence [6, 7, 9, 10].

② Predicting prosody from word-level feature sequence. There has been research trying to predict prosody features from word-level features [11], whereas few works used prosody features predicted from word-level features to help speech synthesis.

③ Imitating the prosody of a reference utterance, by conditioning the speech synthesis of arbitrary texts on prosody features extracted from the reference, usually used in prosody transfer tasks [1, 2, 3, 4, 5].

④ Sampling prosody from a prior distribution [2, 3, 4]

This paper will focus on inferring prosody from text inputs, that is, phoneme and word feature sequences.

It has been shown that hierarchical structures of prosody intrinsically exist in spoken language [12]. Some previous work utilizes the hierarchical property of language to help speech synthesis. [13, 14] used an autoregressive model to predict frame-level prosody values and mel-spectrogram from hierarchical linguistic features. However, their model only handles prosody prediction at frame-level, and the underlying prosody embedding is not fine-grained. [15] proposed a TTS framework with hierarchical fine-grained prosody modeling, where the word-level and phoneme-level prosody features are encoded in two separate embedding spaces. The previous work presented a multi-level prosody modeling structure, but the generation of prosody embedding at inference time is independent of text inputs. In our work, both word-level and phoneme-level prosody is modeled. At inference time, the multi-level prosody is predicted from phoneme-level

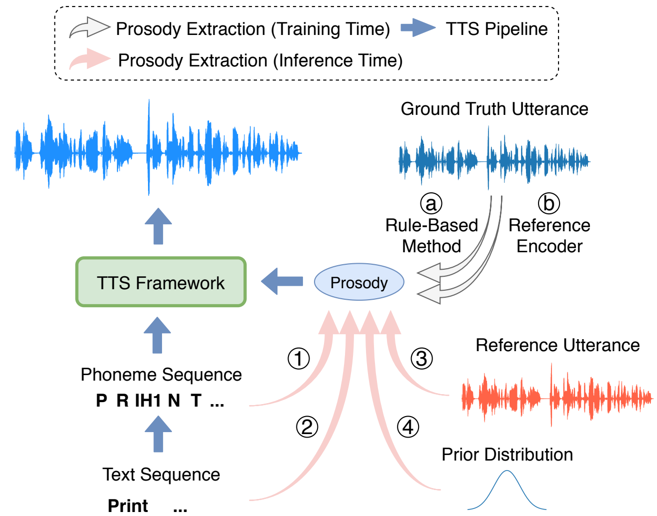

Fig. 1: General pipeline for TTS systems with prosody modeling.

and word-level features, which are derived from the input text sequence. Our model is compatible with any pretrained word embedding, so the word-level prosody prediction benefits from the widely developed word representation models pretrained from large amounts of unlabeled texts.

3. MODEL ARCHITECTURE

In this section we introduce our model architecture based on FastSpeech 2 [6]. The architecture comprises the basic components that are essential for generating valid speech signals and prosody-modeling components, as shown in Fig. 2. The proposed hierarchical architecture is in Section 3.4.

3.1. Basic components

The basic components in FastSpeech 2 include two Feed-Forward Transformer (FFT) [16] stacks and a duration predictor. The configuration and hyper-parameters of the FFT blocks and the duration predictor in our implementation follow FastSpeech 2. The first FFT stack converts the phoneme sequence into a phoneme-level feature sequence in the resolution of the input sequence length. The duration predictor takes the phoneme-level feature sequence as inputs to predict the duration of each phoneme. Then the phoneme-level features are expanded in time according to each phoneme's duration to match the dimensionality of the output mel-spectrogram. At training time, ground-truth duration obtained with the Montreal Forced Aligner (MFA) [17] is used for expansion while at inference time predicted duration is used. The expanded feature sequence is then fed into the second FFT block, followed by a linear layer to predict mel-spectrogram.

There are some differences between our implementation and the original FastSpeech 2. We use an additional Post-Net, which is the same as that used in Tacotron 2 [18], to post-process the output. Mean Squared Error (MSE) loss is

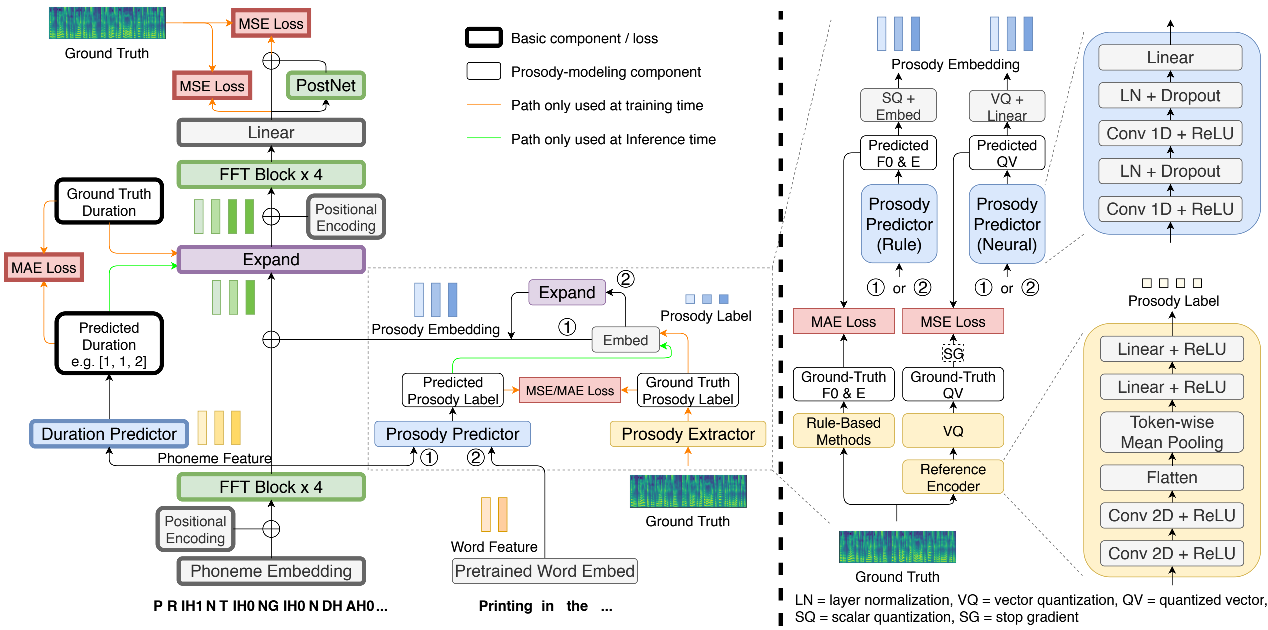

Fig. 2: **Left**: An overview of our model architecture. **Right**: A close-up view of the prosody-modeling components.

computed for both before- and after- post-processing output, while the original FastSpeech 2 uses Mean Absolute Error (MAE). We also replace the MSE loss for duration prediction with MAE loss. These minor modifications are beneficial to training efficiency and stability. The entire network is trained in an end-to-end manner with the duration prediction loss and the mel-spectrogram prediction losses, including before- and after- post-processing parts.

3.2. Prosody extraction

The prosody-modeling components include the prosody extractors and the prosody predictors, as depicted in Fig. 2. At training time, fine-grained prosody labels are extracted from the ground-truth mel-spectrogram. Each prosody label contains the prosody information of a short speech segment (i.e., a word or a phoneme, collectively referred to as *token*). The prosody labels are scalars or low-dimensional vectors. They are mapped to a prosody embedding whose dimension equal to the dimension of the phoneme-level feature. The prosody embedding sequence is then added to the phoneme-level feature sequence and fed into the second FFT stack for mel-spectrogram prediction.

We use two different configurations for prosody extractor, the rule-based one and the neural-based one, to extract the fine-grained prosody information. We will go through their model architecture in detail in the following paragraphs.

3.2.1. Rule-based prosody extractors

Following FastSpeech 2, we use F0 and energy as the rule-based prosody labels. We use the DIO algorithm [19] for F0 estimation and the L2-norm of each Short-Time Fourier Transform (STFT) frame for energy estimation to extract frame-level prosody labels. The F0 and energy values are then averaged over the duration of each token to get token-level prosody labels, as proposed in [7]. The averaged values are then quantized into 256 bins, and transformed into a prosody embedding sequence by an embedding lookup[2].

3.2.2. Neural-based prosody extractors

Inspired by [9], we use a Vector-Quantized Variational Autoencoder (VQ-VAE) [20] as the reference encoder. The reference encoder is jointly learned with the TTS model to extract the prosody information from the ground-truth mel-spectrogram. The reference encoder extracts a 3-dimensional latent representation for each token from the ground-truth mel-spectrogram. The quantization in VQ-VAE serves as an information bottleneck, restricting the amount of information flow from the ground-truth to the mel-spectrogram prediction and thus encouraging the reference encoder to extract prosody information, which is not contained in the text inputs.

The structure of the reference encoder can also be found in Fig. 2. The reference encoder comprises a stack of two 2D convolution layers, each composed of 32 filters with 3×3 kernel size and 1×1 stride. A flatten layer followed, and a token-wise mean pooling is used to transform the frame-level feature sequence into a token-level feature sequence. Then the following two linear layers projected into a 3-dimensional latent space. All the layers are followed by a dropout layer with a 0.2 dropout rate.

[2]The only difference between the workflow here and FastSpeech 2 is that we use token-level instead of frame-level prosody labels. However, in FastSpeech 2, the prosody labels are actually predicted from an expanded phoneme-level feature sequence, which only differs from the inputs of our phoneme-level prosody prediction network by an expansion.

A VQ codebook, consists of 256 codewords, is used to quantize the 3-dimensional latent vector to the nearest codeword (measured with L2 distance). These prosody labels are passed to a linear layer to get prosody embeddings, which are then added to the phoneme feature sequence to predict mel-spectrogram. A VQ loss is used to push the latent vectors and the codewords towards each other.

The quantized vectors from the reference encoder are treated as ground-truth prosody labels. The prosody predictor is then learned to predict the ground-truth prosody labels from the phoneme-level or word-level features from text input. The goal of reference encoder is to provide the ground-truth prosody labels, so its parameters are fixed when training the prosody predictor, and not used at inference time.

3.3. Prosody prediction

At inference time, since ground-truth prosody labels are not available, we have to predict the prosody of an utterance from the text input. As described in Section 2, both phoneme-level features and word-level features can serve as inputs for prosody prediction. For word-level features, there are a variety of pretrained language models [21, 22] that can be used as a good word-embedding. For phoneme-level features, we follow the configuration of FastSpeech 2, in which the output of the first FFT stack is used to predict the prosody labels.

The model architectures of the prosody predictors are shown in Fig. 2, whose network architectures are the same as the duration model in FastSpeech 2. When the prosody predictor takes word-level features as input, the predicted word-level prosody embedding sequence is expanded in time according to the word's phoneme number. The prosody predictors are trained with MAE loss if rule-based prosody labels are used, and MSE loss for neural-based prosody labels, regardless of input features.

3.4. Hierarchical prosody modeling

Under our model architecture, it is possible to select different combinations of inputs (phoneme features or word features) and targets (rule-based or neural-based prosody labels) to train the prosody predictor. Our preliminary experiments showed that word features provide more accurate prosody prediction than phoneme features. However, the low-resolution of word-level prosody embeddings hinders the quality of synthesized audio samples, since the TTS model tends to predict a blurred mel-spectrogram if there is not enough information given. As a result, we design a hierarchical prosody modeling architecture, which is composed of a concatenation of a word-level prosody predictor and a phoneme-level prosody predictor, as shown in Figure 3.

In the proposed architecture, word-level prosody is first predicted by the word-level prosody predictor and expanded to match the phoneme sequence length. The expanded embedding sequence is then added to the phoneme feature sequence. By feeding the summed feature sequence into the

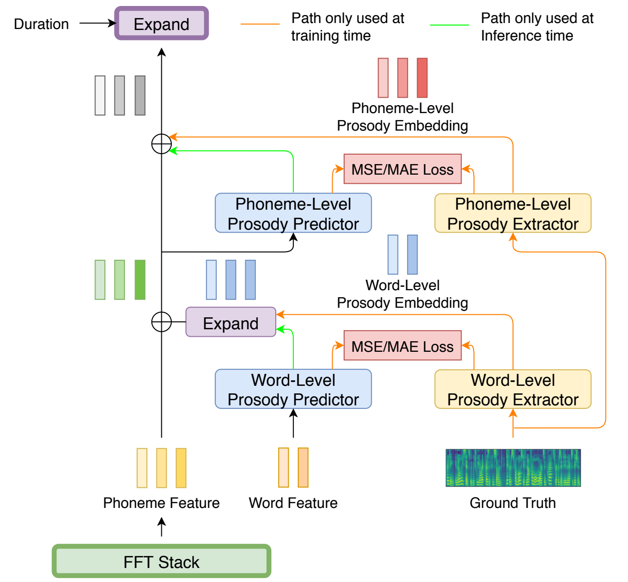

Fig. 3: Proposed hierarchical prosody modeling architecture.

phoneme-level prosody predictor, the prediction of the fine-grained phoneme-level prosody is thus conditioned on the result of the coarse-grained word-level prosody prediction.

The proposed hierarchical prosody modeling benefits from high-resolution phoneme-level prosody labels and accurate word-level prosody prediction simultaneously at inference time. In this framework, both rule-based and phoneme-based prosody labels can be used to model phoneme-level or word-level prosody. We will compare different hierarchical paradigms with the non-hierarchical ones in Section 4.

4. EXPERIMENTS

4.1. Setup

All of our models are trained on the LJSpeech dataset [23], which contains 13100 English utterances spoken by a female speaker. We keep 892 sentences (with document title LJ001, LJ002, and LJ003) for testing, and the remaining are used for training. MFA is used to convert the transcripts into phoneme sequences and discover the alignments between the phoneme sequences and the utterances. The audio samples are converted to 80-dimensional mel-spectrograms for training, and predicted mel-spectrograms are converted to raw waveform with pretrained MelGAN vocoder. Because the source code of FastSpeech 2 has not been released up to the time the paper was written, we used our implementation in the experiments [3].

At training time, the models are trained with Adam optimizer [24] (with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$) for 300k steps with batch size 16, and the learning rate scheduling proposed in [16] is applied. The only exception is the word-level prosody predictor. Since the loss and gradient flow of the word-level prosody predictor are independent of all other

[3] FastSpeech 2: `https://github.com/ming024/FastSpeech2`

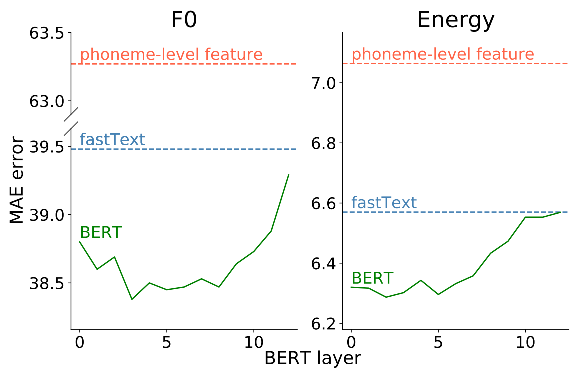

Fig. 4: MAE error of F0 and energy predictor on the testing set, with phoneme-level features, fastText features and the outputs of the 0-th to 12-th layer of BERT as inputs.

modules, we train this module separately with Adam optimizer (with learning rate 10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$) for 30k steps with batch size 16.

4.2. Predictability of prosody labels from different features

This experiment is designed to compare the performance of predicting prosody labels from phoneme-level features and word-level features. We use FastText [21] and pretrained BERT [22, 25] to generate word-level feature sequences, which are used as the input of the word-level prosody predictor. For a fair comparison, pretrained BERT is treated as a static feature extractor without fine-tuning. To further study the influence of context information in prosody prediction, we also use BERT's hidden features from different layers as the input of the word-level prosody predictor.

We use the MAE loss of the rule-based prosody predictors over the testing set for evaluation. The result is shown in Figure 4. All word-level features outperform phoneme-level features, verifying the hypothesis that word features can model prosody variation better than phoneme features. For different word embeddings, despite that BERT's performance is better than fastText, we cannot conclude that contextualized information helps prosody prediction. Instead, we think contextualized information does not make any noticeable difference since the hidden features from high layers do not beat low-layer features[4]. We will use the 0-th layer BERT features (i.e., the input word embedding of BERT) as the word-level features in all following experiments. Similar results are also observed in neural-based prosody label prediction, which is not presented due to space limitation.

4.3. Objective evaluation

It is difficult to evaluate the quality and naturalness of synthesized audio samples with objective metrics. However, there

[4]The higher layers of BERT are generally considered to be more contextualized [26].

	GPE	VDE	FFE	F-MAE	E-MAE
vanilla	.4063	.2856	.4493	42.829	8.205
P+R	.4084	.2836	.4660	41.806	7.363
P+N	.4113	.2898	.4549	43.385	7.441
P+N, rand.	.5278	.3436	.5278	57.119	9.290
W+R	.3952	.2800	.4498	40.202	7.264
W+N	.3977	.2972	.4494	42.096	8.050
W+N, rand.	.4759	.2983	.4824	48.924	8.265
H(W+R, P+R)	.3998	.2812	.4614	40.190	7.308
H(W+R, P+N)	**.3886**	**.2758**	.4499	**39.597**	7.263
H(W+N, P+R)	.3971	.2832	.4529	40.240	**7.145**
H(W+N, P+N)	.3994	.2908	**.4434**	42.074	7.512

vanilla = no prosody modeling, P = Phoneme-level feature, W = Word-level feature, R = Rule-based prosody labels (i.e. F0 and energy), N = Neural-based prosody label (i.e. VQ-VAE codeword), H=Hierarchical model, rand. = prosody label randomly sampled from the uniform prior of VQ-VAE.

Table 1: Objective prosody scores for TTS models with different prosody-modeling frameworks.

are several metrics used in early works to evaluate pairwise prosody similarity [1, 27]:

- Gross Pitch Error (GPE): measuring the pitch similarity between two utterances.
- Voice Decision Error (VDE): measuring the difference of voiced/unvoiced decision between two utterances.
- F0 Frame Error (FFE): combination of GPE and VDE.

For the above metrics, We use the DIO algorithm for the estimation of F0 and the frame-wise voiced/unvoiced decision of both ground-truth and synthesized audio samples. To match the length of two utterances, Dynamic Time Warping (DTW) [28] is used to align the mel-spectrograms, and the resulted alignment is applied to the F0 values and voiced/unvoiced decisions of both utterances.

In addition to the metrics above, we also use the MAE of F0 and energy, which is more compatible with our training objective, to evaluate the synthesized utterances' prosody. Let f_t, f'_t be the estimated F0, v_t, v'_t be the binary voiced/unvoiced decision, e_t and e'_t be the L2 norm of the t-th STFT frame (after alignment) of the ground-truth and the synthesized audio sample, and T be the number of frames after alignment. We define two metrics *F-MAE* and *E-MAE* as below:

$$\textit{F-MAE} = \frac{\sum_{t=1}^{T} |f_t - f'_t| \mathbb{1}[v_t] \mathbb{1}[v'_t]}{\sum_{t=1}^{T} \mathbb{1}[v_t] \mathbb{1}[v'_t]},$$

$$\textit{E-MAE} = \frac{\sum_{t=1}^{T} |e_t - e'_t|}{T}$$

To reduce the differences caused by vocoding, we use the utterances reconstructed by MelGAN from ground-truth mel-spectrograms as the ground-truth for all the objective metrics.

The result is shown in Table 1. It can be seen that for the non-hierarchical models, word-level features achieve better performance than phoneme-level features (W+* v.s. P+*). The models trained with rule-based prosody labels are slightly better than those trained with neural-based features in most cases (*+R v.s. *+N). We hypothesize it is because the neural-based labels are too complicated to predict them accurately. However, the scores of the utterances synthesized with predicted neural-based labels are still much better than randomly sampled prosody labels (P+N, rand., W+N, rand.).

The scores of the four hierarchical models are generally better than their non-hierarchical counterparts. However, it is not clear which hierarchical model is the best, that is, which type of prosody label is the most effective one at which level. One should note that the metrics used in Table 1 only measure the similarity of prosody to ground truth, not the quality of audio. The issue will be addressed in the next subsection.

4.4. Subjective evaluation

We conducted three different subjective assessments, including Mean Opinion Score (MOS), Comparison MOS (CMOS) [29] and the AXY test [1], for audio quality and prosody naturalness evaluation. 100 sentences randomly sampled from the evaluation set are used for all the assessments. In each assessment, every utterance (or utterance pair, for CMOS and AXY) received five ratings from crowdsourced human raters.

4.4.1. MOS score

MOS score is used to evaluate the quality of audio samples synthesized by proposed hierarchical models and other models. The results are listed in Table 5. All TTS models with prosody modeling achieved better performance than the vanilla baseline. Looking into the non-hierarchical models, we find that despite accurate prosody prediction observed in Table 1, word-level prosody modeling is still inferior to phoneme-level prosody modeling in terms of MOS scores. We carefully inspect the audio samples and find that there are more artifacts and noises in audio samples synthesized with word-level models. The information provided by the coarse-grained prosody labels is not specific enough, hindering the model from generating clear speech. It can also be seen that the non-hierarchical models with rule-based prosody labels are better than those with neural-based prosody labels.

The MOS scores of hierarchical models and phoneme-level models are very close, which implies that the granularity of prosody modeling is important for perceptual audio quality. The audio samples of the best hierarchical model H(W+R, P+N) are even comparable to the samples reconstructed from ground-truth mel-spectrograms, reflecting the effectiveness of prosody modeling.

4.4.2. Pairwise subjective scores

We further conduct CMOS [29] and AXY test [1] to compare hierarchical models and non-hierarchical models pairwisely.

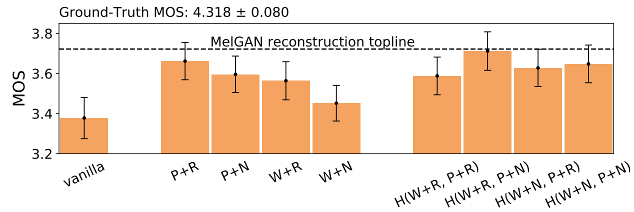

Fig. 5: 5-point scale MOS naturalness evaluation of the TTS models. The error bars indicate the 95 % confidence intervals.

Table 2: CMOS score and AXY reference similarity score of the proposed hierarchical model (W+R, P+N) against non-hierarchical models. Both scores are in a -3 $\sim$ 3 scale, with 0 begin neutral and the larger the better. The t-test p-values are also reported.

In the CMOS test, the human raters are asked to select one audio sample with better quality and naturalness out of two. In the AXY test, the raters are presented two synthesized audio samples and a ground-truth, and they are asked to select one sample with prosody more similar to the ground-truth, regardless of audio quality and any other factors. Before the AXY test, four audio samples are used to instruct the raters to ignore audio quality and focus on prosody. The hierarchical model with word-level rule-based labels and phoneme-level neural-based labels is used for all the pairwise comparison tests since it outperformed any other hierarchical models in terms of MOS score. The results are listed in Table 2.

The CMOS scores are consistent with the MOS scores, which shows that the hierarchical models are slightly better than the phoneme-level models in terms of naturalness, while the difference is not significant. The AXY test reflects that the prosody of audio samples synthesized with the proposed hierarchical model is better than all the non-hierarchical models, consistent with Table 1. We further find that the phoneme-level models are better than the word-level models, which contradicts the objective scores. We think that it is because the raters are easily affected by the quality of speech samples, thus unaware of subtle prosody variation.

5. CONCLUSION

The result shows that word-level prosody modeling achieves more accurate prosody prediction than phoneme-level ones, but subjective tests show that the perceptual quality degrades due to coarse granularity. The proposed hierarchical prosody modeling framework combines the advantages of both word- and phoneme-level, thus capable of generating high-quality audio samples with accurate prosody.

6. REFERENCES

[1] RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif A. Saurous, "Towards end-to-end prosody transfer for expressive speech synthesis with tacotron," in *Proceedings of the 35th International Conference on Machine Learning*, Jennifer Dy and Andreas Krause, Eds., Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018, vol. 80 of *Proceedings of Machine Learning Research*, pp. 4693–4702, PMLR.

[2] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ-Skerry Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *Proceedings of the 35th International Conference on Machine Learning*, Jennifer Dy and Andreas Krause, Eds., Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018, vol. 80 of *Proceedings of Machine Learning Research*, pp. 5180–5189, PMLR.

[3] Y. Zhang, S. Pan, L. He, and Z. Ling, "Learning latent representations for style control and transfer in end-to-end speech synthesis," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6945–6949.

[4] Wei-Ning Hsu, Yu Zhang, Ron Weiss, Heiga Zen, Yonghui Wu, Yuxuan Wang, Yuan Cao, Ye Jia, Zhifeng Chen, Jonathan Shen, Patrick Nguyen, and Ruoming Pang, "Hierarchical generative modeling for controllable speech synthesis," in *International Conference on Learning Representations*, 2019.

[5] Y. Lee and T. 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)*, 2019, pp. 5911–5915.

[6] 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," 2020.

[7] Adrian Łańcucki, "Fastpitch: Parallel text-to-speech with pitch prediction," 2020.

[8] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, "Fastspeech: Fast, robust and controllable text to speech," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. dÁlché-Buc, E. Fox, and R. Garnett, Eds., pp. 3171–3180. Curran Associates, Inc., 2019.

[9] G. Sun, Y. Zhang, R. J. Weiss, Y. Cao, H. Zen, A. Rosenberg, B. Ramabhadran, and Y. 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)*, 2020, pp. 6699–6703.

[10] 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)*, 2018, pp. 595–602.

[11] Aarne Talman, Antti Suni, Hande Celikkanat, Sofoklis Kakouros, Jörg Tiedemann, and Martti Vainio, "Predicting prosodic prominence from text with pre-trained contextualized word representations," in *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, Turku, Finland, Sept.–Oct. 2019, pp. 281–290, Linköping University Electronic Press.

[12] Antti Suni, Juraj Šimko, Daniel Aalto, and Martti Vainio, "Hierarchical representation and estimation of prosody using continuous wavelet transform," *Computer Speech and Language*, vol. 45, pp. 123 – 136, 2017.

[13] Tom Kenter, Vincent Wan, Chun-an Chan, Rob Clark, and Jakub Vit, "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, 9-15 June 2019, Long Beach, California, USA*, Kamalika Chaudhuri and Ruslan Salakhutdinov, Eds. 2019, vol. 97 of *Proceedings of Machine Learning Research*, pp. 3331–3340, PMLR.

[14] Vincent Wan, Jonathan Shen, Hanna Silen, and Rob Clark, "Modelling intonation in spectrograms for neural vocoder based text-to-speech," in *Speech Prosody 2020*, 2020.

[15] G. Sun, Y. Zhang, R. J. Weiss, Y. Cao, H. Zen, and Y. 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)*, 2020, pp. 6264–6268.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia 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., pp. 5998–6008. Curran Associates, Inc., 2017.

[17] Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi," 08 2017, pp. 498–502.

[18] 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)*, 2018, pp. 4779–4783.

[19] Masanori Morise, Hideki Kawahara, and Haruhiro Katayose, "Fast and reliable f0 estimation method based on the period extraction of vocal fold vibration of singing voice and speech," in *Audio Engineering Society Conference: 35th International Conference: Audio for Games*, Feb 2009.

[20] Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu, "Neural discrete representation learning," 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., pp. 6306–6315. Curran Associates, Inc., 2017.

[21] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov, "Fasttext.zip: Compressing text classification models," *arXiv preprint arXiv:1612.03651*, 2016.

[22] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew, "Huggingface's transformers: State-of-the-art natural language processing," *ArXiv*, vol. abs/1910.03771, 2019.

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

[24] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun, Eds., 2015.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, June 2019, pp. 4171–4186, Association for Computational Linguistics.

[26] Kawin Ethayarajh, "How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings," in *EMNLP/IJCNLP*, 2019.

[27] Wei Chu and A. Alwan, "Reducing f0 frame error of f0 tracking algorithms under noisy conditions with an unvoiced/voiced classification frontend," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 3969–3972.

[28] John Kominek, Tanja Schultz, and Alan W. Black, "Synthesizer voice quality of new languages calibrated with mean mel cepstral distortion," in *SLTU*, 2008.

[29] Philipos C. Loizou, *Speech Quality Assessment*, pp. 623–654, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.