PnG BERT: Augmented BERT on Phonemes and Graphemes for Neural TTS

Ye Jia, Heiga Zen, Jonathan Shen, Yu Zhang, Yonghui Wu

Google Research

{jiaye,heigazen,jonathanasdf,ngyuzh,yonghui}@google.com

Abstract

This paper introduces *PnG BERT*, a new encoder model for neural TTS. This model is augmented from the original BERT model, by taking both phoneme and grapheme representations of text as input, as well as the word-level alignment between them. It can be pre-trained on a large text corpus in a self-supervised manner, and fine-tuned in a TTS task. Experimental results show that a neural TTS model using a pre-trained PnG BERT as its encoder yields more natural prosody and more accurate pronunciation than a baseline model using only phoneme input with no pre-training. Subjective side-by-side preference evaluations show that raters have no statistically significant preference between the speech synthesized using a PnG BERT and ground truth recordings from professional speakers.
Index Terms: neural TTS, self-supervised pre-training, BERT

1. Introduction

The advances of neural network-based text-to-speech (TTS) synthesis in the past a few years have closed the gap between synthesized speech and professional human recordings in terms of naturalness [1, 2]. Such neural networks typically consist of an encoder which encodes the input text representation into hidden states, a decoder which decodes spectrogram frames or waveform samples from the hidden states, and an attention or a duration-based upsampler connecting the two together [1–13].

Many early such works take characters of text as input, to demonstrate their capability of being an "end-to-end" model (*i.e.*, text-analysis-free) [1, 3, 8]. More recent works often use phonemes as input, to achieve better stability and generalize better beyond their training sets [2, 9–16]. However, phoneme-based models may suffer from the ambiguity of the representation, such as on homophones. For example, the sentence "*To cancel the payment, press one; or to continue, two.*" is a pattern that used frequently by conversational AI agents for call centers. In the phoneme representation, the trailing "*..., two.*" can be easily confused with "*..., too.*", which is used more frequently in English. However, in natural speech, different prosody is expected at the comma positions in these two patterns – the former expects a moderate pause at the comma, while having no pause sounds more natural for the latter.

Phoneme and grapheme representations have been combined before in TTS [17–20], most commonly by concatenating grapheme-based embeddings to phoneme embeddings [18–20]. However, such approaches face challenges on handling alignment between phonemes and grapheme-based tokens, often require using word-level embeddings (thus a large vocabulary but still with out-of-vocabulary cases) [18, 19], or discarding a portion of tokens when subword embeddings are used [20]. Another approach is to use a multi-source attention [19], attending to both phoneme sequence and grapheme sequence. However, this approach is restricted to attention-based models only, making it inapplicable to duration-based models. More importantly, it may not fully exploit phoneme-grapheme relationships because of the simple architecture of the attention mechanism.

Self-supervised pre-training on large text corpora, using language model (LM) or masked-language model (MLM) objectives, has proven to be successful in natural language processing in the past decade. Such pre-training has been applied to TTS for improving the performance both in low-resource scenarios [19] and high-resource scenarios [20–22], by using embeddings at subword-level [20, 21], word-level [19], or sentence-level [21, 22]. Among the model architectures used for pre-training, BERT is one of the most successful ones, and is often adopted for related works in TTS, such as in [20–23]. However, in these works, the pre-training is only performed on graphemes; no phoneme-grapheme relationship is learned during the pre-training.

This paper introduces *PnG BERT*, an augmented BERT model that can be used as a drop-in replacement for the encoder in typical neural TTS models, including attention-based and duration-based ones. PnG BERT can benefit neural TTS models by taking advantages of both phoneme and grapheme representation, as well as by self-supervised pre-training on large text corpora to better understand natural language in its input. Experimental results show that the use of a pre-trained PnG BERT model can significantly improve the naturalness of synthesized speech, especially in terms of prosody and pronunciations. Subjective side-by-side preference evaluations show that raters have no statistically significant preference between the speech synthesized using PnG BERT and ground truth recordings from professional speakers. Audio samples are available online[1].

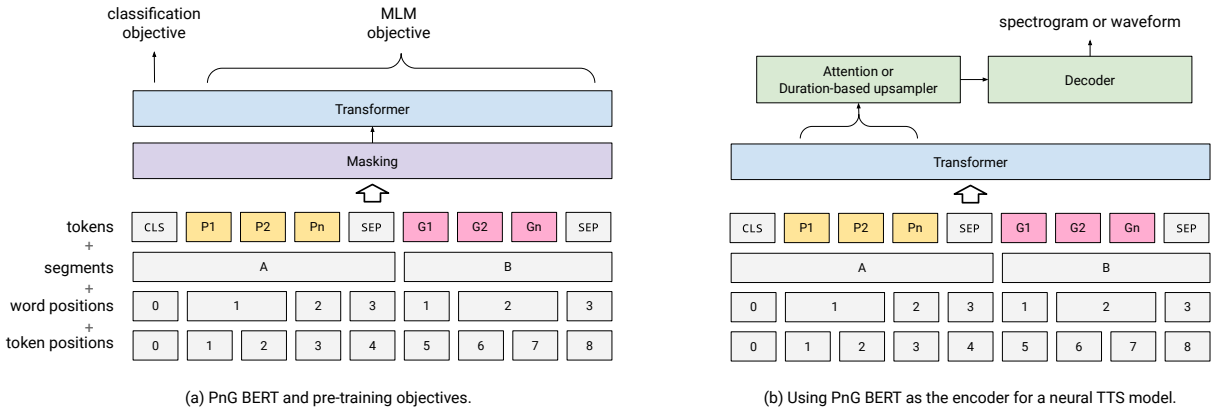
2. PnG BERT

The PnG BERT model is illustrated in Figure 1. It takes both phonemes and graphemes as input, and can be used directly as an input encoder in a typical neural TTS model. It follows the original BERT architecture [24], except for differences in the input and the output, and pre-training and fine-tuning procedures. It also bears some similarity to XLM [25], a cross-lingual language model.

2.1. Background: BERT

The original BERT model [24] is essentially a Transformer encoder [26], pre-trained using a masked language model (MLM) objective on a text corpus. It can take inputs composed of multiple sentences, identified with a segment ID. A special token `CLS` is prepended to the first segment, which can be used in an extra classification objective in addition to the MLM objective. Another special token `SEP` is appended to each segment, indicating segment boundaries. The input to BERT is represented as the sum of a token embedding, a segment embedding and a position embedding.

[1] https://google.github.io/tacotron/publications/png_bert/



(a) PnG BERT and pre-training objectives.

(b) Using PnG BERT as the encoder for a neural TTS model.

Figure 1: *The pre-training and fine-tuning of PnG BERT for neural TTS. Phonemes are displayed in yellow, graphemes in pink.*

2.2. Input representation

The input to PnG BERT is composed of two segments (i.e. subsequences), which are two different representations of the same content. The first segment is a phoneme sequence, containing individual IPA phonemes as tokens. The second segment is a grapheme sequence, containing subword units from text as tokens[2]. The special tokens `CLS` and `SEP` are added in the same way as in the original BERT. All tokens share the same ID space for embedding lookup and MLM classification.

In addition to the token, segment and position embeddings the original BERT uses, a fourth word-position embedding is used by PnG BERT, providing word-level alignment between phonemes and graphemes. This embedding is implemented with the same sinusoidal functions as the regular position embedding, but using the index of the word each phoneme or grapheme belongs to as the input position. An additional learned linear projection is applied to the sinusoidal signals in order to avoid confusion with the regular position embedding. All four embeddings are summed together.

Figure 1 shows a visualized example of such construction of the input.

2.3. Pre-training

Similar to the original BERT model, PnG BERT can be pre-trained on a plain text corpus in a self-supervised manner. Phonemes are obtained using an external grapheme-to-phoneme (G2P) conversion system, while graphemes can be characters, bytes, or obtained using a subword text tokenizer, such as WordPiece [27], SentencePiece [28], or byte-pair encoding (BPE) [29]. Only the MLM objective is used during pre-training.

2.3.1. Masking strategy

The input to PnG BERT consists of two distinct sequences representing essentially the same content. If random masking was applied as in the original BERT model, the counterpart of a token masked in one sequence could present in the other sequence. This would make the MLM prediction significantly easier and would reduce the effectiveness of the pre-training. To avoid this issue, we apply random masking at word-level, consistently masking out phoneme and grapheme tokens belonging to the same word.

We use the same masking categories and ratios used during the original BERT pre-training process: both the phonemes and graphemes for 12% random words are replaced by `MSK` tokens; both the phonemes and graphemes for another 1.5% random words are replaced by random phonemes and graphemes, respectively; both the phonemes and graphemes for another 1.5% random words are kept unchanged. The MLM loss is computed only on tokens corresponding to these 15% words.

Alternative strategies may be used. In Section 3.1 we experimented with increasing the masking ratios of the original random masking. Another choice is to apply masking in phoneme-to-grapheme (P2G) and G2P-like manners, *i.e.*, masking out all tokens in one segment and keeping all tokens in the other.

2.4. Fine-tuning

The PnG BERT model can be used as an input encoder for typical neural TTS models. Its weights can be initialized from a pre-trained model, and further fine-tuned during TTS training. We freeze the weights of the embeddings and the lower Transformer layers, and only fine-tune the higher layers, to prevent degradation due to the smaller TTS training set and to help the generalization ability of the trained TTS model. The MLM objective is not used during TTS fine-tuning.

Only the hidden states from the final Transformer layer on the phoneme token positions are passed to the downstream TTS components (e.g., the attention or the duration-based upsampler), as illustrated in Figure 1. Despite these hidden states are only from the phoneme positions, they can carry information from the graphemes because of the self-attention machanism in the PnG BERT model itself.

In the experiments in the next section, we use a NAT TTS model [2], and replace the original RNN-based encoder with PnG BERT. We anticipate that the PnG BERT model can be applied to other neural TTS model architectures, too.

3. Experiments

To evaluate the performance of PnG BERT, we conducted experiments on a monolingual multi-speaker TTS task. PnG BERT is used to replace the original RNN-based encoder in NAT [2], a duration-based neural TTS model. It produces mel-spectrograms which are converted into the final waveforms

[2]The terms "grapheme" and "subword unit" are considered to be equivalent in the current context, and are used interchangeably in the remainder of this paper.

Table 1: *Pre-training performance of PnG BERT, measured by token prediction accuracy using 3 different evaluation-time masking. (MLM: same random masking as training time; G2P / P2G: masking out all phonemes / graphemes, respectively.)*

Model	MLM	G2P	P2G
PnG BERT	67.9	50.9	33.2
w/o consistent masking	96.8	99.1	95.2
w/o word alignment	94.9	41.3	39.5

using a WaveRNN-based neural vocoder [30].

In all the experiments, the PnG BERT model used 6 Transformer layers with hidden size 512 and 8 attention heads, pre-trained using the SM3 optimizer [31] with a batch size of 24K for 1M steps and a maximum sequence length of 480 tokens. The remainder of the TTS models used the same architecture and hyperparameters as in [2], except that we trained them with a larger batch size of 512 for 450K steps (same for the standard NAT model used as a baseline). The top 2 Transformer layers in PnG BERT were fine-tuned, with the rest part frozen.

The performance of the TTS models were evaluated through subjective listening tests on the naturalness, including 5-scale mean opinion score (MOS) tests and 7-scale (-3 to 3) side-by-side (SxS) preference tests. In the SxS preference tests, loudness normalization[3] is applied consistently on the all samples in both side. All subjective tests were conducted using at least 1,000 samples. Each rater was allowed to rate no more than 6 samples in one test.

3.1. Pre-training performance

We pre-trained the PnG BERT model on a plain text corpus mined from Wikipedia, containing 131M English sentences. A proprietary text normalization engine [32] was used to convert the text into the corresponding phoneme sequences. A SentencePiece model [28] with a vocabulary of 8,192 tokens was used for tokenizing text into subwords (*i.e.*, graphemes).

We compared multiple pre-training strategies for PnG BERT (Sec. 2.3.1). For word-level consistent masking, the same masking ratios (15% in total) as the original BERT were used; otherwise, we doubled the masking ratios (to 30% in total) since the objective was easier. As Table 1 shows, the word-level consistent masking makes the MLM prediction significantly harder than simply doubling the random masking ratios; it also makes the trained model perform worse in the G2P and P2G evaluations. Providing word-level alignment between phonemes and graphemes (*i.e.*, word position embedding) significantly helped the MLM prediction and brought G2P and P2G accuracy to a very high level. However, as shown in the following subsections, these metrics did not foretell the performance on the downstream TTS tasks.

Figure 2 shows plots of the self-attention probabilities across PnG BERT layers, without providing word-level alignment. The trident-shaped distribution in the higher layers indicates that the model learns the alignment between phonemes and graphemes, and effectively combines information from both. Such trident-shaped distribution emerges starting from the bottom layer when word-level alignment is provided, suggesting that the PnG BERT learns more effectively in such case, which is consistent with the other experiment results.

[3]Loudness normalization is critical in SxS preference tests, as raters tend to be biased towards louder samples.

Table 2: *Performance of the TTS models, measured with 10 speakers (5 male and 5 female) on 3 evaluation sets.*

Model	MOS	SxS (vs Baseline)	
	Generic lines	Hard lines	Questions
NAT (Baseline)	4.41 ± 0.05	-	-
NAT w/ PnG BERT	4.47 ± 0.05	0.28 ± 0.05	0.15 ± 0.06
w/o consistent masking	4.44 ± 0.05	0.14 ± 0.06	0.10 ± 0.06
w/o word alignment	4.45 ± 0.05	0.15 ± 0.06	0.12 ± 0.06
w/o pre-training	4.30 ± 0.06	0.12 ± 0.05	0.06 ± 0.05

Table 3: *Performance of the TTS models on recordings held-out from training, measured with the same 10 speakers (5 male and 5 female) as in Table 2, each with 100 samples.*

Model	MOS	SxS (vs Ground truth)
NAT (Baseline)	4.45 ± 0.05	-0.11 ± 0.09
NAT w/ PnG BERT	4.47 ± 0.05	0.02 ± 0.10
Ground truth	4.47 ± 0.05	-

3.2. TTS performance

We trained multi-speaker TTS models on a proprietary dataset consisting of 243 hours of American English speech, recorded by 31 professional speakers, downsampled to 24 kHz. The amount of recording per speaker varies from 3 to 47 hours.

3.2.1. Ablation studies

We conducted ablation studies using three evaluation text sets slightly modified from [20]: a "generic lines" set with 1,000 generic lines (the same set as in [2]), including long inputs (corresponding to up to more than 20 seconds of audio); a "hard lines" set with 286 lines[4], containing, e.g., titles and long noun compounds, expected to be hard; and a "questions" set with 300 questions, as questions are prosodically different from statements. These lines were synthesized using 10 speakers (5 male and 5 female), in a round-robin fashion for the generic lines set (resulting in 1,000 utterances), and a cross-join fashion for the hard lines and questions sets (resulting in 2,860 and 3,000 utterances, respectively).

The results are reported in Table 2. It can be seen that using pre-trained PnG BERT significantly improved the performance of NAT. The best result was achieved by using word-level consistent masking, providing the word-level phoneme-grapheme alignment, and pre-training on a large text corpus. When PnG BERT was not pre-trained, the TTS model performed worse than the baseline on the generic line set. Further check revealed that it was due to the trained model not generalizing well to inputs longer than what was seen during training, which is a known limitation of Transformer [33, 34]. Nevertheless, it outperformed the baseline on shorter evaluation sets, possibly from the benefit of using both phoneme and grapheme as input.

It is interesting to note that the PnG BERT model with the highest G2P and P2G accuracy during pre-training performed worse than the ones with significantly lower G2P and P2G accuracy. This could be because it had the easiest objective during pre-training (reflected by the highest MLM accuracy) and,

[4]Available at https://google.github.io/chive-prosody/chive-bert/dataset

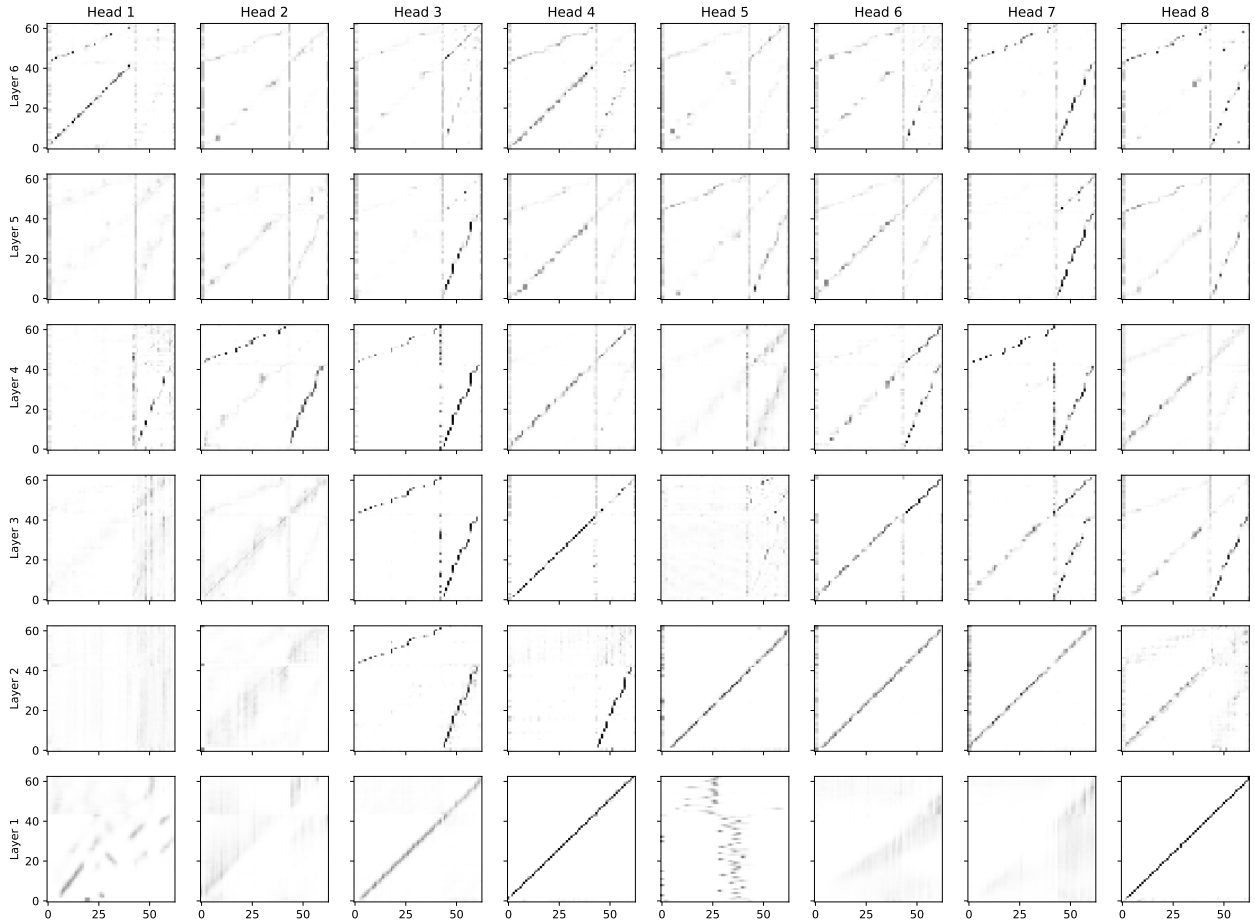


Figure 2: *Self-attention probabilities of each layer and head in a PnG BERT without using word-level alignment. The input text is "To cancel the payment, press one; or to continue, two.", which is converted into an input sequence composed of a first segment of 42 phonemes and 2 special tokens, and a second segment of 18 graphemes and 1 special token. The trident-shaped distributions in the higher layers indicate that the model learns the alignment between phonemes and graphemes, and encodes information across the two segments. When word-level alignment is provided, such trident-shaped distributions emerge starting from the bottom layer.*

as a result, did not learn as much as the models with other pre-training strategies did. This finding suggests that the benefit of PnG BERT primarily lies in better natural language understanding, rather than potential improvements on G2P conversion.

3.2.2. Subjective rater comments

In the SxS tests, on positive examples, raters comments often mentioned better "prosody", "tone", "stress", "inflection", or "pronunciation"; on negative examples, the comments often mentioned worse "inflection", "stress", or "unnatural distortion". These comments are consistent with the foregoing analysis, confirming that the improvement are primarily from better natural language understanding and thus improved prosody in the synthesized speech.

3.2.3. Comparison to ground truth recordings

Lastly, we conducted subjective SxS preference tests against ground truth recordings held out from training. We randomly sampled 100 utterances of each of the 10 speakers, and compared the synthesized audios with the ground truth recordings. The results in Table 3 show that raters had no statistically significant preference between the ground truth recordings from professional speakers and the speech synthesized using PnG BERT.

4. Conclusions

We proposed *PnG BERT*, an augmented BERT model that takes both phoneme and grapheme representations of text as its input. We also described a strategy for effectively pre-training it on a large text corpus in a self-supervised manner. PnG BERT can be directly used as an input encoder for typical neural TTS models. Experimental results showed that PnG BERT can significantly improve the performance of NAT, a state-of-the-art neural TTS model, by producing more natural prosody and more accurate pronunciation. Subjective side-by-side preference evaluation showed that raters had no statistically significant preference between the speech synthesized using PnG BERT and the ground truth recordings from professional speakers.

5. Acknowledgements

The authors would like to thank the Google TTS Research team, in particular Tom Kenter for his support on the evaluation and comprehensive suggestions on the writing.

6. References

[1] 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. Agiomyrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions," in *Proc. ICASSP*, 2018.

[2] J. Shen, Y. Jia, M. Chrzanowski, Y. Zhang, I. Elias, H. Zen, and Y. Wu, "Non-Attentive Tacotron: Robust and controllable neural TTS synthesis including unsupervised duration modeling," *arXiv preprint arXiv:2010.04301*, 2020.

[3] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. C. Courville, and Y. Bengio, "Char2Wav: End-to-end speech synthesis," in *ICLR workshop*, 2017.

[4] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[5] S. Ö. Arık, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman *et al.*, "Deep Voice: Real-time neural text-to-speech," in *Proc. ICML*, 2017.

[6] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, "Deep Voice 2: Multi-speaker neural text-to-speech," in *Proc. NeurIPS*, 2017.

[7] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep Voice 3: Scaling text-to-speech with convolutional sequence learning," in *Proc. ICLR*, 2018.

[8] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech*, 2017.

[9] I. Elias, H. Zen, J. Shen, Y. Zhang, Y. Jia, R. Weiss, and Y. Wu, "Parallel Tacotron: Non-autoregressive and controllable TTS," in *Proc. ICASSP*, 2021.

[10] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with Transformer network," in *Proc. AAAI*, 2019.

[11] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech: Fast, robust and controllable text to speech," in *Proc. NeurIPS*, 2019.

[12] Y. Ren, C. Hu, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech 2: Fast and high-quality end-to-end text-to-speech," in *Proc. ICLR*, 2021.

[13] R. J. Weiss, R. Skerry-Ryan, E. Battenberg, S. Mariooryad, and D. P. Kingma, "Wave-Tacotron: Spectrogram-free end-to-end text-to-speech synthesis," in *Proc. ICASSP*, 2021.

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

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

[16] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. Lopez-Moreno, and Y. Wu, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Proc. NeurIPS*, 2018.

[17] K. Kastner, J. F. Santos, Y. Bengio, and A. Courville, "Representation Mixing for TTS Synthesis," in *Proc. ICASSP*, 2019.

[18] H. Ming, L. He, H. Guo, and F. K. Soong, "Feature reinforcement with word embedding and parsing information in neural TTS," *arXiv preprint arXiv:1901.00707*, 2019.

[19] Y.-A. Chung, Y. Wang, W.-N. Hsu, Y. Zhang, and R. Skerry-Ryan, "Semi-supervised training for improving data efficiency in end-to-end speech synthesis," in *Proc. ICASSP*, 2019.

[20] T. Kenter, M. K. Sharma, and R. Clark, "Improving prosody of RNN-based english text-to-speech synthesis by incorporating a BERT model," in *Proc. Interspeech*, 2020.

[21] T. Hayashi, S. Watanabe, T. Toda, K. Takeda, S. Toshniwal, and K. Livescu, "Pre-trained text embeddings for enhanced text-to-speech synthesis." in *Proc. Interspeech*, 2019.

[22] G. Xu, W. Song, Z. Zhang, C. Zhang, X. He, and B. Zhou, "Improving prosody modelling with cross-utterance BERT embeddings for end-to-end speech synthesis," in *Proc. ICASSP*, 2021.

[23] Y. Zhang, L. Deng, and Y. Wang, "Unified Mandarin TTS front-end based on distilled BERT model," *arXiv preprint arXiv:2012.15404*, 2020.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, 2019.

[25] G. Lample and A. Conneau, "Cross-lingual language model pretraining," in *Proc. NeurIPS*, 2019.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NeurIPS*, 2017.

[27] M. Schuster and K. Nakajima, "Japanese and Korean voice search," in *Proc. ICASSP*, 2012.

[28] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proc. EMNLP*, 2018.

[29] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. ACL*, 2016.

[30] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *Proc. ICML*, 2018.

[31] R. Anil, V. Gupta, T. Koren, and Y. Singer, "Memory-efficient adaptive optimization," in *Proc. NeurIPS*, 2019.

[32] P. Ebden and R. Sproat, "The Kestrel TTS text normalization system," *Natural Language Engineering*, vol. 21, no. 3, p. 333, 2015.

[33] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser, "Universal Transformers," in *Proc. ICLR*, 2019.

[34] M. Hahn, "Theoretical limitations of self-attention in neural sequence models," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 156–171, 2020.