CONTEXT-AWARE MASK PREDICTION NETWORK FOR END-TO-END TEXT-BASED SPEECH EDITING

Tao Wang[1,2], *Jiangyan Yi*[1], *Liqun Deng*[4], *Ruibo Fu*[1], *Jianhua Tao*[1,2,3], *Zhengqi Wen*[1]

[1]NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3]CAS Center for Excellence in Brain Science and Intelligence Technology, Beijing, China
[4]Huawei Noah's Ark Lab, Shenzhen, China

ABSTRACT

The text-based speech editor allows the editing of speech through intuitive cutting, copying, and pasting operations to speed up the process of editing speech. However, the major drawback of current systems is that edited speech often sounds unnatural and it is not obvious how to synthesize records according to a new word not appearing in the transcript. This paper proposes a novel end-to-end text-based speech editing method called context-aware mask prediction network (CampNet), which avoids the unnatural phenomenon caused by cut-copy-paste operation in the traditional method and can synthesize a new word not appearing in the transcript. Besides, three text-based speech editing operations based on CampNet are designed: deletion, replacement, and insertion. These operations can comprehensively cover different kinds of situations that text-based speech editing can face. The subjective and objective experiments on VCTK and LibriTTS data sets show that the speech editing results based on CampNet are better than TTS technology, manual editing, and VoCo method (the combination of speech synthesis and speech conversion). We also conducted detailed ablation experiments to explore the effect of the CampNet structure on its performance. Examples of generated speech can be found at https://hairuo55.github.io/CampNet-demo.

Index Terms— text-based speech editing, speech synthesis, end-to-end model, mask and prediction

1. INTRODUCTION

With the development of internet, the transmission of information has been accelerated, such as movies, podcasts, YouTube videos, etc. The production of these media is inseparable from speech editing. Typical speech editing interfaces [1] present a visualization of the speech such as waveform and/or spectrogram and provide the user with standard select, copy, paste, volume adjustment, etc. Such tools provide a great convenience for media producers [2]. Some state-of-the-art systems allow the editor to perform select, cut, and paste operations in the text transcript of the speech and apply the changes to the waveform accordingly, which is called text-based speech editing [3]. It mainly faces two challenges.

One is that the edited speech often sounds unnatural because the edited region does not match the prosody of speech context [4]. To solve this problem, speech manipulation includes fast pitch-shifting and time-stretching techniques is applied. They are efficient and suitable for real-time interactive applications but will produce audible artifact [5, 6]. Neural vocoders such as WaveNet [7], etc. [8, 9] can obtain higher perceptual quality, but can not perform context-aware generation for text-based speech editing. To achieve context-aware generation, context-aware prosody correction [4] is applied to modify the prosodic information of the target segment. This method combines neural network and digital signal processing method, which can effectively improve the speech quality. However, an obvious limitation of this system is that the words to insert or replace may not be found in the available speech data of the target speaker, which limits the application of text-based speech editing.

The second problem is that the ability to synthesize new words that do not appear in the transcript is not supported. With the development of TTS, the task of text-based speech editing can be completed with the help of TTS [10], such as the Tacotron [11] and WaveNet [7]. Some transfer learning works can generate the speech of the target speaker, such as global style token (GST) [12], etc [13–15]. However, it is inconvenient to edit the specific words in the synthesis speech. To achieve text-based speech editing, the previous work was completed with the help of TTS and voice conversion (VC) system [16], which is called VoCo [3, 17]. This cascade system can ensure the stability and feasibility of the whole process. However, because each module is independent of the other, it will accumulate errors and bring difficulties to the construction of the system. Besides, we still need to synthesize the edited region first and then copy and paste it, which will face mismatched prosodic in the edited speech.

Different from solving the above two problems separately, this paper proposes an end-to-end text-based speech editing method, which can avoid the above two problems at the same time. Firstly, the context-aware mask prediction network (CampNet) is proposed to simulate the process of text-based speech editing. Secondly, three text-based speech editing operations based on CampNet are designed: deletion, replacement, and insertion. Overall, the main contributions of this paper are:

- CampNet is designed for the text-based speech editing task (Sec. 2.1), which can avoid the unnatural phenomenon caused by cut-copy-paste operation in the traditional method and can synthesize a new word not appearing in the transcript. To our best knowledge, CampNet is the first text-based speech editing model that can be trained in end-to-end form.
- Based on CampNet, we design three speech editing operations, corresponding to delete, replace and insert operations, respectively (Sec. 2.2). These operations can comprehensively cover different kinds of situations that text-based speech editing can face.

The rest of the paper is organized as follows. Section 2 describes the methods. Experiments and results are analyzed in Section 3. The conclusions are discussed in Section 4.

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

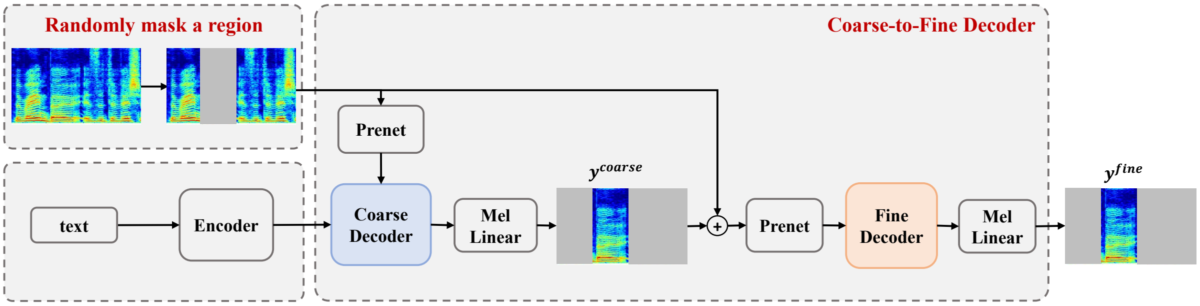

Fig. 1. An overview of CampNet. The acoustic model is based on the encoder and decoder model, and LPCNet is used as the vocoder.

2. PROPOSED METHOD

The text-based speech editing can be view as two steps. We take the replacement operation as an example. First, we mask part of the original speech that needs to be edited. Then the masked region is predicted according to the modified transcription and speech context. In fact, some other operation of text-based speech editing, such as deletion and insertion can also be viewed as the process of masking and prediction, which will be described in detail in Sec 2.2. The advantage of this view is that text-based speech editing can be described by an end-to-end model, which can directly generate edited speech according to the context, synthesize any content speech, and also ensure natural prosody.

In the section, we will first introduce the proposed end-to-end context-aware mask prediction network (CampNet). Second, we will show how to use CampNet for text-based speech editing tasks, including delete, replace and insert operations.

2.1. Context-Aware Mask Prediction Network (CampNet)

The task of end-to-end text-based speech editing model can be view as modifying part of original speech to match the edited transcription. Assume that the original acoustic features is $y = (y_1, \dots, y_n, \dots, y_m, \dots, y_T)$ and its transcription is $x = (x_1, \dots, x_a, \dots, x_b, \dots, x_M)$, where $(x_a, \dots, x_b)$ is aligned with acoustic features $(y_n, \dots, y_m)$. When $(x_a, \cdots, x_b)$ is edited and the new transcription x' is $(x_1, \dots, x'_a, \dots, x'_{b'}, \dots, x_M)$, the target acoustic feature is $y' = (y_1, \dots, y'_n, \dots, y'_m, \dots, y_T)$. We assume that the length of the new speech in the editing region is consistent with that of the original region. If it is inconsistent at the inference stage, an additional duration model [18] can be used to predict the duration of edited words. Then we can add or delete some fragments on the original part to achieve consistent length. Since $(y'_n, \dots, y'_m)$ is independent of text sequences x and related to text x', the problem of text-based speech editing can be formulated in terms of estimating the condition probability $P(y'|y, x'; \theta)$ of the target acoustic feature, and θ is corresponding model parameters, where

$$P(y'|y, x'; \theta) = P(y'_n, \dots, y'_m|y, x'; \theta) \tag{1}$$

In addition, since $(y'_n, \dots, y'_m)$ and $(y_n, \dots, y_m)$ only have the same position, but their contents are different. To remain the position information of edited region $(y'_n, \dots, y'_m)$ accurately, and not be interfered by the content information of $(y_n, \dots, y_m)$ in the original speech, $(y_n, \dots, y_m)$ can be replaced with a new token $< mask >$, and the masked source acoustic features y_{mask} can be expressed as:

$$y_{mask} = (y_1, \dots, < mask >, \dots, < mask >, \dots, y_T) \tag{2}$$

Where the n position in y_{mask} is the starting point of the mask, and m is the ending point of the mask.

Then the condition probability $P(y'|y, x'; \theta)$ can be formulated as:

$$P(y'|y, x'; \theta) = P(y_{n'}, \dots, y_{m'}|y_{mask}, x'; \theta) \tag{3}$$

From the Eq. 3, the task of text-based speech editing can be decomposed into the following two process. First, mask the region of the original speech y that needs to be edited, and get the masked acoustic features y_{mask}. Then, combined with the masked acoustic feature y_{mask} and the edited text sequence x', neural network is used to predict the edited region $(y'_n, \dots, y'_m)$. Because x' and y_{mask} have different lengths, and it has been proved that transformer can effectively fuse the context information of sequences with different lengths [19], an encoder-decoder framework based on transformer is adopt as the structure of CampNet, which is shown in Fig. 1.

The CampNet consists of two processing stages: encoder and decoder. First, the encoder module processes the input sentence and converts it into a hidden representation. This representation is used to guide the decoder to predict the acoustic feature of the edited speech. Second, a random region of acoustic features is masked as the ground truth to condition the decoder at the decoding stage. The decoder is divided into two steps. The first step is to learn the alignment information between the masked ground truth and the text representation through the multi-head attention mechanism and predict coarse acoustic features. Then, the second step is to predict finer acoustic features based on the coarse acoustic features and original speech context by a fine decoder, which can further fuse the context information of speech to make the predicted speech more natural. We call the process of masking part of the acoustic features and predicting the mask region as the ”context-aware mask prediction”.

2.2. Text-based Speech Editing Operations based on CampNet

Some operations of speech editing, such as deletion, insertion, and replacement, can be carried out based on a pre-trained CampNet model. These operations are shown in Fig. 2 and we will introduce these in detail in this section,.

2.2.1. Delete operation

The deletion operation allows the user to remove a region of speech waveform that corresponds to certain specified words. We divide the process into three steps. The first step is to manually delete the target region and the corresponding words in the text. Due to manual deletion, unnatural phenomena will appear at the connection, such as fundamental frequency discontinuity. To repair this problem, taking the connection point as the center and masking the left and right fragments of speech in a small range. Finally, we input the masked speech and the text after deleting the target word into CampNet to re-predict the masked region, obtaining a more natural connection.

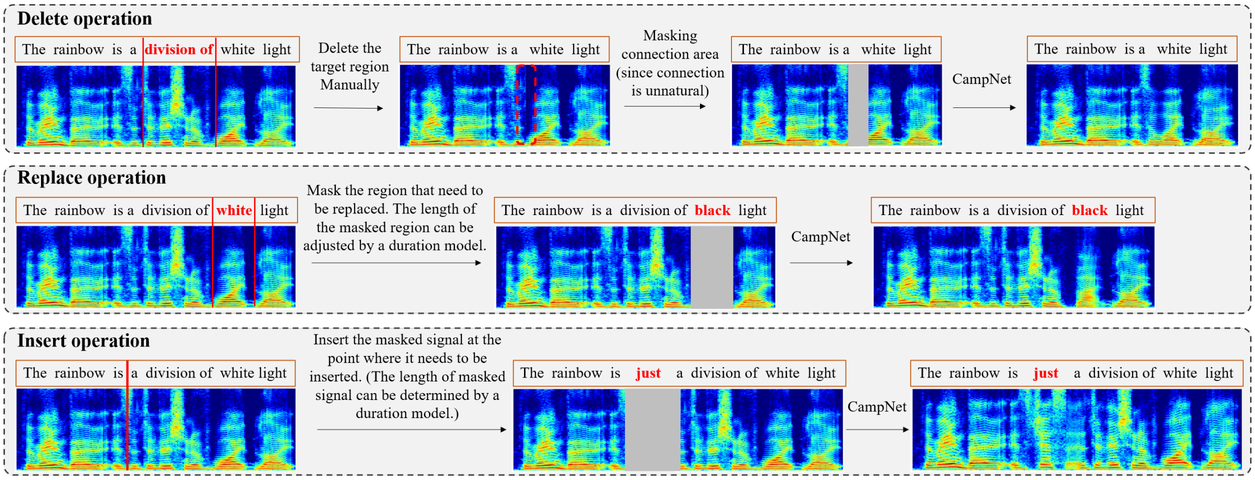
Delete operation
The rainbow is a division of white light
Delete the target region Manually
The rainbow is a white light
Masking connection area (since connection is unnatural)
The rainbow is a white light
CampNet
The rainbow is a white light
Replace operation
The rainbow is a division of white light
Mask the region that need to be replaced. The length of the masked region can be adjusted by a duration model.
The rainbow is a division of black light
CampNet
The rainbow is a division of black light
Insert operation
The rainbow is a division of white light
Insert the masked signal at the point where it needs to be inserted. (The length of masked signal can be determined by a duration model.)
The rainbow is just a division of white light
CampNet
The rainbow is just a division of white light

Fig. 2. At the inference stage, three different operations of text-based speech editing based on CampNet are proposed.

2.2.2. *Replace operation*

The replace operation allows the user to replace a section of speech with another speech. This operation can be divided into two cases. One is that the length of the replaced segment is close to the target pronunciation. The other is that there is a large gap between them. For the former, it can be divided into two steps. The first step is to define the word boundary to be replaced, mask it according to the word boundary and then modify the text. It is worth noting that the range of masking can be larger than the actual boundary when masking. In this way, the model can learn more natural connections. The second step is to input the masked speech and the modified text into CampNet. The model will predict the replaced speech according to the modified text.

If there is a big difference between the length of the replaced speech and the original speech, such as adding some words or deleting some words, a pre-training duration model can be used to predict the length of the replaced region. The duration model is widely used in traditional TTS task [18]. Here, we use the duration model to obtain the duration of the replaced word. Then according to the predicted duration, the masked region can be added or deleted some fragment to ensure the consistency of the duration.

2.2.3. *Insert operation*

The insert operation allows the user to insert a speech into the edited speech. Firstly, we can use a pre-trained duration model to predict the duration of the words to be inserted. Then insert the masked signal with the predicted duration into the original speech. Then input the modified text and speech into CampNet, and predict the inserted speech.

3. EXPERIMENTS AND RESULTS

3.1. Tasks and Datasets

We compare the replacement operation of CampNet and other systems, which is easy to evaluate with the original speech[1]. We conduct experiments on VCTK [20] corpus and LibriTTS [21] corpus. Specifically, we select four speakers from the VCTK dataset as the test set, and the rest utterances are divided into 90% training set and 10% validation set. We use the training set to train the models. We also randomly select 100 sentences from the LibriTTS corpus as the test set to verify the model's performance on the cross dataset. To ensure that the synthesized speech of different systems is consistent with the content of the original speech and facilitate the comparison between objective metrics and subjective metrics, we randomly choose 80 words that span 3 to 10 phonemes from 80 different sentences for each test set. For each sentence, we remove the region of the corresponding words in the speech and then use different systems to predict the removed region. All wav files are sampled at 16KHz.

3.2. Model Details

Acoustic features are extracted with a 10 ms window shift. LPCNet [9] is utilized to extract 32-dimensional acoustic features. Five systems are compared in our experiments.

- **Synth** We train a neural TTS system to synthesize the speech and copy the target region to insert into the edited speech. To make the voice of the synthesized speech as similar as that of the target speaker as possible, Tacotron2 based on global style token (GST [12]) is used as the acoustic model. The structure of Tacotron2 is the same as that in paper [22].
- **VoCo** We train the neural TTS and VC system to realize the VoCo system. For the TTS system, the configuration is the same as **Synth**. We select two speakers (one male and one female) from training set as source speakers. Then, the TTS system is used to synthesize speech, and the VC system synthesizes the voice similar to the target speaker. We copy the target region from the output speech and insert it into the edited speech. The VC system is based on phonetic posteriorgrams (PPGs) [16]. Furthermore, to make the VC system have the one-shot ability, we use GST as speaker embedding to train a multi-speaker VC model.
- **Edit** We use the editing interface to refine the speech further if it improves on Synth/VoCo.
- **Real** The actual recording without modification.
- **CampNet** The structures of encoder, coarse decoder and fine decoder is based on transformer [23]. We input the phoneme sequence into a 3-layer CNN [24] to model the longer-term context in the input character sequence. Each phoneme has a trainable embedding of 256 dims, and the output of each convolution layer has 256 channels, followed by batch normalization and ReLU activation and a dropout layer as well [25–27]. The transformer blocks of the encoder and fine decoder are 3. The transformer block of the coarse decoder is 6. The hidden dimension of the transformer is 256. At the training stage, we set the masked region to be 12% of the total speech length.

[1] Examples of more operations and coarse-to-fine decoding can be found at https://hairuo55.github.io/CampNet-demo.

Table 1. OBJECTIVE EVALUATION RESULTS OF DIFFERENT MASK RATIO AT INFERENCE STAGE ON THE VCTK TEST SET

	VCTK						LibriTTS					
Metrics	M-6%	M-8%	M-10%	M-12%	M-14%	M-16%	M-6%	M-8%	M-10%	M-12%	M-14%	M-16%
MCD(dB)	0.465	0.387	0.391	**0.380**	0.383	0.398	0.746	0.661	0.631	**0.628**	0.634	0.650
F0-RMSE(dB)	10.511	9.723	9.407	**8.637**	9.255	9.114	22.895	22.242	21.049	**20.201**	21.086	21.820
V/UV error	1.989	1.658	1.610	1.635	**1.492**	1.750	5.679	4.136	**3.635**	3.675	4.000	4.259
F0-CORR	0.971	0.976	0.977	**0.981**	0.978	0.978	0.952	0.943	0.949	**0.954**	0.948	0.945

Table 2. OBJECTIVE EVALUATION RESULTS ON THE TEST SETS OF VCTK AND LIBRITTS

	Metrics	**Synth**	**VoCo**	**Edit**	**CampNet**
VCTK	MCD(dB)	0.594	0.589	0.582	**0.380**
	F0-RMSE(Hz)	10.463	10.555	10.451	**8.637**
	V/UV error	1.944	1.843	1.937	**1.635**
	F0-CORR	0.973	0.972	0.975	**0.981**
LibriTTS	MCD(dB)	0.871	0.894	0.870	**0.628**
	F0-RMSE(Hz)	21.898	2.093	21.308	**20.201**
	V/UV error	3.916	4.347	3.956	**3.675**
	F0-CORR	0.940	0.945	0.939	**0.954**

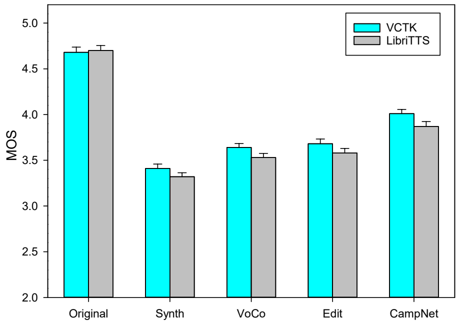

Fig. 3. MOS score of each system in the two datasets.

3.3. Comparison between CampNet and Some other Methods

We compare the performance of our proposed **CampNet** with three speech editing methods by objective and subjective evaluations.

First, F0 RMSE (root of mean square errors of F0), MCD (Mel-cepstrum distortion), VUV (the error rate of voiced/unvoiced flags), and F0 CORR(correlation factor of F0) were adopted as metrics and were calculated on the test sets of VCTK and LibriTTS. The objective results are listed in Table 2. In general, it can be found that the metrics on the two test sets of CampNet are the best among all the systems. Specifically, in the frequency domain, the CampNet obtained the lowest MCD, which means that human perception would be better. Besides, the F0 has a significant influence on speech perception. We can find that CampNet achieves the best performance in F0-related metrics (F0-RMSE, V/UV error, and F0-CORR). The results show that CampNet can obtain more accurate fundamental frequency information.

Second, subjective evaluations are conducted to compare the performance of CampNet with other systems in terms of the naturalness of edited speech. In this evaluation, twenty utterances in each test set are selected and edited using the proposed method and other systems, including Synth, VoCo, and Edit. Twenty listeners took part in the evaluation. They are told in advance which word is predicted. The listeners were asked to listen and rate the quality of the restored sentence on a Likert scale. They can play the recording multiple times. Fig. 3 shows the MOS score of each system. The results show that the CampNet is better than the other three systems in each test set. This is also consistent with the previous analysis of objective metrics.

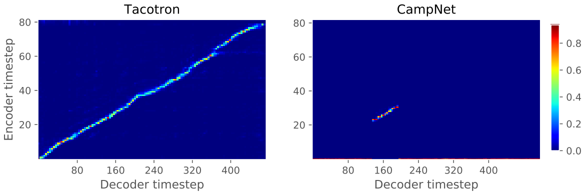

Fig. 4. Comparison of Tacotron and CampNet alignment. The alignment of Tacotron is to align with the text in a complete time step. CampNet only aligns the edited region with the edited text.

3.4. Comparison of Alignment with Tacotron

To understand the process of speech editing better, we visualize the alignments of text and speech in the multi-head attention of coarse decoder. The alignment of local sensitive attention in Tacotron is also visualized for comparison, as shown in Fig. 4. We can find that the alignment of Tacotron is aligned in the whole time step, where each column denotes the attention probabilities corresponding to different encoder states for one decoder step. On the contrary, the alignment of CampNet is only in a small region, that is, the edited region. Aligning only a small region can help the model pay more attention to the edited region.

3.5. The Effect with Different Mask Ratio

We explore the influence of different mask ratios during training. Specifically, the mask ratios at the training stage are set to 6%, 8%, 10%, 12%, 14%, and 16%, respectively. The trained models are represented by M-6%, M-8%, M-10%, M-12%, M-14%, and M-16%. All models are trained for 2 million steps with the same structure and hyper-parameters. We calculate the objective metrics of each model on the test set, which is shown in Table 1. It can be found that when the mask ratio is set to 12%, the model has the best effect on most indicators (MCD, F0-RMSE, and F0-CORR) in the both test sets.

4. CONCLUSION

This paper has proposed a context-aware mask prediction network for the end-to-end text-based speech editing task, which can delete, replace and insert the speech at the word level by editing the transcription. To simulate the speech editing process at the training stage, the text-based speech editing task is viewed as a two-stage process: masking and prediction, and a coarse-to-fine decoding method is proposed to achieve context-aware prediction. At the inference stage, three operations are designed based on CampNet, corresponding to the deletion, insertion, and replacement operations. The experimental results demonstrate that the CampNet is better than the TTS, VoCo, and manual editing in subjective and objective evaluation in the text-based speech editing.

5. ACKNOWLEDGMENT

This work is supported by the National Key Research and Development Plan of China (No.2020AAA0140003), the National Natural Science Foundation of China (NSFC) (No.62101553,No.61901473, No.61831022), the Key Research Project (No.2019KD0AD01) , and is also partially funded by Huawei Noah's Ark Lab.

6. REFERENCES

[1] Roger Derry, *PC audio editing with Adobe Audition 2.0: Broadcast, desktop and CD audio production*, CRC Press, 2012.

[2] Steve Whittaker and Brian Amento, "Semantic speech editing," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 2004, pp. 527–534.

[3] Zeyu Jin et al., "Speech synthesis for text-based editing of audio narration," 2018.

[4] Max Morrison, Lucas Rencker, Zeyu Jin, Nicholas J Bryan, Juan-Pablo Caceres, and Bryan Pardo, "Context-aware prosody correction for text-based speech editing," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7038–7042.

[5] Manu Airaksinen, Lauri Juvela, Bajibabu Bollepalli, Junichi Yamagishi, and Paavo Alku, "A comparison between straight, glottal, and sinusoidal vocoding in statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 9, pp. 1658–1670, 2018.

[6] Qiong Hu, Korin Richmond, Junichi Yamagishi, and Javier Latorre, "An experimental comparison of multiple vocoder types," in *Eighth ISCA Workshop on Speech Synthesis*, 2013.

[7] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[8] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6199–6203.

[9] Jean-Marc Valin and Jan Skoglund, "Lpcnet: Improving neural speech synthesis through linear prediction," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5891–5895.

[10] Daxin Tan, Liqun Deng, Yu Ting Yeung, Xin Jiang, Xiao Chen, and Tan Lee, "Editspeech: A text based speech editing system using partial inference and bidirectional fusion," *arXiv preprint arXiv:2107.01554*, 2021.

[11] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al., "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[12] 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 *International Conference on Machine Learning*. PMLR, 2018, pp. 5180–5189.

[13] Ye Jia, Yu Zhang, Ron J Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, et al., "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," *arXiv preprint arXiv:1806.04558*, 2018.

[14] Eliya Nachmani, Adam Polyak, Yaniv Taigman, and Lior Wolf, "Fitting new speakers based on a short untranscribed sample," in *International Conference on Machine Learning*. PMLR, 2018, pp. 3683–3691.

[15] Shi-Xiong Zhang, Zhuo Chen, Yong Zhao, Jinyu Li, and Yifan Gong, "End-to-end attention based text-dependent speaker verification," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 171–178.

[16] Lifa Sun, Kun Li, Hao Wang, Shiyin Kang, and Helen Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *2016 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2016, pp. 1–6.

[17] Zeyu Jin, Gautham J Mysore, Stephen Diverdi, Jingwan Lu, and Adam Finkelstein, "Voco: Text-based insertion and replacement in audio narration," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–13, 2017.

[18] Zhizheng Wu, Oliver Watts, and Simon King, "Merlin: An open source neural network speech synthesis system.," in *SSW*, 2016, pp. 202–207.

[19] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu, "Neural speech synthesis with transformer network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 6706–6713.

[20] Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al., "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 2017.

[21] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu, "Libritts: A corpus derived from librispeech for text-to-speech," *arXiv preprint arXiv:1904.02882*, 2019.

[22] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al., "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[24] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio, *Deep learning*, vol. 1, MIT press Cambridge, 2016.

[25] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[26] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *Icml*, 2010.

[27] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.