The NeteaseGames System for fake audio generation task of 2023 Audio Deepfake Detection Challenge

Haoyue Zhan*, Yang Zhang and Xinyuan Yu

NetEase Games AI Lab, Guangzhou, China

Abstract

This paper presents the description of our speech synthesis system for the Audio Deep Synthesis Detection Challenge (ADD 2023). We utilizes a FastPitch-based model augmented with a BERT-based prosody feature and an utterance embedding predictor to model the generated speech. We incorporate these components to improve one-to-many generation modeling. Evaluation results indicate a significant advantage over some false speech detection models, earning a second-place ranking in the competition overall.

Keywords

speech synthesis, fake audio, post-processing, ADD challenge

1. Introduction

The proliferation of deepfake technology has underscored the importance of reliable methods to detect and prevent the malicious use of deepfakes. The Audio Deepfake Detection Challenge (ADD 2023) is a deep learning competition that aims to promote research in the detection and analysis of deepfake audio[1]. The primary objective of the competition is to accelerate the development of more robust and reliable deepfake detection tools for audio and encourage the exploration of cutting-edge techniques in this field.

The competition comprises four tracks, each with a unique focus and set of challenges. In this paper, we describe the speech synthesis system we employed in the fake audio generation task (Track 1.1). This Track centers on "Adversarial Attacks," which involves generating speech to deceive a model with false detection capabilities. Participants are required to train their models using the provided training dataset and evaluate their performance on a test dataset. The success rate of generating speech that can fool the detection model determines the results of this task.

The primary challenge of Track 1.1 is the development of effective adversarial attacks that can generate speech that is difficult for detection models to distinguish from genuine audio. Deepfake audio generated using advanced machine learning techniques can be challenging to detect, even for humans. The competition offers an opportunity for participants to benchmark and compare the performance of their models against those of other participants in a fair and transparent setting.

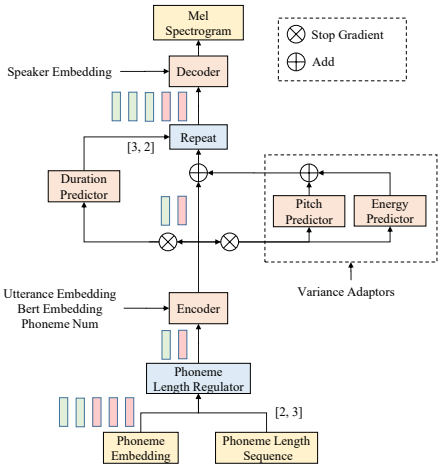


Figure 1: Acoustic Model

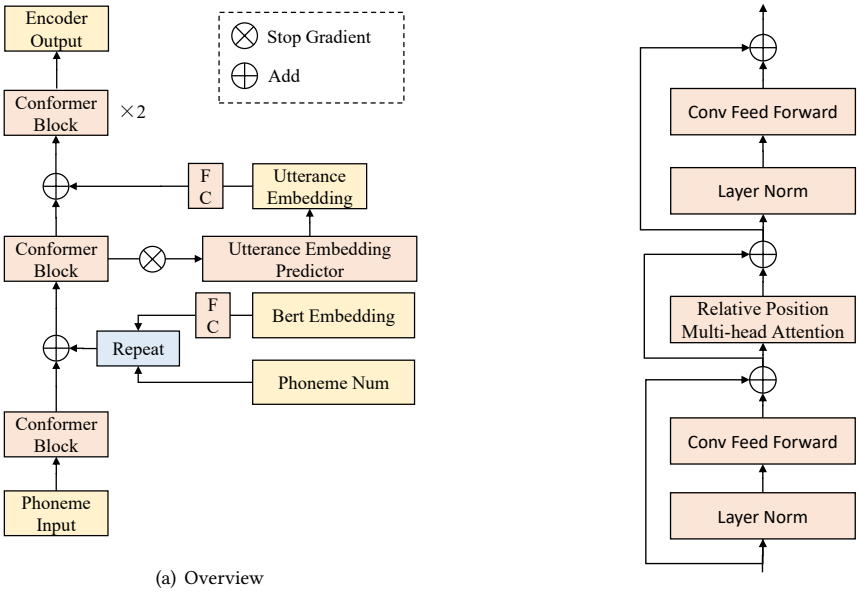
Speech synthesis technology, based on deep learning, has the capability to generate counterfeit speech that mimics a target speaker's voice from text and the target speaker's voice data[2, 3, 4]. Currently, speech synthesis is primarily achieved through two methods: multi-stage synthesis and end-to-end synthesis. Multi-stage synthesis can be categorized further into autoregressive models based on Tacotron[5] and non-autoregressive models based on FastSpeech[6]. For our competition system, we have employed the latter, FastPitch-based model as the acoustic model framework[7].

This paper is organized as follows: Section 2 outlines our data preprocessing process, Section 3 describes our model structure, Section 4 provides detail of our competition results, and finally, the conclusion is presented.

IJCAI 2023 Workshop on Deepfake Audio Detection and Analysis (DADA 2023), August 19, 2023, Macao, S.A.R

*Corresponding author.

zhanhaoyue@corp.netease.com (H. Zhan);
zhangyang09@corp.netease.com (Y. Zhang);
yuxinyuan02@corp.netease.com (X. Yu)



(a) Overview

(b) Conformer block

Figure 2: The Architecture of Encoder

2. Data preprocessing

In accordance with the requirements of Track 1.1 (generation task), the AISHELL-3[8] dataset has been utilized. This extensive Chinese speech corpus comprises over 88 thousand utterances, which amount to 85 hours of speech.

2.1. Encoder Input

Rhythm modeling in speech synthesis plays a crucial role in enhancing the naturalness and fluency of generated speech, particularly with respect to rhythm pauses in unpunctuated long sentences. To address this, we have introduced a pre-trained embedding modeling method based on BERT [1]. However, the length of the BERT embedding is commensurate with the number of words, while the base model of FastSpeech-based aligns longer rhythm pauses as silence in the speech-text alignment data. This misaligns the input BERT embedding sequence with the phoneme sequence length, necessitating some adjustments to the data preprocessing process. To this end, we utilized the MFA[9] tool to force-align speech text. Silences below 250 ms were marked as rhythm pauses and merged with the previous phoneme, while pauses ranging from 250 to 500 ms were marked as regular pauses, and those exceeding 500 ms were marked as long pauses to align punctuation with these two types of pauses. Redundant punctuation was tagged with no duration. To account for the silence mark that may exist at the beginning of a sentence, we introduced a placeholder in the text and phoneme sequence to ensure length matching. Additionally, since a Chinese character in the MFA phoneme set may correspond to an indefinite length of phoneme representation, we recorded Word corresponding Phoneme Num to ensure that the length after upsampling the BERT embedding matches the phoneme sequence exactly. Although AISHELL3 is a purely Chinese dataset, we adapted the language dependent phoneme(LDP) in MFA to IPA and added a phoneme length regulator to the preprocessing process to accommodate potential cross-lingual TTS scenarios[7].

Speech synthesis is a clear one-to-many generation task, wherein differences in prosodic pauses may occur in addition to variations in speech rate, pitch, and energy, even for the same text and speaker. To better capture the variances of generated speech, we have incorporated an utterance predictor modeling method, in addition to BERT embedding. However, unlike the delightful TTS model[10], we have utilized the intermediate vector representation of a pre-trained emotion classification model[2] as the supervisory target for our utterance predictor. The positions of these Encoder inputs in our model are illustrated in Figures 1 and 2.

[1]https://github.com/Executedone/Chinese-FastSpeech2

[2]https://github.com/audeering/w2v2-how-to

2.2. Pitch and Energy

In the first step, we extracted energy features from the linear spectrogram. Subsequently, we utilized the aligned LDP duration to average the frame-level energy sequence and generate the LDP-level energy sequence. The resulting sequence was quantized to aid in subsequent processing. In the second step, we utilized the WORLD[11] tool to extract the pitch sequence for each speaker's speech. Following this, we normalized the sequence using the speaker's average and variance. Subsequently, we obtained the LDP-level normalized real-valued pitch sequence based on the aligned LDP duration.

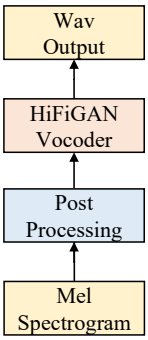


Figure 3: Vocoder Module

3. Acoustic Modeling

Figure 1 showes the overall architecture of the proposed TTS model. Our system can be divided into four components and they are introduced in detail in the following sections.

3.1. Encoder

To overcome any potential pronunciation interference that may arise due to direct superposition of the BERT embedding input and the aggregated phoneme input, we employ a Conformer block to fuse the information. This is accomplished by leveraging a linear layer to adjust the dimension of the BERT embedding input. Analogously, the utterance embedding is similarly inserted and processed through a Conformer block. Ultimately, we obtain the Encoder output by passing the input through two Conformer blocks. The Conformer block utilized in our approach is akin to the one described in the delightful TTS. However, we have made certain modifications, such as removing the Depthwise convolution and substituting the self-attention with Relative Position Multi-head Attention[12] to circumvent any potential instability in pronunciation that may arise towards the close of long sentences due to absolute position information. The detail of the Encoder inputs in the model are illustrated in Figure **??**.

3.2. Explicit modelings

During the training stage, we adopt the approach proposed in [13] and employ a 1-D convolution layer to transform the pitch value into pitch embedding. Similarly, we use a lookup table to convert quantized energy values into energy embedding. During inference stage, we leverage variance adaptors, including the duration predictor, in line with the methodology described in [14]. To ensure that the training of these variance adaptors does not negatively impact the training of the encoder, we implement a stop-gradient operation on the input of all variance adaptors. This is illustrated in the middle part of Figure 1. These measures are imperative to ensure that the encoder is immune to any adverse influence exerted by the variance adaptors, as has been demonstrated in [15, 16, 17].

3.3. Decoder

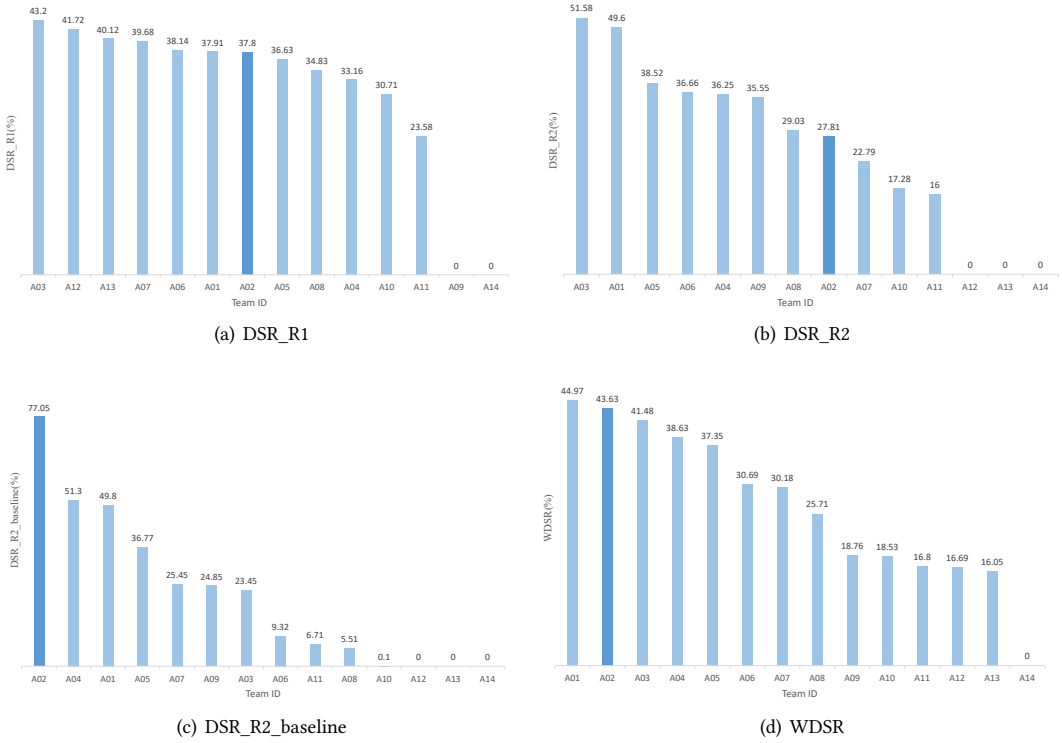
The Decoder architecture, akin to the Encoder, is comprised of four Conformer blocks. In this approach, we fuse the Encoder output, pitch embedding, and energy embedding, and pass the resultant through two Conformer blocks. Subsequently, we incorporate the speaker embedding obtained from the lookup table into the architecture. Finally, we pass this through two Conformer blocks and a linear layer to obtain the mel spectrum.

3.4. Vocoder and Post-processing

The log-mel spectrogram generated by the proposed approach are converted into speech signals using a universal and fine-tuned HiFi-GAN vocoder[18]. This vocoder has been pre-trained on the AISHELL3 dataset. In our experimental setup, we make a post-processing for the input log-mel spectrum before feeding it to the vocoder. This trick has been demonstrated to alleviate some high-frequency noise. The processing steps are outlined below:

$$mel = (mel + s - 1) * s, \quad default\ s = 1.2 \tag{1}$$

while mel denotes the value of log-mel spectrogram and s is a factor used to adjust mel. This adjustment increases the difference between values around 0, which may explain why high-frequency noise will be reduced. However, if the coefficient is multiplied by excessively large value, it can cause an increase in low-frequency energy, which changes the speech quality and reduces



(a) DSR_R1 (b) DSR_R2

(c) DSR_R2_baseline (d) WDSR

Figure 4: Result of Track 1.1 Generation task (FG-G) in ADD Challenge;(a)Round 1 deception success rate (b)Round 2 deception success rate (c)Round 2 baseline model deception success rate (d) Weighted deception success rate

perceived quality. We used default values based on our experience. Additionally, we added an offset due to the logarithmic distribution not being symmetric about the zero-point.

4. Results

4.1. Experimental setup

In our experimental setup, we downsample the AISHELL-3 audio from 44.1 kHz to 24 kHz for the TTS model and to 16 kHz for all evaluations. We represent the audio features as a sequence of 80-dimensional log-mel spectrogram frames. These frames are computed from 40 ms windows that are shifted by 10 ms. The hidden size of the Conformer blocks in our proposed model is set to 128. Each feed-forward layer of the Conformer blocks comprises of two 1-D convolution layers, each with a kernel size of 3 and 1024 intermediate channels. Regarding the acoustic model, we utilize an l1 loss function for the mel spectrum, while the mean-squared error (MSE) loss function is applied to other components. The duration and energy predictor compute the loss in the logarithmic domain.

4.2. Metrics

Track 1.1 of the ADD challenge is an adversarial game that requires participants to generate adversarial samples and enhance the anti-attack capabilities of the audio deepfake detection model from two opposing sides. The generation and detection tasks of Track 1 are evaluated separately. For the generation task (Track 1.1), the deception success rate (DSR) is chosen as the metric. The DSR metric quantifies the extent to which the audio deepfake detection model is deceived by the generated utterances and is defined as follows:

$$DSR = \frac{W}{A \cdot N} \tag{2}$$

where W denotes the count of wrong detection samples by all the detection models on the condition of achieving their respective equal error rate (EER) performance, A is the number of evaluation samples, and N represents the number of detection models. In the first round, the DSR against the Track 1.2 submissions constitutes the overall generation performance metric. In the

second round, weighted consideration is also given to the DSR against the detection model that we release. Specifically, in the second round, the generation performance metric is defined as:

$$WDSR = \alpha DSR_baseline + \beta DSR \quad (3)$$

where α=0.4 and β=0.6, and they denote the respective weights for EER and DSR in our consideration.

4.3. Evaluations

In Track 1.1, participants are tasked with generating attack samples while adhering to the specified text and speaker identities. Our team id is A02. As depicted in Figures 4(a) and 4(b), our deception success rate (DSR) ranks 7th and 8th in the first and second rounds, respectively. Notwithstanding, in Figure 4(c), which displays the DSR scores of the baseline models provided by the organizers in the second round, we secured the first position and outperformed the second-ranking system by a significant margin. Ultimately, our speech synthesis system obtained the second position in the overall scoring shown in Figure 4(d).

5. Conclusion

This paper presents our current speech synthesis system, which has yielded promising results in the adversarial process with the fake audio detection model. Specifically, our system has demonstrated significant advantages over baseline models, thereby validating its efficacy in the task of fake audio generation.

References

[1] Y. Jiangyan, T. Jianhua, F. Ruibo, Y. Xinrui, W. Chenglong, W. Tao, Z. Chuyuan, Z. Xiaohui, Z. Yan, R. Yong, X. Le, Z. Junzuo, G. Hao, W. Zhengqi, L. Shan, L. Zheng, N. Shuai, L. Haizhou, Add 2023: the second audio deepfake detection challenge, in: IJCAI 2023 Workshop on Deepfake Audio Detection and Analysis (DADA 2023), 2023.

[2] J. Kim, J. Kong, J. Son, Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech, in: International Conference on Machine Learning, ICML, 2021.

[3] M. Jeong, H. Kim, S. J. Cheon, B. J. Choi, N. S. Kim, Diff-tts: A denoising diffusion model for text-to-speech, arXiv preprint arXiv:2104.01409 (2021).

[4] X. Tan, J. Chen, H. Liu, J. Cong, C. Zhang, Y. Liu, X. Wang, Y. Leng, Y. Yi, et al., Naturalspeech: End-to-end text-to-speech synthesis with human-level quality, arXiv preprint arXiv:2205.04421 (2022).

[5] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, et al., Tacotron: Towards end-to-end speech synthesis, in: Proc. Interspeech, 2017.

[6] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, T.-Y. Liu, Fastspeech: Fast, robust and controllable text-to-speech, in: Advances in Neural Information Processing Systems(NeurIPS), 2019, pp. 3171–3180.

[7] H. Zhan, X. Yu, H. Zhang, Y. Zhang, Y. Lin, Exploring Timbre Disentanglement in Non-Autoregressive Cross-Lingual Text-to-Speech, in: Proc. Interspeech 2022, 2022.

[8] Y. Shi, H. Bu, X. Xu, S. Zhang, M. Li, Aishell-3: A multi-speaker mandarin tts corpus and the baselines, arXiv preprint arXiv:2010.11567 (2020).

[9] M. McAuliffe, M. Socolof, S. Mihuc, et al., Montreal forced aligner: Trainable text-speech alignment using kaldi., in: Interspeech, 2017.

[10] Y. Liu, Z. Xu, G. Wang, K. Chen, B. Li, X. Tan, J. Li, L. He, S. Zhao, Delightfultts: The microsoft speech synthesis system for blizzard challenge 2021, arXiv preprint arXiv:2110.12612 (2021).

[11] M. Morise, F. Yokomori, K. Ozawa, World: a vocoder-based high-quality speech synthesis system for real-time applications, IEICE Transactions on Information and Systems 99 (2016) 1877–1884.

[12] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, D. Eck, Music transformer, arXiv preprint arXiv:1809.04281 (2018).

[13] A. Łańcucki, Fastpitch: Parallel text-to-speech with pitch prediction, in: 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021.

[14] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, T.-Y. Liu, Fastspeech 2: Fast and high-quality end-to-end text-to-speech, in: International Conference on Learning Representations(ICLR), 2021.

[15] J. Kim, S. Kim, J. Kong, S. Yoon, Glow-tts: A generative flow for text-to-speech via monotonic alignment search, in: Advances in Neural Information Processing Systems, 2020.

[16] T. Raitio, R. Rasipuram, D. Castellani, Controllable Neural Text-to-Speech Synthesis Using Intuitive Prosodic Features, in: Proc. Interspeech, 2020.

[17] C. Gong, L. Wang, Z. Ling, S. Guo, J. Zhang, J. Dang, Improving naturalness and controllability of sequence-to-sequence speech synthesis by learning local prosody representations, in: 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021.

[18] J. Kong, J. Kim, J. Bae, Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis, Advances in Neural Information Processing Systems (2020).