listed in Table 1.

2.2.1 Pitch

We extract the pitch contours from singing voice and speech, and compare the range and stability of the pitch. Here we use semitone as the unit of pitch.

Pitch Range Generally speaking, the range of the pitch in singing voice is larger than that in speech. Loscos et al. [15] have pointed out that the frequency range in singing voice can be much larger compared to that in speech. For each sentence, we calculate the pitch range (the maximum pitch value minus the minimum pitch value in this sentence). After averaging the pitch range of the overall 10K sentences in the corpus, the average values are listed in *Pitch Range* in Table 1.

Pitch Stability The pitch of each frame in a certain syllable (when it is corresponding to a note) in singing voice remains almost constant whereas in speech the pitch changes freely along with the audio frames in a syllable. We call the characteristic of maintaining local stability within a syllable as *Pitch Stability*. Specifically, we calculate the pitch difference between every two adjacent frames in a sentence and average it across the entire corpus of 10K sentences. The larger the value of *Pitch Stability*, the more stable the pitch. The results can be seen in *Pitch Stability* of Table 1.

2.2.2 Duration

We also analyze and compare the range and stability of the syllable duration in singing voice and speech. The duration of each syllable varies a lot along with the melody in singing voice. While in speech, it depends on the pronunciation habits of the certain speaker.

Duration Range For each sentence, we calculate the difference between the duration of the longest syllable and the shortest syllable as the duration range. The average values of the duration ranges in the entire corpus are shown as *Duration Range* in Table 1.

Duration Variance We calculate the variance of the duration of syllables in each sentence and average the variances of all sentences in the whole corpus. The results are listed as *Duration Variance* in Table 1 to reflect the flexibility of duration in singing voice.

Property	*Speech*	*Singing Voice*
Pitch Range (semitone)	12.71	14.61
Pitch Stability	0.93	0.84
Duration Range (s)	0.44	2.40
Duration Variance	0.01	0.11

Table 1: The differences of acoustic properties between speech and singing voice.

Besides the differences in the characteristics we mentioned above, sometimes singers may add vibrato in some long vowels or make artistic modifications to the pronunciation of some words to make them sound more melodious, though it will result in a loss of intelligibility. Considering that some characteristics are hard to be quantified, in this work, we start with pitch and duration to build a prototype and propose PDAugment to augment ALT training data by adjusting pitch and duration of speech at syllable level according to music scores.

3. METHOD

3.1 Pipeline Overview

For automatic lyrics transcription, we follow the practice of existing automatic speech recognition systems and choose Conformer encoder [19] and Transformer decoder [20] as our basic model architecture. Different from standard ASR systems, we add a PDAugment module in front of the encoder as shown in Figure 1, to apply syllable-level adjustments to the pitch and duration of the input natural speech according to the information of aligned notes. When the input of ALT system is singing voice, we just do not enable PDAugment module. When the input is speech, PDAugment module takes the note information extracted from music scores as extra input to adjust the pitch and duration of speech, and then adds the adjusted speech into training data to enhance the ALT model. The loss function

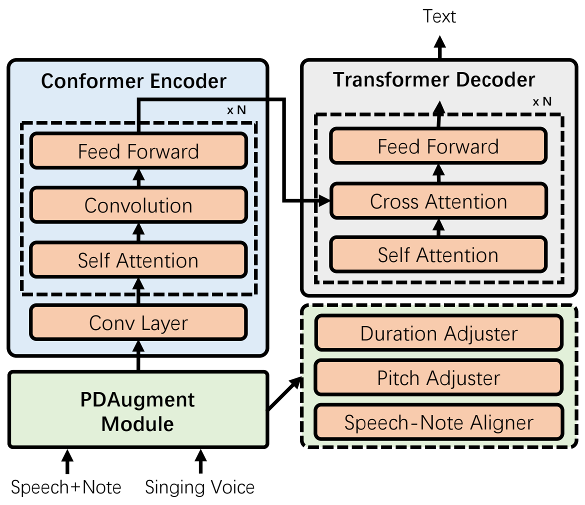

Figure 1: The overall pipeline of the ALT system equipped with PDAugment.

of the ALT model consists of decoder loss $\mathcal{L}_{dec}$, and ctc loss (on top of the encoder) $\mathcal{L}_{ctc}$: $\mathcal{L} = (1-\lambda)\mathcal{L}_{dec}+\lambda\mathcal{L}_{ctc}$, where λ is a hyperparameter to trade off the two loss terms. Considering that lyrics may contain more musical-specific expressions which are rarely seen in natural speech, the probability distributions of lyrics and standard text are quite different. We train the language model with in-domain lyrics data and then fuse it with ALT model in the beam search of the decoding stage.

We try to make the speech fit the patterns of singing voice more naturally as well as to achieve the effect of "singing out" the speech by PDAugment, so we adjust the pitch and duration of speech at syllable level according to those of corresponding notes in music scores instead of applying random adjustments. To do so, we propose PDAugment module, which consists of three key components:

1) speech-note aligner, which generates the syllable-level alignment to decide what the corresponding note of a certain syllable is for subsequent adjusters; 2) pitch adjuster, which adjusts the pitch of each syllable in speech according to that of aligned notes; and 3) duration adjuster, which adjusts the duration of each syllable in speech to be in line with the duration of the corresponding notes. We introduce each part in the following subsections.

3.2 Speech-Note Aligner

According to linguistic and musical knowledge, in singing voice, syllables can be viewed as the smallest textual unit corresponding to notes in the melodies. PDAugment adjusts the pitch and duration of natural speech at syllable level under the guidance of note information obtained from the music scores. In order to apply the syllable-level adjustments, we propose a speech-note aligner, which aims to align the speech and note (in melody) at syllable level.

The textual content can serve as the bridge of aligning notes of melody with the speech. Specifically, our speech-note aligner aligns the speech with notes (in melody) in the following steps:
1) In order to obtain the syllable-level alignment of text and speech, we first convert the text to phoneme using phonemizer [1] and then align the text with speech audio using the Montreal forced aligner (MFA) [21] tool [2] at phoneme level. Next, we group several phonemes into a syllable according to the linguistic rules [22] and get the syllable-level alignment of text and speech.
2) For the syllable-to-note mappings, we set one syllable to correspond to one note by default, because in most cases of singing voice, one syllable is aligned with one note [23]. Only when the time length ratio of the syllable in speech and the note in melody exceeds the predefined thresholds (the lower bound of the ratio is set to 0.5 as the upper bound to 2 from the perspective of preventing severe distortion of the sound), we generate one-to-many or many-to-one mappings to prevent audio distortion after adjustments.
3) We aggregate the syllable-level alignment of text and speech, and the syllable-to-note mappings to generate the syllable-level alignment of speech and note (in melody) as the input of the pitch and duration adjusters.

3.3 Pitch Adjuster

Pitch adjuster adjusts the pitch of input speech at syllable level according to the aligned notes. Specifically, we use WORLD [24], a fast and high-quality vocoder-based speech synthesis system to implement the adjustment. The WORLD system [3] parameterizes speech into three components: fundamental frequency (F0), aperiodicity, and spectral envelope and can reconstruct the speech with only estimated parameters. We use WORLD to estimate the three parameters of natural speech and only adjust the F0 contours according to that of corresponding notes. Then we synthesize speech with adjusted F0 accompanied by original aperiodicity and spectral envelope. Figure 2 shows the F0 contours before and after the pitch adjuster.

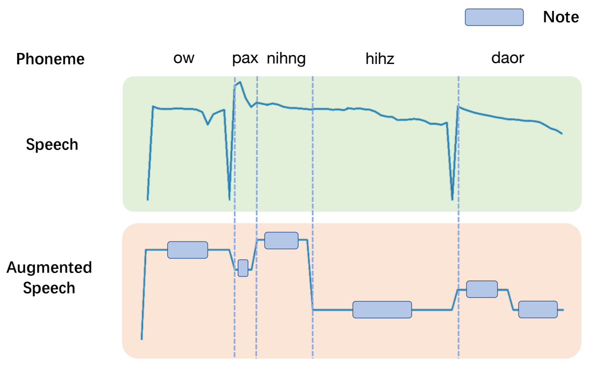

Figure 2: The change of F0 contour after pitch adjuster. The content of this example is "opening his door".

Pitch adjuster calculates the pitch difference between speech and note with syllable-level alignment and adjusts F0 contours of speech accordingly. Some details are as follows: 1) Considering that the quality of synthesized speech will drop sharply when the range of adjustment is too large, we need to keep it within a reasonable threshold. Specifically, we calculate the average pitch of the speech and the corresponding melody respectively. When the average pitch of speech is too different from that of the corresponding melody (e.g., exceeding a threshold, which is 5 semitones in our experiment), we shift the pitch of the entire note sequence to make the difference within the threshold and use the shifted note for adjustment, otherwise, keep the pitch of the original note unchanged; 2) To maintain smooth transitions in synthesized speech and prevent speech from being interrupted, we perform pitch interpolation for the frames between two syllables. 3) When a syllable is mapped to multiple notes, we segment the speech of this syllable in proportion to the duration of notes and adjust the pitch of each segment according to the corresponding note.

3.4 Duration Adjuster

Duration adjuster changes the duration of input speech to align with the duration of the corresponding note. As shown in Figure 3, instead of scaling the whole syllable, we only scale the length of vowels and keep the length of consonants unchanged, because the duration of consonants in singing voice is not significantly longer than that in speech, while long vowels are common in singing voice [5]. There are one-to-many mappings and many-to-one mappings in the syllable-level alignment. For the first case, we calculate the total length of the syllables and adjust the length of all vowels in these syllables in proportion. For the second case, we adjust the length of the vowel, so that the total length of this syllable is equal to the total length of these notes.

[1] https://github.com/bootphon/phonemizer
[2] https://github.com/MontrealCorpusTools/Montreal-Forced-Aligner
[3] https://github.com/JeremyCCHsu/Python-Wrapper-for-World-Vocoder

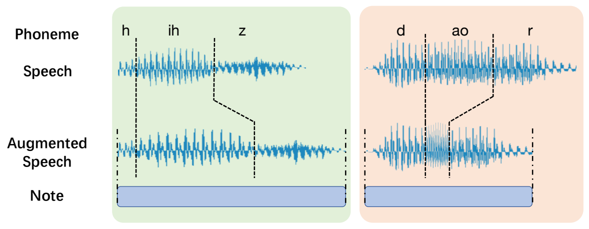

Figure 3: The change of duration after duration adjuster. The content of this example is "his door". The left block shows the case of lengthening the duration of speech and the right shows the case of shortening the duration.

4. EXPERIMENTS AND RESULTS

In this section, we first describe the experimental settings, including datasets, model configuration, and the details of training, inference, and evaluation. Then we report the experiment results, visualize the effect of PDAugment, and further conduct more analyses to verify the effectiveness of PDAugment.

4.1 Experimental Settings

4.1.1 Datasets

Singing Voice Datasets We conduct experiments on two singing voice datasets to verify the effectiveness of our PDAugment: DSing30 dataset [10] and Dali corpus (v2.0) [18, 25]. DSing30 dataset consists of about 4K monophonic Karaoke recordings of English pop songs with nearly 80K utterances, performed by 3,205 singers. We use the partition provided by [10] to make a fair comparison with them. Dali corpus is another large dataset of synchronized audio, lyrics, and notes. It consists of 1200 English polyphonic songs (with background music) for a total duration of 70 hours. Following [6], we use the sentence-level annotation of the dataset provided by [18] [4] and divide the dataset into training, development, and test with a proportion of 8:1:1 without any singers overlapping in each partition. We convert all the singing voice waveforms in our experiments into mel-spectrogram following [26] with a frame size of 25 ms and hop size of 10 ms.

Natural Speech Dataset Following the common practice in previous ASR work [19], we choose the widely used LibriSpeech [17] corpus as the natural speech dataset in our experiments and use the official training partition. The LibriSpeech corpus contains 960 hours of speech sampled at 16 kHz with 1129 female speakers and 1210 male speakers. Similar to singing voice, we convert the speech into mel-spectrogram with the same setting [27].

Music Score Dataset In this work, we choose the pop music subset of FreeMidi dataset [5] because almost all of the songs in our singing voice datasets are pop music. The pop music subset of FreeMidi has about 4000 MIDI files, which are used to provide note information for PDAugment module.

[4] https://github.com/gabolsgabs/DALI
[5] https://freemidi.org/genre-pop

Lyrics Corpus In order to construct a language model with more in-domain knowledge, we deliberately collected a large amount of lyrics data to build our lyrics corpus for language model training. Besides the training text of DSing30 dataset and Dali corpus, we collect lyrics of English pop songs from the Web. We crawl about 46M lines of lyrics and obtain nearly 17M sentences after removing duplicates including DSing30 dataset and Dali corpus. We attach a subset of the collected lyrics corpus in the supplementary material.

4.1.2 Model Configuration

We choose Conformer encoder [19] and Transformer decoder [20] as the basic ALT model architecture in our experiments since the effectiveness of the structure has been proved in ASR. Our language model is based on Transformer encoder. More details about the model configuration, training/inference settings, and codes are attached in the supplementary materials.

4.2 Results

In this subsection, we report the experimental results of the ALT system equipped with PDAugment in two singing voice datasets. We compare our results with several basic settings as baselines: 1) *Naive ALT*, the ALT model trained with only singing voice dataset; 2) *ASR Augmented*, the ALT model trained with the combination of singing voice dataset and ASR data directly; 3) *Random Aug* [5], the ALT model trained with the combination of singing voice dataset and randomly adjusted ASR data. The pitch is adjusted ranging from -6 to 6 semitones randomly and the duration ratio of speech before and after adjustment is randomly selected from 0.5 to 1.2. All 1), 2), and 3) are using the same model architecture as *PDAugment*. Besides the above three baselines, we compare our PDAugment with the previous systems which reported the best results in two datasets respectively. We compare our PDAugment with [11] using RNNLM for DSing30 dataset and compare the results with [6] for Dali corpus, All the results are listed in Table 2.

Method	DSing30		Dali	
	Dev	*Test*	*Dev*	*Test*
Naive ALT	28.2	27.4	80.9	86.3
ASR Augmented	20.8	20.5	75.5	75.7
Random Aug [5]	17.9	17.6	69.8	72.1
Previous Work [11]	17.7	15.7	-	-
Previous Work [6]	-	-	75.2	78.9
PDAugment	**10.1**	**9.8**	**53.4**	**54.0**

Table 2: The WERs (%) of DSing30 and Dali dataset. The audios of Dali corpus contain background music.

As can be seen, *Naive ALT* performs not well and gets high WERs in both DSing30 dataset and Dali corpus, which demonstrates the difficulty of the ALT task. After adding ASR data for ALT training, the performances of *ASR Augmented* setting in both of the two datasets have been improved slightly compared to *Naive ALT*, but still with relatively high WERs, which indicates the limitation of using ASR training data directly.

When applying the adjustments, a question is if we adjust the pitch and duration with random ranges without note information from music scores, how well will the ALT system perform? The results of *Random Aug* can perfectly answer this question. As the results show, *Random Aug* can slightly improve the performance compared with *ASR Augmented*, demonstrating that increasing the volatility of pitch and duration in natural speech helps ALT system training, which is the same as what [5] claimed. *PDAugment* is significantly better than *Random Aug*, which indicates that adjusted speech can better help the ALT training with the guidance of music scores.

Besides, it is obvious that *PDAugment* greatly outperforms the previous work in both datasets. [11] performs worse than *PDAugment* because of not taking advantage of the massive ASR training data. Compared with [6] that replaced F0 contours of speech directly, *PDAugment* can narrow the gap between natural speech and singing voice in a more reasonable manner and achieve the lowest WERs among all the above methods. The results in both datasets show the effectiveness of PDAugment for ALT task and reflect the superiority of adding music-specific acoustic characteristics into natural speech.

4.3 Ablation Studies

We conduct more experimental analyses to deeply explore our PDAugment and verify the necessity of some design details. More ablation studies (different language models) can be found in the supplementary materials. The ablation studies are carried out on DSing30 dataset in this section.

4.3.1 Augmentation Types

In this subsection, we explore the effects of different kinds of augmentations (only adjust pitch, only adjust duration, and adjust pitch & duration) on increasing the performance of ALT system. We generate three types of adjusted speech by enabling different adjusters of PDAugment module and conduct the experiments on DSing30. The results are shown in Table 3.

Setting	*DSing30 Dev*	*DSing30 Test*
PDAugment	**10.1**	**9.8**
Disable Pitch Adjuster	13.6	13.4
Disable Duration Adjuster	13.8	13.8
Disable both Adjusters	20.8	20.5

Table 3: The WERs (%) of different types of augmentation of DSing30 dataset. All of the settings are trained on DSing30 and the original or adjusted LibriSpeech data.

As we can see, when we enable the whole PDAugment module, the ALT system can achieve the best performance, indicating the effectiveness of the pitch and duration adjusters. When we disable the pitch adjuster, the WER on DSing30 is 3.6% higher than *PDAugment*. The same thing happens when we disable the duration adjuster, the WER is 4.0% higher than *PDAugment*. And if both the pitch and duration are not adjusted, which means using the speech data directly for ALT training, the WER is the worst among all settings. The results demonstrate that both pitch and duration adjusters are necessary and can help with improving the recognition accuracy of the ALT system.

4.4 Adjusted Speech Analyses

Following Section 2.2, we analyze the acoustic properties of the original natural speech and the adjusted speech by random and PDAugment, and list the results in Table 4.

Property	*Original*	*Random*	*PDAugment*
Pitch Range (semitone)	12.71	13.87	14.19
Pitch Stability	0.93	0.59	0.69
Duration Range (s)	0.44	0.42	0.59
Duration Variance	0.01	0.01	0.05

Table 4: The differences of acoustic properties between original speech and adjusted speech.

Combining the information of Table 1 and Table 4, we can clearly find that the distribution pattern of acoustic properties (pitch and duration) after PDAugment is closer to singing voice compared with the original speech, which indicates that our PDAugment can change the patterns of pitch and duration in original speech and effectively narrow the gap between natural speech and singing voice. To avoid the distortion of adjusted speech, we limit the adjustment degree within a reasonable range, so the statistics of adjusted speech can not completely match that of singing voice. Nonetheless, adjusted speech is still good enough for ALT model to capture some music-specific characteristics.

In order to visually demonstrate the effect of PDAugment module, we plot the spectrograms of speech to compare the acoustic characteristics before and after different types of adjustments. The visualization of an example adjusted speech is attached in the supplementary materials.

5. CONCLUSION

In this paper, we proposed PDAugment, a data augmentation method by adjusting pitch and duration to make better use of natural speech for ALT training. PDAugment transfers natural speech into singing voice domain by adjusting pitch and duration at syllable level under the instruction of music scores. PDAugment module consists of speech-note aligner to align the speech with notes, and two adjusters to adjust pitch and duration respectively. Experiments on two singing voice datasets show that PDAugment can significantly reduce the WERs of ALT task. We also explore different types of augmentation, and further verify the effectiveness of PDAugment. In the future, we will consider narrowing the gap between natural speech and singing voice from more aspects such as vibrato, and try to add some music-specific constraints in the decoding stage.

6. ACKNOWLEDGEMENT

This work was supported by the Key Project of Natural Science Foundation of Zhejiang Province (No.LZ19F020002), the Key R&D Program of Zhejiang Province (No.2022C03126), and Project of Key Laboratory of Intelligent Processing Technology for Digital Music (Zhejiang Conservatory of Music), Ministry of Culture and Tourism (No.2022DMKLB001). Thanks to Future Design Laboratory of Zhejiang University for the help in dataset collection. Thanks are also due to Shen Zhou for bringing insights in music.

7. REFERENCES

[1] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[2] A. M. Kruspe, "Bootstrapping a system for phoneme recognition and keyword spotting in unaccompanied singing." in *ISMIR*, 2016, pp. 358–364.

[3] A. Mesaros and T. Virtanen, "Adaptation of a speech recognizer for singing voice," in *2009 17th European Signal Processing Conference*. IEEE, 2009, pp. 1779–1783.

[4] C.-P. Tsai, Y.-L. Tuan, and L.-s. Lee, "Transcribing lyrics from commercial song audio: the first step towards singing content processing," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5749–5753.

[5] A. M. Kruspe, "Training phoneme models for singing with" songified" speech data." in *ISMIR*, 2015, pp. 336–342.

[6] S. Basak, S. Agarwal, S. Ganapathy, and N. Takahashi, "End-to-end lyrics recognition with voice to singing style transfer," *arXiv preprint arXiv:2102.08575*, 2021.

[7] C. Gupta, H. Li, and Y. Wang, "Automatic pronunciation evaluation of singing." in *Interspeech*, 2018, pp. 1507–1511.

[8] C. Gupta, E. Yılmaz, and H. Li, "Automatic lyrics alignment and transcription in polyphonic music: Does background music help?" in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 496–500.

[9] S. Ahlbäck, "Mstre-net: Multistreaming acoustic modeling for automatic lyrics transcription," in *ISMIR2021, International Society for Music Information Retrieval*, 2021.

[10] G. R. Dabike and J. Barker, "Automatic lyric transcription from karaoke vocal tracks: Resources and a baseline system." in *INTERSPEECH*, 2019, pp. 579–583.

[11] E. Demirel, S. Ahlbäck, and S. Dixon, "Automatic lyrics transcription using dilated convolutional neural networks with self-attention," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[12] H. Fujihara, M. Goto, J. Ogata, K. Komatani, T. Ogata, and H. G. Okuno, "Automatic synchronization between lyrics and music cd recordings based on viterbi alignment of segregated vocal signals," in *Eighth IEEE International Symposium on Multimedia (ISM'06)*. IEEE, 2006, pp. 257–264.

[13] J. Parekh, P. Rao, and Y.-H. Yang, "Speech-to-singing conversion in an encoder-decoder framework," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 261–265.

[14] D.-Y. Wu and Y.-H. Yang, "Speech-to-singing conversion based on boundary equilibrium gan," in *InterSpeech*, 2020.

[15] A. Loscos, P. Cano, and J. Bonada, "Low-delay singing voice alignment to text," in *ICMC*, vol. 11, 1999, pp. 27–61.

[16] A. Mesaros, "Singing voice identification and lyrics transcription for music information retrieval invited paper," in *2013 7th Conference on Speech Technology and Human-Computer Dialogue (SpeD)*. IEEE, 2013, pp. 1–10.

[17] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[18] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, "Dali: A large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, 2018, pp. 431–437.

[19] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[21] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi." in *Interspeech*, vol. 2017, 2017, pp. 498–502.

[22] D. M. Kearns, "Does english have useful syllable division patterns?" *Reading Research Quarterly*, vol. 55, pp. S145–S160, 2020.

[23] E. Nichols, D. Morris, S. Basu, and C. Raphael, "Relationships between lyrics and melody in popular music," in *ISMIR 2009-Proceedings of the 11th International Society for Music Information Retrieval Conference*, 2009, pp. 471–476.

[24] M. Morise, F. Yokomori, and K. Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[25] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, "Creating dali, a large dataset of synchronized audio, lyrics, and notes," *Transactions of the International Society for Music Information Retrieval*, vol. 3, no. 1, 2020.

[26] C. Zhang, X. Tan, Y. Ren, T. Qin, K. Zhang, and T.-Y. Liu, "Uwspeech: Speech to speech translation for unwritten languages," *arXiv preprint arXiv:2006.07926*, 2020.

[27] C. Zhang, Y. Ren, X. Tan, J. Liu, K. Zhang, T. Qin, S. Zhao, and T.-Y. Liu, "Denoispeech: Denoising text to speech with frame-level noise modeling," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7063–7067.

PARAMETER SENSITIVITY OF DEEP-FEATURE BASED EVALUATION METRICS FOR AUDIO TEXTURES

Chitralekha Gupta **Yize Wei** **Zequn Gong**
Purnima Kamath **Zhuoyao Li** **Lonce Wyse**
National University of Singapore
chitralekha@u.nus.edu, yize.wei@u.nus.edu, zequn.gong@u.nus.edu
purnima.kamath@u.nus.edu, zhuoyaoli@u.nus.edu, lonce.wyse@nus.edu.sg

ABSTRACT

Standard evaluation metrics such as the Inception score and Fréchet Audio Distance provide a general audio quality distance metric between the synthesized audio and reference clean audio. However, the sensitivity of these metrics to variations in the statistical parameters that define an audio texture is not well studied. In this work, we provide a systematic study of the sensitivity of some of the existing audio quality evaluation metrics to parameter variations in audio textures. Furthermore, we also study three more potentially parameter-sensitive metrics for audio texture synthesis, (a) a Gram matrix based distance, (b) an Accumulated Gram metric using a summarized version of the Gram matrices, and (c) a cochlear-model based statistical features metric. These metrics use deep features that summarize the statistics of any given audio texture, thus being inherently sensitive to variations in the statistical parameters that define an audio texture. We study and evaluate the sensitivity of existing standard metrics as well as Gram matrix and cochlear-model based metrics in response to control-parameter variations for audio textures across a wide range of texture and parameter types, and validate with subjective evaluation. We find that each of the metrics is sensitive to different sets of texture-parameter types. This is the first step towards investigating objective metrics for assessing parameter sensitivity in audio textures.

1. INTRODUCTION

Audio textures are rich and varied sounds that are produced by a superposition of multiple acoustic events. Unlike the sound of an individual event, such as a footstep, or the complex spectrotemporal structures and sequences of speech or music, an audio texture is defined by properties or parameters that remain constant over time [1] despite statistical variation in the sound over time or between instances. For parametric audio texture synthesis, the goal is to generate novel sounds with descriptive parameters that match those of a target texture. Standard objective evaluation metrics for audio texture synthesis include diversity and quality based measures such as the Inception score [2], and Fréchet Audio Distance (FAD) [3]. However, the existing metrics have not been studied for their sensitivity to systematic variation in the parameter values that define the synthesized audio textures.

Various audio texture synthesis algorithms have been applied to modeling a wide range of natural audio texture types including rhythmic (eg. drums, tapping), pitched (eg. windchimes, churchbells) and other natural sounds (eg. rain, wind), but with different degrees of success. Evaluation of parametric variation textures has typically been through human listening experiments [4, 5].

"Deep features" (activation patterns across layers of neural networks) have been explored for various evaluation tasks such as out-of-distribution detection in audio classification [6], perceptual image similarity evaluation [7], audio distortion assessment [3] and audio texture synthesis [1, 5, 8]. In this work we study several existing objective metrics and also explore deep features and cochlear channel model statistics for potential evaluation metrics sensitive to parameter variations. We make use of a controlled audio texture dataset [9] to study these metrics for a wide range of audio textures - rhythmic, pitched, and non-rhythmic non-pitched under systematic parametric variation. We present a comparative study of the parameter sensitivity of the metrics and validate their reliability through human listening tests.

2. RELATED WORK

2.1 Why is a parameter sensitive metric needed?

McDermott and Simoncelli [1] developed a set of statistics based on a cochlear model to describe the perceptually relevant aspects of a given audio texture. To synthesize a new audio texture, a random input is iteratively perturbed until its statistics match those of a target. This algorithm produces convincing audio for many natural textures such as insect swarms, a stream, or applause. However, human listeners give low scores on realism for resynthesized pitched and rhythmic textures, such as wind chimes, drum break, walking on gravel, and church bells. Besides the audio quality, the statistical parameters associated with these sounds can fail to be faithfully preserved in the synthesized sounds [4]. Moments derived from the time-averaged scattering coefficients when used for resynthesis of audio textures have also shown similar performance but using fewer coefficients [10, 11]. An objective metric for predicting these human evaluations of texture synthesis failures would be valuable.

Gatys et al. [12] did seminal work on image texture synthesis that replaced hand-crafted statistics with Gram matrix statistics computed as the correlation between hidden feature activations from layers of a trained convolutional neural network (CNN). Iteratively perturbing a random input to match Gram matrices produces compelling and novel image textures. Similarly, Ulyanov et al. [8] improved the quality of synthetic textures based on the dataset from McDermott and Simoncelli [1]. Antognini et al. [5] extended Ulyanov's work by modifying the architecture with 6 parallel single-layered untrained CNNs each with a different convolutional kernel size, and computed autocorrelation and diversity losses in addition to the original Gram matrix loss. They demonstrated some improvements, but found failure modes similar to that of McDermott and Simoncelli [1] using a combination of objective and subjective evaluation. Caracalla and Roebel [13] also relied on human listening tests to show the improvement in sound quality achieved in this kind of iterative texture synthesis using a complex spectrogram audio representation.

The key take-away from these previous studies is that there is a need for evaluating the preservation of statistical parameters in synthesised audio texture but a lack of an objective metric for that purpose.

2.2 The existing audio synthesis evaluation metrics

The evaluation of generative models in terms of perceptual realism is challenging. However, various objective metrics have been formulated for assessing the quality of synthesized audio textures. For example, L2 distance, cosine distance, and signal-to-distortion ratio use a clean reference signal to compare with the modified or enhanced synthetic signal. Such signal-level metrics are agnostic to the type of audio that is being enhanced. However, these metrics may not always correlate with human perception. FAD [3] takes a different approach as a reference-independent metric that, instead of looking at individual clips, computes the distance between the distribution of embedding statistics generated on a large set of clips.

The Inception score [2, 14, 15], passes generated examples through a pre-trained classifier. The mean KL divergence between the conditional output class probabilities and the marginal distribution of the same are then calculated. The Inception score penalizes models whose examples are not easily classified into a single class, as well as models whose examples collectively belong to only a few of the possible classes. However, the Inception score is not the right tool for evaluating a model's sensitivity to parameter differences between sounds within a class.

The goal of most of these audio quality assessment metrics has been to quantify the degradation of an enhanced/modified audio signal. However, to the best of our knowledge, there is a lack of a systematic study of the sensitivity of these metrics to parameter variations in synthesized audio. Such a study is particularly challenging for audio texture synthesis because of the inherent variations that correspond to a particular parametric description.

3. METRICS

We study two standard metrics and three Gram matrix metrics for sensitivity to parametric changes to audio textures.

3.1 Existing Metrics

3.1.1 Fréchet Audio Distance (FAD)

FAD [3] is a reference-independent metric for audio quality assessment that computes the Fréchet distance [16] between the multi-variate Gaussian distributions of the embeddings of train and test set audio data. These 128 dimensional embeddings are extracted from a VGG-ish model [17] pre-trained on clean audio data for classification.

$$FAD = ||\mu_b - \mu_t||^2 + tr(\Sigma_b + \Sigma_t - 2\sqrt{\Sigma_b \cdot \Sigma_t}) \quad (1)$$

where the training and test data embeddings are assumed to have multivariate Gaussian distributions $\mathcal{N}(\mu_b, \Sigma_b)$ and $\mathcal{N}(\mu_t, \Sigma_t)$, respectively. We used the open-source model and FAD computation code [3].

3.1.2 L2 Distance

As another standard metric for our study, we compute the Euclidean distance between the two matrices X_A and X_B, representing the spectrograms of the two input audio signals to be compared, x_A and x_B respectively.

3.2 Gram matrix based Metrics

3.2.1 Gram matrix Metric (GM)

Following Ulyanov and Lebedev [8] and Antognini et al. [5], we use 2D spectrogram representations of audio for iterative updates through the CNNs used to compute a summary statistic in the form of a Gram matrix. Two audio textures sound similar if the computed Gram matrices are similar. Here, we leverage on this property of Gram matrix to develop a metric designed to be sensitive to parameter variations. We define the Gram matrix metric as the mean squared error between the Gram matrices of two textures. We hypothesize that this metric would be sensitive to parametric variations in the statistics of the same texture type such as different rates of flowing water.

Formally, we define the Gram matrix metric GM as,

$$GM = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{D}||G_A^n - G_B^n||_2^2 \quad (2)$$

where, G_A^n and G_B^n are flattened Gram matrices derived from audio clips A and B respectively for the n^{th} CNN in an ensemble of N CNNs, and D is the total number of elements in the Gram matrix. In this work, we adopted the