Black-Box Audio Adversarial Example Generation Using Variational Autoencoder

Wei Zong(✉), Yang-Wai Chow(✉), and Willy Susilo

Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Wollongong, NSW, Australia
{wzong,caseyc,wsusilo}@uow.edu.au

Abstract. Automatic speech recognition (ASR) applications are ubiquitous these days. A variety of commercial products utilize powerful ASR capabilities to transcribe user speech. However, as with other deep learning models, the techniques underlying ASR models suffer from adversarial example (AE) attacks. Audio AEs resemble non-suspicious audio to the casual listener, but will be incorrectly transcribed by an ASR system. Existing black-box AE techniques require excessive requests sent to a targeted system. Such suspicious behavior can potentially trigger a threat alert on the system. This paper proposes a method of generating black-box AEs in a way that significantly reduces the required amount of requests. We describe our proposed method and presents experimental results demonstrating its effectiveness in generating word-level and sentence-level AEs that are incorrectly transcribed by an ASR system.

Keywords: Adversarial example · Automatic speech recognition · Deep learning · Machine learning · Variational autoencoder

1 Introduction

Automatic speech recognition (ASR) applications, such as Google Assistant, play an important role in our daily lives. Modern ASR systems are commonly based on deep learning. However, deep neural networks can be fooled by adversarial examples (AEs). AEs were first investigated in the image domain [21]. In general, AEs that look indistinguishable from their original images can be generated by adding small perturbations to the original input images. Despite appearing to be indistinguishable to the human visual system, these AEs will be misclassified by deep learning models.

While much of the research community has focused on investigating AEs in the image domain [7,14,16], there is less research on AEs in the audio domain. Audio AEs can be classified as targeted and non-targeted. The aim of non-targeted audio AEs is to make an ASR model incorrectly transcribe speech in input audio, while the aim of targeted audio AEs is to cause an ASR model to output a specific transcription injected by an adversary. This paper focuses on non-targeted audio AEs.

D. Gao et al. (Eds.): ICICS 2021, LNCS 12919, pp. 142–160, 2021.
https://doi.org/10.1007/978-3-030-88052-1_9

To date, many audio AE generation techniques adopt a white-box threat model, whereby an adversary knows the internal workings of the target ASR model [4,18,20]. However, a white-box threat model is not practical in the real-world, since commercial ASR application developers do not typically reveal the internal workings of their systems. Thus, black-box audio AE generation techniques are a more practical alternative. Under a black-box assumption, an adversary can only probe the ASR system with input audio and analyze the resulting transcription. Given this challenging problem, there are few studies on generating black-box audio AEs [2,12,22]. These black-box AE generation techniques are all based on the use of genetic algorithms.

Techniques for generating white-box audio AEs are usually formulated as an optimization problem, where input audio is optimized in the direction of the gradient of the loss function [4,18]. This is different for black-box audio AEs, since no internal information can be used to guide the generation process. As such, current black-box audio AEs employ the powerful searching capabilities of genetic algorithms to explore a large target space [2,12,22]. However, genetic algorithms are inefficient due to their non-deterministic nature, as they necessitate making many requests to a target ASR system. This may not be feasible in practice, as such suspicious behavior can be used to trigger a threat alert on the targeted system to block further requests.

This paper proposes a method to efficiently generate black-box audio AEs. The generation exploits the gap between the recognition capabilities of humans and machines. In particular, we interpolate two audio signals in the latent space of a variational autoencoder (VAE) [13] to a point at which the ASR model incorrectly transcribes the speech, but humans can still understand it. This is useful in the situation where users do not want an ASR system to automatically eavesdrop on a conversation. This paper discusses our method for generating word-level and sentence-level audio AEs. Our experiments demonstrate that generating audio AEs using the proposed method requires a low number of probing requests to the target ASR system as compared with other existing audio AE methods. In addition, our sentence-level audio AEs can circumvent temporal dependency detection, which is able to efficiently detect state-of-the-art audio AEs [25].

2 Variational Autoencoder

Variational autoencoder (VAE) is a probabilistic generative model [13] that constructs a relationship between random latent variables $z \sim p_\theta(z)$ and observations $x \sim p_\theta(x|z)$, where the prior $p_\theta(z)$ and conditional likelihood $p_\theta(x|z)$ are parameterized by θ. The marginal likelihood $p_\theta(x)$ and posterior $p_\theta(z|x)$ are intractable, as they both require the integral $\int p_\theta(x|z)p_\theta(z)dz$ to be calculated. As a solution, VAE introduces a recognition model $q_\phi(z|x)$ to approximate the posterior $p_\theta(z|x)$. Thus, $\log p_\theta(x)$ can be rewritten as shown in Eq. 1, where $\mathcal{L}(\theta, \phi; x)$ is the variational lower bound to optimize.

The prior $p_\theta(z)$ is assumed to be centered isotropic multivariate Gaussian $\mathcal{N}(z; 0, I)$, where I is an identity matrix. This is because by using a sufficiently complicated function, we can map a set of d normally distributed variables to

any d-dimensional distribution. Besides, both the generative model $p_\theta(x|z)$ and recognition model $q_\phi(z|x)$ are considered as diagonal Gaussian distributions. Neural networks are used to calculate their mean and covariance.

$$\begin{aligned} \log p_\theta(x) &= KL[q_\phi(z|x)||p_\theta(z|x)] + \mathcal{L}(\theta, \phi; x) \\ &\geq \mathcal{L}(\theta, \phi; x) \\ &= -KL[q_\phi(z|x)||p_\theta(z)] + \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] \end{aligned} \tag{1}$$

In practice, VAE introduces the reparameterization trick to make the networks differentiable. First, ϵ is sampled from $\mathcal{N}(0, I)$. Then, sampling z from $\mathcal{N}(\mu, \sigma^2 I)$ is transformed as $z = \mu + \sigma \odot \epsilon$, where $\odot$ represents element-wise product.

3 Problem Definition

This study investigates a method of generating non-targeted word-level and sentence-level audio AEs. In word-level AEs, each audio only contains a single spoken word, whereas sentence-level AEs contain a spoken sentence or phrase with multiple words. The aim is for the target ASR model to incorrectly transcribe these audio AEs, while humans can still understand the speech.

Formally, given a speech waveform x, the ground truth transcription y and the target ASR model $f(.)$, our objective is to generate an AE x' by interpolating x in the latent space of VAE, so that x' is perceived to be similar to x by humans, while y' (i.e. the transcription of x') is different from y, where $y' = f(x')$ and $y = f(x)$.

A successful word-level AE requires the edit distance (also known as Levenshtein distance) between y and y' to be larger than a predefined value. Edit distance refers to the minimum number of operations, including deletions, insertions and substitutions, needed to modify letters of a transcribed AE text to match the ground truth transcription. Similarly, a successful sentence-level AE requires the word error rate (WER) between y and y' to exceeds a predefined value. WER is defined as the total number of operations, including deletions, insertions and substitutions, needed to change words of a transcribed AE sentence to match the ground truth transcription, divided by the number of words in the ground truth transcription.

To quantify the difference in human perception of x and x', we compute the Euclidean distance of log-scaled mel spectrograms between them. This distance gives an indication of the similarity of the audio x and x' to a human. The smaller the distance, the more similar the audio will sound to a human.

Threat Model and Assumptions. The proposed method assumes a black-box threat model, in which an adversary has no knowledge of the internal workings of the target ASR model. To probe the ASR system, an adversary can only input audio into the target ASR model and receive the corresponding transcriptions in text format. No other information is available to the adversary. In addition, we assume an over the line attack. This means that digital files are sent directly to the target ASR system for transcription, as opposed to playing back audio files

over the air through speakers. We also assume that an adversary cannot probe the target system over thousands of times within a short period of time, since this suspicious behavior will be noticed by the system.

The Target ASR Model. To test the proposed method, DeepSpeech [10] was used as the target ASR model due to its state-of-the-art performance. Note that even though we have access to the internal workings of this open source model, the method proposed in this study treats it as a black-box. DeepSpeech version 0.6.1 was used in the experiments, as it was the most recent version at the time, and no language model was deployed, since related research [4,22,24] did not report on the use of any language models.

4 Related Work

Most current research on generating audio AEs assume a white-box threat model. Early work on white-box audio AEs can be found in [6], in which the authors successfully generated non-targeted audio AEs. However, their proposed method performed unsatisfactorily on targeted audio AEs, because target phrases are required to sound similar to the input audio. This limitation was overcome in [4], who examined targeted attacks where the input audio could be transcribed by DeepSpeech to certain target phrases. In their approach, the connectionist temporal classification loss [8] of the target phrase and the perturbed audio was minimized until DeepSpeech produced the desired transcription. In addition, [18] incorporated expectation over transform [3] into the generation process.

In contrast to white-box audio AEs, there is limited work in the area of black-box audio AEs. Previous work was conducted to fool a light-weight keyword spotting model using genetic algorithms [2]. This work was subsequently extended by targeting the DeepSpeech ASR model [22]. However, the methods presented in both of these studies accessed the prediction scores or logits of the targeted model. Such information is not normally available in commercial ASR products. Thus, as asserted in [12] the audio AEs generated in [2] and [22] are not strictly black-box models.

Black-box audio AEs based on resulting transcripts from the target ASR model using multi-objective evolutionary optimization was proposed in [12]. The fitness function adopted in their method contained two objectives: Euclidean distance of Mel-Frequency Cepstral Coefficients (MFCCs) for measuring the similarity of audio samples, and edit distance for measuring the similarity of transcriptions. When generating non-targeted audio AEs, a large edit distance from the original audio is preferrable, while a small edit distance between the AE transcription and the desired transcription is more appropriate in the generation of targeted audio AEs. The generating of black-box audio AEs using genetic algorithms [2,12,22] necessitates making a large number of requests to the target ASR model, due to the non-deterministic property of these algorithms. This unusual behavior can be used by such systems to detect an attack. Although the method proposed in this paper focuses on non-targeted audio AEs, the generation process requires much fewer requests. Recent work in [1] also investigated black-box untargeted audio AEs. They decomposed audio via singular spectrum

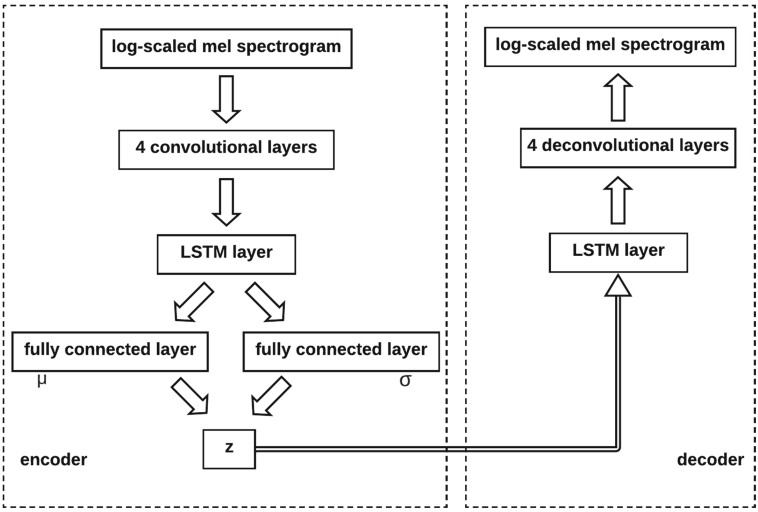


Fig. 1. Conceptual depiction of the VAE architecture.

analysis and discrete Fourier transforms. Audio AEs were then generated by removing components under thresholds.

There has also been an interest in the use of deep probabilistic generative models and VAE [13] in the audio domain. This is because the interpolation of audio in the latent space of VAE can results in meaningful samples. This property was exploited to smoothly interpolate audio samples [11].

5 Proposed Method

Audio Preprocessing. The audio data is first segmented based on word boundaries by forced alignment using corresponding transcriptions. Since we want the generated audio AEs to be similar to the input audio, we utilize log-scaled mel spectrograms as audio features. The commonly used MFCCs in ASR models is not suitable because it discards pitch variation. After alignment, each single-word audio is transformed into mel spectrograms with dimensions $D_t \times D_m$, where D_t is the number of time steps and D_m is the number of mel bands. In our experiments, we set $D_m = 80$ and round D_t down to a multiple of 8.

Model Architecture. In the proposed method, the VAE architecture from Hsu et al. [11] was extended by adding Long Short-Term Memory (LSTM) and extra convolutional layers. The reason for the LSTM layers is because audio data is sequence data, and LSTM is capable of learning temporal dependency in audio. In addition, extra convolutional layers were introduced to further down-sample the audio in the time dimension. In this manner, there are less time steps in the LSTM layers and the training is faster.

A conceptual depiction of the architecture for our VAE is provided in Fig. 1. The VAE is divided into encoder and decoder sections. There are 4 convolutional

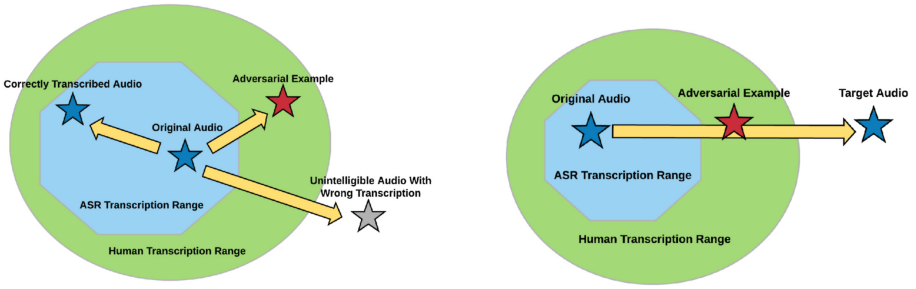


Fig. 2. Generating an audio AE. (a) A successful audio AE should be outside the ASR transcription range, but within the human transcription range. (b) Proposed AE generation method, where an AE is generated by interpolating between an original audio and a target audio in the latent space of VAE.

layers in the encoder. The number of filters for these layers are 64, 128, 256, and 256, respectively. The first convolutional layer has a $1 \times (D_m - 7)$ filter size with a 1×1 stride. The other 3 convolutional layers have a 3×3 filter size with a 2×2 stride. No padding is applied to the first layer, whereas the same padding was applied to the other layers.

The dimensions of the input log-scaled mel spectrogram is $D_t \times D_m$ so after the 4 convolutional layers, the audio is downsampled to $(D_t/8) \times 1 \times 256$. Specifically, $D_t/8$ is the time steps while 1×256 is the dimension of features. Then, a LSTM layer with hidden dimension $D_h = 128$ is applied to this down-sampled audio. The last hidden state with dimension D_h is passed to the Gaussian layer via fully connected layers. The mean value with dimension D_h output from the Gaussian layer is considered as a latent representation of the input audio and the corresponding space with dimension D_h is the latent space.

The decoder section of the architecture mirrors the stages in the encoder. First, a latent representation with dimension D_h is sampled using the mean and covariance output by the Gaussian layer. This sampled latent representation is replicated $(D_t/8)$ times and passed to a LSTM layer with hidden dimension equaling to 256 to reconstruct the $(D_t/8) \times 1 \times 256$ audio. Then, this audio is up-sampled using 4 deconvolutional layers until a log-scaled mel spectrogram with dimension $D_t \times D_m$ is reconstructed. After some experimentation, we found that the model performs well when the convolutional and deconvolutional layers used Leaky ReLU as activation functions, while the tanh activation function was adopted in the LSTM layers.

5.1 Generating Word-Level Adversarial Examples

This section describes our method for generating non-targeted audio AEs at word-level. The purpose of the method is to modify audio that contains a spoken word to a point at which the target ASR model incorrectly transcribes it, while humans can still correctly understand the word.

Figure 2(a) illustrates the concept that was used for generating an audio AE. In the figure, audio data is depicted as being projected into 2-dimensional space. In this space, the closer that two audio points are to each other, the more similar the two audio will sound. The ASR transcription range depicts the space in which speech in audio will be correctly transcribed by that ASR model. Similarly, the human transcription range depicts the space in which humans will correctly understand speech in the audio. The original audio will be within the space of both the ASR and human transcription ranges. A successful AE is defined as one where the projected audio AE point lies outside the ASR transcription range, but within the human transcription range. Speech in the audio is still intelligible to humans, but is incorrectly transcribed by the ASR model.

Black-box AE generation [12,22] makes use of genetic algorithms to randomly modify audio until a successful AE is obtained. The drawback of this non-deterministic method is that it requires the generation of many random samples. Each sample must be sent to the ASR model for transcription to determine whether or not the sample falls within the ASR transcription range. This typically results in a large number of transcription requests being sent to probe the ASR system, which may not be feasible in practice as the shear number of similar transcription requests can alert a system of such an attack.

In contrast, the purpose of our method is to significantly reduce the number of required samples by interpolating between two audio in the latent space of VAE. This is illustrated in Fig. 2(b), where the original audio, which contains a spoken word, is within the ASR and human transcription ranges. Audio which contains a different spoken word is then selected, i.e. the target audio. As the target audio contains a different spoken word, it will definitely lie outside both the ASR and human transcription ranges of the original audio. By interpolating between the original audio and the target audio in the latent space of VAE, the resulting audio will be somewhere in-between the two. The projected location of the resulting audio depends on the interpolation strength. By increasing the interpolation strength, the resulting audio will move closer to the target audio. An audio AE is successfully generated when the resulting audio moves outside the ASR transcription range, but remains within the human transcription range. In this way, the guided approach in our proposed method is more efficient than a genetic algorithm's random sampling.

Formally, let A_W be audio that contains a single spoken word. Our goal is to generate an AE, A'_W, such that A'_W will still be interpreted as A_W by a human, but with $Dist(ASR(A_W), ASR(A'_W)) \geq t$, where $ASR()$ represents the output of the ASR model, $Dist()$ is the function to compute the edit distance between transcribed audio, and t is a predefined threshold that indicates minimum error in the transcription.

Let M_W be the log-scaled mel spectrogram of A_W with dimension $D_t \times D_m$. Let L_W be the latent representation of M_W with dimension D_l. Thus, $L_W = Enc(M_W)$, where $Enc()$ is the encoding function of the VAE, which accepts the log-scaled mel spectrogram as input, and outputs the latent representation with dimension D_l. Let A_T be a target audio that contains a spoken word that is

different from A_W. The procedure for generating A'_W by interpolating between A_W and A_T is described as follows. Let M_T and $L_T = Enc(M_T)$ be the log-scaled mel spectrogram and latent representation of A_T, respectively. The interpolated latent representation: L'_W, is calculated using linear interpolation $L'_W = L_W \times (1-s) + L_T \times s$, where strength s is a real number within $[0, 1]$ that controls the extent of the interpolation. Let $M'_W = Dec(L_{W'})$ be the reconstructed log-scaled mel spectrogram given $L_{W'}$, where $Dec()$ is the decoding function of the VAE, which accepts a latent representation of dimension D_l as input, and outputs the reconstructed log-scaled mel spectrogram with dimension $D_t \times D_m$. A'_W is reconstructed from M'_W by using Griffin-Lim spectrogram inversion [9].

A'_W is considered to be a successful audio AE if the word is incorrectly transcribed, i.e. $Dist(ASR(A_W), ASR(A'_W)) \geq t$. It should be mentioned that increasing the value of s will increase the likelihood that the ASR model will incorrectly transcribe A'_W, because it will increase the difference between the original audio A_W and the modified audio A'_W. However, a successful AE also requires that A'_W sounds similar to A_W from the perspective of human auditory perception, and that a human will interpret the word in A'_W as the same word in A_W. Thus, the value of s should only be as large as it needs to be for the ASR to incorrectly transcribe A'_W. If the resulting A'_W does not satisfy the conditions of a successful AE, either increase the value of s or a different target word can be used for A_T.

A naive way of selecting A_T and s is to sequentially select different audio from a dataset and to gradually increase the value of s. However, this potentially results in a large number of requests to the ASR system. In general, it is better to specify the minimum value of s, and select A_T from a set of different words. Then, only increase the value of s if each word in the set fails to produce a successful AE.

Algorithm 1 in the Appendix details the word-level AE generation procedure that was described above. Given a set of predetermined s values, it finds the minimum value of s that will produce a successfully AE. Let A_{Ti}, where $i \in \{1, 2, ..., n\}$, be a audio set that contain spoken words, and let s_j where $j \in \{1, 2, ..., m\}$, be a set of values for s with increasing strength. The input to the algorithm is an original audio, A_W, the set of target audio, A_{Ti}, the set of interpolation strengths, s_j, and the predefined minimum error threshold, t. For a given set of target audio and interpolation strengths, if a successful AE cannot be found, the algorithm can be repeated on a different set of target audio.

5.2 Generating Sentence-Level Adversarial Examples

This section describes our method for generating sentence-level AEs. An important point to highlight is that simply concatenating the word-level AEs of each word in a sentence, in an attempt to produce a sentence-level AE will not work. This is because state-of-the-art ASR models employ deep learning to learn temporal dependencies in audio. This means that ASR models apply a holistic approach to transcribing an entire sentence, as opposed to independently transcribing each word. In the word-level AE generation approach described above,

temporal dependency does not have to be considered as the ASR model is only supplied with a single word without any context. However, for sentence-level AE generation to be successful, this has to be taken into account.

To describe the sentence-level AE generation method, let A_S be audio containing a sentence of spoken words. Let $(A_{w1}, A_{w2}, ..., A_{wp})$ represent all words in the sentence, and let $(M_{w1}, M_{w2}, ..., M_{wp})$ be the corresponding log-scaled mel spectrograms of the respective words, where p is the total number of words in the sentence. Let $WER()$ be the function to calculate the word error rate, which represents the transcription error in a sentence, and t_S be a predefined value indicating the minimum required transcription error. Our goal of generating a sentence-level AE, $A'_S \equiv (A'_{w1}, A'_{w2}, ..., A'_{wp})$, is to produce an A'_S such that $WER(ASR(A'_S), ASR(A_S)) \geq t_S$, while the words in A'_S are still interpreted by a human as being the same as the words in A_S.

Words in the sentence are sequentially replaced with modified audio, while ensuring that the transcription error increases. To describe the method for generating A'_S, the method starts with $A_S \equiv (A_{w1}, A_{w2}, ..., A_{wp})$, where p is the total number of words in the sentence. The audio of each word A_{wi} in A_S is replaced with a modified audio A'_{wi} in a sequential manner, where $i \in \{1, 2, ..., p\}$. The method for generating A'_{wi} is based on the word-level AE generation method described in the previous section. Given a set of audio that each contain a spoken word that is different from A_{wi}, A_{Tj} where $j \in \{1, 2, ..., n\}$, and a set of interpolation strengths, s_k where $k \in \{1, 2, ..., m\}$, A_{wi} and A_{Tj} are interpolated based on s_k to produce A'_{wi}. The value of s_k is within the range $[0, 1]$, and determines the degree of interpolation between A_{wi} and A_{Tj}.

Let $A_{Si} \equiv (A'_{w1}, ..., A'_{wi}, ..., A_{wp})$ represent a sentence in which the audio of the ith word and all words preceding the ith word, have been replaced with corresponding A'_{wi}. In other words, the audio of each word in the sentence have been sequentially replaced with a modified audio up until the ith word. We consider the inclusion of each A'_{wi} to be successful if $ASR(A_{Si}) \neq ASR(A_{Si-1})$ and $WER(ASR(A_{Si}), ASR(A_S)) \geq WER(ASR(A_{Si-1}), ASR(A_S))$. This means that the modification of the audio for the ith word is deemed to have succeeded if the ASR model produces different transcriptions for A_{Si} and A_{Si-1}, and the transcription error has increased, or is at least the same, with the modification of the ith word. Note that to initialize the transcription error, A_{S0} is set as A_S, hence, initially $WER(ASR(A_{S0}), ASR(A_S)) = 0$. Finally, the generating of the sentence-level AE is deemed to be successful if $WER(ASR(A'_S), ASR(A_S)) \geq t_S$. The procedure for generating sentence-level AE is detailed in Algorithm 2 in the Appendix.

6 Results

In this section, we first show that when compared with the original audio, distortion in the reconstructed audio increases as the interpolation strength increases. We then present results demonstrating word-level and sentence-level audio AEs generated using our method. Finally, we show that the audio AEs generated

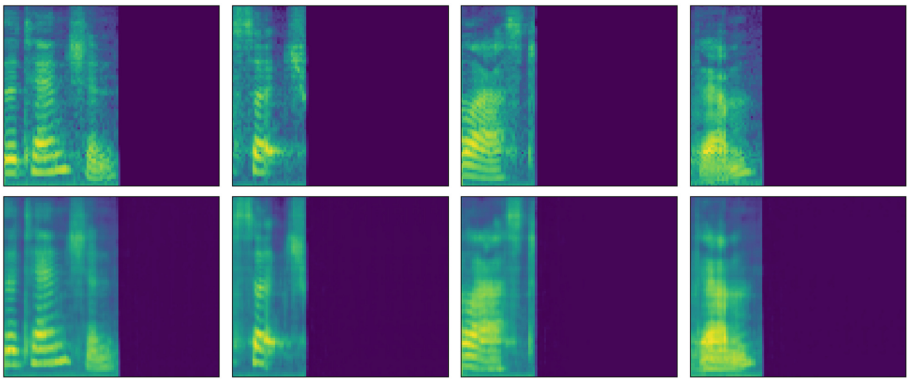


Fig. 3. Reconstructed mel spectrograms without interpolation; first row: mel spectrograms of original audio; second row: the corresponding reconstructed mel spectrograms.

using our proposed method can succeed in thwarting the AE detection method proposed in Yang et al. [25] and transformation defense.

The LibriSpeech [17] dataset was the audio dataset used in the experiments. This dataset contains approximately 1,000 h of read English speech, where the speech duration of a single speaker is usually longer than 20 min. In experiments presented in this paper, all audio data from speaker number 19 in LibriSpeech was used to train the VAE model as well as to generate the AEs. In addition, we used the LibriSpeech alignments produced by Lugosch et al. [15].

The version of DeepSpeech used in experiments is 0.6.1. To improve the performance of DeepSpeech, we deployed the language model, which can be downloaded together with the pre-trained DeepSpeech model.

6.1 Distortion vs Interpolation Strength

A key assumption in our method is that the smaller the interpolation strength, the less distortion produced in reconstructed audio. In this manner, the generation of AEs should use as small an interpolation strength as possible. To investigate the correctness of this assumption, we interpolate randomly selected audio of spoken words with all other words in the dataset using various interpolation strengths. We determined the amount of distortion by calculating the median Euclidean distance between log-scaled mel spectrograms of the original and reconstructed audio.

First, we randomly selected and plotted reconstructed log-scaled mel spectrograms without interpolation to show that the model has been well trained. As shown in Fig. 3, the reconstructed mel spectrogram highly resembles the original ones. This indicates successful training of the model. Then, three randomly selected audio of spoken words were interpolated with the audio of all other spoken words in the dataset. The three words were: “attention”, “such” and “last”.

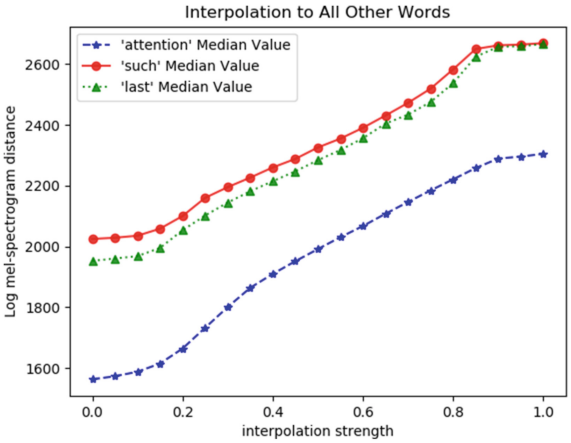
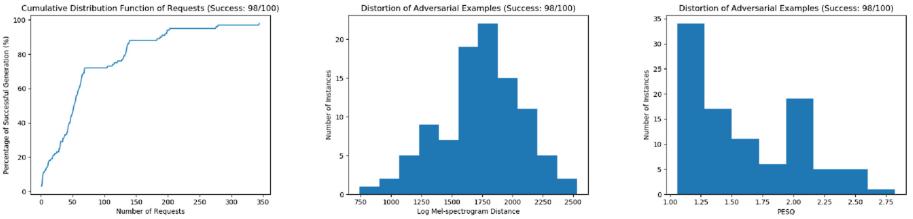


Fig. 4. Example distortion in reconstructed audio as a result of interpolation strength.



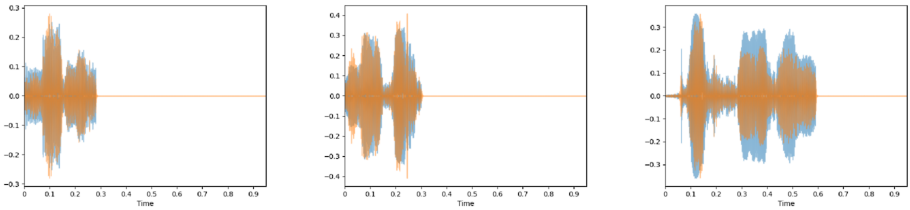
(a) (b) (c)

Fig. 5. (a) CFD of requests made to DeepSpeech. (b) Distribution of distortion in audio AEs based on log-scaled mel spectrogram distance. (c) Distribution of distortion in audio AEs based on PESQ.

The results of this are shown in Fig. 4. We can see that the distortion increases as the interpolation strength increases for all these words. This is intuitive in the sense that audio is distorted more and more as the interpolation strength increases, and with a large enough interpolation strength, the modified audio will eventually be more similar to the target audio than the original audio. Overall, the results presented in Fig. 4 show that smaller interpolation strength leads to less distortion in the reconstructed audio.

6.2 Word-Level Adversarial Examples

To make the non-targeted AEs non-trivial, we generate AEs for spoken words which are correctly transcribed by DeepSpeech. In addition, their corresponding reconstructed audio without interpolation must also be correctly transcribed by DeepSpeech. This is to verify the effectiveness of the proposed method. We applied Algorithm 1 to at most five different subsets of the LibriSpeech [17]



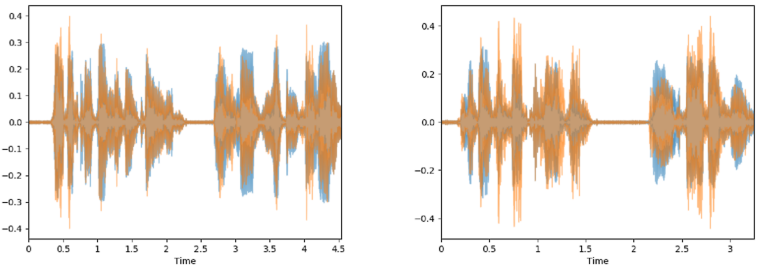
(a) (b) (c)

Fig. 6. Example depicting resulting audio for word-level AEs; original audio (blue), corresponding AE (orange). (a) DeepSpeech could not transcribe the AE for the word "many"; (b) DeepSpeech could not transcribe the AE for the word "never"; (c) The AE for the word "perceiving" was incorrectly transcribed as "curving". (Color figure online)

dataset. Of the key parameters that were used in the experiments, the number of words in A_{Ti} was 10, the values in s_j were {0.017, 0.033, 0.050, 0.067, 0.083, 0.100, 0.117} and the value of the threshold t was set as half the length of the input word.

In Fig. 5(a), we show a cumulative distribution function (CDF) of requests made to DeepSpeech in order to generate 100 AEs. The success rate of AE generation was 98/100, with the majority of AEs successfully generated by making in the order of tens of requests to DeepSpeech. A small number of AEs required requests in the order of hundreds of requests. In comparison, Taori et al. [22] reported that their AE approach requires thousands of iterations on average to generate one audio AE. Hence, our approach achieves its purpose of reducing the number of required requests. Wang et al. [24] pointed out that the method proposed by Khare et al. [12] did not consider success rates and the number of queries. As such, a comparison with that method could not be done adequately. Figure 5(b) and 5(c) in turn show the distortion distribution of the generated AEs. In addition to the distance of log-scaled mel spectrogram, we also measured Perceptual Evaluation of Speech Quality (PESQ), which was proposed as an automatic evaluation metric for measuring speech degradation in the context of telephony [19]. We can see that the log-scaled mel spectrogram distance is centered around 1800, while most PESQ values are between 1.0 to 2.2. It should be noted that we can potentially lower the distortion by making more requests to DeepSpeech. Specifically, we can increase the number of samples in each subset so that a successful AE is more likely to be generated with smaller s.

We present examples showing visual comparisons between original audio and their corresponding audio AEs in Fig. 6. It can be seen from Fig. 6 that waveforms of the audio AEs resemble the waveforms of their original audio. This suggests the acoustic similarity between the AEs and their respective original audio. The AEs either cannot be transcribed by DeepSpeech, or are transcribed as different words. For example, in Fig. 6(a) and 6(b), the AEs for the words "many" and "never" could not be transcribed by the ASR model, whereas in Fig. 6(c), the AE for the word "perceiving" was incorrectly transcribed as "curving".



(a) (b)

Fig. 7. Example depicting resulting audio for sentence-level AEs; original audio (blue), corresponding AE (orange). (a) AE sentence "hated confinement and cleanliness and loved nothing so well as" was transcribed as "kate could find mental cleanliness came loves nothing said well"; (b) AE sentence "she brought herself to read them and though there seemed" was transcribed as "to put herself to madam watho that she". (Color figure online)

6.3 Sentence-Level Adversarial Examples

In this section, we present experimental results obtained from generating sentence-level AEs. As mentioned in Sect. 5.2, simply concatenating word-level AEs into a sentence will not result to a successful sentence-level AE. This is because temporal dependencies in the audio data will help ASR models in correctly transcribing the sentence. In the experiments, the same parameters for word-level AEs generation were used for sentence-level AEs generation.

We selected 30 sentences, which contain at most 10 words, to demonstrate the effectiveness of our method. The results show that the mean WER is 0.75, with the minimum and maximum values equal to 0.4 and 1.0, respectively. If we set the minimal WER for success to 0.7, the success rate is 77%. The mean PESQ value of these 30 audio AEs is 1.37. On average, it needs 38.8 queries for each word in a sentence. It means there are expected to be only 390 queries needed for a sentence with 10 words. This is less than the thousands of queries required by the method in Taori et al. [22].

Two example experiment results are presented in Fig. 7. We can see that the waveforms of these AEs resemble their original audio. Both sentences shown in Fig. 7(a) and 7(b) were incorrectly transcribed with WER both equal to 0.8. It should be mentioned that we did not modify words with a duration of less than 15 ms, e.g., words such as "a" and "an". This is because these short words do not convey important information and they are too short to be successfully modified without overly distorting the resulting audio. Moreover, experiment results show that modifying words before or after such short words are sufficient to cause short words to be incorrectly transcribed. This is also due to temporal dependency in the audio data. In addition, skipping the modification of such short words also reduces the number of requests made to the ASR system.

6.4 Circumventing Temporal Dependency Detection

A recent audio AE detection method based on temporal dependency has been shown to be efficient in detecting state-of-the-art audio AEs [25]. This detection is based on the observation that previous audio AEs generation methods do not preserve temporal dependency. The detection procedure is briefly described as follows. First, only a portion of a spoken sentence, of length k, is fed into the ASR model and transcribed into transcription $T_{\{k\}}$. Then, the whole sentence is fed into the ASR model and transcribed. The portion of the whole sentence transcription that corresponds to length k is extracted as $T'_{\{k\}}$, and compared with $T_{\{k\}}$. An audio AE is detected if $T_{\{k\}}$ is significantly different from $T'_{\{k\}}$.

Since word-level AEs do not have temporal dependency, we only discuss how our sentence-level AEs circumvent temporal dependency detection. This is because in our proposed method, every time a word in the sentence is modified our method computes the error of the whole sentence, not just a portion of the sentence. Hence, temporal dependency in the modified sentence is preserved.

To demonstrate this, we extract various portions from the original audios and audio AEs in the same way as in [25], e.g. $k = \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \frac{5}{6}$. We then send all of these sub-sentences to be transcribed by the ASR system to see whether temporal dependency is preserved. Table 3 in the Appendix presents an example demonstrating this. It can be seen that the generated AE has a similar temporal dependency property as the original audio, because the preceding transcribed words are the same even when the number of words in the sub-sentence is increased. This indicates that the detection method proposed in [25] will not work on the sentence-level AEs generated using our proposed method, because no matter the value of k in $T_{\{k\}}$, the result of $T'_{\{k\}}$ will be consistent with $T_{\{k\}}$.

We implement the temporal detection method to detect the 30 sentence-level AEs generated in Sect. 6.3. The results in Table 1 suggest that the sentence-level AEs can successfully circumvent temporal dependency detection due to low AUC values.

6.5 Robustness Against Transformation

In addition to temporal dependency, preprocessing input audio using various transformations has also been shown to be an effective defense against audio AEs [5]. We conducted experiments to determine whether transformation would recover the original transcripts of our 30 sentence-level audio AEs. The following transformations were considered: Gaussian noise with different standard deviation, filtering by high pass filters with different cutoff frequencies and resampling the input audio at different rates. Experimental results are shown in Table 2. As mentioned above, the mean WER of our 30 sentence-level audio AEs is originally 0.75. We can see that the transformation cannot recover the original transcripts of the audio because the average WER only decreases slightly.

Table 1. Detection of our audio AEs through temporal dependency.

	WER	CER	LCP
$k = \frac{1}{2}$	0.728	0.735	0.593
$k = \frac{2}{3}$	0.728	0.716	0.593
$k = \frac{3}{4}$	0.567	0.557	0.589
$k = \frac{4}{5}$	0.592	0.623	0.644
$k = \frac{5}{6}$	0.629	0.635	0.638

Table 2. Robustness against transformation

	Gaussian noise			High pass filtering			Resampling		
	1e−4	1e−3	1e−2	200 Hz	300 Hz	400 Hz	8000	10000	12000
Mean WER	0.72	0.67	0.69	0.61	0.60	0.68	0.74	0.75	0.75

7 Limitations and Future Work

There are two main limitations in our proposed method. First, our method heavily relies on techniques to accurately align the lengths of spoken words in audio. Although alignment of speech is a well studied problem [15], it would be more efficient in future work if unaligned speech could be used directly. Another limitation of our proposed method is that some generated AEs may not be easily interpreted by a human due to the noise introduced in the resulting AE. This is a consequence of the black-box model, because unlike a white-box model, a black-box model assumes no knowledge of the internals workings of an ASR system. In future work, we will incorporate psychoacoustics and the use of a vocoder, such as Wavenet [23], to increase the quality of the reconstructed audio.

Furthermore, VAE trained in this paper is only based on a single speaker. It should be mentioned that VAE can be speaker-independent by integrating speech from various speakers in the training set. However, this would result in significantly more effort to successfully train a VAE that can produce clear speech. The complexity of the proposed architecture may also increase. We leave speaker-independent VAE as an investigation for future work.

8 Conclusion

This paper presents a novel black-box non-targeted audio AE generation method. The proposed method makes use of a VAE model to produce black-box audio AEs. This black-box generation process relies solely on transcription results produced by an ASR model, without requiring any information about the internal workings of the ASR model. Methods for generating both word-level AEs and sentence-level AEs are described. In addition, this paper presents experiment results demonstrating that our method can successfully generate audio

AEs using a smaller number of requests to an ASR system as compared with other methods that rely on the use of non-deterministic genetic algorithms.

Appendix

Table 3. Example of circumventing temporal dependency detection.

k	Original audio transcription	AE transcription
$\frac{1}{2}$	Her mother was a woman of	It mother let alone
$\frac{2}{3}$	Her mother was a woman of useful	It mother let alone useful
$\frac{3}{4}$	Her mother was a woman of useful plan	It mother let alone useful
$\frac{4}{5}$	Her mother was a woman of useful plans	It mother let alone useful and
$\frac{5}{6}$	Her mother was a woman of useful plain	It mother let alone useful in
1	Her mother was a woman of useful plain sense with	It mother let alone useful insense what

Algorithm 1. Word-level AE generation

Input: original audio, A_W; a set of target audio, A_{Ti} (where $i \in \{1, 2, ..., n\}$); a set of interpolation strengths, s_j (where $j \in \{1, 2, ..., m\}$); and a minimum edit distance, t

Output: word-level AE, A'_W

For each s_j, where $j \in \{1, 2, ..., m\}$, do
  For each A_{Ti}, where $i \in \{1, 2, ..., n\}$, do
    $M_W \leftarrow$ log-scaled mel spectrogram for A_W
    $M_T \leftarrow$ log-scaled mel spectrogram for A_{Ti}
    $M'_W \leftarrow$ interpolate between M_W and M_T by s_j
    $A'_W \leftarrow$ reconstruct audio from M'_W using
        Griffin-Lim spectrogram inversion
// check whether AE generation succeeded (i.e. if resulting
// edit distance above threshold)
    If $Dist(ASR(A_W), ASR(A'_W)) \geq t$
      return A'_W
    End If
  End For
End For
// if AE generation unsuccessful, use a different set of
// target audio

Algorithm 2. Sentence-level AE generation

Input: original audio containing a sentence, A_S; a set of target audio, A_{Tj} (where $j \in \{1, 2, ..., n\}$); a set of interpolation strengths, s_k (where $k \in \{1, 2, ..., m\}$)
Output: sentence-level AE, A'_S

WordLoop:
For each word A_{wi} in sentence A_S do
  For each s_k, where $k \in \{1, 2, ..., m\}$, do
    For each A_{Tj}, where $i \in \{1, 2, ..., n\}$, do
      $M_{wi} \leftarrow$ log-scaled mel spectrogram of A_{wi}
      $M_{Tj} \leftarrow$ log-scaled mel spectrogram of A_{Tj}
      $M'_{wi} \leftarrow$ interpolate between M_{wi} and M_{Tj} by s_k
      $A'_{wi} \leftarrow$ reconstruct audio from M'_{wi} using
          Griffin-Lim spectrogram inversion
// check whether the transcription of the modified
// sentence is different from the previous sentence
// and that the error has not decreased
      If $ASR(A_{Si}) \neq ASR(A_{Si-1})$ and
        $WER(ASR(A_{Si}), ASR(A_S)) \geq$
          $WER(ASR(A_{Si-1}), ASR(A_S))$
// save the modified sentence, and proceed to modify
// the next word in the sentence
          $A'_S \leftarrow A_{Si}$
          goto WordLoop
      End If
    End For
  End For
End For
// verify the sentence-level AE
If $WER(ASR(A'_S), ASR(A_S)) \geq t_S$
  return A'_S
End If

References

1. Abdullah, H., et al.: Hear "no evil", see "kenansville": efficient and transferable black-box attacks on speech recognition and voice identification systems. In: 2021 IEEE Symposium on Security and Privacy (SP), Los Alamitos, CA, USA, May 2021, pp. 142–159. IEEE Computer Society (2021)
2. Alzantot, M., Balaji, B., Srivastava, M.B.: Did you hear that? Adversarial examples against automatic speech recognition. CoRR, abs/1801.00554 (2018)
3. Athalye, A., Engstrom, L., Ilyas, A., Kwok, K.: Synthesizing robust adversarial examples. In: Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018, pp. 284–293 (2018)
4. Carlini, N., Wagner, D.A.: Audio adversarial examples: targeted attacks on speech-to-text. In: 2018 IEEE Security and Privacy Workshops, SP Workshops 2018, San Francisco, CA, USA, 24 May 2018, pp. 1–7 (2018)

5. Chen, G., et al.: Who is real bob? Adversarial attacks on speaker recognition systems. CoRR, abs/1911.01840 (2019)
6. Cissé, M., Adi, Y., Neverova, N., Keshet, J.: Houdini: fooling deep structured visual and speech recognition models with adversarial examples. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017, pp. 6977–6987 (2017)
7. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015, Conference Track Proceedings (2015)
8. Graves, A., Fernández, S., Gomez, F.J., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, 25–29 June 2006, pp. 369–376 (2006)
9. Griffin, D., Lim, J.: Signal estimation from modified short-time Fourier transform. IEEE Trans. Acoust. Speech Signal Process. **32**(2), 236–243 (1984)
10. Hannun, A.Y., et al.: Deep speech: scaling up end-to-end speech recognition. CoRR, abs/1412.5567 (2014)
11. Hsu, W., Zhang, Y., Glass, J.R.: Learning latent representations for speech generation and transformation. In: Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, 20–24 August 2017, pp. 1273–1277 (2017)
12. Khare, S., Aralikatte, R., Mani, S.: Adversarial black-box attacks for automatic speech recognition systems using multi-objective genetic optimization. CoRR, abs/1811.01312 (2018)
13. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. In: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 14–16 April 2014, Conference Track Proceedings (2014)
14. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017, Workshop Track Proceedings (2017)
15. Lugosch, L., Ravanelli, M., Ignoto, P., Tomar, V.S., Bengio, Y.: Speech model pre-training for end-to-end spoken language understanding. CoRR, abs/1904.03670 (2019)
16. Moosavi-Dezfooli, S., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017, pp. 86–94 (2017)
17. Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: LibriSpeech: an ASR corpus based on public domain audio books. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, 19–24 April 2015, pp. 5206–5210 (2015)
18. Qin, Y., Carlini, N., Cottrell, G.W., Goodfellow, I.J., Raffel, C.: Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In: Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, California, USA, 9–15 June 2019, pp. 5231–5240 (2019)
19. Rix, A.W., Beerends, J.G., Hollier, M.P., Hekstra, A.P.: Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2001, Salt Palace Convention Center, Salt Lake City, Utah, USA, 7–11 May, 2001, Proceedings, pp. 749–752. IEEE (2001)

20. Schönherr, L., Kohls, K., Zeiler, S., Holz, T., Kolossa, D.: Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding. In: 26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, 24–27 February 2019 (2019)
21. Szegedy, C., et al.: Intriguing properties of neural networks. In: 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 14–16 April 2014, Conference Track Proceedings (2014)
22. Taori, R., Kamsetty, A., Chu, B., Vemuri, N.: Targeted adversarial examples for black box audio systems. In: 2019 IEEE Security and Privacy Workshops, SP Workshops 2019, San Francisco, CA, USA, 19–23 May 2019, pp. 15–20 (2019)
23. van den Oord, A., et al.: WaveNet: a generative model for raw audio. In: The 9th ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA, 13–15 September 2016, p. 125 (2016)
24. Wang, Q., Zheng, B., Li, Q., Shen, C., Ba, Z.: Towards query-efficient adversarial attacks against automatic speech recognition systems. IEEE Trans. Inf. Forensics Secur. **16**, 896–908 (2021)
25. Yang, Z., Li, B., Chen, P., Song, D.: Characterizing audio adversarial examples using temporal dependency. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019 (2019)