Weighted-Sampling Audio Adversarial Example Attack

Xiaolei Liu,[1] **Kun Wan,**[2] **Yufei Ding,**[2] **Xiaosong Zhang,**[1*] **Qingxin Zhu**[1]
[1]University of Electronic Science and Technology of China, [2]University of California, Santa Barbara
luxaole@gmail.com, {kun, yufeiding}@cs.ucsb.edu, {johnsonzxs, qxzhu}@uestc.edu.cn

Abstract

Recent studies have highlighted audio adversarial examples as a ubiquitous threat to state-of-the-art automatic speech recognition systems. Thorough studies on how to effectively generate adversarial examples are essential to prevent potential attacks. Despite many research on this, the efficiency and the robustness of existing works are not yet satisfactory. In this paper, we propose *weighted-sampling audio adversarial examples*, focusing on the numbers and the weights of distortion to reinforce the attack. Further, we apply a denoising method in the loss function to make the adversarial attack more imperceptible. Experiments show that our method is the first in the field to generate audio adversarial examples with low noise and high audio robustness at the minute time-consuming level [1].

Introduction

In recent years, machine learning algorithms are widely used in various fields. However, studies show that existing learning-based algorithms are vulnerable to adversarial attacks (Szegedy et al. 2013; Goodfellow, Shlens, and Szegedy 2014). Currently, majority of the research on adversarial examples are in the image recognition field (Kurakin, Goodfellow, and Bengio 2016; Carlini and Wagner 2017; Chen et al. 2018; Su, Vargas, and Sakurai 2019), while others investigate fields such as text classification (Jia and Liang 2017), traffic classification (Liu et al. 2018), and malicious software classification (Grosse et al. 2016; Hu and Tan 2017; Liu et al. 2019).

Automatic speech recognition (ASR) is another vital field where machine learning algorithms are also frequently applied (Hinton et al. 2012). To date, it has been proved that audio adversarial examples can mislead ASR to transfer any audio to any targeted phrases (Carlini and Wagner 2018). However, it is much more difficult to generate adversarial examples for audio than images. To generate an effective audio adversarial example, there are still several technical challenges to be addressed:

(C1) Generating audio adversarial examples demands significant computational resources and huge time overhead. It takes over one hour or more to generate an effective audio adversarial example by recently proposed approaches (Carlini and Wagner 2018; Kreuk et al. 2018; Yuan et al. 2018; Qin et al. 2019). Such inefficiency significantly undermines the practicability of the attack.

(C2) Recording and replaying, which are common operations for audio, could easily introduce extra noise. Therefore, the robustness of adversarial examples against noise is crucial. Nevertheless, the adversarial examples prepared over hours are still poor in robustness. The state-of-the-art audio adversarial examples (Carlini and Wagner 2018; Alzantot, Balaji, and Srivastava 2018) become invalid after adding imperceptible pointwise random noise.

(C3) Different from the image domain where l_p-based metrics are carefully studied as a part of the loss function to generate adversarial examples, there are no investigations on which kind of metric is more suitable for constructing audio adversarial examples.

In this paper, we achieve a fast, robust adversarial example attack to ASR by proposing two novel techniques named **Weighted Perturbation Technology (WPT)** and **Sampling Perturbation Technology (SPT)**.

WPT adjusts the weights of distortion at different positions of audio during the generation process, and thus generates adversarial examples faster and improves the attack efficiency (addressing C1).

Meanwhile, by reducing the number of points to perturb based on the characteristics of context correlation in the speech recognition model, SPT can increase the robustness of audio adversarial examples (addressing C2).

To best of our knowledge, we are the first in the field to both take the factors of the weights and the numbers of perturbed points into consideration during the generation of audio adversarial examples. And the two techniques are always complementary to all existing ASR adversarial attacks,

*Correspondence author is Xiaosong Zhang. This research was supported by National Key R&D Program of China (2017YFB0802900), National Natural Science Foundation of China (61572115,61902262), Sichuan Science and Technology Program (2019JDRC0069).

[1]We encourage you to listen to these audio adversarial examples on this website: https://sites.google.com/view/audio-adversarial-examples/.

which by default modify every value of the entire audio vector.

Further, we also investigate different metrics as parts of the loss function to generate audio adversarial examples and provide a reference for future researchers in this field (addressing C3).

Finally, our experiments show that our method can generate more *robust* audio adversarial examples in a short period of *4 to 5 minutes*. This is a substantial improvement compared to the state-of-the-art methods.

Related Work

Audio adversarial example attacks can be mainly divided into two categories, speech-to-label, and speech-to-text (Yang et al. 2018). Speech-to-label classifies audio into different categories and the output is a specific label. This method is inspired by a similar method on images (Alzantot, Balaji, and Srivastava 2018; Cisse et al. 2017). Since the target phrases can only be chosen from a certain amount of labels, the practicality of such a method is limited.

The speech-to-text method directly converts audio semantic information into text. Carlini & Wagner (Carlini and Wagner 2018) are the first to work on audio adversarial examples for the speech-to-text models and they can let ASR transcribe any audio into a pre-specified text. However, the audio robustness is compromised and most of their examples will lose the adversarial labels by adding imperceptible random noise.

Later on CommanderSong (Yuan et al. 2018) achieved practical over-the-air audio adversarial attacks, but they only validated their method on the music clips. Additionally Yakura & Sakuma (Yakura and Sakuma 2018) proposed another physical-world attack method. Regardless, these two methods will introduce non-negligible noise to the original audio. Unfortunately, all of these methods would require several hours to generate only one audio adversarial example, including the most recent work (Qin et al. 2019).

To the best of our knowledge, there is no method to generate audio adversarial examples with low noise and high robustness at the minute level. Our proposed method can be applied with all these current methods to achieve a trade-off among quality, robustness and convergence speed.

Background

Threat Model. Before digging into details of the audio adversarial example attack, an ASR model should be selected as the potential threat model. Following the common practice in the field we summarize three basic requirements for it:

- Its core component should be Recurrent Neural Networks (RNNs) such as LSTM (Hochreiter and Schmidhuber 1997), which is widely adopted in current ASR systems;
- It is vulnerable to the state-of-the-art audio adversarial attack methods, and the corresponding results could be used as baselines in our experiments;
- It has to be open-source and thus we can directly conduct white-box tests on it.

Given requirements above, we choose the speech-to-text model, Deepspeech (Hannun et al. 2014), as our experimental threat model, which is an open-source ASR with Connectionist Temporal Classification (CTC) method (Graves et al. 2006) and LSTM as its main components. Notice that our approach can be also applied to other RNN-based ASR systems.

Considering that there are many ways to convert the black-box model to a white-box model (Papernot, McDaniel, and Goodfellow 2016; Oh et al. 2017; Ilyas et al. 2018), which is another research direction, and most of the previous work also assume they know the parameters of models, hence our research is also based on the white-box model.

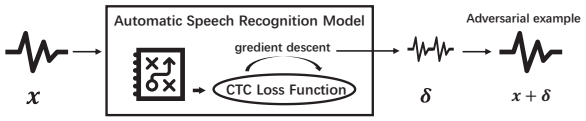

Figure 1: General process of audio adversarial example attack.

Audio Adversarial Examples. Figure 1 shows the general process of audio adversarial example attack. Specifically, let x be the input audio vector and δ is the distortion to the original audio. Audio adversarial example attacks are defined as by adding some perturbations δ, ASR recognizes $x+\delta$ as specified malicious texts t (formally: $f(x+\delta)=t$), while there is no perceivable difference for humans. The process of generating adversarial examples can be regarded as a process of updating x using gradient descent on a predefined loss function $\ell(\cdot)$ shown in Eq. 1. The iterative process stops until the adversarial example meets our evaluation requirements.

$$\begin{aligned}\ell(x,\delta,t) = \ell_{model}(f(x+\delta),t) \\ + c\cdot\ell_{metric}(x,x+\delta)\end{aligned} \tag{1}$$

In Eq. 1, ℓ_{model} is the loss function used in the ASR models. For example, Carlini & Wagner (Carlini and Wagner 2018) uses CTC-loss as the ℓ_{model}. ℓ_{metric} is used to measure the difference between the generated adversarial examples and the original samples. Different from the image domain where l_p-based metrics are commonly used, there is no consensus on which ℓ_{metric} should be applied in the audio field. For instance, so far various ℓ_{metric} such as SNR (Yuan et al. 2018), psychoacoustic hearing thresholds (Schönherr et al. 2018) and frequency masking (Qin et al. 2019) have been adopted. We will also elaborate the choices of l_{metric} in this paper.

Evaluation Metric. Based on the characteristics of the audio and the common practice in the field, the following evaluation metrics are chosen in this paper.

- **SNR**(Signal-to-noise ratio) measures the noise level of the distortion δ relative to the original audio x. The smaller distortion is, the larger SNR will be,

$$\text{SNR} = 10\log_{10}\frac{P_x}{P_\delta}, \tag{2}$$

where P_x and P_δ represent the energies of the original audio and the noise respectively.

- **WER**, i.e., the word error rate, is a common evaluation metric in the ASR domain,

$$\text{WER} = \frac{S + D + I}{N} \times 100\%, \tag{3}$$

where S, D and I are the numbers of substitutions, deletions and insertions respectively, and N is the total number of words.

- **Success Rate** is the ratio of examples which can be successfully recognized as the malicious target texts by ASR,

$$\text{Success Rate} = \frac{N_{adv}}{N_{total}} \times 100\%, \tag{4}$$

where N_{adv} is the number of adversarial examples that can be transcribed as target phrases and N_{total} is the total number of adversarial examples generated.

- **Robustness Rate**. Adding noise to the audio x is the same as applying transformation function $t \sim \mathcal{T}$ over the input x. Here we define the robustness rate as the success ratio of examples that can still retain adversarial property after transformed by $t(\cdot)$,

$$\text{Robustness Rate} = \frac{N_{t(adv)}}{N_{total}} \times 100\%, \tag{5}$$

where $N_{t(adv)}$ is the number of adversarial examples that can still be transcribed as target phrases after transformed by $t(\cdot)$.

Methodology

In this section, first, we will show the details of sampling perturbation technology and weighted perturbation technology. We will also explain why these methods are able to increase the robustness of adversarial examples and accelerate the attack. Finally, we will investigate different metrics and try to find out an experiential standard to refer to, instead of directly using the l_p-based metrics on the image domain.

Sampling perturbation technology

We propose SPT to increase the robustness of audio adversarial examples. It works by reducing the number of perturbed points. Here we will explain the reason why SPT works, taking the CTC loss as an example. Actually it's a general method for current audio adversarial attacks.

We use x denote an audio vector, p denotes a phrase which is the semantic information of x and y denotes the probability distribution of x decoded to p. x_i is one frame of x and y^i is the probability distribution over the character which is transformed by x_i.

In CTC process (shown in Figure 2 left), the process from x to p is: Input x (Step 1) and get the sequences of tokens π (Step 2). Then merge the repeated characters and drop '-' tokens (Step 3). Output the predicted phrase p (Step 4).

Because π is the sequence of tokens to x, we say the probability of π under y is the product of the likelihoods of each $y^i_{\pi^i}$. For a given phrase p with respect to y, there will be a set of predicted sequences $\pi \in \prod(p, y)$. Finally, we calculate $\Pr(p|y)$, the probability of phrase p under the distribution y, by summing the probability of each π in the set:

$$\Pr(p|y) = \sum_{\pi \in \prod(p,y)} \prod_{i=0}^{n} y^i_{\pi^i} \tag{6}$$

In traditional audio adversarial example attack, if we want to transcribe audio x to target t, we will add slight distortion on each π^i to let $t = \arg\max_p \Pr(p|y)$. However, we can also get the same result by fixing part of $\prod_j^{n-m} y^j_{\pi^j}$ and perturbing the other part to let $\prod_k^m y^k_{\pi^k} = \prod_k^m y'^k_{\pi'^k}$, where y'^k is the new probability distribution of perturbed π'^k and $y^k_{\pi^k} \neq y'^k_{\pi'^k}$:

$$\begin{aligned} t &= \arg\max_p \Pr(p|y) \\ &= \arg\max_p \sum_{\pi \in \prod(p,y)} \prod_{i=0}^{n} y^i_{\pi^i} \\ &= \arg\max_p \sum_{\pi' \in \prod(p,y')} \prod_{j=0}^{n-m} y^j_{\pi^j} \prod_{k=0}^{m} y'^k_{\pi'^k} \end{aligned} \tag{7}$$

Based on Formula 7, we can shorten the perturbed number of audio vector from n to m. Our evaluations give the support that m can be much smaller than n.

Since most of the points in our adversarial examples are exactly the same as those in the original audio, this makes our adversarial examples show very similar properties to the original audio. Compared with the adversarial examples that all points are perturbed, environmental noise has a lower probability of affecting the SPT-based adversarial examples.

Athalye et al. (Athalye et al. 2017) proposed the Expectation Over Transformation (EOT) algorithm to construct adversarial examples that are able to maintain the adversarial property over a chosen transformation distribution $\mathcal{T}$. Unfortunately, the limitation of the EOT is that it only increases robustness under the same or similar $\mathcal{T}$-distribution noise. Without the assumption of similar distribution, the adversarial property will be largely compromised. As a comparison, our method does not need to have prior knowledge regarding the distribution when generating adversarial examples, thus we could have better general robustness. Meanwhile, our method is complementary to EOT.

Weighted perturbation technology

WPT can reduce the time cost by adjusting the weights of distortion in a different position. We first point out the limitations of traditional loss function $\ell(\cdot)$ (Eq. 1) and then give our solution. (Again, we introduce WPT based on CTC sequence loss and WPT is a general method and can be easily applied to attack other ASR systems.)

Current Problem. By analyzing the process of generating audio adversarial examples, we found that the closer the currently transcribed phrase p' is to the target text t, the

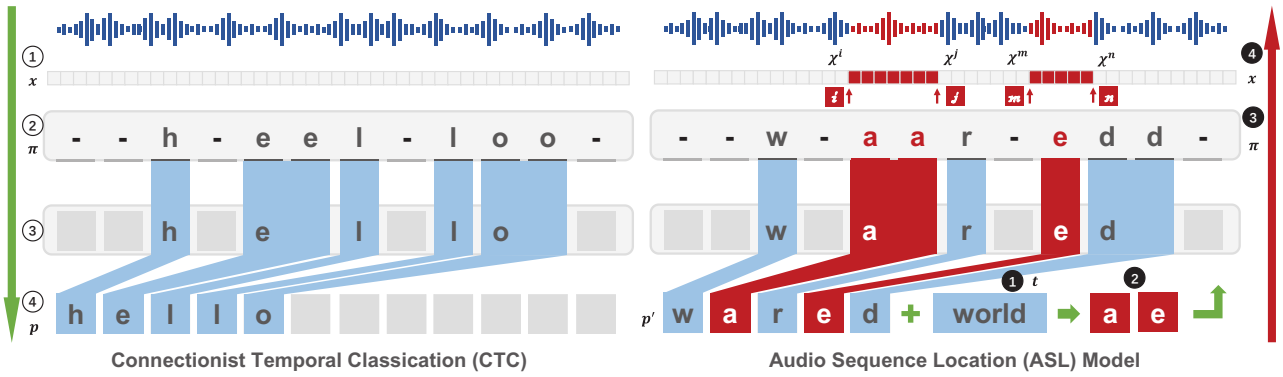

Figure 2: Overview of CTC and ASL.

longer it takes. In order to divide this process into different stages, we introduce the Levenshtein Distance (Levenshtein 1966), which is a string metric for measuring the minimum number of single-character edits (i.e. insertions, deletions or substitutions) required to change one string into the other. According to our statistics, the average percentage of time loss spent on the Levenshtein distance from 3 to 2, 2 to 1 and 1 to 0 are, respectively, 7.52%, 15.43%, and 32.16%. Their sum exceeds 55% of the generation time. The reason for spending a lot of time at these stages is that when Levenshtein Distance is small, most of the current points no longer need to be perturbed, except for those points which cause the Levenshtein Distance not to be 0. We name these points as key points.

On the one hand, if we can give these key points larger weights, the time spent at this stage will be reduced; on the other hand, if the global search step size can be reduced with the number of iterations, then we can avoid missing a more perfect adversarial example due to over perturbing. These two aspects will make the overall speed be accelerated.

Steps of WPT: Accordingly, we implement WPT in two steps. The **first step** focuses on shortening the time cost when Levenshtein Distance equals to 1 by increasing the weights of key points. Therefore, we need to know which points are key points.

Audio Sequence Location(ASL) is a model to help us locate these key points in the audio. As shown in Figure 2 (right), the inputs of ASL are current transcribed phrase p' and target t (Step 1). After comparing p' and t, we get the different characters (Step 2). Find the positions of these characters in the sequence of tokens π (Step 3). Output the intervals set χ^k in audio vector x (Step 4). Finally, the distortion corresponding to these k positions in χ^k are multiplied by weights ω. Our improved formulation of $\ell(\cdot)$ is,

$$\ell(x,\delta,t) = \ell_{model}(f(x+\alpha\cdot\delta),t) + c\cdot\ell_{metric}(x,x+\delta),$$

$$\alpha_i = \begin{cases} \omega, & \text{if } i \in \chi^k \\ 1, & \text{else} \end{cases}, \ \omega > 1, \tag{8}$$

where α is a weights vector to δ, and if the vector subscript i belongs to the intervals set χ^k, we give these key points bigger weights ω.

Besides, when we shorten Levenshtein Distance to 0, WPT goes to its **second step** to reduce the learning rate lr:

$$lr \leftarrow \beta \cdot lr, \tag{9}$$

where constant β satisfies $\beta \in (0,1)$. After updating lr, we can calculate the perturbations δ on each iteration:

$$\delta_0 = 0, \ \ \delta_{n+1} \leftarrow \delta_n - lr \cdot sign(\nabla_\delta \ell(x,\delta,t)), \tag{10}$$

where $\nabla_\delta \ell(x,\delta,t)$ is the gradient of ℓ with respect to δ.

Advantages: Carlini&Wagner try to set different weights to each character of the sequence of π to solve this problem (Carlini and Wagner 2018). Actually it will cost prohibitive computation to find the most suitable weight for each character. So, they have to get a feasible solution x_0 which is found by using the normal CTC-loss function first and then using their improved method based on x_0. However, this is not a perfect solution to solve the problem mentioned before. There are three advantages to our WPT:

1. Their method has to find a feasible solution x_0 first, which means they can not shorten the time cost before generating a successful adversarial example. This period time accounts for more than 55% of the total time. We can use ASL at any iterations to get the key location intervals χ^k without having to obtain x_0 first. Then we make converge faster by adjusting the weight ω of δ.

2. WPT is effective against both a greedy decoder and beam-search decoder (Graves et al. 2006), which are two searching ways combined with CTC to obtain the alignment π, while their method is only effective against greedy decoder. The reasons are **a)** Instead of adjusting the weight of a single character or token, we adjust the weights of a continuous interval on the audio vector corresponding to the character. This distortion based on the continuous interval is effective for beam-search decoder. **b)** WPT updates weights ω according to the current alignment π instead of a fixed π_0. So our method won't be limited to the greedy decoder.

3. The learning rate lr, that gradually decreases as the distortion δ is reduced, can help us avoid the problem of excessive perturbations due to too long steps so that better adversarial examples can be found more quickly.

Table 1: Evaluation of our adversarial attack with Commander Song and C&W's attack.

Attack Approach	Target phrase	Proportion ↓	Efficiency(s) ↓	Success Rate ↑	$dB_x(\delta)\uparrow$ **	SNR ↑
Our attack	Random phrases*	**75%**	**251**	**1**	**46.92**	**31.9**
C&W's attack	Random phrases*	all points	≈3600	**1**	38	-**
CommanderSong	echo open the front door	all points	3600	**1**	-**	17.2
	okay google restart phone now	all points	4680	**1**	-**	18.6

* As is selected in C&W's work: target phrase is chosen at random such that (a) the transcription is incorrect (b) it is theoretically possible to reach that target.

** $dB_x(\delta)$ is a l_∞ metric defined by C&W (Carlini and Wagner 2018). And'-' means no relevant data was provided in their papers. '↑' means the bigger the better.

Investigation of metrics

As for ℓ_{metric}, which is the other part of $\ell(\cdot)$, also plays an important role in the generation of adversarial audio. Different from the image domain where mainly l_p-based metrics are used as ℓ_{metric}, there is no study on which metric should be selected.

The purpose of ℓ_{metric} is to limit the difference between the adversarial examples and the original samples. Therefore, we introduce the Total Variation Denoising (TVD) to reduce the noise perturbed and let adversarial examples sound more like the original audio. TVD is based on the principle that signals with excessive and possibly spurious detail have high total variation and is mostly used in the process of noise removal (Rudin, Osher, and Fatemi 1992). After the TVD process, we can remove most of the impulse in the adversarial examples and make the distortion more imperceptible. The ℓ_{metric} based on TVD can be calculated via the sum of closeness $E(\delta)$ and total variation $V(x+\delta)$:

$$\begin{aligned}\ell^{tvd}_{metric}(x,\delta) &= E(\delta) + \gamma \cdot V(x+\delta) \\ &= \frac{1}{n}\sum_{i=0}^{n}(\delta_i)^2 + \gamma \cdot \sum_{j=1}^{n-1} |(x_{j+1} \\ &+ \delta_{j+1}) - (x_j + \delta_j)|,\end{aligned} \tag{11}$$

where γ is a trade off adjusted of $E(\delta)$ and $V(x+\delta)$. Besides, we also investigate other three types of similarity metrics which are selected in terms of 1) l_∞ in image domain; 2) l_2-based in current audio domain; and 3) cosine distance in information retrieval domain; as shown in Formula 12.

$$\begin{aligned}\ell^1_{metric}(x,\delta) &= l_\infty(x, x+\delta) \\ \ell^2_{metric}(x,\delta) &= l_2(x, x+\delta) \\ \ell^3_{metric}(x,\delta) &= (1 - cor(x, x+\delta))\end{aligned}, \tag{12}$$

where $l_\infty(\cdot)$, $l_2(\cdot)$ and $cor(\cdot)$ are, respectively, the measurement of l_∞ distance, l_2 distance and cosine distance between two audio vectors.

A good choice of ℓ_{metric} not only accurately reflects the auditory difference between the two audio frequencies but also avoids the optimization process oscillating around a solution without converging (Carlini and Wagner 2017). We will give a comparison of the effects of various loss functions in the experimental section.

Experimental results

In this section, we show the evaluation of our adversarial attack used the technologies introduced in the Methodology Section. We also study the performance of different ℓ_{metric} on success rate, SNR and $dB_x(\delta)$. Our experimental results show that our approach has faster generation speed, better SNR, higher success rate, and stronger robustness than other attacks.

Dataset and experimental settings

Dataset. Mozilla Common Voice dataset[2] (MCVD): MCVD is an open and publicly available dataset of voices that everyone can use to train speech-enabled applications. It consists of voice samples require at least 70GB of free disk space. We follow the convention in the field and use the first 100 test instances of this dataset to generate audio adversarial examples. **Unless otherwise specified, all our experimental results are averaged over these 100 instances.**

Environment. All experiments are carried out on an Ubuntu Server (16.04.1) with an Intel(R) Xeon(R) CPU E5-2603 @ 1.70GHz, 16G Memory and GTX 1080 Ti GPU.

Experiments

Evaluating adversarial examples In order to illustrate the effectiveness of our approach, we compared it with other two methods, Carlini & Wagner's attack (Carlini and Wagner 2018) and CommanderSong (Yuan et al. 2018). Table 1 gives the success probability, average SNR, $dB_x(\delta)$ and efficiency for our method and two other state-of-the-art methods. For our method, we use SPT and WPT to improve the generation, use Eq. 7 as ℓ_{model} and set ℓ^{tvd}_{metric} as our ℓ_{metric} (Eq. 11), and the proportion of perturbed points is chosen to be 75%.

As shown in Table 1, our fast approach shortens the generation time from one hour to less than 5 minutes by focusing on the key points and dynamic learning rate to accelerate the converge. In addition, our adversarial examples also have a better average of $dB_x(\delta)$ and SNR, that is, we use less calculation time and get better results. More importantly, our approach has better robustness which is shown in the next section.

[2]https://voice.mozilla.org/en/datasets

Table 2: The robustness against noise from $\triangle = 5$ to $\triangle = 30$.

Approach	$\triangle = 5$		$\triangle = 15$		$\triangle = 30$	
	Robustness ↑	WER ↓	Robustness ↑	WER ↓	Robustness ↑	WER ↓
baseline (C&W's attack)	0.23	0.49	0.04	0.81	0.01	0.93
EOT-based ($\triangle = 5$)	0.25	0.46	0.06	0.74	0.02	0.94
EOT-based ($\triangle = 15$)	0.63	0.16	0.07	0.56	0.02	0.92
EOT-based ($\triangle = 30$)	0.74	0.04	0.29	0.23	0.04	0.70
SPT-based (proportion =5%)	0.71	0.04	0.4	0.22	0.27	0.39
SPT-based (proportion =30%)	0.58	0.15	0.21	0.33	0.11	0.58
SPT-based (proportion =75%)	0.42	0.21	0.18	0.50	0.06	0.72
SPT-EOT-based (75%,30)	0.85	0.03	0.30	0.19	0.09	0.53

Table 3: Evaluation of different loss functions in our adversarial attacks.

Loss functions *	SNR ↑	$dB_x(\delta)$ ↑	Success Rate ↑
$\ell_0 = \ell_{model} + c_0 \cdot \ell^{tvd}_{metric}$	**31.9**	**46.92**	**1**
$\ell_1 = \ell_{model} + c_1 \cdot \ell^{1}_{metric}$	29.17	44.55	0.97
$\ell_2 = \ell_{model} + c_2 \cdot \ell^{2}_{metric}$	30.2	44.91	1
$\ell_3 = \ell_{model} + c_3 \cdot \ell^{3}_{metric}$	30.1	44.63	0.98

* We tried our best to tune every constant c of different ℓ_{metric} for a fair comparison. We refer interested readers to Implementation Details Section for setting details.

Evaluating robustness to noise As we mentioned in Eq.5, we evaluate the robustness of audio adversarial examples by adding noise to them and checking their adversarial properties. The process of adding noise is equal to apply transformation function $t \sim \mathcal{T}$ over the input audio. In our experiments, we set $\mathcal{T}$ as the uniform distribution with the boundary of $\pm\triangle$. We respectively added noise to SPT-based, EOT-based and SPT-EOT-based adversarial examples. Then we transcribed the newly obtained audio and finally calculate WER and Robustness Rate. If the newly transcribed phrase is the same as before, we say that this audio successfully bypassed the noise defense.

As shown in Table 2, mostly the SPT-based method performs better than the EOT-based method in terms of WER. The EOT-based audio has a higher Robustness Rate when its distribution is the same or similar to the noise distribution. However, the SPT-based audio exhibits more general robustness. More specifically, in SPT, the smaller the proportion, the better the robustness, but too small proportion results in a decrease in SNR and success rate. Fortunately, the SPT-EOT-based approach combines the advantages of both methods and performs well in all aspects. We recommend using the SPT-EOT-based approach to increase robustness in future work.

Investigation of different ℓ_{metric} In this experiment, we generate adversarial examples based on ℓ_{model} and four different ℓ_{metric} (from Eq. 11 to Eq. 12). For each specific loss function, we conduct adversarial attacks both with SPT (under the proportion of 75%) and WPT.

The results in Table-3 suggest that ℓ_0 has the best performance on SNR, $dB_x(\theta)$ and success rate. Besides, because the TVD process eliminates the harsher impulse noise, the added perturbation "sounds" more imperceptible. As a result, although the SNR and $dB_x(\theta)$ of ℓ_0 are not greatly improved in numerical value, its adversarial audio sounds quite better. Here we again suggest you listen to our demos on the website has given before.

The overall performances of ℓ_1 and ℓ_3 are not satisfactory. Since the maximum value in the audio vector is impossible to measure the magnitude of the two small disturbances under the same maximum value. It also proves that the character of the cosine distance is more suitable for audio similarity measurement. Because l_2 distance can reflect all the perturbation of audio, ℓ_2 has a better performance than ℓ_1 and ℓ_3, especially on the success rate. However, it's still worse than ℓ_0 on SNR and $dB_x(\theta)$.

Combined the experimental results and the process of tuning, we conclude that a good loss function should satisfy the following three characteristics: 1) The value ranges of ℓ_{model} and $c \cdot \ell_{metric}$ should be the same order of magnitude; 2) It should ensure that the value of ℓ_{model} are relatively larger in the initial stage, so that a feasible solution can be found as soon as possible; after finding a feasible solution, the weight of $\ell metric$ should increase, because it is necessary to find a feasible solution that is closer to the original sample; 3) Considering the characteristic of sound, a metric in audio area instead of general metric can lead to a more imperceptible adversarial audio.

Implementation Details

For reproducibility, here we give the hyperparameters used in our experiments.

Evaluating adversarial examples In this experiment, we generate audio adversarial examples with SPT and WPT and we select ℓ^{tvd}_{metric} as ℓ_{metric}. The searching decoder is a beam-search decoder. The max iteration is set to be 500, which is enough for our method to generate imperceptible adversarial examples. In SPT, the proportion of perturbed

points is 75%. In WPT, we set the weights of key points to be 1.2, the learning rate begins with 100 and β is set to be 0.8. lr will be updated by $\beta \cdot lr$, if the iterations$\%50 == 0$ and we have generated at least one adversarial example by now. The hyperparameters c and γ are 0.001 and 10.

Evaluating robustness to noise Most of the hyperparameters are set to be the same as the first experiment except that the proportion of perturbed points are 5%, 15%, 30%, 75%, respectively.

Investigation of different l_{metric} Most of hyperparameters are set to be the same as the first experiment. And c_1, c_2, c_3 are 0.01, 0.001, 1, respectively.

Transcription Examples

Some of the transcription examples are shown in Table 4. All of the phrases are selected randomly from the MCVD.

Table 4: Some of the transcription examples.

Original phrase 1	he thought of all the married shepherds he had known
Targeted phrase 1	we're refugees from the tribal wars and we need money the other figure said
Original phrase 2	i told him we could teach her to ignore people who waste her time
Targeted phrase 2	down below in the darkness were hundreds of people sleeping in peace
Original phrase 3	but finally the merchant appeared and asked the boy to shear four sheep
Targeted phrase 3	it seemed so safe and tranquil
Original phrase 4	this is no place for you
Targeted phrase 4	but finally the merchant appeared and asked the boy to shear four sheep
Original phrase 5	some of the grey ash was falling off the circular edge
Targeted phrase 5	we're refugees from the tribal wars and we need money the other figure said

Notations and Definitions

All notations and definitions used in our paper are listed in Table 5.

Table 5: Notations and Definitions used in our paper.

x	The original input audio
δ	The distortion to the original audio
t	The targeted texts
$f(\cdot)$	The threat model
$\ell(\cdot)$	The loss function to generate audio adversarial examples
$\ell_{model}(\cdot)$	The loss function to measure the difference between the current output of the model and the targeted texts
$\ell_{metric}(\cdot)$	The loss function to limit the difference between the adversarial examples and the original samples
p	The phrase of the semantic information of original audio
p'	The current transcribed phrase by ASL
y	The probability distribution over the transformed characters
π	The sequence of tokens
n	The length of the original audio vector
m	The length of the perturbed audio vector
χ	the key location interval set
c	A hyperparameter to balance the importance of ℓ_{model} and ℓ_{metric}
ω	The weights of key points
α	The weights of δ
lr	The learning rate in gradient descent
β	A hyperparameter to control the rate of decrease of the learning rate
$\nabla_\delta \ell(\cdot)$	The gradient of $\ell(\cdot)$ with regard to δ
$E(\cdot)$	The sum of closeness
$V(\cdot)$	The total variation
γ	A hyperparameter to balance the importance of $E(\cdot)$ and $V(\cdot)$
$l_p(\cdot)$	The l_p distance, such as l_0, l_2, and l_∞ etc.
$cor(\cdot)$	The cosine distance

Conclusion

This paper proposes a weighted-sampling audio adversarial example attack. The experimental results show that our method has faster speed, less noise, and stronger robustness. More importantly, we are the first to introduce the factor of the numbers and weights of perturbed points into the generation of audio adversarial examples. We also introduce TVD to improve the loss function. The study of the effectiveness of loss function shows there are some differences between the adversarial examples of image and audio. It also guides us on how to construct a more appropriate loss function in the future. Our future work will focus on the defense of audio adversarial examples.

References

Alzantot, M.; Balaji, B.; and Srivastava, M. 2018. Did you hear that? adversarial examples against automatic speech recognition. *arXiv preprint arXiv:1801.00554*.

Athalye, A.; Engstrom, L.; Ilyas, A.; and Kwok, K. 2017. Synthesizing robust adversarial examples. *arXiv preprint arXiv:1707.07397*.

Carlini, N., and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, 39–57. IEEE.

Carlini, N., and Wagner, D. 2018. Audio adversarial ex-

amples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, 1–7. IEEE.

Chen, P.-Y.; Sharma, Y.; Zhang, H.; Yi, J.; and Hsieh, C.-J. 2018. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Cisse, M.; Adi, Y.; Neverova, N.; and Keshet, J. 2017. Houdini: Fooling deep structured prediction models. *arXiv preprint arXiv:1707.05373*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Graves, A.; Fernández, S.; Gomez, F.; and Schmidhuber, J. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, 369–376. ACM.

Grosse, K.; Papernot, N.; Manoharan, P.; Backes, M.; and McDaniel, P. 2016. Adversarial perturbations against deep neural networks for malware classification. *arXiv preprint arXiv:1606.04435*.

Hannun, A.; Case, C.; Casper, J.; Catanzaro, B.; Diamos, G.; Elsen, E.; Prenger, R.; Satheesh, S.; Sengupta, S.; Coates, A.; et al. 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Hinton, G.; Deng, L.; Yu, D.; Dahl, G. E.; Mohamed, A.-r.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T. N.; et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine* 29(6):82–97.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Hu, W., and Tan, Y. 2017. Generating adversarial malware examples for black-box attacks based on gan. *arXiv preprint arXiv:1702.05983*.

Ilyas, A.; Engstrom, L.; Athalye, A.; and Lin, J. 2018. Black-box adversarial attacks with limited queries and information. *arXiv preprint arXiv:1804.08598*.

Jia, R., and Liang, P. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

Kreuk, F.; Adi, Y.; Cisse, M.; and Keshet, J. 2018. Fooling end-to-end speaker verification with adversarial examples. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1962–1966. IEEE.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

Levenshtein, V. I. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, 707–710.

Liu, X.; Zhang, X.; Guizani, N.; Lu, J.; Zhu, Q.; and Du, X. 2018. Tltd: a testing framework for learning-based iot traffic detection systems. *Sensors* 18(8):2630.

Liu, X.; Du, X.; Zhang, X.; Zhu, Q.; Wang, H.; and Guizani, M. 2019. Adversarial samples on android malware detection systems for iot systems. *Sensors* 19(4):974.

Oh, S. J.; Augustin, M.; Schiele, B.; and Fritz, M. 2017. Towards reverse-engineering black-box neural networks. *arXiv preprint arXiv:1711.01768*.

Papernot, N.; McDaniel, P.; and Goodfellow, I. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*.

Qin, Y.; Carlini, N.; Goodfellow, I.; Cottrell, G.; and Raffel, C. 2019. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. *arXiv preprint arXiv:1903.10346*.

Rudin, L. I.; Osher, S.; and Fatemi, E. 1992. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena* 60(1-4):259–268.

Schönherr, L.; Kohls, K.; Zeiler, S.; Holz, T.; and Kolossa, D. 2018. Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding. *arXiv preprint arXiv:1808.05665*.

Su, J.; Vargas, D. V.; and Sakurai, K. 2019. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Yakura, H., and Sakuma, J. 2018. Robust audio adversarial example for a physical attack. *arXiv preprint arXiv:1810.11793*.

Yang, Z.; Li, B.; Chen, P.-Y.; and Song, D. 2018. Characterizing audio adversarial examples using temporal dependency. *arXiv preprint arXiv:1809.10875*.

Yuan, X.; Chen, Y.; Zhao, Y.; Long, Y.; Liu, X.; Chen, K.; Zhang, S.; Huang, H.; Wang, X.; and Gunter, C. A. 2018. Commandersong: A systematic approach for practical adversarial voice recognition. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 49–64.