実世界でも攻撃可能なAudio Adversarial Example

矢倉 大夢[1,2]　佐久間 淳[1,2,3]

概要：深層学習モデルに意図した通りの結果を出力させる Adversarial Example について様々な研究がなされており，スマートスピーカで用いられているような音声認識モデルを対象とした手法も提案されている．ただし，モデルに音声波形を直接入力するのではなく，実世界で再生・録音した場合では意図した文字列を認識させることはできず，脅威は限定的だとされていた．そこで，本研究では，再生・録音で生まれる変換をシミュレーションし，生成処理に組み込むことで，実世界でも攻撃可能な Adversarial Example を生成する手法を提案する．また，評価実験や聴取実験により，提案手法によって生成された Adversarial Example が現実的な脅威となる可能性を示す．

キーワード：Adversarial Example, 音声認識モデル

1. はじめに

近年，深層学習は飛躍的な精度向上を遂げ，特に，画像分類や音声認識といった分野では，すでに実用的にも利用されている [1]．一方で，こうした深層学習手法には，Adversarial Example と呼ばれる弱点があることが指摘されている [2,3]．Adversarial Example とは，データに意図的に小さな摂動を加えることで，学習済みの深層学習モデルの出力を誤らせることができるという現象，あるいは，そうして作られたデータサンプルのことを指す．

具体的には，**図 1**（A）のようにパンダの画像にわずかな摂動を加えることで，人間の目には依然としてパンダに見えるものの，画像分類モデルにはテナガザルとして分類されるようになる．また，図 1（A）は画像をモデルに直接入力した場合を対象にしたものだが，紙に印刷し，それをカメラで撮影した場合でも誤分類させられるロバストな Adversarial Example の生成手法も提案されている [4]．こういった手法を用いて，深層学習モデルを用いて標識を認識している自動運転車を，「止まれ」の標識に摂動を加えるような半透明のシールを被せることで，暴走させることができる可能性も指摘されている [5]．

一方で，音声認識モデルを対象とした Adversarial Example の生成は，入力された音声波形をメル周波数ケプストラム係数（MFCC）に変換して特徴抽出を行うという非線形な処理が含まれている上に，時系列データを扱うためにネットワークが複雑であり，難しいとされていた [6]．しかし，Carlini ら [7] は，特徴抽出までを生成処理に組み込んで，音声波形のドメインで直接 Adversarial Example を生成することで，この課題を乗り越えた．そして，DeepSpeech [8] という state-of-the-art の音声認識モデルを対象に，Adversarial Example を生成できることを示した．ただし，この手法は，図 1（B）のように生成された音声波形をモデルに直接入力した場合を対象にしており，図 1（C）のようにスピーカで再生し，マイクで録音した場合でも誤認識させるには至っていない．これは，再生・録音した環境によって生じる反響や，マイクやスピーカなどから発生するノイズの影響で，モデルに直接入力した場合と音声波形が大きく変わってしまうためだと考えられる．

そこで，本研究では，実世界の再生・録音にも耐えうるロバストな Adversarial Example を生成する手法を提案する．提案手法では，反響やノイズの影響をシミュレーションし，生成処理に組み込むことで，前述の問題を解決した．我々の知る限り，DeepSpeech のような複雑な音声認識モデルに対して，実世界で攻撃可能な Adversarial Example の生成に成功したのは，本研究が初めてである．

これにより，例えば，生成された Adversarial Example をテレビ CM で再生することで，スマートスピーカを設置しているすべての家庭で気づかれないうちにピザを注文するというようなことができる可能性がある．言い換えると，音声による Adversarial Example は，テレビやラジオ，インターネットを通して拡散することができるため，様々な攻撃シナリオを考えることができる．Gilmer ら [9] も，前述のような，画像の Adversarial Example によって標識を書き換えるというようなシナリオについては，そもそも標識を切り倒してしまえば同じ目的を達成できることから，現実性がないとする一方で，音声の Adversarial Example を用いた攻撃については，さらなる検討を進めなければならないとしている．その点で，実世界でも攻撃可能な Adversarial Example の生成を可能にした本研究は，大きな意義を持つと考える．

また，画像分類モデルは，Adversarial Example によって

1　筑波大学
2　理化学研究所 革新知能統合研究センター
3　JST CREST
*1　（A）の画像は Goodfellow et al. [3] より引用した．

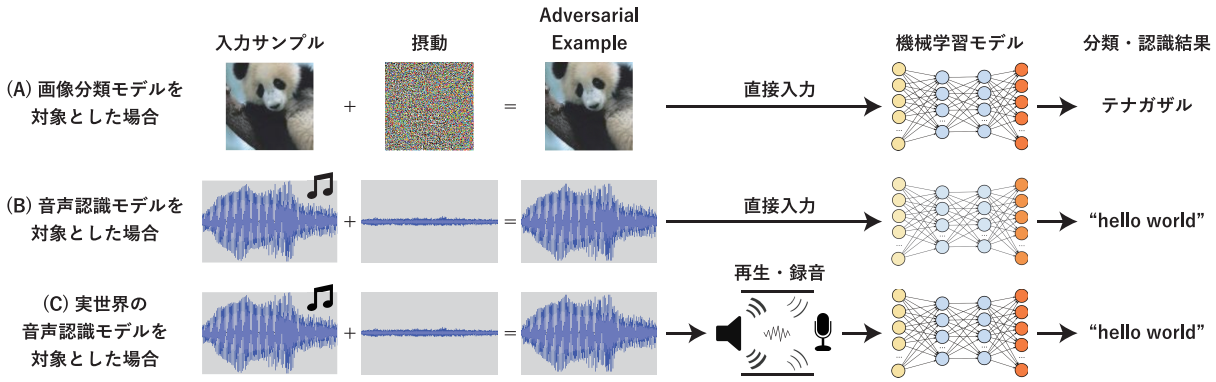

図 1: Adversarial Example のイメージ図[*1]．（A）のように画像分類モデルを対象にした生成手法が多数提案されてきたが，Carlini ら [6] は（B）のように音声認識モデルを対象にした手法を提案した．しかし，（C）のように，実世界での再生・録音に耐えうる手法の提案は本研究が初めてである．

誤らせられることのないように学習する Adversarial Training [3] を導入することで，分類精度が向上すると知られている．同様に，本研究で得られる Adversarial Example によって誤らせられることのないよう，音声認識モデルを学習することで，認識精度の向上にもつながる可能性がある．

2. 背景

本章では，Adversarial Example の定義と，その生成手法について述べた上で，音声認識システムを対象とした生成手法について詳しく説明する．

2.1 Adversarial Example とは

Adversarial Example とは，学習済み識別モデル $f : \mathbb{R}^n \to \{1, 2, \cdots, k\}$ とある入力サンプル $\boldsymbol{x} \in \mathbb{R}^n$ に対し，以下のように定義できる．

$$\tilde{\boldsymbol{x}} \in \mathbb{R}^n \, s.t. \, f(\boldsymbol{x}) \neq f(\tilde{\boldsymbol{x}}) \wedge \|\boldsymbol{x} - \tilde{\boldsymbol{x}}\| \leq \delta$$

ここで，δ は入力サンプルに加える摂動の大きさを制限するパラメータで，人間には違いがわからないという状況を生み出すために導入されている．

また，誤らせたいラベル $l \in \{1, 2, \cdots, k\}$ を指定する場合には，以下のように定義する．

$$\tilde{\boldsymbol{x}} \in \mathbb{R}^n \, s.t. \, f(\tilde{\boldsymbol{x}}) = l \wedge \|\boldsymbol{x} - \tilde{\boldsymbol{x}}\| \leq \delta \qquad (1)$$

ここで，$\boldsymbol{x} + \boldsymbol{v} = \tilde{\boldsymbol{x}}$ として，入力サンプルに付加する摂動 $\boldsymbol{v}$ に着目すると，式 1 は以下のように書き換えられる．

$$\boldsymbol{v} \in \mathbb{R}^n \, s.t. \, f(\boldsymbol{x} + \boldsymbol{v}) = l \wedge \|\boldsymbol{v}\| \leq \delta \qquad (2)$$

このとき，式 2 の算出は，識別モデル f の損失関数 $Loss_f$ を用いて，以下のような最適化問題として扱える．

$$\underset{\boldsymbol{v}}{\operatorname{argmin}} \, Loss_f(\boldsymbol{x} + \boldsymbol{v}, l) + \epsilon \|\boldsymbol{v}\| \qquad (3)$$

これは，深層学習モデルの学習処理では入力データを固定してパラメータを最適化するのに対して，Adversarial Example の作成はパラメータを固定して入力を最適化する処理と捉えることができる．

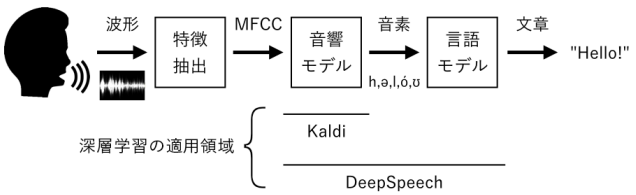

図 2: 一般的な音声認識モデルの構造．従来手法は音響モデルと言語モデルを分けており，Kaldi [10] は音響モデルのみに深層学習を用いるが，DeepSpeech [8] は両者を統合した End-to-End の深層学習モデルを用いる．

ただし，式 2 の $f(\boldsymbol{x} + \boldsymbol{v})$ という点からわかるように，式 3 はモデルに直接 Adversarial Example を与えることを想定しており，画像を印刷し，撮影するといった操作が介在する場合には適用できない．そこで，Athalye ら [4] は，印刷・撮影によって生まれる変換をシミュレーションし，生成処理に組み込む手法を提案した．これは，拡大・縮小，回転，明るさの変更，ノイズの付加などからなる，画像に対する変換の集合 $\mathcal{T}$ を用いて，以下のように表される．

$$\underset{\boldsymbol{v}}{\operatorname{argmin}} \, \mathbb{E}_{t \sim \mathcal{T}} \Big[Loss_f(t(\boldsymbol{x} + \boldsymbol{v}), l) + \epsilon \|t(\boldsymbol{x}) - t(\boldsymbol{x} + \boldsymbol{v})\| \Big] \qquad (4)$$

これにより，変換を経た上でも Adversarial Example として機能するように，生成処理が行われる．

2.2 音声認識モデルに対する Adversarial Example

本章では，一般的な音声認識モデルの構造について説明した上で，それらを対象とした Adversarial Example の生成手法について述べる．

2.2.1 音声認識モデルの構造

従来研究されてきた音声認識モデルは，**図 2** のような処理を行う．まず，音声波形から MFCC を抽出した上で，音響モデルによって音素列を得る．そして，言語モデルによって，音素列から最も尤度の高い文字列を推定する．

これまでは，音響モデルに混合ガウスモデル（GMM），言

語モデルに隠れマルコフモデル（HMM）を用いたGMM-HMM型が一般的だったが，深層学習の普及に伴い，音響モデルにDeep Neural Network（DNN）を用いたKaldi [10]のようなDNN-HMM型も用いられるようになった．DNN－HMM型では，MFCCの各フレームごとに1つの音素を出力するようDNNを学習させるため，DNN自体は時系列を考慮する必要はない．逆に，時系列を考慮しようとした場合，MFCCのフレーム数と得られる文字数が大きく異なるため，Recurrent Neural Network（RNN）やLong－Short Term Memory（LSTM）のような時系列モデルを直接用いることができない．

そこで導入されたのが，Connectionist Temporal Classification（CTC）[11]である．CTCでは，文字列の各文字と時系列モデルの各フレームへの出力の対応関係（アラインメント）について，認識すべき文字列が得られるようなアラインメント全体の生起確率を求める．これにより，生起確率の負の対数尤度を損失としたときに，時系列モデルの各出力における損失を求めることができるようになり，一般的な時系列モデルと同様にBack Propagation Through Time [12]による学習が可能になる．

CTCについて，より具体的に述べるにあたり，まず時系列モデル $f : \mathcal{X}^T \to [0,1]^{T\cdot|\mathcal{Y}|}$ を考える．ここで，$\mathcal{X}$ は入力として与えられるMFCCの集合，$\mathcal{Y}$ は出力される文字集合に「何も出力しない」という意味の ϕ を加えたラベルの集合を表すものとし，f はMFCCの T 個のフレームのそれぞれに対して，文字ごとの生起確率を出力するモデルとする．また，この生起確率から得られる系列を文字列に変換する操作 $C : \mathcal{Y}^T \to (\mathcal{Y} \setminus \{\phi\})^N$ を導入する．この $C(\cdot)$ は，連続した同じ文字を削除した後に，ϕ を削除するという処理を行う．つまり，$C(aab\phi\phi b)$ と $C(a\phi b\phi bb)$ は共に abb となる．

このとき，入力MFCC系列 $\boldsymbol{x} \in \mathcal{X}^T$ に対する，文字列 $\boldsymbol{l} \in (\mathcal{Y} \setminus \{\phi\})^N$ の生起確率は以下のように表される．

$$\mathbb{P}_{am}(\boldsymbol{l}|\boldsymbol{x}) = \sum_{\boldsymbol{\pi}|C(\boldsymbol{\pi})=\boldsymbol{l}} \mathbb{P}(\boldsymbol{\pi}|\boldsymbol{x}) = \sum_{\boldsymbol{\pi}|C(\boldsymbol{\pi})=\boldsymbol{l}} \prod_{t=1}^{T} f(\boldsymbol{x})^t_{\pi_t} \quad (5)$$

ここで，$\boldsymbol{\pi} \in \mathcal{Y}^T$ は $C(\boldsymbol{\pi}) = \boldsymbol{l}$ という条件から，文字列 $\boldsymbol{l}$ を得るような系列（abb に対して $aab\phi\phi b$ や $a\phi b\phi bb$ など）であり，$f(\boldsymbol{x})^t_{\pi_t} \in [0,1]$ は時系列モデルの出力から得られる，t フレーム目での文字 π_t の生起確率を示す．そして，この生起確率の負の対数尤度をCTCの損失関数とする．

$$\underset{f}{CTCLoss}(\boldsymbol{x}, \boldsymbol{l}) = -\log \mathbb{P}_{am}(\boldsymbol{l}|\boldsymbol{x}) \quad (6)$$

Hannunら [8]は，時系列モデルとCTCを用いて，音響モデルと言語モデルを1つのネットワークに統合したEnd-to-Endの音声認識モデルDeepSpeechを提案した．DeepSpeechでは，式5の生起確率を用いて，以下のように認識結果 $\boldsymbol{l}$ を得る．

$$\underset{\boldsymbol{l}}{argmax} \log \mathbb{P}_{am}(\boldsymbol{l}|\boldsymbol{x}) + \alpha \log \mathbb{P}_{lm}(\boldsymbol{l}) + \beta\, WordCount(\boldsymbol{l}) \quad (7)$$

ここで，$\mathbb{P}_{lm}(\boldsymbol{l})$ は言語モデルによる生起確率を表しており，得られた結果の $\boldsymbol{l}$ の自然性を担保する．DeepSpeechは，評価実験によってKaldiなどの既存手法より認識精度が大きく向上したと確かめられている上に，Mozillaが実装を公開している[*2]ことから，深層学習ベースの音声認識モデルのベースラインとなりつつある．

2.2.2 音声認識モデルに対するAdversarial Exampleの生成

現在のところ，音声認識モデルに対するAdversarial Exampleの生成手法として，2件の代表的な提案がある [7,13]．

Yuanら [13]は，Kaldi [10]のDNN（音響モデル）に対するAdversarial Exampleの生成手法を提案した．2.2.1章で述べた通り，対象とするDNNはフレームごとに独立の処理を行うことから，式2の処理をフレーム数だけ行えばよいと考えることができるが，出力が音素列であるために認識させたい文字列を直接指定することはできない．そこで，事前に認識させたい文字列を人間が発話したものを録音しておき，それをKaldiに認識させて，途中で得られる音素列 $\boldsymbol{b} = (b_1, b_2, \cdots, b_t)$ を抽出する．そして，DNNがその音素列を出力するような摂動を，以下のように求める．

$$\underset{v_t}{\operatorname{argmin}} \underset{f}{Loss}(x'_t + v_t, b_t) + \epsilon \|v_t\|$$

ここで，$\boldsymbol{x'}$ は，対象とする環境で事前に入力サンプル $\boldsymbol{x}$ を再生し，録音したものを指す．これにより，同じ機器で再生・録音した場合であれば，70%程度の精度で意図した文字列を認識させられる，ロバストなAdversarial Exampleを作成できたと述べている．

一方で，Carliniら [6]は，DeepSpeech [8]を対象としたAdversarial Exampleの作成手法を提案した．先述のように，DeepSpeechは時系列モデルを使用しているため，Yuanら [13]のようにフレーム単位で処理することができない．そこで，MFCCによる特徴抽出までをTensorFlowの計算グラフ上に実装し，対象とする文字列 $\boldsymbol{l}$ に対してCTCで得られる損失（式6）から以下のように生成する．

$$\begin{aligned} &\underset{\boldsymbol{v}}{\operatorname{argmin}}\ Loss(\boldsymbol{x} + \boldsymbol{v}) + \epsilon \|\boldsymbol{v}\| \\ &where\ \ Loss(\tilde{\boldsymbol{x}}) = \underset{f}{CTCLoss}(MFCC(\tilde{\boldsymbol{x}}), \boldsymbol{l}) \end{aligned} \quad (8)$$

ここで，$MFCC(\cdot)$ は音声波形からのMFCCの抽出を表しており，摂動は音声波形のドメインで求められることになる．この手法では，音声認識モデルに直接，音声波形を入力した場合は100 %成功したものの，実世界で再生・録音した場合は全く成功しなかったと述べている．

[*2] https://github.com/mozilla/DeepSpeech にて公開されている．

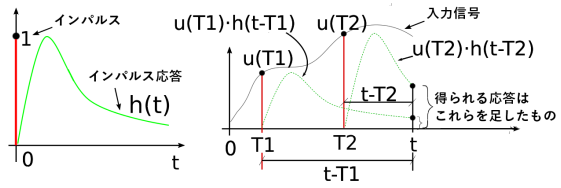

図 3: インパルス応答を用いた畳み込みのイメージ図.

3. 提案手法

本研究では，DeepSpeech [8] を対象に，実世界の再生・録音にも耐えうる，ロバストな Adversarial Example を生成する手法を提案する．基本的なアイデアは，Athalye ら [4] が画像分類モデルを対象に提案した手法と同様に，再生・録音によって生まれる変換を，生成処理に組み込むというものである．具体的には，バンドパスフィルタとインパルス応答の導入と，ホワイトガウスノイズの付加により，変換をシミュレーションする．

3.1 バンドパスフィルタの導入

人間の可聴周波数帯域は 20 ~ 20000Hz と言われており，それに合わせて一般的なスピーカは，可聴域外の周波数帯をそもそも再生できないように作られている．また，マイクについても，雑音を低減するため，可聴域外を自動的にカットするように作られていることが多い．そのため，特定の帯域外に摂動があったとしても，再生・録音の間にカットされて Adversarial Example として機能しなくなる．

そこで，生成処理において，明示的に周波数帯を制限するバンドパスフィルタを導入する．今回は，簡易的な事前実験の結果から，歪みの少なかった 1000 ~ 4000Hz に制限することとした．このとき，生成処理は式 8 をベースに，以下のように表すことができる．

$$\underset{\boldsymbol{v}}{\operatorname{argmin}}\ Loss\left(\boldsymbol{x}+\underset{1000\sim 4000\mathrm{Hz}}{BPF}\left(\boldsymbol{v}\right)\right)+\epsilon\left\|\boldsymbol{v}\right\|$$
$$where\ \ Loss\left(\tilde{\boldsymbol{x}}\right)=\underset{f}{CTCLoss}\left(MFCC\left(\tilde{\boldsymbol{x}}\right),\boldsymbol{l}\right)\quad(9)$$

これにより，生成される Adversarial Example が，スピーカやマイクで一部の周波数帯がカットされた場合でも機能するようなロバスト性を獲得すると期待できる．

3.2 インパルス応答の導入

インパルス応答とは，ある環境でインパルスと呼ばれるごく短い信号を発生させた際に得られる応答 $h(t)$ を指す．特に，音響処理においては，インパルス応答を用いて，その環境である音声 $x(t)$ を再生したときの反響を畳み込みによって $y(t)=\int_{-\infty}^{\infty}x(u)\,h(t-u)\,du$ と表現できる（**図 3**）．

インパルス応答を用いて，様々な環境での反響を再現することでロバスト性を高めるという手法は，音声認識モデルの精度向上を目的に提案されており [14]，これを Athalye ら [4] と同様にして，Adversarial Example の生成処理に組み込むこととした．つまり，様々な環境で収録したインパルス応答の集合を $\mathcal{H}$，インパルス応答 h を用いた畳み込みを $Conv_h$ として，式 9 を式 4 のような形に拡張する．

$$\underset{\boldsymbol{v}}{\operatorname{argmin}}\ \mathbb{E}_{h\sim\mathcal{H}}\left[Loss\left(\underset{h}{Conv}\left(\boldsymbol{x}+\underset{1000\sim 4000\mathrm{Hz}}{BPF}\left(\boldsymbol{v}\right)\right)\right)+\epsilon\left\|\boldsymbol{v}\right\|\right]$$
$$where\ \ Loss\left(\tilde{\boldsymbol{x}}\right)=\underset{f}{CTCLoss}\left(MFCC\left(\tilde{\boldsymbol{x}}\right),\boldsymbol{l}\right)\quad(10)$$

これにより，生成された Adversarial Example が，再生・録音した環境で生じる反響に対しても影響をされないロバスト性を獲得すると期待できる．

3.3 ホワイトガウスノイズの付加

ホワイトガウスノイズとは，$\mathcal{N}\left(0,\sigma^2\right)$ から得られるノイズであり，自然界で発生する雑音を模倣する目的で，音声認識モデルのロバスト性の評価の際にも用いられている [15]．そこで，雑音に対するロバスト性を向上させることを目的として，Adversarial Example の生成処理においても，ホワイトガウスノイズを付加することとした．このとき，生成処理は，式 10 を拡張して以下のように表される．

$$\underset{\boldsymbol{v}}{\operatorname{argmin}}\ \mathbb{E}_{h\sim\mathcal{H},\boldsymbol{w}\sim\mathcal{N}(0,\sigma^2)}\left[Loss\left(\underset{h}{Conv}\left(\boldsymbol{x}+\underset{1000\sim 4000\mathrm{Hz}}{BPF}\left(\boldsymbol{v}\right)\right)+\boldsymbol{w}\right)+\epsilon\left\|\boldsymbol{v}\right\|\right]$$
$$where\ \ Loss\left(\tilde{\boldsymbol{x}}\right)=\underset{f}{CTCLoss}\left(MFCC\left(\tilde{\boldsymbol{x}}\right),\boldsymbol{l}\right)\quad(11)$$

これにより，生成された Adversarial Example が，再生・録音に用いた機器や環境で生じる雑音に対しても影響をされないロバスト性を獲得すると期待できる．

4. 評価実験

4.1 実装

Carlini ら [7] の著者実装[*3]をベースに，式 11 の処理を実装した．ただし，損失の期待値を算出することは難しいため，イテレーションごとにバッチサイズだけインパルス応答をランダムに選択し，損失を計算するという手法を採った．また，実装上，最適化アルゴリズムに Adam [16] を用いており，このような適応的なアルゴリズムにおいて L2 正則化を行う場合には，損失にノルムを加えるのではなく Weight Decay を用いることで汎化性能が高くなるとされている [17] ことから，それに従った．

4.2 実験設定

入力サンプルとしては，Carlini らと同様に[*4]ヨハン・ゼ

[*3] https://github.com/carlini/audio_adversarial_examples/ にて公開されている．
[*4] https://nicholas.carlini.com/code/audio_adversarial_examples/ にて公開されている．

図 4: 評価実験を行った会議室の様子．マイクとスピーカを 50cm 程度の距離となるように設置し，生成された Adversarial Example の再生・録音を行った．

バスティアン・バッハの無伴奏チェロ組曲第 1 番の冒頭 4 秒分を使用した．生成に用いるインパルス応答としては，残響除去の研究で用いられているデータベース [18–21] を中心に，615 件を収集した．また，対象とする文字列は “hello world”，“open the door[*5]”，“ok google[*6]” の 3 パターンを用意し，生成処理の 100 イテレーションごとに得られた Adversarial Example を出力するようにした．そして，**図 4** のように，静かな会議室内でマイク（SONY ECM-PCV80U）及びスピーカ（JBL CLIP2）を用いて，1 つの Adversarial Example ごとに 10 回再生・録音を行い，DeepSpeech [8] での認識結果を確かめた．

4.3 評価基準

得られた Adversarial Example の評価基準としては，摂動の Signal-to-Noise Ratio（SNR）と，認識させたい文字列と再生・録音を経て認識された文字列との編集距離の平均を用いた．SNR は，入力サンプルと摂動のパワーを $P_x = \frac{1}{T}\sum_{t=1}^{T} x_t^2, P_v = \frac{1}{T}\sum_{t=1}^{T} v_t^2$ として，$10\log_{10}\frac{P_x}{P_v}$ で求められる．つまり，SNR が大きいほど摂動が小さく，人間に気づかれにくくなると言える．

編集距離とは，1 文字の挿入・削除・置換という操作について，ある文字列をもう一方の文字列に変形するのに必要な手順の最小回数として定義される．例えば，“cafe” と “coffee” という文字列の場合，“cafe”→“caffe”→“caffee”→“coffee” という手順で変形できるため，編集距離は 3 となる．ここでは，10 回の再生・録音で得られた編集距離の平均を求めているため，平均が 0 であれば 10 回とも意図した文字列を認識させられていることを，平均が 0.1 であれば 1 回のみ 1 文字だけ異なる認識結果となったことを示している．

4.4 結果

結果は，**図 5** に示す通りである．例えば，“hello world”

[*5] 既存研究 [13] でも用いられている通り，音声コマンドとして用いた攻撃シナリオの検討につながるため，これを選んだ．

[*6] Google Home（`https://store.google.com/product/google_home`）のトリガーワードであり，スマートスピーカを用いた攻撃シナリオの検討につながるため，これを選んだ．

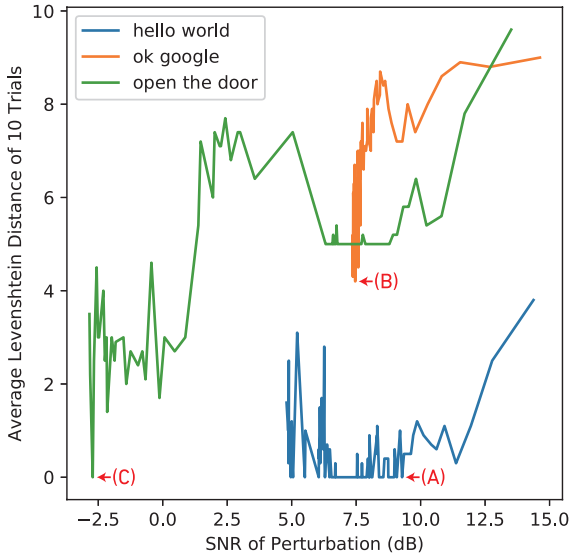

図 5: 提案手法によって生成された Adversarial Example の性能と摂動の大きさの関係．縦軸は認識させたい文字列と認識された文字列の編集距離の平均を，横軸は摂動の SNR を示している．

と認識させる Adversarial Example については，図 5（A）において SNR が 9.3dB の時点で 100%成功している．これに対し，Kaldi [10] を対象とした既存手法 [13] では，再生・録音に耐えうる Adversarial Example を生成したところ，すべてのケースで SNR が 2.0dB 未満になったと述べている．つまり，提案手法は，既存手法よりも気づかれにくいような Adversarial Example を生成することができた．

また，“open the door” のような，実世界での脅威となりうるコマンド文字列についても，図 5（C）の通り，100% 認識するような Adversarial Example を生成することができた．これは “hello world” の場合より SNR が小さくなっているものの，SNR が 1.4dB の時点で，“open the door” の認識に 50%成功する Adversarial Example が生成できたことも確認できた． つまり，2 回に 1 回は成功するということであり，攻撃シナリオによっては十分な脅威になりうると考えられる．

一方で，図 5 からは，対象とする文字列によって認識精度が大きく変わっていることもわかる．これは，式 7 で示したように DeepSpeech には言語モデルも統合されているために，単語の生起確率によって認識させやすさが変わることに起因すると考えられる．例えば，**表 1** は図 5（A∼C）について詳細な認識結果を示したものであるが，このうち “ok google” を対象とした（B）を見ると “google” という単語が全く認識されていないことがわかる．これは，DeepSpeech の学習に用いられている Common Voice Dataset[*7]に “google” という単語が含まれていないために，単語の生起確率が低くなっていることによると説明できる．

[*7] `https://voice.mozilla.org/` にて公開されている．

表 1: 図 5（A∼C）の具体的な認識結果．（A・C）は 10 回とも認識させられているが，（B）は 1 回も認識できていない．

	認識させたい文字列	SNR	認識結果
(A)	hello world	9.3dB	hello world, hello world, hello world, hello world, hello world hello world, hello world, hello world, hello world, hello world
(B)	ok google	7.5dB	oh god, oh good, oh good, oh good, oh good oh good, oh god, oh good, oh good, oh good
(C)	open the door	−2.7dB	open the door, open the door, open the door, open the door, open the door open the door, open the door, open the door, open the door, open the door

表 2: 聴取実験において参加者に提示した選択肢の一覧．音声コマンドとして用いることのできるような文字列を中心に，“hello world” や “open the door” と同じような長さの簡単なフレーズを選んだ．

ok google	turn off	open the door	happy birthday
good night	call john	hello world	airplane mode on

表 3: 図 5（A・C）の聴取実験の結果．正常でないと感じた参加者は一定数いるものの，ほとんどが選択肢を提示した場合でも内容を聞き取ることはできなかった．

	正常でないと感じた	文字列を聞き取れた	選択肢に対して 正解	不正解	わからない
(A)	36.0%	0.0%	4.0%	28.0%	68.0%
(C)	64.0%	0.0%	12.0%	16.0%	72.0%

5. 議論

5.1 生成された Adversarial Example は人間に認識できるか

生成された Adversarial Example による攻撃シナリオを考える上では，人間が認識できるかどうかという点が重要となる．もし，人間が気づかないうちに意図した文字列を認識させられるのであれば，1 章で述べたようなスマートスピーカを悪用する攻撃シナリオを考えることができる．

そのため，Kaldi [10] に対する Adversarial Example の生成手法を提案した Yuan ら [13] は，Amazon Mechanical Turk（AMT）を用いて聴取実験を行っている．その結果，65%近くの参加者が異音には気づいたものの，歌詞が変わっていると気づいたのは 2.2%のみだったと報告している．

そこで，本研究では，同様に AMT を用いた聴取実験によって，人間が気づくかどうかを確かめた．対象の Adversarial Example は，意図した文字列を 100%認識させることができた図 5（A・C）とし，それぞれ 25 名が参加した．参加者は，Adversarial Example を 3 度聴取し，聞き終えるごとに「何か正常でない点があったか」「（何かメッセージが含まれていることを提示した上で）聞き取ることはできたか」「（**表 2** の選択肢を 8 つ提示した上で）含まれているメッセージはどれだったか」という質問に順に答えた．

結果は，**表 3** に示す通りである．（A）について正常でないと感じたのは 36%のみで，それらも「音が不明瞭」や「バックグラウンドノイズがある」「鳥の鳴き声のようなものが聞こえる」といった指摘であった．（C）については 64%が正常でないと感じたが，「電話や Skype 越しのようなノイズが聞こえる」といった（A）と同様の指摘がほとんどで，メッセージや発話が含まれているというような指摘をしたものはいなかった．そして，（A）と（C）のどちらについても，文字列を聞き取れたものはいなかった．

さらに，選択肢を提示した場合でも，過半数が「わからない」と答えた．ただし，（A）については，選択肢のいずれかを選んだ参加者の正解率は 12.5%で，ランダムな選択と変わらなかったが，（C）については 42.9%で，（A）よりも気づきやすいということが示唆された．これは，表 1 で示した通り，（A）よりも（C）の方が摂動が大きくなっていることに起因すると説明できる．なお，これらは，明示的に Adversarial Example を聴取するよう指示した上で，具体的な選択肢を提示した場合の結果であり，また，「わからない」を含めると正解率は 12.0%のみである．つまり，実世界で気づかれないように再生するという攻撃シナリオであれば，大きな問題にはならないと考えられる．

以上より，生成された Adversarial Example は，ほとんど人間には気づけないか単なる雑音のように聞こえ，現実的な脅威となる可能性があると言える．また，得られたコメントから，鳥の鳴き声を入力サンプルとする，あるいは，電話を通して再生するような攻撃シナリオを考えるというようなアプローチにより，より攻撃シナリオとしての現実性を高められるのではないかと示唆される．

5.2 生成された Adversarial Example は他の音声認識モデルにも転移可能か

画像を対象とした Adversarial Example は，転移性（transferability）を持つことがあると知られている [2, 3]．これは，ある画像分類モデルを対象に生成された Adversarial Example で，異なるデータで学習された，あるいは，異なる構造を持つ別の画像分類モデルを誤らせることができるという性質を指す．

また，Yuan ら [13] は音声ドメインでも Adversarial Example が転移性を持つことを確かめられたと述べている．具体的には，Kaldi [10] が “open the door” や “good night”

表 4: 図 1（A∼C）の Kaldi [10] での認識結果．いずれの場合も，全く認識できていない．

	認識させたい文字列	認識結果
(A)	hello world	[noise], uh-huh, uh-huh, uh-huh uh-huh, uh-huh, uh-huh uh-huh, [noise], uh-huh
(B)	ok google	oh yeah, oh yeah, oh yeah, oh no oh yeah, oh yeah, oh yeah oh yeah, oh yeah, oh yeah
(C)	open the door	oh [noise], [noise], [noise], [noise] oh wow [noise], oh [noise], oh wow [noise] oh wow [noise], oh [noise], oh wow [noise]

表 5: 図 1（A）を DeepSpeech の音響モデルのみで認識した場合の認識結果．言語モデルを組み合わせると 10 回とも “hello world” と認識されたが，音響モデルのみでは 1 回も認識できていない．

helo worl	elo worl	thelo worl	helo worl	helo worl
helo worl	helo worl	hello worl	hello worl	ahelo worl

と認識するよう生成した Adversarial Example を再生・録音し，iFlytek[*8] という別の音声認識システムに与えた結果，100%の精度で意図した文字列を認識させられたという．

そこで，本研究では，提案手法で得られた Adversarial Example を再生・録音し，Kaldi に認識させることで，転移性を持つかどうかを確かめた．ここでは，Adversarial Example として図 1（A∼C）を用いたが，**表 4** に示した結果の通り，いずれも全く認識しなかった．

これは，提案手法が対象としている DeepSpeech は，式 7 に示したように音響モデルと言語モデルを同時に用いて認識処理を行うことによると考えられる．逆に Kaldi を誤認識させるには，音響モデルから出力される音素列の時点で（言語モデルにとって）自然な系列となっている必要があるため，より制約が強いと言える．これにより，4 章の評価実験にて，Kaldi を対象にした既存手法 [13] より小さな摂動の Adversarial Example を得られたことも説明できる．

実際に，生成された Adversarial Example に対して，DeepSpeech の音響モデルのみから得られる出力（式 5 における $\boldsymbol{\pi}$ に対して $C(\hat{\boldsymbol{c}})\ where\, \hat{\boldsymbol{c}} = \underset{\boldsymbol{c}}{argmax} \prod_i^T \pi_{c_i}$ で得られる）を求めると，**表 5** に示す通り，意図した文字列とは異なる認識結果となった．この結果から，音響モデルのみからの出力では意図した文字列となっていない場合でも，言語モデルの認識結果の自然性を保つ作用によって，Adversarial Example として機能するようになっていると言える．

5.3 生成された Adversarial Example は防御可能か

音声認識モデルへの Adversarial Example に対する防御手法として，Yang ら [22] により信号処理に基づく手法と時系列連続性に基づく手法が提案されている．信号処理に基づく手法としては，量子化，平滑化，ダウンサンプリングの 3 種類の変換を取り上げ，いずれについても，音声波形に適用してから音声認識モデルに入力することで，意図した文字列が認識されにくくなったと述べている．一方で，時系列連続性に基づく手法は，Carlini らの手法 [7] で生成された Adversarial Example について，音声波形の前半のみを切り出して入力した場合に認識結果が大きく変わるという現象に基づいている．例えば，“this is an adversarial example” と認識される Adversarial Example の前半のみを切り出すと，“tho lits an aters o” という結果になったという．こうした場合，全体の認識結果と，前半の認識結果の最長共通接頭辞（LCP）の長さをチェックすることで，Adversarial Example かどうかを判定できると述べている．

表 6: Adversarial Example をそのまま認識させた場合と，信号処理に基づく防御手法を適用した場合の，認識結果の編集距離の平均．数値が大きいほど Adversarial Example の生成者の意図した通りに認識させにくくなっていることを表しており，いずれの手法も一定の有効性があると言えるが，一方で量子化とダウンサンプリングは自然音声の認識結果も大きく変えてしまっている．

SNR	16bit から 8bit へ量子化	前後 4 フレームの中央値で平滑化	16000Hz から 8000Hz へダウンサンプリング
9.3dB	8.4	7.4	4.8
6.8dB	5.1	6.1	4.1
6.5dB	6.4	5.3	1.8
5.0dB	2.4	5.0	4.0
自然音声	8.9	1.4	4.1

そこで，本研究では，提案手法で得られた Adversarial Example に対して，こうした防御手法が有効かを確かめた．ここでは，比較のため，4 章の評価実験で 10 回とも意図した通り “hello world” と認識された Adversarial Example のうちから，SNR の異なる 4 つを選んだ．また，防御手法が Adversarial Example でない正常な音声の認識に与える影響を確認するため，筆者が “hello world” と発話した音声に，それぞれの防御手法を適用した場合の認識結果とも比較した．なお，信号処理に基づく手法において，量子化ビット数やサンプリング周波数といったパラメータは，Yang ら [22] に従った．

表 6 は，信号処理に基づく手法を適用した場合に，認識結果が “hello world” とどれほど変化したかを，編集距離によって示している．この結果から，量子化，平滑化，ダウンサンプリングのいずれも生成者の意図した文字列とは異なる認識結果を得ており，Adversarial Example から防御する作用があると言える．一方で，量子化とダウンサンプリングは自然音声の認識結果も大きく変えてしまってお

[*8] http://www.iflytek.com/en/mobile/iflyime.html

表 7: 時系列連続性に基づく防御手法の通り，“hello world” と認識される Adversarial Example の前半のみを入力とした場合の認識結果．自然音声の場合とほとんど変わらず，最長共通接頭辞（LCP）で見分けることは難しいと言える．

SNR	“hello world” との LCP の平均長	認識結果
9.3dB	4.0	hell, hell, hell, hell, hell hell, hell, hell, hell, hell
6.8dB	4.5	hello, hello, hello, hello, hello hello, hello, hello, hello, well
6.5dB	4.7	hell, hell, hell, hello, hello hello, hello, hello, hello, hello
5.0dB	4.8	hell, hell, hello, hello, hello hello, hello, hello, hello, hello
自然音声	5.0	hello, hello, hello, hello, hello hello, hello, hello, hello, hello

り，音声認識モデルの精度を著しく損なうことから，実用的な防御手法とは言い難い．ただし，平滑化についても，摂動が大きくなるほど編集距離が小さくなる傾向があり，さらに摂動を大きくすれば回避できる可能性が示唆される．

表 7 は，時系列連続性に基づく防御手法の通り，Adversarial Example の前半のみを入力とした場合にどのような認識結果となったかを示している．この結果からは，前述のような，半分に切り出すことで認識結果が大きく変わるという現象は見られず，防御手法としての有効性を確かめることはできなかった．以上より，提案手法によって生成された Adversarial Example に対して防御を行うには，認識精度の多少の低下を許容して平滑化を適用するか，新たなアプローチを考える必要があると言える．

6. おわりに

本研究では，音声認識モデル DeepSpeech を対象として，実世界でも攻撃可能な Audio Adversarial Example を生成する手法を提案した．提案手法では，バンドパスフィルタとインパルス応答の導入と，ホワイトガウスノイズの付加により，再生・録音によって生まれる変換をシミュレーションし，生成処理に組み込むことで，ロバストな Adversarial Example の生成を可能にした．そして，評価実験によって，より簡易な音声認識モデルを対象にした既存手法より，摂動が小さく，気づかれにくいような Adversarial Example が生成できていることを確かめた．また，我々の知る限り，DeepSpeech のような End-to-End の音声認識モデルに対して，実世界で攻撃しうる Adversarial Example の生成に成功したのは，本研究が初めてである．

今後は，Audio Adversarial Example を用いた，詳細な攻撃シナリオ，及び，防御手法の検討に取り組みたい．また，Adversarial Example の生成を学習処理に組み込んだ Adversarial Training によって，よりロバストで高性能な音声認識モデルの実現も目指したいと考えている．

謝辞 本研究は JST CREST JPMJCR1302 及び科学研究費 16H02864 の助成を受けた．

参考文献

[1] LeCun, Y. et al.: Deep learning, *Nature*, Vol. 521, No. 7553, pp. 436–444 (2015).
[2] Szegedy, C. et al.: Intriguing properties of neural networks, *ICLR*, pp. 1–10 (2014).
[3] Goodfellow, I. J. et al.: Explaining and Harnessing Adversarial Examples, *ICLR*, pp. 1–11 (2015).
[4] Athalye, A. et al.: Synthesizing Robust Adversarial Examples, *arXiv*, 1707.07397, pp. 1–19 (2017).
[5] Eykholt, K. et al.: Robust Physical-World Attacks on Deep Learning Visual Classification, *CVPR*, pp. 1625–1634 (2018).
[6] Carlini, N. et al.: Hidden Voice Commands, *USENIX Security*, pp. 513–530 (2016).
[7] Carlini, N. and Wagner, D. A.: Audio Adversarial Examples: Targeted Attacks on Speech-to-Text, *DLS*, pp. 1–7 (2018).
[8] Hannun, A. Y. et al.: Deep Speech: Scaling up end-to-end speech recognition, *arXiv*, 1412.5567, pp. 1–12 (2014).
[9] Gilmer, J. et al.: Motivating the Rules of the Game for Adversarial Example Research, *arXiv*, 1807.06732, pp. 1–33 (2018).
[10] Povey, D. et al.: The Kaldi Speech Recognition Toolkit, *ASRU*, pp. 1–4 (2011).
[11] Graves, A. et al.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks, *ICML*, pp. 369–376 (2006).
[12] Werbos, P. J.: Backpropagation Through Time: What It Does and How To Do It, *Proceedings of the IEEE*, Vol. 78, No. 10, pp. 1550–1560 (1990).
[13] Yuan, X. et al.: CommanderSong: A Systematic Approach for Practical Adversarial Voice Recognition, *USENIX Security*, pp. 1–16 (2018).
[14] Peddinti, V. et al.: Reverberation robust acoustic modeling using i-vectors with time delay neural networks, *INTERSPEECH*, pp. 2440–2444 (2015).
[15] Hansen, J. H. L. and Pellom, B. L.: An effective quality evaluation protocol for speech enhancement algorithms, *ICSLP*, pp. 2819–2822 (1998).
[16] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, *ICLR*, pp. 1–13 (2015).
[17] Loshchilov, I. and Hutter, F.: Fixing Weight Decay Regularization in Adam, *arXiv*, 1711.05101, pp. 1–13 (2017).
[18] Kinoshita, K. et al.: The reverb challenge: Acommon evaluation framework for dereverberation and recognition of reverberant speech, *WASPAA*, pp. 1–4 (2013).
[19] Nakamura, S. et al.: Acoustical Sound Database in Real Environments for Sound Scene Understanding and Hands-Free Speech Recognition, *LREC*, pp. 965–968 (2000).
[20] Jeub, M. et al.: A binaural room impulse response database for the evaluation of dereverberation algorithms, *ICDSP*, pp. 1–5 (2009).
[21] Wen, J. Y. C. et al.: Evaluation of speech dereverberation algorithms using the MARDY database, *IWAENC*, pp. 1–4 (2006).
[22] Yang, Z. et al.: Towards Mitigating Audio Adversarial Perturbations, `https://openreview.net/forum?id=SyZ2nKJDz` (2018).