Imperceptible, Robust, and Targeted
Adversarial Examples for Automatic Speech Recognition

Yao Qin [1]   **Nicholas Carlini** [2]   **Ian Goodfellow** [2]   **Garrison Cottrell** [1]   **Colin Raffel** [2]
[1] **University of California, San Diego**   [2] **Google Brain**

Abstract

Adversarial examples are inputs to machine learning models designed by an adversary to cause an incorrect output. So far, adversarial examples have been studied most extensively in the image domain. In this domain, adversarial examples can be constructed by imperceptibly modifying images to cause misclassification, and are practical in the physical world. In contrast, current targeted adversarial examples applied to speech recognition systems have neither of these properties: humans can easily identify the adversarial perturbations, and they are not effective when played over-the-air. This paper makes advances on both of these fronts. First, we develop *effectively imperceptible* audio adversarial examples (verified through a human study) by leveraging the psychoacoustic principle of auditory masking, while retaining 100% targeted success rate on arbitrary full-sentence targets. Next, we make progress towards physical-world over-the-air audio adversarial examples by constructing perturbations which remain effective even after applying realistic simulated environmental distortions.

1. Introduction

Adversarial examples (Szegedy et al., 2013) are inputs that have been specifically designed by an adversary to cause a machine learning algorithm to produce a misclassification (Biggio et al., 2013). Initial work on adversarial examples focused mainly in the domain of image classification. In order to differentiate properties of adversarial examples on neural networks in general from properties which hold true only on images, it is important to study adversarial examples in different domains. Indeed, adversarial examples are known to exist on domains ranging from reinforcement learning (Huang et al., 2017) to reading comprehension (Jia & Liang, 2017) to speech recognition (Carlini & Wagner, 2018). This paper focuses on the latter of these domains, where (Carlini & Wagner, 2018) showed that any given source audio sample can be perturbed slightly so that an au-

tomatic speech recognition (ASR) system would transcribe the audio as any different target sentence.

To date, adversarial examples on ASR differ from adversarial examples on images in two key ways. First, adversarial examples on images are imperceptible to humans: it is possible to generate an adversarial example without changing the 8-bit brightness representation (Szegedy et al., 2013). Conversely, adversarial examples on ASR systems are often perceptible. While the perturbation introduced is often small in magnitude, upon listening it is obvious that the added perturbation is present (Schönherr et al., 2018). Second, adversarial examples on images work in the physical world (Kurakin et al., 2016) (e.g., even when taking a picture of them). In contrast, adversarial examples on ASR systems do not yet work in such an "over-the-air" setting where they are played by a speaker and recorded by a microphone.

In this paper, we improve the construction of adversarial examples on ASR and match the power of attacks on images by developing adversarial examples which are imperceptible, and make steps towards robust adversarial examples[1].

In order to generate imperceptible adversarial examples, we depart from the common ℓ_p distance measure widely used for adversarial example research. Instead, we make use of the psychoacoustic principle of auditory masking, and only add the adversarial perturbation to regions of the audio where it will not be heard by a human, even if this perturbation is not "quiet" in terms of absolute energy.

Further investigating properties of adversarial examples which appear to be different from images, we examine the ability of an adversary to construct physical-world adversarial examples (Kurakin et al., 2016). These are inputs that, even after taking into account the distortions introduced by the physical world, remain adversarial upon classification. We make initial steps towards developing audio which can be played over-the-air by designing audio which remains adversarial after being processed by random room-environment simulators (Scheibler et al., 2018).

Finally, we additionally demonstrate that our attack is ca-

---

[1] The project webpage is at `http://cseweb.ucsd.edu/~yaq007/imperceptible-robust-adv.html`

pable of attacking a modern, state-of-the-art Lingvo ASR system (Shen et al., 2019).

2. Related Work

We build on a long line of work studying the robustness of neural networks. This research area largely began with (Biggio et al., 2013; Szegedy et al., 2013), who first studied *adversarial examples* for deep neural networks.

This paper focuses on adversarial examples on automatic speech recognition systems. Early work in this space (Gong & Poellabauer, 2017; Cisse et al., 2017) was successful when generating *untargeted* adversarial examples that produced incorrect, but arbitrary, transcriptions. A concurrent line of work succeeded at generating targeted attacks in practice, even when played over a speaker and recorded by a microphone (a so-called "over-the-air" attack) but only by both (a) synthesizing completely new audio and (b) targeting older, traditional (i.e., not neural network based) speech recognition systems (Carlini et al., 2016; Zhang et al., 2017; Song & Mittal, 2017).

These two lines of work were partially unified by Carlini & Wagner (2018) who constructed adversarial perturbations for speech recognition systems targeting arbitrary (multi-word) sentences. However, this attack was neither effective over-the-air, nor was the adversarial perturbation completely inaudible; while the perturbations it introduces are very quiet, they can be heard by a human (see § 7.2). Concurrently, the CommanderSong (Yuan et al., 2018) attack developed adversarial examples that are effective over-the-air but at a cost of introducing a significant perturbation to the original audio.

Following this, concurrent work with ours develops attacks on deep learning ASR systems that either work over-the-air or are less obviously perceptible.

- Yakura & Sakuma (2018), create adversarial examples which can be played over-the-air. These attacks are highly effective on short two- or three-word phrases, but not on the full-sentence phrases originally studied. Further, these adversarial examples often have a significantly larger perturbation, and in one case, the perturbation introduced had a *higher* magnitude than the original audio.
- Schönherr et al. (2018) work towards developing attacks that are less perceptible through using "Psychoacoustic Hiding" and attack the Kaldi system, which is partially based on neural networks but also uses some "traditional" components such as a Hidden Markov Model instead of an RNN for final classification. Because of the system differences we can not directly compare our results to that of this paper, but we encourage the reader to listen to examples from both papers.

Our concurrent work manages to achieve both of these results (almost) simultaneously: we generate adversarial examples that are both nearly imperceptible and also remain effective after simulated distortions. Simultaneously, we target a state-of-the-art network-based ASR system, Lingvo, as opposed to Kaldi and generate full-sentence adversarial examples as opposed to targeting short phrases.

A final line of work extends adversarial example generation on ASR systems from the white-box setting (where the adversary has complete knowledge of the underlying classifier) to the black-box setting (Khare et al., 2018; Taori et al., 2018) (where the adversary is only allowed to query the system). This work is complementary and independent of ours: we assume a white-box threat model.

3. Background

3.1. Problem Definition

Given an input audio waveform x, a target transcription y and an automatic speech recognition (ASR) system $f(\cdot)$ which outputs a final transcription, our objective is to construct an imperceptible and targeted adversarial example x' that can attack the ASR system when played over-the-air. That is, we seek to find a small perturbation δ, which enables $x' = x + \delta$ to meet three requirements:

- **Targeted**: the classifier is fooled so that $f(x') = y$ and $f(x) \neq y$. Untargeted adversarial examples on ASR systems often only introduce spelling errors and so are less interesting to study.
- **Imperceptible**: x' sounds so similar to x that humans cannot differentiate x' and x when listening to them.
- **Robust**: x' is still effective when played by a speaker and recorded by a microphone in an over-the-air attack. (We do not achieve this goal completely, but do succeed at simulated environments.)

3.1.1. ASR MODEL

We mount our attacks on the **Lingvo** classifier (Shen et al., 2019), a state-of-the-art sequence-to-sequence model (Sutskever et al., 2014) with attention (Bahdanau et al., 2014) whose architecture is based on the Listen, Attend and Spell model (Chan et al., 2016). It feeds MFCCs into an encoder consisting of a stack of convolutional and LSTM layers, which conditions an LSTM decoder that outputs the transcription. The use of the sequence-to-sequence framework allows the entire model to be trained end-to-end with the standard cross-entropy loss function.

3.1.2. THREAT MODEL

In this paper, as is done in most prior work, we consider the white box threat model where the adversary has full

access to the model as well as its parameters. In particular, the adversary is allowed to compute gradients through the model in order to generate adversarial examples.

When we mount over-the-air attacks, we do not assume we know the exact configurations of the room in which the attack will be performed. Instead, we assume we know the *distribution* from which the room will be drawn, and generate adversarial examples so as to be effective on any room drawn from this distribution.

3.2. Adversarial Example Generation

Adversarial examples are typically generated by performing gradient descent with respect to the input on a loss function designed to be minimized when the input is adversarial (Szegedy et al., 2013). Specifically, let x be an input to a neural network $f(\cdot)$, let δ be a perturbation, and let $\ell(f(x), y)$ be a loss function that is minimized when $f(x) = y$. Most work on adversarial examples focuses on minimizing the max-norm ($\|\cdot\|_\infty$ norm) of δ. Then, the typical adversarial example generation algorithm (Szegedy et al., 2013; Carlini & Wagner, 2017; Madry et al., 2017) solves

$$\text{minimize } \ell(f(x + \delta), y) + \alpha \cdot \|\delta\|$$
$$\text{such that } \|\delta\| < \epsilon$$

(where in some formulations $\alpha = 0$). Here, ϵ controls the maximum perturbation introduced.

To generate adversarial examples on ASR systems, Carlini & Wagner (2018) set ℓ to the CTC-loss and use the max-norm which has the effect of adding a small amount of adversarial perturbation consistently throughout the audio sample.

4. Imperceptible Adversarial Examples

Unlike on images, where minimizing ℓ_p distortion between an image and the nearest misclassified example yields a visually indistinguishable image, on audio, this is not the case (Schönherr et al., 2018). Thus, in this work, we depart from the ℓ_p distortion measures and instead rely on the extensive work which has been done in the audio space for capturing the human perceptibility of audio.

4.1. Psychoacoustic Models

A good understanding of the human auditory system is critical in order to be able to construct imperceptible adversarial examples. In this paper, we use *frequency masking*, which refers to the phenomenon that a louder signal (the "masker") can make other signals at nearby frequencies (the "maskees") imperceptible (Mitchell, 2004; Lin & Abdulla, 2015). In simple terms, the masker can be seen as creating a "masking threshold" in the frequency domain. Any signals which fall under this threshold are effectively imperceptible.

Because the masking threshold is measured in the frequency domain, and because audio signals change rapidly over time, we first compute the short-time Fourier transform of the raw audio signal to obtain the spectrum of overlapping sections (called "windows") of a signal. The window size N is 2048 samples which are extracted with a "hop size" of 512 samples and are windowed with the modified Hann window. We denote $s_x(k)$ as the kth bin of the spectrum of frame x.

Then, we compute the log-magnitude power spectral density (PSD) as follows:

$$p_x(k) = 10 \log_{10} \left| \frac{1}{N} s_x(k) \right|^2. \tag{1}$$

The normalized PSD estimate $\bar{p}_x(k)$ is defined by Lin & Abdulla (2015)

$$\bar{p}_x(k) = 96 - \max_k \{p_x(k)\} + p_x(k) \tag{2}$$

Masking Threshold Given an audio input, in order to compute its masking threshold, first we need to identify the maskers, whose normalized PSD estimate $\bar{p}_x(k)$ must satisfy three criteria: 1) they must be local maxima in the spectrum; 2) they must be higher than the threshold in quiet; and 3) they have the largest amplitude within 0.5 Bark (a psychoacoustically-motivated frequency scale) around the masker's frequency. Then, each masker's masking threshold can be approximated using the simple two-slope spread function, which is derived to mimic the excitation patterns of maskers. Finally, the global masking threshold $\theta_x(k)$ is a combination of the individual masking threshold as well as the threshold in quiet via addition (because the effect of masking is additive in the logarithmic domain). We refer interested readers to our appendix and (Lin & Abdulla, 2015) for specifics on computing the masking threshold.

When we add the perturbation δ to the audio input x, if the normalized PSD estimate of the perturbation $\bar{p}_\delta(k)$ is under the frequency masking threshold of the original audio $\theta_x(k)$, the perturbation will be masked out by the raw audio and therefore be inaudible to humans. The normalized PSD estimate of the perturbation $\bar{p}_\delta(k)$ can be calculated via:

$$\bar{p}_\delta(k) = 96 - \max_k \{p_x(k)\} + p_\delta(k). \tag{3}$$

where $p_\delta(k) = 10 \log_{10} |\frac{1}{N} s_\delta(k)|^2$ and $p_x(k) = 10 \log_{10} |\frac{1}{N} s_x(k)|^2$ are the PSD estimate of the perturbation and the original audio input.

4.2. Optimization with Masking Threshold

Loss function Given an audio example x and a target phrase y, we formulate the problem of constructing an imperceptible adversarial example $x' = x + \delta$ as minimizing

the loss function $\ell(x, \delta, y)$, which is defined as:

$$\ell(x, \delta, y) = \ell_{net}(f(x + \delta), y) + \alpha \cdot \ell_\theta(x, \delta) \quad (4)$$

where ℓ_{net} requires that the adversarial examples fool the audio recognition system into making a targeted prediction y, where $f(x) \neq y$. In the Lingvo model, the simple cross entropy loss function is used for ℓ_{net}. The term ℓ_θ constrains the normalized PSD estimation of the perturbation $\bar{p}_\delta(k)$ to be under the frequency masking threshold of the original audio $\theta_x(k)$. The hinge loss is used here to compute the loss for masking threshold:

$$\ell_\theta(x, \delta) = \frac{1}{\lfloor \frac{N}{2} \rfloor + 1} \sum_{k=0}^{\lfloor \frac{N}{2} \rfloor} \max\left\{\bar{p}_\delta(k) - \theta_x(k), 0\right\}, \quad (5)$$

where N is the predefined window size and $\lfloor x \rfloor$ outputs the greatest integer no larger than x. The adaptive parameter α is to balance the relative importance of these two criteria.

Stability The existence of the $\log$ function in the threshold $\theta_x(k)$ and the normalized PSD estimate of the perturbation $\bar{p}_\delta(k)$ leads to instability during back-propagation. Therefore, we remove the term $10\log_{10}$ in the PSD estimate of $p_\delta(k)$ and $p_x(k)$ and then they become:

$$p_\delta(k) = \left|\frac{1}{N} s_\delta(k)\right|^2, \quad p_x(k) = \left|\frac{1}{N} s_x(k)\right|^2 \quad (6)$$

and the normalized PSD of the perturbation turns into

$$\bar{p}_\delta(k) = \frac{10^{9.6} p_\delta(k)}{\max_k\{p_x(k)\}}. \quad (7)$$

Correspondingly, the threshold $\theta_x(k)$ becomes:

$$\theta_x(k) = 10^{\frac{\theta_x}{10}} \quad (8)$$

4.2.1. TWO STAGE ATTACK

Empirically, we find it is difficult to directly minimize the masking threshold loss function via backpropagation without any constraint on the magnitude of the perturbation δ. This is reasonable because it is very challenging to fool the neural network and limit a very large perturbation to be under the masking threshold in the frequency domain at the same time. In contrast, if the perturbation δ is relatively small in magnitude, then it will be much easier to push the remaining distortion under the frequency masking threshold.

Therefore, we divide the optimization into two stages: the first stage of optimization focuses on finding a relatively small perturbation to fool the network (as was done in prior work (Carlini & Wagner, 2018)) and the second stage makes the adversarial examples imperceptible.

In the first stage, we set α in Eqn 4 to be zero and clip the perturbation to be within a relatively small range. As a result, the first stage solves:

$$\begin{aligned} &\text{minimize } \ell_{net}(f(x + \delta), y) \\ &\text{such that } \|\delta\| < \epsilon \end{aligned} \quad (9)$$

where $\|\delta\|$ represents the $\|\cdot\|_\infty$ max-norm of δ. Specifically, we begin by setting $\delta = 0$ and then on each iteration:

$$\delta \leftarrow \text{clip}_\epsilon(\delta - lr_1 \cdot \text{sign}(\nabla_\delta \ell_{net}(f(x + \delta), y))), \quad (10)$$

where lr_1 is the learning rate and and $\nabla_\delta \ell_{net}$ is the gradient of ℓ_{net} with respect to δ. We initially set ϵ to a large value and then gradually reduced during optimization following Carlini & Wagner (2018).

The second stage focuses making the adversarial examples imperceptible, with an *unbounded* max-norm; instead, δ is only constrained by the masking threshold constraints. Specifically, initialize δ with δ^*_{im} optimized in the first stage and then on each iteration:

$$\delta \leftarrow \delta - lr_2 \cdot \nabla_\delta \ell(x, \delta, y), \quad (11)$$

where lr_2 is the learning rate and $\nabla_\delta \ell$ is the gradient of ℓ with respect to δ. The loss function $\ell(x, \delta, y)$ is defined in Eqn. 4. The parameter α that balances the network loss $\ell_{net}(f(x + \delta), y)$ and the imperceptibility loss $\ell_\theta(x, y)$ is initialized with a small value, *e.g.,* 0.05, and is adaptively updated according to the performance of the attack. Specifically, every twenty iterations, if the current adversarial example successfully fools the ASR system (i.e. $f(x + \delta) = y$), then α is increased to attempt to make the adversarial example less perceptible. Correspondingly, every fifty iterations, if the current adversarial example fails to make the targeted prediction, we decrease α. We check for attack failure less frequently than success (fifty vs. twenty iterations) to allow more iterations for the network to converge. The details of the optimization algorithm are further explained in the appendix.

5. Robust Adversarial Examples

5.1. Acoustic Room Simulator

In order to improve the robustness of adversarial examples when playing over-the-air, we use an acoustic room simulator to create artificial utterances (speech with reverberations) that mimic playing the audio over-the-air. The transformation function in the acoustic room simulator, denoted as t, takes the clean audio x as an input and outputs the simulated speech with reverberation $t(x)$. First, the room simulator applies the classic Image Source Method introduced in (Allen & Berkley, 1979; Scheibler et al., 2018) to create the room impulse response r based on the room configurations (the

room dimension, source audio and target microphone's location, and reverberation time). Then, the generated room impulse response r is convolved with the clean audio to create the speech with reverberation, to obtain $t(x) = x * r$ where $*$ denotes the convolution operation. To make the generated adversarial examples robust to various environments, multiple room impulse responses r are used. Therefore, the transformation function t follows a chosen distribution $\mathcal{T}$ over different room configurations.

5.2. Optimization with Reverberations

In this section, our objective is to make the perturbed speech with reverberation (rather than the clean audio) fool the ASR system. As a result, the generated adversarial examples $x' = x + \delta$ will be passed through the room simulator first to create the simulated speech with reverberation $t(x')$, mimicking playing the adversarial examples over-the-air, and then the simulated $t(x')$ will be fed as the new input to fool the ASR system, aiming at $f(t(x')) = y$. Simultaneously, the adversarial perturbation δ should be relatively small in order not to be audible to humans.

In the same manner as the Expectation over Transformation in (Athalye et al., 2018), we optimize the expectation of the loss function over different transformations $t \sim \mathcal{T}$ as follows:

$$\text{minimize } \ell(x, \delta, y) = \mathop{\mathbb{E}}_{t \sim \mathcal{T}} \left[\ell_{net}(f(t(x + \delta)), y) \right] \tag{12}$$

$$\text{such that } \|\delta\| < \epsilon.$$

Rather than directly targeting $f(x + \delta) = y$, we apply the loss function l_{net} (the cross entropy loss in the Lingvo network) to the classification of the transformed speech $f(t(x + \delta)) = y$. We approximate the gradient of the expected value via independently sampling a transformation t from the distribution $\mathcal{T}$ at each gradient descent step.

In the first I_{r_1} iterations, we initialize ϵ with a sufficiently large value and gradually reduce it following Carlini & Wagner (2018). We consider the adversarial example successful if it successfully fools the ASR system under a single random room configuration; that is, if $f(t(x + \delta)) = y$ for just one $t(\cdot)$. Once this optimization is complete, we obtain the max-norm bound for δ, denoted as ϵ_r^*. We will then use the perturbation δ_r^* as an initialization for δ in the next stage.

Then in the following I_{r_2} iterations, we finetune the perturbation δ with a much smaller learning rate. The max-norm bound ϵ is increased to $\epsilon_r^{**} = \epsilon_r^* + \Delta$, where $\Delta > 0$, and held constant during optimization. During this phase, we only consider the attack successful if the adversarial example successfully fools a set of randomly chosen transformations $\Omega = \{t_1, t_2, \cdots, t_M\}$, where $t_i \sim \mathcal{T}$ and M is the size of the set Ω. The transformation set Ω is randomly sampled from the distribution $\mathcal{T}$ at each gradient descent step. In other words, the adversarial example $x' = x + \delta$ gener-

ated in this stage satisfies $\forall t_i \in \Omega, f(t_i(x + \delta)) = y$. In this way, we can generate robust adversarial examples that successfully attack ASR systems when the exact room environment is not known ahead of time, whose configuration is drawn from a pre-defined distribution. More details of the algorithm are shown in the appendix.

It should be emphasized that there is a tradeoff between imperceptibility and robustness (as we will show experimentally in Section 7.2). If we increase the max amplitude of the perturbation ϵ_r^{**}, the robustness can always be further improved. Correspondingly, it becomes much easier for humans to perceive the adversarial perturbation and alert the ASR system. In order to keep these adversarial examples mostly imperceptible, we therefore limit the ℓ_∞ amplitude of the perturbation to be in a reasonable range.

6. Imperceptible and Robust Attacks

By combining both of the techniques we developed earlier, we now develop an approach to generate both imperceptible and robust adversarial examples. This can be achieved by minimizing the loss

$$\ell(x, \delta, y) = \mathop{\mathbb{E}}_{t \sim \mathcal{T}} \left[\ell_{net}(f(t(x + \delta)), y) + \alpha \cdot \ell_\theta(x, \delta) \right], \tag{13}$$

where the cross entropy loss function $\ell_{net}(\cdot)$ is again the loss used for Lingvo, and the imperceptibility loss $\ell_\theta(\cdot)$ is the same as that defined in Eqn 5. Since we need to fool the ASR system when the speech is played after random perturbations, the cross entropy loss $\ell_{net}(f(t(x + \delta)), y)$ forces the transcription of the transformed adversarial example $t(x + \delta)$ to be y (again, as done earlier).

To further improve these adversarial examples to be imperceptible, we optimize $\ell_\theta(x, \delta)$ to constrain the perturbation δ to fall under the masking threshold of the clean audio in the frequency domain. This is much easier compared to optimizing the hinge loss $\ell_\theta(t(x), t(\delta)) = \max\{\bar{p}_{t(\delta)}(k) - \theta_{t(x)}(k), 0\}$ because the frequency masking threshold of the clean audio $\theta_x(k)$ can be pre-computed while the masking threshold of the speech with reverberation $\theta_{t(x)}(k)$ varies with the room reverberation r. In addition, optimizing $\ell_\theta(x, \delta)$ and $\ell_\theta(t(x), t(\delta))$ have similar effects based on the convolution theorem that the Fourier transform of a convolution of two signals is the pointwise product of their Fourier transforms. Note that the speech with reverberation $t(x)$ is a convolution of the clean audio x and a simulated room reverberation r, hence:

$$\mathscr{F}\{t(x)\} = \mathscr{F}\{x * r\} = \mathscr{F}\{x\} \cdot \mathscr{F}\{r\} \tag{14}$$

where $\mathscr{F}$ is the Fourier transform, $*$ denotes the convolution operation and $\cdot$ represents the pointwise product. We apply the short-time Fourier transform to the perturbation and the raw audio signal first in order to compute the power

spectral density $\bar{p}_{t(\delta)}$ and the masking threshold $\theta_{t(x)}$ in the frequency domain. Since most of the energy in the room impulse response falls within the spectral analysis window size, the convolution theorem in Eqn 14 is approximately satisfied. Therefore, we arrive at:

$$(\bar{p}_{t(\delta)} - \theta_{t(x)}) \approx (\bar{p}_\delta - \theta_x) \cdot \mathscr{F}\{r\}. \quad (15)$$

As a result, optimizing the imperceptibility loss $\ell_\theta(x, \delta)$ can help in finding the optimal δ and in constructing the imperceptible adversarial examples that can attack the ASR systems in the physical world.

Specifically, we will first initialize δ with the perturbation δ_r^{**} that enables the adversarial examples to be robust in Section 5. Then in each iteration, we randomly sample a transformation t from the distribution $\mathcal{T}$ and update δ according to:

$$\delta \leftarrow \delta - lr_3 \cdot \nabla_\delta \big[\ell_{net}(f(t(x+\delta)), y) + \alpha \cdot \ell_\theta(x, \delta))\big], \quad (16)$$

where lr_3 is the learning rate and α, a parameter that balances the importance of the robustness and the imperceptibility, is adaptively changed based on the performance of adversarial examples. Specifically, if the constructed adversarial example can successfully attack a set of randomly chosen transformations, then α will be increased to focus more on imperceptibility loss. Otherwise, α is decreased to make the attack more robust to multiple room environments. The implementation details are illustrated in the appendix.

7. Evaluation

7.1. Datasets and Evaluation Metrics

Datasets We use the LibriSpeech dataset (Panayotov et al., 2015) in our experiments, which is a corpus of 16KHz English speech derived from audiobooks and is used to train the Lingvo system (Shen et al., 2019). We randomly select 1000 audio examples as source examples, and 1000 separate transcriptions from the test-clean dataset to be the targeted transcriptions. We ensure that each target transcription is around the same length as the original transcription because it is unrealistic and overly challenging to perturb a short audio clip (*e.g.,* 10 words) to have a much longer transcription (*e.g.,* 20 words). Examples of the original and targeted transcriptions are available in the appendix.

Evaluation Metrics For automatic speech recognition, we evaluate our model using the standard word error rate (WER) metric, which is defined as WER $= \frac{S+D+I}{N_W} \times 100\%$, where S, D and I are the number of substitutions, deletions and insertions of words respectively, and N_W is the total number of words in the reference.

We also calculate the success rate (sentence-level accuracy) as Accuracy $= \frac{N_s}{N_a} \times 100\%$, where N_a is the number of
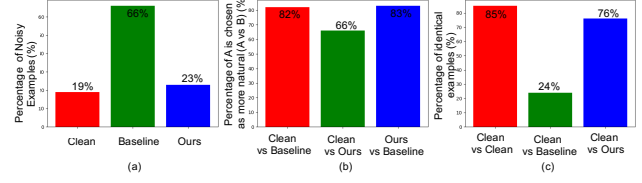

Figure 1. Results of human study for imperceptibility. Here baseline represents the adversarial example generated by Carlini & Wagner (2018), and ours denotes the imperceptible adversarial example generated following the algorithm in Section. 4.

audio examples that we test, and N_s is the number of audio examples that are correctly transcribed. Here, "correctly transcribed" means the original transcription for clean audio and the targeted transcription for adversarial examples

7.2. Imperceptibility Analysis

To attack the Lingvo ASR system, we construct 1000 imperceptible and targeted adversarial examples, one for each of the examples we sampled from the LibriSpeech test-clean dataset. Table 1 shows the performance of the clean audio and the constructed adversarial examples. We can see that the word error rate (WER) of the clean audio is just 4.47% on the 1000 test examples, indicating the model is of high quality. Our imperceptible adversarial examples perform even better, and reach a 100% success rate.

7.2.1. QUALITATIVE HUMAN STUDY

Of the 1000 examples selected from the test set, we randomly selected 100 of these with their corresponding imperceptible adversarial example. We generate then generate an adversarial example using the prior work of Carlini & Wagner (2018) for the same target phrase; this attack again succeeds with 100% success. We perform three experiments to validate that our adversarial examples are imperceptible, especially compared to prior work.

Experimental Design. We recruit 80 users online from Amazon Mechanical Turk. We give each user one of the three (nearly identical) experiments, each of which we describe below. In all cases, the experiments consist of 20 "comparisons tasks", where we present the evaluator with some audio samples and ask them questions (described below) about the samples. We ask the user to listen to each sample with headphones on, and answer a simple question about the audio samples (the question is determined by which experiment we run, as given below). We do not explain the purpose of the study other than that it is a research study, and do not record any personally identifying information.[2] We randomly include a small number of questions

---

[2]Unfortunately, for this reason, we are unable to report aggregate statistics such as age or gender, slightly harming potential reproducibility.

Input	Clean	Adversarial
Accuracy (%)	58.60	100.00
WER (%)	4.47	0.00

Table 1. Sentence-level accuracy and WER for 1000 clean and (imperceptible) adversarially perturbed examples, fed without over-the-air simulation into the Lingvo model. In "Clean", the ground truth is the original transcription. In"Adversarial", the ground truth is the targeted transcription.

Input	Clean	Robust ($\Delta = 300$)	Robust ($\Delta = 400$)	Imperceptible & Robust
Accuracy (%)	31.37	62.96	64.64	49.65
WER (%)	15.42	14.45	13.83	22.98

Table 2. Sentence-level accuracy and WER for 100 clean and adversarially perturbed examples, fed with over-the-air simulation into the Lingvo model. The ground truth for "clean" inputs is the original transcription while the ground truth is the targeted transcription for the adversarial inputs. The perturbation is bounded by $\|\delta\| < \epsilon_r^* + \Delta$.

with known, obvious answers; we remove 3 users from the study who failed to answer these questions correctly.

In all experiments, users have the ability to listen to audio files multiple times when they are unsure of the answer, making it as difficult as possible for our adversarial examples to pass as clean data. Users additionally have the added benefit of hearing 20 examples back-to-back, effectively "training" them to recognize subtle differences. Indeed, a permutation test finds users are statistically significantly better at distinguishing adversarial examples from clean audio during the second half of the experiment compared to the first half of the experiment, although the magnitude of the difference is small: only by about 3%.

Figure 1 summarizes the statistical results we give below.

Experiment 1: clean or noisy. We begin with what we believe is the most representative experiment of how an attack would work in practice. We give users one audio sample and ask them to tell us if it has *any* background noise (e.g., static, echoing, people talking in the background).

As a baseline, users believed that 19% of original clean audio samples contained some amount of noise, and 66% of users believed that the adversarial examples generated by Carlini & Wagner (2018) contained some amount of noise. In comparison, only 23% of users believe that the adversarial examples we generate contain any noise, a result that is not statistically significantly different from clean audio ($p > .05$). That is, when presented with just one audio sample in isolation, users do not believe the adversarial examples we generate are any noisier than the clean samples.

Experiment 2: identify the original. We give users two audio samples and inform them that one of the audio samples is a modified version of the other; we ask the user to select the audio sample corresponding to the one which sounds like the *more natural* audio sample. This setup is much more challenging: when users can listen to both the before and after, it is often possible to pick up on the small amount of distortion that has been added. When comparing the original audio to the adversarial examples generated by

Carlini & Wagner (2018), the evaluator chose the original audio 82% of the time. When we have the evaluator compare the imperceptible adversarial examples we generate to those of Carlini & Wagner (2018), our imperceptible examples are selected as the better audio sample 83% of the time—a difference that is not statistically distinguishable from comparing against the clean audio.

However, when directly comparing the adversarial examples we generate to the clean audio, users prefer the clean audio still 66% of the time. Observe that the baseline percentage, when the samples are completely indistinguishable, is 50%. Thus, users only perform 16% better than random guessing at distinguishing our examples from clean examples.

Experiment 3: identical or not. Finally, we perform the most difficult experiment: we present users with two audio files, and ask them if the audio samples are identical, or if there are *any* differences. As the baseline, when given the same audio sample twice, users agreed it was identical 85% of the time. (That is, in 15% of cases the evaluator wrongly heard a difference between the two samples.) When given a clean audio sample and comparing it to the audio generated by Carlini & Wagner (2018), users only believed them to be identical 24% of the time. Comparing clean audio to the adversarial examples we generate, user believed them to be completely identical 76% of the time, $3\times$ more often than the adversarial examples generated by the baseline, but below the 85%-identical value for actually-identical audio.

7.3. Robustness Analysis

To mount our simulated over-the-air attacks, we consider a challenging setting that the exact configuration of the room in which the attack will be performed is unknown. Instead, we are only aware of the distribution from which the room configuration will be drawn. First, we generate 1000 random room configurations sampled from the distribution as the training room set. The test room set includes another 100 random room configurations sampled from the same distribution. Adversarial examples are created to attack the Lingvo ASR system when played in the simulated test

rooms. We randomly choose 100 audio examples from LibriSpeech dataset to perform this robustness test.

As shown in Table 2, when fed non-adversarial audio played in simulated test rooms, the WER of the Lingvo ASR degrades to 15.42% which suggests some robustness to reverberation. In contrast, the success rate of adversarial examples in (Carlini & Wagner, 2018) and our imperceptible adversarial examples in Section 4 are 0% in this setting. The success rate of our robust adversarial examples generated based on the algorithm in Section 5 is over 60%, and the WER is smaller than that of the clean audio. Both the success rate and the WER demonstrate that our constructed adversarial examples remain effective when played in the highly-realistic simulated environment.

In addition, the robustness of the constructed adversarial examples can be improved further at the cost of increased perceptibility. As presented in Table 2, when we increase the max-norm bound of the amplitude of the adversarial perturbation $\epsilon_r^{**} = \epsilon_r^* + \Delta$ (Δ is increased from 300 to 400), both the success rate and WER are improved correspondingly. Since our final objective is to generate imperceptible and robust adversarial examples that can be played over-the-air in the physical world, we limit the max-norm bound of the perturbation to be in a relatively small range to avoid a huge distortion toward the clean audio.

To construct imperceptible as well as robust adversarial examples, we start from the robust attack ($\Delta = 300$) and finetune it with the imperceptibility loss. In our experiments, we observe that 81% of the robust adversarial examples [3] can be further improved to be much less perceptible while still retaining high robustness (around 50% success rate and 22.98% WER).

7.3.1. QUALITATIVE HUMAN STUDY

We run identical experiments (as described earlier) on the robust and robust-and-imperceptible adversarial examples.

In **experiment 1**, where we ask evaluators if there is any noise, only 6% heard any noise on the clean audio, compared to 100% on the robust (but perceptible) adversarial examples and 83% on the robust and imperceptible adversarial examples. [4]

In **experiment 2**, where we ask evaluators to identify the original audio, comparing clean to robust adversarial examples the evaluator correctly identified the original audio 97%

---

[3]The other 19% adversarial examples lose the robustness because they cannot successfully attack the ASR system in 8 randomly chosen training rooms in any iteration during optimization.

[4]Evaluators stated they heard noise on clean examples 3× less often compared to the baseline in the prior study. We believe this is due to the fact that when primed with examples which are obviously different, the baseline becomes more easily distinguishable.

of the time versus 89% when comparing the clean audio to the imperceptible and robust adversarial examples.

Finally, in **experiment 3**, where we ask evaluators if the audio is identical, the baseline clean audio was judged different 95% of the time when compared to the robust adversarial examples, and the clean audio was judged different 71% of the time when compared to the imperceptible and robust adversarial examples.

In all cases, the imperceptible and robust adversarial examples are statistically significantly less perceptible than just the robust adversarial examples, but also statistically significantly more perceptible than the clean audio. Directly comparing the imperceptible and robust adversarial examples to the robust examples, evaluators believed the imperceptible examples had less distortion 91% of the time.

Clearly the adversarial examples that are robust are significantly easier to distinguish from clean audio, even when we apply the masking threshold. However, this result is consistent with work on adversarial examples on images, where completely imperceptible physical-world adversarial examples have not been successfully constructed. On images, physical attacks require over 16× as much distortion to be effective on the physical world (see, for example, Figure 4 of Kurakin et al. (2016)).

8. Conclusion

In this paper, we successfully construct imperceptible adversarial examples (verified by a human study) for automatic speech recognition based on the psychoacoustic principle of auditory masking, while retaining 100% targeted success rate on arbitrary full-sentence targets. Simultaneously, we also make progress towards developing robust adversarial examples that remain effective after being played over-the-air (processed by random room environment simulators), increasing the practicality of actual real-world attacks using adversarial examples targeting ASR systems.

We believe that future work is still required: our robust adversarial examples do not play fully over-the-air, despite working in simulated room environments. Resolving this difficulty while maintaining a high targeted success rate is necessary for demonstrating a practical security concern.

As a final contribution of potentially independent interest, this work demonstrates how one might go about constructing adversarial examples for non-ℓ_p-based metrics. Especially on images, nearly all adversarial example research has focused on this highly-limited distance measure. Devoting effort to identifying different methods that humans use to assess similarity, and generating adversarial examples exploiting those metrics, is an important research effort we hope future work will explore.

References

Allen, J. B. and Berkley, D. A. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 65(4):943–950, 1979.

Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K. Synthesizing robust adversarial examples. In *ICML*, 2018.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 387–402. Springer, 2013.

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

Carlini, N. and Wagner, D. A. Audio adversarial examples: Targeted attacks on speech-to-text. *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 1–7, 2018.

Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C., Wagner, D., and Zhou, W. Hidden voice commands. In *USENIX Security Symposium*, pp. 513–530, 2016.

Chan, W., Jaitly, N., Le, Q., and Vinyals, O. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pp. 4960–4964. IEEE, 2016.

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

Gong, Y. and Poellabauer, C. Crafting adversarial examples for speech paralinguistics applications. *arXiv preprint arXiv:1711.03280*, 2017.

Huang, S., Papernot, N., Goodfellow, I., Duan, Y., and Abbeel, P. Adversarial attacks on neural network policies. *arXiv preprint arXiv:1702.02284*, 2017.

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

Khare, S., Aralikatte, R., and Mani, S. Adversarial black-box attacks for automatic speech recognition systems using multi-objective genetic optimization. *arXiv preprint arXiv:1811.01312*, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Lin, Y. and Abdulla, W. H. Principles of psychoacoustics. In *Audio Watermark*, pp. 15–49. Springer, 2015.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Mitchell, J. L. Introduction to digital audio coding and standards. *Journal of Electronic Imaging*, 13(2):399, 2004.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. Librispeech: an asr corpus based on public domain audio books. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pp. 5206–5210. IEEE, 2015.

Scheibler, R., Bezzam, E., and Dokmanić, I. Pyroomacoustics: A python package for audio room simulation and array processing algorithms. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 351–355. IEEE, 2018.

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

Shen, J., Nguyen, P., Wu, Y., Chen, Z., Chen, M. X., Jia, Y., Kannan, A., Sainath, T., Cao, Y., Chiu, C.-C., et al. Lingvo: a modular and scalable framework for sequence-to-sequence modeling. *arXiv preprint arXiv:1902.08295*, 2019.

Song, L. and Mittal, P. Inaudible voice commands. *arXiv preprint arXiv:1708.07238*, 2017.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pp. 3104–3112, 2014.

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

Taori, R., Kamsetty, A., Chu, B., and Vemuri, N. Targeted adversarial examples for black box audio systems. *arXiv preprint arXiv:1805.07820*, 2018.

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

Yuan, X., Chen, Y., Zhao, Y., Long, Y., Liu, X., Chen, K., Zhang, S., Huang, H., Wang, X., and Gunter, C. A. Commandersong: A systematic approach for practical adversarial voice recognition. *arXiv preprint arXiv:1801.08535*, 2018.

Zhang, G., Yan, C., Ji, X., Zhang, T., Zhang, T., and Xu, W. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 103–117. ACM, 2017.

Appendix

A. Frequency Masking Threshold

In this section, we detail how we compute the frequency masking threshold for constructing imperceptible adversarial examples. This procedure is based on psychoacoustic principles which were refined over many years of human studies. For further background on psychoacoustic models, we refer the interested reader to (Lin & Abdulla, 2015; Mitchell, 2004).

Step 1: Identifications of Maskers

In order to compute the frequency masking threshold of an input signal $x(n)$, where $0 \leq n \leq N$, we need to first identify the maskers. There are two different classes of maskers: tonal and nontonal maskers, where nontonal maskers have stronger masking effects compared to tonal maskers. Here we simply treat all the maskers as tonal ones to make sure the threshold that we compute can always mask out the noise. The normalized PSD estimate of the tonal maskers $\bar{p}_x^m(k)$ must meet three criteria. First, they must be local maxima in the spectrum, satisfying:

$$\bar{p}_x(k-1) \leq \bar{p}_x^m(k) \leq \bar{p}_x(k+1), \quad (17)$$

where $0 \leq k < \frac{N}{2}$.

Second, the normalized PSD estimate of any masker must be higher than the threshold in quiet ATH(k), which is:

$$\bar{p}_x^m(k) \geq \text{ATH}(k), \quad (18)$$

where ATH(k) is approximated by the following frequency-dependency function:

$$\text{ATH}(f) = 3.64(\frac{f}{1000})^{-0.8} - 6.5 \exp\{-0.6(\frac{f}{1000} - 3.3)^2\}$$
$$+ 10^{-3}(\frac{f}{1000})^4. \quad (19)$$

The quiet threshold only applies to the human hearing range of 20Hz $\leq f \leq$ 20kHz. When we perform short time Fourier transform (STFT) to a signal, the relation between the frequency f and the index of sampling points k is

$$f = \frac{k}{N} \cdot f_s, \quad 0 \leq f < \frac{f_s}{2} \quad (20)$$

where f_s is the sampling frequency and N is the window size.

Last, the maskers must have the highest PSD within the range of 0.5 Bark around the masker's frequency. Human's main hearing range between 20Hz and 16kHz is divided into 24 non-overlapping critical bands, whose unit is Bark, varying as a function of frequency f as follows:

$$b(f) = 13 \arctan(\frac{0.76f}{1000}) + 3.5 \arctan(\frac{f}{7500})^2. \quad (21)$$

As the effect of masking is additive in the logarithmic domain, the PSD estimate of the masker is further smoothed with its neighbors by:

$$\bar{p}_x^m(\bar{k}) = 10 \log_{10}[10^{\frac{\bar{p}_x(k-1)}{10}} + 10^{\frac{\bar{p}_x^m(k)}{10}} + 10^{\frac{\bar{p}_x(k+1)}{10}}] \quad (22)$$

Step 2: Individual masking thresholds

An individual masking threshold is better computed with frequency denoted at the Bark scale because the spreading functions of the masker would be similar at different Barks. We use $b(i)$ to represent the Bark scale of the frequency index i. There are a number of spreading functions introduced to imitate the characteristics of maskers and here we choose the simple two-slope spread function:

$$\text{SF}[b(i), b(j)] = \begin{cases} 27[b(j) - b(i)], & \text{if } [b(j) - b(i)] \leq 0. \\ G(b(i)) \cdot [b(j) - b(i)], & \text{otherwise} \end{cases} \quad (23)$$

where $G(b(i)) = [-27 + 0.37 \max\{\bar{p}_x^m(b(i)) - 40, 0\}]$ and $b(i)$ and $b(j)$ are the Bark scale of the masker at the frequency index i and the maskee at frequency index j respectively. Then, $T[b(i), b(j)]$ refers to the masker at Bark index $b(i)$ contributing to the masking effect on the maskee at Bark index $b(j)$. Empirically, the threshold $T[b(i), b(j)]$ is calculated by:

$$T[b(i), b(j)] = \bar{p}_x^m(b(i)) + \Delta_m[b(i)] + \text{SF}[b(i), b(j)], \quad (24)$$

where $\Delta_m[b(i)] = -6.025 - 0.275b(i)$ and $\text{SF}[b(i), b(j)]$ is the spreading function.

Step 3: Global masking threshold

The global masking threshold is a combination of individual masking thresholds as well as the threshold in quiet via

addition. The global masking threshold at frequency index i measured with Decibels (dB) is calculated according to:

$$\theta_x(i) = 10 \log_{10}[10^{\frac{ATH(i)}{10}} + \sum_{j}^{N_m} 10^{\frac{T[b(j),b[i]]}{10}}], \quad (25)$$

where N_m is the set of all the selected maskers. The computed θ_x is used as the frequency masking threshold for the input audio x to construct imperceptible adversarial examples.

B. Notations and Definitions

The notations and definitions used in our proposed algorithms are listed in Table 3.

C. Implementation Details

The adversarial examples generated in our paper are all optimized via Adam optimizer (Kingma & Ba, 2014). The hyperparameters used in each section are displayed below.

C.1. Imperceptible Adversarial Examples

In order to construct imperceptible adversarial examples, we divide the optimization into two stages. In the first stage, the learning rate lr_1 is set to be 100 and the number of iterations T_1 is 1000 as (Carlini & Wagner, 2018). The max-norm bound ϵ starts from 2000 and will be gradually reduced during optimization. In the second stage, the number of iterations T_2 is 4000. The learning rate lr_2 starts from 1 and will be reduced to be 0.1 after 3000 iterations. The adaptive parameter α which balances the importance between ℓ_{net} and ℓ_θ begins with 0.05 and gradually updated based on the performance of adversarial examples. Algorithm 1 shows the details of the two-stage optimization.

C.2. Robust Adversarial Examples

To develop the robust adversarial examples that could work after being played over-the-air, we also optimize the adversarial perturbation in two stages. The first stage intends to find a relatively small perturbation while the second stage focuses on making the constructed adversarial example more robust to random room configurations. The learning rate lr_1 in the first stage is 50 and δ will be updated for 2000 iterations. The max-norm bound ϵ for the adversarial perturbation δ starts from 2000 as well and will be gradually reduced. In the second stage, the number of iterations is set to be 4000 and the learning rate lr_2 is 5. In this stage, ϵ is fixed and equals the optimized ϵ_r^* in the first stage plus Δ. The size of the transformation set Ω is set to be $M = 10$.

C.3. Imperceptible and Robust Attacks

To construct imperceptible and robust adversarial examples, we begin with the robust adversarial examples generated

---

Algorithm 1 Optimization with Masking Threshold

Input: audio waveform x, target phrase y, ASR system $f(\cdot)$, perturbation δ, loss function $\ell(x, \delta, y)$, hyperparameters ϵ and α, learning rate in the first stage lr_1 and second stage lr_2, number of iterations in the first stage T_1 and second stage T_2.
\# Stage 1: minimize $\|\delta\|$
Initialize $\delta = 0$, $\epsilon = 2000$ and $\alpha = 0$.
for $i = 0$ **to** $T_1 - 1$ **do**
    $\delta \leftarrow \delta - lr_1 \cdot \text{sign}(\nabla_\delta \ell(x, \delta, y))$
    Clip $\|\delta\| \leq \epsilon$
    if $i \% 10 = 0$ and $f(x + \delta) = y$ **then**
        if $\epsilon > \max(\|\delta\|)$ **then**
            $\epsilon \leftarrow \max(\|\delta\|)$
        end if
        $\epsilon \leftarrow 0.8 \cdot \epsilon$
    end if
end for

\# Stage 2: minimize the perceptibility
Reassign $\alpha = 0.05$
for $i = 0$ **to** $T_2 - 1$ **do**
    $\delta \leftarrow \delta - lr_2 \cdot \nabla_\delta \ell(x, \delta, y)$
    if $i \% 20 = 0$ and $f(x + \delta) = y$ **then**
        $\alpha \leftarrow 1.2 \cdot \alpha$
    end if
    if $i \% 50 = 0$ and $f(x + \delta) \neq y$ **then**
        $\alpha \leftarrow 0.8 \cdot \alpha$
    end if
end for
Output: adversarial example $x' = x + \delta$

---

in Section. C.2. In the first stage, we focus on reducing the imperceptibility by setting the initial α to be 0.01 and the learning rate is set to be 1. We update the adversarial perturbation δ for 4000 iterations. If the adversarial example successfully attacks the ASR system in 4 out of 10 randomly chosen rooms, then α will be increased by 2. Otherwise, for every 50 iterations, α will be decreased by 0.5.

In the second stage, we focus on improving the less perceptible adversarial examples to be more robust. The learning rate is 1.5 and α starts from a very small value of 0.00005. The perturbation will be further updated for 6000 iterations. If the adversarial example successfully attacks the ASR system in 8 out of 10 randomly chosen rooms, then α will be increased by 1.2.

D. Transcription Examples

Some examples of the original phrases and targeted transcriptions from the LibriSpeech dataset (Panayotov et al., 2015) are shown in Table 4.

x	The clean audio input
δ	The adversarial perturbation added to clean audio
x'	The constructed adversarial example
y	The targeted transcription
$f(\cdot)$	The attacked neural network (ASR)
$\mathscr{F}(\cdot)$	Fourier transform
k	The index of the spectrum
N	The window size in short term Fourier transform
$s_x(k)$	The k-th bin of the spectrum for audio x
$s_\delta(k)$	The k-th bin of the spectrum for perturbation δ
$p_x(k)$	The log-magnitude power spectral density (PSD) for audio x at index k
$\bar{p}_x(k)$	The normalized PSD estimated for audio x at index k
$p_\delta(k)$	The log-magnitude power spectral density (PSD) for audio δ at index k
$\bar{p}_\delta(k)$	The normalized PSD estimated for audio δ at index k
$\theta_x(k)$	The frequency masking threshold for audio x at index k
$\ell(x, \delta, y)$	Loss function to optimize to construct adversarial examples
$\ell_{net}(\cdot, y)$	Loss function to fool the neural network with the input $(\cdot)$ and output y
$\ell_\theta(x, \delta)$	Imperceptibility loss function
α	A hyperparameter to balance the importance of ℓ_{net} and ℓ_θ
$\|\cdot\|$	Max-norm
ϵ	Max-norm bound of perturbation δ
$\nabla_\delta(\cdot)$	The gradient of $(\cdot)$ with regard to δ
lr_1, lr_2, lr_3	The learning rate in gradient descent
r	Room reverberation
$t(\cdot)$	The room transformation related to room configurations
$\mathcal{T}$	The distribtion from which the transformation $t(\cdot)$ is sampled from
δ_{im}^*	The optimized δ in the first stage in constructing imperceptible adversarial examples
ϵ_r^*	The optimized ϵ in the first stage in constructing robust adversarial examples
δ_r^*	The optimized δ in the first stage in constructing robust adversarial examples
ϵ_r^{**}	The max-norm bound for δ used in the second stage in constructing robust adversarial examples
δ_r^{**}	The optimized δ in the second stage in constructing robust adversarial examples
Δ	The difference between $\epsilon_r^{**} - \epsilon_r^*$
Ω	A set of transformations sampled from distribution $\mathcal{T}$
M	The size of the transformation set Ω

Table 3. Notations and Definitions used in our algorithms.

Original phrase 1	the more she is engaged in her proper duties the less leisure will she have for it even as an accomplishment and a recreation
Targeted phrase 1	old will is a fine fellow but poor and helpless since missus rogers had her accident
Original phrase 2	a little cracked that in the popular phrase was my impression of the stranger who now made his appearance in the supper room
Targeted phrase 2	her regard shifted to the green stalks and leaves again and she started to move away

Table 4. Examples of the original and targeted phrases on the LibriSpeech dataset.