Private Speech Adversaries

Lucy Chai **Thavishi Illandara** **Zhongxia Yan**

Abstract

The fact that deep learning models are surprisingly fragile – yielding drastically different outputs upon small changes in input – has been well studied in the image recognition space, but this property is shared across other input modalities. In the audio domain, minor perturbations to a waveform impact speech-to-text translation and allow an attacker to arbitrarily change a recognized phrase while remaining barely noticeable to an outside listener. We demonstrate that that these perturbations can also be used for good – by applying simple learned perturbations to audio signals, we are able to prevent successful transcription in speech-to-text systems; our approach defends against malicious eavesdropping and mass transcription of private audio content while still maintaining human interpretability of the signal. Furthermore, our learned perturbations, including a fixed universal perturbation, do not require optimizing towards a specific audio sequence or having the full audio recording available in advance. These perturbations are simple to apply in real-time and provide a step towards ensuring greater privacy in largely unmonitored digital environments.

1 Introduction

Advances in machine learning and deep learning have enabled the widespread use of software-based inference, classification, and decision making systems in many domains such as facial recognition, spam, malware filtering, and online fraud detection systems. As computing resources become cheaper, these applications have become more scalable, allowing organizations to apply these methods to large quantities of data. As a result, it is now possible for malicious parties to automatically collect and analyze large amounts of data that may jeopardize people's privacy, such as in illegal surveillance.

Speech-to-text (STT) software is one application enhanced by deep learning that may be used by malicious parties. For example, large communication companies or the National Security Agency can wiretap audio transmissions and use STT software to create searchable transcripts. Deep learning in this case enables automatic, passive, and programmable surveillance which do not require much human labor to be applied on a large scale.

Threat Model: In this paper we assume that there is an eavesdropper whose goal is to obtain text-searchable transcripts of many audio recordings, such as phone calls, in order to obtain certain relevant information from sparse targets within the recordings. As a substantial amount of phone calls take place at a given moment, it is impractical to employ humans to eavesdrop on or transcribe each individual call to determine the relevance of every segment. Therefore, an automated STT translation algorithm must be used. We assume that the eavesdropper is able to record waveforms and run the recordings through the STT algorithm to generate transcripts on a large scale. We also assume that the eavesdropper does not have active access to the users' recording devices.

Defending against these kinds of attacks requires processing speech waveform so that it appears significantly distorted to the STT system, but remains undistorted to a human listener (as we assume the recordings are meant to be understood by humans). Furthermore, the processing must be occur in real-time, so that conversations are not delayed.

Existing studies in adversarial attacks on deep learning models show that inputs into a machine learning system can be designed to manipulate the system's output Szegedy et al. [2013]. Adversarial

inputs can be created by introducing small perturbations that are imperceptible or hardly perceptible to humans. For instance, adversarial images have been generated to cause misclassifications or targeted classification in Xiao et al. [2018]. Traffic signs may be created to appear as stop signs to human eyes but speed limit signs to traffic sign recognition systems [Evtimov et al., 2017]. Carlini et al. [2016] manipulated voice interfaces by generating hidden voice commands that are unintelligible to human listeners. These adversarial attacks are examples of *white-box* adversarial methods, in that they assume that the attackers of the machine learning models have access to the model and can backpropagate gradients through the models. Addressing this assumption, Papernot et al. [2017] shows that under certain conditions, attacks can be designed in a *white-box* manner against substitute machine learning models, and then applied in a *black-box* manner to target models, without having full access to the target models.

We seek to apply an adversarial attack technique as a **defense** against the threat of eavesdroppers as described in the Threat Model. We design our attack in a "white-box" setting against a commonly used open-sourced STT system. Our contribution is a perturbation technique that can be applied to a speech waveform to prevent recognition by an STT algorithm without affecting human listeners. The perturbation should have a small magnitude that could either be simple function of the waveform, or a fixed universal perturbation. Therefore, this perturbation can be applied to the waveform in real-time, without the knowledge of the entire speech sequence in advance. We leave the task of addressing the *black-box* problem, that of designing a defense against arbitrary unknown STT systems, to future work.

2 Methods

We learn perturbation vectors against an open-source implementation of the DeepSpeech speech-to-text translation engine [Hannun et al., 2014][1]. Following the implementation by Carlini and Wagner [2018], we optimize a perturbation to transcribe an input audio waveform to a desired target phrase in a white-box setting on the DeepSpeech model. However, unlike the targeted adversarial attack demonstrated in Carlini and Wagner [2018], we also require that our perturbation can be easily applied in real-time deployment with minimal delays.

2.1 DeepSpeech Speech-to-Text Engine

The DeepSpeech transcriber [Hannun et al., 2014] is a five-hidden-layer bidirectional model trained end-to-end to recognize spoken words from an input audio spectrogram. The initial layer transforms the spectrogram into an intermediate feature representation using a fixed-width context window of neighboring timesteps, and is followed by two subsequent feed-forward layers. The fourth layer introduces bi-directional recurrent units, in which the hidden activations are are computed sequentially both forward in time from $t = 1$ to $t = T$ and backward from $t = T$ to $t = 1$. In the final layer, the model takes the result of both forward and backward activations and outputs a softmax distribution over character tokens, or a silence token, for each timestep.

The objective in speech-to-text translation is to predict the correct sequence of characters from an audio waveform, but the precise timestep when each character is spoken is not necessarily known in the training data. Therefore, the loss function must allow for flexibility in the alignment of output characters. DeepSpeech is trained with Connectionist Temporal Classification (CTC) loss [Graves et al., 2006], which disregards the exact position of each character in the output but retains ordinal information. Specifically, DeepSpeech outputs a sequence prediction with length proportional to the length of the input audio, e.g. one prediction for every 320 audio samples (0.02 second of audio), and this output sequence is always longer than the target text; each prediction is a probability distribution over characters a-z and silence token, with potentially multiple predictions corresponding to each token in the target.

To compute the CTC loss with respect to a target, we sum the total probability of all alignments of the target t to the prediction sequence y, which can be done efficiently with dynamic programming; the loss is simply the negative log of the total probability. Here $\Pi(t, y)$ denotes all possible alignments

[1] `https://github.com/mozilla/DeepSpeech`

of target t to sequence y, and y_π^i denotes the probability at the ith time step along a single alignment:

$$P(t|y) = \sum_{\pi \in \Pi(t,y)} \prod_{i=1}^{T} y_\pi^i$$
$$\mathcal{L}(y|t) = -\log P(t|y)$$

At decoding time, the target is not given, and it's intractable (exponential over sequence length) to iterate over all possible targets. This search is approximated more practically by either (a) greedily taking the most probable character at each prediction time step and removing extra characters or (b) using a beam-search decoding of fixed width to simultaneously evaluate a number of possible alignments and select the most likely alignment from these choices.

Furthermore, the decoding can be improved by querying a pretrained language model to help decided which combination of tokens to output, based on prior probabilities about co-occurrences in the language (English). We direct readers curious about implementation details of the language model to Hannun et al. [2014]. We only note that we utilize DeepSpeech *without* the language model during training time, but utilize DeepSpeech *with* the language model at evaluation time.

2.2 Learning a Perturbation Vector or Function

Following Carlini and Wagner [2018], we optimize for an perturbation vector which minimizes CTC loss toward a given target phrase, which we set to be silence. However, we also want to constrain the magnitude of the perturbation to prevent it from becoming arbitrarily large, which would make the perturbed audio waveform unintelligible. We formulate this loss as:

$$\begin{aligned} \min_{\delta} \quad & \text{CTC}(x+\delta, t) \\ \text{s.t.} \quad & ||\delta||_\infty < \tau \end{aligned} \tag{1}$$

for some target phrase t, perturbation δ, audio waveform x, and clipping threshold τ. A lower clipping threshold means that the perturbation will be less audible. However, this loss formulation used in Carlini and Wagner [2018] requires that the entire audio waveform be present at the time of loss minimization, so the perturbation cannot be applied in real-time.

Universal Perturbations Because we aim to find a perturbation which can be applied in real-time, we also need to enforce that the learned perturbation vector generalizes. The simplest way to accomplish this is to find a δ which fools DeepSpeech over multiple audio waveforms, which lets us apply a fixed sequence of perturbations to any audio waveform to prevent recognition by DeepSpeech. A perturbation of this form is analogous to universal adversarial perturbations in images, in which the same perturbation vector can be applied to any given image and cause it to be misclassified with high probability Moosavi-Dezfooli et al. [2017]. To this end, we minimize the expectation of CTC loss over the training data:

$$\begin{aligned} \min_{\delta} \quad & \mathbb{E}_x \left[\text{CTC}(x+\delta, t)\right] + \lambda ||\delta||_n \\ \text{s.t.} \quad & ||\delta||_\infty < \tau \end{aligned} \tag{2}$$

Here δ is some fixed length perturbation, whereas x is a continuous subsequence of the same length selected from an audio in the training set. At test time, if an input audio is longer than this fixed length, we repeatedly apply the fixed length δ to the audio.

The hyperparameter λ is the multiplier for L_n regularization to decrease the magnitude of δ, and we take $\lambda = 0$ unless otherwise mentioned.

Parameterized Perturbations An alternative approach is to parametrize the perturbation, allowing it to be a function of the input audio waveform. Here, we consider the perturbation to be an affine transformation of the input waveform, $\delta(x) = w * |x| + b$ where w and b are learned parameters of some fixed length, and $*$ and $+$ denotes element-wise multiplication and addition. As above, we still consider x to be a continuous subsequence of the same length as w and b. The modified objective becomes:

$$\begin{aligned} \min_{w,b} \quad & \mathbb{E}_x \left[\text{CTC}(x+\delta(x), t)\right] + \lambda ||\delta(x)||_n \\ \text{s.t.} \quad & ||\delta||_\infty < \tau \quad \text{and} \quad \delta(x) = w * |x| + b \end{aligned} \tag{3}$$

Entropy Objective The CTC loss objective compares a transcribed waveform with a target phrase of silence, but for our purposes of preventing accurate transcription, we do not necessarily need to silence the transcription. We investigate replacing the CTC loss with an entropy maximization objective over the prediction distribution of characters at each timestep. This objective reduces the log likelihood of any specific transcription sequence, making it more difficult for the decoding to output the correct sequence of characters when the perturbation vector is applied. Using the fixed perturbation, our loss objective in this case, which minimizes the negative entropy, becomes:

$$\begin{aligned} \min_{\delta} \quad & \mathbb{E}_{x,t}\left[\frac{1}{T}\sum_{t=0}^{T}\sum_{c} p_{c,t}(x)\log p_{c,t}(x)\right] \\ \text{s.t.} \quad & ||\delta||_{\infty} < \tau \end{aligned} \tag{4}$$

where t indexes over timesteps, c indexes over all character tokens in the vocabulary, and $p_{c,t}(x)$ refers to the DeepSpeech predicted probability of the c-th token at the t-th timestep on input waveform x. We could similarly use a parameterized form of δ rather than a fixed δ.

Optimization We optimize for the perturbation vector using a signed gradient approach – we compute the gradient of the loss with respect to the learned parameters, and then step the parameters according to the *sign* of the gradient during gradient descent rather than the magnitude, scaled by a specific step size per parameter (we consider a global learning rate of 1). To set our clipping threshold τ, we follow the method in Carlini and Wagner [2018] to first solve the optimization problem for a sufficiently large constant τ, and subsequently reduce τ when the optimization succeeds in finding a successful perturbation vector.

We learn the δ perturbation using the TIMIT corpus of spoken American English sentences from 630 speakers of eight dialects [Garofolo et al., 1993] and evaluate the success of our perturbation on a held-out validation set of audio waveforms: if the applied perturbation transcribes to silence on 80% of the validation audio recordings (for the CTC objective) or a BLEU score less than 0.1 (for the Entropy objective; see following evaluation section), we further reduce τ and continue optimization until no better solution is found within a maximum number of iterations.

Evaluation We consider a number of metrics for evaluating the effect of applying a perturbation vector on the resulting transcription. The first metric we use is the length of the transcribed sentence in terms of the number of characters. When we optimize for the CTC loss towards a silence target, sentences with *smaller* length are preferable; however, when with optimize for the Entropy objective, we do not expect a reduction in length of the transcribed sentence.

Secondly, we consider the Levenshtein (edit) distance between the transcribe sentences with and without the applied perturbation vector. Levenshtein distance measures the minimum number of insertion, deletion, and substitution operations to transform a source sentence (with the applied perturbation) to a target sentence (without the applied perturbation). A *larger* Levenshtein distance indicates a more successful perturbation.

We also use the BLEU score as an alternative approach of similarity, which yields a normalized measure of n-gram overlap between source and target sentences Papineni et al. [2002]. Compared to Levenshtein distance, the BLEU score is less sensitive to the precise ordering of the characters but also measures how similar a source sentence is to a given target. A *smaller* BLEU score indicates a more successful perturbation.

3 Experiments

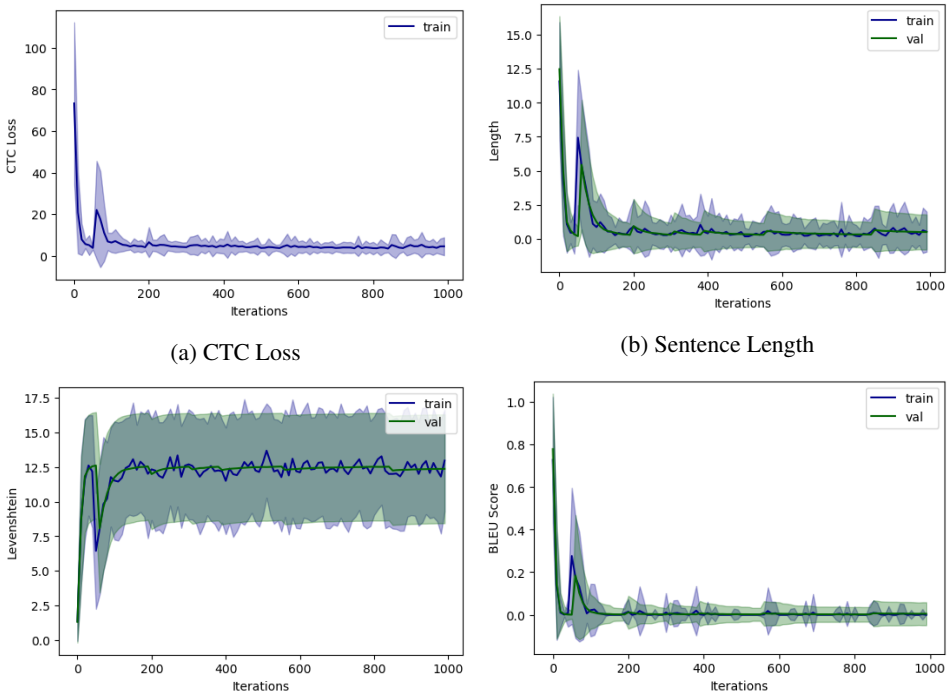
We split the TIMIT corpus into a training set of 4158 audio waveforms, an validation set of 462 waveforms, and a test set of 1680 waveforms. Each audio has a sample rate of 16kHz, ranges in length from 0.92 seconds to 7.78 seconds, and ranges in amplitude from -18000 to 18000. During training process, we split the audio from the training and validation sets into segments of 1 second (16000 samples). Thus, the δ in each universal perturbation and the w and b in each affine function perturbation are all length 16000.

We run the optimization for 1000 gradient update steps on the perturbation vector parameters with a learning rate of 1 (gradient step sizes for parameters specified for individual experiments) with

batches of 100 sample training segments. Notably, we run the optimization against the DeepSpeech model that *does not* incorporate a language model.

Fixed Noise Baseline. For a baseline, we generate a fixed perturbation δ of length 16000 by sampling amplitude values uniformly between -120 and 120.

Fixed Perturbation Optimizing CTC Objective. We optimize for the CTC loss objective as defined in (2), utilizing a step size of 10 for optimizing the δ. The perturbation successfully reduces the mean BLEU score of audio segments to around 0 during training (Figure 1). The constant δ produced by this vector has maximum magnitude of 119 and mean magnitude of 66, both of which are similar to the baseline noise.



(a) CTC Loss

(b) Sentence Length

(c) Levenshtein Distance

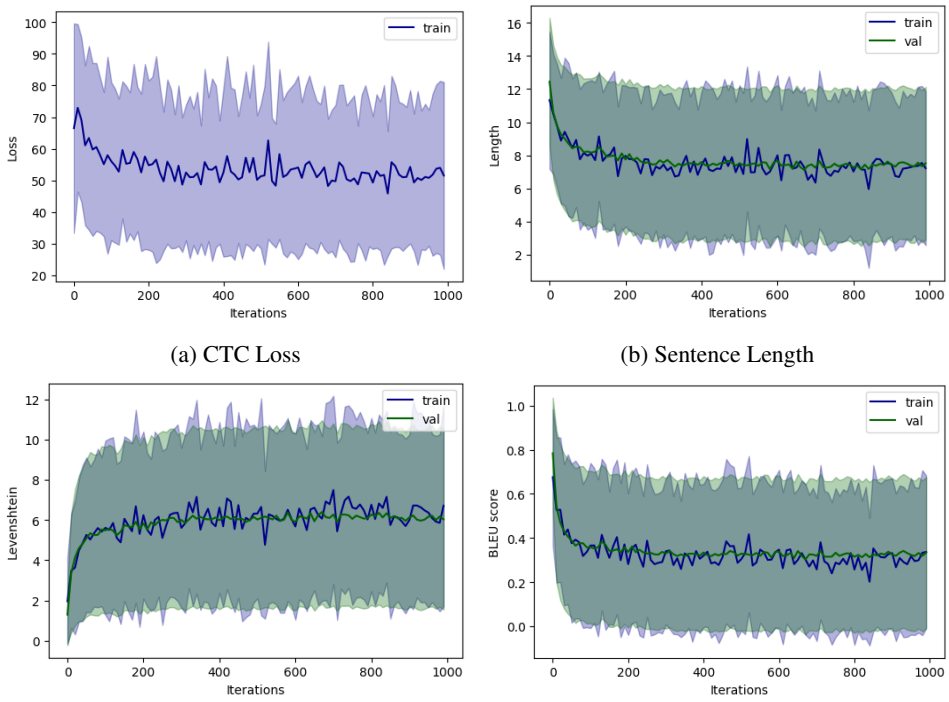
(d) BLEU Score

Figure 1: **Optimizing an perturbation vector to reduce CTC loss without regularization.** All metrics are computed on segments of training or validation data of length 16000. Note that the CTC loss to the silence target, BLEU score, and sentence approach 0 with more training iterations, which shows that the perturbations fools transcription by DeepSpeech at training time. The spikes (e.g. at iteration 50) in the metrics mark when the clipping threshold of the perturbation is decreased to force the perturbation to have lower magnitude.

Fixed Perturbation Optimizing CTC Objective, with Regularization. In an attempt to reduce the noise magnitude and promote noise sparsity, we experimented with applying L_1 and L_2 regularization. We ran experiments with the same hyperparameters as above, except with L_2 or L_1 regularization of multiplier 0.1 or 0.01. The regularization limits the effectiveness of the pertubation, as the mean BLEU score of the segment transcriptions during training does not approach 0 (Figure 2). While the maximum δ magnitude was 327, the mean magnitude was only 19, suggesting that the delta produced exhibits significant sparsity.

Constant Perturbation Optimizing the Entropy Objective. We optimize for the Entropy objective as defined in (4) with a constant perturbation, otherwise using the same hyperparameters as optimizing the CTC objective with no regularization. The δ has a maximum amplitude magnitude of 130 and a mean magnitude of 66, which is comparable to optimizing the CTC objective.

Affine Perturbation Optimizing CTC Objective, with Regularization. We optimize for the CTC objective with an parameterized affine perturbation as defined in (3). We specify a gradient update step size of 0.01 for w and 10 for b. We applied L_2 regularization with multiplier 0.01. The mean BLEU score on segments during training stayed between 0.1 and 0.2, which is similar to what we observed for the constant perturbation with regularization experiment.



(a) CTC Loss

(b) Sentence Length

(c) Levenshtein Distance

(d) BLEU Score

Figure 2: **Optimizing a perturbation vector to reduce CTC loss with L_2 regularization multiplier of 0.1.** All metrics are computed on segments of training or validation data of length 16000. Note that the CTC loss and BLEU score do not decrease to 0 here, which shows that regularization limits the effectiveness of fooling DeepSpeech with the perturbation.

4 Results

We apply the pertubation parameters (δ or w and b) that the training process produces to full audio waveforms from the training, validation, and test sets; we apply the perturbation to the input in segments of 16000. We use the DeepSpeech model *with* a language model to transcribe both the original audio and the perturbed audio, and then we compare the quality of original and perturbed transcription pairs. Unless otherwise stated, we discuss the results on the test set.

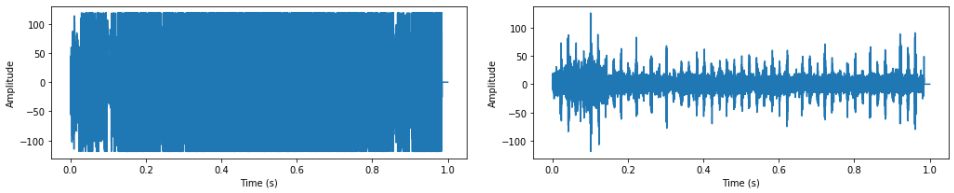
4.1 Baseline and Experiments without Regularization

The baseline noise does not perform well on average, as the mean BLEU score of the perturbed waveforms is 0.59 (Table 1). The noise sampled with this range of amplitudes also is audible to human ears. Optimizing for the CTC objective and Entropy objective with constant δ both produce perturbation vectors of similar magnitude as the random noise, but these perturbations are far more effective. For CTC objective, the perturbed transcriptions are on average only 0.12 times the length of the original transcriptions, suggesting that the perturbation successfully suppresses transcriptions. For the Entropy objective, the modified transcriptions are on average 1.2 times longer than the original transcriptions, suggesting that the perturbation does introduce some additional characters into the transcription. The BLEU score (mean of 0.026) for the optimizing the CTC objective is superior to both the noise and optimizing for the Entropy objective (mean of 0.10).

Although the CTC and entropy objectives greatly diminish transcription performace as desired, the perturbation is clearly audible to a human observer, especially on stretches of audio where the original amplitude is close to 0. We plot the amplitudes of the learned δ in Figure 3a.

Translated Sentence Length			
Partition	Baseline (↓)	Entropy (↑)	CTC (↓)
Train	0.90±0.10	1.21±0.25	**0.15±0.23**
Validation	0.90±0.13	1.21±0.31	**0.14±0.20**
Test	0.89±0.13	1.23±0.33	**0.12±0.19**
Levenshtein Distance (↑)			
Partition	Baseline	Entropy	CTC
Train	0.26±0.17	0.87±0.23	**0.88±0.17**
Validation	0.27±0.19	**0.89±0.26**	**0.89±0.16**
Test	0.28±0.19	**0.91±0.29**	0.90±0.15
BLEU Score (↓)			
Partition	Baseline	Entropy	CTC
Train	0.62±0.24	0.13±0.13	**0.045±0.11**
Validation	0.59±0.27	0.10±0.14	**0.027±0.091**
Test	0.59±0.26	0.10±0.14	**0.026±0.095**

Table 1: **Universal perturbations without regularization.** Evaluation of a fixed perturbation vector δ learned without regularization or distortion penalties. Length measures the number of characters in the transcribed waveform, Levenshtein distance measures the number of single-character edits between the trasncriptions of the original audio source and perturbed signal, and BLEU score measures normalized n-gram count similarity; ↑, ↓ indicate if higher or lower is better, and we report mean and standard deviation. Levenshtein distance and sentence length are normalized by the length of the original transcribed sentence. We compare the efficacy of δ vectors using a noise baseline, learned to maximize predicted softmax entropy, and learned to optimize CTC loss towards a silence target phrases.



(a) δ without L_2 (b) δ with L_2

Figure 3: **Learned perturbation vectors** using the CTC loss objective, with and without L_2 regularization on waveform distortion. Adding L_2 regularization increases the sparsity of the perturbation, making it less audible. We also experiment with L_1 regularization but find that it has a smaller effect on the transcribed sentence.

4.2 Experiments with Regularization

We next experiment with adding forms of regularization during the optimization process, with the aim of reducing audible distortion of the speech waveform – a distortion that is too large impacts the ability of a listener to understand the speaker, even though STT transcription fails.

Regularization reduces the effectiveness of the perturbation vector on transcription, but tend to reduce the audible noise in the signal (Figure 3b). L_2 regularization of multiplier 0.01 with constant δ performed the best, with perturbed waveforms having a mean BLEU score of around 0.19 (Table 2, compared to 0.59 for baseline and 0.02 for optimizing CTC objective with no regularization (Table 1). The produced waveforms still contain audible noise, but is less noticeable than transcriptions produced without regularization.

We do not report the results of optimizing the affine objective with regularization in tables, but we note that the mean BLEU score on full audio at evaluation time is around 0.35, which is significantly higher than the mean BLEU score during training (0.2 at most). This suggests that the affine perturbation with the current hyperparameters suffers from overfitting to the training set, and more tuning may be required to obtain more robust parameters.

Translated Sentence Length (↓)					
Partition	Baseline	L_2=0.1	L_2=0.01	L_1=0.1	L_1=0.01
Train	0.9±0.1	**0.5±0.3**	0.85±0.13	0.67±0.24	0.65±0.26
Validation	0.9±0.13	**0.44±0.29**	0.82±0.15	0.64±0.25	0.6±0.25
Test	0.89±0.13	**0.41±0.3**	0.82±0.16	0.62±0.27	0.58±0.28
Levenshtein Distance (↑)					
Partition	Baseline	L_2=0.1	L_2=0.01	L_1=0.1	L_1=0.01
Train	0.26±0.17	**0.58±0.26**	0.24±0.16	0.44±0.21	0.47±0.22
Validation	0.27±0.19	**0.63±0.26**	0.27±0.17	0.46±0.23	0.5±0.24
Test	0.28±0.19	**0.65±0.27**	0.27±0.18	0.48±0.25	0.51±0.25
BLEU Score (↓)					
Partition	Baseline	L_2=0.1	L_2=0.01	L_1=0.1	L_1=0.01
Train	0.62±0.24	**0.23±0.23**	0.64±0.26	0.37±0.27	0.35±0.25
Validation	0.59±0.27	**0.2±0.24**	0.6±0.25	0.35±0.28	0.32±0.27
Test	0.59±0.26	**0.19±0.25**	0.61±0.26	0.34±0.29	0.31±0.28

Table 2: **Universal perturbations with regularization.** Evaluation of a fixed perturbation vector δ learned with regularization and optimizing the CTC loss towards a silence target phrase. We obtain the best results using $L_2 = 0.1$ regularization across the sentence length, Levenshtein distance, and BLEU score metrics.

4.3 Generalization beyond TIMIT Corpus

To test whether our perturbations generalize to arbitrary human speech, we recorded nine audio clips of ourselves reading sentences from "The Devil's Due." using the microphone on a Macbook Pro. We compare the ground truth text to the sentence recognized by speech-to-text translation, before and after applying the fixed universal δ from optimizing the CTC loss with L_2 regularization of multiplier 0.1 (Table 3). Applying speech-to-text directly on the audio waveform transcribes the spoken text with high fidelity. However, adding the learned perturbation successfully prevents recognizable transcription of the audio recordings, even though δ was learned on the TIMIT dataset. We also encourage the reader to listen to the original and perturbed audio waveforms.

In Figure 4 we visualize one of our recordings and the associated spectrogram, both before and after application of the δ perturbation. A limitation of our fixed δ vector is that it is applied independently of silent regions in the source audio, and therefore will be more noticeable at those times. We may be able to mitigate this effect by dampening the magnitude of δ during sections of silence, during which the speech-to-text system should not be recognizing characters regardless. In the spectrograms the main vocal formants are still present in the modified audio signal, however the added perturbation introductes some high frequency noise artifacts. As a potential adversary may be able to detect and filter out these artifacts; evaluating whether the perturbations still work in the absence of these artifacts, is a topic for future investigation.

Ground Truth	They heard the sound of a can being kicked ahead of them, just around the corner at the end of the block
Transcription	they heard the sound of a canbikeepedthehead of them is round a corner at the end of the block
Perturbed	
Ground Truth	Then, there was the crashing sound of a bottle being smashed angrily against a wall.
Transcription	then there was the prasinsonoffobottlebesmashed angrily against the wall
Perturbed	then there was the prasingsoutovofbotobesmashedahvangrieagainsttewa
Ground Truth	A voice spoke out, "Not a god damn drop in that bottle. I need money damn it!"
Transcription	a as spoke not have got them drop in that bottle and need money demon
Perturbed	
Ground Truth	I ran to the corner store weaving my way through groups of children out trick-or-treating
Transcription	i ran to the corner or weaving a read through groups of chigranourtrick-ontreae
Perturbed	i fukoutnotocotkthermdropedinthetpottoeeheboytei
Ground Truth	I noticed that the groups of kids thinned out quickly, with almost none at all on the second block up
Transcription	i no it is that the group of kids kinedoquickly with almost on at all on the second looka
Perturbed	yes i am
Ground Truth	Goodbye sir, and thank you for the talk
Transcription	good by sir and thank you for the talk
Perturbed	i i
Ground Truth	Little did he know just how right he was
Transcription	little did he know just how right he was
Perturbed	it all
Ground Truth	she said nothing but still continued to be a loyal and loving wife
Transcription	she said nothing but still continued to be a loyal in loving life
Perturbed	hmm
Ground Truth	Heartless would have described me better
Transcription	heartless would have described me better
Perturbed	is it

Table 3: **Generalization of perturbation vectors.** We apply the fixed perturbation vector with L_2 regularization, which obtained best performance on the TIMIT dataset, to audio sequences that we recorded. We show the ground truth text that was read, speech-to-text transcription of the unmodified audio waveform, and the corresponding transcription after applying the perturbation.

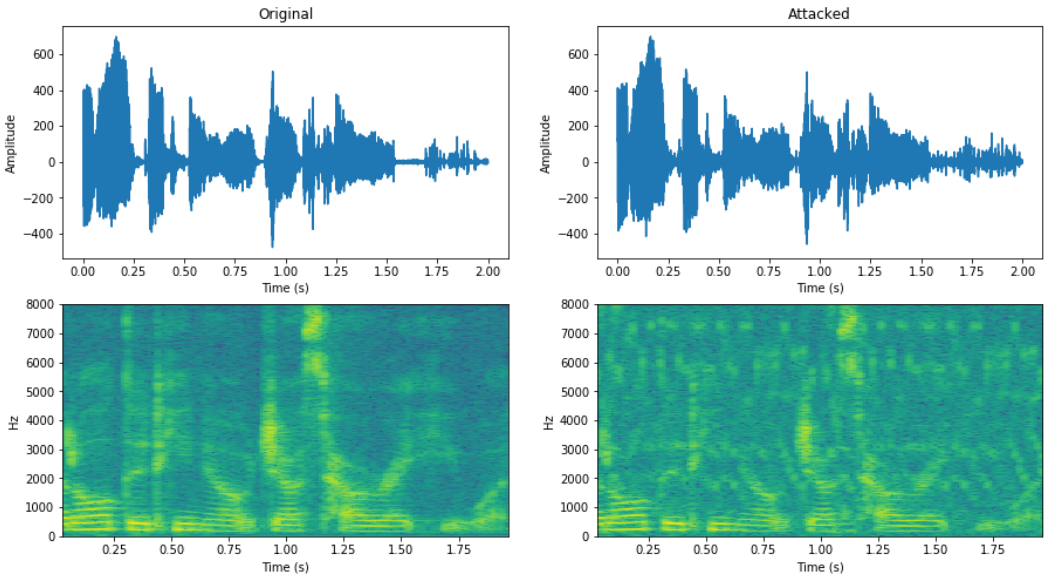


Figure 4: **Recorded Audio waveforms.** We compare the audio waveforms and spectrograms of our own recordings with and without the learned perturbation vector. The general form of the waveform and spectrogram remains similar, but note that a fixed perturbation vector will introduce noise in silent regions and some high-frequency artifacts.

5 Conclusion

Using the DeepSpeech speech-to-text engine in a *white-box* setting, we demonstrate that simple audio perturbations can be learned from data to prevent automated speech-to-text system from generating logical transcriptions. We experiment with various forms of perturbations: using fixed and parametrized vectors, Entropy and Connectionist Temporal Classification loss objectives, and L_1 and L_2 regularization to reduce the audibility of the perturbed signal. Our best results are obtained with a fixed universal perturbation and L_2 regularization, which generalizes to audio waveforms in a separate test set as well as to our own recorded signals. The perturbation can be applied in real-time without inducing signal transmission delays, and modified audio waveform still remains interpretable to a human listener. Our work could be further extended by investigating the effect of our learned perturbation (1) in black-box scenarios, where the weights and gradients of a speech-to-text model are unknown and (2) after high frequency artifacts of the perturbation are removed, particularly in silent sections of the input audio signal. However, these learned perturbations provide an initial step towards increasing digital privacy with respect to mass automation of speech-to-text transcriptions in illegal wiretapping and eavesdropping scenarios.

Acknowledgements. We thank Anish Athalye and the 6.858 course staff for helpful discussions and feedback, and Phillip Isola and Jonas Wulff for project inspiration. Much of our implementation is adapted from Carlini and Wagner [2018], whom we thank for providing the original codebase.

References

Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7. IEEE, 2018.

Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. Hidden voice commands. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 513–530, 2016.

Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. Robust physical-world attacks on machine learning models. *CoRR*, abs/1707.08945, 2017. URL `http://arxiv.org/abs/1707.08945`.

John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon technical report n*, 93, 1993.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM, 2006.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519. ACM, 2017.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *CoRR*, abs/1801.02610, 2018. URL `http://arxiv.org/abs/1801.02610`.