Untargeted Adversarial Attacks on Automatic Video Captioning

Aadarsh Pratik
Artificial Intelligence and Innovation
Carnegie Mellon University
Pittsburgh, PA 15213
apratik@andrew.cmu.edu

Ritika Dhiman
Artificial Intelligence and Innovation
Carnegie Mellon University
Pittsburgh, PA 15213
rdhiman@andrew.cmu.edu

Meiirbek Islamov
Department of Chemical Engineering
University of Pittsburgh
Pittsburgh, PA 15260
mei12@pitt.edu

Siddarth Achar
Computational Modeling and Simulation
University of Pittsburgh
Pittsburgh, PA 15260
ska31@pitt.edu

Raphael Olivier
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
rolivier@andrew.cmu.edu

Abstract

There is an increasing development of complex artificial neural network architectures to improve services in the form of automatic speech recognition. However, recent studies have shown that these networks can *malfunction* by adversarial examples. The adversarial examples operate the neural network model as intended by the adversary, thus leading to manipulation in prediction. Most attack strategies for speech recognition applications were performed for white-box models. However, these white-box attacks are not transferable to black-box models, which leads them to generate unimpressive attacks. One such black-box (Automatic speech recognition) ASR model is the YouTube captioning model. There are several successful black-box attack methods for image classification problems. One such attack methodology is the Meta-Gradient Adversarial Attack (MGAA) for image classification that was recently published. In this work, we performed studies of untargeted white-box and black-box adversarial attacks with the available ASRs. We used noisy audio files from different attack methods to understand the robustness of YouTube's audio captioning ASR. We found that that the white-box attacks resulted in high word error rates (WER). This was expected because the attacks were tailored to the a particular ASR and were tested on the same ASR. However, testing one of these white-box attacks on a different ASR resulted in a low WER, thus demonstrating poor transferability of these attacks. The MGAA on the other hand resulted in a much higher WER than the previous attempt of transfer attack. The MGAA's transferability was also effective with the YouTube's ASR, where the attacks generated by the white-box method were inferior to the MGAA attack for the same example.

GitHub link: https://github.com/apratik137/robust_speech_idl

Preprint. Under review.

1 Introduction

Adversarial attacks are intended to fool a machine learning (ML) system by perturbing the inputs in a way that affects the decision that the model makes while *trying* not to alter that of a human. These ML systems serve as the backbone for automating several services that require great attention to censorship and security. Some situations where the sensitivity of the model's decision-making ability is vital include speech recognition/ translation and face recognition. Studies have shown how vulnerable these models could be with carefully crafted inputs.[Biggio et al., 2013, Carlini and Wagner, 2017, Szegedy et al., 2013] An example of an *undesired* face recognition instance could be when a slight perturbation on the face image may trick the model to approve someone to access someone else's online bank account, among other examples [Goodfellow et al., 2014, Yuan et al., 2019]. An example of an *undesired* speech recognition instance could be when a small perturbation to the audio of a video uploaded on YouTube can generate misinterpreted closed-captions (subtitles). Therefore, it is essential that these systems are not susceptible to adversarial attacks. In this project, we focus on generating adversarial attacks on speech audio with the target application of caption generation. We will be making use of large repositories of automatic speech recognition models (ASRs) to develop attack methodologies.

Generating an attack can be classified based on the *model transparency*. When an adversary has access to the internals of an ASR system, such as the weights, architecture, etc., it is called a *White Box* attack. Generating a white box attack is often of less significance since the internals of an ASR system are generally hidden from the public. A more useful attack would be a *Black Box* attack, where the attacker only has access to the input-output pairs of the model and nothing else. The problem that we are trying to work on is of the second kind, a black box attack, since the details of YouTube's ASR are unavailable to the public. A major *road block* is that reliable strategies to attack black box ASR models have not been developed, and this opens up the avenue for us to focus on. Generating an attack can also be classified based on the *intent*. Models that are intended to make an ASR system generate the wrong output for a given audio sample are called *un-targeted* attack. These attacks are intended to just degrade an ASR system's performance. On the other hand, when the goal is to perturb the audio such that the ASR outputs a desired caption is called *targeted* attack. These attacks can tend to be more dangerous than un-targeted attacks at a time as they can manipulate actions taken by digital assistants such as Alexa. We focus on un-targeted attacks for this project as our main intention is to generate perturbed audio that would get misclassified by YouTube's ASR systems. We would also like to note that our project currently *does not* indent to develop defense strategies to make robust ASR from the attacks that we generate.

As mentioned earlier, black box attack methods are not very well established for ASRs. Before developing a functional black box approach, an obvious first step would be to test out white box models for currently available ASRs. We first used a white box attack method called the projected gradient descent (PGD) [Madry et al., 2017] for our audio examples using several ASR models. We demonstrate the influence of parameters such as "signal to noise ratio" (SNR) and the number of iterations of training the attack vector ($\mathrm{nb\_iter}$) on these white box attacks. We then performed a round of *grey-box* attacks where PGD attacks from one ASR model are tested on a different ASR model from our repository of models. We call them *transfer PGD* attacks. The reason we call these attacks as *grey-box* is that the ASR models being used to generate adversaries and the ones being tested have similar formalism; also, we have access to their architectures. The *hypothesis* for these two types of attacks is that the prediction errors generated by the regular white-box PGD attacks are expected to be higher than the transfer PGD. However, the *central hypothesis* of our project is that advanced attack strategies like meta-gradient adversarial attack (MGAA) [Yuan et al., 2021] can result in more severe attacks on black-box models as compared to the two attacks mentioned above. We have developed and formed similar experiments using the MGAA method for speech recognition. We have detailed discussions of each of these attack methods in the following sections. The quality of the attack can be regarded as *useful* if it is successful in manipulating an established and commonly used black-box model, which is YouTube's audio captioning for our project. Attacks generated by these three methods are tested on YouTube to find some degree of misclassification. We made use of Robust Speech [rob, 2022], which Raphael Olivier is currently developing at Carnegie Mellon University.

The report is structured as follows: we first discuss *Related Work* (literature review) pertaining to our project. Then we give a detailed mathematical description of the *Baseline Model*, which includes

PGD, transfer PDG, and MGAA. We also have a brief description of the ASRs used to generate and test the attacks. We then discuss the dataset involved in generating attacks and the evaluation metric used to understand the quality of attacks. We finally discuss the experiments we performed using PDG, transfer PDG, and MGAA on our dataset. The results from YouTube's prediction of our attacked audio files are reported, and conclusions are drawn from these results. Instead of this being just a comparative study, we also performed a grid search where we varied the SNR and the number of attack iterations for all three attack strategies.

2 Related Work

Recent work has seen rapid development of adversarial attack generation schemes where inputs are almost indistinguishable from natural data yet classified incorrectly by the neural networks.[Madry et al., 2017] proposed PGD, which is a robust first-order adversary to optimize saddle point formulation. Despite the fact that the optimization task involves maximizing of a non-concave function with many distinct local maxima, their values were highly concentrated. With respect to the strongest adversaries in the test suite, the best model of MNIST achieved an accuracy of 89%. In addition, the MNIST network is highly resilient to white-box attacks by an iterative adversary. [Carlini and Wagner, 2018] provides a white-box iterative optimization-based methodology for attacking DeepSpeech end-to-end in Mozilla and show that it is 99.9% successful. This requires optimizing through the MFC pre-processing transformation followed by a recurrent neural network using LSTMs. In addition, the paper proposes that speech-to-text systems can be attacked with targeted adversarial attacks with minimal distortion.

While prior work has focused on models using white-box knowledge, [Abdullah et al., 2019] specialized in attacks that target the pipeline rather than the models. They implement black-box attacks that are transferable and achieve mistranscription and misidentification rates in excess of 100% with only a few frames of audio. The attack exploits a fundamental difference between the human brain and automated speech recognition systems in how they process speech. [Qin et al., 2019] enhanced the construction of adversarial examples on ASR systems and used the psychoacoustic principle of auditory masking in order to develop effective imperceptible audio adversarial examples. As part of their experiment, they generate over-the-air adversarial samples in a virtual world after simulating a realistically distorted environment.

Furthermore, [Alzantot et al., 2018] also introduced a gradient-free genetic algorithm that added small background noise to targeted attacks to achieve 87% success without knowing the underlying model parameters. As a result of their attack, noise is only added to the smallest bits of a subset of audio clip samples and does not impact the human perception of the audio file. In Wav2Vec2, [Kim et al., 2020] proposed a novel adversarial attack that confuses the model about the instance-level identities of the perturbed data samples. The authors presented a framework for adversarially training robust neural networks without labeled data that maximizes the similarity between a random augmentation of a data sample and its instance-wise adversarial perturbation. Robust Contrastive Learning (RoCL) was validated on multiple benchmark datasets, on which it showed significant improvement in robustness against black-box attacks and unknown types of attacks.

Invoking meta-learning, [Yuan et al., 2021] proposes a plug-and-play method for improving the crossmodal transferability of gradient-based attack methods called Meta Gradient Adversarial Attack (MGAA). Based on extensive experiments on the CIFAR10 and ImageNet datasets, this architecture outperforms everything currently available in both black-box and white-box models for image datasets. We discuss the specifics of Meta-Gradient in the following sections.

3 Baseline Model

3.1 Projected Gradient Descent

As a baseline model, we first consider the PGD[Madry et al., 2017], which creates an adversarial attack and defense. The paper introduces the formulation of novel saddle point (min-max) optimization against the adversarial attack in a structured way. This formulation helps achieve a precise security guarantee, i.e., a comprehensive class of attacks the model can withstand instead of only a specific pre-determined attack. Particularly, training of adversarial examples corresponds to this saddle point optimization directly. The PGD attack can be thought of as a universal strongest "first-order

adversary" that utilizes the first-order gradient information of the neural network model. Furthermore, to train a robust model against any general class of the first order attacks, the network needs to have a larger capacity than a model that can only classify the benign examples correctly. This indicates that the decision boundary of the robust model can be more complicated to be able to correctly classify the adversarial examples. The paper uses the PGD robust optimization scheme to train adversarial examples and uses those examples to train a robust model for the image classifier. In this project, we intend to use the same PGD technique to generate adversarial examples for the ASR system to compare it with the newly developed Meta-gradient learning formalism. The saddle point optimization can be considered as the mixture of the inner maximization and the outer minimization problem. The purpose of the inner maximization is to determine the adversarial variant of the data, x, that gives a high loss, which is exactly equivalent to the notion of neural network adversarial attack. Whereas the goal of the outer maximization is to train the parameters of the model so that adversarial loss given by the adversarial example is minimized. This corresponds to creating a robust model using adversarial examples.

The inner maximization problem requires multi-step iterations/training to generate a powerful adversary, which is simply a PGD applied to the negative loss function, which is given by:

$$x^{t+1} = \prod_{x+S} (x^t + \alpha \operatorname{sgn}(\nabla_x L(\theta, x, y)))$$

In general, the saddle-point optimization requires solving both a non-concave inner maximization and the non-convex outer minimization problem. For defense, the PDG experiment on image datasets (MNIST and CIFAR10) revealed that the PGD could be robust against all first-order adversarial examples, which use only first-order information. In other words, it can be challenging to find an example using only the first-order information that can give a better local maximum than PGD.

Although the PGD technique can be great for training a robust model against a first-order adversary, the adversary generated using PGD might not be transferable to other unseen black box models. In order to create an adversary that is transferable, we plan to implement the MGAA learning algorithm for training an adversarial example to attack the ASR system. We label another form of attack called *Transfer PGD*, which is the same as a regular white-box PGD attack but is tested on a different ASR.

3.2 Meta-Gradient adversarial attack

The Meta-Gradient adversarial attack proposed by Qin et al. [Yuan et al., 2021] was used for image recognition. The idea is to include a set of white-box ASR models in the model zoo. For each task iteration, we plan on randomly sampling a set of these ASR models as part of meta-training and then use one of the ASR models as a simulated black-box model for meta-testing. MGAA consists of multiple iterations. A meta-task is composed by randomly selecting $n+1$ models from the model zoo for each iteration. There are two steps in each task: the meta-train step and the meta-test step. In order to generate perturbations, an ensemble of the n sampled models is used at the meta-train step, which can be repeated K times. As a basis for the meta-test step, the adversarial examples x_i, K obtained from the meta-train step are used, with the last sampled model being used to generate perturbations for adversarial attacks. In the final step, x_i , the final adversarial example following the i^{th} task is perturbed by the perturbation generated in the meta-test step $x_{i,mt} - x_{i,K}$. The meta-training is aimed to simulate the white-box attack, whereas the meta-testing phase is to simulate the black-box attack on the basis of the generated adversarial examples from the corresponding meta-training step. This approach is aimed at narrowing the gaps in gradient directions between the meta-train and meta-test steps. The attack examples that we generate after several iterations may not be biased to existing white-box models. Instead it generalizes for any unseen black-box model and thus transferable. Since our target application is to attack black-box models such as YouTube's video captioning, we believe this novel approach is pertinent to our target application.

In order simulate the white-box attack, a total of n models of M_{k_1}, M_{k_2},$\cdots$,M_{k_n} are used. The adversarial attack are performed by using the ensemble of n models, where the logits are given by:

$$l(x_{i,0}) = \sum_{s=1}^{n} w_s l_{k_s}(x_{i,0}),$$

where $l_{k_s}(x_{i,0})$ is the logits of the model M_{k_s}, w_s is each model's ensemble weight, where $w_s \geq 0$ and $\sum_{s=1}^{n} w_s = 1$. Then, the loss of misclassification is calculated using the cross entropy loss:

$$\zeta_{M_{k_1},M_{k_2},\ldots M_{k_n}}(x_{i,0}) = -1_y \cdot \log(\text{softmax}(l(x_{i,0}))),$$

where 1_y is the one-hot encoding of y. The updated rule for the adversarial example to maximize the loss functions is given by:

$$x_{i,j+1} = x_{i,j} + \alpha \cdot \text{sign}(\nabla_{x_{i,j}} \zeta_{M_{k_1},M_{k_2},\ldots M_{k_n}}(x_{i,j})),$$

In order to simulate the black-box attack using the examples $x_{i,K}$ generated during the meta-train step (K is the number of iterations), the final sampled model $M_{k_{n+1}}$ is used. Particularly the cross-entropy loss is calculated by model $M_{k_{n+1}}$:

$$\zeta_{M_{k_{n+1}}}(x_{i,K}) = -1_y \cdot \log(\text{softmax}(l_{k_{n+1}}(x_{i,K}))),$$

Then the adversarial example based on $x_{i,K}$ gets updated by:

$$x_{i,mt} = x_{i,K} + \beta \cdot \text{sign}(\nabla_{x_{i,K}} \zeta_{M_{k_{n+1}}}(x_{i,K})),$$

Finally, in order to update the adversarial example, the perturbation obtained in the meta-test step is added to the previously generated example x_i

$$x_{i+1} = x_i + (x_{i,mt} - x_{i,K}).$$

3.3 ASR models used

We used pretrained models to perform all of our attacks. These models were downloaded from the Huggingface [hug, 2022] website, which were pretrained on LibriSpeech (English) [Panayotov et al., 2015] within SpeechBrain [Ravanelli et al., 2021]. The four ASR used were; `asr-crdnn-transformerlm-librispeech`, `asr-crdnn-rnnlm-librispeech`, `asr-crdnn-transformerlm-librispeech-ctc`, and `asr-transformer-transformerlm-librispeech`. All models have similar formalisms as each of them are composed of 3 different but linked blocks:

1. The first block is a *tokenizer* (unigram) that transforms words into subword units and is trained with the "train" transcriptions of LibriSpeech.
2. The *neutral language model* that is trained on the full 10M words dataset of LibriSpeech. The `asr-crdnn-rnnlm-librispeech` uses an recurrent neural network language model (RNN-LM) [Mikolov, 2011], however, the other three used a transformer language model [Vig and Belinkov, 2019].
3. The *acoustic model* for the first three ASRs contain a CRDNN architecture [Xiang et al., 2021] that is made up of N blocks of convolutional neural networks with normalization and pooling on the frequency domain. A bidirectional LSTM with projection layers is connected to a final DNN to obtain the final acoustic representation that is given to the CTC and attention decoders. For the `asr-transformer-transformerlm-librispeech`, the acoustic model is made of a transformer encoder and a joint decoder with CTC and transformer together. This allows the decoding to incorporate the CTC probabilities.

All ASRs were trained with recordings sampled at 16kHz (single channel).

4 Dataset

We used the LibriSpeech dataset [Panayotov et al., 2015] to perform all of our experiments. The LibriSpeech is a corpus of approximately 1000 hours of 16kHz read English speech that is prepared by Vassil Panayotov with the assistance of Daniel Povey. The data is derived from read audiobooks from the LibriVox project and has been carefully segmented and aligned. The dataset contained `.flac` format audio files and corresponding transcriptions that are sorted based on `chapter_id`. We used the `test-clean` dataset that contains more than 2000 examples.

5 Evaluation Metrics

Choosing an evaluation metric is critical when it comes to measuring the accuracy of natural language predictions like words or characters. In the domain of automatic speech recognition, the word error rate (WER) [Morris et al., 2004] is widely used and has become a "standard" way of measuring how accurate a speech recognition model is. The WER for a prediction is calculated as,

$$\frac{N_S + N_D + N_I}{N_W}$$

where N_S is the number of substitutions, N_D is the number of deletions, and N_I is the number of insertions. The sum of these three quantities is then divided by the total number of words N_W. The metric stats called `ErrorRateStats` by SpeechBrain [Ravanelli et al., 2021] was used to compute the WER for all of our experiments. We also use another commonly used error metric called the character error rate (CER). This is similar to WER but is operated on characters instead of words.

6 Experiments

The attack is solely not governed by the method that is used to generate the attack. There are factors within the attack methodology that lead to an increase or decrease in the number of effects of an adversary. These attack parameters are common among the three attack methods that we performed experiments on.

Iterations of Training of Attack Vector

Projected Gradient Descent performs gradient descent to find the input vector maximises the loss function. By this logic , it follows that a higher number of iterations should result in a higher rate of misclassification or a higher word error rate. Both PGD and transfer PGD use essentially the same attack method, and the only difference is in how they are tested; thus, the meaning of the number of iterations on the attack vector is the same. MMGAA, on the other hand, consists of two nested `for` loops. The inner loop is responsible for perturbing the attack vector for the meta-train step, where an ensemble of the n sampled models is used to perform gradient-based attacks (here PGD) to generate perturbations. The outer loop of MGAA controls the meta-test step that uses the adversarial example from the inner loop (meta-train) as the basis and uses the previously sampled model to generate perturbations for adversarial attacks. The remaining steps are mentioned above for MGAA as explained in the Baseline Models section.

Signal to Noise Ratio

The Signal to Noise Ratio (SNR) is a measure that compares the level of the desired signal to the level of background noise. It is often measured on decibels (dB), and a ratio greater than 1 would indicate that there is more signal than noise. The larger the magnitude of SNR, the lower the noise. This is an important parameter to consider while testing out the PGD attack. The Projected Gradient attack looks for the best possible input (in an adversarial sense) within a limit of perturbation from the original input. This limit of perturbation is decided by the SNR. The experimental section is aimed to perform two objectives; the first is to verify our overall hypothesis of the project and the second is to contribute to the understanding of the sensitivities of hyperparameters that govern each of these attacks. Based on our initial understanding and our attempt to implement MGAA, we would expect the severity of MGAA attacks on black-box models to be higher as compared to the transfer PGD setup.

6.1 PGD

We first performed a preliminary round of experiments using the white-box PGD attack. The purpose of performing this experiment is to have an estimate of how effective the PGD attack methodology is. We performed attacks for three ASRs: `asr-crdnn-transformerlm-librispeech`, `asr-crdnn-rnnlm-librispeech`, `asr-crdnn-transformerlm-librispeech-ctc`. We changed two parameters mentioned above, SNR and number of iterations, to perform our experiments. Because PGD is a gradient-based attack method, the magnitude of predicted errors is expected to be high as compared to a black-box attack. We varied the SNR from 5 to 50 and the number of iterations from 5 to 200. We used 20 clips from the LibriSpeech `test-clean` repository to

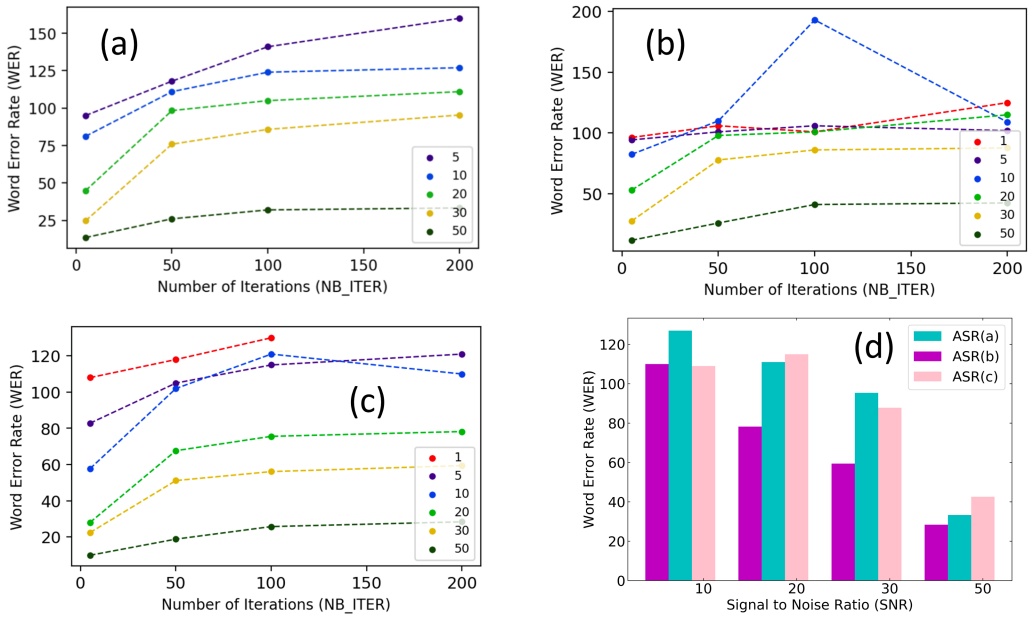

Figure 1: White box attacks: % WER calculated for attack experiments using different ASRs with PGD. (a) `asr-crdnn-transformerlm-librispeech-ctc`, (b) `asr-crdnn-rnnlm-librispeech` and (c) `asr-crdnn-transformerlm-librispeech`. (d) Bar plot that compares the equilibrated (highest number of iterations) % WER for each of the three ASR PGD attacks for varying SNR = 10, 20, 30, and 50.

generate attacks and perform tests. The %WERs were calculated for each set of the SNR and the number of iteration combinations. Figure 1a-c plots the results for all three ASRs. We observe a schematic increase in the WER with decreasing SNR. The extent of attack also seems to equilibrate with the increasing number of attack iterations, which is expected since the perturbation can only be projected within a constraint. Very small values of SNR (such as 1 and 5) lead to a lot of perceivable noise and thus may not be commonly used SNR values for adversarial attacks. We assume that a larger number of attack iterations (greater than 200) can only give WER close to that of the 200, which means that the attack may have reached its limit. We compare the WER for each of these attacks at their proposed limits in Figure 1d. These are labelled as ASR(a), ASR(b) and ASR(c) for `asr-crdnn-transformerlm-librispeech`, `asr-crdnn-rnnlm-librispeech`, and `asr-crdnn-transformerlm-librispeech-ctc`, respectively. Overall we find that ASR(a) resulted in less intense white-box attack as compared to the ASR(b) and ASR(c). We also see from this figure that the effects of the attacks tends to reduce with increasing SNR, which is expected. We plot spectrograms of one of the attacked examples along with the original audio and the amount of perturbation in Figure 2 using ASR(a).

6.2 Transfer PGD

These regular white-box PGD attacks can also be tested on other ASRs. For this experiment, we performed a similar grid search where we varied the SNR from 1 to 50 and the number of attack iterations from 0 to 200. We made use of two ASRs to generate our results. One is the source ASR, here `asr-transformer-transformerlm-librispeech`, and the other is the target ASR, here `asr-crdnn-rnnlm-librispeech`. The source ASR is used to perform PGD and generate attacked audio examples. These attacked examples are then tested on the target ASR, and the WER is computed. This gives us an understanding of how transferable a white-box PGD particular attack is on an unseen ASR. The results are plotted in Figure 3a. Though mild, the only appreciable manipulation of the target ASR can be observed for SNR=1 and SNR=5. Such low SNR at a high number of attack iterations is bound to produce a lot of noise that can lead to audio that is not perceivable to the human ear. An interesting result is the low WER observed when the SNR is greater than 10. This would be

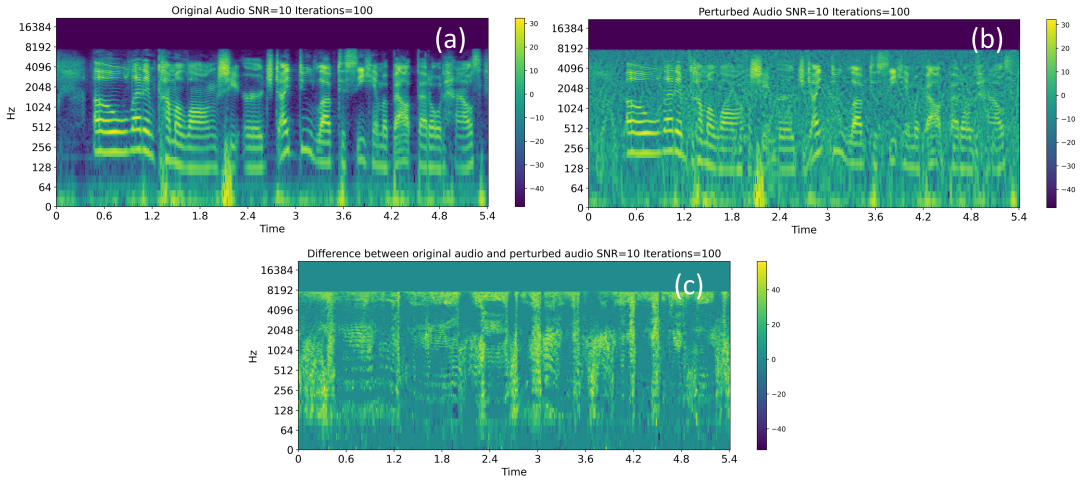

Figure 2: Spectrogram of original (a), perturbed (b) and the difference between original and perturbed audio files (c). The SNR used to generate the perturbed audio file is 10 and total number of iterations to perturb the audio is 100. These attacks are generated by PGD on the `asr-crdnn-rnnlm-librispeech` model.

the general range of adversarial attacks. The fact that these WERs are very low suggests that PGD is not very transferable.

6.3 MGAA

This brings us to the final attack method that is expected to work effectively on black-box models, which is MGAA. We first ran experiments like that was done for the attacks above. The SNR varied from 5 to 50, and the number of attack iterations varied from 1 to 100. Note that MGAA makes use of a “nested attack" mechanism. Thus the attacks take longer to complete for a single MGAA experiment. The target ASR was chosen to be the same target as that in the transfer PGD experiment, `asr-crdnn-rnnlm-librispeech`. We first used two source ASRs for the meta-train steps, `asr-transformer-transformerlm-librispeech` and `asr-crdnn-transformerlm-librispeech-ctc`. The source ASR does not contain the target ASR, making it a fair comparison to the results from the transfer PGD experiment. The results from this first MGAA experiment are plotted in 3b. Unlike PGD and transfer PGD, we observe that MGAA does not follow a trend with an increasing number of attack iterations. This can be attributed to the nested loops that are used to perform attacks and can thus lead to early “convergence" of attacks. The variation in the plots may be due to numerical differences that arise because of the smaller number of test examples that we used. The key point is that MGAA consistently produces higher WER for varying SNR than the transfer PGD experiment.

Increasing the number of models in the ensemble for the meta-train step should theoretically improve the generalization of the attack on black-box models. We performed another experiment where we increased the number of source ASRs to three for MGAA. The target was kept the same as the previous MGAA and the transfer PGD. We included the `asr-crdnn-transformerlm-librispeech` to the original list of source ASRs. This inclusion made the attack even more expensive to run; thus, we only varied the SNR and fixed the number of attack iterations to 10 for each source ASR. Figure 3c compares the three attacks, MGAA (with 2 sources ASRs), MGAA (with 3 source ASRs) and the transfer PGD experiment. Mean, and standard deviation data are plotted only for MGAA (with 2 sources ASRs) and the transfer PGD experiment since we had more data points (varied number of attack iterations). The MGAA (with 3 sources ASRs) results are just single points. As expected, the transfer PGD performed poorly compared to the other two MGAA experiments. There is no overlap of errors for the MGAA and transfer PGD experiments. However, the MGAA (with 3 sources ASRs) experiment seemed to result in a smaller WER than the MGAA (with 2 sources ASRs) experiment. This result was not as expected, which is that increasing the size of the ensemble should lead to a better quality of attacks. We feel this could be due to improper implementation of the MGAA

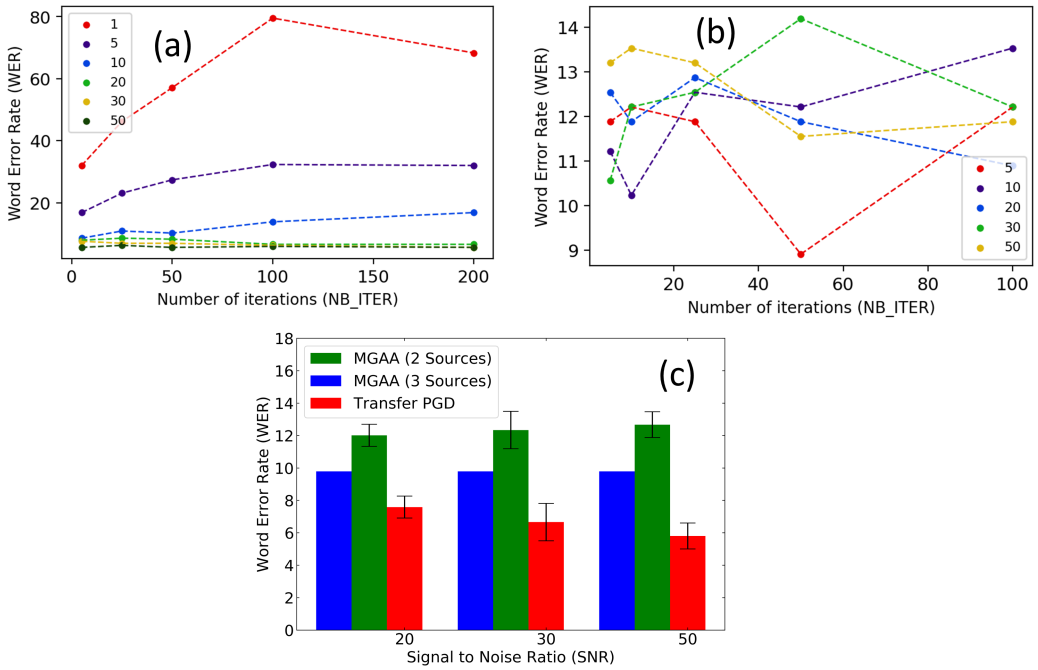

Figure 3: Transfer attacks: % WER calculated for attack experiments using (a) Transfer PGD, (b) MGAA (with source ASR brains). (c) Bar plots comparing the mean (with standard deviation) WER for transfer PGD and MGAA (2 source ASR brains) and MGAA (3 source ASR brains) for varying SNR = 20, 30 and 50.

Table 1: YouTube Transferability tests using MGAA and PGD attack.

Attack	YouTube Transcription
PGD	NOW WHAT HAVE YOU TO SAY CYNTHIA SPROG ISN'T HE THE GREATEST
MGAA	NOW WHAT HAVE YOU TO SAY CYNTHIA SPROUT ISN'T HE THE GREATEST

(with 3 sources ASRs) experiment, or it could be due to the smaller number of attack iterations used to generate our data. The latter may seem plausible because we know that longer attack iterations generally lead to better attacks. However, it was also seen that MGAA does not necessarily vary a lot with small changes in attack iterations, as seen in 3b. This aspect of the problem needs to be investigated further.

6.4 YouTube captioning

As discussed before, PGD is a directed attack that generates attacks specific to a particular ASR model. This should imply that the attack should have a relatively small effect on different ASR models. We have carried out attacks and tested them against YouTube's ASR model to test this. We compiled a few examples that resulted in the comparable WERs from our initial experiments and used those to generate transcriptions from YouTube's ASR. The link to the video can be found here. We found that a regular PGD was incapable of manipulating YouTube's ASR. However, the MGAA attack led to YouTube having one wrongly predicted word. Note that the MGAA (with 2 sources ASRs) was used to generate these adversarial examples.

7 Conclusions and Proposed Extension

We demonstrated that the regular white-box PGD is an effective way to generate adversarial speech examples. We performed a grid search over several parameters, such as the signal-to-noise ratio and the number of attack iterations, to understand the attack trends. We then performed similar transfer PGD tests where a white box PGD attack from one source ASR was tested against another

target ASR that varied in its core structure. We found that this particular attack method failed to generate transferable adversarial examples that could manipulate the target ASR. We then tested our hypothesis that a more transferable attack method, like the MGAA, should lead to more effective attacks on target ASR. We observed that MGAA gave a more consistent WER on test samples than transfer PGD. We used some of these perturbed clips to test against YouTube's ASR. In terms of higher WER, we found that MGAA performed better than regular PGD on YouTub's ASR. Before we can conclude that MGAA is the most effective black box attack method, it would be better to discuss possible reasons why it performed better and how one can make it even more effective. Using an ensemble of ASRs to generate attacks to examples and transfer those examples to the next iteration of MGAA attacks makes it very robust and generalized. However, the preliminary experiment that we ran showed unimpressive results when the number of source ASRs was increased to three instead of two. We will be investigating this particular result. We would want to work on improving our MGAA attacks by incorporating more ASRs and testing it on more extensive test examples. We will also want to compare our MGAA black box attack to other black box ASR attacks.

References

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

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.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2017. URL `https://arxiv.org/abs/1706.06083`.

Zheng Yuan, Jie Zhang, Yunpei Jia, Chuanqi Tan, Tao Xue, and Shiguang Shan. Meta gradient adversarial attack, 2021. URL `https://arxiv.org/abs/2108.04204`.

Robust speech. `https://github.com/RaphaelOlivier/robust_speech`, 2022. Accessed: 2022-04-04.

Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text, 2018. URL `https://arxiv.org/abs/1801.01944`.

Hadi Abdullah, Muhammad Sajidur Rahman, Washington Garcia, Logan Blue, Kevin Warren, Anurag Swarnim Yadav, Tom Shrimpton, and Patrick Traynor. Hear "no evil", see "kenansville": Efficient and transferable black-box attacks on speech recognition and voice identification systems, 2019. URL `https://arxiv.org/abs/1910.05262`.

Yao Qin, Nicholas Carlini, Ian Goodfellow, Garrison Cottrell, and Colin Raffel. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition, 2019. URL `https://arxiv.org/abs/1903.10346`.

Moustafa Alzantot, Bharathan Balaji, and Mani Srivastava. Did you hear that? adversarial examples against automatic speech recognition, 2018. URL `https://arxiv.org/abs/1801.00554`.

Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial self-supervised contrastive learning, 2020. URL `https://arxiv.org/abs/2006.07589`.

Hugging face. `https://huggingface.co/`, 2022. Accessed: 2022-04-04.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 5206–5210. IEEE, 2015.

Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio. SpeechBrain: A general-purpose speech toolkit, 2021. arXiv:2106.04624.

Tomas Mikolov. The rnnlm toolkit. In *RNNLM—recurrent neural network language modeling toolkit, IEEE Automatic Speech Recognition and Understanding Workshop*, 2011.

Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. *CoRR*, abs/1906.04284, 2019. URL `http://arxiv.org/abs/1906.04284`.

Yusheng Xiang, Tian Tang, Tianqing Su, Christine Brach, Libo Liu, Samuel S Mao, and Marcus Geimer. Fast crdnn: Towards on site training of mobile construction machines. *IEEE Access*, 9:124253–124267, 2021.

Andrew Morris, Viktoria Maier, and Phil Green. From wer and ril to mer and wil: improved evaluation measures for connected speech recognition. 01 2004.