Design of Adversarial Samples in ASR Systems

Jingyi Tian and Hongming Yu

University of Victoria

Abstract. In recent years, Automatic Speech Recognition (ASR) technologies which enable the conversion from human speech into text and thus facilitate human-computer interaction have drawn significant attention. However, the concerns on the security of the ASR systems have been ever growing. Some attackers can introduce malicious perturbations into the inputs of ASR, potentially affecting or manipulating the system's functions. Therefore, identifying and defending against such attacks is crucial. One effective approach is to understand the essence of these attacks from the attacker's perspective.

Currently, research in this area primarily involves adding perturbations to speech to generate adversarial audio samples. While these attacks often achieve a high success rate, most studies neglect the imperceptibility of such perturbations to human listeners. The ideal perturbation should be nearly imperceptible to humans yet highly disruptive to ASR systems, causing them to produce incorrect outputs. In our work, first we framed the generation of adversarial audio samples as a mathematical optimization problem. Then we focused on deter-

mining the optimal L_2 norm, which would render the generated perturbation akin to environmental noise. In this process, we employed the Projected Gradient Descent (PGD) method to iteratively create adversarial samples with the best performance. In addition, we set hyperparameters adaptively for the optimization problem to enhance effectiveness. We utilized the open-source ASR system DeepSpeech and conducted tests on the adversarial samples in a relatively quiet environment. The results of our experiments indicate that our method achieves a high attack success rate and shows promising performance SNR (Signal-to-Noise Ratio) and WER (Word Error Rate).

Keywords: ASR System, Adversarial Samples, Defense

1 Introduction

Automatic speech recognition (ASR) is a technology that enables the recognition and translation of spoken language into text by computers. With the proliferation of smart devices and advancements in machine learning technologies, Deep Neural Network (DNN)-based ASR systems are widely used to enhance human-computer interaction capabilities. Smart Home technologies, Siri, and Google Assistant are all good samples of the application of ASR.

However, research suggests that ASR systems can be susceptible to malicious attacks [1]. Specifically, one of the most state-of-the-art methods is to add per-

turbation to the audio input of ASR to cause the system to produce incorrect recognition output. Such audio inputs are also called audio adversarial samples. The adversarial attacks are posing a growing threat to users' data privacy, financial security, or even safety. Therefore, it is of great importance to understand the essence of audio adversarial samples and improve the system's resilience from a defender's perspective.

Essentially, the principle of adversarial sample creation primarily exploits the non-linearity and overfitting characteristics of machine learning models. Due to these characteristics, slight variations in the input data can cause significant changes in the model's classification or regression results, leading to erroneous outputs. Typically, adversarial audio samples have two requirements: imperceptibility to the human ear and erroneous recognition by ASR systems. For instance, as shown in Fig. 1, when the original audio clip conveys only the message 'You are so beautiful', a malicious attacker can introduce perturbations into this audio sample before it is input into the system. In this case, the audio heard by the human ear remains almost unchanged, but the text recognized by the ASR system changes to 'Pay bob 1000 dollars,' prompting the system to perform a transaction, thus causing financial loss to the user.

The academic and industrial communities have already conducted some research on adversarial samples. Essentially, adversarial samples can be classified into black-box attacks, white-box attacks, targeted attacks, and untargeted at-

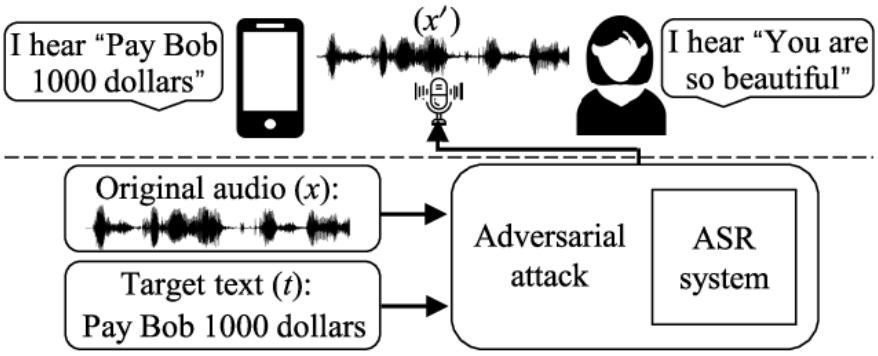

Fig. 1. An illustration of the adversarial attack against ASR system

tacks, etc. There are also algorithms for generating adversarial samples, such as the Fast Gradient Sign Method (FGSM) [2], Projected Gradient Descent (PGD) [3], and One Pixel Attack. Some researchers are investigating how to minimize the impact of adversarial attacks, with common methods based on input transformation or audio processing techniques, such as down-sampling, quantization, signal smoothing, filtering, and audio compression. However, these methods can easily be compromised under adaptive attacks. Therefore, adversarial sample attacks remain a potential threat in ASR systems, and analyzing and designing attack methods is crucial for enhancing the system's defense capabilities.

We noticed that most research on adversarial samples focuses on how to produce incorrect transcriptions in ASR systems, which is undoubtedly an important aspect. However, the imperceptibility of the perturbation is equally important in the sense of performing undetectable attacks. Therefore, we designed a method to generate adversarial samples for both targeted and untargeted attacks that ensures good imperceptibility to human ears while still causing the ASR system

to transcribe incorrectly. In the meanwhile, since our evaluation was conducted on the open-source ASR system DeepSpeech [4], it can be categorized as a white-box attack. Our contributions to the improved design of adversarial samples are summarized as follows:

- With the requirement of ensuring that the perturbations closely resemble environmental noise, we generated both targeted and untargeted adversarial samples that can lead to incorrect transcriptions in ASR systems.
- Using a real dataset, we tested these adversarial samples on the open-source ASR system DeepSpeech in terms of attack success rate, WER, and SNR, demonstrating the effectiveness and robustness of our solution.

The rest of this article is organized as follows: Section 2 introduces work and terms related to adversarial sample attacks, equipping readers with necessary background information. Section 3 delves into the details of our proposed solution. Section 4 discusses the experiment plan and presents the results. Finally, a summary is provided in the conclusion section.

2 Related Work

In this section, we will introduce some existing research related to adversarial samples.

White-box and Black-box Attacks. White-box attacks refer to scenarios where the attacker has full access to the ASR system they are targeting, including training

data, model structure, hyperparameters, activation functions, and model weights. Black-box attacks, on the other hand, occur when the attacker does not have access to the implementation details and parameters of the targeted system, and can only obtain the output of the model being attacked. Current academic research on white-box attacks includes gradient-based optimization attacks designed for the end-to-end ASR system DeepSpeech, with a success rate of 100% [5]; embedding malicious perturbations into popular music [6]; using impulse responses to simulate reverberation to increase the robustness of adversarial samples [7]; and minimizing the perceptibility of adversarial perturbations using psychoacoustic hiding [8]. Research on black-box attacks includes combining genetic algorithms with gradient estimation techniques using CTC [9]; using evolved multi-objective optimization methods to attack ASR systems in untargeted and targeted settings [10]. Improved optimization, genetic algorithms, and particle swarm optimization methods are used to attack black-box ASR systems. The scope of our work only involves white-box attacks.

Targeted and untargeted Attacks. Targeted attacks are those where the generated adversarial samples are misclassified by DNNs as a specific category, usually occurring in multi-classification problems. Untargeted attacks, on the other hand, do not have specific requirements for the transcription results of the generated adversarial samples; as long as the transcription is incorrect, the attack is considered successful, without restrictions on which category the adversarial samples

ultimately belong to. Because untargeted attacks offer more choices and a broader range of outputs, they are easier to implement than targeted attacks. Our design involves both types of attacks.

Generation Methods. The main methods for generating adversarial samples in white-box attacks include the Fast Gradient Sign Method (FGSM) [2], Projected Gradient Descent (PGD) [3], and One Pixel Attack [11]. FGSM is one of the most famous methods for untargeted attacks and can be directly extended to targeted attacks. Its basic idea is to enhance the linear behavior of DNNs in high-dimensional space to produce adversarial samples. It is a single-step optimization method of gradient ascent, with the optimization direction opposite to the direction of gradient descent of the trained model. As FGSM is a single-step method, it is relatively fast. PGD is an improvement of FGSM. FGSM assumes the target is a linear model, where the derivative of the loss function with respect to the input is fixed, and the direction of perturbation is clear. However, DNNs are nonlinear models, and the result generated in a single iteration is not necessarily correct. Therefore, PGD is a multi-iteration algorithm that adjusts the perturbation within a specified range in each iteration and can be seen as FGSM performed K times. One Pixel Attack, as the name implies, allows only one pixel of the image to be modified and uses differential evolution to maximize the difference between input and output. Due to the exhaustive nature of differential evolution, it is very inefficient and generally used only when gradient information cannot be obtained. In

our design, we used the PGD algorithm to generate adversarial samples, ensuring both good results and performance.

Defense Methods. Compared to the image field, there are fewer studies on defenses against adversarial attacks in the audio field. These mainly include down-sampling [12], quantization [13], signal smoothing [14], and audio compression [15]. We tested the attack performance based on down-sampling and quantization, so we will only introduce these two defense methods here. The principle of down-sampling is to reduce the sampling rate of the original signal, thus reducing the amount of sample data per unit of time, i.e., reducing the data volume of the original signal, thereby weakening the attack effect of adversarial samples. Quantization maps the original continuous speech signal to a set of discrete values, reducing the signal resolution to decrease the ASR system's sensitivity to perturbations, which can lead to information loss and reduced recognition accuracy.

3 Solution

3.1 Threat Model

In our work, we evaluate white-box attacks on open-source ASR models under both targeted (e.g., changing a password for a website account) and untargeted scenarios (e.g., obtaining incorrect transcriptions), through which attackers can engage in malicious activities. Here, we first analyze the generation strategies and scenarios of adversarial samples.

Initially, we established two scenarios, E_1 and E_2, set with different and small noise ranges: E_1 between $0dB - 20dB$, and E_1 between $20dB - 70dB$. We expect that if our adversarial samples can achieve good imperceptibility to human ears in milder noise environments, they will also perform well in the noisier environments, thus, we did not conduct further experiments in the higher noise settings. Next, we divided the generation of adversarial samples into two parts: machine transcription errors and human ear imperceptibility, using Connectionist Temporal Classification (CTC) [18] as the loss function and L_2 norm to measure, respectively. Hyperparameters are introduced to balance the weights of these two parts, transforming the generation of adversarial samples into an optimization problem to obtain optimal results, and adapting hyperparameters according to environmental thresholds. All tests were conducted on the ASR model DeepSpeech.

3.2 Problem Formulation

ASR Model The original audio signal is represented as x, and $f(\cdot)$ denotes the processing of the original signal by the ASR model based on DNNs. The correct transcription obtained by the ASR from the original x is denoted as $y = f(x)$.

Adversarial Samples A perturbation δ is added to the original audio signal x, resulting in $x' = x + \delta$, where x' is the adversarial sample we aim to create. The corresponding incorrect transcription is $y' = f(x + \delta) = f(x')$, with $y \neq y'$. For targeted attacks, the goal is to make $f(x + \delta)$ approximate a target t, i.e., $f(x + \delta) = t$. To ensure that the adversarial sample remains a valid audio signal,

we need to constrain the amplitude of the adversarial sample within a certain range, such that $-\varepsilon \leq x + \delta \leq \varepsilon$.

Ambient Noise The ambient noise is denoted as n. To approximate the perturbation δ to the environmental noise as closely as possible, we introduce the L_2 norm $\|\delta - n\|_2$ to control the difference between n and δ, ensuring it is less than the maximum amplitude threshold ε of the adversarial sample, i.e., $\|\delta - n\|_2 \leq \varepsilon$.

Therefore, given the original audio signal x and the ASR model $f(\cdot)$, we need to find a perturbation δ that satisfies the following constraints to obtain the adversarial sample $x' = x + \delta$:

$$\begin{aligned} & f(x+\delta) \neq f(x) \\ \text{such that} \quad & x + \delta \in [-\varepsilon, \varepsilon], \\ & \|\delta - n\|_2 \leq \varepsilon. \end{aligned} \tag{1}$$

Where $f(\cdot)$ represents an ASR system, and our motivation for L_2 norm is its desirable properties, including interpretability, cross-dimensional equilibrium distribution, continuity, and differentiability. The L_2 norm helps to make smoother modifications to the speech signal, thus mitigating the presence of perceived perturbation.

3.3 Targeted Attacks

For targeted attacks, our objective under scenarios E1 and E2 is to obtain a perturbation δ approximating the environmental noise n, such that the ASR's

incorrect transcription $f(x+\delta)$ closely matches the target transcription t. We formulate this problem as an optimization of the objective function for δ based on Equation (1). Our objective function is divided into two parts: (1) ensuring the adversarial sample's incorrect transcription $f(x+\delta)$ is as close as possible to the target transcription t; (2) ensuring the difference between the perturbation δ and the environmental noise n is minimized to enhance imperceptibility to human ears.

For the first part, we introduce the CTC loss function to measure this, denoted as l_{ASR}, and minimizing the difference between $f(x+\delta)$ and t can be represented as:

$$\begin{aligned} &\min l_{\text{ASR}}(f(x+\delta), t) \\ &\text{such that } x+\delta \in [-\varepsilon, \varepsilon] \end{aligned} \tag{2}$$

For the second part, we introduce the L2 norm, and the problem can be represented as:

$$\begin{aligned} &\min \|\delta - n\|_2 \\ &\|\delta - n\|_2 \leq \varepsilon \end{aligned} \tag{3}$$

To integrate these two parts into a single objective function, we use a hyperparameter α_1 to balance the weights of these two parts. Thus, finding the optimal perturbation δ for the adversarial sample is transformed into minimizing

the following objective function:

$$\min T = (1 - \alpha_1) \cdot L_{\text{ASR}}(f(x + \delta), t) + \alpha_1 \cdot \|\delta - n\|_2 \tag{4}$$

3.4 Untargeted Attacks

For untargeted attacks, the search for the perturbation δ can also be divided into two parts: (1) mislead the ASR system to obtain an incorrect transcription $f(x + \delta) \neq f(x)$, ensuring $f(x + \delta)$ is as dissimilar to $f(x)$ as possible; (2) ensure that the gap between the perturbation δ and the environmental noise n is as small as possible, to enhance the imperceptibility to human ears. The first part can be represented as $\max l_{\text{ASR}}(f(x + \delta), y)$, which we transform into a minimization problem:

$$\min -l_{\text{ASR}}(f(x + \delta), y) \tag{5}$$

such that $x + \delta \in [-\varepsilon, \varepsilon]$

The second part is the same as for targeted attacks. After integrating using the hyperparameter α_1, finding the optimal perturbation δ for the adversarial example is transformed into minimizing the following objective function:

$$\min UT = -(1 - \alpha_1) \cdot L_{\text{ASR}}(f(x + \delta), t) + \alpha_1 \cdot \|\delta - n\|_2 \tag{6}$$

3.5 Adversarial Sample Generation

We use the PGD algorithm to solve Equations (4) and (6). To ensure the effectiveness of the adversarial samples, we introduce the equation $\delta = \varepsilon \cdot \tanh(z)$,

which allows the adversarial perturbation δ to find its optimal value under the unconstrained optimization of z, ensuring that the final result falls within our expected range. Hence, the perturbation δ can be iteratively generated through the following equation:

$$\begin{aligned} \delta_0 &= 0 \\ \delta_{t+1} &= \varepsilon \cdot \tanh(\delta_t - \alpha \cdot \text{sign}(\nabla_x L(f(x), y))) \end{aligned} \tag{7}$$

where t is the iteration number and α is the learning rate. The complete process can be described by Algorithm 1.

Algorithm 1: Use PGD method to generate adversarial samples

Input : Speech signal x, target adversarial transcription t, ASR system $f(\cdot)$, hyperparameters a_1, learning rate α, number of steps *steps*

Output: Adversarial samples $x + \delta$

Function `GENERATE()`

    Initialize: $z \leftarrow 0$;

    for $t = 1$ *to steps* **do**

        $l \leftarrow (1 - a_1) \cdot l_{ASR}(f(x + \delta), t) + a_1 \cdot \|\delta - n\|_2$;

        $z \leftarrow z - \alpha \cdot sign(\nabla_z l)$;

        $\delta \leftarrow \epsilon \cdot tanh(z)$;

3.6 Hyperparameter Adaptive

The problem of determining the hyperparameter α_1 in the objective function is resolved by considering the distance between the background noise and the en-

vironmental threshold. As previously mentioned, the two environments we tested are:

$$\text{Scenario} \begin{cases} \theta_1 \leq E_1 \leq \theta_2 \\ \theta_0 \leq E_2 \leq \theta_1 \end{cases}$$

where $\theta_0 = 0\text{dB}, \theta_1 = 20\text{dB}, \theta_2 = 70\text{dB}$. Using bg to represent the background noise and $\max_j(\theta_j)$ to denote the threshold of the environment, we can calculate the distance between them as:

$$ds = |bg - max_j(\theta_j)| \tag{8}$$

When the scenario background approaches the threshold, that is, as the background noise increases, the value of α_1 nonlinearly decreases with the reduction of ds. This results in a reduced weight for the adversarial perturbation δ in approximating the environmental noise n. Here, we use the sigmoid function [16], obtaining the following expression to solve for the value of α_1, allowing α_1 to adaptively change with the environment:

$$a_1 = \begin{cases} 1 - \frac{0.5}{1+e^{-\left(bg\frac{1}{2\left(max_j(\theta_j)\right)}\right)}}, bg \in (\theta_0, \theta_1] \\ 0.5 - \frac{0.5}{1+e^{-(bg-1/2(max_j(\theta_j)))}}, bg \in (\theta_1, \theta_2] \end{cases} \tag{9}$$

4 Experiment

4.1 Experimental Setup

Settings Our experiments were conducted on a host equipped with four GeForce RTX 2080 Ti GPUs, with the Ubuntu 18.04 LTS operating system installed. The Python scripts and TensorFlow were utilized to implement our approach. DeepSpeech 0.4.1 was employed as the ASR model.

Dataset LibriSpeech [17] was chosen as the target dataset. It is a corpus of approximately 1000 hours of read English speech with sampling rate of 16 kHz. Its dev-clean subset comprises a total of 2,703 samples, from which we randomly selected 500 samples for our experiments. These samples have an average duration of 3 seconds and can be transcribed into an average of 9 words.

4.2 Evaluation metrics

Word Error Rate (WER). The word error rate (WER) is used to measure the error rate of the predicted text compared to the standard text and is an important indicator for evaluating ASR systems. The gap between these two texts can be calculated using the Levenshtein distance algorithm, with the formula as follows:

$$WER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C}$$

where S, D, and I respectively represent the number of substitutions, deletions, and insertions needed to transform the predicted text into the standard text.

N represents the total number of words in the standard text. C represents the number of words correctly recognized in the predicted text.

Signal-to-noise ratio (SNR). Signal-to-Noise Ratio (SNR) refers to the ratio of the perturbation δ to the original signal x, used to quantify the amount of disturbance added to the signal, measured in decibels (dB). It can be calculated by the following formula:

$$SNR(dB) = 10 \cdot \log_{10} \left(\frac{P_x}{P_\delta} \right)$$

where P_x and P_δ represent the energies of the original signal and the perturbation, respectively. It can be observed that the higher the SNR ratio, the less distortion is caused by the perturbation.

Success rate of attack (SRoA). The Success Rate of Attack (SRoA) represents the ratio of successful attacks to the total number of attacks. Clearly, the higher the success rate, the better the attack algorithm. It can be calculated using the following formula:

$$SRoA = \frac{N_s}{N_a}$$

where N_a is the total number of audio adversarial samples input into the ASR system, and N_s is the number of audio adversarial samples that are mistranscribed by the ASR system. SRoA is one of the most widely used evaluation metrics in the study of audio adversarial samples. For our work, we aim to achieve a higher SRoA to demonstrate the effectiveness of our adversarial samples.

4.3 Results and Discussion

Attack Performance We use the PGD method to generate adversarial samples, and it can be observed that a parameter ε is used to restrict the amplitude of the adversarial samples in Equation (7). This parameter not only limits the amplitude of the adversarial samples within a certain range to ensure their effectiveness but also limits the similarity between the perturbation δ and the environmental noise n.

We gradually increased the value of ε in steps of 5 and tested the generation time of adversarial samples, SNR, WER, and SRoA under different values of ε to determine its optimal value, with results shown in Tables 1 and 2. Table 1 shows the performance of various metrics under different values of ε in the case of targeted attacks. Due to space limitations, we only display a portion of the results near the optimal value of ε. It can be seen that when ε is set to 20, the generation time of the adversarial samples is approximately 274.05s, the shortest, with SNR at 38.83 dB and WER at 199.47% being the highest, and the success rate of attack also reaching 100%, which is superior compared to other values of ε.

Table 1. Different ϵ under targeted attack

ϵ	Time(s)	SNR(dB)	WER(%)	SRoA(%)
5	276.95	32.04	188.55	96
10	276.34	35.61	194.31	99
20	274.05	38.83	199.47	100
30	278.66	36.17	198.67	100
40	275.14	31.68	190.15	100

Table 2 shows the performance of various metrics under different values of ε in the case of untargeted attacks. It is evident that when ε is set to 70, the generation time of the adversarial samples is approximately 277.17s, with SNR at 33.25 dB and WER at 140.4% being the highest, and the success rate of attack also being 100%. Although the generation time is relatively longer, to achieve the best performance of adversarial samples in terms of SNR, WER, and SRoA, we believe that the optimal value of ε is 70.

Table 2. Different ϵ under untargeted attack

ϵ	Time(s)	SNR(dB)	WER(%)	SRoA(%)
50	280.42	29.89	132.35	97
60	275.03	32.01	138.33	100
70	277.17	33.25	140.4	100
80	271.86	28.29	138.92	100
90	274.59	26.25	135.55	100

When we determined the optimal values of ε as 20 for targeted attacks and 70 for untargeted attacks, we retested the aforementioned metrics in environments E_1 and E_2. The results are shown in Tables 3 and 4. In both environments, all metrics performed well. Time, SNR, and SRoA were not significantly different from the previous results. However, since the environmental noise was restricted to a smaller range, the interference to the ASR system was also reduced, leading to a slight decrease in the WER values.

Hyperparameter a_1 In addition to the parameter ε, we also introduced a hyperparameter α_1 to balance the weights of the two parts of the objective function,

Table 3. Targeted attack at $\epsilon = 20$

	Time(s)	SNR(dB)	WER(%)	SRoA(%)
E2	298.25	38.83	140.29	100
E1	270.29	38.83	130.06	100

Table 4. Untargeted attack at $\epsilon = 70$

	Time(s)	SNR(dB)	WER(%)	SRoA(%)
E2	275.25	33.25	135.59	100
E1	274.69	33.25	138.77	100

with its value varying according to a segmentation function based on the scenario background. Therefore, we also tested the impact of α_1 on WER and SNR. Fig. 2 and 3 show the impact of α_1 on WER during targeted attacks in environments E_1 and E_2, respectively. Fig. 4 and 5 respectively show the impact of α_1 on SNR during targeted attacks in environments E_1 and E_2.

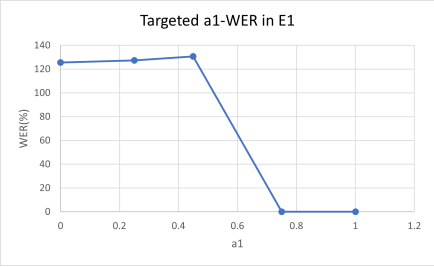

Fig. 2. Targeted a_1-WER in E_1.

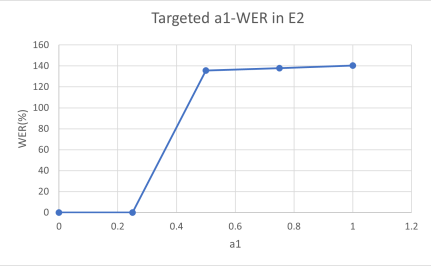

Fig. 3. Targeted a_1-WER in E_2.

Fig. 6 and 7 respectively illustrate the impact of α_1 on WER during untargeted attacks in environments E_1 and E_2. Fig. 8 and 9 respectively demonstrate the impact of α_1 on SNR during untargeted attacks in environments E_1 and E_2.

The results indicate that the impact of α_1 remains relatively stable across different scenarios. The WER metric maintains an average of 130%, while the

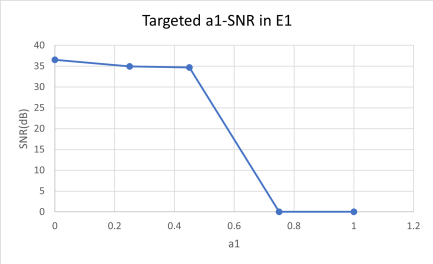

Fig. 4. Targeted a_1-SNR in E_2.

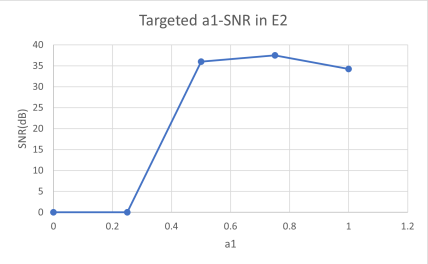

Fig. 5. Targeted a_1-SNR in E_2.

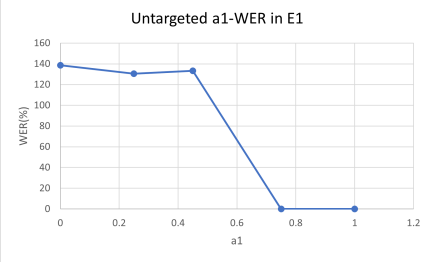

Fig. 6. Untargeted a_1-WER in E_2.

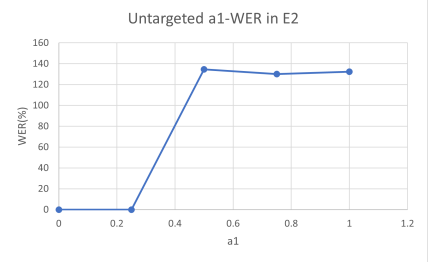

Fig. 7. Untargeted a_1-WER in E_2.

Signal-to-Noise Ratio (SNR) metric stays around 36%. The fluctuation in these values has a consistent effect on the efficacy of the attacks within a reasonable range. Additionally, the adaptive variation of the α_1 value in different scenarios achieves optimal concealment of the perturbation in the given environment.

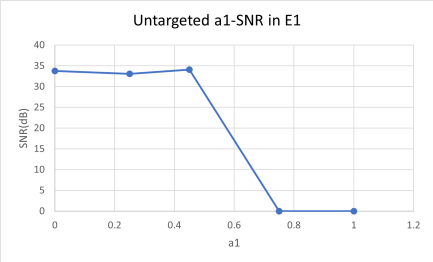

Fig. 8. Untargeted a_1-SNR in E_2.

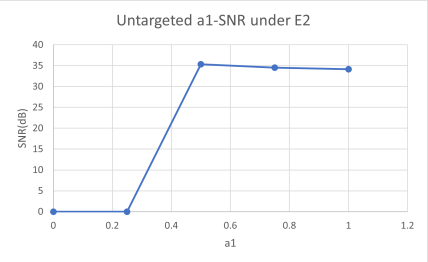

Fig. 9. Untargeted a_1-SNR in E_2.

Defense Robustness We evaluated the attack performance of adversarial samples against common defense methods such as down-sampling and quantization to determine their robustness against these defenses.

As shown in Table 5, to assess the robustness of our adversarial samples against defense methods, we performed down-sampling of the input audio signals at different sampling rates (5.6KHz and 6.4KHz). We also conducted quantized audio preprocessing using different quantization values (256, 512). The WER, SNR, and SRoA of the adversarial samples were tested under these two defense methods.

Table 5. Performance of adversarial samples under defense

	Down-Sampling		Quantization	
	5.6KHz	6.4KHz	256	512
WER(%)	92.77	95.36	105.14	100.81
SNR(dB)	12.03	14.81	15.15	13.59
SRoA(%)	100	100	100	100

The results indicate that for the down-sampling method, WER decreased by approximately 17%, and SNR decreased by about 15%. For the quantization method, WER decreased by approximately 8%, and SNR decreased by about 11%. However, successful attacks with a 100% success rate could still be achieved under these defense measures, indicating that our adversarial sample scheme maintains good robustness even in the presence of certain defense mechanisms.

5 Conclusion

Research shows that ASR systems are susceptible to adversarial attacks, which means that the privacy and financial security of ASR users are at risk. Analyzing and enhancing adversarial attacks is crucial for improving the defense capabilities of ASR systems. Our work focused on generating enhanced adversarial samples. Specifically, we have framed the generation of adversarial samples as a mathematical optimization problem, searching for the optimal solution while restricting the difference between the perturbation and environmental noise. Experimental results indicate that our designed adversarial samples can successfully attack ASR systems, showing good performance in metrics such as SNR and WER. Additionally, we have verified that some common defense schemes are ineffective against our adversarial samples, demonstrating their robustness.

6 Individual Contributions

Jingyi Tian

- Design scheme and develop solutions
- Conduct experiments
- Draft this report

Hongming Yu

- Set up experimental environment

– Conduct experiments

– Create report charts and graphs and compile report

References

1. Bock, Kevin, et al. "unCaptcha: A Low-Resource Defeat of reCaptcha's Audio Challenge." 11th USENIX Workshop on Offensive Technologies (WOOT 17). 2017.
2. Madry, Aleksander, et al. "Towards deep learning models resistant to adversarial attacks." arXiv preprint arXiv:1706.06083 (2017).
3. Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).
4. Hannun, Awni, et al. "Deep speech: Scaling up end-to-end speech recognition." arXiv preprint arXiv:1412.5567 (2014).
5. Carlini, Nicholas, and David Wagner. "Audio adversarial examples: Targeted attacks on speech-to-text." 2018 IEEE security and privacy workshops (SPW). IEEE, 2018.
6. Yuan, Xuejing, et al. "CommanderSong: A systematic approach for practical adversarial voice recognition." 27th USENIX security symposium (USENIX security 18). 2018.
7. Yakura, Hiromu, and Jun Sakuma. "Robust audio adversarial example for a physical attack." arXiv preprint arXiv:1810.11793 (2018).

8. Schönherr, Lea, et al. "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding." arXiv preprint arXiv:1808.05665 (2018).

9. Taori, Rohan, et al. "Targeted adversarial examples for black box audio systems." 2019 IEEE security and privacy workshops (SPW). IEEE, 2019.

10. Khare, Shreya, Rahul Aralikatte, and Senthil Mani. "Adversarial black-box attacks for automatic speech recognition systems using multi-objective genetic optimization." arXiv preprint arXiv:1811.01312 (2018).

11. Su, Jiawei, Danilo Vasconcellos Vargas, and Kouichi Sakurai. "One pixel attack for fooling deep neural networks." IEEE Transactions on Evolutionary Computation 23.5 (2019): 828-841.

12. Hussain, Shehzeen, et al. "WaveGuard: Understanding and mitigating audio adversarial examples." 30th USENIX Security Symposium (USENIX Security 21). 2021.

13. Yang, Zhuolin, et al. "Characterizing audio adversarial examples using temporal dependency." arXiv preprint arXiv:1809.10875 (2018).

14. Eisenhofer, Thorsten, et al. "Dompteur: Taming audio adversarial examples." 30th USENIX Security Symposium (USENIX Security 21). 2021.

15. Das, Nilaksh, et al. "Adagio: Interactive experimentation with adversarial attack and defense for audio." Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part III 18. Springer International Publishing,

2019.

16. Han, Jun, and Claudio Moraga. "The influence of the sigmoid function parameters on the speed of backpropagation learning." International workshop on artificial neural networks. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995.
17. Panayotov, Vassil, et al. "Librispeech: an asr corpus based on public domain audio books." 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2015.
18. Graves, Alex, et al. "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks." Proceedings of the 23rd international conference on Machine learning. 2006.