DEGREE PROJECT IN ELECTRICAL ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, *SWEDEN 2021*

A General Approach to Inaudible Adversarial Perturbations in a Black-box Setting

JOHAN SÖRELL

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

A General Approach to Inaudible Adversarial Perturbations in a Black-box Setting

JOHAN SÖRELL

Master in Information and Network Engineering
Date: 11 January, 2021
Supervisor: Borja Rodríguez Gálvez
Examiner: Ragnar Thobaben
School of Electrical Engineering and Computer Science
Swedish title: En Generel Metod för att Generera Ohörbara Kontradiktoriska Störningar på Okända Klassificeringssystem

Acknowledgement

I would first like to thank my supervisor, PhD student Borja Rodríguez Gálvez who offered guidance and encouragement during the whole project. Our discussions were always interesting and fulfilling while simultaneously challenging me to further increase my understanding of the problem explored in the thesis. His insightful feedback was invaluable for improving the quality of the thesis.

I would also like to thank my examiner Prof. Ragnar Thobaben for thoroughly going through the thesis and offering thoughtful and useful feedback, which was very helpful during the final thesis work.

Abstract

Deep learning is currently being deployed in many speech recognition systems. While these systems can achieve state-of-the-art performance, they are known to be susceptible to adversarial perturbations. These are minor perturbations to the input data, crafted specifically to cause erroneous behavior from the system. Some previous work have put effort into placing the perturbations in accordance with psychoacoustics, meaning placing the perturbations in areas of a signal that are perceptually limited for humans. In this work, a general method for optimizing perturbations according to psychoacoustics is presented. The formulation allows for a non-gradient based optimization strategy to be implemented. Two greedy optimization algorithms are developed using the proposed method. Inaudible perturbations are shown to be ineffective, which conform with the current academic understanding. However, when allowing the perturbations to be 18 dB stronger than the psychoacoustical defined perceptual limit, targeted success-rate of 64% and untargeted success-rate of 87% is achieved on a keyword spotting task.

Sammanfattning

Djupinlärning är idag den föredragna metoden för att skapa taligenkänningssystem. Det har dock visat sig att denna typ av system är känsliga för små specialgjorda störningar, så kallade kontradiktoriska störningar (adversarial perturbations). Dessa är specifikt genererade för att orsaka en felaktig respons från systemet. Tidigare forskning har försökt att anpassa de kontradiktoriska störningarna för att minimera den subjektiva inverkan på ljudet. Detta har gjorts genom att modellera den mänskliga perceptionen av ljud, vilket kallas för psykoakustisk modellering. I det här arbetet utvecklas en ny metod för att optimera störningar i enighet med en psykoakustisk modell. Metoden är utvecklad för att kunna appliceras på system som saknar en differentierbar kostnadsfunktion. För att demonstrera metoden presenteras även två giriga optimeringsalgoritmer vars attackeffektivitet utvärderas. Det konstateras att effektiva ohörbara störningar är svåra att skapa, vilket överensstämmer med liknande arbeten. Genom att öka störningarnas inverkan till 18 dB högre än den hörbara gränsen definierad av den psykoakustiska modellen uppnås 64% effektivitet för riktade attacker och 87% effektivitet för oriktade attacker på det undersökta klassificeringssystemet.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Machine learning (ML) algorithms are a crucial tool for problem solving in a variety of domains. Modern technologies such as autonomous vehicles and voice assistants are some applications made possible by the advances within the field. More specifically deep learning has become the go to ML method for achieving state-of-the-art performance in computer vision and speech recognition. With the deployment of deep learning models in real world applications, one would expect these systems to be secure. However, it has been shown that real world systems can be tricked by feeding malicious data to the system [1–4]. Such an input is known as an adversarial example (AE), which in practice is specifically created with the intent of producing a counter intuitive response from the system, while simultaneously appearing benign to a human beholder. The early research done on the topic mainly considered attacks on visual classification systems, in which the attacker has perfect control over the classifier and the input fed to the classifier [5, 6]. For such a scenario one can often generate an adversarial perturbation ρ that when added to the original image signal x is imperceptible to the naked eye. An AE $\tilde{x}$ is thus created as $\tilde{x} = x + \rho$, where ρ is often specifically crafted for a particular x. Today there exist a plethora of algorithms for generating AEs in this particular setting [7–11]. In comparison, attacks applied in real world scenarios require larger and often perceivable perturbations [1–4].

Research on adversarial perturbations on audio is more limited in comparison to its visual counterpart. The most common problem of classification on audio domain data is speech recognition, meaning identifying words/sentences in a speech signal. Such technology is becoming a part of many people's everyday life through the usage of voice assistants in smartphones and speakers; e.g, Google Assistant, Apple Siri, and Amazon Alexa. Speech classification systems similar to those used by these assistants have been shown to be exploitable [4, 12–14]. While these systems are given the authority to perform more and more tasks, the number of possible exploitation of the systems increases.

When generating AEs one often formulate the problem as a minimization problem in which the perceptual distortion between the AE and the original signal is minimized while $\tilde{x}$ is being misclassified. In the image domain, distortions metrics such as ℓ_0,

ℓ_2, and ℓ_∞ have been used successfully [5, 7–11]. On audio data one would instead like to have an optimization loss that captures the perceptual difference between the signal x and the AE $\tilde{x}$ experienced by the human auditory system (HAS). Psychoacoustics is the field of research concerned with modelling the HAS. Some research on AEs has put effort into utilizing psychoacoustics to limit the perceptual distortion of the generated AEs [13, 15]; these methods have used gradient based optimization techniques too generate AEs. In this thesis the intriguing prospect of creating inaudible adversarial examples is explored in depth in a more general setting compared to previous work. This is done by developing black-box optimization strategies which place perturbations in time-frequencies that are perceptually limited according to the psychoacoustic model.

A hazardous quality of AEs is their transferability property, meaning that an AE effective at fooling one model is likely to be effective on another model aimed at solving the same task, but using a different architecture and/or trained on a different dataset [5]. This phenomena has been exploited in a variety of attacks, by so called substitute attacks [2].

The existance of AEs shows that there is some intrinsic flaw in the way our learning methods generalize the statistical properties of the data. The generalization gap between human intelligence and artificial intelligence has been explored in [16]. Research on AEs provides insight of this generalization gap and are thus a useful tool for better understanding the problems of current state-of-the-art ML systems.

Problem Setting. Generally there are two types of speech recognition systems: automatmatic speech recognition (ASR) and keyword spotting (KWS). In an ASR system whole sentences are transcribed at once, while in a KWS system only one word is classified at a time. KWS is similar to image classification in the sense that the dataset consists of all classes that the system should be able to transcribe. On the other hand, in an ASR system the classifier must be able to handle new unseen transcriptions due to the large variety of sentences that can be formed. The developed attacks in this thesis focuses on the problem of KWS, this makes the evaluation of the algorithms less computationally expensive, since KWS systems can be light weight compared to ASR systems. The KWS system is implemented to classify the 10 class speech command dataset [17] using deep architectures inspired by [18]. The developed methodology for inserting inaudible perturbations in signals is nevertheless applicable for other settings too. Further the optimization algorithms could also be adapted to ASR by changing the metric used for optimization.

The amount of information the attacker has regarding the classifier highly effects what optimization strategies can be used for generating AEs. A setting were the attacker has perfect knowledge of the classifier is known as a white-box setting. Such attacks are infeasible in most real world scenarios, but can be used to highlight a worst case scenario. If the known classifiers architecture is differentiable, gradient based optimization can be used, which can produce strong attacks [7, 8]. On the contrary,

in a black-box adversarial attack, the attacker has limited knowledge of the classifier, hence this setting is more applicable to real world scenarios. Both white-box [19, 20] and black-box [14] attacks have been demonstrated on audio. In this work a general setting is used i.e. black-box setting.

Research Question 1: *Is it possible to generate inaudible adversarial perturbations in a Black-box setting for a KWS task?*

By answering this question one gains insight into the feasibility of performing such attacks on real-world systems. This is highly relevant from a security perspective since inaudible perturbations would be impossible for humans to perceive, and thus one could perform attacks without raising any suspicion.

Research Question 2: *Are such adversarial examples transferable between different classifiers or different classifier parameters?*

By evaluating transferability one gains insight of how general the generated adversarial perturbations are.

1.1 Limitations

Since this work concerns black-box attacks, which eliminates the possibility of using gradient based optimization, finding adversarial examples is computationally difficult. Even though a somewhat small architecture (7-layers) by today's standard is explored, it becomes computationally infeasible to perform extensive testing on the effectiveness of the algorithms due to limited computational resources. As a consequence the results presented considers a limited set of signals. Hence, it is unclear to what extend these results generalise to the larger problem of AEs in KWS tasks.

1.2 Ethical Analysis

It can be argued that publicizing new algorithms for generating AEs and exploring transferability increases the chances of AEs being used to exploit real world systems. This might seem like a strong argument for not exploring this topic from an ethical perspective. However, there are multiple reasons to why developing the understanding of AEs is necessary. Algorithms for generating AEs is an effective way of evaluating robustness. In algorithms for generating AEs, the perturbations are being specifically crafted to fool the system while remaining perceptually benign. Although the generated AEs might be unlikely to occur naturally, it still gives insight to worst case scenarios and can thus be used to evaluate how much trust and authority can be given to a particular system. The claim that adversarial examples doesn't occur naturally is somewhat disproved by [21], however, this type of natural AEs are natural images and thus have different statistical properties to many of the algorithmically generated AEs.

Evaluating robustness using AEs isn't only useful for determining the trust to be placed in a system, it is also a valuable tool for improving state-of-the-art ML. By determining a system's weak spot, strategies of improving the system can be proposed. Adversarial training is one such technique, which involves training the classification systems on AEs. Recent findings [22] suggest that this is an effective way of increasing the robustness of image classifiers in a scalable manner. The end goal of research in AEs is to create more robust ML that don't have the generalisation problem associated with modern state-of-the-art ML, and hence be secure against AEs. Thereby reducing the problem of adversarial examples and allowing more trust to be given to ML systems. Today, the existence of adversarial examples in modern ML is the problem, and not if they can be found or not.

Depending on the scenario, attacking a system might actually be considered ethical. For example, in [23] glasses that evade face detection are presented. Hence, such work could make mass surveillance more difficult and help preserve individuals privacy. A similar case can be made for audio, where one might want to find disturbing signals that are perceptually limited to humans but are effective at causing misclassification in speech recognition applications.

As of yet, it seems like no real world exploitation has been done using AEs. This might be considered an insurance that the research in AEs doesn't pose a threat. However, it is known that many state-of-the-art image classification systems acting in the real world are susceptible to AEs. The reason for AEs not being used for exploitation yet might be that there are easier ways to exploit systems through weaknesses in cyber security.

Considering the contributions of this work in particular, it is shown that inaudible AEs are difficult to find, which is similar to what others have found in white-box settings [13, 15]. Hence, the feasibility of performing such an attack as of right now seems very low. The setting explored in this work assumes that the attacker has perfect control over the input fed to the classifier, which would not be the case in reality. When performing real world attacks one needs to consider the acoustic response of the environment, which has been shown to further increase the difficulty of generating effective AEs [13]. In this work, it is also assumed that the attacker has access to the soft classification labels, which in a real world scenario would likely not be available. Further, since the generated AEs are shown to have weak transferability, the possibility of performing substitute black-box attacks is small. The attacks presented in this work are hence far from being applicable to any real world scenario. Attack scenarios focusing on real world application have already been explored in detail in works such as [4], hence an attacker with ill intent has more to learn from such research compared to this work.

Chapter 2
Background

The chapter starts of by briefly introducing the problem of classification in modern machine learning (ML) from a statistical perspective. Since the method of generating adversarial examples proposed is applicable on any classifier architecture, how a classifier is implemented in practice isn't central to the problem, and hence not presented in detail. Then follows an introduction to the field of adversarial examples. Finally, some common methods of generating adversarial examples are presented as well as the current academic understanding of adversarial examples.

2.1 Audio Classification

The problem of classification can be formulated as finding a function $Q : \mathcal{X} \to \mathcal{Y}$ that maps the realisations of the random variable $X \in \mathcal{X} \subseteq \mathbb{R}^N$ to the correct realisation of the statistically dependent random variable $Y \in \mathcal{Y} \subseteq \{l_1, l_2, ..., l_M\}$. Although a wide variety of problems can be formulated in this manner, the classical problem of classification within the audio domain is speech recognition, meaning identifying words/sentences in a speech signal. Whereby x_i is a speech signal produced by a speaker uttering the word/sentence y_i.

In practice, the system can be divided into two parts: (1) Feature extraction $Q_{ext} : \mathcal{X} \to \mathcal{T}$ which extracts relevant features $T \in \mathcal{T} \subseteq \mathbb{R}^A$ from X, and (2) Discrimination $Q_{dis} : \mathcal{T} \to \mathcal{Y}$ which uses the relevant features to predict the most likely class association of X. The whole classification system is then described by: $Q = Q_{dis} \circ Q_{ext}$.

2.1.1 Feature Extraction

The goal of Q_{ext} is to find a representation T of X which contains less statistically irrelevant information of X for predicting Y than X. At the same time, the statistically dependent information between X and Y must be retained in order to not inhibit classification performance. Such a representation is known as a sufficient statistic and can be formulated in terms of the mutual information I as:

Definition 2.1.1. ***Sufficient Statistic***
Let X, Y, T be statistically dependant random variables, following the Markov chain $Y \leftrightarrow X \rightarrow T$. Then, if

$$I(Y;X) = I(Y;T), \tag{2.1}$$

T is a sufficient statistic of X for Y.

By obtaining a sufficient statistic T, no information about the task is lost, hence, by identifying the statistically dependent information between T and Y, a discriminator Q_{dis} would in theory be able to estimate the most likely class y_i for any x_i using $Q_{ext}(x_i)$ as well as would be possible with x_i itself.

If T instead would exclude all statistically irrelevant information of X w.r.t. Y while retaining all statistically relevant information, T would be a minimal sufficient statistic for Y w.r.t. X and for X w.r.t. Y. Defined in terms of mutual information as follows:

Definition 2.1.2. ***Minimal Sufficient Statistic***
Let X, Y, T be statistically dependant random variables, following the Markov chain $Y \leftrightarrow X \rightarrow T$. Then, if

$$I(Y;X) = I(Y;T) = I(X;T), \tag{2.2}$$

T is a minimal sufficient statistic for X w.r.t. Y, and for Y w.r.t. X.

In practice, finding a feature transformation that creates minimal sufficient representations is unfeasible, and hence there will always be a trade-off between retaining high $I(T;Y)$ and reducing $I(T;X)$, making it a fundamental problem in statistics and machine learning, which is commonly known as the Information Bottleneck Problem [24].

In the field of audio classification there are some feature transformations that have become widely accepted as being good representations. Instead of being motivated from an information theoretical perspective, these transformations have rather gained validity by other desirable properties such as being computationally efficient, applicable for a variety of tasks, and/or having an intuitive design. Some common features for speech signals are the spectrogram, the Mel frequency cepstral coefficients (MFCC) [25, 26], the zero-crossing rate and the fundamental frequency [27].

More recently, with the development of deep neural networks, the classifier can learn effective feature transformations on its own. This practice has been common practice for visual data for several years, but has more recently started being utilized on audio [18, 28].

Short-time Fourier Transform

For many audio classification tasks, both temporal and frequency domain information is useful. For example, in speech recognition the phonemes are preferably analysed

in the frequency domain, but in order to construct words from the phonemes the temporal information is required. To accommodate for this, feature extractors commonly divide the signal into analysis windows and then perform some spectral transformation on each window, usually the Discrete Time Fourier Transform (DTFT). This transformation is known as the Short-time Fourier Transform (STFT). Following is the definition of the STFT used throughout this report.

Definition 2.1.3. ***Short-Time Fourier Transform***
Let $x_{i,win}[m,n]$ be an audio sample in window m at sample index n $(0 \leq n < N)$ of the audio signal x_i.

Then using the N-sized Hanning window,

$$w[n] = 0.5 - 0.5\cos\left(\frac{2\pi n}{N-1}\right), \tag{2.3}$$

the DTFT *is applied on the piece-wise multiplied $x_{i,win}[m] \odot w$ as:*

$$\text{STFT}^N[m,k] = \frac{1}{N}\sum_{n=0}^{N-1} x_{i,win}[m,n]w[n]e^{j2\pi k/N}, \quad 0 \leq k \leq N/2 \tag{2.4}$$

resulting in what is known as the short-time Fourier transform. The $\text{STFT}^N[m,k]$ *defines the real and imaginary component of the spectral component k at frame index m.*

Note that in this work the STFT is only applied on real valued data, hence the limited range of k. If not explicitly stated otherwise, the frames used in STFT^N have an overlap of 50%, meaning that $x_{win}[m]$ and $x_{win}[m+1]$ share 50% of their samples. For reasons that will become dear during the design of the method for generating adversarial examples it is worth noting that the STFT is an invertible transformation if $\sum_m w^2[n-m(N/2)] \neq 0\ \forall n$, which is known as the nonzero overlap add (NOLA) constraint[1]. For the specified window and overlap the constraint is fulfilled.

Spectrogram

A spectrogram is a 3-dimensional representation of a signal that visualizes the power of a spectral component k, at frame index m. Usually in a spectrogram, time is represented along the x-axis, frequency along the y-axis, and the power by the intensity of the pixel at coordinates (m,k). The Spectrogram is formally defined as follows.

Definition 2.1.4. ***Spectrogram***
Let STFT_x^N *be the Short-Time Fourier Transform of x, as defined in Eq. 2.4. Then the Spectrogram of x is:*

$$\text{Spectrogram}_x^N[m,k] = \frac{8}{3}|\text{STFT}_x^N[m,k]|^2. \tag{2.5}$$

[1]This formulation of the NOLA constraint is only applicable for 50% overlapping frames.

The gain constant $\frac{8}{3}$ is included to compensate for the average reduction in power from the multiplication with the Hanning window during the evaluation of the STFT.

Based on Eq. 2.5 one can note that the only information that is discarded in the spectrogram, compared to the original signal, is the phase. Taking this into account the spectrogram is more accurately described as an alternative visual representation of the input rather than a feature representation. In fact, feature extraction methods developed with the intention of being applied on images have been used on Spectrograms. Music identification robust to pitch shifting was achieved in [29] by applying the Scale-Invariant Feature Transform (SIFT) on spectrograms. Spectrograms can also be used as input to convolutional neural network classifiers to let the network learn appropriate feature transformations in conjunction with the discriminator [30]. State-of-the-art performance was achieved on automatic chord recognition using this approach in [31]. As noted in [32], there are differences between spectrograms and normal images that make 2d-convolutional feature extractors hard to motivate; given a pixel from an object in an image, generally the neighboring pixels belong to the same object, which is not the case in a spectrogram since overtones belonging to the same sound object often lay far apart. Another dissimilarity is that different sound sources/objects can have overlapping frequency content, and thus one pixel of the spectrogram can belong to multiple sounds.

A common alteration when visualizing the spectrogram for speech signals is to use a logarithmic frequency axis. This is motivated by considering that humans perception of frequencies behave logarithmically. The mel-scale is a logarithmic frequency scale, first introduced in [33], designed to mimic the human perception of distance in frequency. A spectrogram with frequency components placed linearly in mel-scale is called mel-spectrogram. MFCC is another common use case for the mel-scale.

Another alteration of the original definition of the spectrogram is to represent the component strength in dB, which better models signal strength perceived by humans. The dBspectrogram for a signal x_i is defined as:

$$\text{dBspectrogram}^N_{x_i}[m,k] = 10\log_{10}\text{Spectrogram}^N_{x_i}[m,k]. \tag{2.6}$$

2.1.2 Machine Learning Discriminators

Traditional methods for solving the ASR problem rely on the discriminator Q_{dis} finding the statistical dependence between the target Y and the feature T, where T is defined by a hand crafted transformation Q_{ext} of X. For a variety of machine learning methods such an approach is used. Specifically, for ASR systems Hidden Markov Models (HMM) have proven to be highly effective [34]. More recently, such traditional classification methods are gradually being replaced by DNNs, which instead parameterize the whole classification system. The method/architecture used to parameterize the classifier will be denoted as F and the model parameters associated

with the architecture will be denoted θ, creating the density $q_{\theta,F}(\cdot)$. An example of this approach being used with great success is [18], in which the architecture F consists of layered $1d$-convolutions acting on raw audio data.

Multiclass Classification

Following is a brief description of how the general classification problem fits into a statistical framework. The data-label pairs found in the dataset D are realizations of the random variables X and Y and are assumed to be i.i.d. from the joint density $p(X, Y)$. However, it is the conditional density $p(Y|X)$ that is sought to be approximated by $q_{\theta,F}(Y|X)$. Note that $p(Y|X)$ and the "conditional density of $Y|X$" refers to either the Probability Density Function (PDF) if $Y|X$ is continuous or the Probability Mass Function (PMF) if $Y|X$ is discrete. The dataset D can be regarded as a subset of all possible data-label pairs $\mathcal{D}$ associated with the classification problem.

The approximated conditional density $q_{\theta,\mathrm{F}}(Y|X = x_i)$ is known as a soft classification of x_i, i.e. the output is interpreted as the classification confidence for each label in $\mathcal{Y}$. A hard classification of x_i, meaning the class label with highest prediction confidence, will be referred to as:

$$\hat{y}_i = \hat{q}_{\theta,\mathrm{F}}(Y|X = x_i) = \underset{l \in \mathcal{Y}}{\operatorname{argmax}}\, q_{\theta,\mathrm{F}}(Y = l|X = x_i). \tag{2.7}$$

Optimizing θ

In a supervised learning setting, the data used for improving the estimated conditional density $q_{\theta,F}(Y|X)$ of the true density $p(Y|X)$ consists of both the input data x_i and its corresponding ground truth class label y_i. The dataset containing the whole set of known input-output mappings can be expressed as $D = \{(x_i, y_i)\}_{i=1}^{M} \subset \mathcal{D}$. Usually, the dataset is split into a training set D_{train}, a validation set $D_{validation} \backslash (D_{train} \cup D_{eval})$, and an evaluation set $D_{eval} = D \backslash (D_{train} \cup D_{validation})$. The training partition D_{train} is used to improve the decision regions of the classifier by optimizing θ. The classification performance is then gauged using the validation set in order to select the architecture F. By evaluating the accuracy on D_{eval}, one effectively estimate how well the model would perform on unseen data from the same distribution as the evaluation set.

2.2 Introduction to Adversarial Examples (AEs)

Definition 2.2.1. ***Untargeted Adversarial Example*** *If x_i is a realisation of the generative model $X|Y = y_i \in \mathcal{X}$ and $q_{\theta,F}(Y|X)$ is a classifier approximating the density $p(Y|X)$, then if $\tilde{x}_i$ satisfies the inequality:*

$$d(\tilde{x}_i, x_i) < \epsilon \;\; \textit{such that} \;\; \hat{q}_{\theta,F}(Y|X) \neq y_i, \tag{2.8}$$

$\tilde{x}$ is an untargeted adversarial example (AE), where $d(\cdot)$ is some distance measure upper bounded by the constant ϵ. The value of $d(\tilde{x}_i, x_i)$ will in words be called the perceptive distortion of $\tilde{x}$, while the metric $d(\cdot)$ is called perceptual distortion metric. Although, $d(\cdot)$ might in some cases not correspond well to human perception.

In practice, there are no standard definition of perceptual distortion. Its meaning is highly dependant on what one considers is acceptable for a specific setting. Hence, $d(\cdot)$ and ϵ are chosen in conjunction to define the space deemed suitable for a particular setting. Using Definition 2.2.1, any algorithm for generating adversarial examples can be fit into the definition by properly choosing $d(\cdot)$ and ϵ. However in many cases $d(\tilde{x}, x_i)$ isn't actually formalized as a metric, but rather becomes abstracted by the intrinsic properties of the algorithm generating the AE and its settings. As an example, some research focused on finding $\tilde{x}$ that are perceivable but inconspicuous [2, 35]. To fit their AEs into the Definition 2.2.1, one must include the criteria of inconspicuousness into the distortion measure. In practice, their method for generating AEs limits the search space for the perturbation to an area which is chosen to imitate graffiti, hence $d(\cdot)$ doesn't have to be specified as long as graffiti is considered to be inconspicuous. Although note that there are likely other perturbations that would be considered inconspicuous too, therefore only a subspace of inconspicuous AEs can be found using such an approach. In [36], the perceptual distortion metric used is the ℓ_0-distance; in their case the ℓ_0 of the perturbation is set before optimizing and hence it will always result in $d(\cdot) < \epsilon$, although it might not fulfill the misclassification criteria.

Having a differentiable definition of $d(\cdot)$ opens up the possibility of a using gradient based optimization technique to find adversarial examples, as done in [5, 19], both of which use the ℓ_2-distance on image domain data.

2.2.1 AE Settings

Adversarial attacks can be divided into the following two categories.

- White-box attacks: the attacker have access to the some or all of the following information regarding the victim model: architecture, parameters, training method, and/or training data.
- Black-box attacks: the attacker only has access to the model output $q_{\theta,F}(Y|X = x_i)$ (sometimes called semi black-box setting), or in the stricter case only the prediction of the class with highest confidence $\hat{y}_i = \hat{q}_{\theta,F}(Y|X = x_i)$, or denoted as $\tilde{y}_i = \hat{q}_{\theta,F}(Y|X = \tilde{x}_i)$ when predicting an AE.

White-box attacks provide insight of a model's flaws in a worst case scenario, but aren't feasible attacks in model's acting in the real world, since the attacker would only have a limited understanding of the model. Black-box attacks on the other hand, are more applicable to real world scenarios.

Another important aspect of black-box attacks is that for some domains there are limitations in the control one has over the input fed to the classifier, due to physical conditions. For example, when attacking the visual classification system of an autonomous vehicle, as done in [2], one has to consider possible transforms between the perturbation and the classifiers input, such transformations include but are not limited to, distance, rotation, lighting, or weather. Similarly, to create adversarial perturbations on audio that remain effective once played over-the-air, it is beneficial to account for reverberation and frequency response of the environment [13].

Adversarial attacks can further be divided into targeted and untargeted attacks. Eq. 2.8 presented the definition of an untargeted adversarial example. In a targeted attack $\tilde{x}$ should instead be classified as a specific target class $t \in \mathcal{Y}$:

Definition 2.2.2. ***Targeted Adversarial Example*** *If x_i is a realisation of the generative model $X|Y = y_i$, $q_{\theta,F}(Y|X)$ is a classifier approximating the density $p(Y|X)$ and $\tilde{x}_i \in \mathcal{X}$ satisfies the equality:*

$$t = \hat{q}_{\theta,\mathrm{F}}(Y|X = \tilde{x}_i) \text{ such that } \; d(x_i, \tilde{x}_i) < \epsilon, \text{ and } t \in \mathcal{Y}\backslash\{y_i\} \tag{2.9}$$

then, $\tilde{x}$ is a targeted AE for class t, where d is some distance measure upper bounded by the constant ϵ.

Targeted AEs are often harder to generate, meaning requiring higher perceptual distortion. This can be understood by considering that the nearest decision boundary with regard to perceptual distortion is:

$$\underset{\tilde{x}_i}{\operatorname{argmin}}\, d(x_i, \tilde{x}_i) \text{ such that } \; \tilde{y}_i \neq y_i, \tag{2.10}$$

and hence is a lower bound on the perceptual distortion for an untargeted adversarial example $\tilde{x}_i$. A targeted attack will only have the same lower bound if the target class equals the solution class defined by $\tilde{y}_i$ in Eq. 2.10 or if multiple classes (including the perceptually nearest class and t) have equal perceptual distance to x_i.

2.3 Adversarial Attacks

There is a plethora of algorithms for generating adversarial examples, of which some of the most influential are presented here with the intention of giving the reader an overview of the field of adversarial examples. Note that in some of the following optimization strategies the "AE" isn't formally casted to its original space $\mathcal{X}$. That is because the original formulation of those attacks doesn't include the operation during the optimization procedure, but instead it is applied on the final optimized "AE".

Box Constrained L-BFGS AEs

Box constrained L-BFGS, introduced by Szegedy *et al*.[5], was one of the early method for generating adversarial examples. L-BFGS stands for Limited Memory Broyden-Fletcher-Goldfarb-Shanno, which was the optimization strategy used to find an approximate solution to the hard problem of minimizing $d(\tilde{x}, x) = \|\rho\|_2$ such that Eq. 2.9 is satisfied. This method is defined as

$$\underset{\tilde{x}}{\text{argmin}}\, c\,\|\rho\|_2 + \mathcal{L}(\tilde{x}, t), \tag{2.11}$$

where $\mathcal{L}(\tilde{x}, t)$ is the loss function of the model being attacked and c specifies the impact of the ℓ_2-regularization. In the case of a convex loss this optimization would yield the optimal solution in terms of ℓ_2-distance [5]. However, solving Eq. 2.11 for the non-convex case doesn't guarantee that an AE is found for any particular value of $c > 0$. Hence, to find an AE with the least perceptual distortion, c has to be optimized. In practice this involves solving the optimization in Eq. 2.11 multiple times while performing some one-dimensional optimization on c.

A key advantage to formulating the problem as a general optimization problem is that more criteria of the adversary can easily be included. For example changing the ℓ_2 distance metric to some other function which better models perception depending on the application domain [37].

Fast Gradient Sign Method (FGSM)

The fast gradient sign method introduced by Goodfellow *et al*.[10] disproved the general belief that the existence of adversarial examples stemmed from the non-linear nature of neural networks [5]. This was done by linearizing the cost function at θ. The linearly optimal attempt for generating an AE constrained by the max-norm of ρ can then be calculated as

$$\tilde{x} = x + \epsilon\ \text{sign}(\nabla_x \mathcal{L}(\theta, x, y)), \tag{2.12}$$

where $\nabla_x \mathcal{L}(\theta, x, y)$ is the gradient of the loss $\mathcal{L}$ with respect to x and ϵ determines the max norm of the perturbation. While it isn't guaranteed that increasing the model's loss with any specific amount results in misclassification, the loss value for a misclassified instance is by definition larger than otherwise, hence increasing the loss is a sensible direction for the optimization [37].

It is not guaranteed that FGSM produces an actual adversarial example as defined in Eq. 2.8 regardless of the ϵ value specified. One easy way of seeing this is to consider the methods inherent assumption regarding additive contributions across dimensions. While one data point in x can enhance adversarial properties by a small step in the direction of the gradient, there is no guarantee that this holds true when changing all data points in x simultaneously in a non-linear system. Although, it should be stated

that FGSM is likely the most used and well known method for generating AEs due to reasons such as: generally producing strong attacks that are transferable (if no proper countermeasures are taken) and being computationally efficient, making adversarial training feasible [22].

Many works have improved upon the original FGSM formulation, e.g. iterating the algorithm with a small step size [38] or applying a random perturbation on x before performing the FGSM. The later attack is known as R+FGSM [11].

Jacobian-based Saliency Map Attack (JSMA)

The Jacobian-based Saliency Map Attack introduced in [9] restricts the number of dimensions perturbed ($\|\rho\|_0$) and hence can be used to create sparse perturbations. Each input dimensions impact on the classification is gauged by evaluating $\nabla_x(q_{\theta,F}(Y = t|X = x_i))$. Assuming linear properties in $q_{\theta,F}(Y|X = x_i)$ around x_i, a higher gradient for a particular input dimension indicates that this dimension is beneficial to perturb. In JSMA the strength of the gradient determines the order in which to perturb dimensions. The dimensions with the estimated highest influence are then greedily modified in pairs until either $\tilde{x}_i$ is classified as t or the maximum number of allowed input dimensions has been changed. In the later case, the attack has failed to find a AE with less perceptual distortion than what is defined by the maximum number of perturbations.

Carlini & Wagner (C&W)

Carlini and Wagner [7] introduced a family of attacks where the optimization strategy is specifically designed for the distortion metric used (they presented algorithms for ℓ_2, ℓ_0, and ℓ_∞). The attacks use backpropagation to perform iterative optimizations on the input data using the Adam optimizer [39].
The ℓ_2-attack is formulated as[2]:

$$\underset{\rho}{\text{minimize}}\ c\,\|\rho\|_2^2 + \text{obj}(\tilde{x}), \tag{2.13}$$

where the objective function (obj) is defined as:

$$\text{obj}(\tilde{x}_i) = \max\left(\max_{j\in\mathcal{Y}\setminus\{t\}}(q_{\theta,F}(Y = j|X = \tilde{x}_i)) - q_{\theta,F}(Y = t|X = \{\tilde{x}\}), -\kappa\right), \tag{2.14}$$

where κ determines the confidence with which the misclassification occurs (usually $\kappa = 0$ is used). In practice, this objective results in optimizing to increase the prediction confidence of t while reducing the prediction confidence of the class with highest confidence apart from t.

[2] In [7] the impact of c on minimization is the inverse to the formulation used in this report.

The ℓ_0-attack is more complicated since the ℓ_0 distance metric is non-differentiable and hence ill-suited for the optimization problem in Eq. 2.13. Instead, the ℓ_0 is minimized through an iterative process in which the least classification significant dimensions of the input data are eliminated. The process of eliminating dimensions works by, on each iteration evaluating the ℓ_2-attack and removing the dimensions with the lowest gradient $\nabla_x(q_{\theta,F}(Y = t|X = x_i))$. The iterations are performed until the set of perturbable dimensions has been reduced so that the ℓ_2-attack no longer finds an AE, in which case the last successful ℓ_2-attack is used. Since the attack on each iteration chooses dimensions greedily, the found solution is not guaranteed to be minimal.

The authors of [7] note that the ℓ_0-attack works by reducing the whole set of input dimensions, in contrast to JSMA, which works by growing a set of perturbations. Comparing these two specific methods, the principle difference of reducing instead of growing the set is significantly more computationally efficient in this particular case.

One-Pixel Attack

As the name suggest, the one-pixel attack [36] can be used to find $\|\rho\|_0 = 1$ adversarial perturbations on images. The algorithm works in a black-box setting by utilizing the meta-heuristic optimization strategy: differential evolution (DE). The implementation of differential evolution used in [36] works by randomly initializing a set of candidates $S_{\text{candidates}} = r_{1,1}, ..., r_{1,A}$, in which every individual has the following attributes:

- Spatial indexes, defining the dimension of the input to perturb (x-y coordinates)
- Perturbation values (RGB), defining the values of the perturbation at the individuals spatial indexes

The following process is then iterated: A set of children $S_{\text{children}} = r_{2,1}, ..., r_{2,A}$ are produced according to the DE formula:

$$r_{2,j_1} = r_{1,j_2} + \frac{1}{2}(r_{1,j_3} - r_{1,j_4}), \quad j_2 \neq j_3 \neq j_4, \tag{2.15}$$

which is evaluated per child index (j_1) and j_2, j_3, and j_4 are randomly drawn. If a child (r_{2,j_1}) has a higher fitness score than its parent r_{1,j_1}, the child replaces the parent ($r_{1,j_1} = r_{2,j_1}$). The fitness score for an individual r, in a targeted attack is given by $q_{\theta,F}(Y = t|X = x'_{i,r})$, in which $x'_{i,r}$ represents the perturbed x_i caused by the individual r.

2.3.1 Black-Box Attacks in Practice

Given an adversarial example $\tilde{x}_1$ created for a classifier q_{θ_1,F_1}, it is likely that $\tilde{x}_1$ has adversarial properties on another classifier q_{θ_2,F_2} solving the same task as q_{θ_1,F_1}, even

if the model architecture differs ($F_1 \neq F_2$) and the model parameters (θ_1 and θ_2) have been optimized on a different dataset [5]. Note that this isn't only applicable to DNN architectures, it holds true for a variety of machine learning methods [40]. What enables transferability and how to best enhance robustness towards them is an open research questions and the current academic understanding is mostly based on experimental results. For example, a discovery by Kuratin *et al*.[38] suggests that stronger attacks exploiting nonlinear behaviors in $q_{\theta,F}$ such as iterative methods have less transferability than simpler methods exploiting linearity, such as the FGSM. A possible explanation for this is that iterative methods "overfit" to a particular θ and F [41]. It was also shown that the ϵ was positively correlated to the transferability. It has been empirically shown that untargeted attacks are more likely to be transferable than targeted ones [42].

The main concern of transferability is the possibility to perform substitute black-box attacks, in which the attacker trains a substitute model for generating AEs in a white-box setting and then uses them on the actual black-box target classifier. Performing such an attack has been proven effective for a variety of scenarios [1–3, 41, 42].

2.3.2 Adversarial Perturbations on Audio

Experimental results originally indicated that adversarial attacks on audio are more challenging than its visual counterpart [20], meaning that effective perturbations with negligible perceptual distortion are easier to find for visual classification systems than for audio classification systems. Since then a lot of effort has been put into generating adversarial examples on audio for a variety of settings, and there now exists a number of methods for generating effective targeted attacks [13, 15, 19, 43]. It does however still seem difficult to achieve targeted over-the-air attacks with low perceptual distortion.

Following is a brief presentation of techniques tested for leveraging perceptual distortion, computational cost, and attack strength specifically in an audio setting.

White-box Attacks

Many effective algorithms for generating targeted AEs on audio use gradient optimization techniques. It is however common for ASR systems to not be fully differentiable, as in the commonly used Kaldi system [34]. In [13] the gradient in conjunction with a psychoacoustic distortion metric is optimized consecutively to limit the perceptual distortion of the AE. A slightly different approach utilizing psychoacoustic modelling was used in [15], where the perceptual distortion is used to scale the gradient during each iteration of the optimization.

Black-box Attacks

Genetic algorithms have been used to perturb MFCC representations of audio in black-box settings [43, 44]. In [44] the perturbations were optimized using the distortion metric ℓ_2 in the MFCC domain. Optimizing the perturbations in the MFCC domain is especially useful if one knows that the classifier internally calculates the MFCC, since the search space contains all AEs while still being reduced, which is especially useful for non-gradient based optimization strategies. This is something discussed in detail during the motivation of the algorithm proposed in this thesis in Section 4.1.

Inconspicuous Perturbations

Definition 2.3.1. *For two statistically dependent random variables $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ with conditional density $p(Y|X)$, then if $q_{\text{human}}(Y|X)$ is the average human estimation of $p(Y|X)$ and $\tilde{x}_i$ is a realisation of $X|Y = y_i$, δ_i is an inconspicuous perturbation on x_i if:*

$$q_{\text{human}}(Y = y_i|X = x_i) = q_{\text{human}}(Y = y_i|X = x_i + \delta_i).$$

Meaning that the average human wouldn't experience any difference in confidence of the generative class y_i. In other words, the perturbation δ_i is considered by q_{human} to be statistically irrelevant for determining the true class.

To find if an AE is actually inconspicuous according to this definition would be difficult. However, as found by Vadillo and Santana [45], attacks that attempt to be inconspicuous on audio are often perceived as sounding artificial to the listener. One would arguably assume that the prediction confidence of class y_i for the artificial sounding audio signal $x_i + \rho$ would be less than for the non-artificial sounding signal x_i. Hence, the findings in [45] should be applicable to Definition 2.3.1, meaning that most methods for generating "inconspicuous" adversarial perturbations on audio may not comply with the definition used here. Although, in [15] a listening test was performed which resulted in the finding that small perceivable perturbations do not necessarily inhibit the human classification performance.

"Inaudible" Adversarial Perturbations

Definition 2.3.2. *Consider an audio signal x_i and a distortion metric $d(\cdot)$ that models human perception of audio, where $d(\cdot) = 1$ specifies the threshold at which humans starts being able to perceive a difference between two signals. An inaudible perturbation δ_i is then defined as:*

$$\delta_i \text{ such that } d(x_i + \delta_i, x_i) \leq 1. \tag{2.16}$$

One intriguing aspect of designing adversarial attacks for the audio domain is the possibility of "hiding" perturbations in masked frequencies, and by doing so making the adversarial perturbations inaudible to humans. To do so one needs to acquire an accurate model of human perception of audio. The field of research concerned with doing so is called psychoacoustics, and is explained in further detail in Section 3.1. The prime example of utilizing psychoacoustic modelling to limit the perceptual impact of the adversarial perturbation is found in [13], where the audio adapted C&W-attack [19] and a psychoacoustic constraint is optimized in turns, producing 100% accuracy of attacks on targeted full length sentences using a state-of-the-art automatic speech recognition system.

2.4 Evaluating Properties

2.4.1 Success Rate

When evaluating the effectiveness of a method for generating AEs the percentage of successful misclassifications is a natural metric to use, it is typically called success rate. For the untargeted case Eq. 2.8 defines the misclassification and for the targeted attack Eq. 2.9 is used.

2.4.2 Transfer Rate

Transferability can be measured in both the untargeted and targeted case. It is measured by the success rate of the attack using adversarial examples generated for the model q_{θ_1,F_1} applied on the target q_{θ_2,F_2}, where $\theta_1 \neq \theta_2$ or $F_1 \neq F_2$ and $\theta_1 \neq \theta_2$.

2.4.3 Distortion Metric

By approximating the human perception of audio using a psychoacoustic model, one can use the psychoacoustic model as a replacement for the human perception distortion metric in Definition 2.3.2. Then, if an inserted perturbation δ_i on an audio signal x fulfills $d(x_i, x_i + \delta_i) \leq 1$ one can guarantee that δ_i is inaudible, provided that the psychoacoustic model is accurate. Note that due to the non-linear behavior of human audio perception, a change between $d(\cdot) = 2$ and $d(\cdot) = 4$ doesn't necessarily mean that the perturbation is perceived as exactly twice as loud, however, this non perfect measure will be used later as an indication of the perceptual distortion of the generated adversarial examples.

Chapter 3
Psyhoacoustics

3.1 Psychoacoustic Modelling

Bosi Marina [46] defines psychoacoustics as the science studying the statistical relationship between acoustic stimuli and human hearing sensation. Researching psychoacoustics has largely been motivated by the development of audio codecs. Accurately modelling psychoacoustics allows quantization noise to be placed in areas where human perception is limited. Such audio encoders are called perceptual audio encoders, where MPEG-1 audio Layer-3 (MP3) is a prime example of one such standard [47]. This section begins by presenting the principles of psychoacoustics, and ends by presenting each implementation step for the pschoacoustic model-1, developed for the MPEG-1 codec.

3.1.1 Principles of Psychoacoustics

Acoustic stimula reaches the human Audio Peripheral Systems (APS), or ears, in the form of pressure waves. Our APS can sense pressure differences u in the range $[10^{-5}, 10^{2}]$ Pascal (Pa). Commonly, this range is described in decibels using the metric sound pressure level (SPL), according to

$$SPL = 10 \log_{10} \left(\frac{u}{u_h} \right)^2 , \tag{3.1}$$

in which u_h Pa corresponds to the weakest human perceivable pressure variation of a 1 kHz tone [46]. Although u_h is defined with regards to human hearing, SPL doesn't correspond well to the human perception of audio strength, instead loudness is the preferred measure. The concept of loudness was first formalized in the 1920s with the purpose of describing perceived sound intensity, thus loudness isn't a physical attribute, but rather a psychological one. From this point on, the whole hearing system will be referred to as the human auditory system (HAS), meaning the APS together with any psychological phenomena involved in hearing. The unit of loudness is called

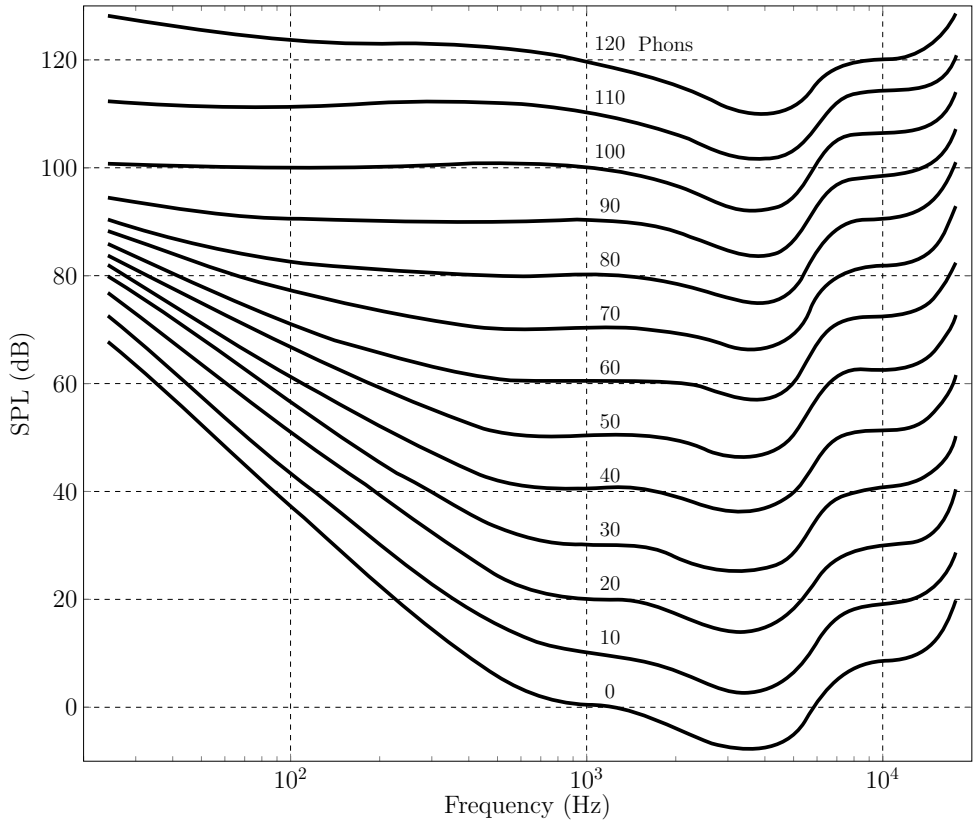

Figure 3.1: Equal loudness curve for different phons, by definition the loudness (phon) at 1 kHz equals its SPL. This plot is based on the equal loudness plot in [48].

Phon, and is an empirically determined loudness measures [48]. Phon is frequency independent, i.e. the perceived tonal strength for a particular phon is the same for all frequencies. By plotting the SPL over frequencies for a particular phon, an equal-loudness contour is created. In Figure 3.1 some equal-loudness contours are shown. Since the shape of the equal-loudness curve changes over frequencies depending on the Phon, the playback volume of a sound effects the perceived tonal balance of the sound, given that the sound covers a wide variety of frequencies [49]. By definition a phon describes the perceived strength of a single sine wave at different frequencies. As a consequence, it doesn't account for how more complex sounds are perceived, for example multiple sine waves with similar frequencies. In such sounds not all of the spectral components are necessarily perceived by the HAS, this phenomena is known as frequency masking.

Frequency Masking

In Figure 3.2, simultaneous auditory masking (also known as frequency masking) is visualised. Any signal below the masking threshold cannot be perceived by the HAS, thus allowing certain modifications of the frequency components in the audio signals to be made without effecting the perceived sound [49]. The shape of the masking threshold around a masker is in practice approximated by a spreading function.

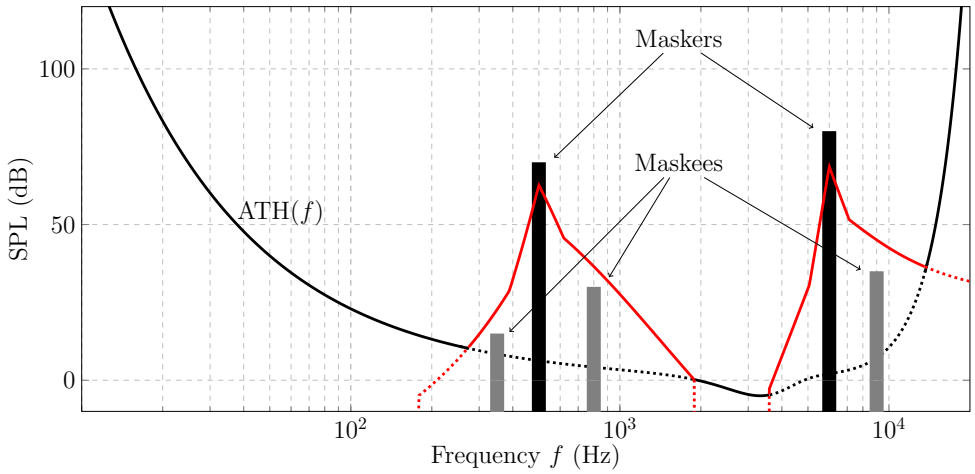

Figure 3.2: Example of maskers and maskees. The curve which defines the masking threshold around a masker is called the spreading function(shown in red), and is an approximation of the perceptual spread of a tone perceived by humans. The solid line is approximates the masking threshold.

Nonsimultaneus Auditory Masking

Masking is not solely a phenomena effecting nearby frequencies of the masker, it also effects perception of nearby sounds in the temporal domain. Interestingly, it can mask sounds occurring up to 20 milliseconds prior to the masker, this is known as pre-masking. Post-masking on the other hand is the perceptual decay of the masker. This phenomena is present up to 150 milliseconds after the masker ends. A visualization of temporal masking is shown in Figure 3.3.

Transformations by the APS

In machine learning terms, the APS is a basic feature extractor, it translates air pressure variations into activations in specific nerves depending on the frequency of the variations, which then are used by the brain to analyse the audio. The APS can be

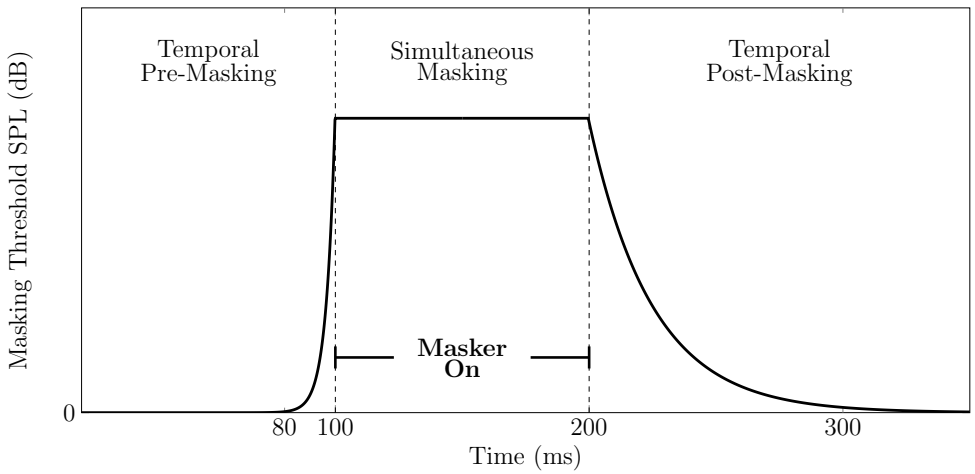

Figure 3.3: Visualization of maskers effect on perception in time

thought of as three distinct parts, referred to as the inner, middle, and outer ear. The outer ear seems to mostly be useful for sound source localisation, which is done by changing the tonal balance of a sound depending on its azimuthal angle of arrival. The middle ear has two functions, the first being to favor transmission of frequency components between 500 to 4000 Hz to the inner ear. Human speech has a roughly similar frequency range and hence the APS enhances perception of human voices. A second important function of the middle ear is impedance matching. This is necessary since the inner ear has an acoustic impedance higher than air and without impedance matching 99.9 % of the sound energy would be reflected without reaching the inner ear. Finally the inner ear performs frequency-to-place conversion by making different frequencies activate different regions of nerves [46].

A noteworthy property of the inner ear is that the frequency resolution is not uniform over the range of human hearing. Through measuring peoples perceived distance between sinusoids researchers have been able to create normalized frequency scales. One example of such a scale is the bark scale:

$$\text{bark}(f) = 13 \cdot \arctan\left(0.00076 f\right) + 3.5 \cdot \arctan\left(\frac{f^2}{7500^2}\right), \tag{3.2}$$

which is used in the MPEG-1 standard. The mapping between bark and Hz is shown in Figure 3.4. This non-uniformity gives the HAS a finer frequency resolution for low frequencies.

In speech processing the frequency-to-place conversion is often imitated as a filterbank. To capture the HAS frequency resolution the filters are usually designed to be overlapping triangular filters with a constant distance in bark scale [50]. In the

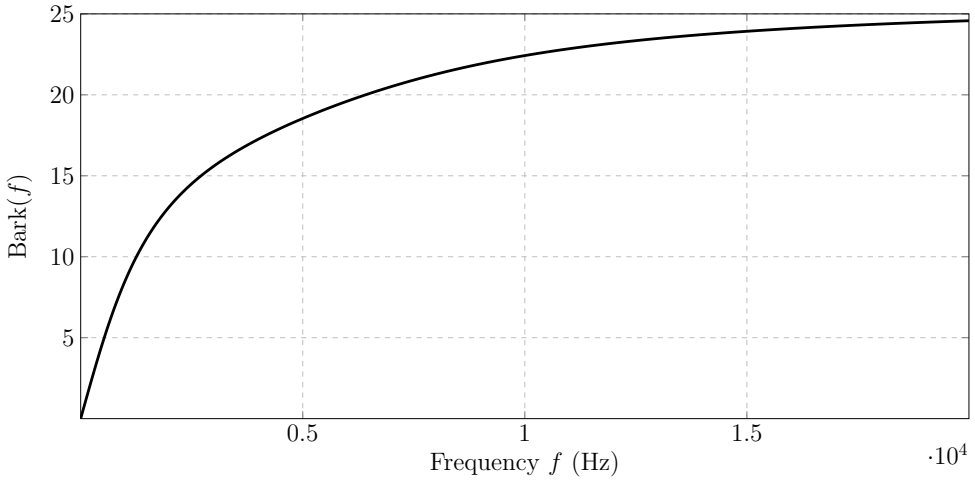

Figure 3.4: Bark as a function of Hz. A frequency change of plus one bark is perceived as the same relative change independent of the starting bark

psychoacoustic model-1 the human frequency-to-place conversion is imitated by bins with constant spacing in bark-scale. To create an accurate model for audio perception the psychological phenomenas must also be accounted for. These are hard to motivate from a biological perspective, and have instead been determined through extensive empirical testing [49].

An example of this is the absolute threshold of hearing, which is the minimum SPL which can be perceived for different frequencies f. The threshold is shown in Figure 3.2, and the function for calculating the threshold is given by:

$$\text{ATH}(f) = 3.64\left(\frac{f}{1000}\right)^{-0.8} - 6.5e^{-0.6\left(\frac{f}{1000}-3.3\right)^2} + \frac{1}{1000}\left(\frac{f}{1000}\right)^4. \qquad (3.3)$$

3.1.2 Psychoacoustic Model-1 Implementation

Given an audio signal x_i transformed into a time-frequency representation z_i, psychoacoustic modelling can be used to estimate a surface in the space of z, which defines the boundary between audible and inaudible noise, limited in time and frequency. This surface is called the minimum masking threshold (MMT_i).

As can be inferred by the foregoing section about the HAS, there are many factors that needs to be taken into account when designing an accurate psychoacoustic model. All of which have to be put in perspective to their computational cost, and the accuracy required by the application [49]. The psychoacoustic Model-1 presented in the MPEG-1 standard is generally regarded as a computationally efficient but somewhat

crude model. One limitation of the model is that it doesn't consider temporal masking. For many applications estimating the effects of temporal masking isn't crucial since the effects of frequency masking often is much greater [46]. Another simplification is the spreading function used by the model. In this section the steps for calculating the MMT_i as defined in the psychoacoustic model-1 will be explained. The implementation presented closely follows the implementation described in [49].

The psychoacoustic Model-1 is defined for a variety of sampling frequencies, the following is a description for the one using a sampling frequency $\text{Fs}_z = 44.1$ kHz. Before beginning the calculation of the MMT_i, the input signal needs to be resampled to the specified sampling rate. If the input signal x has a sampling rate of Fs_x, then the resampling can be expressed as $x \uparrow \text{Fs}_z \downarrow \text{Fs}_x$. In practice the resampling is done by windowed sinc interpolation. Note that the resampling step is not included in the definition of psychoacoustic model-1, but becomes a necessity in this work since the application developed should be able to handle any sampling frequency Fs_x.

Step 1: STFT and SPL normalization

The first step is to acquire an accurate spectral estimate for the audio, this is done by calculating the $\text{dBspectrogram}_x^{512}$ (2.6). Note that in the original MPEG-1 standard the frames are non overlapping, it has however been found that overlapping frames can have beneficial effects for some applications, for example the audio watermarking method developed in [49]. In this work, overlapping frames will be used, and the specific amount of overlap is described and motivated in Section 4.1.1.

Since the perceived tonal balance from Figure 3.1 is affected by the playback level of audio (which is unknown), a playback level is assumed within the model. The maximal SPL in dB is chosen to 96 dB and the frame is normalized accordingly:

$$v[m,k] = 96 - \max_l(\text{dBspectrogram}_x^{512}[m,l]) + \text{dBspectrogram}_x^{512}[m,k]. \quad (3.4)$$

From this point on the calculations of the MMT_i will be presented for one frame index m, thereby simplifying the equations ($v[k]$ is used instead of $v[m,k]$). This simplification is justified by considering that every frame is treated equivalently, and there are no inter-frame dependencies.

step 2: Identify Tonal & Non-Tonal Maskers

The next step is to create sets of tonal and non-tonal maskers. The tonal set S_{TM} consists of all indices of $v[k]$ that are local maximas and fulfill the constraint:

$$|v[k] - v[k+i]| \leq 7, \quad \forall\, i \in \begin{cases} \{\pm 2\}, & 2 < k < 63 \\ \{\pm 2, \pm 3\}, & 63 < k < 127 \\ \{\pm 2, \pm 3, \ldots \pm 6\}, & 127 < k < 251 \end{cases} \quad (3.5)$$

For all $v[k]$ in S_{TM} the strength of the tonal maskers $v_{TM}[k]$ is determined as:

$$v_{TM}[k] = 10\log_{10}(10^{\frac{v[k-1]}{10}} + 10^{\frac{v[k]}{10}} + 10^{\frac{v[k+1]}{10}}), \quad \forall\, k \in S_{TM}. \tag{3.6}$$

The nontonal maskers (v_{NM}) are calculated over frequency bands. There are a total of 25 frequency bands which are roughly equally spaced in bark scale with distance 1. The non tonal maskers are considered to belong to index $\bar{k}$, which is the spectral index closest to the geometric mean of the lowest frequency $f_l[\bar{k}]$ and highest frequency $f_h[\bar{k}]$ of each frequency band. The frequencies of each band are defined in Appendix A.1b. The power of the non-tonal maskers is then calculated by summation over all remaining components in each critical-band that are not in the S_{TM} as:

$$v_{NM}[\bar{k}] = 10\log_{10}\sum_{k\in\Omega} 10^{\frac{P(k)}{10}} \qquad \Omega = \{k : k \notin S_{TM}, f_l[\bar{k}] < f[k] < f_h[\bar{k}]\}, \tag{3.7}$$

where $f[k]$ is the frequency in Hz for the frequency index k.

Step 3: Decimation

The sets S_{TM} and S_{NM} are decimated by evaluating two criteria that every masker should fulfill. Firstly, every masker should be higher than the hearing threshold in quiet (3.3), ATH($f[k]$) for tonal maskers and ATH($f[\bar{k}]$) for the nontonal maskers. Secondly, the frequency distance between maskers must be greater than 0.5 bark. If two or more maskers exist within the same bark range of 0.5, only the strongest one is preserved in its corresponding set, i.e.

$$\begin{aligned} S_{TM,NM} = \{k : v_{TM,NM}[k] \geq \text{ATH}(f[k])\} \\ \cap\ \{k : v_{TM,NM}[k] = \underset{k_0\in-0.5,0.5}{\text{argmax}}\ v_{TM,NM}[k+k_0]\}, \end{aligned} \tag{3.8}$$

where k_0 is all the spectral indices within the bark($f[k]$) $\pm\, 0.5$ bark range.

Step 4: Calculating Masking Threshold

S_{TM} and S_{NM} can now be used to create a masking threshold for the corresponding set, lets call these L_{TM} and L_{NM}. The masking threshold is created by considering how every masker affects every maskee. The number of maskees used in the calculation consists of a subset of all possible spectral components. For a sampling frequency of 44.1 kHz, 106 subsamples are used. How the 106 subsamples relate to the actual frequency components defined by k is listed in Appendix A.1a. Let the maskee subsamples be indexed using j which has a corresponding frequency $f[j]$, then L_{TM} is calculated by:

$$L_{TM}[k,j] = v_{TM}[k] - 6.025 - 0.275\text{bark}(f[k]) + \text{SF}(v_{TM}[k],k,j), \tag{3.9}$$

and L_{NM} is calculated by:

$$L_{NM}[k,j] = v_{NM}[k] - 2.025 - 0.165\text{bark}(f[k]) + \text{SF}(v_{NM}[k], k, j), \tag{3.10}$$

in which the spreading function SF used is defined as follows:

$$10\log_{10}\text{SF}(v,k,i) = \begin{cases} 17(\text{bark}(f[k])-\text{bark}(f[i]))-0.4v[k]+11 & -3 \le \text{bark}(f[k])-\text{bark}(f[i]) < -1, \\ (0.4v[k]+6)(\text{bark}(f[k])-\text{bark}(f[i])) & -1 \le \text{bark}(f[k])-\text{bark}(f[i]) < 0, \\ -17(\text{bark}(f[k])-\text{bark}(f[i])) & 0 \le \text{bark}(f[k])-\text{bark}(f[i]) < 1, \\ -17(\text{bark}(f[k])-\text{bark}(f[i])) & 1 \le \text{bark}(f[k])-\text{bark}(f[i]) < 8. \\ \quad +0.15v[k](\text{bark}(f[k])-\text{bark}(f[i])-1) & \end{cases} \tag{3.11}$$

The tonal masking threshold L_{TM} for 2 maskers is shown in Figure 3.2.

Step 5: Global Masking Threshold

The global masking threshold is a sum over the masking contribution of the tonal and non-tonal masking thresholds, and the hearing threshold in quiet:

$$L_G[k] = 10\log_{10}\left[10^{\frac{\text{ATH}(k)}{10}} + \sum_{j=1}^{N_{TM}} 10^{\frac{L_{TM}[k,j]}{10}} + \sum_{j=1}^{N_{NM}} 10^{\frac{L_{NM}[k,j]}{10}}\right]. \tag{3.12}$$

Step 6: Determining the MMT

The MMT has a spectral resolution of 32, these subbands are uniformly spaced between 0 Hz and the Nyquist frequency. The mapping between maskees to subbands can be found in Appendix A.1a. The MMT is then calculated as:

$$\text{MMT}[k] = \min_{j \in \text{subband}(k)} L_G[j]. \tag{3.13}$$

Chapter 4
Algorithm for Generating Inaudible Adversarial Perturbations

In this chapter a new method aimed at generating inaudible adversarial examples is proposed. This is achieved by placing perturbations in accordance with the MMT defined by the psychoacoustic model from Section 3.1. First the domain in which perturbations are optimized is presented, then follows an explanation of two optimization strategies explored.

To avoid confusion throughout this chapter, it should be clarified that a *perturbation* applied on the perturbation domain representation z_i refers to the act of changing a single data point in z_i. An *adversarial perturbation* instead refers to ρ_i, which is added to the time-domain signal x_i to create an AE, $\tilde{x}_i = x_i + \rho_i$.

4.1 Perturbation Domain

The perturbation domain representation $z_i \in \mathcal{Z} \subseteq \mathbb{R}^{M \times \left(\frac{N}{2}+1\right)}$ of signal x_i, is calculated through the transform $g(\cdot)$ as:

$$(z_i, c_i) = g(x_i), \tag{4.1}$$

in which z_i is a lossy representation of x_i, and c_i is the information of x_i discarded in z_i; allowing x_i to be reproduced by the inverse transformation $g^{-1}(z_i, c_i)$.

To facilitate the goal of creating inaudible perturbations there are two criteria imposed on the choice of perturbation domain:

1. $g(\cdot)$ and $g^{-1}(\cdot)$ must fulfill $x_i = g^{-1}(g(x_i)) \; \forall \; x_i \in \mathcal{X}$.
2. The MMT must be expressible in $\mathcal{Z}$, by transformation $h(\cdot)$, i.e. $h(\text{MMT}_i) \in \mathcal{Z}$, where $h(MMT_i)$ describe inaudible perturbations.

If these criteria are fulfilled one can create an audio signal containing inaudible perturbations by calculating the inverse transform of the perturbed z_i,

$$g^{-1}(z_i + h(\text{MMT}_i), c_i). \tag{4.2}$$

4.1.1 Perturbation Domain in Practice

In this work, $\mathcal{Z}$ was chosen to be similar to the domain of the MMT. In practice this results in the initial transformations performed during the calculations of the psychoacoustic model being included in $g(\cdot)$, i.e. resampling x_i to the sampling rate used in the psychoacoustic model ($\text{Fs}_z = 44100$), calculating the dBspectrogram^N (2.6), and normalizing the dBspectrogram (3.4). If the original sampling frequency of x_i (Fs_x) is less than Fs_z, the resampling can be lossless. The last two transformations are in either case lossy, hence the information discarded is saved in c_i to enable invertability. More specifically, when calculating the dBspectrogram (2.6) the phase information of the original signal is saved, and during the normalization (3.4) the initial maximum dB of each frame is saved.

Definition 4.1.1 ($\mathcal{Z}$-domain)**.** *The transform to $\mathcal{Z}$ can mathematically be expressed as*

$$10^{\frac{z_i[m,k]}{10}} = 96 - \max_j(\textit{dBspectrogram}^N_{x_i\uparrow \text{Fs}_z \downarrow \text{Fs}_x}[m,j]) + \textit{dBspectrogram}^N_{x_i\uparrow \text{Fs}_z \downarrow \text{Fs}_x}[m,k], \quad (4.3)$$

where $\max(\cdot)$ *is calculated along the spectral components for each frame* m*.*

Note that this formulation defines z_i in magnitude instead of dB. This eases the notation since the perturbations are added in magnitude and not dB.

In practice, the calculations in $g(\cdot)$ are done with 32-bit floating point precision, hence during the calculation of the inverse transform the output needs to be cast to its original type, which for wav-files commonly is 16 or 24-bit integers.

The transform $h(\cdot)$ also includes the same dB to magnitude conversion as done in $g(\cdot)$. By then defining $h(\cdot)$ to resample the temporal and spectral components of the MMT_i to the same size as $\mathcal{Z}$, no assumption regarding the spectrogram parameter N has been made. This provides a more general setting compared to previous attempts at creating inaudible perturbations, where N has been fixed. A potential negative consequence of upsampling the MMT in the frequency domain is a reduction in accuracy of the masking threshold, hence the original spectral resolution is a natural starting point for the hyperparameter N. Similarly, upsamling the temporal resolution might cause an inaccurate masking threshold. Increasing the overlap used during the initial STFT evaluation fulfills the same goal as upsamling, but without reducing the model's accuracy of the masking. In this work a time resolution 4 times greater than the original is used in the calculation of the MMT. This results in a masking threshold being calculated at regular intervals of less than 3 ms apart. Due to the effects of temporal masking becoming more evident on short time intervals (see Figure 3.3), one can expect diminishing returns in the accuracy of the masking threshold when increasing the temporal resolution of the MMT. Hence, if an even higher temporal resolution is required for a specific N upsampling is performed.

The dimensionality difference between MMT_i and z_i is ≈ 4 (varying slightly depending on how the length of the signal x_i aligns with the hop-size in the STFT calculation, and the fact that the MMT has slightly different spectral resolution). This can intuitively be understood by considering that the specified psychoacoustic model calculates a dBspectrogram with twice the temporal resolution as the dBspectrogram used during the transform to $\mathcal{Z}$, but then decimates the spectral resolution by a factor $\frac{257}{32} \approx 8$, hence the overall difference in dimensionality is approximately 4. Therefore a relative increase in dimensionality of this factor is required in $h(\cdot)$. Which dimension is upsampled depends on how N is chosen. In practice, both dimensions are resampled to the size of z_i, which allows N to be chosen freely. Note that with this formulation N could also be chosen smaller than the original spectral resolution.

4.1.2 Motivation for $\mathcal{Z}$

Proposition 1. ***Existence of AEs***

Let $X \in \mathcal{X}$, $T \in \mathcal{T}$, and $Y \in \mathcal{Y}$ be statistically dependant random variables following the Markov chain $Y \leftrightarrow X \rightarrow T$
Also let $q(Y|T)$ be a non perfect approximation of the true density $p(Y|T)$. Further, it is assumed that T is a sufficient statistic according to Definition 2.1.1, but is not a minimal sufficient statistic according to Definition 2.1.2. Then, for a realisation $x_i \sim p(X)$ there can exist a statistically insignificant perturbation δ_i that causes the approximate density to change, i.e,

$$q(Y|T = t(x_i)) \neq q(Y|T = t(x_i + \delta_i)). \tag{4.4}$$

Proof. From the sufficient statistic criteria it is known that

$$p(Y|X) = p(Y|T), \tag{4.5}$$

which is expressed for the realisation $x_i \sim p(X)$ as

$$p(Y|X = x_i) = p(Y|T = t(x_i)). \tag{4.6}$$

A statistically insignificant perturbation δ_i can then be expressed as

$$\delta_i \text{ such that } \; p(Y|T = t(x_i)) = p(Y|T = t(x_i + \delta_i)). \tag{4.7}$$

Due to T not being minimal, it is possible that $t(x_i) \neq t(x_i + \delta_i)$, and therefore there might exist a

$$\delta_i \text{ such that } \; q(Y|T = t(x_i)) \neq q(Y|T = t(x_i + \delta_i)). \tag{4.8}$$

Thus there might exist statistically insignificant perturbations δ that can be used to change the approximate density $q(Y|X = x)$ to $q(Y|X = x + \delta)$. □

The practical implication of Proposition 1 is that the non minimal statistic T can be exploited to create a statistically counter-intuitive response from $q_{\theta,F}(Y|X)$. Keeping this in mind, the optimal perturbation domain would be T, since all adversarial examples are defined in this domain. However in a black-box attack T is unknown, and hence a perturbation domain involving a lossy transformation $g(\cdot)$ cannot be used if one wishes to retain all possible AEs in the search space. Using the designed Z-space, *perturbations* that have a psychoacoustic distortion greater than ϵ are not accessible from the Z-space. Note that when using the definition (2.8) of an AE, this reduction of the search space doesn't exclude any *adversarial perturbations* for a particular ϵ and N. When creating inaudible AEs, ϵ is set to 1. As described previously, the phase of the perturbations is aligned to the phase of the original signal, hence perturbations depending on phase are not accessible from Z-space either, however, the same argument as before is applicable; changing the phase in a speech signal isn't a transformation motivated by psyhoacoustics and therefore shouldn't be used for inaudible AEs.

When creating adversarial examples for image domain data, considerations like this can largely be ignored since the algorithms often work on the actual image data and not a perturbation domain representation. Some black-box attacks on audio have utilized MFCC as the perturbation domain, which based on Proposition 1 don't remove any AEs from the search space, if the classifier calculates the same representation internally. However, using such an approach is hard to motivate from a perspective of psychoacoustics, and for unknown classifier architectures.

4.1.3 Limitation of MMT usage

The way perturbations are inserted into the signal (see (4.2)) considers a simplified analysis of the MMT. A rigorous calculation of inaudible perturbations is more complex, following is an explanation of this discrepancy. Firstly, tonal and non-tonal maskers might change due to the insertion of perturbations, thus if a non-tonal masker is reduced by its corresponding value of the MMT, the MMT might no longer be correct for frequencies masked by that particular masker. This could potentially be avoided by only allowing time-frequency components which doesn't effect the MMT to be perturbed, however, this would require extensive processing, and the usage of non-tonal maskers in the psychoacoustic model results in the majority of time-frequency components affecting the MMT.

Another similar limitation is that a masked time-frequency component might become unmasked due to the fact that the MMT is added to that time-frequency component instead of replacing it. However, replacing the component is difficult to motivate since that could cause maskers to be severely reduced and hence change the MMT even more.

Ideally one would like to replace, instead of add, the MMT to time-frequency components that doesn't effect the MMT. This is a potential improvement for the

perturbations domain described. Thus an inaudible perturbation should fulfill

$$MMT_i = MMT_i', \tag{4.9}$$

where MMT_i is the MMT of the original signal x_i and MMT_i' is the MMT of the perturbed signal x_i'. This limitation will not be considered further.

4.2 Optimization Algorithm

Many methods for generating adversarial perturbations have positive correlation between the number of perturbations inserted and the introduced human perceivable distortion. This is one motivation for using the JSMA and C&W-ℓ_0 methods, which focus the perturbations to the most influential dimensions. For some black-box attacks such as the one-pixel attack, limiting ℓ_0 has other benefits, namely, reducing the search-space of the optimization.

In an attack using psychoacoustic modelling, one isn't concerned by the number of perturbations introduced since they have no consequences on the perception of the signal. For a one second long audio signal x_i, its z_i representation has 45507 dimensions for $N = 64$. Any combination of the dimensions can be perturbed without audible effects by adding the corresponding $h(\text{MMT})$ to z_i, i.e.

$$z_i'[m,k] = z_i[m,k] \pm h(\text{MMT}_i)[m,k]. \tag{4.10}$$

Finding a global optimum is hence infeasible, and greedy optimization strategies are preferred.

4.2.1 Masking

The perturbation space, meaning the whole set of possible perturbations can be written as

$$z_i + \gamma \odot h(\text{MMT}_i), \quad \gamma \in [-\epsilon, \epsilon]^{M \times \left(\frac{N}{2}+1\right)} \tag{4.11}$$

where $\odot$ is element-wise multiplication, and ϵ specifies the strength of the perturbations in proportion to the MMT. ϵ is an important parameter since it provides a trade-off between perceptual distortion and the strength of the perturbation. Setting $\epsilon \geq 1$ will according to the psychoacoustic model yield perceivable distortion but it was also found to be required in order to produce effective adversarial examples.

Note that the perturbation doesn't contain any phase distortion, it is instead aligned with the phase of the original signal. This is in line with how the audio watermarking application in [49] utilizes the MMT. Choosing to align the phase to the original signal can be motivated by considering that phase distortion might cause perceivable effects due to constructive or destructive interference between frequencies.

4.2.2 Optimization Metric

Perceptual similarity enforcing regularization is commonly used in algorithms for generating adversarial examples. With the specified perturbation space that is unnecessary since the perceptual model is included within the search space. Although, for values of $\epsilon \geq 1$ regularization might be beneficial for reducing the perceivable distortion, but it won't be explored in this work. The metric sought to optimize therefore solely depends on the classifiers prediction confidence. For an untargeted attack the metric sought to minimize is $q_{\theta,F}(Y = y_i|X = x_i)$, i.e. confidence in the true label. For the targeted attack, one instead aims to maximize $q_{\theta,F}(Y = t|X = x_i)$, i.e. confidence in the target class.

4.2.3 Randomly-Greedy Adversarial Perturbations (R-GAP)

The first algorithm proposed is called randomly-greedy adversarial perturbations (R-GAP). In each iteration it picks a random index pair (m, k) in z_i and replaces it with $z_i[m, k] + \gamma \cdot h(\text{MMT}_i)[m, k]$ where $\gamma = \mathcal{U}(-\epsilon, \epsilon)$. If the replaced index improve adversarial properties the perturbation is kept for the next iteration, else it is reverted to its original value. The algorithmic implementation is presented in Algorithm 1.

An example of the perturbation domain representation of an audio signal is presented in Figure 4.1, together with an adversarial example $\tilde{x}_i$ generated by R-GAP using $N = 32$, $\epsilon = 32$, and optimized for 20000 iterations. The corresponding $MMT(x_i)$ is also shown and the perturbations.

Early Stopping

Early stopping is implemented in order to reduce the perceptual distortion of the adversarial example. This also comes with the added benefit of reducing the time required for generating adversarial examples. The optimization in an untargeted attack setting is stopped when $q_{\theta,F}(Y = y|X = x_i + \rho) < 0.1$, and targeted attacks are attacks are stopped when $q_{\theta,F}(Y = t|X = x_i + \rho) > 0.9$. These thresholds were set by considering that some margin in the classifiers confidence is likely beneficial when generating adversarial examples that are tested for transferability. For brevity, the early stopping is not presented in Algorithm 1. The early stopping criteria is evaluated on each iteration.

Greedy Solution

The algorithm can't find solutions which require two or more dimensions to be changed simultaneously. If every dimensions effect on classification was independent this wouldn't limit the set of possible solutions. However this is likely not the case, hence only localy optimal solutions can be found. In a sense, this is similar to the assump-

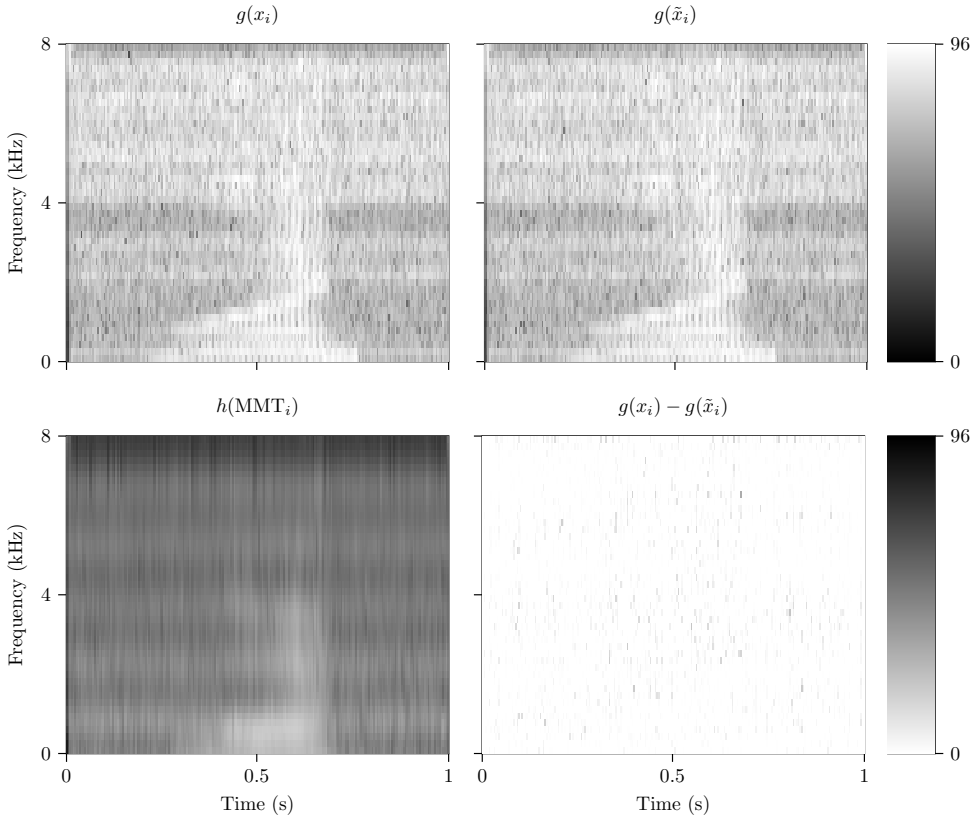

Figure 4.1: Visualisation of $g(x_i)$, $g(\tilde{x}_i)$, $h(MMT(x_i))$, $g(x_i) - g(\tilde{x}_i)$. To ease visualization $g(x_i)$, $g(\tilde{x}_i)$, and $h(MMT(x_i))$ are shown using the upper color mapping while $g(x_i) - g(\tilde{x}_i)$ is shown using the lower color mapping. Further, $g(\cdot)$ uses $N = 256$, while the AE was generated using $N = 32$, as a consequence perturbations become spread over multiple frequencies in the visualization.

tion made for the FGSM, were all dimensions are perturbed in the direction that would enhance adversarial properties for a linear classifier.

Comparing the proposed algorithm with other methods that limits the total number of perturbations (ℓ_0-norm), JSMA perturbs pairs of dimensions which allows it to find solutions requiring multiple dimensions to be changed simultaneously, however performing the JSMA optimization becomes infeasible when the number of candidate dimensions increases [19]. In C&W-ℓ_0 the search space is reduced by an assumption of independence between dimensions. The dimensions that have the least impact on classification at x are removed from the optimization procedure, however, these are not guaranteed to be the same dimensions at $\tilde{x}$, resulting in a limited solution space.

4.2.4 Locally-Greedy Adversarial Perturbations (L-GAP)

The second algorithm starts by initializing a population of U perturbations. In every iteration of the algorithm each individual perturbations 4 neighboring dimensions, together with one random index pair of z_i, are evaluated with the replacement $z_i[m,n] + \gamma \cdot h(\text{MMT}_i)[m,n]$, where $\gamma = \mathcal{U}(-\epsilon, \epsilon)$.

Thus, for every iteration each perturbation is locally optimized. The same early stopping to that in R-GAP is also applied. however, in this algorithm it is mostly useful for reducing the computational cost since the number of perturbations inserted is constant. The algorithmic implementation of L-GAP is presented in Algorithm 2. Note that the early stopping is not shown in the algorithm but is evaluated on each iteration.

Greedy Solution

During initialization, U perturbations are randomly placed on z_i, thereafter only one perturbation is optimized at a time, leading to the limitation that solutions requiring multiple changes simultaneously can't be found. This also results in the possibility of the initial state being less "adversarial" than the original signal. This could potentially be a problem, especially if the set of AEs in $\mathcal{Z}$ is a small subset of $\mathcal{Z}$. However this initial random perturbation resembles how the R+FGSM which has been found to overcome adversarial training defense strategies. Adversarial training isn't explored in this work but it is worth noting that the random initialization might be beneficial in some scenarios.

4.2.5 Reducing Computational Complexity

If $\text{Fs}_x < \text{Fs}_z$, the amount of spectral components that will yield an effect on x is reduced to $\lceil \frac{N(\text{Fs}_x)}{2\text{Fs}_z} \rceil$. Using this knowledge one can reduce the search space of the algorithm. If one knew the sampling rate of the classifier (Fs_q), that information could be utilized to further improve the adversarial search space. However, the sampling rate used in a classifier should be considered unknown in a black-box attack, thus if the classifier downsamples the audio before processing it, that reduction of spectral components can't be utilized. Discussing the sampling rate of the classifier isn't really applicable in the attack scenario investigated in this work, where the attacker directly feeds the audio data to the classifier. For over-the-air-attacks this would be of more interest, since x_i could use a different sampling rate than what is employed by the classification system.

The formulation of both algorithms can be adapted to generate batches of AEs instead, which was done during the implementation of the algorithms.

4.3 Masked FGSM (M-FGSM)

It is difficult to compare the proposed method with previous works on black-box methods since the hyperparameter ϵ doesn't translate into other commonly used measures of perturbations strengths, for example signal-to-noise ratio. To circumvent this issue, the proposed method is instead compared to the FGSM acting on the same perturbation domain representation, thus allowing the hyperparameter to remain the same for the different attacks. To further favor comparability, the FGSM generated adversarial perturbation is limited to the same number of perturbations (U) as the other methods, i.e. only the U most influential dimensions of the perturbation domain representation is effected by the attack. This attack will be referred to as M-FGSM.

Algorithm 1: R-GAP

Input:
x_i, initial audio signal
y_i, corresponding class label
t, target class label (False if untargeted attack)
Output:
$\tilde{x}$, adversarial example
Parameters:
N, size of STFT
ϵ, perturbation strength
J, number of iterations

$j \leftarrow 0$
$(z, c) \leftarrow g(x_i)$
$M \leftarrow$ temporal size(z) // infer temporal size from z
mask $\leftarrow h(\text{MMT}_i)$
$z_{\text{test}} \leftarrow z$
$x_{\text{test}} \leftarrow g^{-1}(z_{\text{test}}, c)$
$x_{\text{best}} \leftarrow x_{\text{test}}$ **for** $j \leftarrow 0$ **to** J **do**
  $(m, k) \leftarrow$ draw random index pair from $[M] \times [\frac{N}{2} + 1]$
  $\gamma \leftarrow \mathcal{U}(-\epsilon, \epsilon)$
  $z_{\text{test}} \leftarrow z_{\text{best}}$
  $z_{\text{test}}[m, k] \leftarrow z[m, k] + \gamma * \text{mask}[m, k])$
  $x_{\text{test}} \leftarrow g^{-1}(z_{\text{test}}, c)$
  if t **then**
    if $q_{\theta,F}(Y = t|X = x_{\text{test}}) > q_{\theta,F}(Y = t|X = x_{\text{best}})$ **then**
      $z_{\text{best}} \leftarrow z_{\text{test}}$
      $x_{\text{best}} \leftarrow x_{\text{test}}$
    end
  else
    if $q_{\theta,F}(Y = y|X = x_{\text{test}}) < q_{\theta,F}(Y = y|X = x_{\text{best}})$ **then**
      $z_{\text{best}} \leftarrow z_{\text{test}}$
      $x_{\text{best}} \leftarrow x_{\text{test}}$
    end
  end
end
$\tilde{x} \leftarrow x_{\text{best}}$

Algorithm 2: L-GAP

Input:
x_i, initial audio signal
y_i, corresponding class label
t, target class label (False if untargeted attack)
Output:
$\tilde{x}$, adversarial example
Parameters:
N, size of STFT
ϵ, perturbation strength
J, number of iterations
U, number of perturbations

$(z, c) \leftarrow g(x_i)$
$M \leftarrow$ temporal size(z) // infer temporal size from z
mask $\leftarrow h(\text{MMT}_i)$
$S_p \leftarrow$ draw uniformly from $\left([M] \times [\frac{N}{2} + 1]\right)^{\times U}$ // initalize set of perturabtion indices
for $(m, k) \in S_p$ **do**
  $\gamma \leftarrow \mathcal{U}(-\epsilon, \epsilon)$
  $z_{\text{best}}[m, k] \leftarrow z[m, k] + \text{mask}[m, k]$ // calculate initial z_{best}
end
$x_{\text{best}} \leftarrow g^{-1}(z_{\text{best}}, c)$
for $j \leftarrow 0$ **to** J **do**
  for $u \leftarrow 0$ **to** U **do**
    $(m_u, k_u) \leftarrow S_p[u]$
    $(m_r, k_r) \leftarrow$ draw random index pair from $([M] \times [\frac{N}{2} + 1])$
    $S_{\text{neighbors}} \leftarrow \{(m_u + 1, k_u), (m_u, k_u + 1), (m_u - 1, k_u), (m_u, k_u - 1), (m_r, k_r)\}$
    for $(m_n, k_n) \in S_{\text{neighbors}}$ **do**
      $\gamma \leftarrow \mathcal{U}(-\epsilon, \epsilon)$
      $z_{\text{test}} \leftarrow z_{\text{best}}$
      $z_{\text{test}}[m_n, k_n] \leftarrow z[m_n, k_n] + \gamma * \text{mask}[m_n, k_n])$
      $x_{\text{test}} \leftarrow g^{-1}(z_{\text{test}}, c)$
      if t **then**
        if $q_{\theta,F}(Y = t|X = x_{\text{test}}) > q_{\theta,F}(Y = t|X = x_{best})$ **then**
          $p[u] \leftarrow (m_n, k_n)$
          $z_{\text{best}} \leftarrow z_{\text{test}}$
          $x_{\text{best}} \leftarrow x_{\text{test}}$
        end
      else
        if $q_{\theta,F}(Y = y|X = x_{\text{test}}) < q_{\theta,F}(Y = y|X = x_{\text{best}})$ **then**
          $p[u] \leftarrow (m_n, k_n)$
          $z_{\text{best}} \leftarrow z_{\text{test}}$
          $x_{\text{best}} \leftarrow x_{\text{test}}$
        end
      end
    end
  end
end
$\tilde{x} \leftarrow x_{\text{best}}$

Chapter 5
Experimental Results

5.1 Evaluating Adversarial Properties

In this chapter, the proposed algorithms are evaluated for targeted and untargeted attacks in a variety of settings. A short description of the two classifiers used for testing is presented in 5.1.1, and all the tests and results relating to adversarial properties are presented in later sections.

5.1.1 Classifier Structures

Two different classifier architectures will be used for evaluation, the architecture is inspired by [18], where 1d-convolutions are applied on raw audio data. Each convolution layer is followed by batch normalization, ReLU activation, and max pooling layers. After the last max pooling the signal is fed through a global average pooling, which reduces the temporal resolution to 1. A fully connected layer is then applied to reduce the size from the channel width to the number of classes (10), on which the softmax function is applied.

The two classifier architectures used will be called F7 and F10, where the integer describe how many convolution layers the classifier uses. A full description of the architectural parameters of the layers is presented in Appendix B.

Dataset

The dataset used for classification is the speech command dataset [17]. The dataset is reduced to 10 classes, using the data from each of the following classes, ”Yes”, ”No”, ”Up”, ”Down”, ”Left”, ”Right”, ”On”, ”Off”, ”Stop”, and ”Go”. The dataset is randomly split into a training partition using 80% of the dataset, a validation set using 15%, and an evaluation set of the remaining 5%. This yields a total number of samples in each partition of $\|D_{train}\|_0 = 18484$, $\|D_{validation}\|_0 = 3467$ and $\|D_{eval}\|_0 = 1158$. From D_{eval} 100 samples are randomly drawn which are used to explore the proposed method of generating adversarial examples, this set is referred to as D_{AE}.

Training Classifers

Each classifier is trained using a batch size of 50. After every 368 batches (corresponding to 1 epoch) the validation set is evaluated Training is discontinued when the accuracy on the validation set hasn't improved during the last 5 evaluations of the validation set. The Adam optimizer is used with the standard parameter values used by PyTorch: $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$. The performance of the classifiers used is listed in Table 5.1

Table 5.1: Classification performance of classifiers, the probability is evaluated as the arithmetic mean of samples fulfilling the equality within $p(\cdot)$

Classifier	$p(\hat{y} = y)$		
	D_{val}	D_{eval}	D_{AE}
$q_{\theta_1,\text{F7}}$	0.903	0.901	0.91
$q_{\theta_2,\text{F7}}$	0.866	0.879	0.88
$q_{\theta_3,\text{F10}}$	0.878	0.884	0.87

5.2 Choosing Hyperparameters (ϵ & N)

Setting $N = 64$ yields the most similar spectral resolution of $\mathcal{Z}$ to that of the MMT. This provides a natural starting point for testing different N. In table Table 5.2a, results are also presented for an untargeted attack for the neighboring power of twos $(32, 128)$. These come with the disadvantage of reducing the similarity to the original psychoacoustic model, and thus the psychoacoustic modelling might be less accurate.

Table 5.2: Accuracy for Hyperparameters on R-GAP

(a) Varying N for $\epsilon = 8$

N	U	Accuracy
32	1653.1	0.55
64	726.0	0.87
128	701.1	0.87

(b) Varying ϵ for $N = 32$

ϵ	U	Accuracy
1	1402.0	0.85
2	1548.0	0.80
4	1658.3	0.68
8	1653.1	0.55
16	1503.8	0.34
32	1270.2	0.19

For testing ϵ the best performing spectral resolution from Table 5.2a is used i.e. $N = 32$. The values of ϵ tested lay in the range of 1 to 32. This range was chosen by

considering the perceptual distortion, in which 1 makes the perturbations impossible to hear according to the psychoacoustic model, and 32 makes for a clearly distorted signal. Apart from the accuracy presented in table Table 5.2b, the change in prediction confidence in the correct class is visualized over iterations in Figure 5.1 and Figure 5.2. The same hyperparamaters will be used during evaluation for both R-GAP and L-GAP, since this favors comparability between the attack algorithms at similar perceptual distortion. Although, it is likely that the same ϵ produces differences in audible distortion due to the fact that the number of perturbations inserted (U) varies between the algorithms. Therefore, one should not consider comparing the miss classification rate between the attack algorithms as fair regarding the average perceptual distortion.

$\epsilon = \{1, 32, 64\}$ were chosen to do more extensive tests on. The values 32 and 64 were chosen taking the performance presented in Table 5.2b into account and the perceptual distortion of the adversarial examples produced. Readers are encouraged to listen to the adversarial examples[1]. Although $\epsilon = 1$ seems to have marginal effect on the classifiers accuracy this value is interesting since it defines the boundary between audible and inaudible perturbations as estimated by the psychoacoustic model. Hence, evaluating this gives insight to the possibilities of creating inaudible adversarial examples. For all tests $N = 32$ is used since it seems to have highly favourable performance in this particular setting.

Looking at the rate of change in the classifiers correct class confidence (shown by the arrows in the plot), it seems that the average rate of change is highest at about 50% correct class confidence, and low between $90 - 100\%$. Effectively this means that $q_\theta(Y = y|X = x)$ is an indication of how computationally expensive it will be for the algorithm to produce an adversarial example from x, where a low initial $q_\theta(Y = y|X = x)$ indicate that the signal is likely to be successfully perturbed while a high confidence in the correct class instead indicates that it is difficult to perturb the signal.

5.3 Untargeted

The result from the untargeted attacks is presented in Table 5.3. The hyperparameters used are the ones presented in Subsection. 5.2, i.e. $\epsilon = \{1, 32, 64\}$ and $N = 32$. These values of epsilon correspond to roughly $\{0, 15, 18\}$ in dB.

5.3.1 Time Complexity

For the chosen number of iterations and perturbations, L-GAP evaluates $g^{-1}(\cdot)$ and $q_{\theta,F}$ a maximum of $100,000$ times depending on the number of iterations before the

[1] `https://github.com/Srll/Adversarial-Attacks-Pytorch`

Table 5.3: Untargeted Attack Success Rate

Algorithm	ϵ	U	$p(\tilde{y} \neq y)$	Time (s/AE)
R-GAP	1	1402.0	0.15	316.3
R-GAP	32	1270.2	0.81	184.3
R-GAP	64	1014.9	0.87	116.9
L-GAP	1	1402	0.09	1542.4
L-GAP	32	1270	0.79	931.7
L-GAP	64	1015	0.86	809.2
M-FGSM	1	1402	0.09	3.6
M-FGSM	32	1402	0.13	2.8
M-FGSM	64	1402	0.15	3.1

early stopping criteria is met. Similarly, R-GAP performs at most $20,000$ evaluations. Thereby, R-GAP is expected to be approximately 5 times faster at generating AEs. For reference $q_{\theta,F}$ is evaluated on a Nvidia RTX2060, while the remaining calculations are performed on the CPU (Intel I7-8700k). With this setup it is possible to run 4 R-GAP or L-GAP algorithms in parallel. The generation time presented in Table 5.3 does not account for running 4 algorithms in parallel.

5.4 Targeted

The result from targeted attacks is shown in Table 5.4. Due to computational complexity only one target class will be considered when testing the targeted attack. The target class was set to class 3 ("down"), the mean targeted accuracy is calculated as the percentage of "AEs" fulfilling (2.8). The evaluation is done on 91 samples, which happened to be the number of samples that were not equal to the target class out of the 100 samples in D_{AE}.

5.5 Transferability

Transferability is evaluated both on different classifiers (cross-technique transferability) and different realisations of the parameters θ on the same architecture (intra-technique transferability) for the untargeted and targeted attacks. The same 100 AEs from the untargeted attack, and the 91 samples from the targeted attack is reused for the evaluation of transferability. The result is shown in Table 5.5 for untargeted attacks, and Table 5.6 for targeted attacks. The difference in success rate between the original classifier $q_{\theta_1,\text{F7}}$ and the classifier being tested for transferability is denoted

as $\Delta q_{\theta_1,\text{F7}}$, i.e. $p(\hat{q}_{\theta_2,\text{F7}}(Y|X=\tilde{x})=y) - p(\hat{q}_{\theta_1,\text{F7}}(Y|X=\tilde{x})=y)$ for the intra-technique transferability, and $p(\hat{q}_{\theta_3,\text{F10}}(Y|X=\tilde{x})=y) - p(\hat{q}_{\theta_1,\text{F7}}(Y|X=\tilde{x})=y)$ for the cross-technique transferability.

Table 5.4: Targeted Attack Success Rate

Algorithm	ϵ	U	$p(\tilde{y}=t)$
R-GAP	32	1999.2	0.48
R-GAP	64	1749.9	0.64
L-GAP	32	1999	0.53
L-GAP	64	1750	0.67
M-FGSM	32	1999	0.0
M-FGSM	64	1750	0.0

Table 5.5: Transfer Rate for Untargeted Attack

Algorithm	Classifier	ϵ	$p(\tilde{y}\neq y)$	$\Delta q_{\theta_1,\text{F7}}$
R-GAP	$q_{\theta_2,\text{F7}}$	32	0.17	–0.64
R-GAP	$q_{\theta_2,\text{F7}}$	64	0.20	–0.67
R-GAP	$q_{\theta_3,\text{F10}}$	32	0.17	–0.64
R-GAP	$q_{\theta_3,\text{F10}}$	64	0.15	–0.72
L-GAP	$q_{\theta_2,\text{F7}}$	32	0.18	–0.56
L-GAP	$q_{\theta_2,\text{F7}}$	64	0.18	–0.58
L-GAP	$q_{\theta_3,\text{F10}}$	32	0.15	–0.58
L-GAP	$q_{\theta_3,\text{F10}}$	64	0.17	–0.64

Table 5.6: Transfer Rate for Targeted Attack

Algorithm	Classifier	ϵ	$p(\tilde{y}=t)$	$\Delta q_{\theta_1,\text{F7}}$
R-GAP	$q_{\theta_2,\text{F7}}$	32	0.20	–0.28
R-GAP	$q_{\theta_2,\text{F7}}$	64	0.24	–0.40
R-GAP	$q_{\theta_3,\text{F10}}$	32	0.02	–0.46
R-GAP	$q_{\theta_3,\text{F10}}$	64	0.02	–0.62
L-GAP	$q_{\theta_2,\text{F7}}$	32	0.19	–0.17
L-GAP	$q_{\theta_2,\text{F7}}$	64	0.25	–0.35
L-GAP	$q_{\theta_3,\text{F10}}$	32	0.00	–0.33
L-GAP	$q_{\theta_3,\text{F10}}$	64	0.00	–0.53

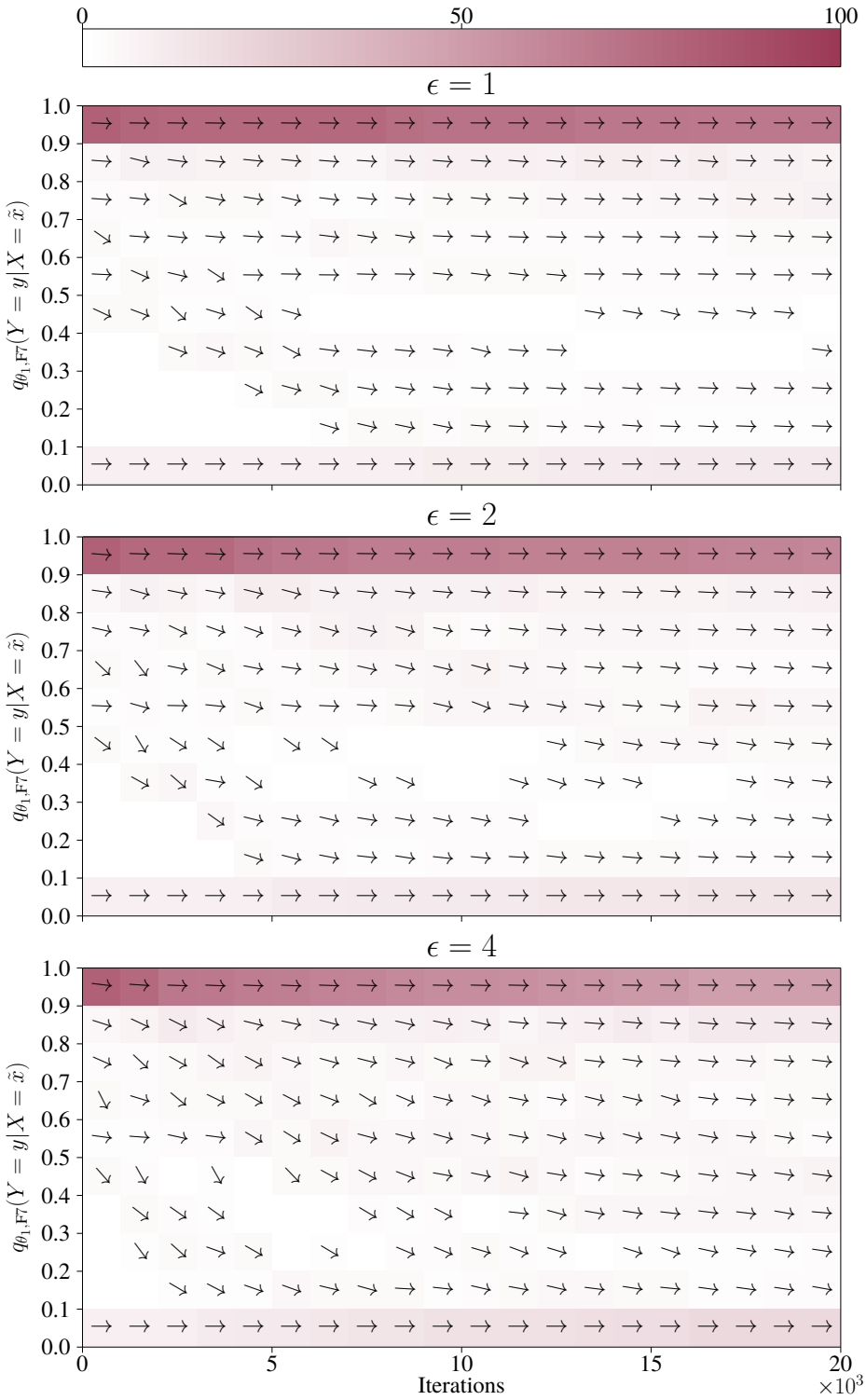

Figure 5.1: Change in accuracy over iterations of R-GAP, for $\epsilon = \{1, 2, 4\}$, the angle of the arrows indicate the mean rate of change of the samples in each bin over 1000 iterations. The color intensity correspond to the number of signals in each bin.

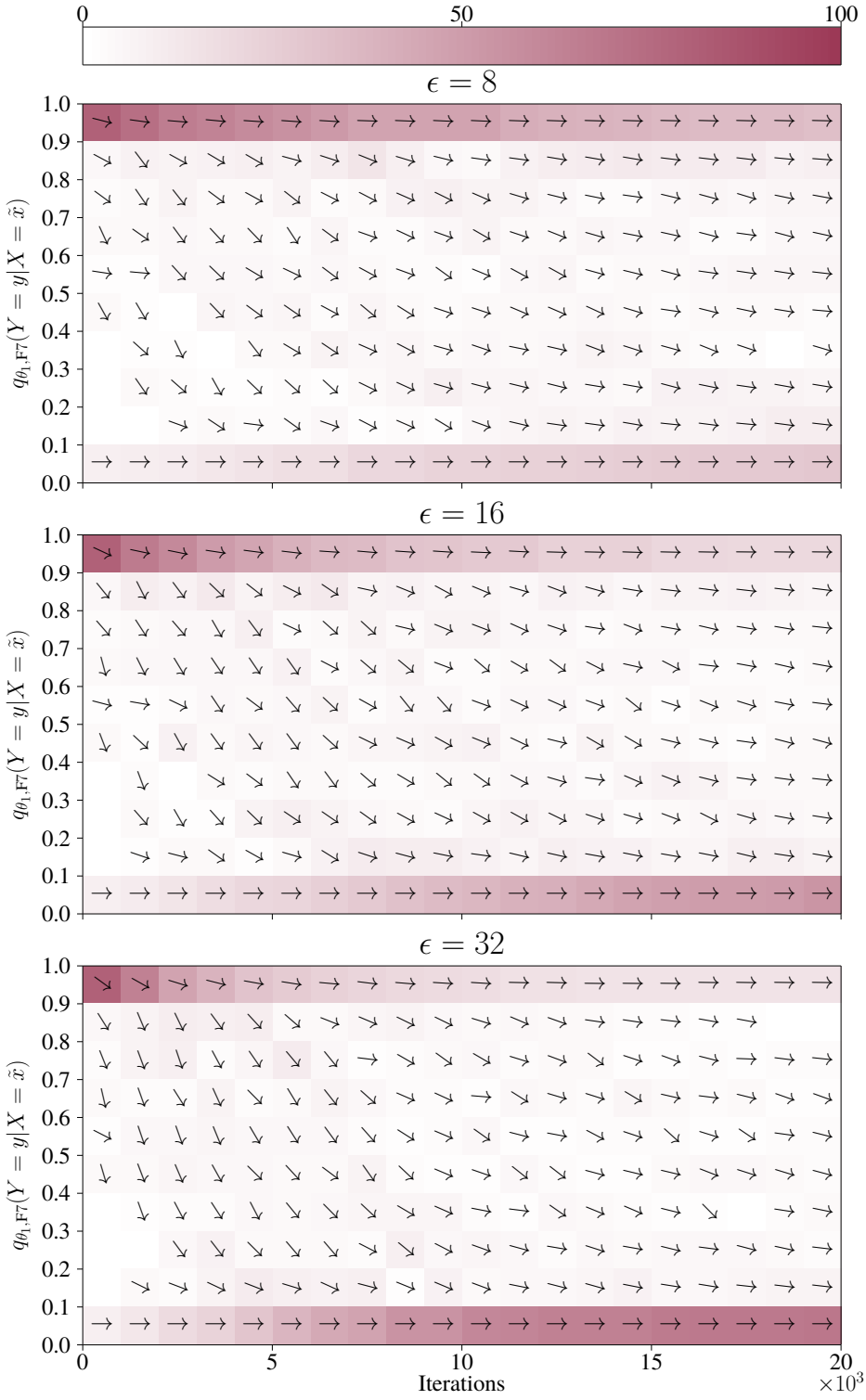

Figure 5.2: Change in accuracy over iterations of R-GAP, for $\epsilon = \{8, 16, 32\}$, the angle of the arrows indicate the mean rate of change of the samples in each bin over 1000 iterations. The color intensity correspond to the number of signals in each bin.

Chapter 6
Discussion

The main contribution of this thesis is to provide a well motivated method of utilizing psychoacoustic modelling for generating adversarial examples. The method developed does not make any inherent assumptions regarding the classification system and can thus be used on any KWS system with access to the soft classification output, while simultaneously retaining all inaudible perturbations in the search space.

The developed algorithms for generating adversarial perturbations in accordance with the masking threshold exploits that the ℓ_0-norm of the perturbations does not effect the perceptual distortion of the perturbation. In practise this means that a vast number of perturbations can be inserted, consequently the search space becomes large and greedy optimization strategies is used to find local optimal solutions. The algorithms developed showcase how the masking threshold can be used to create inaudible perturbations. The algorithms should work effectively on any type of KWS system, since the optimization strategies similarly to the perturbation domain have little inherent assumptions of the classification problem, in the case of optimization this means the curvature of the optimization problem. This is especially true for R-GAP, in contrast to L-GAP, which is expected to perform favourably on locally smooth optimization problems. Whether or not this is the case for general KWS problems is unclear. The results do however show that R-GAP generate stronger untargeted adversarial attacks than L-GAP. This is somewhat surprising since the number of randomly chosen perturbations are the same between the two algorithms (20,000), and that L-GAP also locally optimizes the perturbations. This could possibly be explanation by the initialization process of perturbations in L-GAP. The problem of bad initial states should be more severe in the case of targeted attacks, since the set of favourable initialization is smaller. This does however not seem to be the case, since L-GAP yield more effective attacks.

The results show that the M-FGSM acting on the perturbation domain representation provides limited adversarial properties in comparison to the black-box methods proposed. This isn't surprising when considering that the M-FGSM applies a single linear perturbation on the perturbation domain representation, unlike the black-box methods which perform multiple non-linear optimization steps. It's worth noting that

the limited capabilities of the FGSM in this scenario indicate that the problem of finding inaudible adversarial perturbations is a non-linear problem.

The results from Table 5.3 indicate that in order to create effective untargeted adversarial perturbations using the R-GAP and L-GAP algorithms one needs to set $\epsilon > 1$, which causes the perturbations to be audible according to the psychoacoustic model. This does not prove that inaudible perturbations are impossible. However, the results conform with the results in [13, 15], where psychoacoustic modelling was utilized to limit the perceptual impact of the perturbations, true inaudible perturbations was not achieved for the ASR systems tested. It is possible that other optimization strategies could find perturbations with superior adversarial properties to those found by R-GAP and L-GAP, thus effective inaudible adversarial perturbations might still be possible. This work demonstrates that in order to find such perturbation the naive greedy algorithms tested can't be used to find effective adversarial examples for this particular classifier and dataset. For classification robustness this finding should be considered positive, since it is indicated that inaudible adversarial perturbations are difficult to produce for classifier which hasn't been created without any strategies for achieving robustness to adversarial perturbations.

According to Table 5.2a, higher temporal resolution performs favourably to high spectral resolution of the perturbation domain. Using a small N results in spectral components being limited in time such that the period length become shorter than the frame of the STFT. This likely results in perturbations with small N being perceived as transients rather than tonal components. When listening to generated AEs this become apparent, the perturbations are perceived similar to noise rather than harmonics. Whether or not this is desirable is unclear, arguably non-tonal noise is a more naturally occurring phenomena in speech recordings compared to tonal noise, hence it could be a desirable property if ones goal is to create perturbations that sound less suspicious. In any case, by increasing N, the proposed perturbation domain could also be used to create tonal noise.

Although the goal of generating adversarial inaudible adversarial perturbations was not achieved, it is demonstrated that by increasing ϵ, psychoacoustically motivated perturbations can be generated that are effective for both untargeted Table 5.3 and targeted Table 5.4 attacks in a black-box setting. As expected, targeted AEs are more difficult to generate than untargeted. This is materialized when comparing the percentage of successful attacks of the targeted attack and the accuracy on AEs generated in the untargeted attack.

The transferability property of the generated AEs was demonstrated to be weak, see Table 5.5 and Table 5.6. This is somewhat expected due to transferability often being a property that stems from linearity in attacks.

Chapter 7
Conclusions

The main contribution of this work to the field of AEs is the development of a method for inserting inaudible perturbations in an audio signal. Although this has been explored in previous work, the suggested algorithm uses the full psychoacoustic model-1 with the limited hyperparameters, ϵ which is psychoacoustically motivated and the spectral resolution N which is motivated from a signal processing perspective. The perturbation domain representation of the signal facilitates that any optimization strategy that doesn't require a gradient easily can be implemented to produce adversarial perturbations that are motivated by pschoacoustic modelling.

Finally, to recap on the research questions posed in the beginning:

Research Question 1: *Is it possible to generate inaudible adversarial perturbations in a Black-box setting for a KWS task?*

Although effective inaudible adversarial perturbations were not achieved for the KWS problem explored, by allowing perturbations to be audible but psychoacoustically limited, AEs can be produced with low perceptual distortion that are effective at creating counter intuitive responses from the KWS system.

Research Question 2: *Are such adversarial examples transferable between different classifiers or different classifier parameters?*

Since inaudible perturbations weren't achieved, it is difficult to answer whether or not the resulting AEs can be transferable or not. However, the AEs found when allowing the perturbations to be perceivable seem to largely not be transferable for the optimization algorithms tested.

Bibliography

[1] Hiromu Yakura and Jun Sakuma. “Robust Audio Adversarial Example for a Physical Attack”. In: *CoRR* abs/1810.11793 (2018). arXiv: 1810.11793. URL: http://arxiv.org/abs/1810.11793.

[2] Ivan Evtimov et al. “Robust Physical-World Attacks on Machine Learning Models”. In: *CoRR* abs/1707.08945 (2017). arXiv: 1707.08945. URL: http://arxiv.org/abs/1707.08945.

[3] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. “Adversarial examples in the physical world”. In: *CoRR* abs/1607.02533 (2016). arXiv: 1607.02533. URL: http://arxiv.org/abs/1607.02533.

[4] Nicholas Carlini et al. “Hidden Voice Commands”. In: *USENIX Security Symposium*. 2016.

[5] Christian Szegedy et al. “Intriguing properties of neural networks”. In: *CoRR* abs/1312.6199 (2014).

[6] Battista Biggio et al. “Evasion Attacks against Machine Learning at Test Time”. In: *CoRR* abs/1708.06131 (2017). arXiv: 1708.06131. URL: http://arxiv.org/abs/1708.06131.

[7] Nicholas Carlini and David Wagner. *Towards Evaluating the Robustness of Neural Networks*. 2016. arXiv: 1608.04644 [cs.CR].

[8] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. *DeepFool: a simple and accurate method to fool deep neural networks*. 2015. arXiv: 1511.04599 [cs.LG].

[9] Nicolas Papernot et al. “The Limitations of Deep Learning in Adversarial Settings”. In: *2016 IEEE European Symposium on Security and Privacy (EuroS&P)* (Mar. 2016). DOI: 10.1109/eurosp.2016.36. URL: http://dx.doi.org/10.1109/EuroSP.2016.36.

[10] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. *Explaining and Harnessing Adversarial Examples*. 2014. arXiv: 1412.6572 [stat.ML].

[11] Florian Tramèr et al. *Ensemble Adversarial Training: Attacks and Defenses*. 2017. arXiv: 1705.07204 [stat.ML].

[12] Lea Schönherr et al. *Imperio: Robust Over-the-Air Adversarial Examples for Automatic Speech Recognition Systems*. 2019. arXiv: 1908.01551 [cs.CR].

[13] Yao Qin et al. *Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition*. 2019. arXiv: 1903.10346 [eess.AS].

[14] Rohan Taori et al. *Targeted Adversarial Examples for Black Box Audio Systems*. 2018. arXiv: 1805.07820 [cs.LG].

[15] Lea Schonherr et al. "Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding". In: *Proceedings 2019 Network and Distributed System Security Symposium* (2019). DOI: 10.14722/ndss.2019.23288. URL: http://dx.doi.org/10.14722/ndss.2019.23288.

[16] Robert Geirhos et al. "Generalisation in Humans and Deep Neural Networks". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 2018, pp. 7549–7561.

[17] Pete Warden. *Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition*. 2018. arXiv: 1804.03209 [cs.CL].

[18] W. Dai et al. "Very deep convolutional neural networks for raw waveforms". In: (2017), pp. 421–425.

[19] Nicholas Carlini and David Wagner. *Audio Adversarial Examples: Targeted Attacks on Speech-to-Text*. 2018. arXiv: 1801.01944 [cs.LG].

[20] Moustapha Cisse et al. *Houdini: Fooling Deep Structured Prediction Models*. 2017. arXiv: 1707.05373 [stat.ML].

[21] Dan Hendrycks et al. *Natural Adversarial Examples*. 2020. arXiv: 1907.07174 [cs.LG].

[22] Eric Wong, Leslie Rice, and J. Zico Kolter. *Fast is better than free: Revisiting adversarial training*. 2020. arXiv: 2001.03994 [cs.LG].

[23] Mahmood Sharif et al. "Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS '16. Vienna, Austria: Association for Computing Machinery, 2016, pp. 1528–1540. ISBN: 9781450341394. DOI: 10.1145/2976749.2978392. URL: https://doi.org/10.1145/2976749.2978392.

[24] Naftali Tishby, Fernando Pereira, and William Bialek. "The Information Bottleneck Method". In: *Proceedings of the 37th Allerton Conference on Communication, Control and Computation* 49 (July 2001).

[25] Arne Leijon and Gustav Eje Henter. *Pattern Recognition: Fundamental Theory and Exercise Problems*. 2015 ed. Stockholm, Sweden: School of Electrical Engineering, KTH Royal Institute of Technology, 2012.

[26] Jonathan T. Foote. "Content-based retrieval of music and audio". In: *Multimedia Storage and Archiving Systems II*. Ed. by C.-C. Jay Kuo, Shih-Fu Chang, and Venkat N. Gudivada. Vol. 3229. International Society for Optics and Photonics. SPIE, 1997, pp. 138–147. DOI: 10.1117/12.290336. URL: https://doi.org/10.1117/12.290336.

[27] D. Gerhard and University of Regina. Dept. of Computer Science. *Audio Signal Classification : History and Current Techniques*. Technical report (University of Regina. Dept. of Computer Science). Department of Computer Science, University of Regina, 2003. ISBN: 9780773104563. URL: https://books.google.se/books?id=mOqBtgAACAAJ.

[28] Li Deng, Geoffrey Hinton, and Brian Kingsbury. "New types of deep neural network learning for speech recognition and related applications: An overview". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE. 2013, pp. 8599–8603.

[29] Xiu Zhang et al. "SIFT-based local spectrogram image descriptor: a novel feature for robust music identification". eng. In: *EURASIP Journal on Audio, Speech, and Music Processing* 2015.1 (2015), pp. 1–15. ISSN: 1687-4722.

[30] Yandre M.G. Costa, Luiz S. Oliveira, and Carlos N. Silla. "An evaluation of Convolutional Neural Networks for music classification using spectrograms". In: *Applied Soft Computing* 52 (2017), pp. 28–38. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2016.12.024. URL: http://www.sciencedirect.com/science/article/pii/S1568494616306421.

[31] E. J. Humphrey and J. P. Bello. "Rethinking Automatic Chord Recognition with Convolutional Neural Networks". In: *2012 11th International Conference on Machine Learning and Applications*. Vol. 2. 2012, pp. 357–362.

[32] Lonce Wyse. "Audio Spectrogram Representations for Processing with Convolutional Neural Networks". In: *CoRR* abs/1706.09559 (2017). arXiv: 1706.09559. URL: http://arxiv.org/abs/1706.09559.

[33] S. S. Stevens, J. Volkmann, and E. B. Newman. "A Scale for the Measurement of the Psychological Magnitude Pitch". In: *The Journal of the Acoustical Society of America* 8.3 (1937), pp. 185–190. DOI: 10.1121/1.1915893. eprint: https://doi.org/10.1121/1.1915893. URL: https://doi.org/10.1121/1.1915893.

[34] Daniel Povey et al. "The Kaldi speech recognition toolkit". In: *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding* (Jan. 2011).

[35] Tom B. Brown et al. “Adversarial Patch”. In: *CoRR* abs/1712.09665 (2017). arXiv: 1712.09665. URL: http://arxiv.org/abs/1712.09665.

[36] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. “One pixel attack for fooling deep neural networks”. In: *CoRR* abs/1710.08864 (2017). arXiv: 1710.08864. URL: http://arxiv.org/abs/1710.08864.

[37] Rey Reza Wiyatno et al. *Adversarial Examples in Modern Machine Learning: A Review*. 2019. arXiv: 1911.05268 [cs.LG].

[38] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. *Adversarial Machine Learning at Scale*. 2016. arXiv: 1611.01236 [cs.CV].

[39] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. arXiv: 1412.6980 [cs.LG].

[40] Nicolas Papernot, P. McDaniel, and Ian J. Goodfellow. “Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples”. In: *ArXiv* abs/1605.07277 (2016).

[41] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. *Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples*. 2016. arXiv: 1605.07277 [cs.CR].

[42] Y. Liu et al. “Delving into Transferable Adversarial Examples and Black-box Attacks”. In: *ArXiv* abs/1611.02770 (2017).

[43] Moustafa Alzantot, Bharathan Balaji, and Mani Srivastava. *Did you hear that? Adversarial Examples Against Automatic Speech Recognition*. 2018. arXiv: 1801.00554 [cs.CL].

[44] Shreya Khare, Rahul Aralikatte, and Senthil Mani. *Adversarial Black-Box Attacks on Automatic Speech Recognition Systems using Multi-Objective Evolutionary Optimization*. 2018. arXiv: 1811.01312 [cs.CR].

[45] Jon Vadillo and Roberto Santana. *On the human evaluation of audio adversarial examples*. 2020. arXiv: 2001.08444 [eess.AS].

[46] Marina Bosi. *Introduction to digital audio coding and standards*. eng. Kluwer international series in engineering and computer science. 2003. ISBN: 1-4615-0327-2.

[47] Jayaraman Jayaraman Thiagarajan. *Analysis of the MPEG-1 layer III (MP3) algorithm using MATLAB*. eng. Synthesis digital library of engineering and computer science. San Rafael, Calif. (1537 Fourth Street, San Rafael, CA 94901 USA): Morgan & Claypool, 2012. ISBN: 1-60845-802-4.

[48] Harvey Fletcher and W. A. Munson. “Loudness, Its Definition, Measurement and Calculation”. In: *The Journal of the Acoustical Society of America* 5.2 (1933), pp. 82–108. DOI: 10.1121/1.1915637. eprint: https://doi.org/10.1121/1.1915637. URL: https://doi.org/10.1121/1.1915637.

[49] Yiqing Lin and Waleed H Abdulla. *Audio Watermark: A Comprehensive Foundation Using MATLAB*. eng. 2015th ed. Cham: Springer International Publishing, 2014. ISBN: 9783319079738.

[50] Francesco Camastra and Alessandro Vinciarelli. *Machine Learning for Audio, Image and Video Analysis: Theory and Applications*. eng. Advanced Information and Knowledge Processing. London: Springer London, 2015. ISBN: 978-1-4471-6734-1.

Appendix A
Psychoacoustics

Table A.1: Frequency related constants defined in the MPEG-1 standard

(a) Mapping between spectral subsamples and subbands

Subsamples	Frequency Index k	Subbands
0-48	0-48	0-6
48-72	48-96	6-12
72-106	96-224	12-29
106	232-256	29-32

(b) Lower and upper frequency for bins used in calculation of nontonal maskers

Lower Frequency	Upper Frequency
0	100
100	200
200	300
300	400
400	510
510	630
630	770
770	920
920	1080
1270	1480
1480	1720
1720	2000
2000	2320
2320	2700
2700	3150
3150	3700
3700	4400
4400	5300
5300	6400
6400	7700
7700	9500
9500	12000
12000	15500
15000	22100

Appendix B
Classifier Architecture

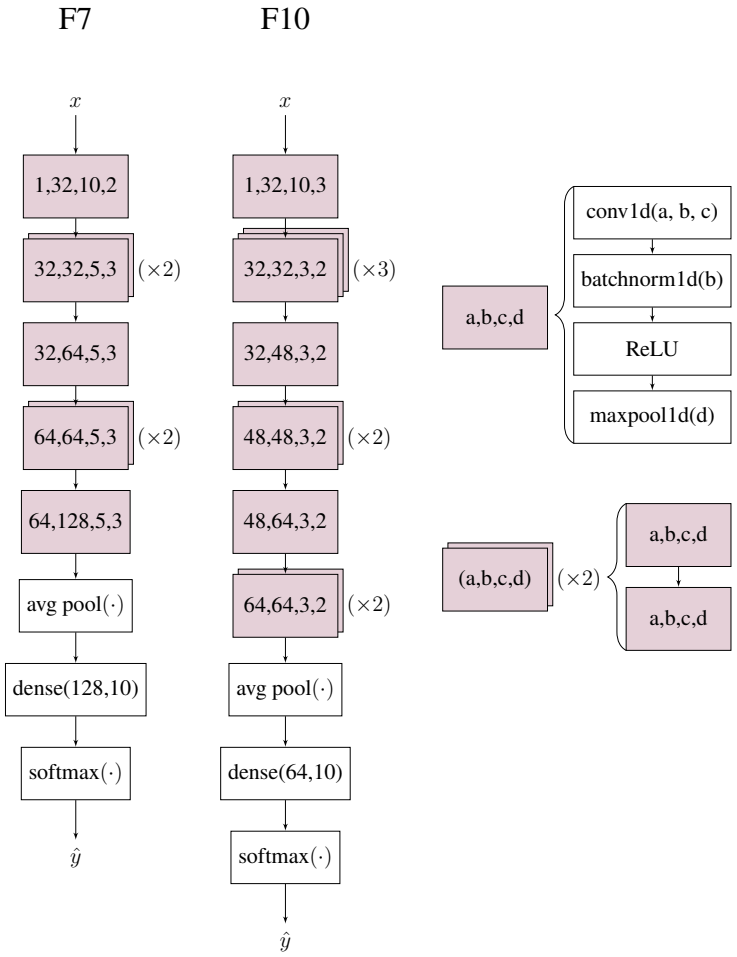

Figure B.1: Classifier architecture, conv1d takes arguments `in_channels`, `out_channels`, `kernel_size`, batchnorm1d takes argument `num_features`, maxpool1d takes argument `kernel_size`. The stride of conv1d is set to 1 for all convolutions. The dense layer reduces the dimensional from the channel depth to the number of classes (10).

TRITA-EECS-EX-2021:90