Computational Models for Auditory Scene Analysis

by

Shariq A. Mobin

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Neuroscience

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Bruno Olshausen, Chair
Professor Frederic Theunissen
Professor Michael Yartsev
Professor Michael DeWeese

Spring 2019

Computational Models for Auditory Scene Analysis

Abstract

Computational Models for Auditory Scene Analysis

by

Shariq A. Mobin

Doctor of Philosophy in Neuroscience

University of California, Berkeley

Professor Bruno Olshausen, Chair

The high levels goals of this thesis are to: understand the neural representation of sound, produce more robust statistical models of natural sound, and develop models for top-down auditory attention. These are three critical concepts in the auditory system. The neural representation of sound should provide a useful representation for building robust statistical models and directing attention. Robust statistical models are necessary for humans to generalize their knowledge from one domain to the plethora of domains in the real world. And attention is fundamental to the perception of sound, allowing one to prioritize information in the raw audio signal.

First, I approach understanding the neural representation of sound using the efficient coding principle and the physiological characteristics of the cochlea. A theoretical model is developed using convolutional filters and leaky-integrate-and-fire (LIF) neurons to model the cochlear transform and spiking code of the auditory nerve. The goal of this model is to explain the distributed phase code of the auditory nerve response but it lays the foundation for much more.

Second, I investigate an algorithm for audio source separation, called deep clustering. Experiments are performed to evaluate it's robustness, and a new neural network architecture is developed to improve robustness. The experiments show that the conventional recurrent neural network performs sub-optimally, and our dilated convolutional neural network improves robustness while using an order of magnitude fewer parameters. This more parsimonious model is a step towards models which are minimally parameterized and generalize well across many domains.

Third, I develop a new algorithm to address the limitations of the previous deep clustering method. This algorithm can extract multiple sources at once from a mixture using an attentional context or bias. It relies on modulating the computation of the bottom-up pathway using a top-down neural signal, which indicates which sources are of interest. A simple idea from the attentional spotlight method is used to do this: to allow for the top-down neural signal to modulate the gain on a set of low level neurons. This computational method demonstrates one way top-down feedback could direct auditory attention in the

brain. Interestingly, this method goes beyond neuroscience, it demonstrates that attention can be about more than efficient computation. The experiments show that it resolves one of the main short comings of deep clustering. The model can extract sources from a mixture without knowing the total number of sources in the mixture, unlike deep clustering.

The major contributions of this work are a theoretical model for the auditory nerve response, a more robust neural network architecture for sound understanding, and a novel and powerful model of top-down auditory attention. I hope that the first contribution will be used to build a better understanding of the complex auditory nerve code. The second to build ever more parsimonious and robust models of source separation. And the third to provide a basis for an under-explored research direction which I believe is the most fruitful for building human-level auditory scene analysis, attention-based source separation.

Acknowledgments

I thank my father for instilling in me perseverance, curiosity, and to have borderline crazy dreams. My mother for humility, levelheadedness, and independence. My brother for crafty thinking, intelligence, and a passion for science and technology. And all of them for their undying support and encouragement over the many years of my education.

I thank Bruno Olshausen, my advisor, for teaching me how to be a great scientist, to ask the right questions, and to dive deeper into subjects. His continuing love of knowledge, passion for inspiring new scientists, and excitement for the future all make the Redwood Center an incredible place to occupy. His ideas and perspectives inspire much of my work.

I thank Brian Cheung for being an incredible friend and collaborator. I spent countless hours bouncing ideas off Brian over the years and his insights were instrumental to my success.

I thank my thesis committee for their feedback and encouragement of my work. Mike Deweese for his astounding excitement, Frederic Theunissen for his pointed comments, and Michael Yartsev for his continued support.

I thank the entire Redwood Center for Theoretical Neuroscience. The Redwood is full of brilliant neuroscientists, statisticians, and physicists who have all contributed a great deal to my scientific skill set. Special thanks to Fritz Sommer who provided early guidance and encouragement to me, leading me to publish my first machine learning paper. I also want to thank my colleagues Alex Anderson, Yubei Chen, Jasmine Collins, Charles Frye, Dylan Paiton, Sophia Sanborn, and Eric Weiss for excellent discussions and banter that kept me going.

To my outside mentor, Dick Lyon, for teaching me about the real-world implications of my research and connecting me with brilliant engineers and scientists at Google Brain.

And finally to all my friends in the Helen Wills Neuroscience Institute and the Bay Area. Especially Shomik Chakravarty, Irene Grossrubatscher, Maimon Rose, Andrew Saada, Daniel Saltiel, and Jessie Temple for their incredible friendship and energy.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

A number of questions have puzzled me during my study of auditory neuroscience. 1) Why does the cochlea choose to represent sound in a time-frequency representation? 2) How do we build human-level robust statistical models of sound? 3) What role does feeback play in attention? While tackling all these problems may seem daunting at first, the belief that the brain computes the most parsimonious model of sound to accomplish a behaviour goal can be quite powerful. But what should that behavioural goal be? One difficulty that all animals face is that the sound waveform that enters our ears is often a mixture of many individual sources. Yet our conscious experience of sound is made up of the individual sources themselves, demixed, which is critical to our survival. In much of my research I will make progress on these three questions by using a combination of learning algorithms, large models that make efficient use of their parameters, lots of data, and this behavioural goal. In this chapter, I provide background on the physiology of the auditory pathway, discuss different representations of sound, and finally review the idea of auditory scene analysis and how neural networks can be applied to it.

1.2 Auditory Physiology

In this section background knowledge on the physiology of the cochlea as well as some details of the entire auditory pathway are provided. The information in this section informs the theoretical model in Chapter 2 but is not required for understanding it, nor the remaining the chapters.

Subcortical Physiology

The Mechanical Representation of Sound

Sound vibrations enter the ear canal and cause vibrations of the ear drum. These vibrations are then amplified by three bones, the Malleus, Incus, and Stapes in order to amplify the input signal in preparation for the fluid-filled chamber of the cochlea. The cochlea acts a mechanical frequency analyzer through the mechanical properties of the basilar membrane and fluids that fill it. The basilar membrane is narrow and stiff at the basal end of the cochlea but progressively more wide and floppy towards the apical end. This mechanical property leads high frequencies to vibrate more basal parts of the basilar membrane while low frequencies vibrate more apical ends. Thus, if the input waveform contains a single frequency then the place of maximal vibration in the basilar membrane gives a good indication of what frequency is contained within the signal. Because of the frequency analysis property of the cochlea it is often described as a biological Fourier analyzer. However, if the cochlea is thought of as a set of mechanical filters, they differ in some fundamental ways as compared to the Fourier filters. These differences include the logarithmic frequency spacing of the cochlear filters as well as their tuning bandwidth being approximately one-sixth of the center frequency of the filter. To address these differences and more, a gamma-tone filter bank is typically used. It is important to note that modeling the cochlea using this filter follows from a linear assumption about the cochlea which is not true. This will become clear a little later. However, in practice, this assumption is a reasonable first approximation.

Transduction: From Vibrations to Voltage

The next stage of auditory processing is the conversion of vibrations in the basilar membrane into a voltage change of the spiral ganglion. This conversion takes place in the Organ of Corti, a structure attached to the basilar membrane. Inside the Organ of Corti lie sensory hair cells which are the specific components that transduce the mechanical vibrations. They accomplish this task through small hairs called stereocilia which sit on top of them. These hairs form mechano-gated potassium channels with the sensory hair cells. That is, in response to vibrations, these stereocilias sway back and forth, causing a potassium channel to open which depolarizes the hair cells. The hair cells voltage will then mimic the basilar membranes vibration as an alternating current (AC), at least up to 1kHz (Fig. 1.1) [52].

Up until now all hair cells have been treated as equivalent. However, an important distinction is made between inner hair cells (IHCs) and outer hair cells (OHCs). Inner hair cells are purely sensory neurons which simply act to transduce the vibrations of the basilar membrane. Outer hair cells also transduce these vibrations but have an additional astounding property - they oscillate the lengths of their cell bodies according to their membrane voltage, thus providing a feedback amplification process. This amplification process is critical to our ability to discriminate between small changes in frequency as well as our ability to hear sounds at the enormous range of over 120dB [66]. Mathematically, this amplification is often

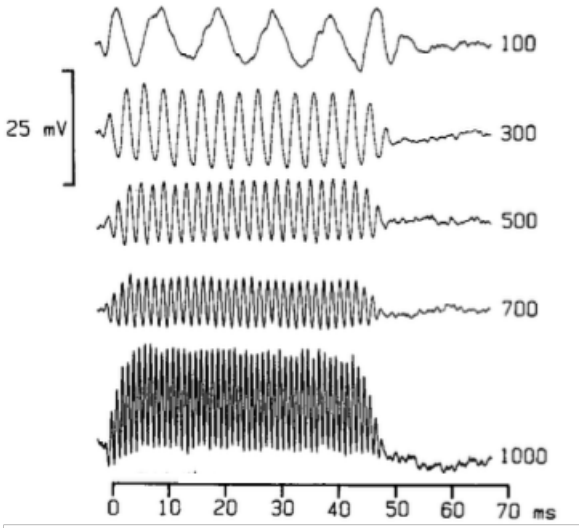

Figure 1.1: Five individual voltage recordings of sensory hair cells, each in response to a pure sinusoidal frequencies indicated on the right. If you count the peaks, you will notice that the voltage oscillates at the frequency of the stimulus. Image adapted from [52].

described as a compressive nonlinearity (Fig. 1.2) [69] which acts to amplify small sounds a great deal, but back off for louder sounds. This nonlinearity will be revisited later.

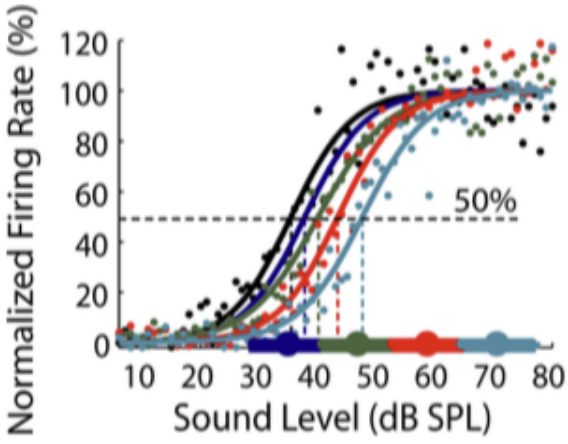

Figure 1.2: Normalized firing rate along the auditory nerve in response to auditory stimulus presented at different volumes (dB SPL). The colors indicate what volume the cochlea was allowed to adapt to before performing the volume sweep. When the cochlea adapts to loud volumes the curve shifts right but the shape of the curve remains similar. Adapted from [69].

Spikes: From Voltage Fluctuations to Action Potentials

The hair cells of the spiral ganglion form glutamatergic, excitatory synaptic connections with long axons that travel through the auditory nerve (also known as the vestibulocochlear nerve) to the cochlear nucleus in the brain stem. The rate of glutamate release is monotonically related to the membrane voltage. Thus, the more the stereocilia vibrate, the greater the current influx, the more depolarized the hair cell membrane voltage is, the greater the glutamate release, and, finally, the more spikes we expect from the auditory nerve fiber. The frequency tuning of auditory nerve fibers is shown in Figure 1.3 [16]. Each curve represents a single auditory nerve fiber, where the height corresponds to the volume of the sound stimulus required to record a threshold number of spikes. The stimulus is a pure sinusoid at the frequency indicated by the x axis value. The lowest point of the curve is known as the characteristic frequency (CF) of the auditory nerve fiber.

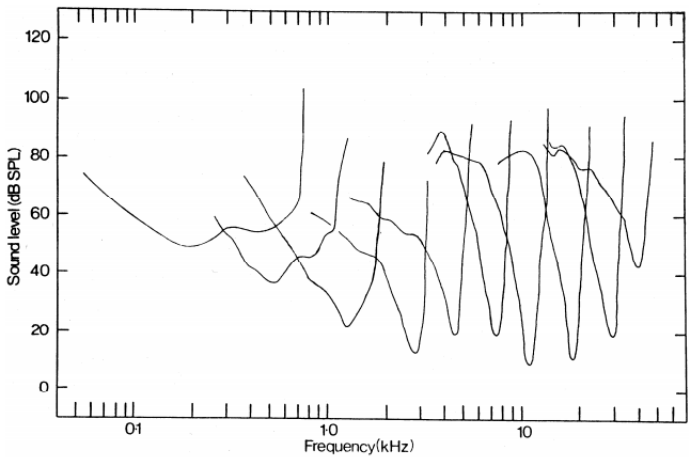

Figure 1.3: Frequency tuning curves of several auditory nerve fibers. Each point on a curve corresponds to the minimum sound level required to record a threshold number of spikes from the fiber. Adapted from [16]

Phase Coding

Because the glutamate release is correlated with the oscillations of inner hair cell membrane potential so too is the auditory nerve response. This phenomenon is known as phase locking. Figure 1.4 [56] indicates the degree to which an auditory nerve fiber is phase locked to the stimulus is negatively correlated with frequency of the stimulus.

Intensity Coding

While one might think that the intensity of the stimulus is simply conveyed by the firing rate of the auditory nerve fiber, the truth is more complicated. Each inner hair cell has between

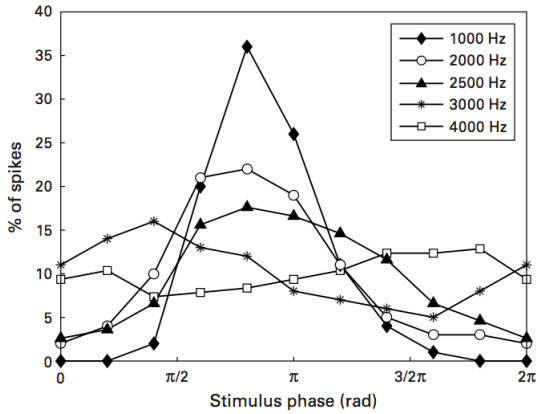

Figure 1.4: The distribution of spike timings across all phases for different frequencies. Notice that as the frequency increases the distribution becomes more uniform, indicating a lack of phase locking. Adapted from [57] with data from [56].

10-20 afferent nerve fibers attached to it. These fibers have a varying degree of spontaneous firing rate which is negatively correlated with the intensity required for it to fire above its spontaneous rate [42]. Thus, some fibers may not start spiking more until the sound has reached 30dB SPL and not saturate until 80dB SPL.

Dynamic Coding

An additional astonishing fact about the outer hair cells is that there are efferent fibers that synapse onto them, originating from the medial olivocochlear nucleus. This is the only sensory system in which feedback from the brain directly alters the sensory neurons response, e.g. unlike the retina. This mechanism is largely believed to be used to shift the compressive nonlinearity function of outer hair cells to match the intensity statistics of the current stimulus. Experimental results for this are shown in Figure 1.2.

Summary

In Chapter 2 these observations will be used to explore computational models of the early auditory system.

High-Level Auditory Pathway

One of the unique characteristics of the auditory pathway is the high prevalence of top-down feedback connections, including efferent connections from cortical layers to sub-thalamic layers and even efferent fibers entering the cochlea itself as shown in Figure 1.5. One hypothesis for what these connections are for is to modulate attention which will be explored from a computational perspective in Chapter 4.

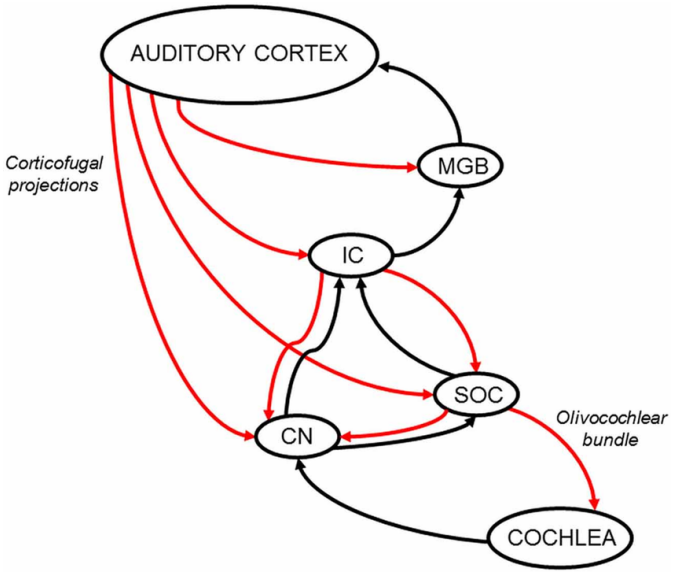

Figure 1.5: A high-level overview of efferent fibers (red) and afferent fibers (black) in the auditory pathway. Image from [62].

1.3 Time-Frequency Representations

A common preprocessing step for sound waveforms is to perform frequency analysis on consecutive segments of the waveform, similar to the process performed by the cochlea. There are in fact an infinite number of ways to do this but the common ones from the signal processing community are the wavelet transform and the Gabor transform. See Figure 1.6 for a visualization of possible transforms. While the cochlear transform is most similar to the wavelet transform, this transform presents difficulties for our neural network models because the time resolution changes per filter. This makes developing models which convolve over the frequency axis difficult, as discussed in Chapter 3. Instead we focus on a transform very similar to the Gabor transform, the Short-Time Fourier Transform (STFT).

Short-Time Fourier Transform (STFT)

The idea of the Short-Time Fourier Transform is to perform frequency analysis on consecutive segments of a long signal, in order to understand how frequencies evolve along a long time horizon as illustrated in Figure 1.7. The Short-Time Fourier Transform is specified by two scalar parameters, the hop size, H, and the window length, L, as well as a window function $w \in \mathbb{R}^L$.

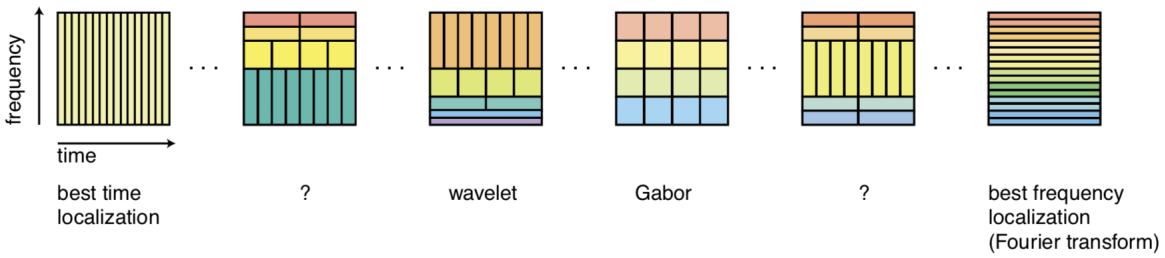

Figure 1.6: The time-frequency plane of different transforms. On the left the time domain is shown, where sound waveforms are recorded, the filters are localized in time and global in frequency. On the far right the opposite is shown, by performing the Fourier Transform, filters are localized in frequency and global in time. In the middle, other possible transforms are shown, including the wavelet and Gabor transform which are common in the signal processing community. Notice that each transform has the same number of filters, and the area of each filter is constant, regardless of transform. This reflects the time-frequency tradeoff. Figure image from [51].

$$s(\tau, t) = w[\tau] * x[t * H + \tau], \ \ \tau \in [0, L) \tag{1.1}$$

$$S(f, t) = \sum_{\tau=0}^{L-1} s(\tau, t) e^{-2\pi j \frac{\tau f}{L}} \ \ // \text{ Discrete-Time Fourier Transform} \tag{1.2}$$

where j is imaginary, and $S \in \mathbb{C}^{F,T}$ is the result of the STFT. The Short-Time Fourier Transform is equivalent to the Gabor transform if a gaussian window function is used.

In Figure 1.8 the spectrogram, which is the magnitude of the short-time fourier transform, of a woman saying "history of france" is shown to give more intuition for this transform.

1.4 Models for Auditory Scene Analysis

As mentioned earlier, one of the most impressive abilities of the human auditory system is the ability to separate sound mixtures into the separate sound events that make them. While one might compare to this to the ability to group adjacent retinal activations into a single object, the cochlear activations are far from adjacent as seen in Figure 1.8. This is because each sound source contains many frequencies across the frequency spectrum. So how does the brain sort out which frequencies belong to which source?

Experimental work into how complex sound mixtures are grouped into the individual sources was pioneered largely by Albert Bregman. In his book, *Auditory Scene Analysis* [8], Bregman describes how the statistical structure of natural sound can be used to group different frequencies into different sources. This statistical structure comes in the form of

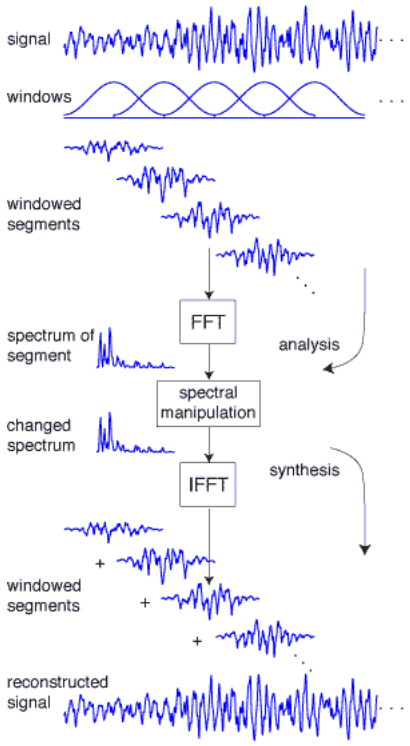

Figure 1.7: Diagram of the Short-Time Fourier Transform process as well as it's inverse. In this diagram $H = \frac{L}{2}$. The analysis consists of segmenting the waveform, multiplying by a window, and taking the fourier transform. The synthesis consists of the inverse operations. Figure image from [59].

common onset, co-modulation, and correlation among frequencies which correspond to a harmonic stack. While this methods can be used to explain how frequencies are grouped in the case of a few simple pure tones, how frequencies are grouped in complex, natural, environments is not well understood. This is because natural sounds in the world are not made up of a few simple pure tones, they experience many transformations, for example vocalizations are transformed by the mouth, and all sounds undergo idiosyncratic reverberations due to their acoustical environment.

In this thesis, *learning* is used to obtain a model which can capture the necessary statistical structure of natural sound. The learning algorithms here rely on neural networks and large amounts of data. Recently neural networks have been shown to be one of the most powerful machine learning frameworks for understanding the structure of high dimensional natural stimuli such as sounds and images [37]. They have been used for obtaining state of the art results in speech recognition [41], source separation [23], and much more. They are therefore excellent models for building computational models of auditory scene analysis.

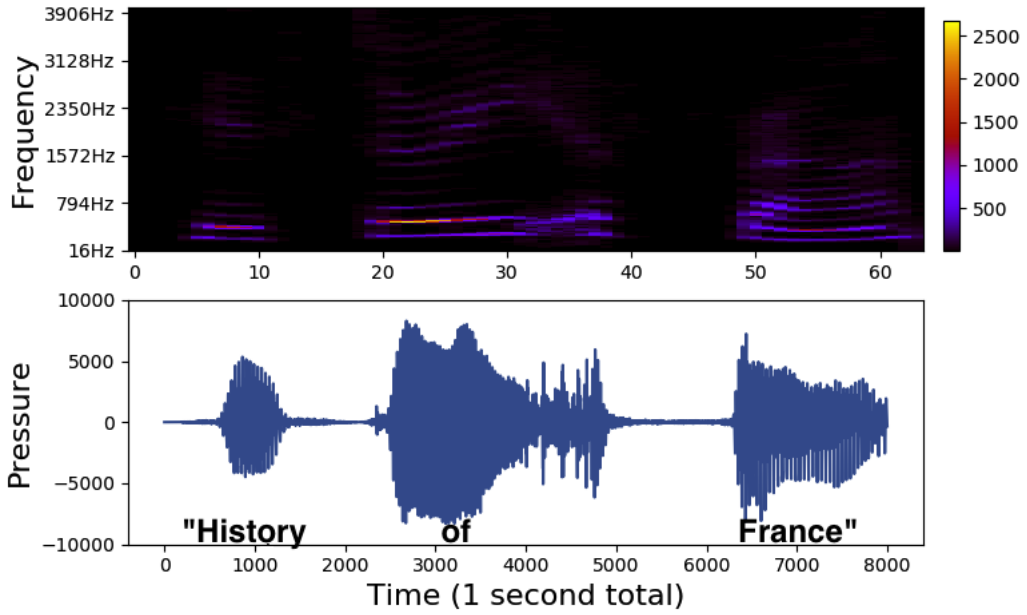

Figure 1.8: (Bottom) The waveform of a woman saying "History of France", 1 second in length at sample rate of 8000hZ. (Top) Spectrogram of the waveform computed using the absolute value of the short-time fourier transform of the waveform.

Neural Networks

A basic background of neural networks is provided here. The equations for a simple two layer neural network are:

$$y_i = \sigma(Wx_i) \tag{1.3}$$
$$z_i = \sigma(Vy_i) \tag{1.4}$$
$$\sigma(u) := \frac{1}{1+e^{-u}} \tag{1.5}$$

where $x_i \in \mathbb{R}^L$ is the input, $y_i \in \mathbb{R}^M$ is the hidden activation, and $z_i \in \mathbb{R}^N$ is the output. $W \in \mathbb{R}^{L,M}$ and $V \in \mathbb{R}^{M,N}$ are the parameters of our network. See Figure 1.9 for an illustration of this process. The parameters of this network must be learned given some objective, or loss function:

$$L = \sum_i^I [T_i - \sigma(V\sigma(Wx_i))]^2 \tag{1.6}$$

where i indexes over example data, I is the total number of examples, and T_i is the desired output of example i. Backpropagation, which is an application of the chain rule, is used to train our networks:

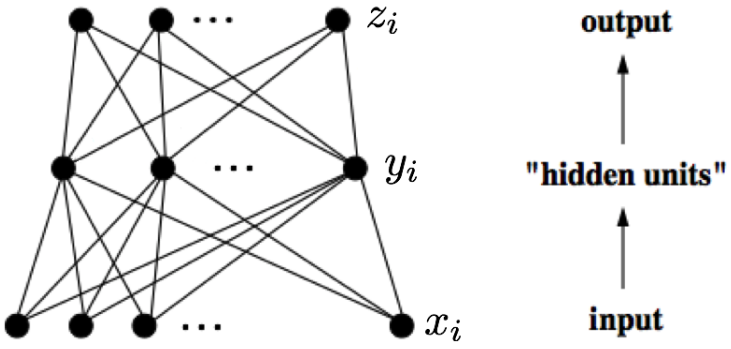

Figure 1.9: Diagram for a 2 layer neural network.

$$\theta := [W, V] \tag{1.7}$$

$$\theta = \theta - \eta * \frac{\partial L}{\partial \theta} \tag{1.8}$$

where η is a learning rate. This simple formula of neural network, loss function, and backpropagation can create very powerful models for studying the auditory system as we will see in the rest of this thesis.

1.5 Overview of Thesis

In Chapter 2 subcortical physiology is used to build a theoretical model that could explain the auditory nerve response. In Chapter 3 an audio source separation model is developed using the Short-Time Fourier Transform in combination with a neural network. In Chapter 4 an algorithm is developed for specifying target speaker extraction using an attentional context, which is a model of feedback sent from higher layers of the auditory pathway down to lower layers.

Chapter 2

Neural Representations of Sound

2.1 Abstract

The theoretical principles of efficient coding have been used to explain the center-surround and gabor receptive fields of retinal ganglion cells and neurons of the primary visual cortex. In this chapter similar ideas are used to explain not only the receptive field properties of the auditory nerve fibers, but their neural response themselves. The neural response has a number of interesting properties which have never been explained: highly distributed, a precise timing code, and an intensity code using multiple fibers. The seminal work of [61] provides the best correspondence between efficient coding and the gammatone receptive fields of the auditory system but does not explain the neural response. In this research I will outline a model to rectify these shortcomings. Our results indicate that with better optimization this method may bring insight into neural representation of sound.

2.2 Introduction

One of the goals of theoretical neuroscience is to provide concise theories for how neurons process information. By having a concise theory we can show that a large body of experimental results follows from a few intuitive ideas. Ideally, these theories are falsifiable, they can be implemented and their results can be compared with experimental data. This allows us to iteratively improve our theories in order to understand the nature of the brains computation. In my project I would like to build such a theory - I will build a theory of why encoding sound in the temporally precise firing patterns of the auditory nerve fiber is optimal from an efficient coding perspective.

I first explain the efficient coding principle and how it has been used to provide a concise theory of early visual processing. I then cover how these theories have been used to explain the early auditory system and reflect on their shortcomings. Next I provide an overview of the physiological results on the early auditory system to make these shortcomings more clear. I then outline two possible theoretical models of early auditory processing that will

start to resolve these differences. Finally, I propose a method to optimize these models using recent advances in Binary Neural Networks [10].

2.3 Coding

Efficient Coding

Animals have evolved their eyes in order to process images in a way that balances biological constraints with the optical performance that is required for their particular environment. This claim is supported by the rich and diverse set of eyes that have evolved across the animal kingdom which can be explained as an adaptation to specific environmental conditions [33]. Therefore, a theoretical explanation of an animals eyes can be provided through the principles of optics, biological constraints, and the environmental conditions. This idea applies to visual processing equally, a theoretical model of visual processing can be obtained using the natural visual environment, an animals biological constraints, and the principles of visual processing. This is then the principle of efficient coding - to represent sensory information accurately with the fewest physical and metabolic resources.

Redundancy Reduction / Data Compression

At its core efficient coding represents a trade-off, a need to preserve more information about the stimulus versus the metabolic cost of representing more information. In order to optimally satisfy this constraint, it is necessary to exploit the statistical structure contained in natural stimuli. Attneave [3] was the first to introduce this idea, that visual perception was related to the statistics of natural images. But it was Barlow [6] who eventually formalized these ideas into the principle of redundancy reduction which states that neurons should minimize their statistical dependences with one another. By minimizing this statistical dependence while encoding as much information as possible they formed an efficient code.

The theory of redundancy reduction was used by Atick and Redlich [2] to explain the center-surround properties of retinal ganglion cells (RGCs). Given the redundancy reduction hypothesis, they claimed that RGCs should have receptive fields which act to pairwise decorrelate their responses:

$$< R_i, R_j > \;\; = \;\; \delta(i-j) \;\; \forall i, j \tag{2.1}$$

Where R_i describes the response of RGC i and the average is, importantly, over samples of *natural images*. In the Fourier domain this means that the amplitude of their responses should be flat, or uniform, over frequency. Since it had been shown by Field [18] that natural scenes posses a $1/f^2$ power spectrum then the amplitude of the optimal receptive field in the Fourier domain should simply rise linearly with spatial frequency. Taking the inverse Fourier transform of this function results a center-surround receptive field. This result was

monumental - it was a powerful example of how the retina exploited the statistics of natural images in order to efficiently encode it.

Robust Coding

While redundancy reduction is an important principle in sensory coding, it is insufficient to completely explain how early sensory neurons encode information [5]. Neurons have a finite capacity which has been estimated to be around 1-2 bits per spike [7]. In order transmit information reliably with neurons that have a fixed capacity it is necessary to have some amount of redundancy, this is has been referred to by Doi et al. [13] as robust coding. Doi and Lewicki [12] created an auto-encoder model where the hidden units were put through a gaussian channel to approximate the limited capacity of neurons. By using an overcomplete representation they were able to overcome the channel noise to arbitrary precision given more units. By imposing an additional structural constraint on the filters, the optimal filters strongly matched the center-surround receptive field structure of regional cells. Thus, they were able to motivate the center-surround filters from a much more general framework. Later, Karklin and Simoncelli [28] used these ideas with a metabolic constraint on the neuron responses to show the optimality of center-surround receptive fields from a yet more general framework.

Sparse Coding

While redundancy reduction and robust coding are important principles for understanding how the brain encodes information efficiently, the brain must do more than this - it must construct a meaningful representations of sensory information. Horace Barlowe was famous for proposing one theory to build these representations, termed the "neuron doctrine of perception", which proposed that neurons exploit the statistics of natural stimuli in order to build sparse representations of it. More precisely, the sensory stimuli at any given moment should be described by a small fraction of neurons, forming a sparse representation. These ideas were formalized mathematically by Olshausen and Field [50] who formed a parameteric, linear, generative model for images with a sparse prior. When optimized the basis functions closely matched the oriented receptive fields of V1 neurons, i.e. gabor functions. This was another powerful result in the field of theoretical neuroscience, the theory of sparse coding in the cortex had been used to explain the oriented receptive fields of V1 neurons.

Auditory Models

Given the success of sparse coding in the visual domain it was natural to wonder if it could explain the receptive field properties of the auditory nerve fibers. Lewicki [39] used a related model, Independent Component Analysis (ICA), on segments of sound and found that the learned filters closely resembled the gammatone filters often used to describe the receptive

field of the auditory nerve fibers. The next breakthrough came from Smith and Lewicki [61] where a convolutional sparse coding model was used to form a generative model of sound:

$$\hat{x}(t) = \sum_i b_i(t) \star S_i(t)$$
$$E = \frac{1}{2}||x(t) - \hat{x}(t)||^2$$

where $b_i(t)$ represents a scalar coefficient at time t, $\star$ indicates convolution, and S_i is a temporal kernel.

The optimization of this model consisted of iterating on the following two steps:

- Matching Pursuit - Find a sparse number of coefficients that best reconstruct the signal
- A gradient update of the kernel using these coefficients: $S_i = S_i + \eta \frac{\partial E}{\partial S_i}$

The Matching Pursuit [46] algorithm itself consists of the following steps:

- Convolve the input signal with each basis function to get potential coefficent responses for the signal.
- Take the maximal response over coefficients and time. If this is less than a pre-defined threshold then stop. If it is above, then remove this component from the input signal and iterate

In this way a sparse number of coefficients is found to represent the entire signal x. In the optimization of this network the kernels converge to the revcor (reverse correlation) filters of the auditory nerve, i.e gammatone filters (Figure 2.1). While the training of this model resulted in kernels which closely match the revcor filters of the auditory system, the coefficients strongly differ from the audtiory nerve **response** in a number of ways.

The first difference is how the coefficients are computed, i.e. through matching pursuit. As reviewed in the introduction, the auditory nerve response is largely a passive response to the input, there is no competition or lateral inhibition between the neurons in the cochlea. In contrast, the matching pursuit algorithm is implicitly a competition between units trying to describe the signal with only one winner in each iteration. This represents an active, inference procedure which is not present in the anatomy. A much more reasonable way to describe the auditory nerve response is as a simple function of the input:

$$b_i(t) = x(t) \star A_i(t) \tag{2.2}$$

This decision has important consequences in the output representation of the network. As can be seen in Figure 2.2 the coefficient representation of the convolutional sparse coding

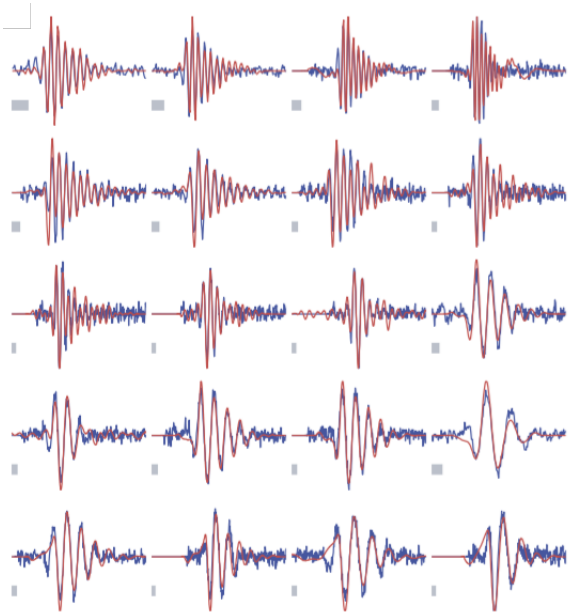

Figure 2.1: The kernel functions resemble gammatone filters (red). Matching teverse Correlation (revcor) filters found from the auditory nerve response of chinchillas [55] are shown in blue. From Smith and Lewicki [61].

model is quite sparse. However, as shown below, experimental recordings of the auditory nerve response are much less sparse, there is a distributed response to input stimuli.

Reflecting on equation (2.2), this is the result of two important properties. The first is that auditory nerve fibers have overlapping receptive fields and the second is that there is no competition or lateral inhibition which would sparsify the solution as in Matching Pursuit. These two properties explain the difference between the theoretical results and the physiological results.

Because of this distributed response sparse coding is an insufficient theory for explaining the auditory nerve response. So what is a sufficient theory? Reflecting on how theory has explained the processing of the retina, the robust coding framework was the most effective. In the robust coding framework the auditory nerve is a group of fixed capacity fibers which are trying to maximize the mutual information between their representation and the input waveform subject to some metabolic constraints. This idea forms the foundation of our theoretical model. To use a set of fixed capacity neurons which attempt to represent the input stimulus as efficiently as possible. The central goal of our model is obtain a distributed spike representation of the input signal which has a strong correspondence to the experimental data.

There are some additional shortcomings of the Smith and Lewicki [61] paper:

- The auditory nerve response consists of binary valued spikes while [61] uses analog spikes

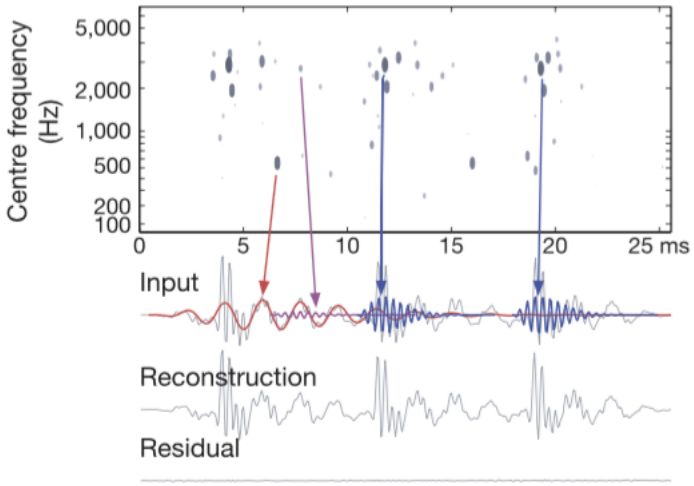

Figure 2.2: A spectrogram-like representation of the input stimulus using the optimal kernels and coefficients. The kernels are ordered vertically by their center frequency. Colors simply denote different kernels in this diagram. From [61].

- The auditory nerve response is known to have an important phase structure, where the timing of the spikes is strongly correlated with the peak of a periodic input. The model in [61] does not replicate this phase structure.

These differences are important because they elucidate why limiting the capacity of neurons is required for understanding the representation used. The auditory nerve response is capacity limited because it can only communicate with binary spikes, further, it is limited in the temporal domain by a refractory period after spiking. We believe that these capacity limitations combined with the efficient coding principle will lead the distributed, phase, and intensity structure to be an emergent property of the system. Using wavelet filters we have constructed an artificial distributed response in Figure 2.3, which represents the desired output of our model.

2.4 Models

Before outlining our robust coding models we review some preliminary work replicating the results of [39] using an auto-encoder framework with a sparse prior in the hidden layer:

$$\begin{aligned} h &= Ax \\ \hat{x} &= Sh \\ E &= \frac{1}{2}||x - \hat{x}||^2 + \lambda|x|_1 \end{aligned}$$

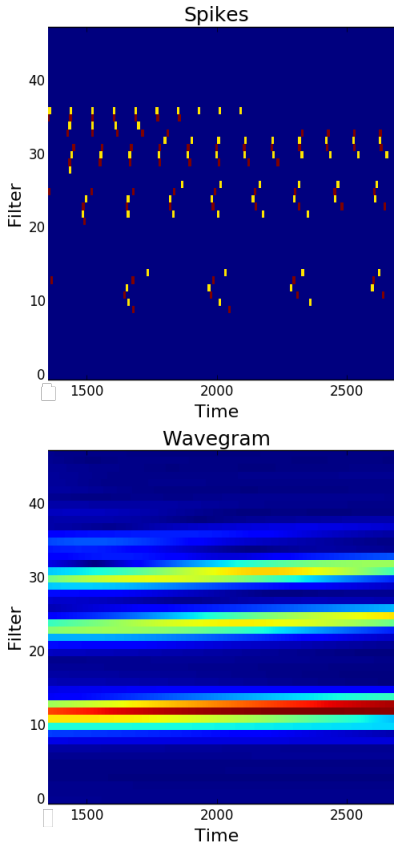

Figure 2.3: Bottom - A spectrogram-like representation of speech from the GRID Speech Corpus using wavelets (similar to gammatone filters) with overlapping receptive fields. Top - An artificial spike based representation using these filters. Note that the sound stimulus is represented in a highly distributed way, there are 5 artificial fibers firing in response to the fundamental frequency contained in the stimulus.

This model closely resembles the work of [34] but we not impose the constraint $S = A^T$. Our results in Figure 2.4 reveal filters which closely resemble the band-pass gamma tone filters found in [39].

This result underscores the ability of the auto encoder framework to recover structure found in the auditory system. We now turn to our two robust coding models for explaining the auditory nerve response. Our code is available at https://github.com/ShariqM/rica_sound.

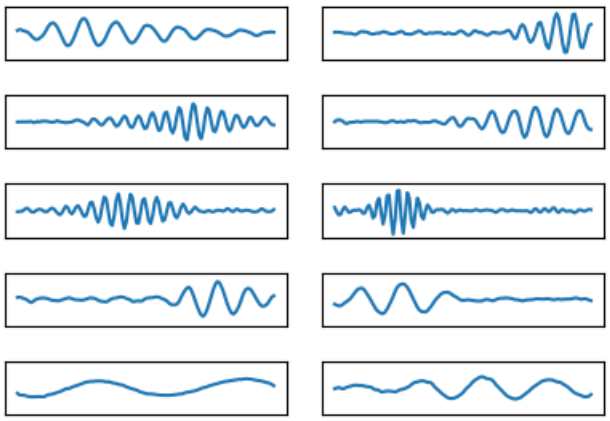

Figure 2.4: Subset of basis functions learned in Model 1. All basis functions can be viewed here: http://i.imgur.com/uUyBgV7.png

Model 1 - Convolutional AutoEncoder

The first model is a convolutional auto encoder where the hidden layer is subjected to noise in order to limit the capacity of neurons as in [13]:

$$\begin{aligned} b_i(t) &= x(t) \star A_i(t) + n_i(t) \\ \hat{x}(t) &= \sum_i b_i(t) \star S_i(t) \\ E &= \frac{1}{2} \sum_t ||x(t) - \hat{x}(t)||^2 + \lambda |b(t)|_1 \\ n_i(t) &\overset{iid}{\sim} \mathcal{N}(0, \sigma_n^2) \end{aligned}$$

We additionally impose a metabolic cost on b, instead of the structural constraint on A, as in [12]. A similar constraint was used in [28]. The goal of this model is to show gamma tone filters can emerge from a simple capacity constraint rather than a sparsity constraint.

Model 2 - Convolutional LIF AutoEncoder

While the previous model would show how the gammatone filters can arise from a fixed capacity constraint the noise model is not realistic. The reason the auditory nerve response is limited in capacity is because it communicates using binary spikes at intervals of at least $\sim$ 1ms due to the refractory period. Thus, the real question we want to answer is - What is the best way to encode the time-varying signal of sound into a spike-based code using only 30,000 nerve fibers, as in the human auditory system. To answer this question we limit the capacity of the hidden layer by using a leaky-integrate-and-fire (LIF) [15] model

which outputs binary spikes. By additionally imposing a refractory period we could limit the temporal precision. A diagram of our model is shown in Figure 2.5.

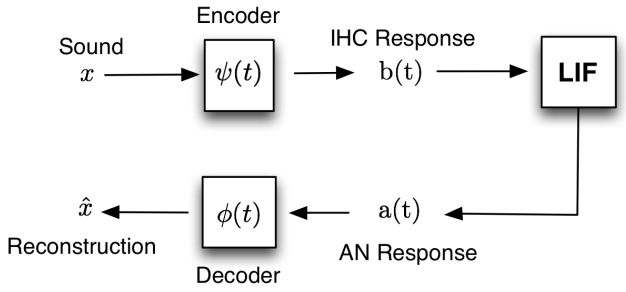

Figure 2.5: Diagram of the model. Inner Hair Cell (IHC). Auditory Nerve (AN).

Encoder:

$$b_i(t) = x(t) \star A_i(t)$$

LIF:

$$\begin{aligned} \bar{v}_i(t) &= b_i(t) + e^{-1/\tau} v_i(t-1) \\ v_i(t) &= \begin{cases} 0, & \bar{v}_i(t) < 0 \\ \bar{v}_i(t), & 0 \leq \bar{v}_i(t) < \theta \\ 0, & \bar{v}_i(t) \geq \theta \end{cases} \\ a_i(t) &= \begin{cases} 0, & \bar{v}_i(t) < \theta \\ 1, & \bar{v}_i(t) \geq \theta \end{cases} \end{aligned}$$

Decoder:

$$\hat{x}(t) = \sum_i a_i(t) \star S_i(t)$$

Objective:

$$E = \frac{1}{2} \sum_t ||x(t) - \hat{x}(t)||^2 + \lambda \sum_i a_i(t)$$

where θ denotes the threshold, v denotes the membrane potential, and a denotes the spiking auditory nerve response. After optimization we would like to see that the analysis filters converge to the gamma tone filters. But more importantly, we would like the response, a, to strongly resemble the auditory nerve response. This would prove that the distributed representation is an emergent property of auditory nerve response that follows from an energy efficient robust coding principle. Further, we would hope that the phase and intensity coding properties are also emergent properties.

We next propose a method for training this model.

2.5 Training

Training a spike-based neural network has been challenging in the field [38, 49]. The main difficulty arises from the gradients of the spike activation and voltage reset functions being zero or even non-differentiable at certain locations. However, recently, there has been promising work in the area of binary neural networks [10]. These methods replace the true gradients with a smooth, non-zero gradient approximation in order to combat the aforementioned issues. Here, we apply those ideas to the LIF model to make our network trainable. For the spike generation function we use a sigmoid activation centered at the threshold instead of the true function:

$$\begin{aligned} \overleftarrow{a_i}(t) &= \sigma(\bar{v}_i(t) - \theta) = \frac{1}{1 + e^{-(\bar{v}_i - \theta))}} \\ \frac{\partial a_i(t)}{\partial \bar{v}_i(t)} &:= \frac{\partial \overleftarrow{a}_i(t)}{\partial \bar{v}_i(t)} \end{aligned}$$

where $\leftarrow$ denotes how the function behaves in the backwards pass in backpropagation. This is displayed visually in Figure 2.6.

For the voltage reset function we make a similar approximation using an unnormalized gaussian:

$$\begin{aligned} \overleftarrow{v}_i(t) &= \theta \, exp(-\frac{(x-\theta)^2}{2\theta}) \\ \frac{\partial v_i(t)}{\partial \bar{v}_i(t)} &:= \frac{\partial \overleftarrow{v}_i(t)}{\partial \bar{v}_i(t)} \end{aligned}$$

This is displayed visually in Figure 2.7.

2.6 Results

Unfortunately even with our smooth gradient optimization tricks, optimization turned out to be extremely brittle. We show the results of one relatively successful experiment in Figure 2.8. The code for our study is available at https://github.com/ShariqM/soundAE.

2.7 Conclusion

Can the auditory nerve response be explained from the principle of efficient coding? Inspired by [13], we have claimed that the properties of the auditory nerve response is a simple consequence of encoding the input waveform with maximum fidelity subject to capacity and metabolic constraints. This capacity constraint is enforced by using LIF neurons with a refractory period which restricts the representation to be a binary, spike-based code. Finally

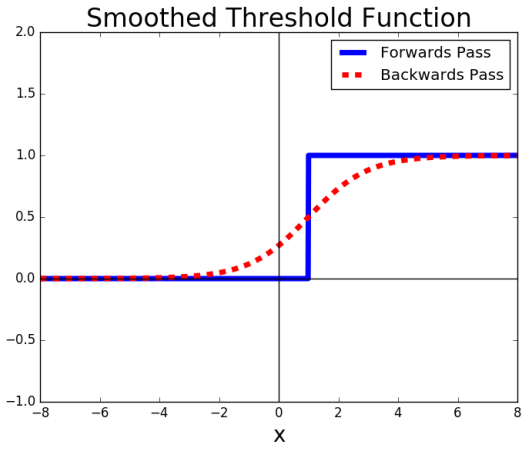

Figure 2.6: A smoothed version of the spike activation function that allows for easier gradient based optimization

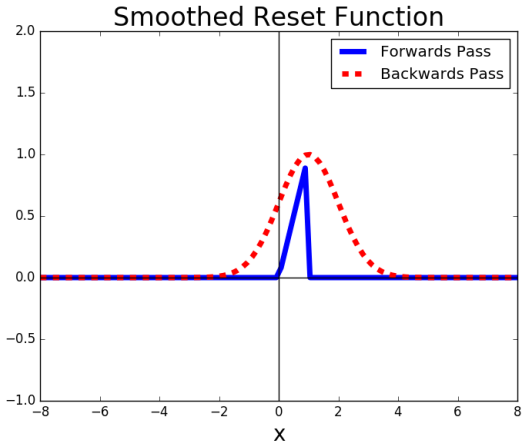

Figure 2.7: A smoothed version of the voltage reset function that allows for easier gradient based optimization

the metabolic constraint is enforced through penalizing the number of spikes used to represent the waveform.

Our model consists of a convolutional auto encoder model where the hidden layer is put through a LIF model. While training this model using gradient based techniques has been challenging recent advances in binary neural networks bring hope to the feasibility of training this model. Unfortunately our experiments were largely unsuccessful so far. Perhaps with additional modifications to the optimization method this model could prove fruitful.

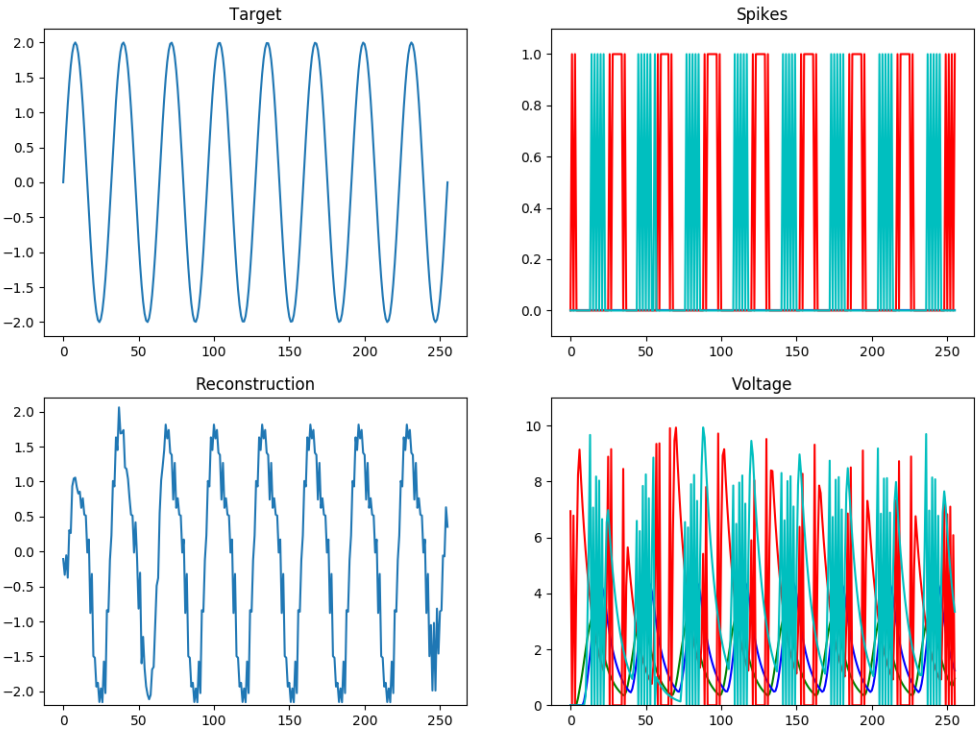

Figure 2.8: Computational experimental results with our model. Time is on the x-axis. This model has four neurons which correspond to the colors in the right column. The reconstruction of the target input is shown in the left column.

Chapter 3

Robust Audio Source Separation

3.1 Abstract

In this chapter a cochlea-like representation, the spectrogram, is used to study audio source separation, which is a major component of auditory scene analysis. This is joint work with my colleague Brian Cheung. Recent work has shown that recurrent neural networks can be trained to separate individual speakers in a sound mixture with high fidelity. Here we explore convolutional neural network models as an alternative and show that they achieve competitive results with an order of magnitude fewer parameters on standard benchmarks. We extend these benchmarks and compare the robustness of these different approaches to generalize under three new test conditions: longer time sequences, intermittent noise, and datasets outside the training domain. For the last condition, a new dataset is created, *RealTalkLibri*, to test source separation in real-world environments. This is used to demonstrate that the acoustics of the environment have significant impact on the overall performance of neural network models, though our convolutional model shows superior ability to generalize to new environments. The code for our study is available at https://github.com/ShariqM/source_separation.

3.2 Introduction

The sound waveform that arrives at our ears rarely comes from a single isolated source, but rather contains a complex mixture of multiple sound sources transformed in different ways by the acoustics of the environment. One of the central challenges of auditory scene analysis is to separate the components of this mixture so that individual sources may be recognized. Doing so generally requires some form of prior knowledge about the statistical structure of sound sources, such as common onset, co-modulation and correlation among harmonic components [8, 11]. Our goal is to develop a model that can learn to exploit these forms of structure in the signal in order to robustly segment the time-frequency representation of a sound waveform into its constituent sources (see Figure 3.1).

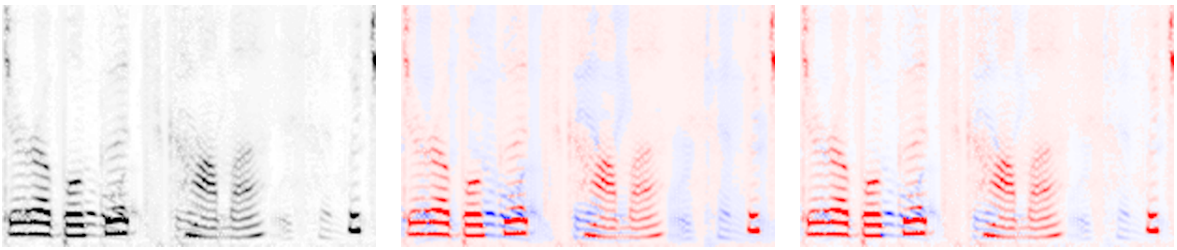

Figure 3.1: Example of source separation using the spectrogram of two overlapped voices as input. *Left*: Spectrogram of the mixture. *Middle*: Ground truth (oracle) source estimates (red and blue). *Right*: Source estimates using our convolutional model.

The problem of source separation has traditionally been approached within the framework of *computational auditory scene analysis* (CASA) [25]. These methods typically rely upon features such as gammatone filters in order to find a representation of the data that will allow for clustering methods to segment the individual speakers of the mixture. Recently, models which make these features learnable have been quite successful. Hershey et al. [23] introduced Deep Clustering (DPCL) which uses a Bi-directional Long short-term memory (BLSTM) [21] neural network to learn useful embeddings of time-frequency bins of a mixture. They formulate an objective function which encourages these embeddings to cluster according to their source so that K-means can be applied to partition the source signals in the mixture. This model was further improved in the work of Isik et al. [26] and Chen et al. [9] which proposed simpler end-to-end models and achieved an impressive ∼10.5dB Signal-to-Distortion Ratio (SDR) in the source estimation signals.

In this work we develop an alternative model for source separation based on a dilated convolutional neural network architecture [73]. We show that it achieves state-of-the-art performance with an order of magnitude fewer parameters and also generalizes better under novel conditions and datasets. In addition, our convolutional approach only requires a small and fixed amount of future information and can therefore operate in real-time.

An under- investigated aspect of deep learning algorithms is their ability to generalize to new datasets. Models are usually trained, tested and compared against on the same data distribution. In vision datasets, categories and tasks vary widely. In contrast, an auditory dataset provides a unique opportunity fix the task, i.e. to extract speech content, while varying the environment in which the data is recorded. In order to study the generalization capabilities of the models, we test them with inputs containing very long time sequences, intermittent noise, and mixtures collected under different recording conditions as in our *RealTalkLibri* dataset.

Success in these more challenging domains is critical for progress to continue in robust source separation, where the eventual goal is to be able to separate sources regardless of speaker identities, recording devices, and acoustical environments. Figure 3.2 shows three examples of how these factors can affect the spectrogram of the recorded waveform. Training models which can generalize to the space of these deformations is challenging. In vision and machine learning, this issue is usually referred to as *dataset bias* [64, 65, 14] where models

perform well on the training dataset but fail to generalize to new datasets. In recent years the main approach to tackling this issue has been through data augmentation. In the speech community, simulators for different acoustical environments [4, 31] have been leveraged to create more data. Recently, Variational Auto-Encoders [30] have begun to be used [24]. In vision and machine learning, Adversarial Network frameworks [20] have been used to improve generalization to different domains [65].

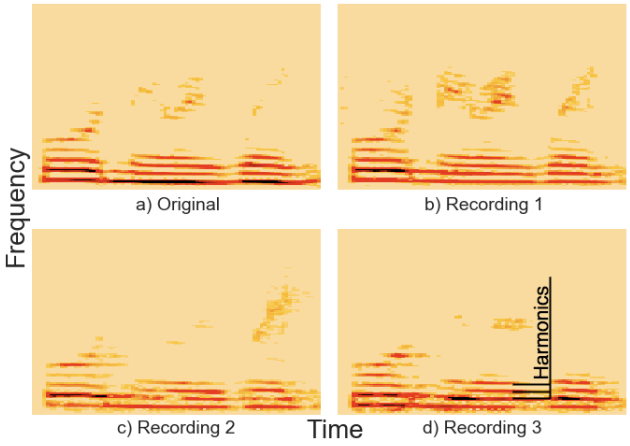

Figure 3.2: *a*: Original recording of a single female speaker from the LibriSpeech dataset; *b,c,d*: Recordings of the original waveform made with three different orientations between computer speaker and recording device.

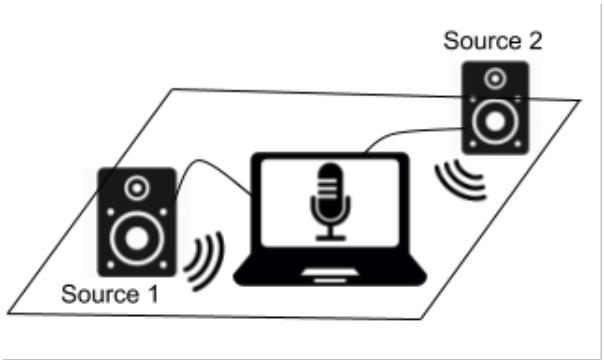

Figure 3.3: Recording setup diagram. The speakers were separated from the recording device by different devices, each computer speaker was responsible for playing a particular source.

Here we show that the choice of model architecture alone can improve generalization. Our choice to use a convolutional architecture was inspired by the generalization power of Convolutional Neural Networks (CNNs) [36, 32, 60] relative to fully connected networks. We compare the performance of our CNN model with the recurrent BLSTM models of previous work and show that while both suffer when tested on new datasets the CNN model exhibits superior performance to the BLSTM.

3.3 Deep Attractor Framework

Notation: For a tensor $T \in \mathbf{R}^{A \times B \times C}$: $T_{\cdot,\cdot,c} \in \mathbf{R}^{A \times B}$ is a matrix, and $T_{a,\cdot,c} \in \mathbf{R}^{B}$ is a vector, and $T_{a,b,c} \in \mathbf{R}$ is a scalar.

Embedding the mixed waveform

Chen et al. [9] propose a framework for single-channel speech separation. $x \in \mathbf{R}^{\tau}$ is a raw input signal of length τ and $X \in \mathbf{R}^{F \times T}$ is its spectrogram computed using the Short-

time Fourier transform (STFT). Each time-frequency bin in the spectrogram is embedded into a K-dimensional latent space $V \in \mathbf{R}^{F \times T \times K}$ by a learnable transformation $f(\cdot;\theta)$ with parameters θ:

$$\bar{V} = f(X;\theta) \tag{3.1}$$

$$V_{f,t,\cdot} = \frac{\bar{V}_{f,t,\cdot}}{||\bar{V}_{f,t,\cdot}||_2} \tag{3.2}$$

In our work, the embeddings are normalized to the unit sphere in the latent dimension k (eq. 3.2).

Generating embedding labels

Assuming that each time-frequency bin can be assigned to one of the C possible speakers, the Ideal Binary Mask (IBM), $Y \in \{\mathbf{0}, \mathbf{1}\}^{F \times T \times C}$, is a one-hot representation of this classification for each time-frequency bin:

$$\bar{Y}_{f,t,c} = \begin{cases} 1, & \text{if } c = \arg\max_{c'}(S_{f,t,c'}) \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

$$\tag{3.4}$$

where $S \in \mathbf{R}^{F \times T \times C}$ is the supervised source target spectrogram.

Threshold Preprocessing for Training

To prevent time-frequency embeddings with negligible power from interfering, the raw classification tensor $\bar{Y}$ is first masked with a threshold tensor $H \in \mathbf{R}^{F \times T}$. The threshold tensor removes time-frequency bins which are below a fraction $0 < \alpha < 1$ of the highest power bin present in X:

$$H_{f,t} = \begin{cases} 0, & \text{if } X_{f,t} < \alpha \max(X) \\ 1, & \text{otherwise} \end{cases} \tag{3.5}$$

$$Y_{\cdot,\cdot,c} = \bar{Y}_{\cdot,\cdot,c} \odot H \tag{3.6}$$

where $\odot$ denotes element-wise product.

Clustering the embedding

An attractor point, $A_c \in \mathbf{R}^K$, can be thought of as a cluster center for a corresponding source c. Each attractor A_c is the mean of all the embeddings which belong to speaker c:

$$A_{c,k} = \frac{\sum_{f,t} V_{f,t,k} Y_{f,t,c}}{\sum_{f,t} Y_{f,t,c}} \tag{3.7}$$

During training the attractor points are calculated using the thresholded oracle mask, Y. In the absence of the oracle mask at test time, the attractor points are calculated using K-means. Only the embeddings which pass the corresponding time-frequency bin threshold are clustered.

Finally the mask is computed by taking the inner product of all embeddings with all attractors and applying a softmax:

$$M_{f,t,c} = \underset{c}{softmax}\Big(\sum_{k} A_{c,k} V_{f,t,k}\Big) \tag{3.8}$$

$$\hat{S}_{\cdot,\cdot,c} = M_{\cdot,\cdot,c} \odot X \tag{3.9}$$

From this mask, we can compute source estimate spectrograms (3.9) which in turn can be converted back to an audio waveform via the inverse STFT. We do not attempt to compute the phase of the source estimates. Instead, we use the phase of the mixture to compute the inverse STFT with the magnitude source estimate spectrogram $\hat{S}$.

The loss function $\mathcal{L}$ is the mean-squared-error (MSE) of the source estimate spectrogram and the supervised source target spectrogram, $S \in \mathbf{R}^{F \times T \times C}$:

$$\mathcal{L} = \sum_{c} ||S_{\cdot,\cdot,c} - \hat{S}_{\cdot,\cdot,c}||_F^2 \tag{3.10}$$

where $||\cdot||_F$ denotes the Frobenius norm. See Figure 3.4 for an overview of our source separation process.

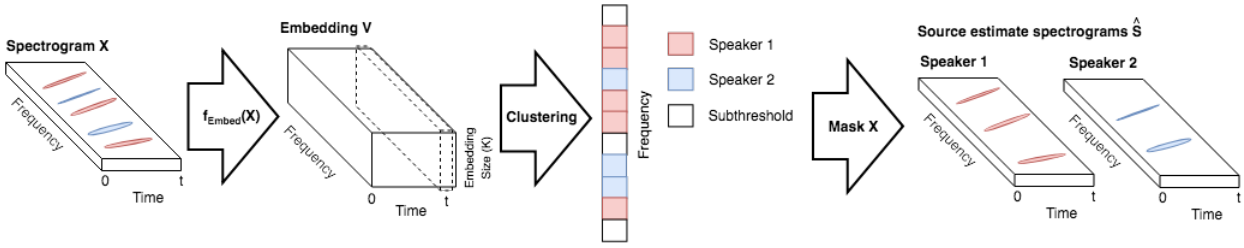

Figure 3.4: Overview of the source separation process.

Network Architecture

A variety of neural network architectures are candidates to parameterize the embedding function in Equation 3.1. Chen et al. [9] use a 4-layer Bi-Directional LSTM architecture which utilizes weight sharing across time, allowing it to process inputs of variable temporal length.

By contrast, convolutional neural networks are capable of sharing weights along both the time and frequency axis. Recently convolutional neural networks have been shown to

perform state-of-the-art music transcription by having filters which convolve over both the frequency and time dimensions [60]. One reason this may be advantageous is that the harmonic series exhibits a stationarity property in the frequency dimension. Specifically, for a signal with fundamental frequency f_0 the harmonics are equally spaced according to the following set: $\{i * f_0 : i = 2, 3, ..., n\}$. This structure can be seen in the equal spacing of successive harmonics in Figure 3.2d.

Another motivation we have for using convolutional neural networks is that they do not incorporate feedback which may allow them to be more stable under novel conditions not seen during training. In the absence of a recurrent memory, filter dilation [73] enables the receptive field to integrate information over long time sequences without the loss of resolution. Furthermore, incorporating a fixed amount of future knowledge in the network is straightforward by having a fixed-lag delay in the convolution as we show in Figure 3.5.

3.4 Model

Dilated Convolution

Yu and Koltun [73] proposed a dilation function $D(\cdot, \cdot, \cdot; \cdot)$ to replace the pooling operations in vision tasks. For notational simplicity, we describe dilation in one dimension. This method convolves an input signal $X \in \mathbf{R}^G$ with a filter $K \in \mathbf{R}^H$ with a dilation factor d:

$$F_t = D(K, X, t; d) = \sum_{dh+g=t} K_h X_g$$

The input receptive field of a unit F_t in an upper layer of a dilated convolutional network grows exponentially as a function of the layer number as shown in Figure 3.5. When applied to time sequences, this has the useful property of encoding long range time dependencies in a hierarchical manner without the loss of resolution which occurs when using pooling. Unlike recurrent networks which must store time dependencies of all scales in a single memory vector, dilated convolutions stores these dependencies in a distributed manner according to the unit and layer in the hierarchy. Lower layers encode local dependencies while higher layers encode longer range global dependencies.

3.5 Datasets

We construct our mixture data sets according to the procedure introduced in [23], which is generated by summing two randomly selected waveforms from different speakers at signal-to-noise ratios (SNR) uniformly distributed from -5dB to 5dB and downsampled to 8kHz to reduce computational cost.

A training set is constructed using speakers from the Wall Street Journal (WSJ0) training dataset [19] si_tr_s.

We construct three test sets:

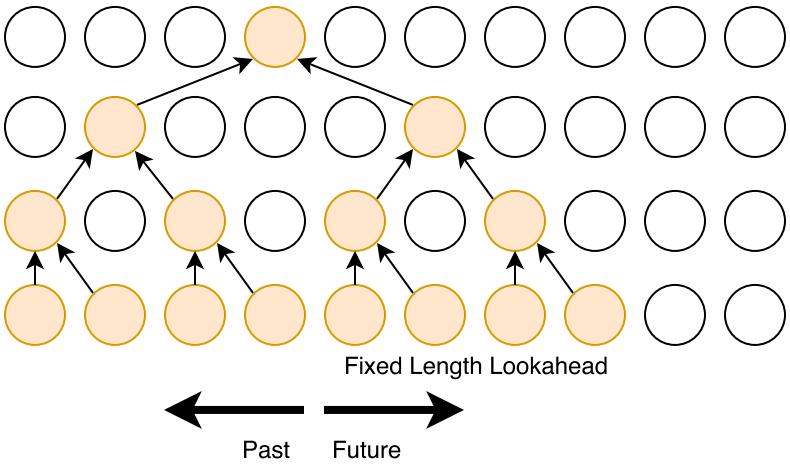

Figure 3.5: One-dimensional, fixed-lag dilated convolutions used by our model with dilation factors of 1, 2 and 4. The bottom row represents the input and each successive row is another layer of convolution. This network has a fixed-lag of 4 timepoints before it can output a decision for the current input.

- *WSJ0*: a test set is constructed identical to the test set introduced in [23] using 18 unheard speakers from si_dt_05 and si_et_05.
- *LibriSpeech*: a test set is constructed using 40 unheard speakers from test-clean.
- *RealTalkLibri*: we generate more realistic mixture using the procedure described below.

RealTalkLibri Dataset

We are interested in collecting a speech mixture dataset with two properties:

1. Speech mixtures are deformed by the acoustics of the room.
2. Availability of ground truth for the source waveforms.

Unfortunately all audio datasets that have property (1) don't have property (2) [31, 4]. These datasets only provide the transcription. In order to fill this void and investigate generalization we created a small test dataset with both these properties.

The *RealTalkLibri* (RTL) test dataset is created starting from the test-clean directory of the open LibriSpeech dataset [53] which contains clean speech from 40 speakers. We first downsampled all waveforms to 8kHz as before. Each mixture in the dataset is created by sampling two random speakers from the test-clean partition of LibriSpeech, picking a random waveform and start time for each, and playing the waveforms through two Logitech computer speakers for 12 seconds. The waveforms of the two speakers are played in separate channels linked to a left and right computer speaker, separated from the microphone of the

computer by different distances. The recordings are made with a sample rate of 8kHz using a 2013 MacBook Pro. Figure 3.3 is a diagram of our data collection process.

In order to obtain ground truth of the individual speaker waveforms each of the waveforms is played twice, once in isolation and once simultaneously with the other speaker. The first recording represents the ground truth and the second one is for the mixture. To verify the quality of the ground truth recordings, we constructed an ideal binary mask Y which performs about as well on the previous simulated datasets, see the Oracle performance in Figure 3.9. The *RealTalkLibri* data set is made up of two recording sessions which each yielded 4.5 hours of data, giving us a total of 9 hours of test data. The dataset is made freely available at https://www.dropbox.com/s/4pscejhkqdr8xrk/rtl.tar.gz?dl=0.

3.6 Experiments

Experimental Setup

We evaluate the models on a single-channel simultaneous speech separation task. The mixture waveforms are transformed into a time-frequency representation using the Short-time Fourier Transform (STFT) and the log-magnitude features, X, are served as input to the model. The STFT is computed with 32ms window length, 8ms hop size, and the Hann window. We use SciPy to compute the STFT and TensorFlow to build our neural networks.

We report our results using the signal-to-distortion ratio (SDR) metric introduced in [67] as a blind audio source separation (BSS) metric. We compute our results using version 3 of the Matlab bsseval toolbox [17]. A python implementation of this code is also available online [54] [1].

Our network consists of 13 dilated convolutional layers [73] made up of two stacks, each stack having its dilation factor double each layer. Batch Normalization is applied to each layer and residual connections [22] are used at every other layer, see Table 3.1 for details.

Layer	1	2	3	4	5	6	7	8	9	10	11	12	13
Convolution	3x3	3x3	3x3	3x3	3x3	3x3	3x3	3x3	3x3	3x3	3x3	3x3	3x3
Dilation	1x1	2x2	4x4	8x8	16x16	32x32	1x1	2x2	4x4	8x8	16x16	32x32	1x1
Residual	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No
Channels	128	128	128	128	128	128	128	128	128	128	128	128	K

Table 3.1: Dilated Convolutional Network Architecture

We reimplement the DANet of [9] with a BLSTM architecture containing 4 layers and 500 hidden units in both the forward and backward LSTM, for a total of 1000 hidden units.

[1] `https://github.com/craffel/mir_eval/blob/master/mir_eval/separation.py`

Model	WSJ0 SDR (dB)	Number of Parameters
DANet (BLSTM)	10.48	17 114 580
DPCL* (BLSTM)	10.8	?
Ours (CNN)	**10.97**	1 650 836
Oracle	13.49	-

Table 3.2: SDR for two competitor models, our proposed convolutional model, and the Oracle. Our model achieves the best results using significantly fewer parameters. Scores averaged over 3200 examples. *: SDR score is from [26] which is at an advantage due to a second enhancement neural network after masking.

We replicated their training schedule using the RMSProp algorithm [63], a starting learning rate of 1e−3, and exponential decay with parameters: decay_steps= 2000, decay_rate= 0.95.

For evaluation we also use the max() function rather than the softmax() for computing the mask in equation (3.8). We calculate an Oracle score using the Ideal Binary Mask (IBM), Y, using the ground truth source spectrograms (Eq. 3.4).

WSJ0 Evaluation

We begin by evaluating the models on the WSJ0 test dataset as in [9]. Our results are shown in Table 3.2. Our model achieves the best score using a factor of ten fewer parameters than DANet. The DPCL score is taken from [26] which has a very similar architecture to DANet and therefore a similar number of parameters. Their model has one important difference however, a second neural network is used to enhance the source estimate spectrogram to achieve their result. Our model is still able to exceed its performance without this extra enhancement network. In addition, our model has a fixed window into the future whereas the BLSTM models have access to the entire future. This indicates that a convolution based architecture is better at solving this source separation task with less information in comparison to a recurrent based architecture.

Embeddings

At test time the K-means must be employed because the labels Y are not available for calculating the attractors (3.7). In order for the attractors to from a good proxy for the K-means algorithm we applied l2 normalization (Eq. 3.2) to the embeddings V. In Figure 3.6 we visualize the embedding outputs of our model using PCA for a single mixture across $T = 200$ timepoints. Each embedding point corresponds to a single time-frequency bin in the mixed input spectrogram. The embeddings are colored in this diagram according to the oracle labelling, red for speaker 1 and blue for speaker 2. Notice that the network has

learned to cluster the embeddings according to the speaker they belong too, i.e. there is a high density of red embeddings on the left and similarly for blue embeddings on the right. This structure allows K-means to easily find cluster centers that match the attractors used at training time.

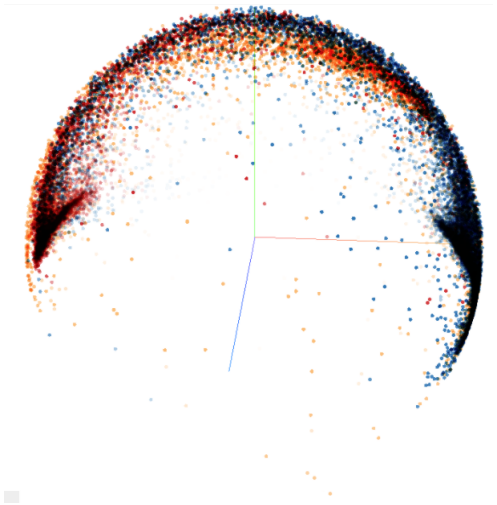

Figure 3.6: Embeddings for two speakers (red & blue) projected onto a 3-dimensional subspace using PCA. The orange points correspond to time-frequency bins where the energy was below threshold (see 3.5). There are $T \times F = 200 \times 129 = 25800$ embedding points in total.

Generalization Experiments

Length Generalization

In the first experiment we study how well these models work under time-sequences 25x longer than they are trained on, i.e. $T = 10000$ ($\sim$ 80s). Previous work [27] has indicated that because recurrent architectures incorporate feedback they can function unpredictably for sequence lengths beyond those seen during training. On the other hand, convolutional network architectures do not incorporate any feedback. This is advantageous for processing time sequences of indefinite length because errors cannot accumulate over time. Since a convolutional network is a stationary process in the convolved dimension, we hypothesize this architecture will operate more robustly over sequence lengths much longer than those seen during training. Our results are shown in Figure 3.7a. Surprisingly, the results indicate that the BLSTM is also able to generalize to sequences of significantly larger length, contrary to our expectations. We discuss possible explanations of this result in the next section. Our CNN model is able to maintain its performance across the long sequence as expected.

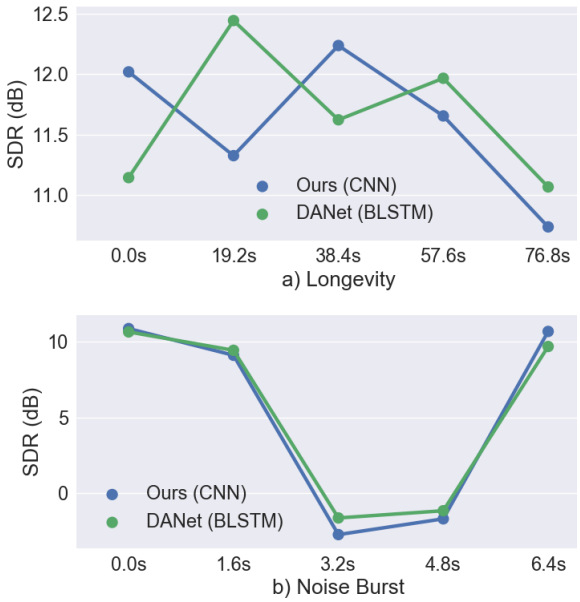

Figure 3.7: Results of the length and noise generalization experiments. For both plots, we plot the SDR starting at the time specified by the x-axis up until 400 time points ($\sim$ 3s) later. Both the BLSTM and CNN are (a) able to operate over very long time sequences and (b) recover from intermittent noise. See Figure 3.8 for the spectrogram in b).

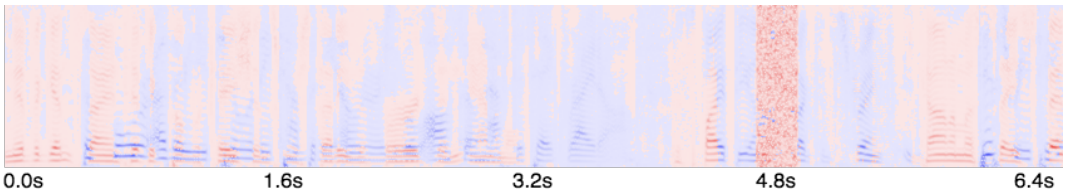

Figure 3.8: Source separation spectrogram for the noise generalization experiment.

Noise Generalization

In the second experiment we are interested in how the models respond to small bursts of input data far outside of the training distribution. We believe the BLSTM model might become unstable as a result of such inputs because its recurrent structure makes it possible for the noise to affect its hidden state indefinitely. We took sequences of length $T = 1200$ ($\sim$ 9s) and inserted white noise for 0.25s in the middle of the mixture to disrupt the models process. Our results are shown in Figure 3.7b. Again, contrary to our belief the BLSTM is very resilient to this noise, the model quickly recovers after the noise passes (last data point). One possible explanation is that the BLSTM is only integrating information over short time scales and therefore "forgets" about previous input data after a short number of time steps. We believe this is because when we construct the input for our models we randomly sample a starting time for each waveform. This may force the BLSTM to learn a stationary function

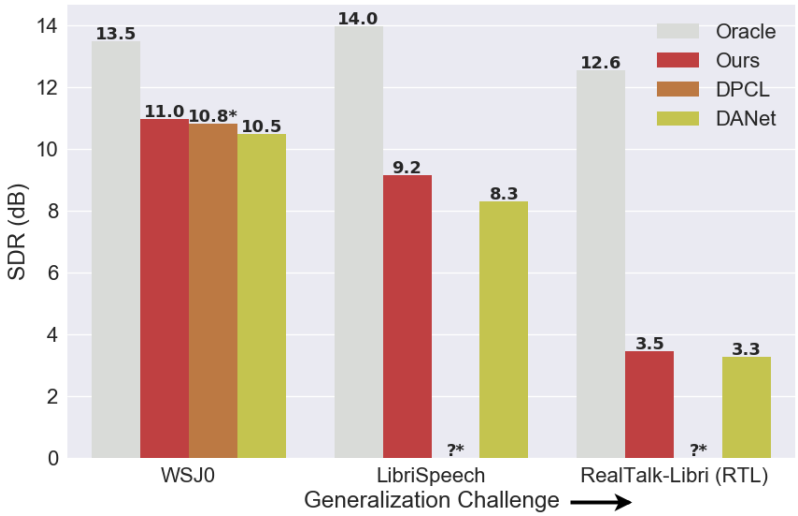

Figure 3.9: Results of the models tested on WSJ0 simulated mixtures, LibriSpeech simulated mixtures, and our RealTalkLibri (RTL) dataset. Our model performs the best on WSJ0, generalizes better to LibriSpeech, but fails alongside the BLSTM at generalizing to the real mixtures of RTL. All models are trained on WSJ0. *: Model has a second neural network to enhance the source estimate spectrogram and is therefore at an advantage. The model wasn't available online for testing against LibriSpeech or RTL.

since it must be able to separate the mixture with or without information from the past in its hidden state.

Data Generalization

In the final experiment we are interested in how well the models generalize to data progressively farther from the training distribution. We trained all the models on the WSJ0 training set and then tested on the WSJ0 test set, the LibriSpeech test set, and *RealTalkLibri* test set. Our results are shown in Figure 3.9. Our model generalizes quite well from the WSJ0 dataset to the LibriSpeech dataset, only losing 1.8dB of performance. Unfortunately it degrades substantially when using the RTL dataset. However, our model still outperforms the DANet model on all datasets.

In Figure 3.10 we visualize the mistakes our network makes under the *RealTalkLibri* dataset. The first example indicates that the model does not have a strong enough bias to the harmonic structure contained in speech, it classifies the frequencies of the fundamental to a different speaker than the harmonic frequencies of that fundamental. The second example indicates that the model also has issues with temporal continuity, the speaker identity of particular frequency bins varies sporadically across time. This indicates that there is still room to improve generalization in these models by modifying model architecture and adding regularization.

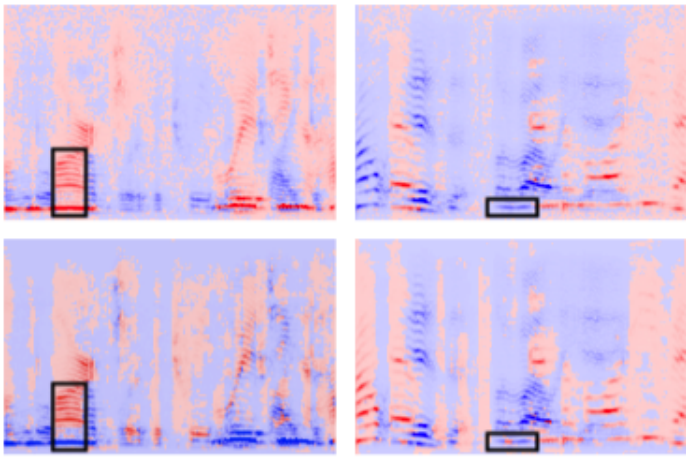

Figure 3.10: *Left and right column*: Two examples of source separation on the *RealTalkLibri* dataset. *Row 1*: Source estimates using the oracle. *Row 2*: The source estimates using our method. The model has difficulty maintaining continuity of speaker identity across frequencies of a harmonic stack (left column) and across time (right column).

3.7 Discussion

Recurrent neural networks have been shown to perform audio source separation with high fidelity. Here we explored using convolutional neural networks as an alternative model. Our state-of-the-art results on the WSJ0 dataset using a factor of ten fewer parameters show that convolutional models are both more accurate and more efficient for audio source separation. Our model has the additional advantage of working online with a fixed-lag response of $\sim$ 1sec.

In order to study the robustness of all models we studied their performance under three different conditions: longer time sequences, intermittent noise, and datasets not seen during training. Our results in the length and noise generalization experiments indicate that the BLSTM learns to behave much like a stationary process in the temporal dimension. We do not observe any substantial degradation in performance after it has been perturbed with noise. It also performs consistently on sequences which are significantly longer than those seen during training.

On the other hand, we get this stationarity property for free with our convolutional model. This further motivates our network architecture in Figure 3.5 which, by design, integrates only local information from the past and future.

In the final experiment we showed that our convolutional neural network also generalized better to both the LibriSpeech dataset and the *RealTalkLibri* dataset we introduced here. Models which are robust to new datasets as well as the deformations caused by the acoustics of different environments are critical to progress in audio source separation. Our *RealTalkLibri* dataset complements other real-world speech datasets [4, 31] by additionally providing approximate ground truth waveforms for the mixture which is currently not available.

Looking forward, we aim to improve the generalization ability of the models on examples such as those shown in Figure 3.10 by introducing a training set for *RealTalkLibri*, developing more robust model architectures, introducing regularizers for the structure of speech, and creating powerful data augmentation tools. We also believe models which can operate under an unknown number of sources is of utmost importance to the field of audio source separation.

Chapter 4

Auditory Attention

4.1 Abstract

In this chapter algorithms for focusing on a specific speaker are investigated, rather than extracting all the sources in the mixture. Recent work has shown that neural networks can perform speaker separation with high fidelity. Most methods have been speaker independent but recently research into methods for speaker biased output have been popular. The main idea is to bias the output of a neural network by feeding it an attentional context, a target speaker embedding that is computed from a target voice snippet. In all previous methods the bias has been for a single speaker which makes the methods inferior to speaker independent methods, which can extract multiple speakers simultaneously. In this chapter a novel idea to perform multi-speaker biased extraction is developed by casting the multi-speaker embedding as a simple superposition of the individual speaker embeddings. This allows our attention models to extract an arbitrary subset of target speakers from a mixture of an unknown number of sources, making the method superior to speaker independent methods, which must know the number of sources in the input. The results show multi-speaker-embeddings perform only slightly worse than single-speaker-embedding demonstrating the effectiveness of the superposition method. Finally, a new method to compute speaker embeddings, called inference-embeddings, is developed which can use substantially more voice data than the voice snippet method of previous work.

4.2 Introduction

While our conscious experience of sound consists of a number of distinct, separate, sound-producing events, the raw auditory stimuli received contains a mixture of them. One of the goals of auditory scene analysis is to separate this complex mixture into the distinct sources that make it [8]. However, this ability super human, humans cannot always focus their attention on every distinct source, and instead must direct their attention to a small set of sources which are behaviourally relevant. This attentional mechanism is thought to be

mediated by top down attention, where high level signals, which correspond to the sources of interest, are fed into lower levels of computation in order to extract the relevant sources [29]. The goal is to develop neural network models which can extract the sources of interest using these high level signals.

Deep learning models of speech separation are not typically viewed from the perspective of top-down attention but there are strong connections in the speaker-biased separation literature. Deep learning based speech separation systems are often classified into two groups: speaker-independent and speaker-biased speech separation. In the first approach the model attempts to separate the sound mixture into all of it's constituent sources at once. Models that follow this approach project components of the sound into a high dimensional space where K-means can be applied to extract the K sources in the mixture. Using deep learning, these methods have been shown to be extremely effective, e.g. deep clustering (DPCL) [22], attractor network [9], etc. [72, 48]. However, a major drawback of this approach is that the number of sources in the mixture, K, must be known before hand otherwise the algorithm fails.

By contrast, the speaker-biased approaches only extract a target speaker, while all other speakers are considered interference. An attention context vector must be supplied to indicate which speaker is of interest to the network. The advantage of this method is that, in theory, this method should be agnostic to the number of inputs in the mixture, which is demonstrated in this chapter. Again using deep learning, this approach has been shown to be quite effective [68, 74, 71] Currently, there are two disadvantages of this method when compared to speaker-independent separation, multiple sources cannot be extracted, and prior knowledge of the speakers must be supplied. More succinctly here are the advantages and disadvantages of these two approaches:

Speaker-Independent

\+ Requires no prior knowledge of speakers

\+ Ability to extract an arbitrary number of sources

\- Requires the number of sources in the mixture to be known

Speaker-Biased

\- Requires prior knowledge of speakers

\- **Can only extract a single source**

\+ Agnostic to the number of sources in the input

But are these current disadvantages inherent to the algorithm? We believe both can be resolved as actual advantages making the speaker-biased approach superior in all respects. In this chapter the focus is on how multiple sources can be extracted. Ideas for how the

inference-embeddings technique could be used to solve the requirement of prior knowledge is explored after.

One of the motivations for solving the multiple source issue is to replicate the human ability to focus attention on a subset of speakers in an environment with many speakers. For example, in the cocktail party problem. The final experiment of this chapter replicates such a scenario.

For speaker-biased methods the prior knowledge of speakers comes from a snippet of reference audio from the target speaker. An additional network computes an embedding from this snippet and hands it to the separation network [71]. This method works well but is limited to using a short piece of reference audio, on the order of 10 seconds. In this chapter, an *inference-embeddings* method is described which can use much more data.

We make the following contributions:

- Extend speaker-biased models to handle the extraction of multiple speakers using embedding superpositions.
- Develop a new method to compute embeddings, termed *inference-embeddings*, which has the advantage of utilizing an arbitrary amount of data rather than a short snippet of reference audio.
- Demonstrate the strongest advantage of speaker-biased models, to be agnostic to the number of sources in the input, using our superposition method.

4.3 Multi-Speaker Extraction

Task Description

We propose a single-channel multi-speaker biased speech separation task. The task is to extract multiple target speakers' voices from a mixed speech waveform. Two sets of speakers are analyzed, known speakers, and new speakers. Known speakers represent speakers our algorithms has substantial data on and help train the baseline model. New speakers represent speakers our algorithms have minimal data on and are challenged to learn to separate well by relying on fine-tuning the baseline model.

Multi-Speaker Extraction Framework

Setup

Notation: For a tensor $T \in \mathbf{R}^{A \times B \times C}$: $T_{\cdot,\cdot,c} \in \mathbf{R}^{A \times B}$ is a matrix, $T_{a,\cdot,c} \in \mathbf{R}^{B}$ is a vector, and $T_{a,b,c} \in \mathbf{R}$ is a scalar.

Let N_k be the number of known speakers, N_n the number of new speakers that must be learned after the initial training on the N_k speakers, and $N = N_k + N_n$ be the total number of speakers.

Each mixture input signal is the sum of two components, a target signal and a interference, or distractor, signal, $x = t + d$, $x \in \mathbb{R}^\tau$. The target and interferer signals are themselves the sum of G and H waveforms respectively, each waveform coming from a different speaker. A special G-hot vector, $B \in \{0,1\}^N$, is used to specify which speakers should be output of the model. More formally:

$$\begin{aligned} z_k &\tilde{\sim} \mathcal{U}\{0, N-1\}, \quad k \in [0, G+H) \\ t &= \sum_{k=0}^{G-1} s_k, \qquad s_k \sim S_{z_k} \\ d &= \sum_{k=G}^{G+H-1} s_k, \quad s_k \sim S_{z_k} \\ x &= t + d \\ B[z_k] &= \begin{cases} 1, & \text{if } k < G \\ 0, & \text{else} \end{cases} \end{aligned}$$

where $\tilde{\sim}$ denotes sampling without replacement, $\mathcal{U}\{\}$ denotes the discrete uniform distribution, and S_i corresponds to another discrete uniform distribution over the waveforms of speaker i. $\mathbb{X} \in \mathbf{R}^{F \times T}$ is the spectrogram of x, computed using the Short-time Fourier transform (STFT), where F corresponds to frequencies and T to time. Analogously $\mathbb{T} \in \mathbf{R}^{F \times T}$ is the spectrogram of t. Before inputting $\mathbb{X}$ to the model we apply a compression function, f_c. Here we use a power-law nonlinearity to compress, inspired by [70]:

$$\begin{aligned} f_c(u) &:= abs(u)^p \\ \bar{\mathbb{X}} &= f_c(STFT(x)), \qquad \bar{\mathbb{T}} = f_c(STFT(t)) \end{aligned}$$

Multi-Speaker Embeddings

All N speakers have a K-dimensional speaker embedding, $E \in \mathbb{R}^{N,K}$. Each multi-speaker embedding, $\mathbb{E} \in \mathbb{R}^K$, is simply a superposition of the desired individual speaker embeddings, which is computed using B. E is included as part of the model parameters and must be learned. $\mathbb{E}$ is an input to the model in addition to the compressed mixture, $\bar{\mathbb{X}}$, so that the model knows which speaker to extract:

$$\begin{aligned} \mathbb{E} &= E^T B \\ O &= f_{NN}(\bar{\mathbb{X}}, \mathbb{E}; \theta) \\ M_{f,t} &= \sigma(O_{f,t}) \\ \hat{\bar{\mathbb{T}}}_{f,t} &= M_{f,t} \odot \bar{\mathbb{X}}_{f,t} \\ \mathcal{L} &= \left\| \bar{\mathbb{T}} - \hat{\bar{\mathbb{T}}} \right\|_F^2 \end{aligned}$$

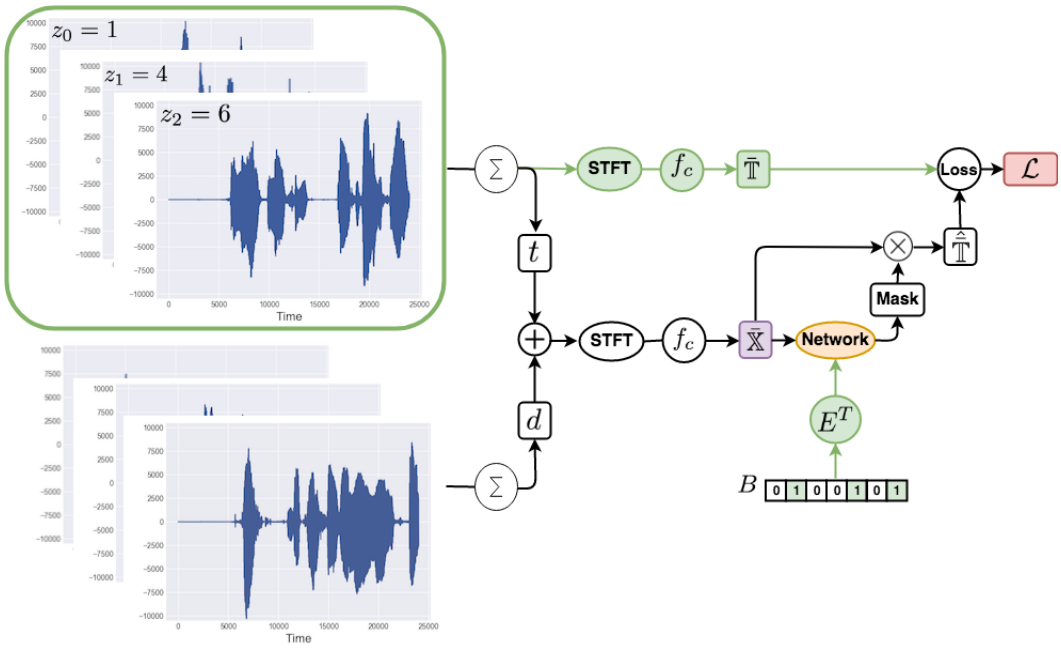

Figure 4.1: Overview of the attention-based multi-speaker separation process. Circles denote operations and rounded squares represent variables. In this diagram G=3, H=3, and N=7, see text for details. Components colored in green represent supervised information about the desired target speaker. Purple indicates the compressed spectrogram, orange the model, and red the loss. f_c corresponds to the compression function used.

where superscript T denotes transpose, f_{NN} is a neural network, θ represents the parameters of the neural network, $O \in \mathbb{R}^{F \times T}$, $M \in [0,1]^{F \times T}$ is a mask, $\odot$ denotes element-wise product, and $||\cdot||_F$ indicates the Frobenius norm. See Figure 4.1 for an overview of the attention processing pipeline. The target estimate can then be recovered using the inverse Short Time Fourier Transform:

$$\hat{t} = ISTFT(f_c^{-1}(\mathbb{T})) \tag{4.1}$$

where the mixture phase from the STFT of x is used since we do not attempt to recover phase information.

Phase 1 Training - Known Speakers

During Phase 1 the neural network parameters and the known speaker embedding parameters are trained:

$$\begin{aligned} \Theta &:= [\theta, E_{i,\cdot}, ..., E_{j,\cdot}], \quad i = 0, j = N_k - 1 \\ \Theta &= \Theta + \eta \frac{\partial L}{\partial \Theta} \end{aligned}$$

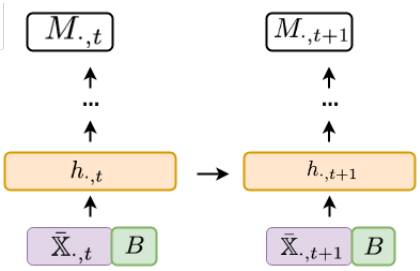

Figure 4.2: LSTM-Append Model. The target specification is encoded in the vector B. To incorporate this into the network B is appended to the normal input at each time step.

where η is the learning rate.

Phase 2 Fine-Tuning - New Speakers

During Phase 2 the model parameters are updated under two different scenarios.

A) Fine-Tuning In this mode all the parameters are updated, making this stage identical to Phase 1 but with new speaker embeddings.

$$\begin{aligned}\Theta &:= [\theta, E_{i,\cdot}, ..., E_{j,\cdot}], \;\; i = N_k, j = N_k + N_n - 1 \\ \Theta &= \Theta + \eta \frac{\partial L}{\partial \Theta}\end{aligned}$$

B) Online Fine-Tuning In this mode new speaker embeddings are treated latent variables that must be inferred given the data. Specifically, the neural network parameters are fixed and only the parameters for the new speaker embeddings are updated:

$$\begin{aligned}\Theta &:= [E_{i,\cdot}, ..., E_{j,\cdot}], \;\; i = N_k, j = N_k + N_n - 1 \\ \Theta &= \Theta + \eta \frac{\partial L}{\partial \Theta}\end{aligned}$$

This method is interesting because it can operate online, the network can continue to perform well on old speakers, while new speaker embeddings are learned without affecting the performance of other speakers. We call this method *inference-embeddings*.

Network Architecture

We use a neural network, BLSTM-FC, which is composed of multiple layers of a Bi-Directional LSTM [21] followed by multiple fully connected layers for the experiments. To incorporate the target information, B is appended to the input and the multi-speaker embedding is then

specified by the weights that connect B to the hidden layer of the BLSTM. This is similar to the approach taken in [71, 68].

$$\begin{aligned} f_{NN}(\bar{\mathbb{X}}, \mathbb{E}; \theta) &= \text{BLSTM-FC}(A) \\ A_{\cdot,t} &:= [\bar{\mathbb{X}}_{t,\cdot}, B], \quad A \in \mathbb{R}^{(F+N)\times T} \end{aligned}$$

For a simple recurrent neural network this would correspond to a bias equivalent to the multi speaker embedding:

$$h_{t+1} = tanh(W_{hh}h_t + W_{ih}\bar{\mathbb{X}}_{t+1,\cdot} + b) \quad // \text{ No Attention (normal RNN)} \tag{4.2}$$

$$h_{t+1} = tanh(W_{hh}h_t + W_{ih}\bar{\mathbb{X}}_{t+1,\cdot} + b + \mathbb{E}) \quad // \text{ Attention} \tag{4.3}$$

where W_{hh} is the hidden to hidden weight matrix, W_{ih} is the input to hidden weight matrix, and b is the normal bias term. See Figure 4.2 for an illustration.

4.4 Experiments

We evaluate the models on a single-channel multi-speaker speech separation task. The mixture waveforms are transformed into a time-frequency representation using the Short-time Fourier Transform (STFT), after which the absolute value and power law function are applied, with $p = 0.3$. The STFT is computed with 32ms window length, 16ms hop size, and the Hann window. The waveform is downsampled to 8kHz to reduce computational cost. The length of the waveforms is set to $\tau = 40000$. We use TensorFlow to compute the STFT and build our neural networks [1].

Results are reported using a distance measure between the source estimate and the true source in the waveform space. Here we use the signal-to-distortion ratio (SDR) which we define as the scale-invariant source-to-noise ratio (SI-SNR) as suggested by [35], and which has also been used in [[22], [9], [44]]:

$$\begin{aligned} t^{proj} &= \frac{\langle \hat{t}, t \rangle t}{\|t\|^2} \\ e_{noise} &= \hat{t} - t^{proj} \\ \text{SDR} := \text{SI-SNR} &:= 10\, log_{10} \frac{\|t^{proj}\|^2}{\|e_{noise}\|^2} \end{aligned}$$

Datasets

To train and evaluate the models the LibriSpeech dataset [53] is used. The train-clean-360 and test-clean portions of the dataset are used for the experiments. The train-clean-360 data set contains 360 hours of speech data spread over $N_k = 916$ speakers, which we call 'known' speakers. The $N_n = 40$ speakers in the test-clean dataset are treated as 'new' speakers. For the known and unknown speakers we split each into two portions for training and evaluation:

Table 4.1: Model hyperparamters

	Model
batch size	128
# of LSTM layers	5
Units per LSTM layer	512
# of FC layers	2
Units per FC layer	512
Embedding Size (K)	512

Known, $N_k = 916$

- known-eval: The first 100 seconds of each speaker in train-clean-360
- known-train: The remaining seconds of data ($\gg 100$) of each speaker in train-clean-360

New, $N_n = 40$

- unknown-train: The first 100 seconds of each speaker in test-clean
- unknown-eval: The remaining seconds of data ($\gg 100$) of each speaker in test-clean

During training sampling is done uniformly from the speaker set, and the target and interferer signal are mixed with a signal-to-noise ratio (SNR) uniformly sampled from [-5dB, 5dB].

Network hyperparameters

The model hyperparameters are shown in Table 4.1. RMSProp [63] is used to optimize the networks.

Single Target, Single Interferer Experiments

In these experiment there is a single target and a single interferer, $G = 1, H = 1$, which is identical to the audio source separation scenario in previous work such as [9].

Known Speakers Experiment

We begin by evaluating the model on the known speaker set. The results are compared with the only deep clustering result for LibriSpeech available [48] in Table 4.2. They indicate that speaker-biased models, can perform comparably to speaker-independent (DPCL) methods which validates the speaker-biased approach.

Table 4.2: SDR of our method on the known-eval partition compared to deep clustering results from Mobin et al. [48] on the test-clean partitions of LibriSpeech. While these partitions are not identical they come from the same dataset and are comparable.

	known-eval	test-clean
Model	10.1 dB	x
DPCL Mobin et al. [48]	x	9.2 dB

Table 4.3: SDR for the Phase 2 fine-tuning on the new speakers. Results are compared to the baseline attention network of Xiao et al. [71]

	Unknown-eval	test-clean
Model Fine-Tuning	8.5dB	x
Model Online-Fine-Tuning	7.8dB	x
Xiao-Attention-Model	x	9.8dB

New Speakers Experiment

Next, we investigate the model's ability to adapt to new speakers that have limited data. 100 seconds of speech data are used per speaker. The baseline model trained on the known speakers is used as a starting point for fine-tuning on the new speakers. Our results are shown in Table 4.3.

Table 4.4: SDR for our Phase 1 training on known speakers where there are multiple targets and interferers. No other neural network model is capable of working on this task.

	Multi-Target-Interferer (known-eval)	Single-Target-Interferer (best result)
Model	9.5 dB	10.1 dB
Xiao-Attention-Model	Not possible	9.8 dB
DPCL	Not Possible	9.2 dB

Multi Target, Multi Interferer Experiments

In this experiment we demonstrate the most powerful component of our methodology. The ability of the network to track multiple voices in the context of an unknown number of speakers in the input mixture. For each batch element:

$$\begin{aligned} G &\sim \mathcal{U}\{1,3\}, \\ H &\sim \mathcal{U}\{1,3\}, \end{aligned}$$

so that there are between 2 and 6 total speakers in the mixture, and between 1 and 3 speakers that the model must extract. To keep the results comparable to previous work there are only two speakers speaking at any given timepoint. This is done by making only one speaker active at any given time point in both the target and interferer signals. The target can be thought of as a conversation between 1 and 3 individuals, and the interferer as another conversation between 1 and 3 individuals that our model tries to ignore. Our results are shown in Table 4.4.

Interestingly, the model is quite robust to the multi-target, multi-interferer scenario, degrading by only 6% in SDR performance compared to the single-target, single-interfere scenario. This is quite impressive as the number of possible embeddings increases from 916 to $\sum_{i=1}^{3} \binom{916}{i} = 12809664$.

4.5 Discussion

Speaker-biased approaches have been of interest recently in the audio deep learning community. However a major limitation of these models when compared to speaker-independent approaches is that they could not extract multiple speakers. Here we showed how using a simple idea, a superposition of embeddings, multi-target extraction is not only possible but can perform at a comparable level to single-target extraction. Further, we demonstrated one of the strongest advantages of speaker-biased approaches in comparison to speaker-independent approaches, to be agnostic to the number of sources in the input. We now summarise the new advantages and disadvantages of speaker-independent vs. speaker-biased approaches:

Speaker-Independent

+ Requires no prior knowledge of speakers
+ Ability to extract an arbitrary number of sources
- Requires the number of sources in the mixture to be known

Speaker-Biased

- Requires prior knowledge of speakers
+ **Ability to extract an arbitrary number of sources**
+ Agnostic to the number of sources in the input

A new method for deriving speaker embeddings was also developed, termed *inference-embeddings*. While this method did not perform as well as the reference-audio approach, little hyperparameter tuning was done on the models. This method is also of interest because it could resolve the remaining disadvantage of the speaker-biased approach. The idea is that once a baseline model is constructed, speaker embeddings are essentially voice fingerprints that can be used to extract a specific voice. When the model wants to learn about a new speaker, it can find a new embedding which is different than all others it knows about, but which also causes the network to output something that sounds like human speech. In this chapter a supervised signal was used to validate human speech output. But instead, a discriminator in a generative adversarial network [20] could be used to ensure the new embedding corresponded to human speech.

Humans do not have the ability to simultaneously track all sources inside a sound mixture. They utilize their attention mechanism to identify and learn about new sources on-the-fly, in the spirit of continual learning. This work is a step towards more human-like source separation which has substantial room for research innovation.

Chapter 5

Conclusion

5.1 Closing remarks

Three key questions were asked: 1) Why did the cochlea evolve to use a time-frequency code? 2) How do we build human-level robust statistical models of sound? 3) How does feedback in the auditory system guide our attention? A theoretical model based on efficient coding and leaky-integrate-and-fire (LIF) neurons was created to explain the auditory nerve response. A more efficient neural network architecture was designed which was more robust to shifts in the data distribution. Finally, a novel algorithm was developed to allow for high-level neural signaling to modulate bottom-up processing to explain auditory attention. This work forms a foundation for others to continue to make progress on these questions.

Future Work

In Chapter 2 gradients of a non-smooth forwards pass, from the LIF neurons, were replaced with a smooth backwards pass to help optimization. This wasn't enough to prevent the model from converging to sub-optimal local minima. One possible explanation is that small changes in the forwards pass still dramatically change the output, despite the changes in the backwards pass. It would be interesting to smooth the forwards pass as well but gradually iterate towards the non-smooth version. This might help optimization become more stable.

In Chapter 3 different neural network architectures were explored to improve generalization to new data domains. Another approach is data augmentation. The idea is to generate additional data that is still in the class of natural sound using existing, fixed datasets. How could we generate more natural sound? One observation is that the real world contains much more reverberation, essentially short-time scale echoes, than existing datasets. Often audio datasets are recorded with headsets that eliminate reverb, or in otherwise less realistic environments. To generate more realistic data we could simulate reverberation using a set of all pass and feedback comb filters as in the work of Schroeder [58].

In Chapter 4 a new algorithm was developed to extract multiple voices from a mixture using an attentional context. One shortcoming of this algorithm is that the speakers, and

audio of the speakers, had to be known before hand. However, humans have no such limitation, we learn the voices of new speakers all the time. It would be very interesting to have the model have a *discovery* mode which allows it to learn new voices. One idea for doing this is to encourage the algorithm to find an embedding that is dissimilar to all speaker embeddings it knows about, but also causes the network to output natural sounding speech. Specifically, gradients could be taken with respect to the embedding of a loss function with two components. The first would be the negative sum of the mean-squared-error between the current embedding and all known embeddings. The second would be the generator loss in a Generative Adversarial Network (GAN), where the output of our current model should be a sound that pleases a discriminator of a GAN. The discriminator in this GAN would be trained to discriminate between speech and not-speech. Using this framework the algorithm could become more autonomous, learning about new voices on-the-fly, and be able to extract new voices given the correct speaker embeddings. This innovation would start to bridge the gap between the capabilities of our attentional models and those of human attention.

5.2 Next Steps

Over the last few years I've thought a lot about hearing science [45], deep learning models of sound, and how we can combine the two to help the hearing impaired hear better. A good place to start is with a problem. In a paper called "Why do people fitted with hearing aids not wear them?" [47] McCormack et. al. found that the most significant reason people didn't wear hearing aids was because a lack of speech clarity. In other words modern hearing aids are unable to help them hear the voices of their peers so they didn't see any point in wearing them. With further research we find that indeed everyone with hearing aids complains that they cannot hear voices in noisy places like restaurants and bars. And it's a serious problem, leading to the 30 million hearing impaired Americans to be twice as likely to develop depression [40] and two to five times more likely to develop dementia [43] due the social isolation they experience.

What can be done about this? In Chapter 4 a model was developed to selectively focus on a subset of voices in the context of many voices. If we think of the restaurant as mixture of many voices, say 100, then the goal of a hearing aid should be to amplify a select few of those voices, say 5, that originate at our table. We then see a perfect application of our research, we build hearing aids that are customized for each user, to attend to the specific voices of their friends and family and ignore everyone else. In Chapter 4 we showed that only a small sample of audio data was needed per target speaker to extract their voice. Therefore each user could simply collect voice samples from their peers to accomplish this. I believe there is a big opportunity to take this research outside the lab and into a product that improves the quality of lives of tens of millions of people around the world. And of course, this idea is just the beginning, continuing research in this domain will lead to even more fruitful methods of helping the hearing impaired.

Bibliography

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[2] Joseph J Atick and A Norman Redlich. What does the retina know about natural scenes? *Neural computation*, 4(2):196–210, 1992.

[3] Fred Attneave. Some informational aspects of visual perception. *Psychological review*, 61(3):183, 1954.

[4] Jon Barker, Ricard Marxer, Emmanuel Vincent, and Shinji Watanabe. The third chime-speech separation and recognition challenge: Dataset, task and baselines. In *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, pages 504–511. IEEE, 2015.

[5] Horace Barlow. Redundancy reduction revisited. *Network: computation in neural systems*, 12(3):241–253, 2001.

[6] Horace B Barlow. Possible principles underlying the transformations of sensory messages. 1961.

[7] Alexander Borst and Frédéric E Theunissen. Information theory and neural coding. *Nature neuroscience*, 2(11):947–957, 1999.

[8] Albert S Bregman. *Auditory scene analysis: The perceptual organization of sound*. MIT press, 1994.

[9] Zhuo Chen, Yi Luo, and Nima Mesgarani. Deep attractor network for single-microphone speaker separation. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 246–250. IEEE, 2017.

[10] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016.

[11] Chris J Darwin. Auditory grouping. *Trends in cognitive sciences*, 1(9):327–333, 1997.

[12] Eizaburo Doi and Michael S Lewicki. A theory of retinal population coding. *Advances in neural information processing systems*, 19:353, 2007.

[13] Eizaburo Doi, Doru C Balcan, and Michael S Lewicki. Robust coding over noisy overcomplete channels. *IEEE Transactions on Image Processing*, 16(2):442–452, 2007.

[14] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014.

[15] Chris Eliasmith and Charles H Anderson. *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT press, 2004.

[16] EF Evans and AR Palmer. Proceedings: Responses of units in the cochlear nerve and nucleus of the cat to signals in the presence of bandstop noise. *The Journal of physiology*, 252(2):60P, 1975.

[17] Cédric Févotte, Rémi Gribonval, and Emmanuel Vincent. Bss_eval toolbox user guide–revision 2.0. 2005.

[18] David J Field. Relations between the statistics of natural images and the response properties of cortical cells. *JOSA A*, 4(12):2379–2394, 1987.

[19] John Garofalo, David Graff, Doug Paul, and David Pallett. Csr-i (wsj0) complete. *Linguistic Data Consortium, Philadelphia*, 2007.

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[21] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Bidirectional lstm networks for improved phoneme classification and recognition. *Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005*, pages 753–753, 2005.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[23] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 31–35. IEEE, 2016.

[24] Wei-Ning Hsu, Yu Zhang, and James Glass. Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation. *arXiv preprint arXiv:1707.06265*, 2017.

[25] Ke Hu and DeLiang Wang. An unsupervised approach to cochannel speech separation. *IEEE Transactions on audio, speech, and language processing*, 21(1):122–131, 2013.

[26] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R Hershey. Single-channel multi-speaker separation using deep clustering. *arXiv preprint arXiv:1607.02173*, 2016.

[27] Łukasz Kaiser and Ilya Sutskever. Neural gpus learn algorithms. *arXiv preprint arXiv:1511.08228*, 2015.

[28] Yan Karklin and Eero P Simoncelli. Efficient coding of natural images with a population of noisy linear-nonlinear neurons. In *NIPS*, volume 24, pages 999–1007, 2011.

[29] Emine Merve Kaya and Mounya Elhilali. Modelling auditory attention. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 372(1714):20160101, 2017.

[30] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[31] Keisuke Kinoshita, Marc Delcroix, Takuya Yoshioka, Tomohiro Nakatani, Armin Sehr, Walter Kellermann, and Roland Maas. The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech. In *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*, pages 1–4. IEEE, 2013.

[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[33] Michael F Land and Dan-Eric Nilsson. *Animal eyes*. Oxford University Press, 2012.

[34] Quoc V Le, Alexandre Karpenko, Jiquan Ngiam, and Andrew Y Ng. Ica with reconstruction cost for efficient overcomplete feature learning. In *Advances in Neural Information Processing Systems*, pages 1017–1025, 2011.

[35] Jonathan Le Roux, JR Hershey, A Liutkus, F Stöter, ST Wisdom, and H Erdogan. Sdr–half-baked or well done? *Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA, Tech. Rep*, 2018.

[36] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[37] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553): 436, 2015.

[38] Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer. Training deep spiking neural networks using backpropagation. *Frontiers in Neuroscience*, 10, 2016.

[39] Michael S Lewicki. Efficient coding of natural sounds. *Nature neuroscience*, 5(4):356–363, 2002.

[40] Chuan-Ming Li, Xinzhi Zhang, Howard J Hoffman, Mary Frances Cotch, Christa L Themann, and M Roy Wilson. Hearing impairment associated with depression in us adults, national health and nutrition examination survey 2005-2010. *JAMA otolaryngology–head & neck surgery*, 140(4):293–302, 2014.

[41] Jinyu Li, Li Deng, Yifan Gong, and Reinhold Haeb-Umbach. An overview of noise-robust automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4):745–777, 2014.

[42] MC Liberman. Single-neuron labeling in the cat auditory nerve. *Science*, 216(4551): 1239–1241, 1982.

[43] Frank R Lin, E Jeffrey Metter, Richard J OBrien, Susan M Resnick, Alan B Zonderman, and Luigi Ferrucci. Hearing loss and incident dementia. *Archives of neurology*, 68(2): 214–220, 2011.

[44] Yi Luo and Nima Mesgarani. Tasnet: time-domain audio separation network for real-time, single-channel speech separation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 696–700. IEEE, 2018.

[45] Richard F Lyon. *Human and machine hearing*. Cambridge University Press, 2017.

[46] Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415, 1993.

[47] Abby McCormack and Heather Fortnum. Why do people fitted with hearing aids not wear them? *International journal of audiology*, 52(5):360–368, 2013.

[48] Shariq Mobin, Brian Cheung, and Bruno Olshausen. Generalization challenges for neural architectures in audio source separation. *arXiv preprint arXiv:1803.08629*, 2018.

[49] Peter O'Connor and Max Welling. Deep spiking networks. *arXiv preprint arXiv:1602.08323*, 2016.

[50] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607, 1996.

[51] Bruno A Olshausen and Kevin N O'Connor. A new window on sound. *nature neuroscience*, 5(4):292, 2002.

[52] AR Palmer and IJ Russell. Phase-locking in the cochlear nerve of the guinea-pig and its relation to the receptor potential of inner hair-cells. *Hearing research*, 24(1):1–15, 1986.

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

[54] Colin Raffel, Brian McFee, Eric J Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel PW Ellis, and C Colin Raffel. mir_eval: A transparent implementation of common mir metrics. In *In Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR*. Citeseer, 2014.

[55] Alberto Recio-Spinoso, Andrei N Temchin, Pim van Dijk, Yun-Hui Fan, and Mario A Ruggero. Wiener-kernel analysis of responses to noise of chinchilla auditory-nerve fibers. *Journal of neurophysiology*, 93(6):3615–3634, 2005.

[56] Jerzy E Rose, John F Brugge, David J Anderson, and Joseph E Hind. Phase-locked response to low-frequency tones in single auditory nerve fibers of the squirrel monkey. *Journal of neurophysiology*, 30(4):769–793, 1967.

[57] Jan Schnupp, Israel Nelken, and Andrew King. *Auditory neuroscience: Making sense of sound*. MIT press, 2011.

[58] Manfred R Schroeder. Natural sounding artificial reverberation. *Journal of the Audio Engineering Society*, 10(3):219–223, 1962.

[59] William Arthur Sethares. *Rhythm and transforms*. Springer Science & Business Media, 2007.

[60] Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon. An end-to-end neural network for polyphonic piano music transcription. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 24(5):927–939, 2016.

[61] Evan C Smith and Michael S Lewicki. Efficient auditory coding. *Nature*, 439(7079): 978–982, 2006.

[62] Gonzalo Terreros and Paul H Delano. Corticofugal modulation of peripheral auditory responses. *Frontiers in systems neuroscience*, 9:134, 2015.

[63] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[64] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1521–1528. IEEE, 2011.

[65] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4, 2017.

[66] Mats Ulfendahl and Åke Flock. Outer hair cells provide active tuning in the organ of corti. *Physiology*, 13(3):107–111, 1998.

[67] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE transactions on audio, speech, and language processing*, 14(4):1462–1469, 2006.

[68] Quan Wang, Hannah Muckenhirn, Kevin Wilson, Prashant Sridhar, Zelin Wu, John Hershey, Rif A Saurous, Ron J Weiss, Ye Jia, and Ignacio Lopez Moreno. Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking. *arXiv preprint arXiv:1810.04826*, 2018.

[69] Bo Wen, Grace I Wang, Isabel Dean, and Bertrand Delgutte. Dynamic range adaptation to sound level statistics in the auditory nerve. *Journal of Neuroscience*, 29(44):13797–13808, 2009.

[70] Kevin Wilson, Michael Chinen, Jeremy Thorpe, Brian Patton, John Hershey, Rif A Saurous, Jan Skoglund, and Richard F Lyon. Exploring tradeoffs in models for low-latency speech enhancement. In *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 366–370. IEEE, 2018.

[71] Xiong Xiao, Zhuo Chen, Takuya Yoshioka, Hakan Erdogan, Changliang Liu, Dimitrios Dimitriadis, Jasha Droppo, and Yifan Gong. Single-channel speech extraction using speaker inventory and attention network. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 86–90. IEEE, 2019.

[72] Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 241–245. IEEE, 2017.

[73] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[74] Katerina Zmolikova, Marc Delcroix, Keisuke Kinoshita, Takuya Higuchi, Atsunori Ogawa, and Tomohiro Nakatani. Speaker-aware neural network based beamformer for speaker extraction in speech mixtures. In *Interspeech*, pages 2655–2659, 2017.