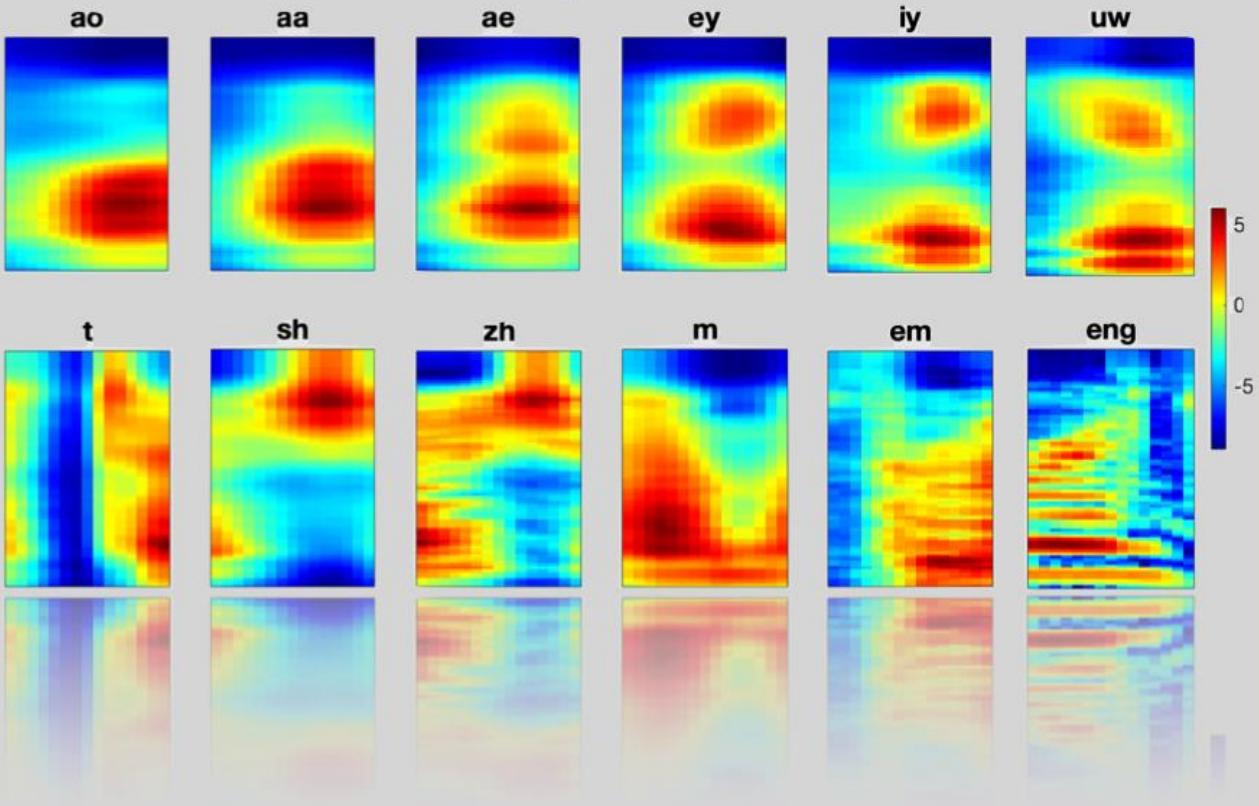
Neural Representation Learning for Speech and Audio Signals



Purvi Agrawal

Neural Representation Learning For Speech and Audio Signals

A dissertation submitted
in partial fulfillment of the
requirements for the Degree of

Doctor of Philosophy

in the **Faculty of Engineering**

by

Purvi Agrawal

Department of Electrical Engineering
INDIAN INSTITUTE OF SCIENCE
BENGALURU-560012
2020

Purvi Agrawal
Neural Representation Learning For Speech and Audio Signals

SUPERVISOR:
Dr. Sriram Ganapathy
LEAP - Learning and Extraction of Acoustic Patterns Lab
Department of Electrical Engineering
Indian Institute of Science, Bengaluru-560012

DEDICATED TO

Everyone who is fascinated by ''What are speech representations? How do machines understand the signals? Can a machine's speech/audio processing be made similar to humans?''

"I think therefore I am.
I share therefore I have a world.''
– in Principles of Philosophy (1644) by René Descartes.

Acknowledgements

I am infinitely indebted to Dr. Sriram Ganapathy for giving me the opportunity to work with him in LEAP lab as his first research student, and providing me with perseverance and freedom throughout my graduate life. During this period, he allowed me to follow my research interests and passion, along with motivation and push for a fit and healthy body. This really helped me in carrying out the balance efficiently, without hustle. The detailed discussions with him in the entire period has immensely contributed to this final thesis. His huge efforts in setting the GPU cluster facility in EE department helped me (and everyone) significantly in carrying out the machine learning experiments.

I am highly grateful to Dr. Neeraj Sharma for the discussions on the research work, his patience in correcting the presentations, and support from the time he started working with LEAP lab. The detailed discussions with him really helped me to understand and think about critical aspects of the work with a different viewpoint. I express my gratitude to Shreyas Ramoji for technical discussions in the lab as well as managing the conference/workshop trips.

I would like to thank my comprehensive examination panel [Dr. Partha Pratim Talukdar, Dr. K.V.S. Hari, Dr. Chandra Sekhar Seelamantula] and departmental curriculum committee [DCC panel - Dr. P.S. Sastry and Dr. Chandra Sekhar Seelamantula] for valuable suggestions towards the work. I would also like to thank my colleagues at Sony R & D, Tokyo where I interned with audio source separation team. The support from Mr. Naoya Takahashi and Mr. Nabarun Goswami led to the smooth execution in whole internship period in Japan and the publication process.

Since my journey at IISc started in Speech and Audio Group (SAG) lab, I am thankful to SAG lab members for the initial support and help. On personal level, I express my gratitude for the love and support from my family members, my friends Lekshmi, Mayank, Mohammadi Zaki, Shubham, and my cousin's family in Bangalore - Ajay and Mahima.

Last but not least, I am grateful to the suggestions from my thesis examination committee members - Prof. S. Umesh, and Prof. Hynek Hermansky. Their comments and suggestions resulted in improving the final thesis significantly.

Contents

Abstract

Representation learning is the branch of machine learning consisting of techniques that are capable of automatically discovering meaningful representations from raw data for efficient information extraction. In the recent years, following the trends in other streams of machine learning, representation learning using neural networks has attracted significant interest. For example, deep representation learning in the text domain using word embeddings has shown interesting semantic properties that make them widely useful for many natural language processing applications. In the speech processing field, representation learning has been a challenging task. This thesis is focused on developing neural methods for representation learning of speech and audio signals, with the goal of improving downstream applications that rely on these representations.

For representation learning, we pursue two broad directions - supervised and unsupervised. In the case of speech/audio signals, we identify two stages of representation learning that are explored. The first stage is the learning of a time-frequency representation (equivalent of spectrogram) from the raw audio waveform. The second stage is the learning of modulation representations (filtering the time-frequency representations along the temporal domain, called rate filtering and spectral domain, called scale filtering).

In the first part of the thesis, we propose representation learning methods for speech data in an unsupervised manner. Using the modulation representation learning as the goal, we explore various neural architecture for unsupervised learning. These include restricted Boltzmann machines (RBM), variational autoencoders (VAE) and generative adversarial networks (GAN). For learning modulation representations that are distinct and irredundant, we propose different learning frameworks like external residual approach, skip connection based approach, and a modified cost function based approach. The methods developed for rate and scale representation learning are benchmarked using an automatic speech recognition (ASR) task on noisy and reverberant conditions. We also illustrate that the unsupervised representation learning can be extended to the first stage of learning time-frequency representations from raw waveforms.

The second part of the thesis deals with supervised representation learning. Here, we propose a two-stage representation learning approach from raw waveform consisting of acoustic filterbank learning (time-frequency representation learning) from raw waveform followed by a modulation representation learning. This two-stage learning is directly optimized for the task at hand. The key novelty in the proposed framework consists of a relevance weighting mechanism that acts as a feature selection module. This is inspired by gating networks and provides a mechanism to weight the relevance of the acoustic and modulation representations for the task involved. The relevance weighting network can also utilize feedback from the previous predictions of the model for tasks like ASR. The proposed relevance weighting scheme is shown to provide significant performance improvements for ASR task and UrbanSound audio classification task. A detailed analysis yields insights into the interesting properties of the relevance weights that are captured by the model at the acoustic and modulation stages for speech and audio signals. In particular, the relevance weights are shown to succinctly capture phoneme characteristics in speech recognition tasks and the audio characteristics in the urban sound classification task.

In summary, the thesis makes strides in the direction of unsupervised and supervised neural representation learning of speech and audio signals. While conventional methods of speech/audio processing involve deriving time-frequency spectrogram representations as the first step in most classification tasks, the work

reported in the thesis argues that data driven representations from the raw signal with minimal assumptions can yield task specific flexibility and interpretability while also providing superior performance.

Publications

Peer-reviewed Journal Papers

(1) P. Agrawal, S. Ganapathy, "Interpretable Representation Learning for Speech and Audio Signals Based on Relevance Weighting," *IEEE Transactions on Audio, Speech and Language Processing*, 28, (2020): 2823-2836.
(2) P. Agrawal, S. Ganapathy, "Modulation filter learning using deep variational networks for robust speech recognition," *IEEE Journal of Selected Topics in Signal Processing, Special Issue on Data Science: Machine Learning for Audio Signal Processing*, 13, no. 2 (2019): 244-253.
(3) P. Agrawal, S. Ganapathy, "Unsupervised Modulation Filter Learning for Noise-Robust Speech Recognition," *Journal of Acoustical Society of America*, 142, no. 3 (2017): 1686-1692.

Peer-reviewed Conference Papers

(1) P. Agrawal, S. Ganapathy, "Representation Learning For Speech Recognition Using Feedback Based Relevance Weighting," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2021.
(2) P. Agrawal, S. Ganapathy, "Robust Raw Waveform Speech Recognition Using Relevance Weighted Representations", *INTERSPEECH*, pp. 1649-1653, 2020.
(3) P. Agrawal, S. Ganapathy, "Unsupervised Raw Waveform Representation Learning for ASR," *INTERSPEECH*, pp. 3451-3455, 2019.
(4) P. Agrawal, S. Ganapathy, "Deep variational filter learning models for speech recognition," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 5731-5735, 2019.
(5) P. Agrawal, S. Ganapathy, "Comparison of unsupervised modulation filter learning methods for ASR," *INTERSPEECH*, pp. 2908-2912, 2018.
(6) N. Takahashi, P. Agrawal, N. Goswami, Y. Mitsufuji, "PhaseNet: Discretized phase modeling with deep neural networks for audio source separation," *INTERSPEECH*, pp. 2713-2717, 2018.
(7) P. Agrawal, S. Ganapathy, "Speech Representation Learning Using Unsupervised Data-Driven Modulation Filtering for Robust ASR," *INTERSPEECH*, pp. 2446-2450, 2017.

Challenges

(1) P. Agrawal, S. Singh, J. Shankar, S. Ganapathy, P. Jyothi, "Interspeech 2018 Special Session: Low Resource Speech Recognition Challenge for Indian Languages - by Microsoft Research," finished 3^{rd} on leader-board with team CSALT-LEAP.
(2) S. Ganapathy, P. Agrawal, "The 5th CHiME Speech Separation and Recognition Challenge 2018," finished in top 10 teams for single-array track with team LEAP.

Softwares and Facilities Used

Software toolkits: The following software tools aided in the computer implementation of the research presented in this thesis.

(1) **KALDI toolbox** for training a conventional speech recognition system (feature extraction, generation of alignments, language modeling, acoustic model training, decoding)
Open-source download link: https://kaldi-asr.org/doc/install.html

(2) **Python with PyTorch library** for design and training of almost all unsupervised and supervised representation learning models discussed in the thesis.
Open-source download link: https://docs.conda.io/en/latest/miniconda.html
https://pytorch.org/

(3) **MATLAB software** for signal processing involved in feature extraction, as well as all plotting work.
Licensed version download link: https://in.mathworks.com/downloads/

Research facilities: The following research facilities played crucial role in conducting the research presented in this thesis.

(1) *Cluster with GPU grid, Cluster Room in the Dept. EE, IISc.* This was used for all the system training on large dataset and was used in all of the experimentation.

(2) *Local computer system in LEAP lab, Dept. EE, IISc.* This was used almost every day for 4 years of PhD duration.

Codes: Some of the codes of the proposed work in the thesis can be found at:
https://github.com/iiscleap?tab=repositories.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Representations

Representations of data refer to the form in which data is stored or processed. The performance of machine learning methods is heavily dependent on the choice of data representation (or features). For this reason, much of the actual effort in deploying machine learning algorithms goes into the design of pre-processing pipelines and data transformations that result in a representation that enables effective machine learning.

The generation of representations of data involves the conversion of raw data to some numerical form that contains the data characteristics. The raw data can be of any form, for example, image, text, speech/audio, etc. The primary goal of having data representation is to be able to succinctly characterize the data.

Data representations can be derived by either using domain knowledge or can be learnt from the data itself. The traditional methods to obtain representations from data are mostly knowledge-driven approaches, termed as feature engineering. The feature engineering is sometimes bio-inspired to imitate the processing of the data in humans or driven by heuristics about domain of the data. The handcrafted feature engineering to obtain representations is labor-intensive and may or may not benefit the downstream task, since they are common generic representations. Feature engineering is also a manual process that requires a good amount of domain knowledge. Therefore, for improved benefits in the downstream task, it may be preferable to learn representations from the data, termed as 'representation learning'.

1.2 Representation Learning

Representation learning refers to the task of learning "meaningful" representations from the data with the output being termed as data-driven representations [9]. The motivation for learning the representations is to make the learning algorithms less dependent on feature engineering or human intervention.

Some examples of representation learning are principal component analysis (PCA), t-stochastic neighbourhood embeddings (t-SNE), multi-dimensional scaling (MDS), linear discriminant analysis (LDA), etc. Representation learning can be carried out in supervised or unsupervised manner. The PCA, t-SNE, MDS belong to the unsupervised learning category. Both PCA and LDA are linear data transformation techniques, however, LDA is a supervised method that requires class labels.

With deep learning advancement in recent years, neural approaches have been explored to learn the representations. The deep learning based representation learning is termed as neural representation learning. The applications of neural representation learning include dimensionality reduction, information retrieval, data denoising, clustering, data abstraction and invariance, etc. With various information extraction systems becoming neural, the use of neural representation learning is attractive as the representation learning block can be integrated with the rest of the model and optimized jointly.

1.2.1 *Example of Representation Learning*

The neural representation learning approach has shown success in many domains. A successful example of this representation learning is the Word2vec model in text domain to learn word embeddings (word

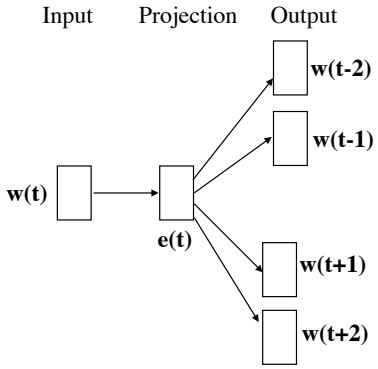


Fig. 1.1: An example of representation learning in text - word2vec model with skip-gram method [70].

representations) [70]. The Word2vec is a two-layer neural network that processes text by 'vectorizing' words. Its input is a word $\boldsymbol{w}(t)$ from text corpus as a one-hot vector and its output is the one-hot vector of context words: words in neighbourhood of the input word, i.e. $\boldsymbol{w}(t\pm2)$, $\boldsymbol{w}(t\pm1)$ (skip-gram method), shown in Figure 1.1. The intermediate representations $\boldsymbol{e}(t)$ are the feature vectors that represent words in that corpus and these representations have shown to embed meaningful semantic properties [70]. The word2vec is an unsupervised representation learning method and has been successfully used for many applications [114, 12].

1.2.2 *Challenges in Speech/Audio Representation Learning*

Natural speech has very complex manifolds and inherently contains information about the message, language, speaker characteristics (gender, age, health status), emotion, etc. All of this information is entangled together, and the disentanglement of these attributes in some latent space is a challenging task that may require extensive training [57, 9]. In addition to source variability, the non-stationary nature of speech/audio data with high levels of temporal variability creates difficulties in representation learning.

There have been attempts recently on representation learning for speech and audio signals. However, compared to vector representations of text (obtained from word2vec) which have shown to embed meaningful semantic properties, the interpretability of speech representations from existing approaches has often been limited. Learning speech representations that are domain invariant, i.e, invariant to variabilities in speakers, language, etc., have been cumbersome. The performance of representations learnt from one corpus may not work well in another corpus having different recording conditions. While there has been some success in learning speaker invariant representations, language invariant representation learning is still very challenging. In addition, the training of unsupervised representation learning models is more difficult in contrast to supervised ones. As highlighted in [9], in supervised learning, there is a clear objective to optimize, for example, the classifiers are trained to learn such representations that minimize the mis-classification errors, which is absent in unsupervised representation learning task. The interpretability remains negligible or limited in most supervised learning approaches as well.

1.3 Feature Engineering in Speech/Audio

Before we discuss representation learning of speech/audio in detail, we briefly highlight major feature engineering approaches popular in speech/audio processing.

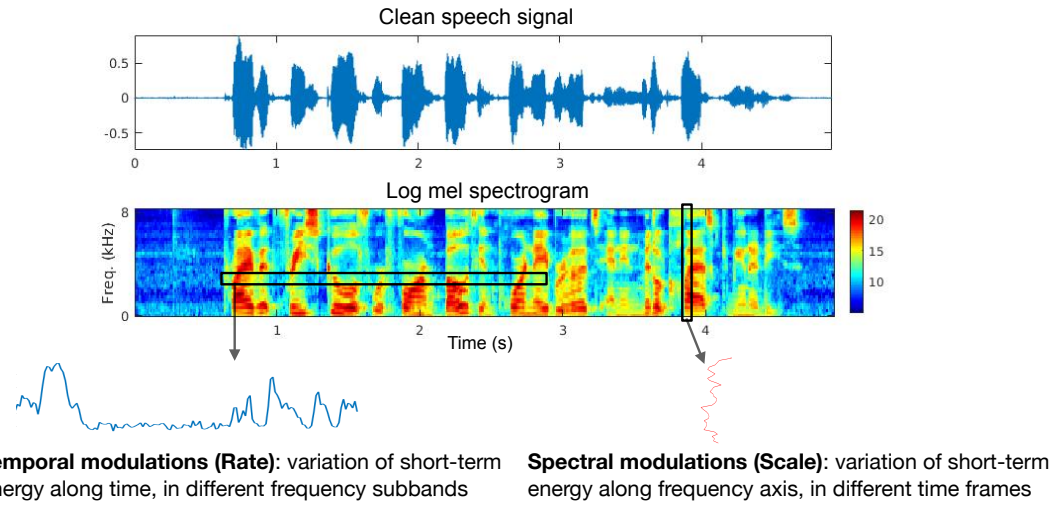


Fig. 1.2: Traditional log mel filterbank energy representation of clean speech signal, and highlighting the modulations in the representation.

1.3.1 *Mel Spectrogram Representation*

The traditional speech/audio representation (features) used in most of the machine learning algorithms is derived from short-term spectral energies called as spectrogram. It is computed using spectral energies of overlapping windows of raw waveform, with window length of around $20-30$ms shifted by around 10ms. The signal is framed using Hamming ($20-30$ms) window and the magnitude of the Fourier transform is computed in each frame (short-term Fourier transform (STFT)). Let x be the input signal, then

$$X(\omega, t) = \mathcal{F}[x(\tau)w(\tau - t)] \tag{1.1}$$

denotes the STFT of input $\boldsymbol{x}$ at frame t, $w(\tau - t)$ is the window function centered at time t, ω represents frequency. The term $|X(\omega, t)|^2$ is termed as power spectra or short-term spectral energies of the signal x. For extracting the mel spectrogram, the mel filterbank is multiplied with the power spectra and the energy in each filter is summed (as filterbank energies). This step warps the frequency axis of the spectrogram representation [18]. The resultant representation is then compressed using log operation and is called as log mel spectrogram representation. Figure 1.2 shows the raw speech signal and the corresponding log mel spectrogram representation.

There are several speech representations (features) useful for speech processing. Some of them include modified group delay features [73], linear prediction (LP) residual based features [113], phase based features [95], zero-crossings [50], pitch based features [63], etc. However, these features are not considered in discussion in this thesis. As the log-mel spectrogram is the most common feature used in deep learning based automatic speech recognition (ASR) and for many audio classification tasks, the scope of the discussion in this thesis is limited to spectrogram-like representation and their derivatives.

1.3.2 *Modulation filtered representation*

The log mel spectrogram representation contains spectral energies at all time frames. For a particular frequency ω_k, $\log(|S(\omega_k, t)|^2)$ represents a time domain function of the evolution of spectral energy. The rate of change of signal energy across time in a frequency sub-band is called as rate (temporal modulations in Hz) in the signal. Correspondingly, for a particular time frame t_j, $\log(|S(\omega, t_j)|^2)$ represents a frequency domain function of the evolution of energy. The rate of change of signal energy across frequency sub-bands in a time frame is termed as scale (spectral modulations in cycles/octave) in the signal. The rate and scale characterize different information present in the speech/audio signal. These definitions are highlighted in Fig. 1.2.

The spectrogram representation of a typical speech utterance is rich in temporal and frequency patterns, with fluctuations of energy across both time and frequency. These energy fluctuations, referred to as modulations, characterize several important cues associated with different sound percepts. Slow temporal modulations ($<$ 10 Hz) are commensurate with the syllable rate in speech, while intermediate and fast modulation rates ($>$ 10 Hz) capture segmental transitions like onsets and offsets [75]. Similarly, slow/broad spectral modulations ($<$ 1 cycles/octave) capture primarily the overall spectral profile and formants, while fast/narrow modulation scales ($>$ 1 cycles/octave) reflect spectral details such as harmonics and sub-harmonic structure of the spectrum. In gerenal, natural sounds are low-pass, showing most of their modulation energy for low temporal and spectral modulations [100]. Animal vocalizations and human speech are characterized by most of the spectral modulation power being found only for low temporal modulation [100].

The spectral and temporal modulation content of the spectrogram can be estimated and filtered to further process the spectrogram representations. It can be done computationally via a bank of modulation-selective filters, termed as modulation filters [14]. The process of filtering the spectrograms (trajectories or 2-D patch) using modulation filters (rate/scale) is referred to as modulation filtering. A Fourier transform of the terms $\log(|S(\omega_k, t)|^2)$ - rate domain, or $\log(|S(\omega, t_j)|^2)$ - scale domain, or $\log(|S(\omega, t)|^2)$ - joint rate-scale domain function can yield the modulation spectrum of speech.

1.3.3 *Past Works on Feature Engineering Representations*

Here, we review some of the popular feature engineering works and their applications.

Patterson et. al., 1987 - Gammatone filterbank [80, 96]
This is a popular work that develops a filterbank based design of speech front-end processing. It describes the development of an auditory filterbank to perform the initial frequency analysis based on models of human hearing and speech perception. It is based on the gammatone function found in physiology studies that summarize measurements of the impulse response of the auditory filter in small mammals. The work in [96] introduced an acoustic feature set based on the Gammatone filterbank for large-vocabulary speech recognition. The authors showed competitive results compared to mel filterbank based features for ASR.

Hermansky et. al., 1994 - RASTA filtering [33]
One of the earliest use of temporal modulations was the RASTA filtering approach [33] . By means of band-pass filtering, the authors show that the modulations relevant to the speech signal can alone be preserved and those pertaining to the channel artifacts can be removed. This is particularly useful for ASR in mis-matched channel conditions, where the channel effects are convolutive in the signal domain and appear as additive component in the log-spectral domain. Modulation processing for RASTA is done on the short-term sub-band energy representations and the processing shows considerable noise robustness in the noisy ASR task. This approached was followed by TRAPS [34] and HATS [13].

Mesgarani et. al., 2006 - Discrimination of speech from non-speech [67]
Here, the authors describe a audio classification algorithm based on multiscale spectro-temporal modulation features. The task explored is to discriminate speech from non-speech consisting of animal vocalizations, music, and environmental sounds. The model captures basic processes occurring from the early cochlear stages to the central cortical areas and generates a multidimensional spectro-temporal representation of the sound, which is then processed and classified by a support vector machine (SVM).

Nemala et. al., 2013 - Multi-stream feature framework [75]
Here, the authors adapt the duality (slow vs. fast: coarse signal dynamics appear to be processed separately from rapidly changing modulations) in a multistream framework for robust phoneme recognition. A multi-path bandpass modulation analysis of speech sounds is proposed with each stream covering an entire range of temporal and spectral modulations. The approach results in substantial improvements for

phoneme recognition in presence of non-stationary noise, reverberation and channel distortions.

Kleinschmidt, 2013 - Gabor filtering [55, 21]
The work proposes a feature extraction scheme for ASR which utilizes two-dimensional spectro-temporal modulation filters. It focuses on the Gabor feature approach, where a feature selection scheme is applied to obtain a suitable set of Gabor-type features for a given task. The optimized feature sets are examined in ASR experiments which report improved robustness. An approach to separable spectro-temporal Gabor filterbank features is proposed in [94].

Thoret et. al., 2017 - Musical Instrument Identification [103]
In this work, modulation spectrum (the two dimensional Fourier transform of a patch of the speech spectrogram captures the spectro-temporal modulation content) has been shown to be a representation that potentially explains the perception of musical instrument sounds. Here, the sounds are processed with filtered spectro-temporal modulations with 2D Gaussian windows. The most relevant regions of this representation for instrument identification were determined for each instrument that are important for their identification. The authors claim that the lower values of spectro-temporal modulations are the most important regions of the modulation spectra for recognizing instruments.

1.4 Physiological and Psycho-acoustical Evidences

The computational auditory model is based on neuro-physiological, biophysical, and psycho-acoustical investigations at various stages of the auditory system. It consists of two major auditory transformations. An early stage models the transformation of the acoustic signal into an internal neural representation (auditory spectrogram). This stage captures monaural processing from the cochlea to the mid-brain. The second stage is a central stage (also called cortical stage) that analyzes the spectrogram to estimate the content of its spectral and temporal modulations. It reflects the more complex spectro-temporal analysis presumed to take place in mammalian auditory cortex. This stage is responsible for extracting the key features for identification and classification of complex sounds.

1.4.1 *Physiological Evidence*

Front-end Cochlear Processing
Various studies have been done on analyzing the front-end cochlear processing in the human auditory system. In humans, the transduction from mechanical vibrations to electrical impulses in neurons occurs in the cochlea [112]. Mechanical vibrations are transmitted into the cochlea via the middle ear and cause the basilar membrane to vibrate. The mechanical properties of the basilar membrane vary along the length of the cochlea. Conceptually, this can be modeled using a bank of bandpass filters with center frequencies and bandwidths that increase logarithmically [32, 117]. These studies have had significant influence in speech based modeling systems.

On the studies including cochlear structure and basilar membrane characteristics, the tonotopic nature of basilar membrane in processing of incoming sounds with non-linear nature of frequency warping motivated the design of Gammatone filterbank [80]. Another group of studies include analytically tractable framework to describe processing in the periphery with a series of transformations [112]. It includes an analysis stage (cochlear filtering as wavelet transform), transduction stage (the fluid-cilia coupling as velocity coupling, the ionic channels as instantaneous nonlinearity, and the membrane potentials as low pass filtering) and reduction stage (lateral inhibitory network as spectral estimation) [112]. Hence, the two-dimensional representation termed as the auditory spectrogram is obtained using an auditory-inspired model of cochlear and midbrain processing.

Cortical Analysis

Several studies have tried to unravel the signal analysis involved in higher levels of auditory processing like the primary auditory cortex [14]. Specifically, much insight about the physiological functions can be gained by the measuring the spectro-temporal receptive fields (STRFs) of the auditory neurons in the cortex of animals and humans. The STRF denotes a two dimensional time-frequency impulse response of a neuron assuming a linear model for the neuron and determines the modulation selectivity of the neuron. In the scope of this thesis, the most relevant aspect of STRFs is the temporal span of these measured responses. Typically, some of these STRFs extend for about 250 ms or more [14] which is about a syllable length in speech signals. If we desire to have a signal analysis scheme which is consistent with these physiological studies, there is a need to process longer context of speech/audio signals than the conventional window of 25 ms.

1.4.2 *Psycho-acoustical Evidences*

Acoustic Filterbank

The design of acoustic filterbank response to derive auditory representations is motivated from human perception. The design of mel filterbank response to warp frequency axis of spectrogram is motivated through these studies [18]. The mel scale was developed through set of psycho-acoustic experiments on humans based on what one hears when a sound reaches his/her ears, along with the systematic relations between stimulus and sensation [101]. The name *mel* comes from the word melody to indicate that this frequency scale is based on perceived pitch comparisons [102].

Modulation Filtering

The importance of various modulation frequencies in speech has been analyzed using a set of psycho-physical experiments [23, 24]. In the first set of experiments, speech envelope in sub-bands (octave bands) is downsampled and filtered using a low-pass rate filter with a variable cut-off frequency [23]. The ratio of filtered envelope to the original envelope is used as a modulation function on the original sub-band signal. Finally, a sub-band re-synthesis is done to obtain a full-band speech signal. The modified speech signal is used for listening experiments on a sentence recognition as well as a phoneme recognition task. By varying the cut-off frequency of the rate filter, the effect of removing the lower and higher modulations is analyzed. The results from these two experiments indicate that most of the speech intelligibility is contained in $1-16$ Hz of temporal modulations with a peak sensitivity at 4 Hz. In order to extract relevant modulation information from a speech signal, the analysis window must be long enough (for example, a window length of 250 ms is needed for representing a 4 Hz modulation component).

In various studies, the sensitivity of the human ear towards slowly varying stimuli explains why human listeners do not seem to pay much attention to a slow change in the frequency characteristics of the communication environment or why the steady background noise does not severely impair human speech communication [33].

1.5 Proposed Deep Representation Learning For Speech and Audio

Based on studies reviewed in previous section, there are several evidences of time-frequency representations and modulation processing. Inspired by these evidences, the representation learning from raw waveform in this thesis is viewed as 2-stage process and has been incorporated in deep framework by performing the following operations in sequential manner. The learning frameworks explored are supervised, unsupervised, or semi-supervised. The proposed deep representation learning framework is shown in Figure 1.3 comprising of:

- **Step-1: Obtaining time-frequency (spectrogram) representation** - The first step involves converting the input raw waveform into "spectrogram-like" representation (short-time spectral energies). This step incorporates the learning of 'acoustic filterbank'. The filterbank can be learnt from raw waveform using convolution operation in time domain or can be learnt in the frequency

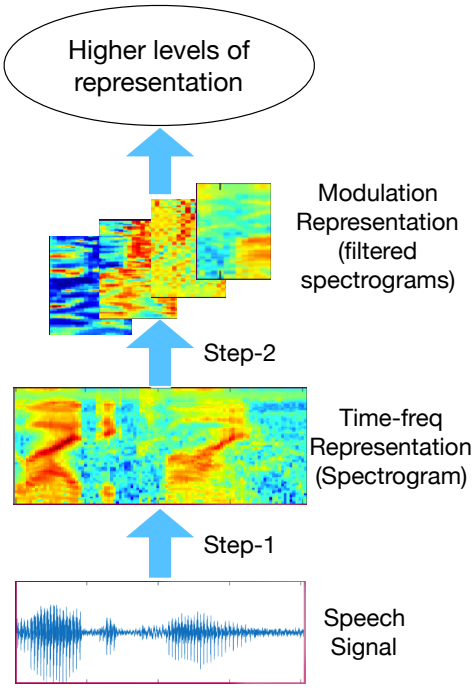


Fig. 1.3: Block schematic of proposed deep representation learning pursued in the thesis.

domain with power spectra as the input.

- **Step 2: Modulation representation** - This step involves filtering the time-frequency representation obtained from previous step using 1-D/2-D modulation filterbank (rate-scale filters). This process is known as learning of modulation filters and the outputs are referred as modulation filtered representations.

1.6 Past Approaches to Speech/Audio Representation Learning

In this section, we discuss the past works on representation learning carried out at both the stages.

1.6.1 *Raw waveform based acoustic filterbank learning*

The broad set of representation learning works can be categorized as supervised and unsupervised. In supervised setting, the main direction pursued has been the learning of the acoustic filterbank parameters from raw waveforms [77, 90, 105]. Compared to the acoustic models using the mel spectrogram features, these models are fed with raw speech signal or power spectra and the first layer of the acoustic model performs a time-frequency decomposition of the signal. The supervised objective function is either a detection or a classification task [90, 39, 91]. Here, we review some of the popular supervised works and their applications.

Palaz et. al., 2013 [77] - The authors proposed the use of convolutional neural networks (CNN) on raw waveform to estimate phoneme class conditional probabilities. The proposed architecture consists of 1 convolutional layer followed by 2 linear layers and the network is trained by maximizing the log likelihood. On TIMIT phoneme recognition task, the authors show the benefit of CNNs with raw input over conventional log mel spectrogram features in a DNN setup.

Sainath et.al., 2013 [90] - Here, the authors used power spectrum as input to the network to learn fil-

terbank for the task of speech recognition. The speech features are produced by multiplying input power spectral features by a set of filters (i.e. first layer weights) and then fed to the CNN based acoustic model. The filterbank (first layer) along with the CNN is learned jointly to minimize cross-entropy loss in a ASR task. The authors used mel filterbank as the initialization of the learnable filters, and show improvements over conventional mel features for the ASR task.

Tüske et. al., 2014 [105] - In this work, the authors use raw speech waveform as the input to DNN with first layer learning the filterbank. The filterbank (the first hidden layer weights) is initialized with auditory-inspired Gammatone filterbank and the DNN is trained for the task of ASR with cross-entropy loss. The analysis presented in the work suggests that, even though the ASR did not improve with filterbank learning over the conventional mel features, the DNN is able to learn a set of band-pass filters in time domain purely from the raw waveform.

Lee et. al., 2017 [60] - Here, the authors propose raw waveform based audio classification for three types of signals - music, speech and acoustic scene sounds. Two types of sample-level deep convolutional neural networks are used that take raw waveforms as input and the filters have fine small granularity. The authors show that their models reach state-of-the-art classification performance for the 3 categories of sound. The visualization of the learnt filters show interesting characteristics for the three categories as well.

For unsupervised learning of acoustic representations, there have been some attempts in the past. Here, we review some of the popular unsupervised learning works and their applications.

Sailor et. al., 2016 [88] - In this work, the authors use convolutional restricted Boltzmann machine (ConvRBM) as a model for learning representations from raw speech signal. The ConvRBM is trained in an unsupervised way to model speech signal of arbitrary lengths. The weights of the model are shown to represent an auditory-like filterbank with non-linear center frequencies. The authors apply the learnt representations for speech recognition task where they show improvements over conventional mel features.

Schneider et. al., 2019 [97] - Here, the authors apply unsupervised pre-training to improve supervised speech recognition. Their model, wav2vec, is a multi-layer convolutional neural network that takes raw audio as input and computes a general representation that can be input to an ASR system. The model is optimized to predict future samples from a given signal context. The objective is a contrastive loss that requires distinguishing a true future audio sample from other samples (noise contrastive binary classification task). The authors show significant improvements in the ASR task with the approach.

Pascual et. al., 2019 [78] - In this work, a self-supervised method is proposed, where a single neural encoder is followed by multiple workers that jointly solve different self-supervised tasks. The authors claim that the joint training of different tasks imposes meaningful constraints on the encoder, contributing to discover general representations and to minimize the risk of learning superficial ones. The authors show that the approach can learn transferable, robust, and problem-agnostic features that extract relevant information from the speech signal.

1.6.2 *Modulation filter learning*

The principle of modulation filtering is based on enhancing perceptually relevant regions of the modulation spectrum (the two dimensional Fourier transform of a patch of the speech spectrogram captures the spectro-temporal modulation content). This is partly inspired by human perceptual studies relating to the importance of temporal modulations (rate) and spectral modulations (scale) [25]. Several works in the past have incorporated the knowledge of spectro-temporal modulation filtering for speech and audio tasks. These approaches define a series of the spectral, temporal, and spectro-temporal modulation filtering operations on the speech spectrogram. We review some of the popular works and their applications here.

Hung and Lee, 2006 - Linear Discriminant Analysis [42]
A supervised data-driven approach using the linear discriminant analysis (LDA) has been attempted for deriving temporal modulation filters. The authors also proposed the use of new optimization criteria of principal component analysis (PCA) and the minimum classification error (MCE) for learning the temporal filters. The paper then reports comparative performance analysis for the features obtained using the three optimization criteria, LDA, PCA, and MCE. The three approaches of deriving temporal modulation filters are shown to improve the noise robustness of speech features used in speech recognition.

Sailor and Patil, 2016 - Convolutional Restricted Boltzmann Machine [89]
In this work, the authors investigate unsupervised representation learning using convolutional restricted Boltzmann machine (ConvRBM) with rectified units for speech recognition task. The temporal modulation representations are learned using log mel spectrogram as an input to ConvRBM. The learnt representations are then used in DNN based ASR setup. The work uses a system combination framework by combining mel filterbank features with modulation features learned by ConvRBM, and report considerable improvements in ASR performance.

Mlynarski et. al., 2018 - Hierarchical Generative Model [71]
Here, the authors designed a hierarchical generative model of natural sounds that learns combinations of spectro-temporal features from natural stimulus statistics. In the first layer, the model forms a sparse convolutional code of spectrograms using a dictionary of learned spectro-temporal kernels. To generalize from specific kernel activation patterns, the second layer encodes patterns of time-varying magnitude of multiple first layer coefficients. When trained on corpora of speech and environmental sounds, some second-layer units learned to group spectro-temporal features that occur together in natural sounds. The authors show that the features trained on a speech corpus were strongly modulated in frequency, while environmental sounds yielded spectro-temporal kernels with faster temporal modulations.

1.6.3 *Interpretable Representation learning*

In many of these previous approaches, the learned representations are assumed to capture distinctive speech features for phonetic discrimination. However, in both supervised and unsupervised framework, the interpretability of the learnt filterbank remain limited, often referred to as "black-box" representations that might make sense for a machine but difficult to interpret by humans.

In the direction of moving towards learning interpretable representations, we review some works with interpretable learning objective.

Seki et. al., 2017 - Gaussian functions [98]
In this work, the authors used Gaussian functions incorporated as first layer of DNN instead of triangular mel-scale filterbanks with input being power spectra to DNN. The means (center frequencies) are initialized as equally spaced values along mel scale, and the bandwidths are set as corresponding bandwidth of the mel filterbank. The means, bandwidth and gain of the Gaussian filters are learnable parameters of the filterbank and are learnt jointly with the rest of the DNN for the task of speech recognition. The authors claim that this parametric technique enables the frequency domain smoothing, and provides improvements for the ASR task.

Zeghidour et. al., 2018 - Gabor wavelets [115]
Here, the authors train a bank of complex filters that operates on the raw waveform and is fed into a convolutional neural network for end-to-end phone recognition. The kernels (filters) of convolutional layer are initialized with Gabor wavelets and its parameters are initialized as an approximation of mel filterbank, and all the filter coefficients of the FIR filters are learned. The authors claim that these time-domain learnable filterbanks outperform conventional mel filterbank features.

Ravanelli et. al., 2018 - SincNet filterbank [84]
The authors use Sinc filters as parametric acoustic filters in the first convolutional layer, with only low and high cutoff frequencies of band-pass filters to be directly learned from data, and filters are learned in supervised manner for the task of speech recognition. They showed that the filter learning is resilient to the presence of band-limited noise.

The authors extended the work in [78] to learn filterbank in a self-supervised framework as problem agnostic speech encoder (PASE). Here, a single neural encoder is followed by multiple workers that jointly solve different self-supervised tasks. The authors claim that the needed consensus across different tasks naturally imposes meaningful constraints to the encoder, contributing to discover general representations and to minimize the risk of learning superficial ones. The experiments and analysis shown in the work reveal that their approach can learn transferable, robust, and problem-agnostic features that carry on relevant information from the speech signal, such as speaker identity, phonemes, and even higher-level features such as emotional cues.

1.6.4 *Comments on Past Methodologies*

Although there have been several attempts on learning spectrogram representation from raw waveform (stage-1 of deep representation learning), compared to vector representations of text which have shown to embed meaningful semantic properties, the interpretability of speech representations has been limited. In addition, there are apparently no studies that analyze the weighting (selectivity) of sub-band representations as all sub-bands may not be equally important for every input speech frame for a task.

For the works with data-driven modulation filtering (stage-2 of deep representation learning), the learning of modulation filters in irredundant manner remains unexplored. Typically, when many kernels in convolutional layer of deep architecture are learnt, they tend to form overcomplete set of representations. In addition, there has been limited attempts on interpretability of the learnt modulation (rate-scale) characteristics through deep networks. The analysis of the weighting (selectivity) of the modulation representations can be explored as all rate-scale modulations may not be equally important for every input speech/audio frame for a task.

In terms of deep representation learning for speech and audio signals, a generic common framework that can be deployed to learn interpretable representations from speech and audio signals has not been explored in the past works.

1.7 This Thesis - Outline of Contributions

The thesis is focused on developing neural methods for representation learning of speech and audio signals, with the goal of improving downstream applications that rely on these representations. The outline is shown schematically in Figure 1.4 highlighting the major contributions of the thesis.

For representation learning, we pursue two broad directions - supervised and unsupervised. In the case of speech/audio signals, we identify two stages of representation learning. The first stage is the learning of a time-frequency representation (equivalent of spectrogram) from the raw audio waveform. The second stage is the learning of modulation representations (filtering the time-frequency representations along the temporal domain, called rate filtering and spectral domain, called scale filtering). In the first part of the thesis, we propose representation learning methods for speech data in an *unsupervised* manner. Using the modulation representation learning as the goal, we explore various neural architectures for unsupervised learning. These include restricted Boltzmann machines (RBM), variational autoencoders (VAE) and generative adversarial networks (GAN). For learning modulation representations that are distinct and irredundant, we propose different learning frameworks like external residual approach, skip connection based approach, and a modified cost function based approach. The methods developed for rate and scale representation learning are benchmarked using an automatic speech recognition (ASR) task on noisy and reverberant conditions. We

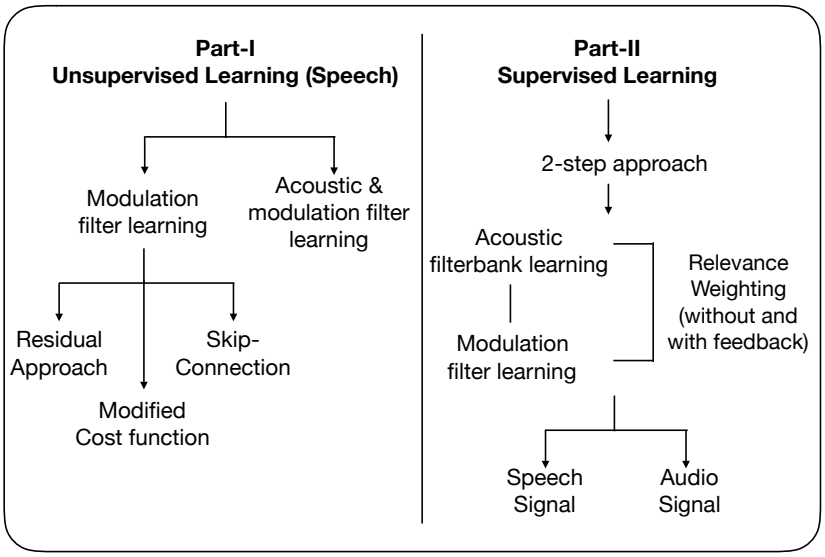


Fig. 1.4: Summary of the thesis contributions.

also illustrate that the unsupervised representation learning can be extended to the first stage of acoustic filterbank learning (time-frequency representation learning) from raw waveforms.

The second part of the thesis deals with *supervised* representation learning. This part is primarily motivated by the need for interpretable representations and explainable deep learning. Here, we propose a two-stage representation learning approach from raw waveform consisting of acoustic filterbank learning from raw waveform followed by a modulation representation learning. This two-stage learning is directly optimized for the task at hand. The key novelty in the proposed framework consists of a relevance weighting mechanism that acts as a feature selection module. This is inspired by gating networks and provides a mechanism to weight the relevance of the acoustic and modulation representations for the task involved. The relevance weighting network can also utilize feedback from the previous predictions of the model for tasks like ASR. The proposed relevance weighting scheme is shown to provide significant performance improvements for ASR task and UrbanSound audio classification task. A detailed analysis yields insights into the interesting properties of the relevance weights that are captured by the model at the acoustic and modulation stages for speech and audio signals.

1.8 Road Map for the Rest of the Thesis

The organization of various chapters in this thesis is shown in Figure 1.5. The rest of the thesis is organized as follows. Chapter 2 sets up the stage for the experiments carried out in the thesis with details of datasets. It also discusses the baseline and other features in comparison.

Chapter 3 dwells into the unsupervised learning part of the work. We begin with modulation filter learning using RBM using the residual approach to learn irredundant filters. This is followed by learning modulation filters using autoencoders (AE) and GANs. The learnt representations are then used for the task of automatic speech recognition (ASR). We then propose modified cost function approach and skip-connection approach to learn multiple irredundant modulation filters in variational framework. The three approaches are compared in terms of learnt filter characteristics and ASR performance from obtained representations. The last part of the chapter extends the unsupervised learning to the raw waveform, where acoustic filterbank is learnt in variational framework to obtain spectrogram representations followed by modulation filter learning.

In Chapter 4, we propose and explore supervised representation learning approaches. The 2-stage (deep) representation learning is carried out for the ASR task. The filterbanks in both the stages are designed

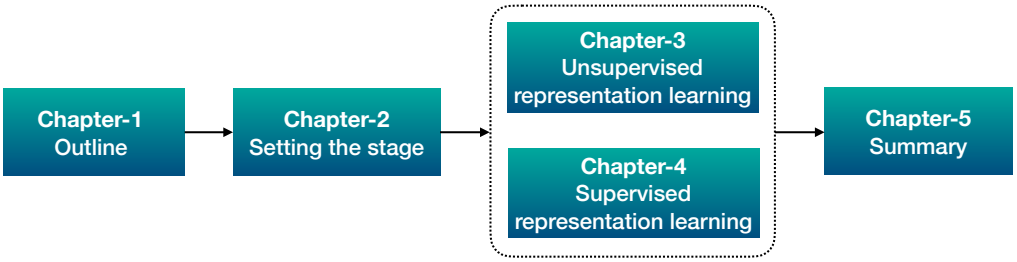


Fig. 1.5: Road map for the thesis chapters.

in parametric fashion for enhanced interpretability. To weigh the sub-bands representations (first stage) and modulation representations (second stage), we propose a relevance weighting scheme at each stage. We then extend the proposed approach to audio sound classification task. The last part of the chapter details about incorporating the target feedback to learn relevance weights.

In Chapter 5, a summary of the various contributions of this thesis is discussed. We also discuss the limitations of the proposed work. This includes determining the applicability of the work for speech/audio systems compared to fixed knowledge-driven representations. This chapter concludes with a discussion on the possible extensions of the deep speech/audio representation learning work as future directions.

1.9 Chapter Summary

The chapter outlined an introduction to the field of representation learning. The discussion is summarized as follows.

- Broad overview of what are representations and the need to learn representations.
- Traditional feature engineering in brief followed by the proposed deep representation learning approach for speech and audio.
- Discussion on physiological and psycho-acoustical evidences towards the need and motivation to learn meaningful representations.
- Past approaches to speech representation learning in the two stages - representation learning from raw waveform and modulation filtering approach.
- Outline of the contributions made in this thesis towards the representation learning of speech and audio.
- Road map for the rest of the thesis.

Chapter 2

Setting the Stage

In this chapter, the background of the experimental setup used in multiple experiments in this thesis are discussed. Section 2.1 reports the details of different features used in the experiments for comparison with the proposed approaches. Section 2.3 discusses the conventional ASR system. The datasets used in the experiments are discussed in Section 2.4.

2.1 Baseline Features

The ASR performance of the proposed approaches is compared with the following features popular for noise robustness:

(1) **Mel FilterBank energies - MFB features**: These are traditional log mel filterbank energy features computed using mel filterbank to warp short-time Fourier transform [18]. The STFT is computed using 25ms window length and 10ms shift. The obtained power spectra is then multiplied with mel filterbank followed by summation inside each sub-band to obtain sub-band energy values. The log compression over these warped sub-band energies is termed as log mel filterbank energy features.

(2) **Power normalized FilterBank energies - PFB features**: These features replace the traditional log nonlinearity of log mel filterbank energy (MFB) features with power-law nonlinearity [49]. Another difference is it uses gammatone filterbank to warp the frequency axis of power spectra instead of mel filterbank. It also involves a noise-suppression algorithm based on asymmetric filtering to estimate the level of the acoustical background noise for each time frame and frequency bin, and a module that accomplishes temporal masking.

(3) **Advanced front-end ETSI feature extraction - ETS features**: ETSI is an advanced front-end feature extraction to create mel-cepstrum parameters [26]. The feature extraction process consists of noise reduction block based on Wiener filter theory. First, the signal spectrum is smoothed along the time (frame) index. Then, frequency domain Wiener filter coefficients are calculated by using both the current frame spectrum estimation and the noise spectrum estimation through voice activity detection (VAD). The linear Wiener filter coefficients are further smoothed along the frequency axis by using a mel filterbank, resulting in a mel-warped frequency domain Wiener filter. The impulse response of this mel-warped Wiener filter is obtained by applying a mel IDCT (mel-warped inverse discrete cosine transform). Finally, the input signal is filtered using the obtained warped impulse response.

(4) **RelAtive SpecTral Amplitude (RASTA) - RAS features**: Relative spectra (RASTA) is method of feature extraction for speech recognition which tries to achieve robustness to channel distortions using principles of modulation spectra [33]. As discussed in Sec. 1.4.2, the important speech information for human perceptual system lies in $1 - 16$ Hz of rate modulations. Some of

the temporal effects introduced by the channel artifacts lie outside this region of the modulation spectrum. By means of band-pass filtering, the modulations relevant to the speech signal can alone be preserved and those pertaining to the channel artifacts can be removed. This is particularly useful in automatic speech recognition (ASR) in mis-matched channel conditions, where the channel effects are convolutive in the signal domain and appear as additive component in the log-spectral domain. Modulation processing for RASTA is done on the short-term sub-band energy representations.

(5) **Linear Discriminant Analysis - LDA features**: These features use LDA for data-driven design of RASTA-like filters [107]. The LDA is applied to rather long segments of time trajectories of features rather than just to a single feature vector or to a relatively short block of feature vectors. This particular application of LDA results in a set of FIR filters in the rate domain. The LDA on time-shifted segments of the trajectories therefore allows an FIR filtering interpretation of the analysis. The frequency responses of the first three discriminant vectors are consistent with the bio-inspired design of RASTA filters [33].

(6) **Gabor filtering - GAB features**: This feature extraction method is based on spectro-temporal Gabor filters with filter selection [56]. The spectro-temporal filtering is performed on log mel filterbank energy features as spectrograms. The approach takes 2-D localized patches from the spectrogram of the speech signal, and creates features for ASR by filtering them using set of Gabor filters followed by feature selection algorithm. The feature selection is based on the rate-scale space spanned by each filter and constructing a filter set according to the resolution of filter parameters such as center frequency and bandwidth.

(7) **Mean Hilbert envelope coefficients (MHEC) - MHE features**: MHEC features are an effective alternative to mel filterbank based features (for robust speaker identification) under noisy and reverberant conditions [87]. These features are based on the Hilbert envelope of Gammatone filterbank outputs. From the filterbank outputs, the temporal envelope of each sub-band output is computed as the squared magnitude of analytical signal obtained using the Hilbert transform. Here, two different compensation strategies are integrated within the feature extraction framework to effectively suppress the reverberation effects. In each sub-band, the smoothed Hilbert envelope is normalized by the long term average computed over the entire utterance, followed by spectral mean subtraction.

2.2 Feature Normalization

All the features used in this thesis (baseline as well as proposed) are normalized using cepstral mean and variance normalization (CMVN) [109, 104]. CMVN uses second order cepstral moment normalization to modify the statistics of noisy speech features closer to that of the clean ones, and hence, CMVN is utilized to decrease the effects of both convolutional and additive distortions and especially to deal with the noise in silence frames [83]. The features (representations) obtained through unsupervised learning framework (Chapter-3) are normalized at utterance level for ASR training using Kaldi tool [82], whereas the representations learnt from raw waveform processing in supervised framework (Chapter-4) are processed with CMVN on a 1sec. running window.

2.3 Automatic Speech Recognition (ASR) System

In this Section, we discuss a conventional ASR system since a large part of experiments in this thesis are on ASR. The task of an ASR system is to convert a speech signal into a sequence of words (or phonemes) in the

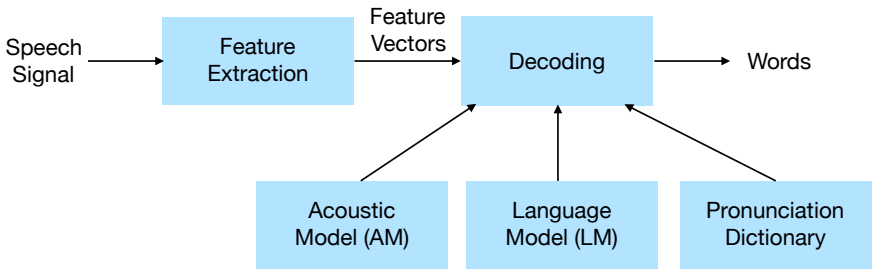


Fig. 2.1: Block schematic of conventional automatic speech recognition (ASR) system [29].

text format. The ultimate goal of the ASR task is to enable people to communicate with the machines in the form of human-computer interaction. There are many potential applications of the ASR that includes call centers, voice dialing, data entry and dictation, command and control, computer-aided language learning, etc. In the recent years, ASR along with text-to-speech synthesis (TTS) is used effectively in chat bots and in mobile phones as an intelligent personal assistant (e.g., Apple Siri, Google Home, Amazon Echo).

The block schematic of a conventional ASR system is shown in Figure 2.1 [29]. Note that we will use the terms *features* and *representations* interchangeably in rest of the thesis. In the block schematic, the input speech signal is first converted into a sequence of feature vectors through feature extraction block. Let $\boldsymbol{X}$ denote the sequence of feature vectors as $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T\}$, where T is number of frames, and $\boldsymbol{Y}$ is the corresponding target word sequence. The goal of the decoder is to find the optimal word sequence $\tilde{\boldsymbol{Y}}$ through the fundamental equations as:

$$\tilde{\boldsymbol{Y}} = \arg\max_{\boldsymbol{Y}} P(\boldsymbol{Y}|\boldsymbol{X}). \tag{2.1}$$

Applying Bayes' rule to the above equation yields

$$\tilde{\boldsymbol{Y}} = \arg\max_{\boldsymbol{Y}} \frac{P(\boldsymbol{Y})P(\boldsymbol{X}|\boldsymbol{Y})}{P(\boldsymbol{X})}, \tag{2.2}$$

$$= \arg\max_{\boldsymbol{Y}} P(\boldsymbol{Y})P(\boldsymbol{X}|\boldsymbol{Y}). \tag{2.3}$$

The likelihood $P(\boldsymbol{X}|\boldsymbol{Y})$ is determined by the acoustic model and the prior $P(\boldsymbol{Y})$ is determined by the language model (LM). In our work, the acoustic model (AM) and language model (LM) are obtained from Kaldi recipe [82]. Note that in second half of the thesis (Chapter-4), raw waveform processing is pursued where 'feature extraction' and 'acoustic model' blocks are merged and learnt jointly.

The role of acoustic model in ASR is to improve the recognition accuracy by learning high-level statistics while combating variations in speakers, dialects, environment, and noise. The Hidden Markov Models (HMM) were the most popular type of statistical model used for acoustic modeling for a long time [58], until deep learning methods took over from last 10 years. Acoustic modeling also includes "pronunciation modeling" that describes how a sequence or multi-sequence of the fundamental speech units (e.g., phones) is used to represent larger speech sound units, such as words or phrases.

Each spoken word w is decomposed into a sequence of N_w basic speech sound units called as base phones, a fixed set of basic sound units for a given language. This sequence is called its pronunciation denoted as $\boldsymbol{q}_w = q_1, q_2, ..., q_{N_w}$. Generally, this pronunciation of words is supplied via the pronunciation dictionary, which contains the phonetic decomposition of words. Multiple pronunciations are allowed by computing the likelihood $P(\boldsymbol{X}|\boldsymbol{Y})$ that can be computed over multiple pronunciations as follows [58]:

$$P(\boldsymbol{X}|\boldsymbol{Y}) = \sum_{\boldsymbol{Q}} P(\boldsymbol{X}|\boldsymbol{Q})P(\boldsymbol{Q}|\boldsymbol{Y}), \tag{2.4}$$

where the summation is over all the valid pronunciation sequences for $\boldsymbol{Y}$, and $\boldsymbol{Q}$ is a particular sequence of pronunciation. Each base phone q is modeled by a continuous density HMM [58]. Given the HMM, $\boldsymbol{Q}$

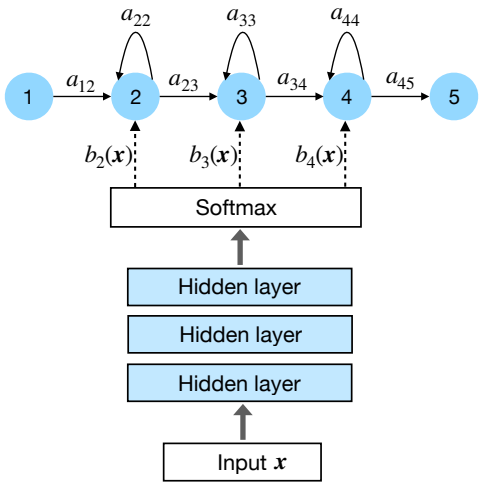


Fig. 2.2: An example of hybrid DNN-HMM approach for automatic speech recognition (ASR) system [16].

is formed by concatenating all of the base phones $\boldsymbol{Q} = \boldsymbol{q}_{w_1}, \boldsymbol{q}_{w_2}, ..., \boldsymbol{q}_{w_L}$. Then, the acoustic model is given by:

$$P(\boldsymbol{X}|\boldsymbol{Q}) = \sum_{\boldsymbol{s}} P(\boldsymbol{X}, \boldsymbol{s}|\boldsymbol{Q}), \tag{2.5}$$

where $\boldsymbol{s}$ is a state sequence through the composite HMM model [41]. Hence, the speech features are modeled and represented by concatenating a sequence of HMM phone models. However, the context-dependent variations in the speech has not been considered yet. For example, the pronunciation of the vowel /a/ in the words "bat" and "ball" are different. The context-free phone models are referred to as monophones [29]. A simple way to incorporate the context in the phones is to use a unique HMM phone model for every possible pair of left and right neighbors of the phones. The resulting HMM models are called as triphones [29].

In our work, we use hybrid DNN-HMM approach to build ASR system with triphone HMM states (called senones). Here, the class labels (triphone HMM states called as senones) are produced by force-alignment in the GMM-HMM training [82].

2.3.1 *Acoustic Model - Hybrid DNN-HMM approach*

The state-of-the-art ASR systems are based on using DNN for the acoustic modeling and HMM for the sequence modeling and decoding. Such an approach is known as the hybrid DNN-HMM approach [72, 11]. In the conventional ASR, the likelihood probabilities $P(\boldsymbol{X}|\boldsymbol{Y})$ are estimated using the GMM-HMM from the acoustic feature vectors. The DNN can estimate posterior probabilities that are related to the emission probabilities and, hence, can be easily integrated with an HMM-based approach [11]. Hence, instead of the GMM, the DNN provides the emission probabilities. In particular, the DNNs can be trained to produce the posterior probability $P(\boldsymbol{s}|\boldsymbol{X})$, i.e., the posterior probability of the HMM state sequence $(\boldsymbol{s} = s_1, s_2, ..., s_T)$ given the acoustic feature vectors $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T\}$. This is done by setting the DNN targets as states of the HMM. The DNN outputs are converted to the emission probabilities (scaled likelihoods) using Bayes' rule [11]:

$$\frac{P(\boldsymbol{X}|\boldsymbol{s})}{P(\boldsymbol{X})} = \frac{P(\boldsymbol{s}|\boldsymbol{X})}{P(\boldsymbol{s})}, \tag{2.6}$$

where $P(\boldsymbol{s}|\boldsymbol{X})$ is the posterior estimated using DNN (DNN output), $P(\boldsymbol{s})$ is the class prior (the relative frequencies of each class as determined from the class labels (triphone HMM states as senones) that are produced by a forced-alignment in the GMM-HMM training), and scaled likelihood $P(\boldsymbol{X}|\boldsymbol{s})$ is used as an

emission probability for the HMM. An example of the hybrid DNN-HMM approach is shown in Figure 2.2 for a 3-layer DNN. The transition probability from state s_i to state s_j is denoted as a_{ij} and the emission probability density $b_j(\boldsymbol{x})$ describes the distribution of the observation vector $\boldsymbol{x}$ at the state s_j. In our work, we use triphone HMM model for acoustic modeling using standard Kaldi recipe for different datasets [82].

Most of the ASR experiments in this thesis (Chapter-3) use a fully-connected DNN in the hybrid DNN-HMM for ASR and triphone HMM states as senones.

2.3.2 *Language Modeling (LM) and Decoding*

The role of the language model in ASR is to provide the value $P(\boldsymbol{Y})$ in the fundamental equation of the ASR (Eq. 2.1). The probabilistic relationship between a sequence of words can be derived and modeled from a text corpus with a large number of words. These probabilistic models are called stochastic language models or N-grams. A language model can be formulated as a probability distribution $P(\boldsymbol{Y})$ over a word string $\boldsymbol{Y}$ that reflects how frequently a string $\boldsymbol{Y}$ occurs as a sentence [41]. The $P(\boldsymbol{Y})$ can be decomposed as:

$$P(\boldsymbol{Y}) = P(y_1, y_2, ..., y_N), \tag{2.7}$$

$$= \prod_{i=1}^{N} P(y_i|y_1, y_2, ..., y_{i-1}), \tag{2.8}$$

where $P(y_i|y_1, y_2, ..., y_{i-1})$ is the probability that y_i will follow given the previous word sequence, $y_1, y_2, ..., y_{i-1}$. For a large vocabulary continuous speech recognition (LVCSR) task, the word history is truncated to $N-1$ words due to computational issues, which leads to an N-gram language model. If the current word depends on the previous word, we have the bi-gram model $P(y_i|y_{i-1})$, and if the word depends on two previous words, we have a tri-gram model $P(y_i|y_{i-2}, y_{i-1})$. The probabilities in the N-gram model are estimated from the training text corpus by counting N-gram occurrences to form the maximum likelihood estimates [58]. In our work, we have used the trigram- based LM (using the standard Kaldi recipe) for all the databases for decoding [82]. We have also used recurrent neural network based language model (RNN-LM) for some experiments [69].

2.4 Databases

Here we discuss the details of different databases used for experiments in the thesis. Table 2.1 summarizes the dataset details in brief.

2.4.1 *Clean and Noisy speech - WSJ Aurora-4*

The WSJ Aurora-4 corpus is used for clean and noisy speech recognition experiments [81]. A total of 14 hours of continuous read speech recordings sampled at 16 kHz is available separately for clean and noisy training, with noisy speech corrupted with one of six different noises (street, train, car, babble, restaurant, and airport) at $10-20$ dB SNR. The noisy data has additive noises and linear convolutional channel distortions, which were artificially synthesized to corrupt clean speech recorded with close-talking microphone. The clean recordings were carried out with two microphones The training data has two sets of 7138 clean and multi condition recordings (14 hours each), respectively (84 speakers). The average train utterance duration is 7.6 secs. Similar to the training data, the same types of noise and microphones are used to generate the validation and test set. The validation data has two sets of 1206 clean and multi condition recordings, respectively (14 speakers) and test data has 330 recordings (8 speakers) for 14 clean and noise conditions. The average utterance length of validation data is 6.7 sec. (total 2.2hrs of data) and test set is 7.3 sec. (total 9.4hrs of data for all 14 conditions). The ASR results are reported for average of 14 test conditions, classified into four groups as: A - clean data, B - noisy data, C - clean data with channel

distortion, and D - noisy data with channel distortion. The dataset has 5000 word vocabulary size with no out of vocabulary words (OOVs) in the evaluation set (referred to as the 5k closed vocabulary task).

2.4.2 *Noisy + Reverberant speech - REVERB Challenge*

The ASR experiments on reverberated speech data are performed using WSJCAM0 corpus, released as a part of REVERB challenge [53]. This database consists of 7861 recordings sampled at 16kHz from 92 training speakers (total 15.5 hours), 1488 recordings from 20 development test (dt) speakers (total 2.8 hrs), and 2178 recordings from two sets of 14 evaluation test (et) speakers, with each speaker providing about 90 utterances (total 4.2 hrs). The clean recordings were carried out with two sets of microphone- head mounted (close-talk single channel) as well as desk microphone positioned about half meter from the speaker's head (an 8 channel circular array microphone with a diameter of 20 cm).

The database consists of training data set (Train) for clean and multi condition reverb training respectively. The multi-condition data consists of simulated reverb training data, a simulated (Sim) test dataset and a naturally reverberant (Real) recording of the test dataset. The simulated (Sim) reverb utterances are artificially synthesized by convolving the clean utterances with 24 measured room impulse responses and adding background noise at an SNR of 20 dB. The reverberation times of the measured impulse responses for this dataset range roughly from 0.2 to 0.8 s. Different recording rooms were used for the Dev set, the Eval set, and the training data. In our work, we use single channel and beamformed audio for the ASR experiments. The BeamformIt algorithm based on weighted delay-and-sum technique [6] is used in the work available with Kaldi toolkit [82].

2.4.3 *Noisy + Reverberant speech - CHiME-3 Challenge*

The CHiME-3 corpus with multi-condition training (real+simulated) is used for ASR experiments [8]. The data sampled at 16kHz is recorded with multi-microphone tablet device being used in everyday environments with four varied environments - cafe (CAF), street junction (STR), public transport (BUS) and pedestrian area (PED). For each environment, two types of noisy speech data are present, real and simulated. The real data consists of 6-channel recordings using an array of microphones in rectangular frame. It consists of sentences from the WSJ0 corpus spoken by 6 male and 6 female talkers in the environments listed above. The simulated data was constructed by artificially mixing clean utterances with environment background recordings.

The training data with a total of 17 hours of data has 1600 (real) noisy recordings and 7138 simulated noisy utterances. The development (dev) and evaluation (eval) data set consists of the same 410 and 330 utterances that make up the corresponding sets in the WSJ0 5k task. For each set, the sentences are read by four different talkers in the four CHiME-3 environments. In each environment, the set is split into four random partitions and each is assigned to a different talker. This results in 1640 (410×4) and 1320 (330×4) real development and evaluation utterances in total. Identically-sized, simulated dev and eval sets are made by mixing recordings captured in the recording booth with the environmental noise recordings. This gives total of 6.4 hrs of dev data and 5.1 hrs of eval data. We use the beamformed audio of train and test data in the work for experiments. The beamforming is performed using BeamformIt algorithm (as part of challenge baseline) in Kaldi toolbox [6, 82].

2.4.4 *Noisy + Reverberant speech - VOiCES Challenge*

The Voices Obscured in Complex Environmental Settings (VOiCES) corpus is a creative commons speech dataset [85], being used as part of VOiCES Challenge [74]. The ASR training data under 'fixed' training condition track contains a subset of Librispeech clean training data in this challenge. The training data set of 80 hours has $22,741$ utterances sampled at 16kHz from 202 speakers, with each utterance having $12-15$s segments of read speech from the Librispeech corpus. This subset was designed in such a way as to have no overlap in speakers with the VOiCES corpus (development or evaluation). The ASR development

Table 2.1: Brief summary of databases used in the work.

Dataset	Train,Test (hrs)	Samp. rate (kHz)	#spk/sources Train, Test	Style	Type of noise	Task	Section
Aurora-4 [81]	14, 9.5	16	84, 8	read	artificial	ASR	3.2 - 3.7, 4.2, 4.4
REVERB [53]	15.5, 4.2	16	92, 14	read	artificial	ASR	3.2 - 3.5
CHiME-3 [8]	17, 5.1	16	76, 4	read	nat. & art.	ASR	3.5 - 3.7, 4.2, 4.4
VOiCES [74]	80, 20	16	202, 200	converse	artificial	ASR	4.2, 4.4
UrbanSound8K [92]	8.7,0.87	44.1	10, 10	natural	natural	USC	4.3
TIMIT [30]	0.8	16	168	read	artificial	Analysis	4.2, 4.4

set consists of 20 hours of distant recordings from the 200 VOiCES dev speakers. It contains recordings from 6 microphones. The evaluation set consists of 20 hours of distant recordings from the 100 VOiCES eval speakers and contains recordings from 10 microphones from an unseen room. We performed a $1-$fold reverberation and noise augmentation of the training data using Kaldi [82] for ASR experiments.

2.4.5 *Urban Sounds - UrbanSound8K*

The UrbanSound8K dataset [92] contains 8732 sound clips (excerpts) sampled at 44.1kHz. The sound clips are of up to 4s in duration (total 8.7 hours), extracted from field recordings crawled from the Freesound online archive. Each slice contains one of 10 possible sound sources: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music. The samples in UrbanSound8K come pre-sorted into 10 folds using a stratified approach which ensures that samples from the same recording are not used both for training and testing. For every urban sound classification (USC) experiment, we run a $10-$fold cross validation using the provided stratified folds.

2.4.6 *TIMIT dataset*

The TIMIT corpus of read speech is small vocabulary dataset designed to provide speech data for acoustic-phonetic studies [30]. The dataset is designed to be phonetically rich dataset with hand-labelled phonetic and word transcriptions available for each utterance. In our work, we use the clean and noisy version of test set of TIMIT consisting of 1344 utterances (total 0.8 hours) from 168 speakers. The noisy version has each utterance artificially corrupted with 4 different noise types, namely babble, exhall, restaurant and subway, at 5 different SNR levels, 0, 5, 10, 15 and 20 dB SNR, respectively. In this thesis, we use TIMIT dataset for analysis and not for any model training.

2.5 Chapter Summary

This chapter discussed the different essentials of the experiments commonly used across the experiments in the work.

- The popular features proposed in the past are detailed and these are used in the experiments to compare with the proposed work.
- The conventional ASR system is discussed in brief with discussion on different components such as acoustic model, language model and decoding.
- The databases used for the speech and audio experiments are discussed.

Chapter 3

Unsupervised Learning of Representations

3.1 Introduction

In this chapter, we propose the learning of representations from the speech data in unsupervised manner, followed by the analysis and comparison of various unsupervised data-driven approaches for representation learning. In particular, the representations are learnt with two approaches: modulation filtering and representation learning from raw waveform. In modulation filtering, the temporal and spectral modulation filters are learnt in unsupervised framework to filter speech spectrograms and obtain speech representations for robust automatic speech recognition (ASR) system. The representation learning from raw waveform is carried out by learning acoustic filterbank in unsupervised manner to obtain spectrogram representations. The framework of unsupervised learning can be divided into distribution learning, representation learning or clustering methods.

A distribution learning method for unsupervised modeling is proposed using the restricted Boltzmann machine (RBM). The RBM learns a binary hidden layer by maximizing likelihood of Boltzmann distribution. A convolutional RBM (CRBM) incorporates the convolutional operation on input to derive hidden representations [37, 76].

An autoencoder (AE) is a neural network which aims at representation learning at the hidden layers by mapping the input to the output using mean square error cost [38]. A convolutional autoencoder (CAE) incorporates convolutional layers in an AE [59, 65]. The traditional CAE consists of an encoder neural network which operates on the input data to derive a bottleneck representation through convolutional operators [65]. The decoder then reconstructs the original data back from the bottleneck representation.

Another approach for representation learning using conditional generative adversarial network (cGAN) attempts to modify the CAE approach with an additional adversarial cost function [31]. Here, a second network is trained in parallel to provide feedback to CAE about matching the distribution of generated ones with the input.

A new approach to learn spectral and temporal modulation filters purely from a variational generative modeling perspective is also proposed. In particular, a filter learning method using the speech spectrogram in conjunction with a two-dimensional (2-D) convolutional variational autoencoder (CVAE) is developed [52]. The encoder learns the distribution of the latent representation and the decoder attempts to reconstruct the original data back from a sample of the latent representation generated from the encoder distribution. Thus, the CVAE functions as a generative model of the input data.

All the models contain an initial convolution layer which learns the desired filters used in ASR. The kernel of the first convolutional layer of these models are interpreted as the modulation filter, that captures modulations derived from large amount of unsupervised speech spectrogram data. Different approaches are used to learn multiple irredundant filters:

- Residual approach: The projection of the input spectrogram on the learnt filter is removed and the residual spectrogram is then used in the same model framework for learning subsequent filters. Here, one filter is learnt at a time.
- Skip connection based residual: The projection of the input spectrogram on the learnt filter is

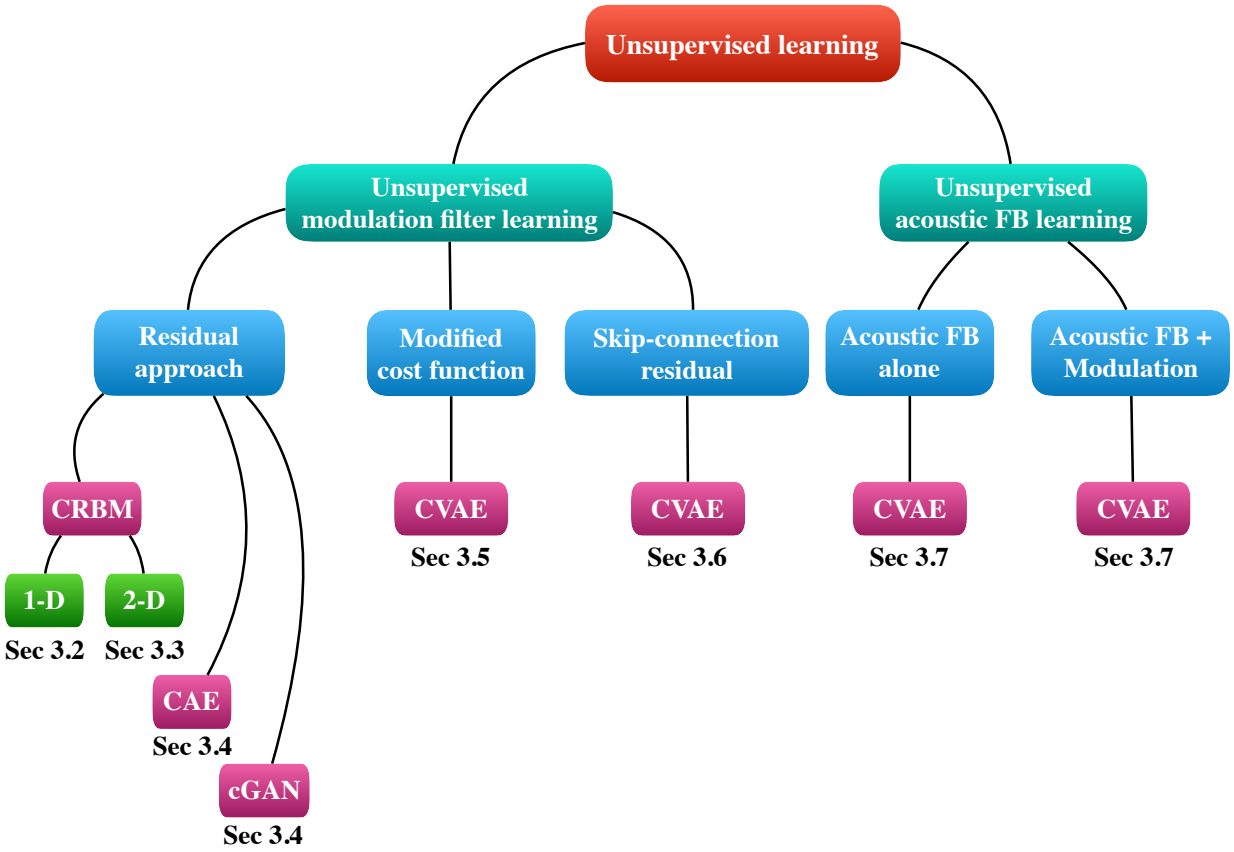


Fig. 3.1: Flowchart of the unsupervised representation learning.

removed using skip connection and the residual spectrogram is then fed to the next layer for learning subsequent filters. Hence, multiple filters are learnt jointly through residual approach.

- Modified cost function: The generative model CVAE is trained with modified loss function to avoid learning redundant filters (and learn non-overlapping filters). Hence, this approach allows learning of multiple filters jointly.

Figure 3.1 shows the flowchart of the unsupervised representation learning approaches explored in this chapter. The CVAE is also used to learn acoustic filterbank (FB) from raw waveform to obtain time-frequency representation. In all these approaches, no prior knowledge of the perceptual filtering studies in auditory processing is applied in filter learning and the data is used to learn the key characteristics (temporal and spectral modulation content as well as acoustic filterbank's parameters). The learnt filters are then used to obtain speech representations which are then used as features for ASR.

The ASR experiments are performed on multiple datasets with noisy and reverb characteristics using a deep neural network (DNN) acoustic model. The results from the experiments indicate that the features derived from the learnt filters provide significant improvements over other noise robust front-ends. Further, the performance of the filtered features in a semi-supervised setting is investigated where availability of labeled data is limited.

In this chapter, the fundamental question of interest is on how to learn representations in unsupervised learning paradigm. In particular, instead of using the standard CNN layer (typically with tens of kernels resulting to many feature maps), we aim to learn modulation representations that are distinct and non-overlapping (with 1-D/2-D kernels). The reason for learning reduced number of kernels is motivated from the fact that without the labels, the problem is ill-constrained and the representations may not be useful for downstream recognition tasks. Hence, to encourage the filters to be non-overlapping, we propose different

learning frameworks like external residual approach (learning 1 kernel at a time and removing the learnt contribution from the total, and learning subsequent kernels), skip-connection based approach (where the residual is computed inside the network itself and the subsequent filters are learnt jointly), and a modified cost function-based approach (which encourages the filters modulation spectra to be orthogonal).

The rest of the chapter is organized as follows in unsupervised learning.

- Section 3.2 describes the convolutional RBM model for learning temporal and spectral modulation filters, followed by residual approach of multiple filter learning criteria and ASR experiments.
- Section 3.3 explores the learning of joint 2-D spectro-temporal modulation filters with rank constraints using CRBM.
- Section 3.4 shows the comparison of CRBM with other models (autoencoder and cGAN) for rate filter learning.
- Section 3.5 describes the learning of Rank-1 2-D spectro-temporal filters through modified cost function in training of CVAE.
- Section 3.6 explores another approach of filter learning through skip-connection based residual approach proposed to learn multiple irredundant filters using CVAE.
- Section 3.7 describes the representation learning from raw waveform using CVAE to learn acoustic filterbank followed by modulation filters.
- Section 3.8 summarizes the chapter.

3.2 Restricted Boltzmann Machine (RBM)

The proposed feature extraction scheme consists of three stages - learning a modulation filter using convolutional RBM architecture, learning multiple irredundant filters and filter selection, and feature extraction for ASR.

3.2.1 *Convolutional Restricted Boltzmann Machine*

The block schematic of the proposed modulation filter learning scheme from speech spectrogram is shown in Figure 3.2 (b). Here, the speech spectrogram is processed with a CRBM architecture (shown in Figure 3.2 (a) for deriving one rate and scale filter). For a 1-D rate filter ($\boldsymbol{w}_R$) learning, the input ($\boldsymbol{v}_R$) consists of temporal energy trajectories of individual subbands for 1.5 sec of speech from training dataset (each of dimension $1 \times N_{V_R}$, with $N_{V_R} = 150$). For 1-D scale filter ($\boldsymbol{w}_S$) learning, the input ($\boldsymbol{v}_S$) consists of all-band energy trajectories of individual speech frames (each of dimension $N_{V_S} \times 1$, $N_{V_S} = 40$ for mel spectrogram). The visible layer and the hidden layer have bias a_R and b_R (a_S and b_S) for rate (scale) filter learning, respectively. The conditional distributions used to perform block Gibbs sampling for rate filtering (similar relations hold for scale filtering also) are:

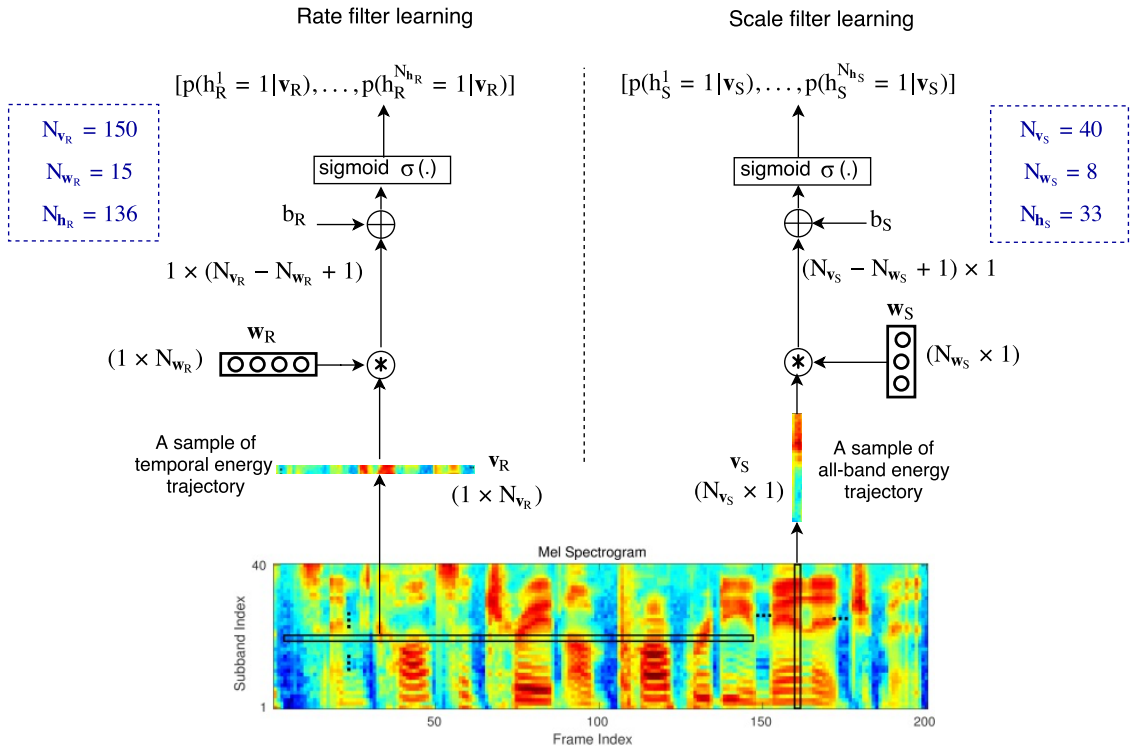
$$P(h_R^j = 1|\boldsymbol{v}_R) = \sigma((\boldsymbol{w}_R \star \boldsymbol{v}_R)_j + b_R) \tag{3.1}$$

$$P(v_R^i = 1|\boldsymbol{h}_R) = \sigma((\boldsymbol{w}_S \star \boldsymbol{h}_R)_i + a_R) \tag{3.2}$$

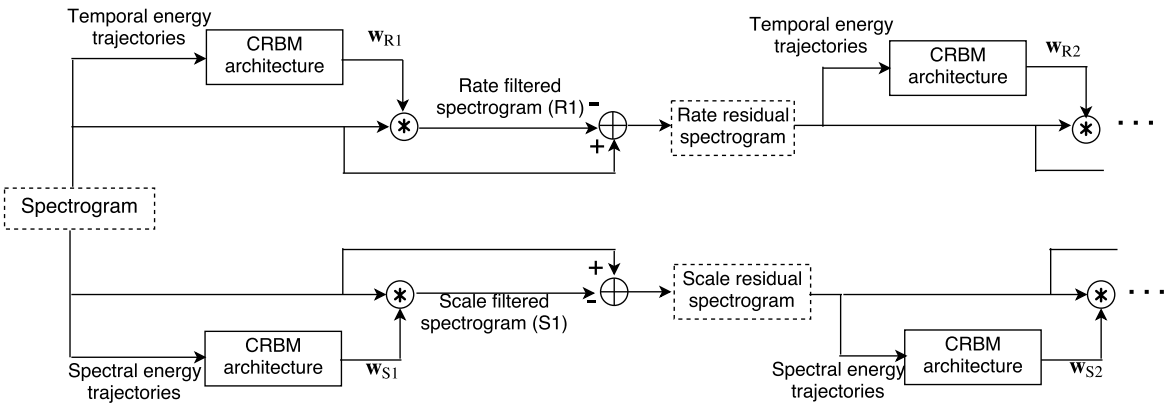
where σ is the sigmoid function, i and j are the index of the visible and hidden layer unit, and $\star$ denotes 1-D convolution. The CRBM training is performed with random initialization of weight vector $\boldsymbol{w}_R$ ($\boldsymbol{w}_S$) and several iterative steps are performed to learn the rate (scale) filter.

3.2.2 *Learning Multiple Irredundant Filters and Filter Selection*

Once a rate (scale) filter is derived, the filtered component of the input spectrogram is removed from the input spectrogram and the residual spectrogram is fed back to CRBM for learning subsequent rate (scale) filters, as shown in Figure 3.2 (b). This method, similar to matching pursuit (MP) algorithm [64], allows us to learn irredundant set of filters. In this work, 3 filters are successively learned from CRBM. The learned



(a) CRBM architecture to learn a rate and scale filter from mel spectrogram



(b) Learning multiple filters using residual approach

Fig. 3.2: The top panel (a) shows the CRBM architecture used for learning a single rate ($\boldsymbol{w}_R$) and a scale ($\boldsymbol{w}_S$) filter separately from the spectrogram (forward pass of CRBM). The bottom panel (b) shows the proposed schematic for learning multiple rate and scale filters.

rate (scale) filters are denoted by $\boldsymbol{w}_{R1}$, $\boldsymbol{w}_{R2}$, $\boldsymbol{w}_{R3}$ ($\boldsymbol{w}_{S1}$, $\boldsymbol{w}_{S2}$, $\boldsymbol{w}_{S3}$), respectively. The corresponding filtered spectrograms are denoted by R1, R2, R3 (S1,S2, S3), respectively.

The magnitude response of the data-driven filters obtained from clean Aurora-4 database are shown in Figure 3.3. In addition, a comparison has been made between the CRBM based rate and scale filters with filters learned from principal component analysis (PCA) (obtained from complex 1-D Fourier representation of the corresponding temporal and spectral energy trajectories) [46], convolutive non-negative matrix factorization (CNMF) with the spectrogram inputs [110] and LDA based filters learned on time trajectory

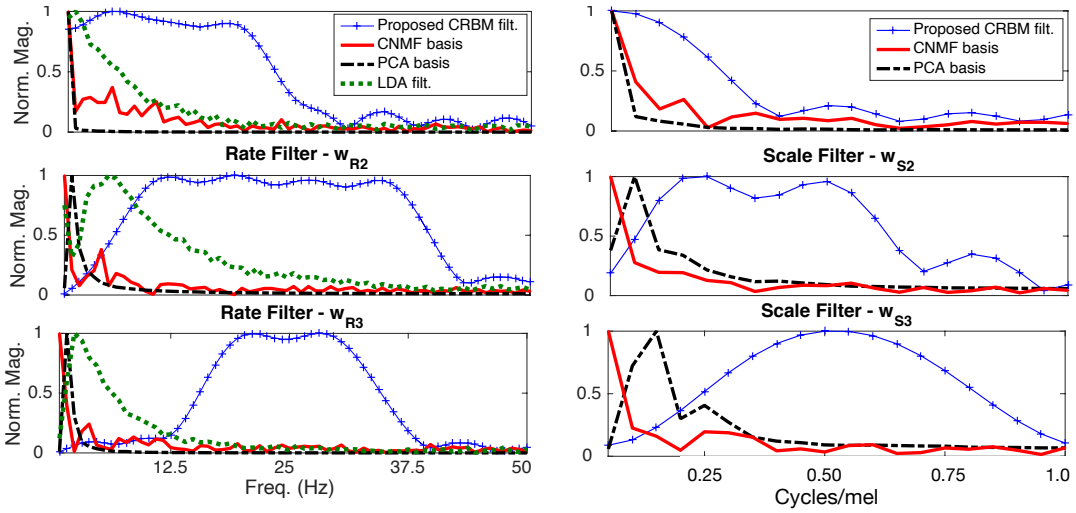


Fig. 3.3: Comparison of magnitude response of the proposed data-driven CRBM filters with the filters obtained from linear discriminant analysis (LDA), complex principle component analysis (PCA) and convolutive non-negative matrix factorization (CNMF). All the filters are derived for mel spectrogram input extracted from Aurora-4 clean training data.

of each subband [107]. As seen here, the proposed CRBM based approach for modulation filter learning provides more smoother filters with broad modulation selectivity similar to those observed in perceptual studies [14, 25]. While a convolutional neural network (CNN) (when used for acoustic modelling) also performs spectro-temporal filtering in ASR, the CNN filters are learnt in a supervised manner (using labelled data). Also, the CNNs typically employed in speech perform local spectro-temporal filtering of $200-300$ ms of temporal context [40]. In the learnt filters here, the filters span entire spectral range for a long temporal context of 1.5 s.

In order to choose the rate and scale filters for ASR, the average hidden activation probability for each filter is computed by a forward pass operation of the input spectrograms through the CRBM. Table 3.1 shows the average hidden activation probability value obtained for clean Aurora-4, multi condition Aurora-4 as well as REVERB challenge database. Based on the highest average activation values from the validation data, the second rate ($\boldsymbol{w}_{R2}$) and first scale filter ($\boldsymbol{w}_{S1}$) is selected to derive R2+S1 features. In ASR using Aurora-4 clean training, it is observed that adding R2+S2 features provided additional improvements along with R2+S1 features. Hence, (R2+S1, R2+S2) features is used for ASR.

3.2.3 *Feature Extraction Overview*

The log-mel spectrogram (MFB) of speech signal is obtained using window length of 25 ms with a shift of 10 ms using 40 mel subband filters between $250-6500$ Hz. The auditory spectrogram (ASp) is obtained using an auditory-inspired model of cochlear processing [14]. The auditory spectrogram is also sampled at 10 ms window shift, with each frame having 113 spectral bands between $250-6500$ Hz. For each of these spectrograms, two streams of joint rate-scale filtered spectrograms are derived (R2+S1 and R2+S2). These spectrogram streams are concatenated and fed to a deep neural network (DNN) based ASR system. The input features are mean-variance normalized at utterance level before DNN training. Figure 3.4 shows the rate-scale filtered spectrogram (R2+S1) on (a) clean test speech file and (b) babble noise test speech file. As seen here, application of modulation filtering can provide more invariant representations compared to conventional mel spectrograms. This may be attributed to the modulation filter characteristics learned from the clean data distribution. For the ASR experiments, the data-driven filters are derived separately for mel spectrogram (MFB) input and the auditory spectrogram (ASp) input and the ASR results are compared.

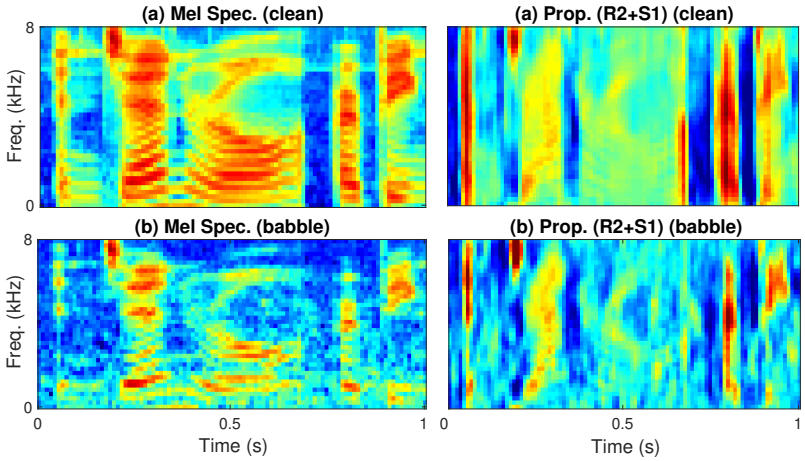


Fig. 3.4: Comparison of mel spectrogram and the data-driven rate-scale filtering of mel spectrogram for (a) clean file (b) babble noise file (different mic.) recorded from a female speaker in Aurora-4 database. The modulation filters with the highest activation probability ($\boldsymbol{w}_{R2} + \boldsymbol{w}_{S1}$) are used in the right side panels to obtain (R2+S1).

Table 3.1: Average hidden activation probability obtained from filtering validation dataset with each of the obtained learned filter individually (averaged over all utterances) in Aurora-4 database on the Mel (MFB) and auditory (ASp) spectrogram and in REVERB database on the Mel (MFB) spectrogram.

Spectrogram	Rate filter			Scale filter		
	$\boldsymbol{w}_{R1}$	$\boldsymbol{w}_{R2}$	$\boldsymbol{w}_{R3}$	$\boldsymbol{w}_{S1}$	$\boldsymbol{w}_{S2}$	$\boldsymbol{w}_{S3}$
Aurora-4: Clean training						
Mel	0.32	**0.38**	0.06	**0.34**	0.23	0.27
Auditory	0.26	**0.30**	0.05	**0.31**	0.26	0.09
Aurora-4: Multi condition training						
Mel	0.28	**0.33**	0.09	**0.35**	0.18	0.26
Auditory	0.23	**0.29**	0.06	**0.30**	0.27	0.08
REVERB training						
Mel	0.30	**0.31**	0.09	**0.39**	0.23	0.22

3.2.4 *Experiments*

3.2.4.1 *Noisy Speech Recognition*

The WSJ Aurora-4 corpus discussed in Section 2.4.1 is used for noisy ASR experiments. The speech recognition Kaldi toolkit is used for building the ASR [82]. A deep belief network- deep neural network (DBN-DNN) with 4 hidden layers having 10 frames of input temporal context and a sigmoid nonlinearity is discriminatively trained using the training data and a tri-gram language model is used in the ASR decoding. The ASR performance of the proposed modulation filtering approach is compared with traditional mel filter bank energy (MFB) features and other features discussed in Section 2.1. The results for the auditory spectrogram features (ASp) [14] is also shown here for reference. The results are reported for average of 14 test data conditions classified into four groups as: A - clean data, B - noisy data, C - clean data with channel distortion, and D - noisy data with channel distortion.

From the ASR performance in clean training condition reported in Table 3.2, it can be observed that PFB and ETS features provide better performance compared to all other baseline features. The data-driven modulation filtering approach on MFB and ASp by joint application of selected rate and scale filtering

Table 3.2: Word error rate (%) in Aurora-4 database for clean training condition with various feature extraction schemes and the proposed (R2+S1,R2+S2) modulation filtering approach applied on the auditory (ASp) and the Mel (MFB) spectrogram.

Cond	Baseline Feature Type								(R2+S1,R2+S2)	
	MFB	ASp	PFB	ETS	RAS	LDA	MHE	GAB	MFB	ASp
A	3.4	3.2	3.3	3.2	3.5	3.7	3.5	**2.6**	3.3	3.3
B	18.9	18.7	16.2	16.3	18.0	20.1	17.4	18.0	13.6	**13.0**
C	15.3	14.4	**11.7**	14.5	16.0	15.9	14.6	**11.7**	13.1	13.1
D	35.2	32.4	32.8	32.0	35.6	36.3	35.4	35.0	29.0	**28.9**
Avg.	24.7	24.0	22.1	21.9	24.4	25.6	23.9	23.8	19.4	**19.1**

Table 3.3: Word error rate (%) in Aurora-4 database for multi condition training condition with various feature extraction schemes and the proposed (R2+S1,R2+S2) modulation filtering approach applied on the auditory (ASp) and the Mel (MFB) spectrogram.

Cond	Baseline Feature Type								(R2+S1,R2+S2)	
	MFB	ASp	PFB	ETS	RAS	LDA	MHE	GAB	MFB	ASp
A	4.2	4.6	4.1	4.5	4.6	4.7	4.0	**3.3**	3.7	4.0
B	7.8	9.7	8.0	8.6	8.5	9.9	8.3	7.4	**7.3**	7.8
C	8.4	10.1	7.8	8.0	9.7	10.0	8.1	**6.1**	7.1	7.8
D	18.5	20.6	19.7	18.8	19.1	21.2	19.6	17.3	**16.2**	17.5
Avg.	12.1	14.0	12.7	12.6	12.8	14.4	12.8	11.2	**10.8**	11.7

(R2+S1,R2+S2) provides significant robustness to noisy and multi-channel test conditions (average relative improvements of 21% over MFB features and 20% over ASp features).

In the matched multi condition training and test scenario in Table 3.3, the GAB features perform better than all other baseline features. We hypothesize that the filter/feature selection criteria in our proposed (data-driven) and the Gabor features (engineered) give them a boost over other methods in most of the test conditions. The best performance is provided by the joint application of selected rate and scale filtering for all the clean and noisy test conditions, improving the baseline MFB results on average by about 11% and improving the ASp results by about 16%. The obtained results with clean and multi condition training also improve over the results obtained from a CNN based ASR system using mel spectrogram [40].

To observe the impact of rate or scale filtering alone on ASR performance, the ASR results with selected rate and scale filters separately is reported in Table 3.4. In clean training condition, the data-driven rate filtering (using R2) gives a average relative improvements of 16% over MFB and 18% over ASp and the scale filtering (concatenation of S1,S2 filtered spectrograms) also provides moderate improvements. In multi-condition training, the rate filter on MFB improves the performance compared to the baseline features (average relative improvements of 9% over MFB).

3.2.4.2 *Reverberant Speech Recognition*

The ASR experiments on reverberated speech data are performed using REVERB challenge dataset discussed in Section 2.4.2 [53]. The rate and scale filters are learnt from mel spectrogram of Train dataset - separately for both clean and multi condition. Table 3.5 shows the ASR performance for clean and multi-condition training conditions using MFB, other features discussed in Section 2.1 and the proposed modulation filtering (R2+S1, R2+S2) applied on MFB.

It can be observed that the proposed features perform better than all the other baseline features under all test conditions with clean and reverb training data. For the clean training, there is an average relative improvement of 24% over MFB features on Sim test data and about 8% with Real test data. For the multi

Table 3.4: Word error rate (%) in Aurora-4 database for clean and multi condition training condition with separate rate and scale filtering applied on the auditory (ASp) and the Mel (MFB) spectrogram.

Cond	Clean Training						Multi condition training					
	Baseline		Rate(R2)		Scale (S1,S2)		Baseline		Rate(R2)		Scale (S1,S2)	
	MFB	ASp	MFB	ASp	MFB	ASp	MFB	ASp	MFB	ASp	MFB	ASp
A	3.4	3.2	2.9	3.2	3.2	3.2	4.2	4.6	3.5	4.1	3.9	4.4
B	18.9	18.7	14.5	13.4	19.0	18.6	7.8	9.7	7.0	8.0	7.9	9.1
C	15.3	14.4	12.7	12.8	14.8	13.8	8.4	10.1	7.7	8.6	7.2	9.0
D	35.2	32.4	31.6	30.3	36.0	33.9	18.5	20.6	16.9	18.4	17.5	19.8
Avg	24.7	24.0	**20.8**	**19.8**	24.8	23.7	12.1	14.0	**11.0**	**12.2**	11.6	13.3

Table 3.5: Word error rate (%) in REVERB Challenge database for clean and multi-condition training with test data from simulated and real reverb environments.

Condition	MFB	PFB	RAS	MHE	R2+S1,R2+S2
Clean training					
Sim_dt	37.2	36.3	32.5	34.5	**28.2**
Sim_et	35.8	35.2	30.4	33.4	**27.2**
Real_dt	70	73.3	67.4	69.0	**63.3**
Real_et	73.1	77	71.0	71.1	**68.9**
Multi condition reverb training					
Sim_dt	11.9	11.3	13.5	11.3	**11.2**
Sim_et	12.2	11.5	12.9	11.6	**11.1**
Real_dt	25.9	25.7	30.7	**25.2**	25.3
Real_et	30.9	30.7	33.6	30.3	**29.4**

condition reverb training (simulated), there is average relative improvement of 7% over MFB features on the Sim test data and about 4% with Real test data. Furthermore, the results with the proposed front-end are better than the previously published results in REVERB Challenge [54].

3.2.4.3 *Semi-supervised Learning*

The semi-supervised ASR requires modeling speech with minimal supervision for ASR training. For semi-supervised ASR training, the Aurora-4 clean condition training set up is used with 70, 50 and 30% of the labeled training data. In this case, the modulation filters were learned using full unsupervised training data available in the clean training set with mel spectrogram input. Note that our semi-supervised ASR set-up does not involve any self-training or confidence-based approach. The ASR architecture and training criteria is same as it is with full data ASR in supervised manner. The semi-supervised ASR set-up trains the ASR with some percentage of the labeled data available.

Figure 3.5 shows the performance comparison of ASR with semi-supervised training using MFB and the proposed feature scheme for clean test data (Cond. A) condition as well as the average of all for test data conditions (average of 14 conditions). It can be observed that the proposed features are more resilient to reduced amounts of labeled training data as compared to the baseline system, even for the matched clean test condition. The proposed features also perform significantly better than MFB on the average of all test conditions (average relative improvement of 29% with use of 30% training data over MFB features).

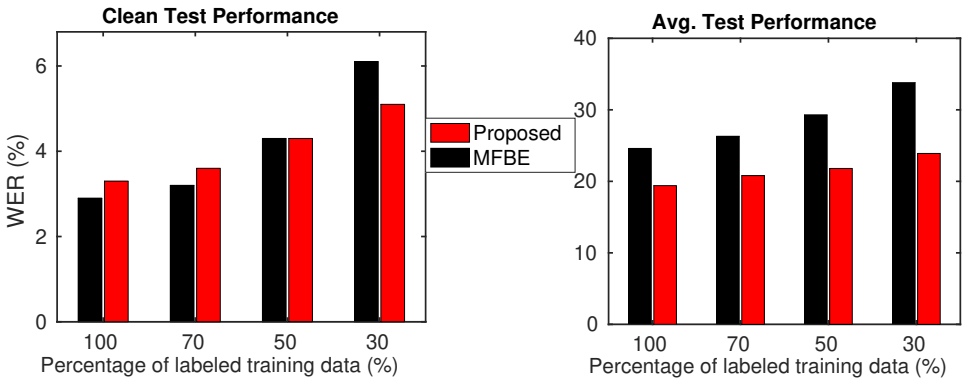


Fig. 3.5: Performance of ASR (WER) versus amount of clean labeled training data. Comparison between MFB and proposed modulation filtering (R2+S1, R2+S2) applied on MFB using cleaning training condition on Aurora-4. Results split for clean test condition (Cond. A) and average of all 14 test conditions.

3.2.4.4 *Brief section summary*

- Proposing an unsupervised data-driven approach to learn spectral and temporal modulation filters with a random initialization using CRBM.
- Obtaining multiple irredundant data-driven filters with the CRBM and residual spectrograms. A filter selection criterion using average hidden activation probability in CRBM.
- Illustrating robustness in noisy and reverberant conditions using the proposed modulation filtering scheme.

3.3 Spectro-Temporal 2-D Filters Using RBM

In the work discussed above, the temporal modulation (rate) filters and spectral modulation (scale) filters are learnt separately through temporal and spectral trajectories of speech spectrograms. However, the spectro-temporal modulation patterns can be learnt jointly from the 2-D patches of spectrograms through 2-D modulation filters. While the 2-D filter learning allow the joint spectro-temporal characteristics to be learnt (simultaneous spectral and temporal processing), there have been works that study the separable spectro-temporal filters and its effects on ASR performance [94]. Here in this section, learning of 2-D spectro-temporal modulation filters with rank constraint is attempted using convolutional RBM.

This is motivated by the evidence from human auditory system which reveal that the inherent robustness may be primarily attributed to the spectro- temporal filtering performed by cortical neurons [66, 25, 67]. Also, a recent approach to separable spectro-temporal Gabor filter bank features is proposed in [94] which motivates us to learn spectro-temporal modulation characteristics jointly from the data with rank constraints.

3.3.1 *2-D Convolutional RBM (CRBM)*

A 2-D CRBM is a probabilistic model where hidden units $\boldsymbol{H}$ (dimension $N_{H_r} \times N_{H_s}$) represent the presence/absence of local features in subwindows of visible units $\boldsymbol{V}$ ($N_{V_r} \times N_{V_s}$) [76]. In this work, the input layer $\boldsymbol{V}$ consists of a batch of 2-D patches sampled from speech spectrogram. The joint energy function of CRBM is given as:

$$E(\boldsymbol{V}, \boldsymbol{H}, \theta) = -\sum_{q} \boldsymbol{H}_q (\boldsymbol{W} \odot \boldsymbol{V}_{(q)}) - \sum_{i} b\boldsymbol{V}_i - \sum_{q} c\boldsymbol{H}_q \tag{3.3}$$

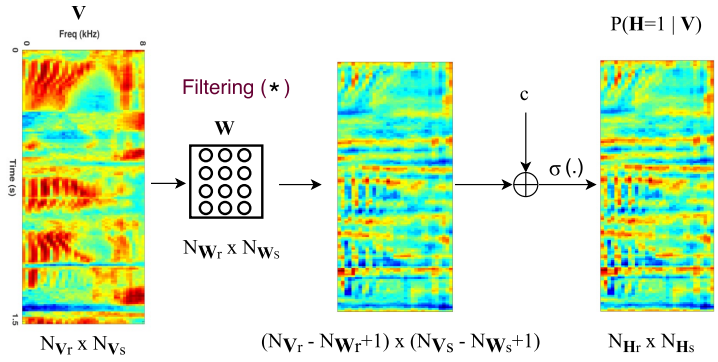


Fig. 3.6: Block schematic of the proposed CRBM architecture for learning modulation filter $\boldsymbol{W}$) (forward pass of CRBM).

Here, $\boldsymbol{W}$ is the weight matrix (filter) of dimension $(N_{W_r} \times N_{W_s} = (N_{V_r} - N_{H_r} + 1) \times (N_{V_s} - N_{H_s} + 1))$, $\boldsymbol{V}_{(q)}$ is subwindow of patch $\boldsymbol{V}$ with top left corner at unit q and with the dimensions same as that of $\boldsymbol{W}$, index q iterates over units of $\boldsymbol{V}$, $\odot$ denotes the dot product of matrices after linearizing its elements, $\theta = (\boldsymbol{W}, b, c)$ are the model parameters, $\boldsymbol{H}_q$ is the element of the matrix $\boldsymbol{H}$ at location q. The conditional probability model is given by:

$$P(\boldsymbol{H}_q = 1|\boldsymbol{V}) = \sigma((\boldsymbol{W} \odot \boldsymbol{V}_{(q)}) + c) \tag{3.4a}$$

$$P(\boldsymbol{V}_p = 1|\boldsymbol{H}) = \sigma((\boldsymbol{W}^\star \odot \boldsymbol{H}_{(p)}) + b), \tag{3.4b}$$

where σ is the sigmoid function, $\boldsymbol{W}^\star$ is the horizontally and vertically flipped version of the original filter, $\boldsymbol{H}_{(p)}$ is sub-window of patch $\boldsymbol{H}$ with top left corner at unit p and size same as that of $\boldsymbol{W}$, index p iterates over units of $\boldsymbol{H}$, $\boldsymbol{V}_p$ is the element of the matrix $\boldsymbol{V}$ at location p. The one-step contrastive divergence (Gibbs sampler) approximation for CRBM is given by:

$$\triangle_{\boldsymbol{W}} J(\boldsymbol{V}; \theta) = < \boldsymbol{V} \star \boldsymbol{H} >_{data} - < \boldsymbol{V} \star \boldsymbol{H} >_{model} \tag{3.5}$$

where $\star$ is the 2-D convolution operation. For reference, this equation is analogous to gradient ascent in 1-D Gaussian RBM with linear layer as

$$\triangle_{\boldsymbol{W}_{ij}} J(\boldsymbol{v}; \theta) = < \boldsymbol{v}_i \boldsymbol{h}_j >_{data} - < \boldsymbol{v}_i \boldsymbol{h}_j >_{model} . \tag{3.6}$$

The weight matrix is updated in an iterative learning process over several steps. The block schematic of the proposed modulation filter learning scheme from speech spectrogram through convolutional RBM (CRBM) is shown in Figure 3.6. The input layer $\boldsymbol{V}$ consists of a batch of 2-D patches sampled from speech spectrogram. Each 2-D patch of $\boldsymbol{V}$ consists of sub-band energy trajectory for 1.5 sec of speech along temporal dimension and an all-band energy trajectory along spectral dimension (40 bands) ($N_{V_r} = 150$, $N_{V_s} = 40$).

3.3.2 *Rank-1 Constraint on Weight Learning*

To constrain the weight matrix $\boldsymbol{W}$ as a separable rank-1 matrix, $\boldsymbol{W}$ is defined as the outer product of 1-D rate filter $\boldsymbol{r}$ and 1-D scale filter $\boldsymbol{s}$, i.e., $\boldsymbol{W} = \boldsymbol{r}\boldsymbol{s}^\top$. The gradient of J is computed with respect to $\boldsymbol{r}$ and $\boldsymbol{s}$ separately (unlike with respect to each element of $\boldsymbol{W}$). Let $\tilde{\boldsymbol{V}} = \boldsymbol{V} \star \boldsymbol{s}^\top$, where $\star$ denotes convolution operation, $\boldsymbol{V}$ is 2-D input spectrogram patch convolved with transposed 1-D scale kernel $\boldsymbol{s}$. The gradient ascent equation with respect to rate filter ($\boldsymbol{r}$) gives:

$$\triangle_{\boldsymbol{r}} J(\boldsymbol{V}; \theta) = < \tilde{\boldsymbol{V}} \star \boldsymbol{H} >_{data} - < \tilde{\boldsymbol{V}} \star \boldsymbol{H} >_{model} \tag{3.7}$$

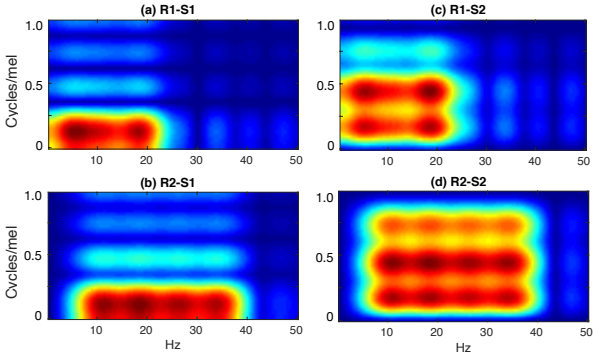


Fig. 3.7: The magnitude response of the proposed 2-D data driven filters (rank-1) obtained from mel spectrogram of clean training data.

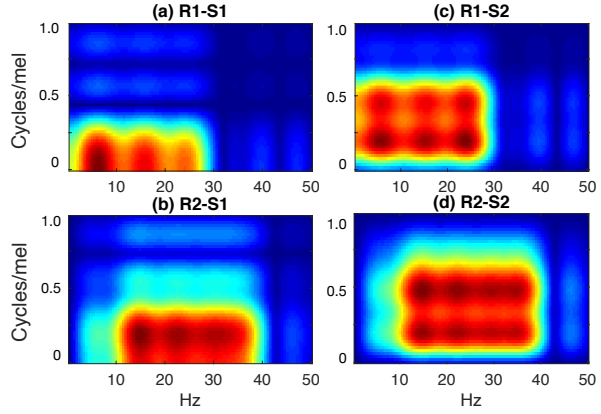


Fig. 3.8: The magnitude response of the proposed 2-D data driven filters (rank-1) obtained from mel spectrogram of multi condition training data.

where $\boldsymbol{s}$ is the scale filter obtained from previous iteration, $\boldsymbol{V}$ and $\boldsymbol{H}$ being the input 2-D patch and hidden activation patch, respectively. Similarly, let $\tilde{\boldsymbol{H}} = \boldsymbol{H} \star \boldsymbol{r}$. The gradient ascent equation with respect to scale filter ($\boldsymbol{s}$) gives:

$$\triangle_{\boldsymbol{s}} J(\boldsymbol{V}; \theta) = < \boldsymbol{V} \star \tilde{\boldsymbol{H}} >_{data} - < \boldsymbol{V} \star \tilde{\boldsymbol{H}} >_{model} \tag{3.8}$$

Hence, the filter update equations become:

$$\boldsymbol{r}' = \boldsymbol{r} + \eta(\triangle_{\boldsymbol{r}} J); \quad \boldsymbol{s}' = \boldsymbol{s} + \eta(\triangle_{\boldsymbol{s}} J) \tag{3.9}$$

where η is the learning rate. Subsequently, the 2-D filter $\boldsymbol{W}$ is updated as $\boldsymbol{W}' = \boldsymbol{r}'\boldsymbol{s}'^{\top}$. Several iterative steps are performed to learn the 2-D filters and the filters are thus learnt purely from a generative modeling perspective.

3.3.3 *Multiple Filter Learning and Selection*

In the analysis, it is observed that the first 2-D filter learnt from the input mel spectrogram is invariably a low-pass in both rate and scale domain (Figure 3.7 (a) and 3.8 (a)). For learning multiple 2-D filters that are less redundant [75], the residual approach is used discussed in Section 3.2.2. After an initial 2-D filter is learnt (we name it R1-S1), the contribution of learnt rate component (R1) is removed from the original spectrogram by subtracting the original spectrogram from the rate filtered spectrogram. This residual (containing the high rate and full scale information) is fed back to CRBM for learning next filter (R2-S1). Similarly, the contribution of learnt scale component (S1) is removed from the original spectrogram and the residual (containing the full rate and high scale information) is fed to CRBM for learning next filter

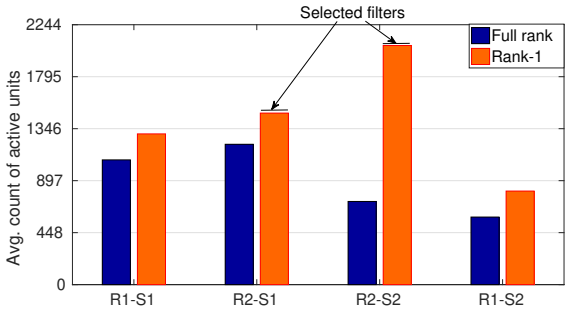


Fig. 3.9: The average count of active hidden units of CRBM model for full rank and rank-1 filters for clean training.

Table 3.6: Word error rate (%) in Aurora-4 database for clean training condition with various feature extraction schemes.

Cond	MFB	PFB	ETS	RAS	CRBM-1D	CRBM-2D
A	3.4	3.3	**3.2**	3.5	3.3	**3.2**
B	18.9	16.2	16.3	18.0	**13.6**	13.8
C	15.3	**11.7**	14.5	13.1	16.0	13.0
D	35.2	32.8	32.0	35.6	**29.0**	29.9
Avg.	24.7	22.1	21.9	24.4	**19.4**	19.9

(R1-S2). The contribution of both (R1) and (S1) from the original spectrogram is also removed for learning filter (R2-S2) from the residual. This method is extension of the 1-D approach to irredundant filter learning discussed in Section 3.2.2. For the CRBM learning, the 2-D weight matrix $\boldsymbol{W}$ is initialized as the outer product of the 1-D rate and scale filters learnt from CRBM using corresponding 1-D inputs sampled from spectrogram. Figure 3.7 shows the magnitude response of the learnt 2-D rank-1 filters obtained from mel spectrogram of clean speech data of Aurora-4 corpus. Similarly, the 2-D rank-1 filters are learned from mel spectrogram of multi condition training data, shown in Figure 3.8. As seen here, deriving the filters using MP style algorithm provides irredundant 2-D filters.

In order to select 2-D filters for ASR (4 learnt from rank-1 and 4 learnt from full rank), the average number of active hidden units of the CRBM (with a total of 4488 hidden units for $N_{\boldsymbol{W}_r} \times N_{\boldsymbol{W}_s} = 15 \times 8$) is computed for each 2-D filter by a forward pass operation of the set of input spectrograms through the CRBM. The average active count is computed using $P(\boldsymbol{H}_q = 1|\boldsymbol{V})$ summed over all q units (count of active units for a given input) and averaged over a number of input patches from the validation data. Based on the highest average active count, the (R2-S1) and (R2-S2) filter with rank-1 constraint gives maximum average active units amongst all filters, as shown in Figure 3.9. Similar trend is observed for 2-D filters for multi condition training data. This criterion represents a data driven unsupervised approach to filter selection. This is again a 2-D extension of the filter selection approach discussed for 1-D CRBM in Section 3.2.2.

The features for ASR are derived using two streams of filtered spectrograms using the rank-1 filters (R2-S1 and R2-S2). These spectrogram streams are concatenated and fed to a DNN based ASR system. The input features are mean-variance normalized at utterance level before DNN training.

3.3.4 *Experiments*

3.3.4.1 *Noisy speech recognition*

The WSJ Aurora-4 corpus discussed in Section 2.4.1 is used for conducting ASR experiments. The ASR performance of the discussed modulation filtering approach is compared with traditional mel filter bank

Table 3.7: Word error rate (%) in Aurora-4 database for multi condition training with various feature extraction schemes.

Cond	MFB	PFB	ETS	RAS	CRBM-1D	CRBM-2D
A	4.2	4.1	4.5	4.6	**3.7**	4.0
B	7.8	8.0	8.6	8.5	**7.3**	**7.3**
C	8.4	7.8	8.0	9.7	**7.1**	7.6
D	18.5	19.7	18.8	19.1	**16.2**	16.7
Avg.	12.1	12.7	12.6	12.8	**10.8**	11.1

Table 3.8: Word error rate (%) in REVERB Challenge database for clean and multi-condition training.

Cond.	MFB	PFB	CRBM-1D	CRBM-2D	MFB	PFB	CRBM-1D	CRBM-2D
	Clean training				Multi training			
Sim_dt	37.2	36.3	**28.2**	**28.2**	11.9	11.3	**11.2**	11.7
Sim_et	35.8	35.2	27.2	**23.6**	12.2	11.5	**11.1**	11.5
Real_dt	70	73.3	**63.3**	63.6	25.9	25.7	**25.3**	26.5
Real_et	73.1	77.0	**68.9**	**68.9**	30.9	30.7	**29.4**	30.6

energy (MFB) features and other baseline features discussed in Section 2.1. The ASR performance in clean training condition is reported in Table 3.6. From the results, it can be observed that PFB and ETS features provide better performance compared to the MFB and RAS features. The data driven modulation filtering approach on mel spectrogram provides significant improvement in noisy and channel distortion scenarios (average relative improvements of 19% over MFB features).

In the multi condition training and test scenario (reported in Table 3.7), the MFB features perform better than all other baseline features. The proposed feature extraction improves the performance of ASR compared to the baseline features by average relative improvements of 9% over MFB.

Comparison with 1-D RBM filtering - If we compare the ASR performance of the proposed features using 2-D rank-1 CRBM (Table 3.6 and Table 3.7), with features derived using 1-D CRBM filtering (Table 3.2 and Table 3.3 discussed in Section 3.2), it can be observed that the ASR with 1-D CRBM performs better than with 2-D rank-1 CRBM in both clean and multi-condition training on Aurora-4 dataset.

3.3.4.2 *Reverberant speech recognition*

The ASR experiments on reverberant speech data are performed using REVERB challenge corpus discussed in Section 2.4.2. The 2-D rank-1 filters are learnt from mel spectrogram of Train dataset - separately for both clean and multi condition. Table 3.8 shows the ASR performance for clean and multi-condition training conditions using MFB, PFB and the proposed modulation filtering (R2-S1+R2-S2) applied on MFB.

It can be observed that the proposed features perform better than MFB and PFB under almost all test conditions with clean and reverb training data. For the clean training, there is an average relative improvement of 29% over MFB features on Sim test data and about 8% with Real test data. The results with the proposed front-end are better than the best published results in REVERB Challenge [53]. For the multi condition reverb training (simulated), there is an average relative improvement of 4% over MFB features on the Sim test data and the performance is similar to MFB with Real test data.

Comparison with 1-D RBM filtering - If we compare the ASR performance of the proposed features using 2-D rank-1 CRBM (Table 3.8), with features derived using 1-D CRBM filtering (Table 3.5 discussed in Section 3.2), it can be observed that the ASR with 1-D CRBM performs comparable to 2-D CRBM in clean training while 1-D CRBM performs better than 2-D rank-1 CRBM in multi condition training.

Table 3.9: Word error rate (%) in Aurora-4 database for clean and multi condition training using lesser amount of labeled training data (70%, 50%, 30%).

Training data	100%		70%		50%		30%	
	MFB	CRBM-2D	MFB	CRBM-2D	MFB	CRBM-2D	MFB	CRBM-2D
Clean	24.6	**19.9**	26.3	**21.1**	29.3	**22.5**	33.8	**24.9**
Multi	12.1	**11.1**	15.8	**14.4**	17.6	**16.3**	21	**19.3**

3.3.4.3 *Semi-supervised training*

For semi-supervised ASR training, the Aurora-4 clean and multi-condition training set up is used with 70, 50 and 30% of the labeled training data. The modulation filters are learnt using full unsupervised clean and multi-condition training data, respectively, available in the training set with mel spectrogram input. The performance comparison of ASR with semi-supervised training is shown in Table 3.9 for MFB and the proposed feature scheme for the average of all test data conditions (14 conditions). These results indicate that the proposed features are more resilient to reduced amounts of labeled training data as compared to the baseline system (especially for clean training condition). The proposed features perform significantly better than MFB features for the average of all test conditions (average relative improvement of 26% for clean training and average relative improvement of 8% for multi-condition training with use of 30% labeled training data).

Furthermore, if we compare the semi-supervised training ASR results of 2-D rank-1 CRBM proposed here with corresponding 1-D CRBM features for clean training reported in Section 3.2.4.3, it can be observed that, again, the features with 1-D CRBM perform slightly better than 2-D CRBM.

3.3.4.4 *Brief section summary*

- Proposing the rank-1 constraint in gradient ascent method to obtain separable 2-D spectro-temporal modulation filters in the CRBM framework.
- Obtaining multiple irredundant filters using residual spectrograms in rate and scale domain, followed by filter selection using average number of active hidden units in the CRBM.
- Comparison of 1-D and rank-1 2-D CRBMs indicate that the 1-D approach gives slightly improved ASR results.

3.4 Comparison of Temporal Filter Learning in RBM with Autoencoder (AE) and Generative Adversarial Network (GAN)

This section describes and compares the filter learning approach through RBM with other two models-autoencoder and generative adversarial network (GAN) used to capture the temporal modulation characteristics from spectrogram data. All the models use temporal trajectories of 1.5 s length derived from speech spectrograms (25 ms frame length with shift of 10 ms containing 80 mel-bands) similar to CRBM learning discussed in previous section.

3.4.1 *Convolutional Autoencoder (CAE)*

Autoencoder (AE) is a neural network that can convert high-dimensional data to low-dimensional codes using encoder, and use a decoder block to reconstruct the data back [38]. The encoder performs the deterministic mapping $f(\theta)$ from a n-dimensional input vector $\boldsymbol{x}$ into a hidden (encoded) representation $\boldsymbol{y}$ as:

$$f_\theta(x) = \boldsymbol{y} = \sigma(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}) \tag{3.10}$$

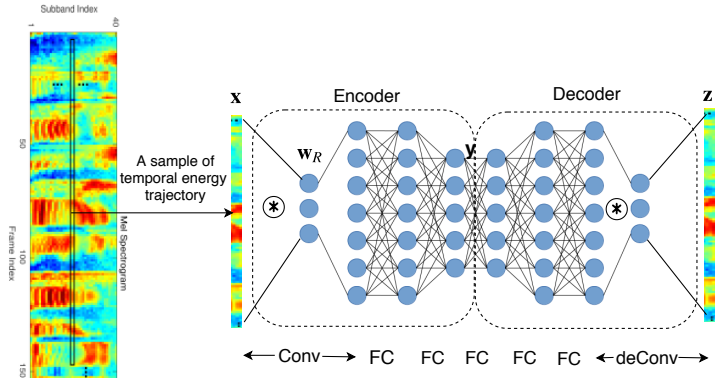


Fig. 3.10: Block diagram of temporal modulation filter learning using CAE from spectrograms.

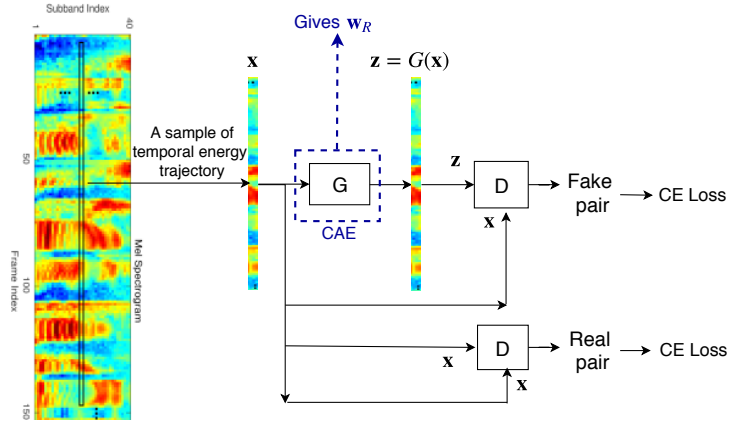


Fig. 3.11: Block diagram of temporal modulation filter learning using GAN - training G in an adversarial framework.

with parameters $\theta = \{\boldsymbol{W}, \boldsymbol{b}\}$, where $\boldsymbol{W}$ is a weight matrix, $\boldsymbol{b}$ is a bias vector and s is the nonlinearity. The resulting encoded representation $\boldsymbol{y}$ is then mapped back (decoded) to a reconstructed d-dimensional vector $\boldsymbol{z}$ in the input space as:

$$g_{\theta'}(\boldsymbol{y}) = \boldsymbol{z} = \sigma(\boldsymbol{W}'\boldsymbol{y} + \boldsymbol{b}') \tag{3.11}$$

with $\theta' = \{\boldsymbol{W}', \boldsymbol{b}'\}$, with a loss function defined as mean squared error,

$$L(\boldsymbol{x}, \boldsymbol{z}) = ||\boldsymbol{x} - \boldsymbol{z}||^2. \tag{3.12}$$

A Convolutional autoencoder (CAE) replaces the fully connected layer with the convolutional layer that is able to learn local patterns by shared weights of connections [59, 65]. In this work, CAE is used to capture the temporal modulations of the speech spectrogram data. The architecture of CAE used is shown in Fig. 3.10. In order to analyze the effect of filtering, only one convolutional layer in encoder (Conv) and one convolutional layer in decoder (deConv) is used. The number of kernels in first Conv layer and last deConv layer is also restricted to one. The kernel (filter) learnt is denoted as $\boldsymbol{w_R}$ in the Fig. 3.10. The other layers are fully connected (FC) layers. To learn multiple non-overlapping filters (corresponding to different modulation characteristics), a sequential filter learning criteria is followed as explained in Section 3.4.3 [1].

3.4.2 *Conditional Generative Adversarial Network (cGAN)*

The GANs are unsupervised generative models that learn to produce realistic samples of input data via an adversarial learning. It consists of two models (usually neural networks) that are trained simultaneously: a

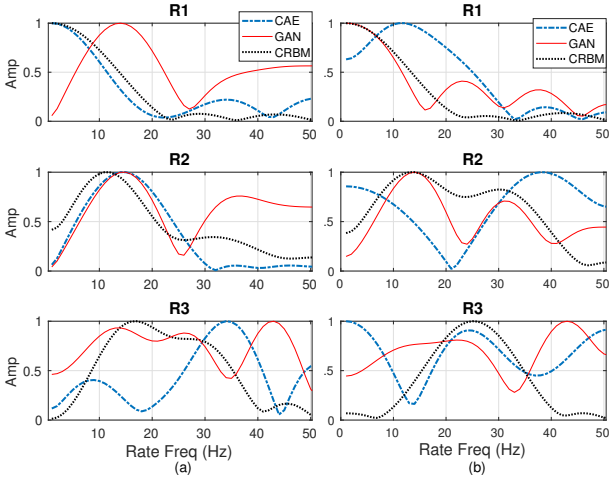


Fig. 3.12: Rate filters learnt from (a) clean WSJ mel spectrogram (b) multi condition WSJ mel spectrogram with residual approach.

generator G that captures the data distribution, and a discriminator D that estimates the probability that a sample came from the training data rather than G [31]. The training procedure for G is to maximize the probability of D making an error. The generator $G(n;\theta_G)$ is learned by mapping noise n to data space $\boldsymbol{x}$, where G is a differentiable function represented by a CAE with parameters θ_G.

The discriminator $D(\boldsymbol{x};\theta_D)$ is a second network that outputs a scalar $D(\boldsymbol{x})$ representing the probability that $\boldsymbol{x}$ is a true data point and not a model generated sample. The D is trained to maximize the probability of assigning the correct label to both training examples and the samples from G. In other words, D and G play the following two-player minimax game with value function $V(G,D)$:

$$\min_G \max_D V(G,D) = E_x[\log D(\boldsymbol{x})] + E_n[\log(1 - D(G(\boldsymbol{n})))], \tag{3.13}$$

where $\boldsymbol{n}$ is a realization of noise sample. In conditional GANs (cGAN) [44], we learn a mapping from observed sample $\boldsymbol{x}$ and random noise vector $\boldsymbol{n}$, to $\boldsymbol{z}$, $G : \{\boldsymbol{x}, \boldsymbol{n}\} \rightarrow \boldsymbol{z}$. In particular, the D observes both real and generated samples as a pair, with the task of detecting whether it is real pair or a fake pair. The objective of a cGAN can be expressed as:

$$\min_G \max_D V(D,G) = E_{\boldsymbol{x},\boldsymbol{z}}[\log D(\boldsymbol{x},\boldsymbol{z})] + E_{\boldsymbol{x},\boldsymbol{n}}[\log(1 - D(\boldsymbol{x}, G(\boldsymbol{x},\boldsymbol{n})))]. \tag{3.14}$$

In this work, cGAN is used to learn modulation characteristics of temporal spectrogram trajectories with aim of reconstructing $\boldsymbol{z} = \boldsymbol{x}$, i.e. identity mapping with G. The block diagram for filter learning with GAN is shown in Fig. 3.11. The input to the generator G is input trajectory $\boldsymbol{x}$ alone (and no noise). The CAE described in previous section is used as G and D is a classifier with two classes as real and fake pair.

3.4.3 *Multiple Filter Learning*

For learning multiple filters that are less redundant [75], the same residual approach is used as discussed earlier in Section 3.2.2. The training is started with random initialization of weights and allows the different unsupervised models to learn modulation characteristics from data. The normalized magnitude response of the filters learnt from the three filter learning schemes discussed in is shown in Figure 3.12 in clean and multi condition training setup.

In the analysis, it is observed that the first filter learnt from the input mel spectrogram is invariably a low-pass in CAE and CRBM, while the first filter from GAN model has a bandpass characteristic in clean condition (Figure 3.12 (a)). As seen here, deriving the filters using MP style algorithm provides irredundant filters. In the case of multi condition filter learning (Figure 3.12 (b)), it is assumed that a filter will learn common underlying representation of all types of input noises.

Table 3.10: Comparison of WER (%) in clean training Aurora-4 database for each filter of corresponding model (average of all 14 test conditions).

Model	R1	R2	R3
CRBM	27.7	**23.0**	23.1
CAE	27.4	**20.7**	21.9
GAN	**20.3**	20.7	22.9

Table 3.11: Word error rate (%) in Aurora-4 database for clean training condition with various feature extraction schemes.

Cond	MFB	PFB	ETS	RAS	CRBM R2	CAE R2	GAN R1
A	3.4	3.3	3.2	3.5	**2.7**	3.0	3.2
B	18.9	16.2	16.3	18.0	16.7	14.0	**13.8**
C	15.3	**11.7**	14.5	16.0	13.9	13.1	13.6
D	35.2	32.8	32.0	35.6	34.1	31.6	**30.8**
Avg.	24.7	22.1	21.9	24.4	23.0	20.7	**20.3**

The features for ASR are derived by filtering the log mel spectrogram using filters learnt from unsupervised models. The features are mean-variance normalized at utterance level before DNN training.

3.4.4 *Experiments*

3.4.4.1 *Noisy Speech Recognition*

The WSJ Aurora-4 corpus described in Section 2.4.1 is used for conducting ASR experiments. We trained the ASR in clean condition for each of the learnt filter R1, R2, R3 individually for all 3 models, and observed that the filter with bandpass characteristic gives the best performance amongst the three. From the average word error rate (WER) reported in Table 3.10, the CRBM and CAE gives the best performance with R2, while GAN having bandpass characteristic in R1 provides the best performance over the other two.

It can be attributed to the fact that the three techniques are expected to describe either distribution of the data or the reconstruction of the data. For eg. CRBM tries to fit in Boltzmann distribution and minimize Boltzmann energy function, whereas CAE learn by minimizing MSE of reconstruction. Both the methods concentrate on high energy regions in the initial filter learning R1 (which turns out to be low-pass in Fig. 3.12 (a)) and then the band-pass filters R2 with residual approach. However, the 'adversarial' cost function in GAN (min-max game) helps to focus on other characteristics (apart from the higher energy region alone) which results in learning BP filter itself as first filter R1 and its variants in successive steps. The effect of the learnt filters on ASR in Table 3.10 shows GAN R1 performs the best among all.

The ASR performance in clean training condition with the best filters is reported in Table 3.11. The results are reported for average of 14 test data conditions classified into four groups as: A - clean data, B - noisy data, C - clean data with channel distortion, and D - noisy data with channel distortion. From the table, it can be observed that PFB and ETS features provide better performance compared to the MFB and RAS features. The data driven modulation filtering approach on mel spectrogram provides significant improvement in noisy and channel distortion scenarios. The GAN features also gave superior performance compared to CAE and CRBM features (average relative improvements of 18% by GAN - R1 over MFB).

In multi condition training scenario, a similar trend was observed for ASR with respect to bandpass characteristic of the learnt filter. The CRBM and GAN gave the best perfromance with R2, while CAE

Table 3.12: Word error rate (%) in Aurora-4 database for multi condition training with various feature extraction schemes.

Cond	MFB	PFB	ETS	RAS	CRBM R2	CAE R1	GAN R2
A	4.2	4.1	4.5	4.6	**3.4**	3.8	3.5
B	7.8	8.0	8.6	8.5	**7.2**	7.6	**7.2**
C	8.4	7.8	8.0	9.7	**7.1**	8.0	7.5
D	18.5	19.7	18.8	19.1	**16.8**	18.4	17.4
Avg.	12.1	12.7	12.6	12.8	**11.0**	12.0	11.3

Table 3.13: Word error rate (%) in REVERB Challenge database for clean and multi condition training.

Cond.	MFB	PFB	GAN	MFB	PFB	GAN
	Clean training			Multi training		
Sim_dt	37.2	36.3	**28.9**	11.9	**11.3**	11.4
Sim_et	35.8	35.2	**27.4**	12.2	11.5	**11.4**
Real_dt	70	73.3	**67.1**	25.9	25.7	**24.8**
Real_et	73.1	77	**69.1**	30.9	30.7	**30.1**

having bandpass characteristic in R1 provides the best performance over the other two. The comparison of the best filters in multi condition are reported in Table 3.12. The filtered features improves the performance of ASR compared to the baseline features. Here, the CRBM provide the best performance which is found to be moderately better than GAN (average relative improvements of 9% with CRBM-R2 and 7% with GAN-R2 over MFB).

3.4.4.2 *Reverberant speech recognition*

The ASR experiments on reverberant speech data are performed using REVERB challenge corpus discussed in Section 2.4.2. The rate filters with GAN are learnt from mel spectrogram of Train dataset - separately for both clean and multi condition. Table 3.13 shows the ASR performance for clean and multi-condition training conditions using MFB, PFB and the selected modulation filtering GAN-R1 (clean) and GAN-R2 (multi condition).

It can be observed that the selected features perform better than MFB and PFB under almost all test conditions with clean and reverb training data. For the clean training, there is an average relative improvement of 23% over MFB features on Sim test data and about 5% with Real test data. For the multi condition reverb training, there is an improvement under all test conditions with average relative improvement of 6% over MFB features on Sim test data and about 3% for Real test data.

3.4.4.3 *Semi-supervised training*

For semi-supervised ASR training, the Aurora-4 clean and multi condition training set up is used with 70, 50 and 30 % of the labeled training data. The modulation filters are learnt using full unsupervised clean and multi condition training data, respectively. The performance comparison of ASR with semi-supervised training is shown in Table 3.14 for MF and the selected (GAN-R1 for clean, GAN-R2 for multi) feature scheme (average WER of all 14 test data conditions). These results indicate that the selected filtered features are more resilient to reduced amounts of labeled training data as compared to the baseline system. These features perform significantly better than MF features (average relative improvement of 32 % in clean training and 22 % in multi condition training with the use of 30 % labeled data).

Table 3.14: Word error rate (%) in Aurora-4 database using lesser amount of labeled training data (70%, 50%, 30%).

Training data	100%		70%		50%		30%	
	MFB	GAN	MFB	GAN	MFB	GAN	MFB	GAN
Clean	24.6	**20.3**	26.3	**20.8**	29.3	**21.4**	33.8	**22.9**
Multi cond.	12.1	**11.3**	15.8	**13.4**	17.6	**14.5**	21.0	**16.4**

3.4.4.4 *Brief section summary*

- Comparing the three unsupervised models for modulation filter learning. The model architectures are designed to learn and interpret kernel as modulation filters.
- From the ASR results, the bandpass temporal modulation region proved to be useful for noise robustness, and the kernels learnt are able to capture these regions.
- The cGAN architecture is more robust compared to CRBM in clean training, however, CRBM gives the best performance for multi-condition training.

3.5 Modified Loss Function - Variational Autoencoder (VAE)

3.5.1 *Variational Autoencoder (VAE)*

The VAE model draws the samples of latent representation from a standard normal distribution, i.e $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ [52]. It maximizes the probability of each input $\boldsymbol{x}$ in the training set under the generative process according to

$$p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}. \tag{3.15}$$

The model involves a two-step process as: (1) a value $\boldsymbol{z}$ is generated from prior distribution $p(\boldsymbol{z})$; (2) a value $\boldsymbol{x}$ is generated from conditional distribution $p_\theta(\boldsymbol{x}|\boldsymbol{z})$. However, in the assumed generative model with a decoder neural network, the function $p_\theta(\boldsymbol{x})$ is not always differentiable w.r.t. θ due to the intractable integral in Eq. 3.15; therefore θ cannot be optimized directly. The VAE framework resolves these problems based on a variational lower bound method by using a $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ (probabilistic encoder with encoder parameters ϕ) that can take value of $\boldsymbol{x}$ and give a distribution over $\boldsymbol{z}$ values (approximates the true posterior distribution $p_\theta(\boldsymbol{z}|\boldsymbol{x})$). The key idea behind the variational autoencoder is to attempt to sample values of $\boldsymbol{z}$ that are likely to have generated $\boldsymbol{x}$, and lower bound the value of $p_\theta(\boldsymbol{x})$ using those. To achieve this, the encoder and decoder parameters, ϕ and θ, respectively, are trained by maximizing the lower bound $\mathcal{L}(\theta, \phi; \boldsymbol{x})$ of the marginal likelihood $\log p_\theta(\boldsymbol{x})$, given as

$$\mathcal{L}(\theta, \phi; \boldsymbol{x}) = -D_{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})] + \mathbb{E}_{\boldsymbol{z}|\boldsymbol{x}\sim q_\phi}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] \tag{3.16}$$

Given the parameters of the encoder network, $\boldsymbol{\mu}_\phi(\boldsymbol{x})$ and $\boldsymbol{\sigma}^2_\phi(\boldsymbol{x})$ which are the mean and variance parameters of $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ - one can sample from $\mathcal{N}(\boldsymbol{\mu}_\phi(\boldsymbol{x}), diag(\boldsymbol{\sigma}^2{}_\phi(\boldsymbol{x})))$ by first sampling $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, then computing $\boldsymbol{z} = \boldsymbol{\mu}_\phi(\boldsymbol{x}) + diag(\boldsymbol{\sigma}_\phi(\boldsymbol{x}))\boldsymbol{\epsilon}$, shown schematically in Fig. 3.13. Stochastic gradient ascent is performed on the lower bound $\mathcal{L}(\theta, \phi; \boldsymbol{x})$ w.r.t. model parameters (θ, ϕ). The negative of the first term in Eq. 3.16, $D_{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})]$ is termed as 'latent loss' (E_{Latent}) which is the KL divergence between two multivariate Gaussian distributions. The second term in Eq. 3.16, $\mathbb{E}_{\boldsymbol{z}|\boldsymbol{x}\sim q_\phi}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})]$, with Gaussian assumptions on $p_\theta(\boldsymbol{x}|\boldsymbol{z})$, reduces to the negative of mean square error (MSE) loss (E_{MSE}), typically used in conventional autoencoder. Thus, VAE loss function can also be viewed as a minimization of regularized MSE loss where the regularization comes from the KL divergence term.

3.5.1.1 *Comparing VAE with other deep generative models.*

The VAE is a generative model which minimizes the MSE of the reconstructed data along with a latent loss. The two other popular models for generative modeling in neural network framework are the restricted

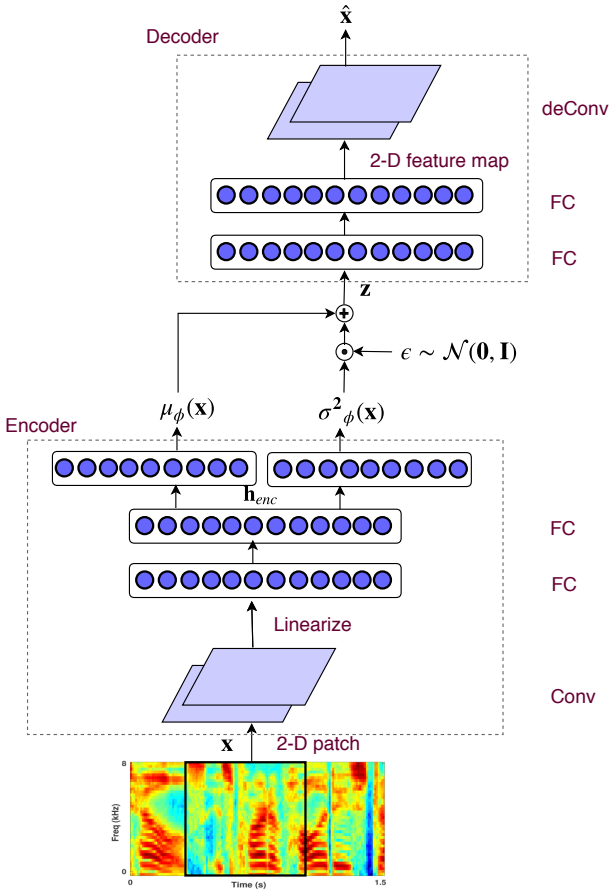


Fig. 3.13: Block schematic of filter learning with CVAE. Here FC denotes fully connected layer and Conv, deConv denotes convolution and deconvolution layer, respectively.

Boltzmann machine (RBM) [38] and the generative adversarial networks (GAN) [31] discussed in previous sections. The RBM model [38] also uses a latent representation similar to the VAE. The model assumes a Boltzmann distribution for the joint density function of the observation and latent variable. For the model parameter learning, a maximum likelihood approach is used where a Gibbs sampling framework is employed. The GAN models [31] also aim to use a latent data distribution to generate the observed data. The model uses a discriminative loss function (fake versus real) to further correct the generative model. The conventional GAN uses an independent distribution to generate the latent vector and does not use the observation data to generate the latent vector (unlike the VAE model).

3.5.2 *Convolutional VAE and Filter Learning*

The block diagram of the VAE model used for filter learning is shown in Figure 3.13. The convolutional VAE (CVAE) used in this work replaces the fully connected layer(s) in the encoder and decoder networks with convolution layer(s). The kernels (convolutional filters) of the deep CVAE trained using spectrogram input are interpreted as the modulation filters learned from the data that characterize the key modulations required to generate speech. The CVAE is trained in multi-condition fashion with a small number of filters (we use two filters in the convolutional layer).

Table 3.15: The architecture of the CVAE model used for filter learning.

Number of layers - encoder	Conv: 1, FC: 2
Number of layers - decoder	FC: 2, deConv: 1
Number of kernels in Conv/deConv	2 (kernel size: 5×5)
Number of nodes in FC	6000
Activation function	tanh
Latent Vector $\boldsymbol{z}$ Dimension	5000
Mini-batch size	1200
Optimization	Adam [51]
Learning rate	0.0001

3.5.2.1 *Implementation of CVAE for filter learning*

As outlined in Figure 3.13, the input to CVAE are the 2-D patches of log mel spectrograms. The mel spectrogram is computed using short-time Fourier transform of speech signal with 25 ms frame length and shift of 10 ms, and warping the frequency axis with 40 mel-bands. The dimension of the 2-D patch of the spectrogram at the input of CVAE is 150×40 (equivalent to 1.5s of speech from 40 mel bands). Table 3.15 gives the details of the CVAE architecture used in this work. The first layer of the 'encoder' is a convolutional layer with number of kernels = 2. The size of the kernels is 5×5 and is constrained to be of rank= 1 in order to learn separable spectro-temporal filters [94]. Hence, the filters $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ of the convolutional layer can be decomposed as the outer product of temporal modulation 'rate' filter $\boldsymbol{r}$ and spectral modulation 'scale' filter $\boldsymbol{s}$ as $\boldsymbol{W}_1 = (\boldsymbol{r}_1 {\boldsymbol{s}_1}^T)$ and $\boldsymbol{W}_2 = (\boldsymbol{r}_2 {\boldsymbol{s}_2}^T)$, respectively.

The output of convolutional (Conv) layer is linearized and fed to fully-connected (FC) layers of the encoder. The decoder then reconstructs the 2-D patch from the latent vector $\boldsymbol{z}$ by reversing the steps in the encoder (fully connected layers followed by a deconvolution layer [65]).

In the implementation of the CVAE, the loss function is modified as follows,

$$E_{Total} = \alpha E_{MSE} + \beta E_{Latent} + \gamma(E_{fr} + E_{fs}) + \delta E_{enc} \tag{3.17}$$

with,

$$E_{MSE}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = ||\boldsymbol{x} - \hat{\boldsymbol{x}}||_2^2 \quad , \quad E_{enc}(\boldsymbol{h}_{enc}) = ||\boldsymbol{h}_{enc}||_1 \tag{3.18a}$$

$$E_{Latent} = D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}), p(\boldsymbol{z})) \tag{3.18b}$$

$$E_{fr}(\boldsymbol{r}_1, \boldsymbol{r}_2) = ||\boldsymbol{r}_1 \star \boldsymbol{r}_2||_2^2 \quad , \quad E_{fs}(\boldsymbol{s}_1, \boldsymbol{s}_2) = ||\boldsymbol{s}_1 \star \boldsymbol{s}_2||_2^2 \tag{3.18c}$$

where $\star$ denotes convolution operation. The L2-norm of convolution of filters are introduced primarily to avoid learning redundant filters (filters with highly overlapping frequency responses). Note that minimizing the convolution loss of filters $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ (or $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$) is equivalent to minimizing the product of frequency response of these filters. The L1 norm loss (E_{enc}) encourages sparse representation in hidden latent dimensions. The sparse regularization term is also beneficial in this case as the latent dimensions in CVAE are quite high (5000). The scaling factors $\alpha, \beta, \gamma, \delta$ are hyper parameters which are set based on validation experiments. The benefits of this modified loss are highlighted in Sec. 3.5.4.

3.5.2.2 *Filter Responses*

The filters $\boldsymbol{r}_1, \boldsymbol{s}_1, \boldsymbol{r}_2, \boldsymbol{s}_2$ are iteratively updated using the gradients of the total loss function in Eq. 3.17. The CVAE is trained using multi condition and clean data of different databases separately. We start the network training with random initialization of the filters and allow the CVAE to learn modulation filter characteristics from data. Fig. 3.14 shows the normalized magnitude frequency response of the filters

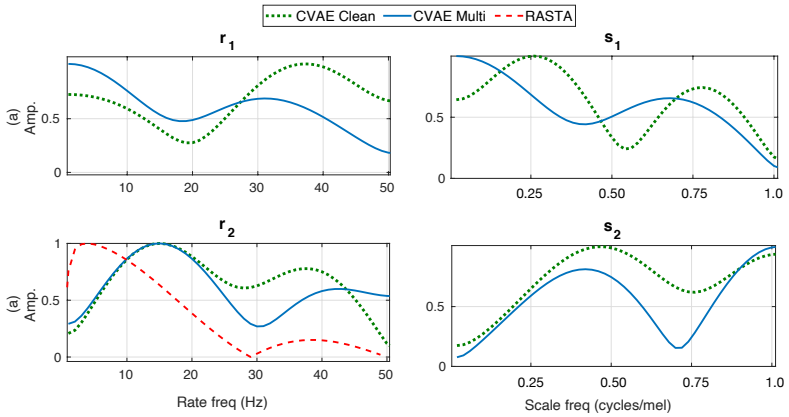


Fig. 3.14: Two sets of rate $(\boldsymbol{r}_1, \boldsymbol{r}_2)$ and scale filters $(\boldsymbol{s}_1, \boldsymbol{s}_2)$ learned from the CVAE model using the clean condition and multi-condition Aurora-4 dataset. The rate filters have low-pass and band-pass characteristics in this case. The RASTA filter is also shown in the $\boldsymbol{r}_2$ plot for reference.

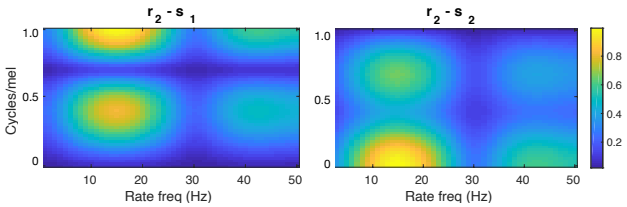


Fig. 3.15: The two 2-D filters $(\boldsymbol{r}_2, \boldsymbol{s}_1)$ and $(\boldsymbol{r}_2, \boldsymbol{s}_2)$ used in feature extraction for ASR in Aurora-4 multi-condition database.

learned using multi-condition and clean Aurora-4 database respectively. Since each 2-D filter is constrained to be rank-1, the frequency response of individual rate and scale components of the filters can be separately plotted. The value of scaling factors in cost function of CVAE used in this case are $\alpha = 1.0$, $\beta = 0.5$, $\gamma = 0.5$ and $\delta = 0.1$.

In the analysis, it is observed that the two rate filters learned from the input mel spectrogram have invariably low-pass and band-pass characteristics in multi-condition data, while it is band-stop and band-pass for clean data (Fig. 3.20). The scale filters jointly span the entire spectral modulation range. It is assumed that the filters will learn the common underlying representation of all types of input noisy speech, which would be dominated by underlying speech characteristics. The second row of Figure 3.14 also shows the comparison with the RASTA filter [33]. As seen here, the learnt data-driven rate filter somewhat resembles the perceptual knowledge driven RASTA filter. Also, the range of modulations captured by $\boldsymbol{r}_1, \boldsymbol{r}_2$ and $\boldsymbol{s}_1, \boldsymbol{s}_2$ are quite similar to the modulation filters found in human perceptual studies [25]. This is interesting in the sense that, even with random initialization, the data-driven generative modeling of a corpus of speech using the framework in CVAE can yield filters that are broadly similar to filters found in various perceptual studies on modulations. The frequency response of filters learned from other datasets is discussed in Sec. 3.5.4.

3.5.2.3 *Feature extraction for ASR*

The features for ASR are derived by filtering the log mel spectrogram using filters learned from the proposed approach. In this work, the rate filter with bandpass characteristic is selected as it has been observed earlier to be crucial for ASR performance [33, 1], while all the scale filters spanning spectral modulation space are used for ASR. Detailed analysis on filter selection and number of filters is given in Sec. 3.5.4. The 2-D filter responses for the filters used in multi-condition ASR for the Aurora-4 dataset are shown in Figure 3.15. The

Table 3.16: Word error rate (%) in Aurora-4 database for clean training condition with various feature extraction schemes and the proposed CVAE modulation filtering approach.

Cond	MFB	PFB	ETS	RAS	LDA	MHE	CRBM-1D	CRBM-2D	CVAE
A	3.4	3.3	3.2	3.5	3.7	3.5	3.3	3.2	**3.0**
B	18.9	16.2	16.3	18.0	20.1	17.4	**13.6**	13.8	15.9
C	15.3	**11.7**	14.5	16.0	15.9	14.6	16.0	13.0	13.6
D	35.2	32.8	**32.0**	35.6	36.3	35.4	**29.0**	29.9	33.0
Avg.	24.7	22.1	**21.9**	24.4	25.6	23.9	**19.4**	19.9	22.1

Table 3.17: Word error rate (%) in Aurora-4 database for multi condition training condition with various feature extraction schemes and the CVAE modulation filtering approach.

Cond	MFB	PFB	ETS	RAS	LDA	MHE	CRBM-1D	CRBM-2D	CVAE
A	4.2	4.1	4.5	4.6	4.7	4.0	**3.7**	4.0	**3.5**
B	7.8	8.0	8.6	8.5	9.9	8.3	**7.3**	**7.3**	7.4
C	8.4	7.8	8.0	9.7	10.0	8.1	7.1	7.6	**6.9**
D	18.5	19.7	18.8	19.1	21.2	19.6	**16.2**	16.7	17.1
Avg.	12.1	12.7	12.6	12.8	14.4	12.8	**10.8**	11.1	11.2

log mel spectrograms are filtered using filters $(\boldsymbol{r}_2, \boldsymbol{s}_1)$ and $(\boldsymbol{r}_2, \boldsymbol{s}_2)$ separately and are concatenated to derive features for ASR. This is motivated from the works in the past about neurophysiological evidence suggesting that processing of speech signals in the brain happens along parallel pathways and encode complementary information in the signal [35, 14]. The proposed features as well as all the other baseline features are mean-variance normalized at the utterance level before the acoustic model training. In all the ASR experiments, no speaker level normalization is performed.

3.5.3 *Experiments and Results*

3.5.3.1 *Datasets*

The Wall Street Journal (WSJ) Aurora-4 corpus discussed in Section 2.4.1 having different types of noises has been used for clean and multi-condition training and testing. The REVERB challenge dataset discussed in Section 2.4.2 having reverberation is the another dataset that has been used for training and testing. The CHiME-3 corpus for ASR discussed in Section 2.4.3 containing multi-microphone tablet device recordings from everyday environments, released as a part of 3rd CHiME challenge is also used [8]. The beamformed audio from 5 multi-channel recordings of REVERB and 8 multi-channel recordings of CHiME-3 dataset respectively are used for CVAE training, ASR training and testing.

3.5.3.2 *Kaldi ASR framework*

The ASR performance of the proposed modulation filtering approach (CVAE) is compared with traditional mel filter bank energy (MFB) features along with with other baseline features discussed in Section 2.1, with CRBM-1D approach discussed in Section 3.2, CRBM-2D discussed in Section 3.3. In particular, the RASTA features (RAS) and LDA features are included as they both perform modulation filtering in the temporal domain using a knowledge driven filter and a supervised data-driven filter, respectively.

Table 3.18: Word error rate (%) in REVERB Challenge database for multi-condition training (simulated) with test data from simulated and real reverberant environments.

Test Cond	MFB	PFB	RAS	MHE	CVAE
Sim_dev	9.1	8.6	10.1	8.4	**8.3**
Sim_eval	9.5	9.2	10.2	9.2	**8.6**
Real_dev	22.0	21.5	24.9	**21.0**	22.7
Real_eval	25.9	25.9	27.9	**24.5**	24.8
Avg.	16.5	16.3	18.3	**15.8**	16.0

3.5.3.3 *Results*

Aurora-4 - The results of various ASR experiments on clean and multi-condition Aurora-4 dataset is shown in Table 3.16 and 3.17 respectively. The results are reported for average of 14 test data conditions classified into four groups as: A - clean data, B - noisy data, C - clean data with channel distortion, and D - noisy data with channel distortion. As seen in Table 3.16 for clean training, the noise robust front-ends improve over the baseline mel-filter bank (MFB) performance. The CVAE approach improves over the baseline models in clean and additive noise conditions. For the experiments with different microphone, the ETSI features provide the best performance. The CVAE approach performs similar to the PFB features and improve over the baseline MFB features as well.

In Table 3.17 for multi-condition training, most of the noise robust front-ends do not improve over the baseline mel-filter bank (MFB) performance (except for condition C), as the acoustic models are trained using multi-condition noisy training data. The CVAE features provide significant improvements in ASR performance over the baseline system (average relative improvements of 7.5%). Furthermore, the improvements in ASR performance are consistently seen across all the noisy conditions of Aurora-4 dataset.

The Aurora-4 results also show comparison with modulation filtering using 1-D CRBM (learning 1-D rate and scale filters separately using residual approach and performing rate-scale filtering to obtain features, discussed in Section 3.2) and 2-D CRBM (learning 2-D rank-1 filters using residual approach and performing 2-D filtering to obtain features, discussed in Section 3.3) in terms of ASR performance. The results reveal 1-D CRBM method as the best filter learning approach among all in both clean and multi-condition training, with significant improvements in clean training condition.

REVERB - The ASR results on REVERB challenge dataset are shown in Table 3.18. The CVAE approach improves over the baseline features in the REVERB challenge dataset for the simulated conditions and for real reverberation in the evaluation dataset. However, the Hilbert envelope based compensation (MHE) improves over the CVAE approach in the evaluation test data for real reverberation.

CHiME-3 - The results for the CHiME-3 dataset are reported in Table 3.19. The CVAE approach to feature extraction provides significant improvements over the baseline system as well as the other noise robust front-ends considered here. On the average, the proposed approach provides relative improvements of 13% in the development set and 20% in the evaluation set. The detailed results on different noises in CHiME-3 are reported in Table 3.20. For all the noise conditions in CHiME-3 in simulated and real environments, the proposed approach shows significant improvements over the baseline system using mel filter bank features (MFB). In the evaluation dataset, the relative improvements over the baseline features for most of the noise conditions are above 20%.

3.5.4 *Discussion*

3.5.4.1 *Statistical significance of the ASR results*

To compare how one system performs better than other in statistical sense, Bootstrap estimate for confidence interval is used [10]. It computes a bootstrapping of WER to extract the 95% confidence interval (CI), and also gives a probability of improvement (POI) by the system-in-test (system with proposed features) over the reference system (baseline system with MFB features). Table 3.21 shows the analysis for various conditions

Table 3.19: Word error rate (%) in CHiME-3 Challenge database for multi-condition training (real+simulated) with test data from simulated and real noisy environments.

Test Cond	MFB	PFB	RAS	MHE	CVAE
Sim_dev	14.3	13.7	14.6	14.4	**12.4**
Real_dev	11.6	12.0	11.8	12.0	**10.2**
Avg.	13.0	12.9	13.2	13.2	**11.3**
Sim_eval	25.5	25.1	23.1	26.4	**19.9**
Real_eval	22.6	23.0	21.6	22.9	**18.9**
Avg.	24.1	24.1	22.4	24.7	**19.4**

Table 3.20: WER (%) for each noise condition in CHiME-3 dataset with the baseline features and the proposed feature extraction.

	Dev Data				**Eval Data**			
Cond.	Sim		Real		Sim		Real	
	MFB	CVAE	MFB	CVAE	MFB	CVAE	MFB	CVAE
BUS	12.6	**10.6**	14.2	**12.6**	18.3	**13.8**	29.2	**23.6**
CAF	17.0	**15.8**	11.4	**10.2**	26.3	**21.4**	23.7	**19.1**
PED	12.0	**10.0**	8.5	**7.4**	29.1	**21.0**	21.1	**18.6**
STR	15.7	**13.2**	12.3	**10.7**	28.3	**23.4**	16.4	**14.3**

Table 3.21: Statistical significance of performance improvements for the proposed method over the baseline MFB system using confidence interval and the probability of improvement (POI) on Aurora-4 dataset. [10].

Test Cond.	Confidence Interval		POI (%)
	MFB	CVAE	
A	[4.0, 5.5]	[3.7, 5.0]	95.0
B	[7.4, 9.7]	[7.1, 9.5]	81.8
C	[7.8, 10.8]	[6.4, 8.9]	100.0
D	[17.5, 23.1]	[16.3, 21.5]	95.3
Avg	–	–	90.0

in the Aurora-4 multi-condition training. The bootstrap estimate of CI is similar for MFB and the proposed CVAE approach. The POI of CVAE system over the MFB is quite high for almost all test conditions, with average POI being 90%.

3.5.4.2 *Choice of number of filters and Filter Selection*

The ASR results thus far in this section are reported with only 2 separable modulation filters with filter size as 5×5. In this subsection, the effect of different number of filters on the ASR performance is analyzed without any filter selection (the 2-D filters obtained from the CVAE model are applied directly). These results are reported in Table 3.22. From the ASR results, it can be observed that the ASR results do not improve for two of the three datasets considered when the number of modulation filters is increased beyond 2. Hence, only 2 modulation filters are used in all the other ASR experiments reported in this section.

3.5.4.3 *Modulation filter selection for ASR*

In ASR experiments, the 2-D filters based on $\boldsymbol{r}_2, \boldsymbol{s}_1$ and $\boldsymbol{r}_2, \boldsymbol{s}_2$ combinations are used. While this was partly motivated by the previous studies on human perception of modulation [33, 25], this choice is validated with

Table 3.22: Performance (Average WER (%) for different number of modulation filters without any filter selection.

No. of 2-D Filters	Aurora-4	REVERB	CHiME-3
2	12.1	**16.3**	**14.9**
3	11.9	16.8	16.4
4	11.6	16.4	16.2
6	**11.5**	16.8	15.9
8	12.0	17.4	16.1

Table 3.23: Average WER (%) with all the filter combinations of Aurora-4, REVERB and CHiME-3 datasets.

Mod. Filter	Aurora-4	REVERB	CHiME-3
(r_1,s_1), (r_1,s_2)	13.2	18.0	15.1
(r_1,s_1), (r_2,s_2)	12.1	16.3	**14.9**
(r_2,s_1), (r_2,s_2)	**11.2**	**16.1**	15.3
(r_1,s_2), (r_2,s_1)	12.2	16.4	15.0

Table 3.24: ASR Performance of proposed 2-D Rank-1 modulation filters and 2-D full-rank joint modulation filters.

Filter Learning Constraint (WER in %)	
Full rank	12.3
Rank-1 (with filter selection)	11.2
Fusion (Feat.) Full-rank + Rank-1	11.2
Fusion (Score) Full-rank + Rank-1	**11.0**

a set of ASR experiments on multi-condition Aurora-4, REVERB and CHiME-3 datasets. The ASR results using all the four combinations of rate $(\boldsymbol{r}_1, \boldsymbol{r}_2)$ and scale filters $(\boldsymbol{s}_1, \boldsymbol{s}_2)$ on the databases are shown in Table 3.23. As seen here, the ASR performance in these experiments validate the claim that the important modulations for ASR lie in the bandpass region of temporal domain and the entire modulation range of the spectral domain, much similar to the human perceptual experiments [25]. In REVERB dataset, a similar trend is also observed. This further validates the filter selection criteria discussed in Section 3.2.2, 3.3.3, 3.4.3. In the CHiME-3 dataset, both rate filters $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are found to have band pass frequency responses with slightly different bandpass regions. In the ASR experiments, inclusion of rate filter $\boldsymbol{r}_2$ instead of $\boldsymbol{r}_1$ results in a degradation in performance.

3.5.4.4 *Full-rank vs. Rank-1 filter learning*

We compare the proposed feature learning approach using rank-1 constraint with features obtained from unconstrained joint 2-D CVAE modulation filters in Table 3.24. The full-rank 2-D filters are learnt using the similar cost function as in Eq. 3.17 except that separable rank-1 constraint is now removed. From the results, it is observed that the full-rank features perform worse than rank-1 filters. It is interesting to note that the concept of separable modulation filters has also been observed in ferret auditory cortex [20]. It is also seen that the feature-level fusion of full-rank and rank-1 filters do not yield any ASR improvements while the score-level fusion yields minor improvements. However, the score-level fusion is computationally expensive as one needs to train two separate ASR systems on each of the feature streams.

Table 3.25: Effect of different learning methods to learn two 2-D rank-1 filters in first convolution layer (with the generative model loss function) on the Aurora-4 ASR experiments in terms of WER.

Discriminative model	WER (%)
CNN (supervised learning of filters)	11.3
Generative model	**WER (%)**
CRBM (Boltzmann Likelihood) [1]	12.2
CAE (Only MSE loss in Eq. 3.17) [2]	11.8
GAN (MSE loss + Adversarial loss) [2]	11.6
Plain CVAE (Only MSE and Latent Loss in Eq. 3.17)	11.6
Prop. CVAE (All four terms in loss function of Eq. 3.17)	**11.2**

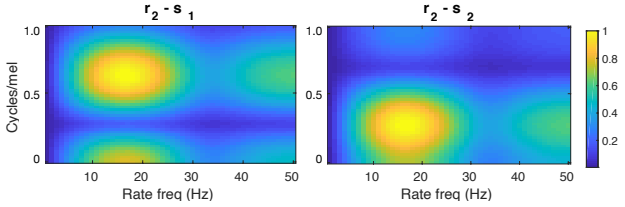


Fig. 3.16: Two 2-D filters $(\boldsymbol{r}_2, \boldsymbol{s}_1)$ and $(\boldsymbol{r}_2, \boldsymbol{s}_2)$ used in feature extraction for ASR learned from the REVERB Challenge database (8 channels) in CVAE.

3.5.4.5 *Choice of generative model loss function for filter learning*

All the ASR experiments reported thus far use the filter learning paradigm of the CVAE model with the total loss function defined in Eq. 3.17. In this subsection, the various components of the loss function are teased apart and we analyze their impact for ASR performance on the Aurora-4 task. Specifically, *two 2-D rank-1 filters* are learnt in first convolutional layer using the conventional CAE model (having only the MSE loss function) as well as the vanilla CVAE model (without the L2 convolution loss or the L1 sparsity loss and having equal weight for the latent loss and the MSE loss). The proposed CVAE framework for filter learning is also compared with the previously proposed convolutional RBM (CRBM) based approach (Section 3.2) and generative adversarial network (GAN) based approach (Section 3.4) with two 2-D rank-1 filters in first convolution layer. All the models are trained in the same framework for learning two filters and use the same training dataset. These ASR experiments on Aurora-4 are reported in Table 3.25. In addition, these generative model approaches are compared with a discriminative model CNN, where 64 filters in a convolution layer are learned jointly with 4-layer DNN for the ASR task. The results indicate that the generative modeling framework of CVAE (Eq. 3.17) provides the best ASR performance in comparison with other choices and it improves over the previously proposed CRBM and GAN framework discussed in Section 3.3 and 3.4. Note that for CRBM, CAE and GAN, two 2-D rank-1 filters are learnt simultaneously in initial convolution layer (instead of one filter at a time) for comparison with the CVAE approach, with primary advantage of being able to able to learn multiple 2-D filters jointly in a single model training. Also, the proposed approach performs marginally better than the supervised CNN approach of learning filters. The features learned from unsupervised model could also be used in the CNN framework to further improve the ASR performance.

3.5.4.6 *Domain specific versus cross-domain filter learning*

In Figure 3.15, the 2-D frequency response of the filters used for feature extraction learned from multi-condition Aurora-4 dataset is shown. The response of the 2-D filters learned from REVERB challenge and CHiME-3 dataset are shown in Figure 3.16 and Figure 3.17 respectively. Comparing the frequency response of the filters learned from each of these datasets, it is observed that the rate filter $\boldsymbol{r}_2$ has relatively lower

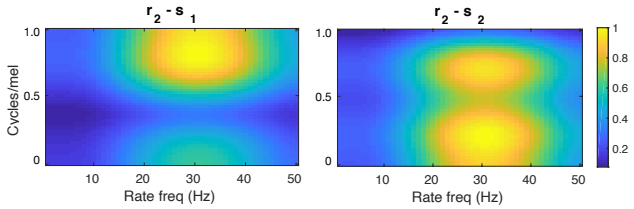


Fig. 3.17: Two 2-D filters $(\boldsymbol{r}_2, \boldsymbol{s}_1)$ and $(\boldsymbol{r}_2, \boldsymbol{s}_2)$ used in feature extraction for ASR with CHiME-3 database.

Table 3.26: WER (%) for cross-domain ASR experiments.

Filters Learned on	**ASR Trained and Tested on**		
	Aurora-4	REVERB	CHiME-3
Aurora-4	11.2	16.2	15.0
REVERB	11.0	16.1	15.2
CHiME-3	11.2	16.1	15.3

frequency range in Aurora-4 compared to the other two datasets. In the case of scale filter characteristics, the filters learned from CHiME-3 dataset show higher rate and scale frequency range.

In a subsequent analysis, a cross-domain ASR experiment is performed, i.e., the filters are learnt from one of the datasets (either Aurora-4, REVERB Challenge or CHiME-3) and these filters are used to train/test ASR on the other two datasets. The results of these cross-domain filter learning experiments are reported in Table 3.26. The rows in the table show the database used to learn filters and the columns show the dataset used to train and test the ASR. The performance reported in this table are the average WER on each of the datasets. The results shown in Table 3.26 illustrate that the filter learning process is relatively robust to the domain of the training data used in the CVAE model. This is a key result and it suggests that ASR system is not affected by the minor changes in the filter characteristics observed in Figure 3.15, 3.16 and 3.17. One could assume that the spectro-temporal modulations in noisy/reverberant speech to be composed of components from clean speech and those from noise/reverberation. The experiments in Table 3.26 lead us to hypothesize that the proposed CVAE based generative model is able to effectively capture the key speech modulations and ignore the spectro-temporal modulations of noise/reverberation. It is also hypothesized that the filters learned using Aurora-4 use more training conditions like 6 different noisy conditions and 2 microphone conditions compared to variabilities in the CHiME-3 dataset. Hence, Aurora-4 filters perform the best for CHiME-3. Using the filters learnt from REVERB dataset, the performance on Aurora-4 is improved mainly due to improvements in microphone mis-match condition *D*.

3.5.4.7 *Semi-supervised ASR training*

In addition to the ASR experiments with full training data, a case is considered when only a fraction of the available training data is labeled. The 2-D filters are learned using CVAE from the entire unlabeled training data and applied for ASR training with the labeled data. The ASR experiments are reported with reduced labeled data (70, 50 and 30% random selection of the original training data). These experiments are shown in Figure 3.18.

It can be observed that the baseline ASR system has a drastic degradation in performance when the amount of training data is reduced. The features using the CVAE model are more resilient to the presence of reduced amounts of labelled training data (a relative improvement of 25% over the baseline for the case with 30% labelled training data). For example, even with 30% of labeled training data, the CVAE feature based ASR attains a WER that is better than the baseline ASR system with 70% labeled training data. In addition, comparison with the CRBM-2D feature learning approach is also shown in the plot, where it can be observed that the CVAE approach performs significantly better than the CRBM-2D approach in all cases of different amount of labeled training data.

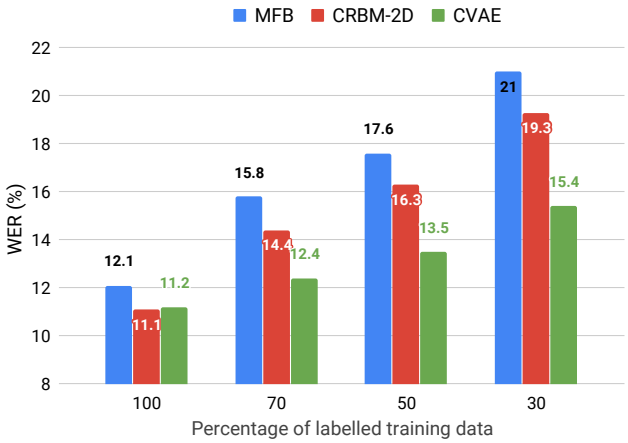


Fig. 3.18: ASR performance in terms of WER (%) in Aurora-4 database (average of 14 test conditions) for multi condition training using lesser amount of labeled training data (70%, 50%, 30%). Here 100% corresponds to 14 h of training data.

3.5.4.8 *Brief section summary*

- Using modified cost function in the CVAE framework that encourages the filters learned to be irredundant and the latent representation to be sparse.
- Exploring the presence of universal modulation characteristics which can be learned from any one of the corpus and generalized to other datasets.
- Discussion on choice of other network parameters and its effect on ASR.
- Comparison of CVAE modified cost function approach with 1-D CRBM and 2-D CRBM residual approach in terms of noisy ASR performance reveal 1-D CRBM as the best filter learning approach among all. In 2-D approach with multiple filter learning performed simultaneously, CVAE-2D improves over CRBM-2D.

3.6 Skip-Connection Based Learning with VAE

To learn multiple irredundant filters, the approach of the skip-connection based model is explored for modulation filter learning. The motivation of using skip-connection is primarily to incorporate residual operation in layers. This approach can learn multiple irredundant filters and overcome the previous residual approach with multiple training of the generative model needed to learn multiple filters (Section 3.2, 3.3, 3.4) and also avoids the use of $(\boldsymbol{r}_1 * \boldsymbol{r}_2)$ or $(\boldsymbol{s}_1 * \boldsymbol{s}_2)$ terms in the loss function E_{fr} or E_{fs} in Eq. 3.17. The CVAE model is used with skip connection in the initial layers of the encoder.

3.6.1 *Convolutional VAE and Filter Learning*

The block diagram of the VAE model used for filter learning is shown in Figure 3.19. As outlined in Figure 3.19, the input to CVAE are the 1-D temporal (spectral) trajectories of log mel spectrograms for rate (scale) filter learning. For rate filter learning, the dimension of the 1-D trajectory as the input to CVAE is 1×150 (equivalent to 1.5 s of speech), and for scale filter learning it is 1×40 (corresponding to all 40 mel bands). Table 3.27 gives the details of the CVAE architecture used in this work. The first layer of the 'encoder' is a convolutional layer with number of kernels = 1 and kernel size as 1×5. Let the output of this layer be $\boldsymbol{h}_1$, where $\boldsymbol{h}_1 = \boldsymbol{x} * \boldsymbol{r}_1$ for rate filter learning and $\boldsymbol{h}_1 = \boldsymbol{x} * \boldsymbol{s}_1$ for scale filter learning. In order to learn multiple irredundant filters, the contribution of the learnt kernel is removed from the input $\boldsymbol{x}$ using skip connection and we feed the *tanh* of the residual $(\boldsymbol{x} - \boldsymbol{h}_1)$ to the next convolutional layer. The next layer (also having one kernel) then learns the modulation characteristics from the residual and generates output

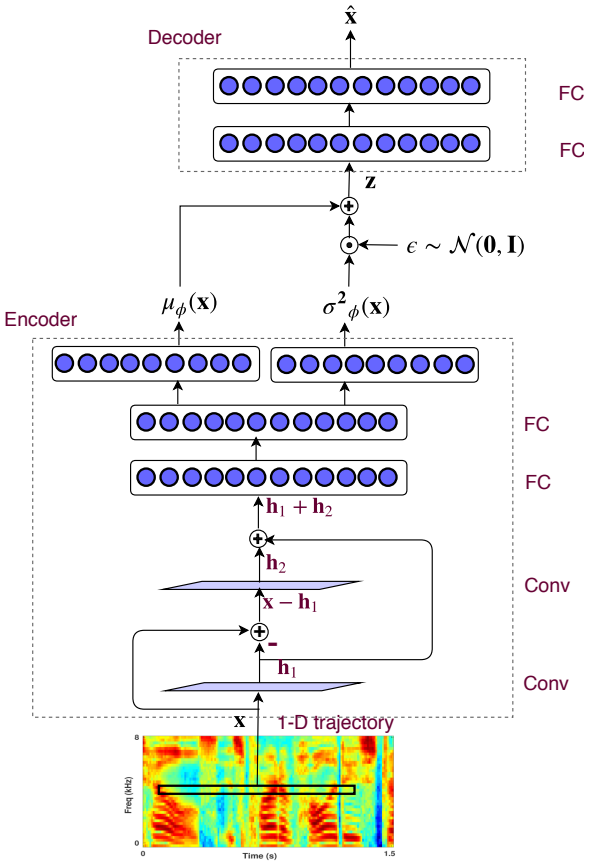


Fig. 3.19: Block schematic (bottom-up) of rate filter learning with CVAE using skip connections in Encoder for residual learning. Here FC denotes fully connected layer, Conv denotes convolution layer.

Table 3.27: The architecture of the CVAE model used for rate and scale filter learning.

Number of layers - encoder	Conv: 2, FC: 2
Number of layers - decoder	FC: 2
Number of kernels, kernel size in Conv	1, 1×5
Activation function	tanh
Mini-batch size	30000
Learning rate, Optimization	0.0001, Adam [51]
Number of nodes in FC - rate / scale	150 / 40
Latent Vector $\boldsymbol{z}$ Dimension - rate / scale	120 / 28

$\boldsymbol{h}_2 = (\boldsymbol{x} - \boldsymbol{h}_1) * \boldsymbol{r}_2$ for rate filter learning and $\boldsymbol{h}_2 = (\boldsymbol{x} - \boldsymbol{h}_1) * \boldsymbol{s}_2$ for scale filter learning. The two filtered (hidden) representations are added and the non-linear activations $tanh(\boldsymbol{h}_1 + \boldsymbol{h}_2)$ are fed to FC layers of the encoder. The latent vector $\boldsymbol{z}$ is calculated from the encoder output as discussed in Section 3.5.1. The decoder then reconstructs the 1-D trajectory from the $\boldsymbol{z}$ [65]. This skip-connection based residual appraoch is similar to the one used in CRBM framework in Section 3.2.

3.6.1.1 *Filter Characteristics*

The CVAE is trained using multi condition data of different databases separately. The training is performed with random initialization of the weights and the model is trained to learn modulation filter characteristics from data. Figure 3.20 shows the normalized magnitude frequency response of the filters learned using

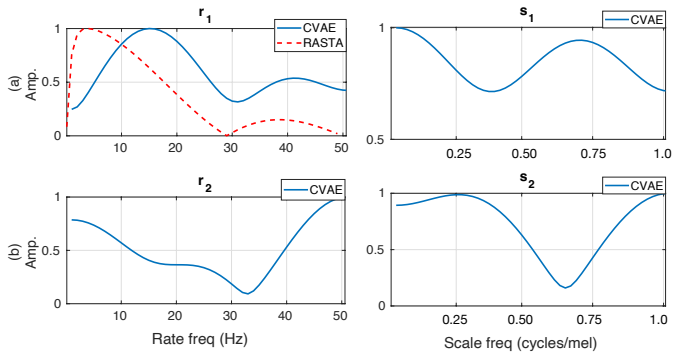


Fig. 3.20: Frequency modulation characteristics of the two rate ($\boldsymbol{r}_1, \boldsymbol{r}_2$) and scale filters ($\boldsymbol{s}_1, \boldsymbol{s}_2$) learned from the CVAE model with skip connections using the Aurora-4 dataset. The RASTA filter is also shown in the $\boldsymbol{r}_1$ plot for reference.

Aurora-4 database. The x axes for the rate and scale filters are rate frequencies (measured in Hz) and scale frequencies (measured in cycles/mel) respectively.

In the analysis, it is observed that the two rate filters learned from the input mel spectrogram have invariably band-pass and band-stop characteristics. The scale filters jointly span the entire spectral modulation space. It is hypothesized that the filters will learn the common underlying representation of all types of input noisy speech, and the overlap of these modulation regions would be representative of modulations of clean speech. The first row of Fig. 3.20 also shows the comparison with the RASTA filter [33]. As seen here, the learnt data driven rate filter resembles the perceptual knowledge driven RASTA filter. Also, it is interesting to note that the range of modulations captured by $\boldsymbol{r}_1, \boldsymbol{r}_2$ and $\boldsymbol{s}_1, \boldsymbol{s}_2$ are quite similar to the modulation filters found in human perceptual studies [25].

3.6.1.2 *Comparison with other architectures*

The previous work in this direction to learn irredundant 1-D and 2-D modulation filters using residual approach has been discussed earlier in Sections 3.2, 3.3, 3.4. In these works, the filter learning is performed by learning one filter at a time using CRBM, CAE or cGAN. The residual is computed externally and is fed to the network for the learning of second filter. Thus, the filters are not jointly optimized. The CVAE network uses a single filter learning framework with skip connections.

3.6.1.3 *Feature extraction for ASR*

The features for ASR are derived by filtering the log mel spectrogram using filters learned from the proposed approach. In this work also, the rate filter with bandpass characteristic ($\boldsymbol{r}_1$ shown in Fig. 3.20) is selected as it has been observed earlier to be important for ASR performance [33, 1], while both the scale filters are used for ASR. The log mel spectrograms are filtered using filters ($\boldsymbol{r}_1, \boldsymbol{s}_1$) and ($\boldsymbol{r}_1, \boldsymbol{s}_2$) separately and are concatenated to derive features for ASR.

3.6.2 *Experiments and Results*

3.6.2.1 *Kaldi ASR framework*

The speech recognition Kaldi toolkit [82] is used for building the ASR. The ASR performance of the proposed modulation filtering approach (CVAE) is compared with traditional mel filter bank energy (MFB) features and other features discussed in Section 2.1. The performance comparison is also done for approaches discussed in Section 3.2 (CRBM-1D), Section 3.3 (CRBM-2D), Section 3.5 (modified cost function in CVAE denoted as CVAE-ModC).

Table 3.28: Word error rate (%) in Aurora-4 database for multi condition training condition with various feature extraction schemes and the CVAE-skip modulation filtering approach.

Cond	MFB	PFB	ETS	CRBM-1D	CRBM-2D	CVAE-ModC	CVAE-skip
A	4.2	4.1	4.5	3.7	4.0	3.5	**3.4**
B	7.8	8.0	8.6	7.3	7.3	7.4	**7.2**
C	8.4	7.8	8.0	7.1	7.6	**6.9**	7.2
D	18.5	19.7	18.8	**16.2**	16.7	17.1	16.8
Avg.	12.1	12.7	12.6	**10.8**	11.1	11.2	11.0

Table 3.29: Word error rate (%) in CHiME-3 Challenge database for multi-condition training (real+simulated) with test data from simulated and real noisy environments.

Test Cond	MFB	PFB	RAS	MHE	CVAE-ModC	CVAE-skip
Sim_dev	14.3	13.7	14.6	14.4	12.4	**12.2**
Real_dev	11.6	12.0	11.8	12.0	10.2	**9.6**
Avg.	13.0	12.9	13.2	13.2	11.3	**10.9**
Sim_eval	25.5	25.1	23.1	26.4	19.9	**19.1**
Real_eval	22.6	23.0	21.6	22.9	18.9	**17.9**
Avg.	24.1	24.1	22.4	24.7	19.4	**18.5**

Table 3.30: WER (%) for each noise condition in CHiME-3 dataset with the baseline features, CVAE-ModC features and the CVAE-skip feature extraction.

	Dev Data					
Cond.	Sim			Real		
	MFB	CVAE-ModC	CVAE-skip	MFB	CVAE-ModC	CVAE-skip
BUS	12.6	10.6	**10.6**	14.2	12.6	**11.5**
CAF	17.0	15.8	**15.7**	11.4	10.2	**9.8**
PED	12.0	10.0	**9.9**	8.5	7.4	**7.0**
STR	15.7	13.2	**12.5**	12.3	10.7	**10.3**
	Eval Data					
Cond.	Sim			Real		
	MFB	CVAE-ModC	CVAE-skip	MFB	CVAE-ModC	CVAE-skip
BUS	18.3	13.8	**13.3**	29.2	23.6	**22.8**
CAF	26.3	21.4	**20.5**	23.7	19.1	**18.0**
PED	29.1	21.0	**20.6**	21.1	18.6	**16.8**
STR	28.3	23.4	**22.1**	16.4	14.3	**13.9**

3.6.2.2 *Noisy speech recognition*

The Aurora-4 corpus discussed in Section 2.4.1 with multi-condition training setup is used for training and testing. The results of various ASR experiments on Aurora-4 dataset is shown in Table. 3.28. As seen in this Table, the CVAE-skip features provide significant improvements in ASR performance over the baseline system (average relative improvements of 9%). Furthermore, the different unsupervised feature learning methods discussed in this chapter are also reported in the Table. It can be observed that the CRBM-1D approach gives best performance among all, with slight differences in performance among all. In clean and noisy condition with same mic. (A and B), the CVAE-skip gives improved results over other architectures.

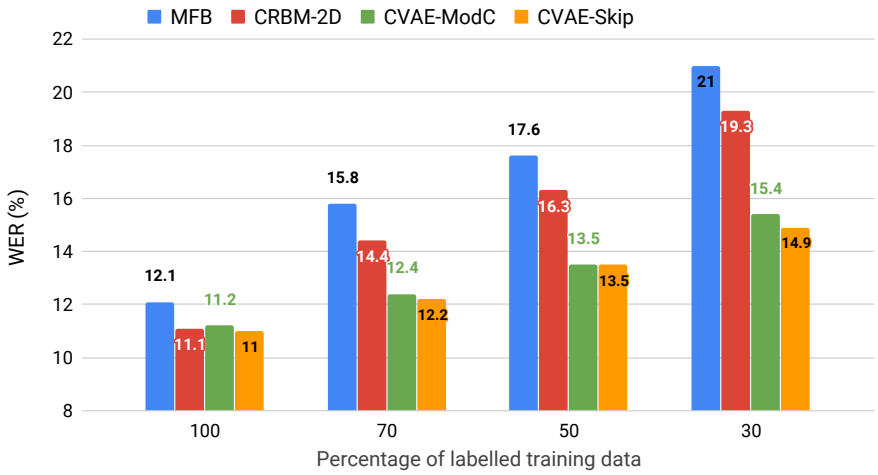


Fig. 3.21: ASR performance in terms of WER (%) in Aurora-4 database (average of 14 test conditions) for multi condition training using lesser amount of labeled training data (70%, 50%, 30%).

3.6.2.3 *Noisy + reverberant speech recognition*

The CHiME-3 corpus discussed in Section 2.4.3 with multi-condition training (real+simulated) is used for training and testing [8]. The results for the CHiME-3 dataset are reported in Table 3.29. The CVAE-skip approach to feature extraction provides significant improvements over the baseline system as well as the other noise robust front-ends considered here. On the average, the CVAE-skip approach provides relative improvements of 16% in the development set and 23% in the evaluation set. The detailed results on different noises in CHiME-3 are reported in Table 3.30. For all the noise conditions in CHiME-3 in simulated and real environments, the discussed approach shows significant improvements over the baseline MFB features. In the evaluation dataset, the relative improvements over the baseline features for most of the noise conditions are above 22%. Furthermore, among the two different unsupervised feature learning methods using CVAE discussed in this chapter (modified cost function as CVAE-ModC and skip-connection based approach), CVAE-skip gives best performance among all, with considerable improvements over the CVAE-ModC approach.

3.6.2.4 *Semi-supervised training*

For semi-supervised ASR training, the Aurora-4 training set up is used with 70, 50 and 30% of the labeled training data. The modulation filters are learnt using full unsupervised training data. The performance comparison of ASR with semi-supervised training is shown in Figure 3.21 for MFB features (as MFB features performed relatively better than other features in Table 3.28), CRBM-2D features, CVAE-ModC features and the CVAE-skip feature scheme (average WER of all 14 test data conditions). These results indicate that the CVAE-skip features are much more resilient to reduced amounts of labeled training data as compared to the baseline MFB system (relative improvement of 29% over the baseline for the case with 30% labeled training data). In addition, CVAE-skip features perform significantly better than the CRBM-2D features in reduced data conditions and performs slightly better than CVAE-ModC features.

3.6.2.5 *Relationship of our approaches with SAGE algorithm [27]*

In space-alternating generalised expectation-maximisation (SAGE) algorithm, if θ is the parameter vector in p-dimensional space , and S is an index set which is subset of the set $\{1, 2, ..., p\}$ with cardinality m. Then, let us use θ_S to denote the m dimensional vector consisting of the m elements of θ and $\theta_{\hat{S}}$ to be the $p - m$ dimensional vector consisting of the remaining elements of θ. In the "group" ascent method, one sequences through different index sets $S = S^i$ and updates only the elements θ_S of θ while holding the

other parameters $\theta_{\hat{S}}$ fixed through conditional log-likelihood paradigm [27].

While this SAGE method learns the partial dimensions of the learnable parameters through EM in alternating fashion, our first residual approach doesn't incorporate residual within the learning framework. After a set of parameters is learnt, the residual is computed externally, with residual spectrogram being fed to the model again for a new training. With other two methods of skip-connection based filter learning and modified cost function method, it can be related with SAGE to some extent to learn the filters jointly.

3.6.2.6 *Brief section summary*

- Obtaining multiple irredundant data-driven modulation filters using skip connection based approach in deep variational network.
- Illustrating robustness in noisy and reverberant conditions using the proposed modulation filtering scheme.
- Comparing all unsupervised feature learning approaches for noisy speech recognition reveal CRBM-1D as the best approach.
- Comparing the two CVAE based filter learning approaches for noisy+reverberant ASR reveal CVAE-skip performing better.

3.7 Representation Learning Using VAE From Raw Waveform

In the work discussed till now, mel spectrogram has been used as the initial speech representation and modulation filters are learnt from it using generative models in unsupervised framework. However, using the 'mel' spectrogram may not be the optimal representation to begin. In this section, we attempt representation learning from raw waveform and without labeled data. A deep representation learning approach is proposed using the raw speech waveform in an unsupervised learning paradigm.

3.7.1 *Acoustic Filterbank Learning*

This section describes the acoustic filterbank learning using CVAE. The use of CVAE is motivated by the goal of learning filterbank (FB) in an unsupervised manner. The kernels (first layer weights) of the deep CVAE trained using raw input are interpreted as the acoustic filters learned from the data.

The acoustic FB layer is expanded in Fig.4.2(b). First, a frame of length 400 samples is taken from raw waveform (corresponding to 25 ms of signal sampled at 16 kHz), and the raw waveform is convolved with the set of 80 filters. The kernels (filters) of the convolution layer are modeled using cosine-modulated Gaussian function as:

$$w_n(t) = \cos 2\pi\mu_n t \times \exp\left(-t^2\mu_n^2/2\right) \tag{3.19}$$

where $w_n(t)$ is the n-th filter impulse response at time t, μ_n is the frequency of the nth filter's cosine function, which represents the mean of the corresponding Gaussian function in frequency domain (center frequency). The number of filter taps (length of filter impulse response) is fixed to $N = 129$, corresponding to 8 ms which has been found to be sufficient to capture temporal variations of speech signal [61]. The parametric approach to filterbank (FB) learning with Gaussian window generates filters with a smooth response in both time and frequency domain and allows the filters to be separable as well. In order to preserve the positive range of frequency values $(0 - 8\text{kHz})$, the μ is chosen to be the sigmoid of a real number (λ) which is scaled, i.e. $\mu = 8000 \times \sigma(\lambda)$ and λ is iteratively updated.

The output of the acoustic FB layer has 80 feature maps, corresponding to convolution of each input frame with each of the 80 filters. Max pooling is applied to the convolved output with pooling kernel of size 8 and pool stride of 4, followed by square operation, sum and logarithmic compression, thereby producing 1×80 sized frame level feature vector. The window is then shifted (120 times) around the raw waveform by a hop size of 10ms and this convolution is repeated to produce patches of size 80×120.

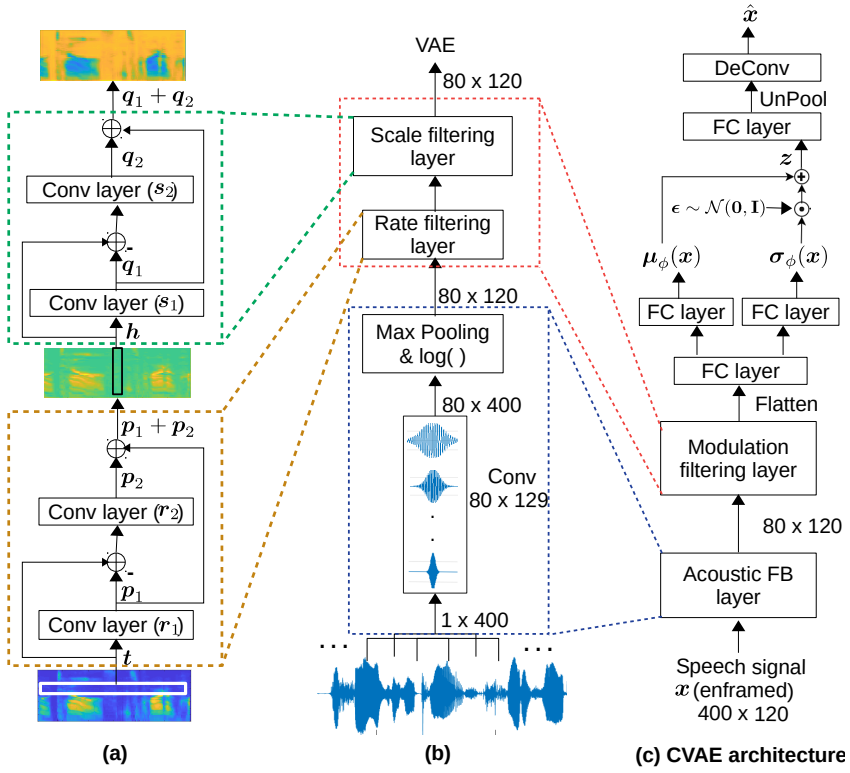


Fig. 3.22: Block diagram of CVAE architecture in (c) to learn acoustic filters in Acoustic FB layer, and modulation filters in Modulation filtering layer. (a) shows expanded modulation filtering layer, (b) shows expanded acoustic FB layer.

3.7.2 *Modulation Filter Learning*

The second layer of the encoder is the modulation filtering layer as shown in Fig. 3.22(c). As outlined in its expanded block (Fig. 3.22(a)), the modulation filtering layer comprises of rate filtering layer followed by scale filtering layer, each of which employs a skip connection based filter learning architecture as discussed in Sec. 3.6.1 [5]. The filters of the convolution layers in rate/scale filtering layer trained using spectrogram trajectories (of previous layer output) are interpreted as the modulation filters.

For rate (scale) filtering layer, the inputs are the temporal (spectral) trajectories of the time-frequency representation obtained from the previous layer output (80×120). The dimension of the 1-D trajectory for rate filtering, denoted as $\boldsymbol{t}$, is 1×120 (equivalent to 1.2 s of speech), and for scale filter learning it is 80×1 (corresponding to all 80 frequency bands). The kernel size is 1×5 in each convolution layer. The resultant output representation is then flattened to be fed to the fully-connected (FC) layer of 120 nodes in the CVAE model (Fig. 3.22(c)). The latent vector $\boldsymbol{z}$ is calculated from the encoder output and the network is trained with the objective of minimizing total loss function calculated as:

$$E_{Total} = \alpha E_{MSE} + \beta E_{MSE-acoustic} + \gamma E_{Latent} \tag{3.20}$$

where E_{MSE} is the mean squared error between input $\boldsymbol{x}$ and the reconstructed output $\hat{\boldsymbol{x}}$, $E_{MSE-acoustic}$ is the mean squared error between the acoustic FB layer output (time-frequency representation) and the reconstructed time-frequency representation in decoder (before deconvolution layer), and E_{Latent} is the latent loss of encoder (KL divergence of $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ with unit Gaussian distribution $p(\boldsymbol{z})$). The values of $\alpha = 0.01$, $\beta = 1.0$ and $\gamma = 0.01$ are used in the experiments. The decoder has one fully connected layer of 120 nodes, followed by an unpooling operation and deconvolution (for reversing the operations performed by the acoustic FB layer of the encoder).

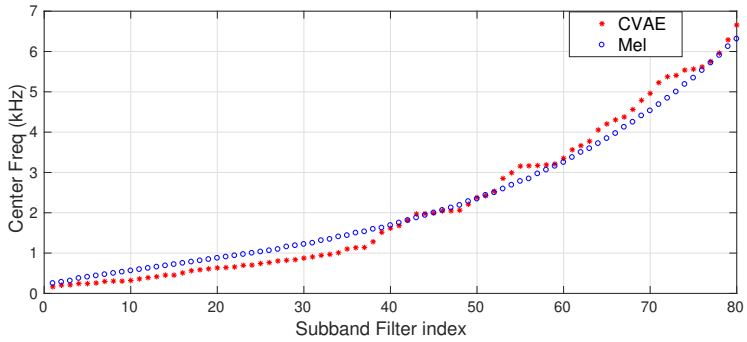


Fig. 3.23: Comparison of center frequency of filterbank learnt using CVAE with clean training data from Aurora-4 dataset, with center frequencies of mel filterbank.

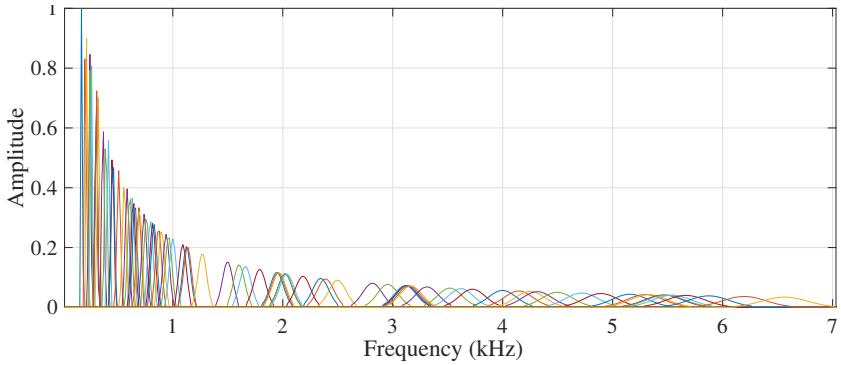


Fig. 3.24: Frequency response of acoustic filterbank learnt using CVAE with clean training data from Aurora-4 dataset.

3.7.2.1 *Filter characteristics*

The acoustic filters in the acoustic FB layer and the filters in the modulation filtering layer are iteratively updated using the gradients of the total loss function and Adam optimizer [51]. The CVAE is trained using data of different databases separately. The training is started with random initialization of the filters and the CVAE learns filter characteristics from data.

Fig. 3.23 shows the the center frequency (μ_n values sorted in ascending order) of the acoustic filters using clean Aurora-4 database and this is compared with the center frequency of the mel filterbank. As can be observed, the proposed filterbank also has nonlinear relationship between center frequencies and the filter index with more number of filters in lower frequencies compared to higher frequencies, similar to traditional acoustic filterbanks [18, 80]. The normalized magnitude response of the learnt filterbank is also plotted for reference in Fig. 3.24. It can be observed that the frequency response magnitude decreases as we go towards higher frequency (because of variance as scaling factor in Fourier transform).

The time-frequency representation obtained from the learnt filterbank is shown in Fig. 3.25(c). The log mel spectrogram is also plotted for reference in Fig. 3.25(b). It can be observed that the obtained representation preserves all information such as formant contours, voiced and unvoiced sounds, even when filters are learnt with a fully unsupervised objective.

3.7.2.2 *Feature extraction for ASR*

The features for ASR are derived by filtering the raw speech waveforms with learnt acoustic filterbank, followed by filtering the time-frequency representation using modulation filters learnt from the discussed CVAE architecture. In this work, the rate filter ($\boldsymbol{r}_2$) with bandpass characteristic is selected as it has been observed earlier (Sections 3.2.2, 3.3.3, 3.4.3) to be important for ASR task [5, 33, 1], while both the scale

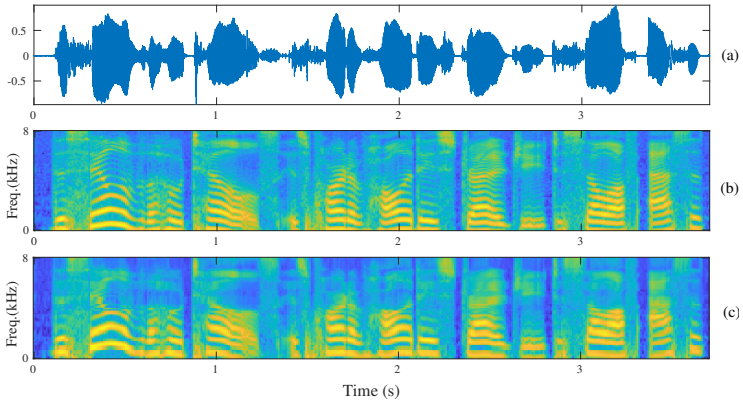


Fig. 3.25: (a) Speech signal, (b) log mel spectrogram (c) spectrogram using learnt cosine-modulated Gaussian filterbank.

Table 3.31: ASR performance comparison for time-frequency representations with different acoustic filterbanks.

Cond	MFB	CRBM[88]	CVAE-A
	Clean Training (Multi condition Training)		
A	3.4 (4.2)	3.4 (3.6)	3.1 (3.8)
B	18.9 (7.8)	23.0 (8.4)	18.7 (7.8)
C	15.3 (8.4)	20.1 (7.2)	14.0 (8.0)
D	35.2 (18.5)	40.0 (19.4)	36.0 (18.8)
Avg.	**24.7** (**12.1**)	28.7 (12.7)	**24.6** (**12.2**)

Table 3.32: Word error rate (%) in Aurora-4 database for clean training condition with various feature extraction schemes.

Cond	MFB	PFB	ETS	MHE	CRBM-1D	CRBM-2D	CVAE-ModC	CVAE-A,M
A	3.4	3.3	3.2	3.5	3.3	3.2	3.0	**2.9**
B	18.9	16.2	16.3	17.4	**13.6**	13.8	15.9	14.4
C	15.3	**11.7**	14.5	14.6	16.0	13.0	13.6	12.9
D	35.2	32.8	32.0	35.4	**29.0**	29.9	33.0	31.9
Avg.	24.7	22.1	21.9	23.9	**19.4**	19.9	22.1	20.6

filters are used. Hence, the obtained time-frequency representation from the acoustic FB layer is filtered using filters $(\boldsymbol{r}_2, \boldsymbol{s}_1)$ and $(\boldsymbol{r}_2, \boldsymbol{s}_2)$ separately (80 dimensional each) and are concatenated as features for ASR.

3.7.3 *Experiments and Results*

The speech recognition Kaldi toolkit [82] is used for building the ASR on two datasets, Aurora-4 and CHiME-3 respectively. For each dataset, the ASR performance with the discussed approach of filtered representation through acoustic FB and modulation filters (CVAE-A,M) is compared with traditional mel filterbank energy (MFB) features along with other baseline features discussed in Section 2.1, with CRBM-1D approach discussed in Section 3.2, CRBM-2D discussed in Section 3.3, modified cost function based CVAE-ModC features discussed in Section 3.5, skip-connection based modulation filter learning approach (CVAE-skip) from mel filterbank features discussed in Section 3.6.

Table 3.33: Word error rate (%) in Aurora-4 database for multi-condition training with various feature extraction schemes.

Cond	MFB	PFB	ETS	CRBM-1D	CRBM-2D	CVAE-ModC	CVAE-skip	CVAE-A,M
A	4.2	4.1	4.5	3.7	4.0	3.5	**3.4**	3.5
B	7.8	8.0	8.6	7.3	7.3	7.4	**7.2**	7.3
C	8.4	7.8	8.0	7.1	7.6	**6.9**	7.2	7.4
D	18.5	19.7	18.8	**16.2**	16.7	17.1	16.8	**17.5**
Avg.	12.1	12.7	12.6	**10.8**	11.1	11.2	**11.0**	11.4

Table 3.34: Word error rate (%) in CHiME-3 Challenge database for multi-condition training (real+simulated) with test data from simulated and real noisy environments.

Test Cond	MFB	PFB	RAS	MHE	CVAE-ModC	CVAE-skip	CVAE-A	CVAE-A,M
Sim_dev	14.3	13.7	14.6	14.4	12.4	**12.2**	14.2	12.3
Real_dev	11.6	12.0	11.8	12.0	10.2	**9.6**	11.5	10.0
Avg.	13.0	12.9	13.2	11.3	11.3	**10.9**	12.8	11.1
Sim_eval	25.5	25.1	23.1	26.4	19.9	**19.1**	26.1	19.7
Real_eval	22.6	23.0	21.6	22.9	18.9	**17.9**	22.5	18.6
Avg.	24.1	24.1	22.4	24.7	19.4	**18.5**	24.3	19.1

3.7.3.1 *Noisy speech recognition*

The WSJ Aurora-4 corpus discussed in Section 2.4.1 is used for conducting noisy ASR experiments. As an initial experiment on Aurora-4 dataset, the ASR performance of the time-frequency representation obtained using different acoustic filterbanks is compared in Table 3.31. The acoustic FB layer output of the proposed model (CVAE-A) is compared with MFB and the acoustic FB output learnt in an unsupervised manner from CRBM [88]. It can be observed that CVAE-A features perform similar to MFB features in both training conditions, under all test conditions and is significantly better than the previous approach to filter learning.

The ASR performance for the discussed (CVAE-A,M) features (joint acoustic and modulation filtering) in clean and multi-condition training condition is shown in Table 3.32 and Table 3.33, respectively. As seen in these results, most of the noise robust front-ends do not improve over the baseline mel filterbank (MFB) performance in both clean and multi-condition training. The CVAE-A,M feature extraction scheme provides improvements in ASR performance over the baseline system with average relative improvements of 16% over MFB in clean training, and 6% in multi condition training. All the explored approaches discussed in previous sections are comparable with the current approach CVAE-A,M. However, the CRBM-1D approach provides the best results among all in both clean and multi-condition training.

3.7.3.2 *Noisy + Reverberant speech recognition*

The CHiME-3 corpus discussed in Section 2.4.3 is another dataset used for ASR training. The beamformed audio has been used for filter learning using CVAE, and for ASR training and testing. The results for the CHiME-3 dataset are reported in Table 3.34. The CVAE-A features perform similar to baseline MFB features in ASR. The discussed (CVAE-A,M) approach to feature extraction (joint acoustic and modulation filtering) provides considerable improvements over the MFB as well as the other noise robust front-ends considered here, while being comparable to the CVAE-ModC and CVAE-skip approaches. On the average, the CVAE-A,M approach provides relative improvements of 15% over MFB features in the dev set and 21% in the eval set. However, among the explored approaches, CVAE-skip provides best results among all. The detailed results on different noises in CHiME-3 are reported in Table 3.35. For all the noise conditions in

Table 3.35: WER (%) for each noise condition in CHiME-3 dataset with the baseline features and the proposed feature extraction.

Cond.	**Dev Data**				**Eval Data**			
	Sim		Real		Sim		Real	
	MFB	CVAE-A,M	MFB	CVAE-A,M	MFB	CVAE-A,M	MFB	CVAE-A,M
BUS	12.6	**10.4**	14.2	**11.8**	18.3	**13.6**	29.2	**23.1**
CAF	17.0	**15.6**	11.4	**10.0**	26.3	**21.4**	23.7	**19.1**
PED	12.0	**9.7**	8.5	**7.2**	29.1	**21.5**	21.1	**17.9**
STR	15.7	**13.3**	12.3	**11.1**	28.3	**22.4**	16.4	**14.4**

CHiME-3 in simulated and real environments, the CVAE-A,M approach shows significant improvements over the baseline MFB features. In the eval dataset, the relative improvements over the baseline features for most of the noise conditions are above 20%.

3.7.3.3 *Brief section summary*

- Proposed a CVAE architecture with the initial two layers of convolutions for speech representation learning from raw waveform with unsupervised learning objective.
- The first layer of convolutions performs acoustic FB learning which is shown to have nonlinear frequency resolution similar to mel FB. In ASR tasks, the proposed acoustic filters perform similar to mel filters and improve over previous unsupervised FB learning method.
- The second layer performs modulation filtering using skip-connection. The features based on joint acoustic and modulation filtering are used for ASR.

3.8 Chapter Summary

The performance summary of the unsupervised representation learning approaches is discussed in Tables 3.36 and 3.37 for modulation filtering and acoustic filterbank learning, respectively, on Aurora-4 database for noisy speech recognition. In the modulation filtering approaches summarized in Table 3.36, the results are reported for baseline system (mel spectrogram features) and compared with different aspects under consideration, like comparison between CRBM 1-D and CRBM 2-D filter learning using residual approach computed externally, comparison between 3 different architectures - CRBM, CAE and cGAN for rate filter learning using residual approach computed externally, multiple filter learning jointly using modified cost function and skip-connection based approach in CVAE.

From the Table, it can be observed that the among 1-D and 2-D CRBM models for rate-scale filtering, CRBM 1-D model outperforms the baseline as well as better than CRBM 2-D rank-1 model. Among the 3 different architectures - CRBM, CAE and cGAN for rate filtering, cGAN 1-D performs best in clean training condition, while CRBM 1-D performs best in multi-condition training. Among the three different joint filter learning techniques (residual computed externally, modified cost function and skip connection approach) in CRBM and CVAE, the CRBM 2-D rank-1 with residual computed externally approach outperforms other two in clean training condition, while in multi-condition, all the three approach are comparable in terms of ASR performance.

For the acoustic filterbank learning from raw waveform and modulation filtering in unsupervised manner reported in Table 3.37, the baseline is mel spectrogram features. The CVAE-A features from learning of acoustic filterbank using CVAE (without modulation filtering) is comparable to the baseline features. The two-stage approach CVAE-A,M of learning acoustic filterbank followed by modulation filters using skip-connection in CVAE gives considerable improvement in clean training as well as multi-condition training.

Semi-supervised setup - While all the discussed models are trained in unsupervised manner, we have observed performance difference in terms of amount of data needed by ASR for training with features

Table 3.36: Summary of all unsupervised approaches of learning modulation filters and their comparison in terms of noisy speech recognition performance on Aurora-4 dataset.

Model	Cost function	Method of learning modulation filters	Type of filtering	ASR (WER %)	
				Clean	Multi
Baseline	-	-	-	24.7	12.1
CRBM 1-D	Boltzmann likelihood	Separate, Residual externally	Rate-scale	**19.4**	**10.8**
CRBM 2-D				19.9	11.1
CRBM 1-D	Boltzmann likelihood	Separate, Residual externally	Rate	23.0	**11.0**
CAE 1-D	Mean Square error			20.7	12.0
cGAN 1-D	Adversarial loss			**20.3**	11.3
CRBM 2-D	Boltzmann likelihood	Joint, Residual externally	Rate-scale	**19.9**	11.1
CVAE-ModC	All 4 terms (Eq. 3.17)	Joint, Modified cost function		22.1	11.2
CVAE-skip	Variational loss	Joint, Skip-connection		20.5	**11.0**

Table 3.37: Summary of unsupervised approaches of learning acoustic filterbank with modulation filters and their comparison in terms of noisy speech recognition performance on Aurora-4 dataset.

Model	Cost function	Method of learning Ac FB	Method of learning mod. FB	Type of mod. filtering	ASR	
					clean	multi
Baseline	-	-	-	-	24.7	12.1
CVAE-A	Variational (Eq. 3.20)	CVAE on raw	-	-	24.6	12.2
CVAE-A,M	Variational (Eq. 3.20)	CVAE on raw	Joint, skip-conn.	rate-scale	**20.6**	**11.4**

Table 3.38: Summary of unsupervised approaches to learn modulation filters with their comparison in terms of semi-supervised ASR in multi-condition training setup on Aurora-4 dataset.

Model	Method of learning modulation filters	Type of filtering	ASR (WER %)			
			100	70	50	30
Baseline	-	-	12.1	15.8	17.6	21.0
CRBM 2-D	Joint, Residual externally	Rate-scale	11.1	14.4	16.3	19.3
CVAE-ModC	Joint, Modified cost function		11.2	12.4	**13.5**	15.4
CVAE-skip	Joint, Skip-connection		**11.0**	**12.2**	**13.5**	**14.9**

derived through these models. Table 3.38 shows the summary for the semi-supervised case with 100, 70, 50 and 30% of the training data. It can be observed that while all the modulation filtered representations provide improvements over baseline with reduced amounts of data, the CVAE-skip approach gives the best results among all in reduced data conditions.

Training time - Here, we compare the training time for each of the unsupervised model explored in the chapter using all Aurora-4 training data. Note that multiple trainings are needed for different types of learning approaches. For example, the CRBM 1-D training with residual approach learns one rate or scale filter in one training. Hence, 4 trainings are needed for 2 rate and 2 scale filter learning. The modified cost function approach with CVAE learns all the filters in a single CVAE training. The skip-connection based approach learns 2 rate/scale filters in one training, hence 2 trainings are required. Table 3.39 shows the training time for each unsupervised model training. It can be observed that one epoch time of CRBM is very less, since RBM is a single layer model. However, due to multiple training required, the total training time for an epoch is comparable to CVAE time, with CVAE-ModC model requiring highest training time among all.

Table 3.39: Comparison of unsupervised models with respect to training time for an epoch, total number of model training required (with residual, modified cost function, skin-connection) to learn modulation filters.

Model	Training time (s)	Total no. of training	Total time (s)
CRBM 1-D	122	4	488
CRBM 2-D	148	4	592
CVAE-ModC	616	1	616
CVAE-skip	298	2	596

Table 3.40: Summary of unsupervised approaches of learning acoustic filterbank and/or modulation filters and their comparison in terms of speech recognition performance on multi-condition training of CHiME-3 dataset.

Model	Method of learning Ac FB	Method of learning mod. FB	Type of mod. filtering	ASR
Baseline MFB	-	-	-	18.5
CVAE-A	CVAE on raw	-	-	18.5
CVAE-ModC	-	Joint, modified cost function	rate-scale	15.3
CVAE-skip	-	Joint, skip connection	rate-scale	**14.7**
CVAE-A,M	CVAE on raw	Joint, skip-connection	Rate-scale	15.1

The summary for unsupervised representation learning approaches for CHiME-3 dataset (noisy + reverberant speech dataset) is discussed in Table 3.40. When no modulation filtering is performed, the learnt acoustic filterbank representation CVAE-A performs similar to the baseline MFB baseline system. With modulation filtering performed, all the approaches perform better than baseline, with CVAE-skip approach, i.e. skip-connection based modulation filtering over mel filterbank features performs the best among all approaches.

Chapter 4

Supervised Learning of Interpretable Representations

4.1 Introduction

While we explored the unsupervised approaches for representation learning in the last chapter, this chapter explores the approaches to interpretable representation learning in supervised manner. The motivation to explore the supervised representation learning is primarily to interpret and analyze the representations at different stages of the network for the task. Therefore, we attempt a two-step representation learning approach from raw waveform. In particular, an acoustic filterbank learning from raw waveform is carried out in supervised framework for the task (senone classification in ASR or sound classification in urban sound classification), respectively. In addition, a relevance weighting mechanism is proposed that allows the interpretability of the learned representations in the forward propagation itself. We also explore incorporating feedback in learning the relevance weighting where senone embeddings of the previous target are used to learn relevance weights. This is followed by modulation filter learning with relevance weighting.

The rest of the neural network architecture performs the task of acoustic modeling for speech recognition/sound classification. All the model parameters including the acoustic layer parameters, acoustic filterbank relevance sub-network, modulation filters and modulation relevance sub-network are learned in a supervised learning paradigm. The subsequent analysis of the relevance sub-network reveals that the weights from the network contain information regarding the acoustic content of the label. For speech recognition, the relevance weights are also adaptive to the presence of noise in the data.

4.1.1 *Motivation*

One of major motivation for the proposed modeling approach of relevance weighting is the evidence of gain enhancement mechanism in human sensory system. Both physiological and behavioral studies have suggested that stimulus-driven neural activity in the sensory pathways can be modulated in amplitude with attention [48]. The recordings of event-related brain potentials indicate that such sensory gain control or amplification processes play an important role in tasks that involve attention [19]. The combined event-related brain potential and neuroimaging experiments provide strong evidence that attentional gain control operates at an early stage of sensory processing [36]. These evidences support feature selection theories of attention and provide a basis for distinguishing between separate mechanisms of attentional suppression (of unattended inputs) and attentional facilitation (of attended inputs).

The second motivation comes from prior works on Mixture of Experts (MoE) models [45, 47]. For neural networks the work proposed by Shazeer et. al. [99] explored multiple parallel neural networks followed by a gating network which performs a combination of the outputs from the networks. This MoE model showed significant promise for a language modeling task. The self attention module successfully inducted in transformer models [108] also incorporates information from multiple streams using a linear combination.

Inspired by these human and machine learning studies, the proposed relevance weighting attempts to model gain enhancement in a neural architecture. The frequency selectivity observed in auditory system can be modeled as relevance weighting in the acoustic filter bank layer while the cortical layer gain enhancement is attempted by modulating the relevance weights of the modulation filtering layer. While attention [108]

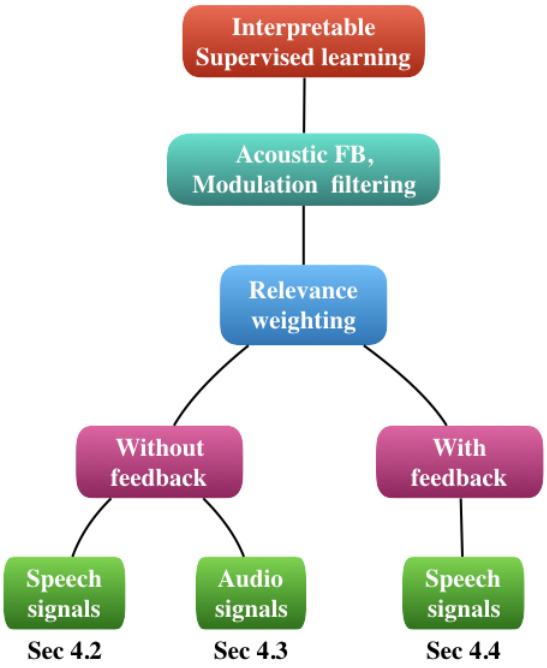


Fig. 4.1: Flowchart of the supervised representation learning approach for speech and audio signals.

and MoE models [99] attempt a combination of input streams that are fed to successive layers, the proposed relevance weighting merely performs a gain enhancement without a linear combination.

The rest of the chapter is organized as follows, with schematic shown in Figure 4.1.

- Section 4.2 describes the two-step representation learning approach using relevance weighting for speech signals, followed by interpretability analysis and experiments.
- Section 4.3 explores the application of similar strategy for audio signals, followed by the analysis and experiments.
- Section 4.4 describes the relevance weighting approach with feedback of target senone embeddings for speech signals.
- Section 4.5 summarizes the chapter.

4.2 Relevance Weighting Based Representation Learning

The block schematic of the relevance weighting based representation learning approach is shown in Figure 4.2. The expanded acoustic filterbank (FB) layer and modulation FB layer are shown in Figure 4.3.

4.2.1 *Acoustic Filterbank Learning with Relevance Weighting*

The first layer of the model performs acoustic filtering from the raw waveforms using a convolutional layer, similar to the one discussed in Section 3.7. The input to the neural network are raw samples windowed into s samples per frame. This frame of raw audio samples is processed with a 1-D convolution using f kernels each of size k. The kernels are modeled using a cosine-modulated Gaussian function [4],

$$g_i(n) = \cos 2\pi\mu_i n \times \exp\left(-n^2\mu_i^2/2\right) \tag{4.1}$$

where $g_i(n)$ is the i-th kernel ($i = 1, .., f$), μ_i is the center frequency of the ith kernel. In order to preserve the positive range of frequency values for μ, we choose the μ to be the sigmoid of a real number which is further scaled to half the sampling frequency of the signal. The mean parameters are updated in a supervised manner for each dataset.

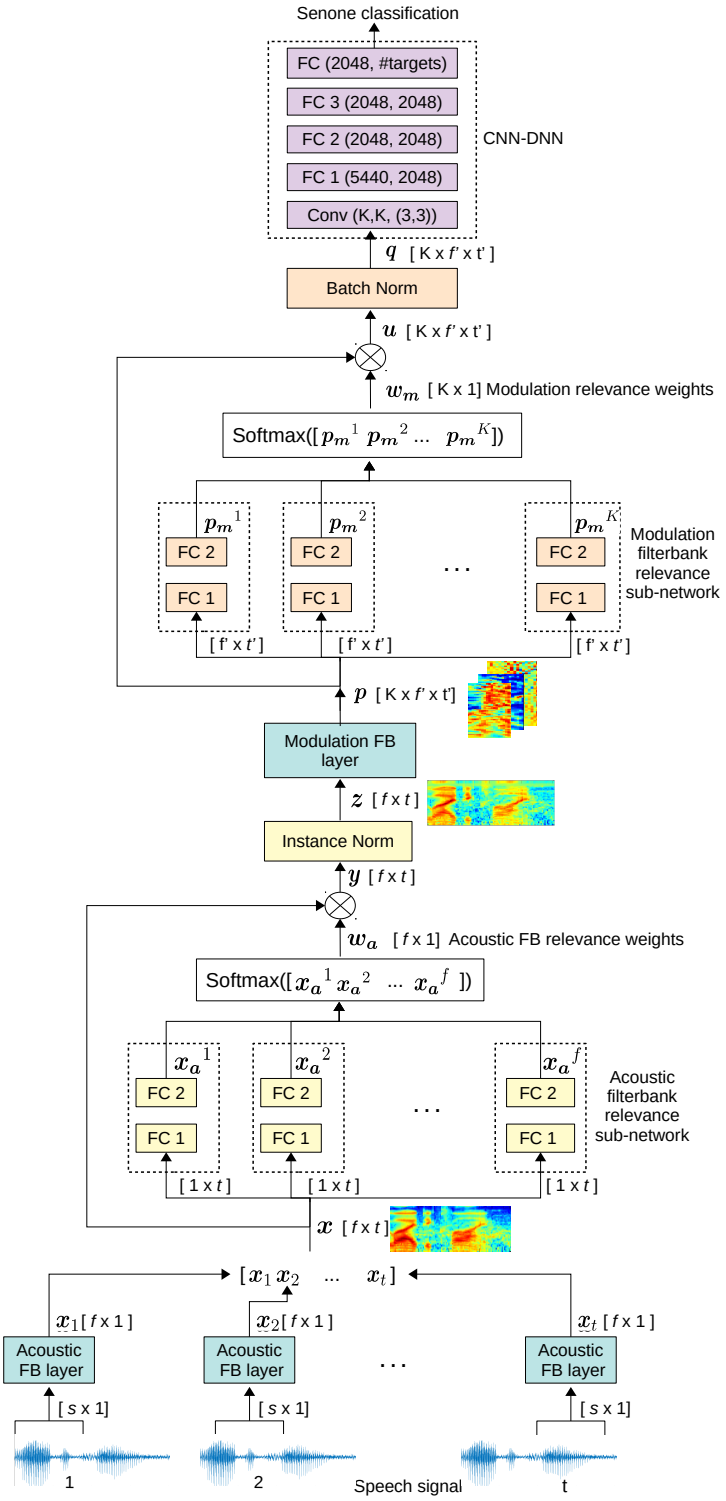


Fig. 4.2: Block diagram of the representation learning approach from raw waveform using relevance weighting approach. Here, FC denotes a fully connected layer and Conv denotes a convolution layer.

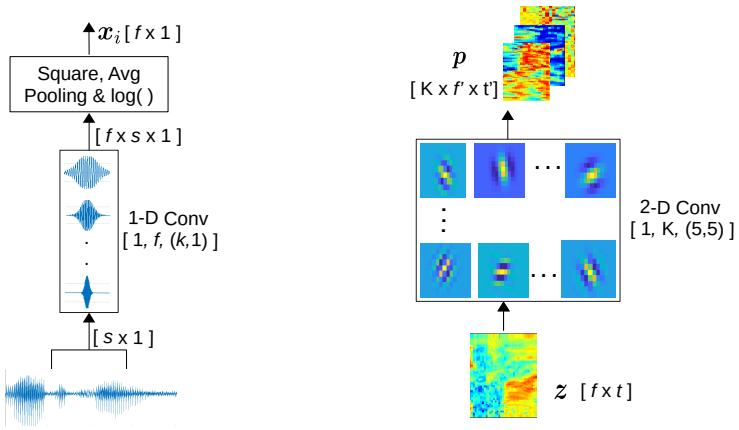


Fig. 4.3: (a) Expanded acoustic filterbank (FB) layer, (b) Expanded modulation FB layer.

The convolution with the cosine-modulated Gaussian filters generates f acoustic feature maps, as shown in Figure 4.3 (a). These outputs are squared, average pooled within each frame and log transformed. This generates $\boldsymbol{x}$ as f dimensional features for each of the t contextual frames. The matrix $\boldsymbol{x}$ is interpreted as the "learned" time-frequency representation.

4.2.1.1 *Relevance weighting*

The relevance weighting paradigm for acoustic FB layer is implemented using a relevance sub-network. This network is fed with the acoustic feature map of dimension $1 \times t$ for each of the f Gaussian kernels (each row of the time-frequency representation $\boldsymbol{x}$). A two layer deep neural network (DNN) with a t dimensional input layer and a scalar output realizes the relevance sub-network. This operation is repeated for all the f acoustic feature maps to generate $\boldsymbol{x_a}$ as f dimensional vector with weights corresponding to each kernel. The relevance weights $\boldsymbol{w_a}$ are generated using the softmax function as,

$$w_a^i = \frac{e^{x_a^i}}{\sum_j e^{x_a^j}}; \quad \text{where } i = 1, 2, ..., f. \tag{4.2}$$

The weights $\boldsymbol{w_a}$ are multiplied with each of the acoustic feature map (rows of $\boldsymbol{x}$) to obtain the relevance weighted time-frequency representation $\boldsymbol{y}$.

The relevance weights in this framework are different from typical attention mechanism [116]. In our framework, relevance weighting is applied over the representation as soft feature selection weights without a linear combination as done in attention models [116].

The first layer outputs ($\boldsymbol{y}$) are also smoothed using instance norm [86, 106]. Let $y_{j,i}$ denote the relevance weighted time-frequency representation for frame j ($j = 1, .., t$) of kernel i ($i = 1, .., f$). The soft weighted output $z_{j,i}$ is given as,

$$z_{j,i} = \frac{y_{j,i} - m_i}{\sqrt{\sigma_i^2 + c}} \tag{4.3}$$

where m_i is the sample mean of $y_{j,i}$ computed over j and σ_i is the sample std. dev. of $y_{j,i}$ computed over j. The constant c acts as a relevance factor and is chosen as $1e^{-4}$. The output of relevance weighting ($\boldsymbol{z}$) is propagated to the subsequent layers for the acoustic modeling.

In the experiments, $t = 101$ is used whose center frame is the senone target for the acoustic model. Also, $f = 80$ kernels each of length $k = 129$ is used. This value of k corresponds to 8 ms in time for a 16 kHz sampled signal which has been found to be sufficient to capture temporal variations of speech signal [61]. The value of s is 400 corresponding to 25ms window length and the frames are shifted every 10ms. The instance norm is not applied globally but rather applied on the given input patch of $t = 101$ time frames

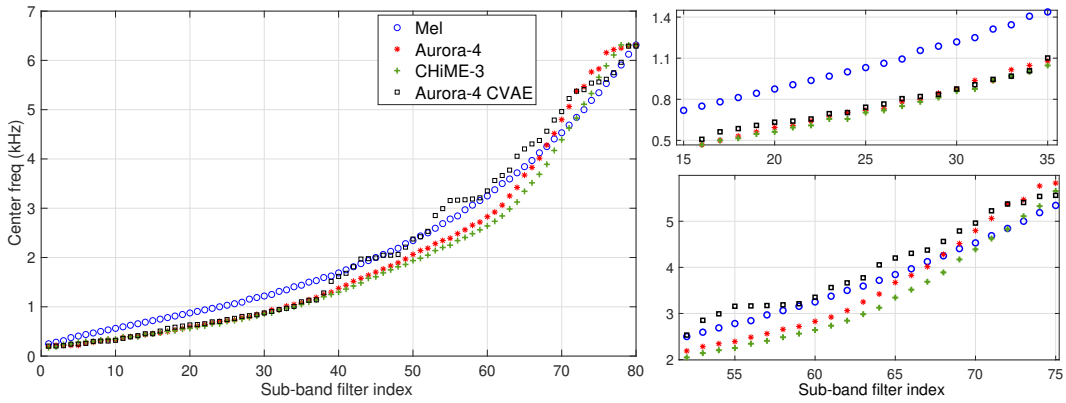


Fig. 4.4: Left: Comparison of center frequency of acoustic filterbank learned using the discussed approach for Aurora-4 and CHiME-3 datasets, with center frequencies of mel filterbank, and center frequency learnt using CVAE in an unsupervised manner (discussed in Section 3.7), Right: zoomed plot for filter indices $15-35$ on top and for indices $50-75$ at the bottom.

(shown in Fig. 4.2). This corresponds to 1-second of length approximately. In the experiments, it is also observed that, after the normalization layer, the number of frames t can be pruned to the center 21 frames for the acoustic model training without loss in performance. This has significant computational benefits and the pruning is performed to keep only the 21 frames around the center frame.

Figure 4.4 shows the center frequency (μ_i values sorted in ascending order) of the acoustic filters obtained using multi-condition Aurora-4 (red curve) and CHiME-3 (green curve) datasets and this is compared with the center frequency of the conventional mel filterbank (blue curve) [18]. The center frequency learnt using CVAE in an unsupervised manner (discussed in Section 3.7) is also plotted (black curve) for comparison. As can be observed, the filterbank (learnt in supervised fashion) has a smoother allocation of center frequencies than the unsupervised one. Also, the supervised center frequencies allocates more filters in lower frequencies compared to the mel filterbank. The CHiME-3 data contains reverberation artifacts which resulted in lower center frequencies compared to the noisy Aurora-4 data center frequency values.

The soft relevance weighted time-frequency representation $\boldsymbol{z}$ obtained from the discussed approach is shown in Figure 4.5(d) for an utterance with airport noise from Aurora-4 dataset (the waveform is plotted in Figure 4.5(a)). The corresponding mel spectrogram (without relevance weighting) is plotted in Figure 4.5(b). The time-frequency representation obtained through the learned filterbank (without relevance weighting) is plotted in Figure 4.5(c). The acoustic filterbank representation with unsupervised learnt FB (acoustic FB layer output of CVAE in Figure 3.22) is also plotted in Figure 4.5(e). It can be observed that, in the supervised learning (Figure 4.5(c) and (d)), the formant frequencies are shifted upwards because of the increased number of filters in the lower frequency region. Among the supervised and unsupervised learnt spectrograms (Figure 4.5(c) and (e), respectively), the supervised one has enhanced formants visibility compared to the unsupervised one. Also, the relevance weighting modifies the representations to preserve only the important details of the spectrogram (Figure 4.5(d)).

4.2.2 *Modulation Filterbank Learning with Relevance Weighting*

The representation $\boldsymbol{z}$ from acoustic filterbank layer is fed to the second convolutional layer which is interpreted as modulation filtering layer (shown in Figure 4.2, expanded in Figure 4.3(b)). Specifically, the modulation filter is characterized using the rate (temporal variations measured in Hz) and scale (spectral variations measured in cyc. per mel) dimensions.

The first layer (acoustic FB) generates time-frequency representations (2-D representations) as the output ($\boldsymbol{z}$ in Fig. 1). These 2-D representations are indexed by time-frame on x-direction and sub-band index in the y-direction. The sub-band indices are ordered in increasing value of center frequency which

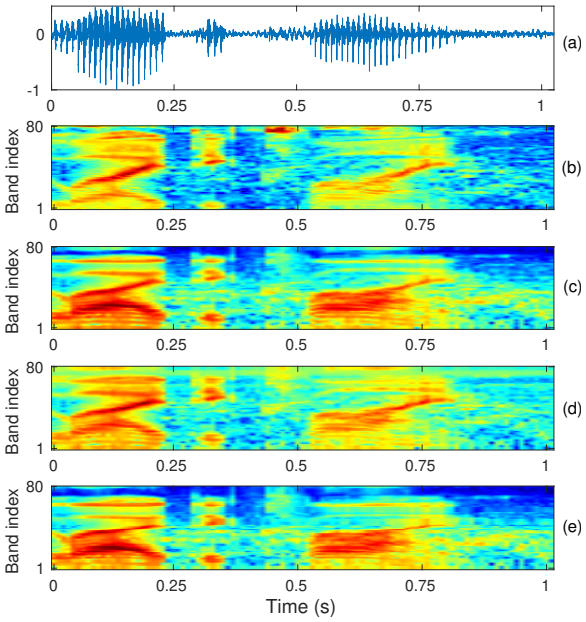


Fig. 4.5: (a) Speech signal from Aurora-4 dataset with airport noise, (b) mel spectrogram representation (c) acoustic filterbank representation ($\boldsymbol{x}$ in Figure 4.2) (d) acoustic filterbank representation with soft relevance weighting ($\boldsymbol{z}$ in Figure 4.2) (e) acoustic filterbank representation with unsupervised learnt FB (acoustic FB layer output of CVAE in Figure 3.22).

is required for modulation filtering in the second layer. The next stage of 2-D modulation filtering (rate-scale) is applied on 2-D time-frequency representation. The column of the time-frequency representation from the first layer constitutes a warped sampling of the spectrum from $0-8$ kHz with center frequencies shown in Figure. 2. Hence, applying a filter along the y-direction of this 2-D time frequency representation constitutes scale filtering and the application of the filter along the x-direction constitutes rate filtering. Note that the sampling in the y-direction is non-linear in frequency.

In this work, the parametric and non-parametric approaches to modulation filter learning are explored. In the parametric approach, the modulation filterbank (kernels of the modulation convolution layer) is designed as 2-D cosine-modulated Gaussian filters. Here, the ith 2-D filter $\boldsymbol{g}_i$ with rate frequency μ_{r_i} and scale frequency μ_{s_i} is designed as:

$$g_i(a,b) = \cos 2\pi(\mu_{r_i} a \pm \mu_{s_i} b) \times \exp\left[(-a^2) + (-b^2)\right] \tag{4.4}$$

with a sampled at 100 Hz (corresponding to 10 ms hop in the time-frequency representation), b sampled at 2 cycles per mel (with 8kHz acoustic range spanning 40 mels, and 80 filters with learnt center frequencies almost uniformly spaced in mel scale leading to sampling rate of 2 cycles/mel), and $i = 1, ..., K$ for K modulation filters. The $\pm$ sign in the cosine term is used to incorporate upward and downward moving patterns in the input time-frequency representation. The cosine frequencies μ_{r_i} and μ_{s_i} are interpreted as rate and scale center frequencies in 2-D frequency response of the modulation filter. The means are the learnable parameters of the 2-D kernels and these are learned jointly with rest of the network parameters. $K = 40$ modulation feature maps are learned using kernels of 5×5 filter tap size.

The learned center frequencies μ_r and μ_s (rate-scale) of the modulation filters are shown in Figure 4.6, from a model trained using Aurora-4 database. The rate frequency values span upto 50 Hz, while the scale frequency can span from 0 to 1 cycles/mel. It can be observed from the plot that the learned filters span the rate-scale space with more density till around 35 Hz rate frequency. The learned center frequencies span low scale regions with most of the filters in the $[-0.1, 0.75]$ cyc. per mel range. We begin with randomly spaced grid of rate-scale values (center frequency) as initialization and let the network update the center

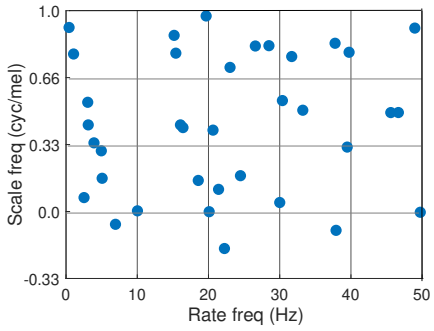


Fig. 4.6: Center frequency of the 2-D parametric modulation filters learned using the 2-stage approach for Aurora-4 dataset.

frequency values.

The modulation feature maps $\boldsymbol{p}$ are pooled, leading to feature maps of size $f' \times t'$. These are weighted using a second relevance weighting sub-network (referred to as the modulation filter relevance sub-network in Figure 4.2). The input to the modulation relevance weighting sub-network is a modulation feature map of size $f' \times t'$ and the model generates a scalar value for each of the K feature maps. Let $\boldsymbol{p_m}$ denote the K dimensional vector from the output of modulation relevance sub-network. Similar to the acoustic relevance network, a softmax function is applied to generate modulation relevance weights $\boldsymbol{w_m}$,

$$w_m^i = \frac{e^{p_m^i}}{\sum_j e^{p_m^j}}; \text{ where } i = 1, 2, ..., K. \tag{4.5}$$

The weights are multiplied with the representation $\boldsymbol{p}$ to obtain relevance weighted modulation feature maps $\boldsymbol{q}$. This weighting performs the adaptive selection of different modulation feature map representations (with different rate-scale characteristics). The resultant weighted representation $\boldsymbol{q}$ is fed to the batch normalization layer [43]. The value of the normalization factor for batch norm is also chosen to be 10^{-4} empirically. Following the acoustic filterbank layer and the modulation filtering layer (including the relevance sub-networks), the acoustic model consists of series of CNN and deep feed forward layers. The configuration details of different model parameters are given in Figure 4.2 and Figure 4.3. The entire model is trained using the cross entropy loss with Adam optimizer [51].

Figure 4.7 shows the obtained feature maps after the second stage of modulation filtering. The input to this layer is shown on left ($\boldsymbol{z}$ in Figure 4.2) and the obtained weighted $K = 40$ feature maps ($\boldsymbol{q}$ in Figure 4.2) are plotted in the right. As can be observed, the modulation filtering layer filters the input patch using rate-scale filters with varying center frequencies.

The described two stage processing is loosely modeled based on our understanding of the human auditory system, where the cochlea performs acoustic frequency analysis while early cortical processing performs modulation filtering [67]. The relevance weighting mechanism attempts to model the feature selection/weighting inherently present in the auditory system (based on the relative importance of the representation for the downstream task).

4.2.3 *Interpretability of the Speech Representations*

We analyze the representations and the relevance weights (outputs of acoustic relevance sub-network and modulation relevance sub-network) learned by the discussed model. The model architecture, shown in Figure 4.2, is trained using the Aurora-4 dataset. This model is tested using the TIMIT dataset (without any retraining). We use clean TIMIT as well as noisy TIMIT corrupted with noise as discussed in Section 2.4.6 [30]. The TIMIT dataset is hand labelled for phonemes, which allows the interpretation of the model representations based on the phoneme identity. The noisy analysis is performed on the noisy test set of TIMIT at 2 different SNR levels, 0 dB and 20 dB SNR.

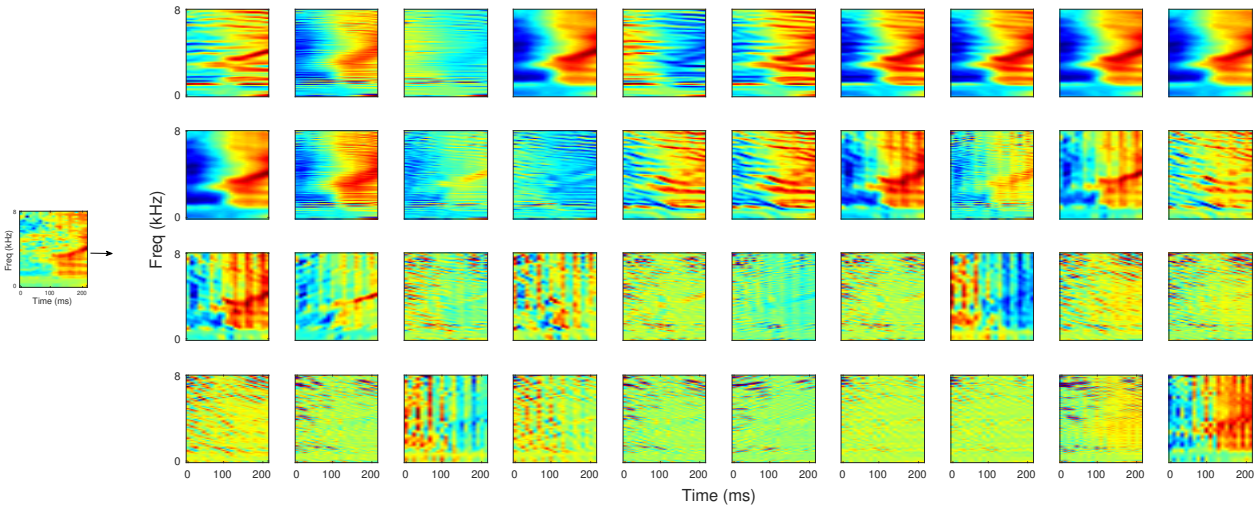


Fig. 4.7: Plot of modulation feature maps ($\boldsymbol{q}$ in Figure 4.2) for an input patch (shown on the left, corresponding to $\boldsymbol{z}$ in Figure 4.2) for an utterance from Aurora-4 dataset - airport noise (feature maps plotted in order of increasing rate frequency).

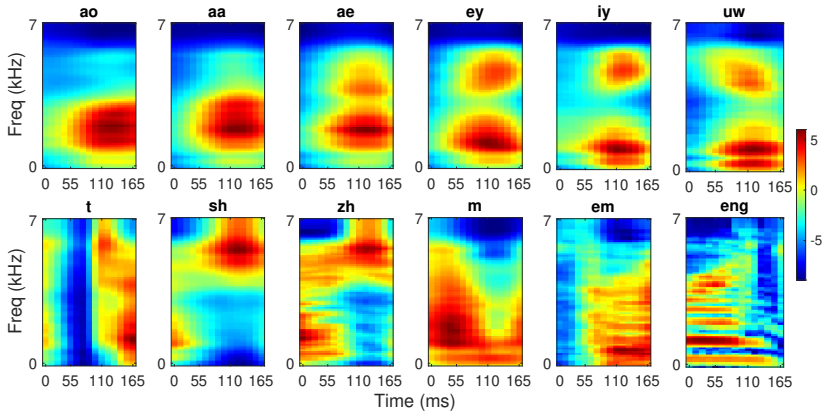


Fig. 4.8: Average time-frequency representation learned by the model for vowel phonemes (top row) and consonant phonemes (bottom row) from the clean TIMIT test set.

The phonemes are analyzed in two groups. A group of vowel phonemes {/ao/, /aa/, /ae/, /ey/, /iy/, /uw/} and a group of consonants from 3 categories: plosives {/t/}, fricatives {/sh/, /zh/}, and nasals {/m/, /em/, /eng/}. The vowel sounds are also organized from back to front with regard to the place of articulation [68].

4.2.3.1 *Mean Time-Frequency Representations*

We analyze the learned time-frequency representation of each phoneme from the first layer (using the representation $\boldsymbol{z}$ with 7 previous and 7 succeeding frames that cover 165 ms duration). For example, the time-frequency representation of all /aa/ vowel exemplars (denoted as $\boldsymbol{z}$ in Figure 4.2) are extracted from clean utterances and averaged to obtain one average time-frequency representation, as shown in Figure 4.8 (top row second column). For the computation of the average spectrogram for each phoneme, a contextual window of $+/-7$ frames are chosen around the center frame (from every exemplar of that phoneme occurrence in the files) irrespective of the phone duration. Thus, all center frames belonging to a phoneme, except the two boundary frames on either side, are used in computing the average time-frequency representation

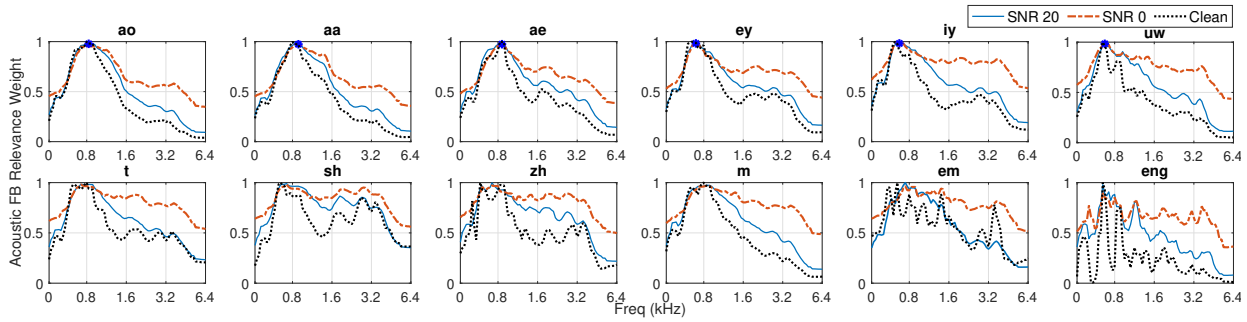


Fig. 4.9: The normalized acoustic FB relevance weight profile for each phoneme: vowels in top row and consonants in bottom row, computed using the relevance weights for clean (black dotted) and noisy TIMIT files with SNR 20 dB (blue-solid) and SNR 0 dB (red dot-dashed), respectively.

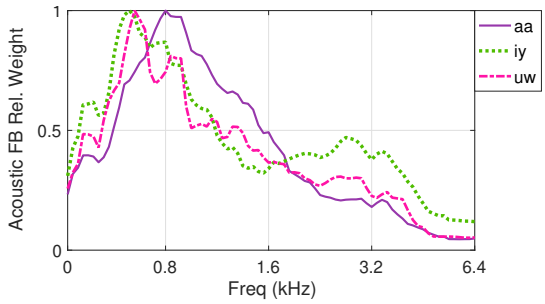


Fig. 4.10: Vowel Analysis - Acoustic filterbank (FB) relevance weights for 3 vowels on clean TIMIT data (black dotted curve for vowels in Figure 4.9). This figure highlights the contrast among vowels for clean condition.

of Figure 4.8.

The averaged time-frequency representation learned by the model reveals that mid/back vowels {/ao/, /aa/, /ae/} have relatively more concentrated activity at low to medium frequencies ($0.5-2$ KHz), whereas front vowels {/ey/, /iy/} have two distinct peaks spaced over a larger frequency range around 0.3 and 4 kHz respectively. This is consistent with the known distribution of the three formants F1, F2, and F3 in these vowels [68]. On the other hand, while the plosive /t/, being a stop-consonant, has a time varying profile, the fricatives {/sh/, /zh/} have dominant energy at high frequencies and the nasal sounds {/m/, /em/} have energy at low to mid frequencies. Thus, the early representations learned by the model attempt to capture distinct phonetic properties. The purpose of this analysis is to relate the relevance weight plots that are analyzed in following sections to the phonetic properties of the time-frequency representation.

4.2.3.2 *Acoustic FB Relevance Weights*

The acoustic filterbank relevance weights $\boldsymbol{w_a}$ are analyzed to understand the weighting incorporated through the relevance sub-network. The relevance weights are averaged across the utterances for each phoneme and SNR level separately. Figure 4.9 shows the relevance weights for clean data and noisy data averaged over all the 4 noise types.

As can be observed for clean and SNR of 20 dB (low noise), for the front vowels {/ey/, /iy/, /uw/}, the filter indices that have the higher relevance weights lie in $500-800$ Hz range. For the mid/back vowels {/ao/, /aa/, /ae/}, more relevance is seen in the filter indices having center frequency above 800 Hz. The filter indices with the highest relevance weights (peak marked with blue star in Figure 4.9) tend to shift towards lower frequency as we move from the back vowel /ae/ to the front vowel /uw/. This is also very similar to the filter representations seen in the auditory system of ferrets and other mammals [68]. Also, in front/closed vowels {/iy/, /uw/}, the relevance weights show a second peak at the higher frequency index

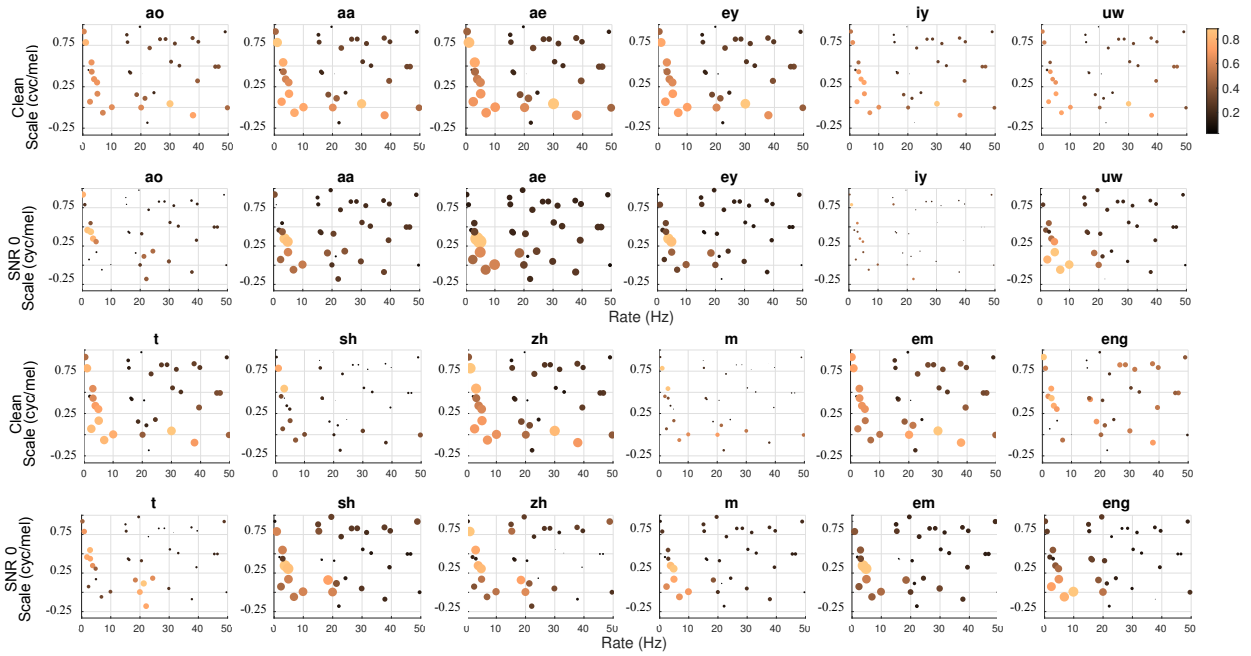


Fig. 4.11: The modulation relevance weights (after removing the mean weights) plotted for each phoneme: vowel phonemes in top two rows for clean and SNR 0 dB condition and consonants in the last two rows for clean and SNR 0 dB condition respectively. The size of the bubble is proportional to the magnitude of the relevance weight.

due to the presence of a higher formant frequency F3 (in the center frequency range of $3.5 - 4$ kHz). For the high noise condition (SNR 0 dB), the relevance weights still preserve some of the phoneme specific selectivity although the weights tend to become more uniform across the frequency range. A more closer look at the relevance weights is given in Figure 4.10, where 3 vowels are analyzed in clean conditions. As seen here, the peak activity moves from 800Hz to 500Hz as we move from back vowels to front vowels.

The acoustic FB relevance weights for consonants is shown in second row of Figure 4.9. The plosive /t/ shows similar relevance weighting as observed in front vowels. The fricatives {/sh/, /zh/} show lesser sub-band selectivity compared to other phonemes (as these phonemes have significant high frequency activity as seen in Figure 4.8). The nasal sounds {/em/, /eng/} also elicit a large range of filter indices that have high relevance weights. Similar to the vowels, the decrease in SNR (higher noise case) reduces the selectivity of the representations as the relevance weights become more uniform. This is again similar to the human auditory processing where noise segregation happens to a very small degree in the peripheral time-frequency representation as compared to higher auditory cortical areas [15].

4.2.3.3 *Modulation Filtering Relevance Weights*

Figure 4.11 shows the bubble plot for modulation filter relevance weights for different phonemes (averaged over all respective exemplars of each phoneme). This plot is obtained by placing the relevance weight at the location of the center frequency (rate-scale) of the corresponding modulation filter and the size of the bubble is proportional to the magnitude of the corresponding relevance weight. The vowels and consonants are arranged in the same order as in acoustic FB relevance weight analysis. In order to highlight the contrast, relevance weights for a given phoneme is plotted after subtracting the mean relevance weights over all the 50 TIMIT phonemes.

The top row shows the weights for vowels under clean condition. It can be observed that almost all the vowels have higher contrast in relevance weight values for the rate frequency in $0 - 10$ Hz rate and -0.1 to 0.75 cyc./mel scale. Additionally, most of the back and mid vowels elicit higher contrast in weights for

Table 4.1: Word error rate (%) in Aurora-4 database for multi-condition training with various feature extraction schemes.

Cond	MFB	PFB	RAS	MHE	A-R,M-R
A	4.2	4.0	5.3	3.8	**3.6**
B	7.2	7.5	8.5	7.4	**6.1**
C	7.2	7.3	9.0	7.3	**6.0**
D	15.9	17.9	17.7	17.3	**14.8**
Avg.	10.7	11.7	12.2	11.4	**9.6**

band-pass rate and low-pass scale frequencies. In the low SNR condition, for vowels (second row of Figure 4.11 with SNR= 0 dB), the relevance characteristics are much more intact compared to the acoustic FB layer relevance weights, with more contrast in lower rate regions. In case of consonants (third and fourth row of Figure 4.11), for the clean case, the plosive /t/ shows similar trend as mid vowels while the fricative /sh/ show a low-pass rate + high-pass scale contrast profile. Additionally, most of the consonants have high contrast towards low-pass rate + high-pass scale and band-pass rate + low-pass scale region. On the other hand, the noisy consonants (SNR 0 dB) show low-pass rate profile scattered over different scale values.

4.2.4 *Experiments - Automatic Speech Recognition*

The speech recognition system is trained using PyTorch toolkit [79] while the Kaldi toolkit [82] is used for decoding and language modeling. The models are discriminatively trained using the training data with cross entropy loss and Adam optimizer [51]. A hidden Markov model - Gaussian mixture model (HMM-GMM) system is used from Kaldi to generate the senone alignments for training the CNN-DNN based model in PyTorch. The notation [A,M] refers to the model of learning acoustic and modulation filterbanks without relevance, [A-R,M] refers to the model that involves acoustic FB with relevance along with modulation FB without relevance, [A, M-R] refers to learning acoustic FB (with no relevance weighting), followed by modulation filter learning with relevance weighting, and [A-R,M-R] refers to the model with learning acoustic and modulation FB and having relevance weighting in both layers. The modulation filters are learnt in all baseline features as the first 2-D CNN layer.

For each dataset, the ASR performance of the discussed approach of learning acoustic representation from raw waveform with acoustic FB (A) with relevance weighting (A-R) and modulation FB (M) with relevance weighting (M-R), is compared with traditional mel filterbank energy (MFB) features and other features discussed in Section 2.1. All the baseline features are processed with cepstral mean and variance normalization (CMVN) on a 1 sec. running window. The baseline MFB features are directly fed to the 2-D modulation filtering layer (without the acoustic FB layer). The other baseline features like PFB, RAS and MHE also generate spectrogram like time-frequency representations which are used similar to the MFB features. For all these features, no relevance weighting is performed. The modulation filtering layer (M) is part of the baseline system and is present with all the other features like PFB, RAS, MHE (without explicit mention in all cases). An additional experiment, where relevance weighting is applied on the baseline MFB features (denoted as MFB-R), is also performed.

For ASR experiments on the proposed model, a non-parametric approach in modulation filter learning is used. The non-parametric approach to modulation filtering involves learning 2-D kernels of the CNN layer that operates on the output of the acoustic FB layer. Empirically, the non-parametric modulation filters had a slight improvement over parametric modulation filters in ASR performance. The batch size of 32 is chosen for all the model training using a learning rate of 10^{-3}. The model training is performed for 10 epochs after which the learning is found to saturate on the validation data.

Table 4.2: Statistical significance of performance improvements for the discussed 2-stage method over the baseline MFB system using confidence interval and the probability of improvement (POI) on Aurora-4 dataset [10].

Test Cond.	Confidence Interval		POI (%)
	MFB	A-R,M-R	
A	[4.1, 5.3]	[3.8, 5.0]	95.1
B	[7.3, 9.7]	[7.2, 9.4]	86.8
C	[7.7, 10.7]	[6.5, 9.1]	99.0
D	[17.4, 23.0]	[16.4, 21.8]	95.3
Avg	–	–	94.0

Table 4.3: Word error rate (%) in CHiME-3 Challenge database for multi-condition training (real+simulated) with test data from simulated and real noisy environments.

Test Cond	MFB	PFB	RAS	MHE	A-R	A-R,M-R
Sim_dev	12.9	13.3	14.7	13.0	12.5	**12.0**
Real_dev	9.9	10.7	11.4	10.2	9.9	**9.6**
Avg.	11.4	12.0	13.0	11.6	11.2	**10.8**
Sim_eval	19.8	19.4	22.7	19.7	19.2	**18.5**
Real_eval	18.3	19.2	20.5	18.5	17.3	**16.6**
Avg.	19.1	19.3	21.6	19.1	18.2	**17.5**

4.2.4.1 *Aurora-4*

The WSJ Aurora-4 corpus discussed in Section 2.4.1 is used for conducting ASR experiments, shown in Table 4.1. The discussed representation learning (two-stage relevance weighting) provides considerable improvements in ASR performance over the baseline system with average relative improvements of 11% over the baseline MFB features. Furthermore, the improvements in ASR performance are consistently seen across all the noisy test conditions.

4.2.4.2 *Statistical Significance*

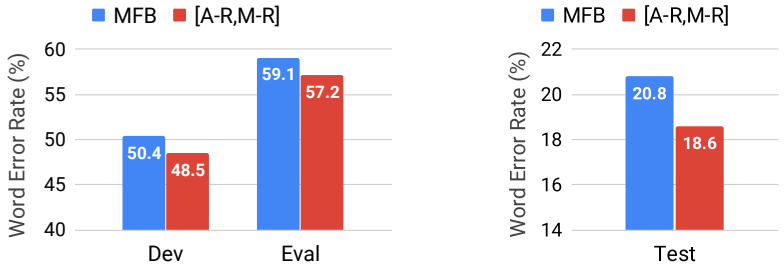
In order to compare how one system performs relative to the other in statistical sense, the bootstrap estimate for confidence interval is used [10]. This method computes a bootstrapping of word error rate (WER) values to extract the 95% confidence interval (CI), and also gives a probability of improvement (POI) for the system-in-test (system with 2-stage representation learning) over the reference system (baseline system with MFB features). Table 4.2 shows the analysis for various test conditions in the Aurora-4 multi-condition training. The POI of discussed system (A-R,M-R) system over the MFB is high for all the test conditions, with average POI being 94%.

4.2.4.3 *CHiME-3*

The CHiME-3 corpus discussed in Section 2.4.3 is used for training ASR with noisy+reverberant conditions. The results for the CHiME-3 dataset are reported in Table 4.3. The approach of acoustic FB learning with relevance weighting alone (A-R) improves over the baseline system (MFB) as well as the other noise robust front-ends considered here. The discussed approach of 2-stage relevance weighting over learned acoustic representations and modulation representations (A-R,M-R) provides significant improvements over baseline features. On the average, the 2-stage approach provides relative improvements of 10% over MFB features in the eval set. The detailed results on different noises in CHiME-3 are reported in Table 4.4. For most of the noise conditions in CHiME-3 in simulated and real environments, the 2-stage approach provides consistent improvements over the baseline features.

Table 4.4: WER (%) for each noise condition in CHiME-3 dataset with the baseline features and the discussed feature extraction [A-R,M-R].

Cond.	**Dev Data**				**Eval Data**			
	Sim		Real		Sim		Real	
	MFB	A-R,M-R	MFB	A-R,M-R	MFB	A-R,M-R	MFB	A-R,M-R
BUS	**10.9**	**10.9**	11.6	**11.3**	13.7	**13.0**	22.5	**21.4**
CAF	16.8	**4.7**	9.8	**9.5**	22.3	**19.6**	18.8	**16.1**
PED	10.4	**9.6**	8.0	**7.5**	20.8	**18.7**	17.7	**15.4**
STR	13.8	**13.0**	10.3	**10.0**	**22.5**	22.7	14.4	**13.3**



(a) (b)

Fig. 4.12: ASR performance in WER (%) for (a) VOiCES database, (b) Librispeech clean test dataset.

Table 4.5: Effect of relevance weighting on different stages of the 2-stage model on ASR with Aurora-4 dataset.

Features	**ASR (WER in %)**				
	A	B	C	D	Avg.
MFB	4.2	7.2	7.2	15.9	10.7
A (Acoustic FB)	4.1	6.8	7.3	16.2	10.7
MFB-R	4.0	7.3	7.1	16.1	10.8
A-R	3.6	6.4	8.1	15.1	10.0
A,M-R	**3.4**	**6.1**	7.9	16.9	10.4
MFB-R, G-R	3.7	6.3	6.4	15.1	10.3
A-R,M-R	3.6	**6.1**	**6.0**	**14.8**	**9.6**

4.2.4.4 *VOiCES*

The VOiCES corpus discussed in Section 2.4.4 with noise and reverberant data augmentation (1-fold) is used to train ASR. The ASR performance of VOiCES dataset with baseline MFB features and the approach of 2-step relevance weighting is reported in Figure 4.12 (a). These results suggest that proposed model is also scalable to relatively larger ASR tasks with large vocabulary where consistent improvements are obtained with such approach in addition to the interpretability of the representations learned.

We also test our proposed features with clean Librispeech test data using the trained VOiCES ASR model (since VOiCES train dataset consists of 80 hours of clean Librispeech data, discussed in Section 2.4.4). The results in Figure 4.12 (b) shows that the improvements are consistent with clean test conditions on large corpus ASR.

Table 4.6: WER (%) for cross-domain representation learning and ASR training experiments.

[A-R,M-R] Learned on	**ASR Trained and Tested on**		
	Aurora-4	CHiME-3	VOiCES
Aurora-4	**9.6**	14.3	57.6
CHiME-3	9.7	**14.2**	59.0
VOiCES	9.9	14.4	**57.2**

4.2.5 *Discussion*

4.2.5.1 *Effect of relevance weighting on different stages*

We compare the ASR performance of the proposed approach of 2-stage relevance weighting (A-R,M-R) with the learned acoustic FB representation (A) without any relevance weighting, relevance weighting on fixed mel filterbank features (MFB-R), acoustic FB relevance weighting without modulation FB relevance weighting (A-R), modulation relevance weighting alone on learned time-frequency representation (A,M-R). We also report performance with handcrafted filterbanks at both the stages, denoted as [MFB-R, G-R]. The hand-crafted modulation filters (G) were chosen as 2-D cosine-modulated Gaussian filters with pre-determined rate-scale center frequencies. Based on prior studies, we use more filters in the band-pass rate and low pass scale regions.

The results are reported in Table 4.5 for Aurora-4 dataset. Since the modulation filtering layer (M) is part of the baseline system and is present with all the features, the notation 'M' alone is omitted. As can be observed, the acoustic FB features (A) perform similar to MFB baseline features on average. The MFB-R features, which denote the application of the relevance weighting over mel filterbank features, does not provide improvements over baseline MFB features. The relevance weighted features (A-R) improve over the acoustic FB (A) features with average relative improvements of 6%. The features (A,M-R) having modulation relevance weighting alone improve over baseline in condition A and B, while there is degradation in C and D. The handcrafted features (MFB-R,G-R) also improve over the baseline. The proposed 2-stage relevance weighting improves in all test conditions over the baseline.

4.2.5.2 *Representation transfer across tasks*

In a subsequent analysis, a cross-domain ASR experiment is performed, i.e., the proposed representations (A-R,M-R) learned from one of the datasets (either Aurora-4, CHiME-3 or VOiCES challenge) is used to train/test ASR on the other dataset. All other layers in Figure 4.2 are learned using the in-domain ASR. The results of these cross-domain representation learning and ASR training experiments are reported in Table 4.6. The performance reported in this table are the average WER on each of the datasets. The results shown in Table 4.6 illustrate that the representation learning process is relatively robust to the domain of the training data, which suggest that the discussed representation learning approach can be generalized for other "matched" tasks (especially between Aurora-4 and VOiCES tasks).

4.2.5.3 *Comparison with other filterbank learning method*

To compare our 2-stage approach with the SincNet method [84], the cosine modulated Gaussian filterbank is replaced with the sinc filterbank as kernels in first convolutional layer. The baseline ASR system with sinc filterbank is trained jointly without any relevance weighting and rest of the architecture is kept same as shown in Fig. 4.2. The ASR system with sincFB and 2-stage relevance weighting is also trained. In addition, we also compare with acoustic FB learning in a non-parametric (ANP) manner (learning the 1-D CNN kernels directly from raw input without using a cosine modulated Gaussian function).

The ASR performance is reported in Table 4.7 for mel filterbank features (MFB), the cosine modulated Gaussian filterbank without relevance weighting (R) and the sinc filterbank from [84] (Sinc). For the experiments without relevance weighting, it can be observed that the parametric sinc filterbank performs similar to the cosine modulated Gaussian filterbank, and both perform similar to mel filterbank. The

Table 4.7: Comparison of MFB with different filterbank learning methods - without and with relevance weighting on Aurora-4 dataset.

Features without relevance weighting	ASR
MFB (mel filterbank)	10.7
A (our cosine modulated Gaussian filterbank)	10.7
Sinc (sinc filterbank from [84])	10.8
Features with relevance weighting	**ASR**
MFB-R,M-R	10.6
A-R,M-R	**9.6**
Sinc-R,M-R	10.0
ANP-R,M-R	11.6

Table 4.8: Unsupervised learning vs. Supervised learning of acoustic filterbank with [A-R,M-R] configuration on Aurora-4 dataset.

Type of acoustic FB learning	ASR
Random Initialization on uniform scale + No fine tuning	11.9
Random Initialization on sigmoidal frequency scale + No fine tuning	11.2
Unsupervised Initialization (& no fine tuning)	10.9
Unsupervised Initialization + supervised fine tuning	**9.6**
Random Initialization + supervised fine tuning	9.9

relevance weighting over sinc FB (Sinc-R,M-R) improves over the baseline with average relative improvement of around 6% over MFB, while the 2-stage [A-R,M-R] representation further improves over the sinc FB approach. The proposed parametric approach to acoustic FB learning also improves over the non-parametric (ANP) approach.

4.2.5.4 *Unsupervised vs. Supervised representation learning*

In the discussed 2-stage approach, the parametric kernels of the acoustic FB layer can be initialized in different ways; (i) initialization using unsupervised training of CVAE [4], (ii) unsupervised initialization + supervised fine tuning in ASR, and (iii) random initialization + supervised fine tuning. Table 4.8 shows the effects of acoustic FB initialization on ASR. All the features are trained with the discussed relevance weighting based model. For the random initialization, we sample the center frequencies uniformly at random or use a sigmoidal mapping function on the uniform random variable. It can be observed that unsupervised initialization alone of acoustic FB parameters (and no fine tuning) doesn't improve over baseline. The random initialization of acoustic FB followed by supervised fine-tuning for ASR gives considerable improvement in ASR performance. The random initialization of acoustic FB means on sigmoidal frequency scale performs better than the uniform scale with no fine tuning. The approach of unsupervised initialization with supervised fine-tuning (approach followed in this chapter) gives the best ASR performance among all choices considered here.

4.2.6 *Choice of Hyper-parameters*

We experiment with various choices of hyper-parameters in acoustic FB learning. The effect of context length of the input patch (t) is analyzed through ASR performance with the proposed [A-R, M-R] approach. The Aurora-4 dataset is used with different context length of $t = 21$, 51, 81, 101 and 131. From the results reported in Table 4.9, it can observed that the performance improves with increasing the context length, with the best performance for $t = 101$ and $t = 131$.

Table 4.9: Effect of context length of the input patch (value of t) on Aurora-4 ASR performance with the [A-R,M-R] approach.

Context length (value of t)	ASR (WER in %)
21	10.2
51	10.1
81	10.0
101	**9.6**
131	9.6

Table 4.10: Effect of different filter length (k) in acoustic FB layer on ASR performance with Aurora-4 dataset.

Filter length (value of k)	ASR (WER in %)
64 (4ms)	10.6
128 (8ms)	**9.6**
256 (16ms)	9.7

Table 4.11: Effect of different non-linearity on configurations of the proposed model for the ASR task on Aurora-4 dataset.

Features	**ASR (WER in %)**
A [Baseline]	10.7
Acoustic Relevance	
A-R [Softmax]	10.0
A-R [Sigmoid]	**9.9**
Acoustic + Mod. Relevance	
A-R,M-R [Softmax]	9.6
A-R,M-R [Sigmoid]	**9.5**

We also observe the effect of acoustic filter length k. While k = 8ms yields a frequency resolution of 125Hz, a 4ms long kernel has a frequency resolution of 250Hz and a 16ms long kernel has a frequency resolution of 62.5Hz. The baseline features like mel filter bank use windowing in the frequency domain with mel-spaced filters. The filters in the lower frequency range in the mel-scale have higher resolution (around 100 Hz). From Table 4.10, it can be observed that while the shorter filter length degrades the ASR performance, the increased filter length beyond 8ms does not improve the ASR performance.

4.2.7 *Effect of Non-linearity in Relevance Sub-networks*

Here, we change the type of non-linearity applied at the output of the relevance sub-networks (sub-networks at both the stages) and see its impact on ASR. While we have explored and analyzed the relevance weights with softmax non-linearity, we can use sigmoid instead of softmax. Table 4.11 shows the effect of non-linearity on the ASR performance. As can be observed, the relevance weight on acoustic FB alone (A-R) improves slightly with sigmoid over the use of softmax. With 2-stage relevance weighting, there is slight improvement again with sigmoid over softmax non-linearity.

4.2.7.1 *Brief Summary*

- Posing the representation learning problem in an interpretable filterbank learning framework using relevance weighting mechanism from raw waveform.
- Using a two-step process for learning representations: first step to obtain time-frequency representation from raw waveform; second step to perform modulation filtering; and relevance weighting

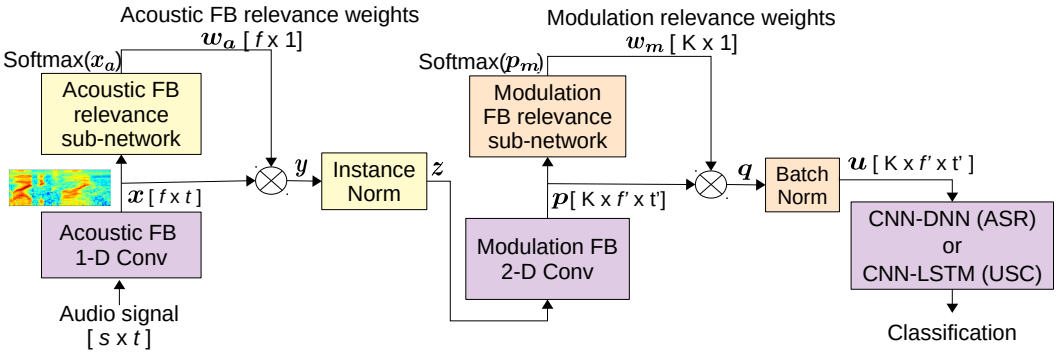


Fig. 4.13: Block diagram of the representation learning from raw waveform using relevance weighting approach for ASR or USC.

Table 4.12: Classifier accuracy (%) in UrbanSound8K database.

Rep.	Accuracy (%)										
	AI	CA	CH	DO	DR	EN	GU	JA	SI	ST	Avg
MFB	32	61	50	65	45	44	54	44	66	61	52
A	32	**67**	60	68	63	55	**82**	48	71	65	58
A-R,M-R	**40**	62	**60**	**68**	**66**	**56**	79	**59**	**74**	**67**	**62**
M3[17]	-	-	-	-	-	-	-	-	-	-	58

sub-networks in both steps.

- Illustrating the benefits of the 2-stage approach in ASR experiments with noise and reverberation.
- Analyzing the relevance weights that capture the underlying phonetic characteristics.

4.3 Representation Learning and Analysis For Audio Sounds

The spectro-temporal characteristics of the different types of audio signals is different, and hence, the time-frequency representation learning can capture the distinctive properties. Therefore, the 2-stage approach of representation learning discussed in previous section can be explored for any type of audio signal, in general, for a given task. While the previous section explored the 2-stage learning approach from raw speech signals for the task of speech recognition, we extend it for the task of urban sound classification (USC) in this section.

A block diagram (generic version of Figure 4.3) for audio representation learning is shown in Figure 4.13. The input to the network is an audio signal (can be speech or an urban sound), and the network is trained for the classification of senones (for ASR task) or classification of urban sounds (for USC task). The 2-stage model architecture and the layer operations are similar to those discussed in detail in Section 4.2 and Figure 4.3. Following the acoustic FB layer and the modulation filtering layer (with the two relevance weighting sub-networks), the model consists of CNN-LSTM layers for the USC task (the LSTM backend provided an improved performance for the USC task but not for the ASR task).

4.3.1 *Experiments*

The Urban Sound Classification (USC) system is trained using UrbanSound8K dataset [92] discussed in Section 2.4.5 using PyTorch [79]. We denote the 10 sound classes as: air conditioner (AI), car horn (CA), children playing (CH), dog bark (DO), drilling (DR), engine idling (EN), gun shot (GU), jackhammer (JA), siren (SI), and street music (ST). The best trained model is chosen through cross-validation loss. In our work, we use 9 folds for training purpose, and the results are reported with average of 10-fold cross

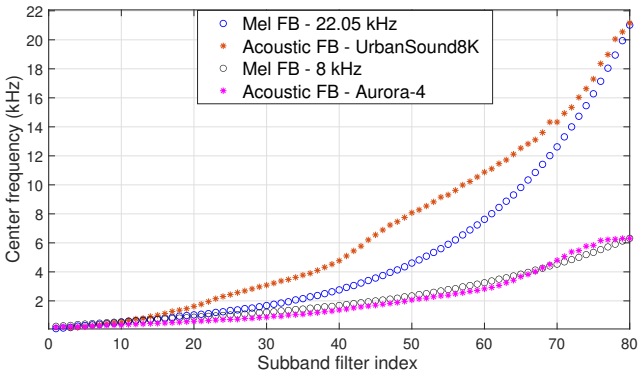


Fig. 4.14: Comparison of center frequency of acoustic FB learned using the discussed 2-stage approach with those of mel FB.

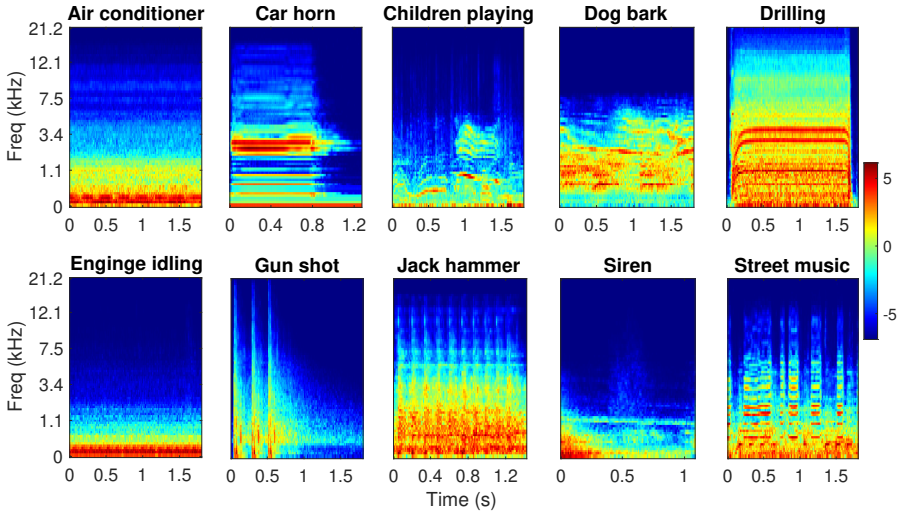


Fig. 4.15: Time-frequency representation learned by the model ($\boldsymbol{x}$ in Fig. 4.13) plotted for a file from each urban sound class.

validation.

Table 4.12 reports the average USC performance for various front-ends for each class type for baseline system and the proposed 2-stage approach of representation learning. As seen in the results, the learnt acoustic FB features (A) alone (without any relevance weighting) performs considerably better than the fixed MFB features, with an absolute improvement of 6%. The 2-stage (A-R,M-R) approach further gives improvement in almost all the classes with absolute improvement of 10% over baseline MFB system. In addition, the results are better than the big CNN model (M3) with high number of kernels (384) on raw audio waveform reported in [17].

4.3.2 *Interpretability of the Audio Signal Representations*

The discussed 2-stage model is used for the analysis of audio signals characteristics. The analysis is performed on the audio files in the cross-validation set. Fig. 4.14 shows the center frequency (μ_i values sorted in ascending order) of the acoustic filters obtained for the audio dataset. A comparison with the mel filter center frequencies is provided for reference. The learnt center frequencies for the ASR task are also plotted for analysis. The speech signals in Aurora-4 dataset are sampled at 16kHz while the audio signals in UrbanSound8K dataset are sampled at 44.1kHz. It can be observed that, while the learnt center frequencies in speech data match closely with the mel spectrogram, the center frequency of learned filters in audio data deviate from the mel counterparts. In particular, the learned representations have more emphasis on higher

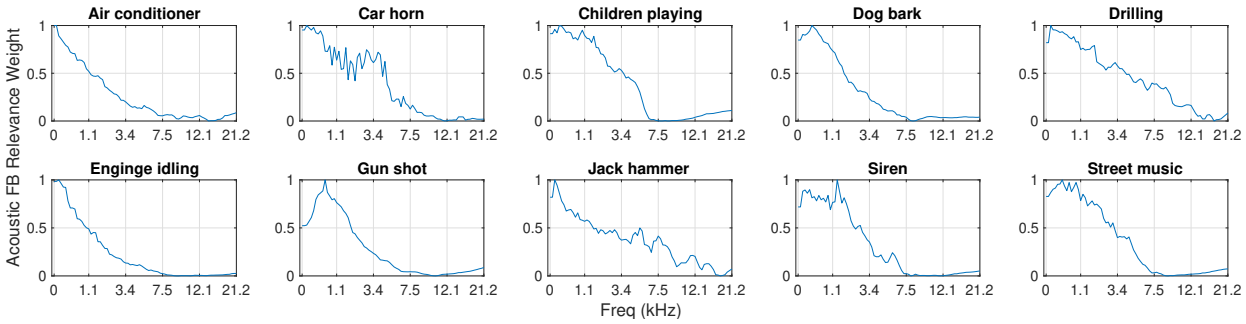


Fig. 4.16: The normalized acoustic FB relevance weight profile ($\boldsymbol{w}_a$ in Fig. 4.2) averaged over audio sounds from UrbanSound8K dataset.

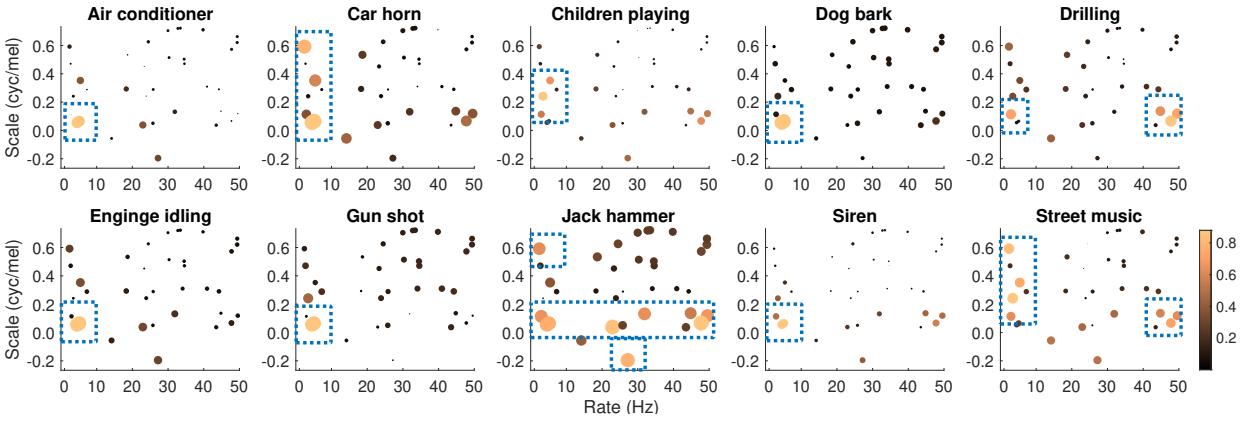


Fig. 4.17: The modulation relevance weights (after removing the mean weights) plotted for urban sound type.

frequencies compared to the mel spectrogram.

4.3.2.1 *Time-Frequency Representations*

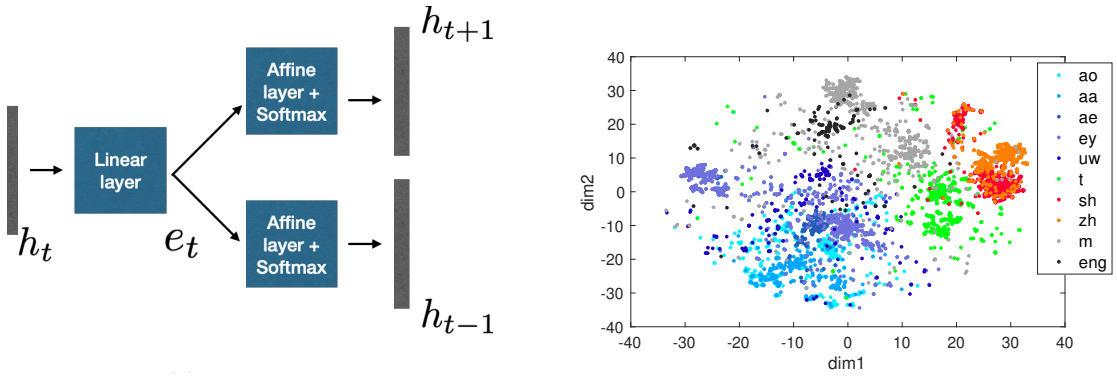
For the audio sounds in the UrbanSound8K dataset, we analyze the learned time-frequency representation of each class in Fig. 4.15. As seen here, air conditioner, engine idling and siren sounds have mostly low frequency content, the car horn, drilling, street music have multiple peaks at different frequencies. The drilling and jack hammer sounds have larger frequency range spanning till 15kHz.

4.3.2.2 *Acoustic FB Relevance Weights*

The acoustic FB relevance weights for the audio signals are analyzed in Fig. 4.16. It can be observed that most of the sounds exhibit higher relevance weight for low to mid acoustic frequencies, with drilling and jackhammer sounds having some weight till the end of the spectral range. The air conditioner and engine idling are low frequency sounds with peak primarily in between $10 - 200$Hz. The car horn, siren and street music sounds exhibit multiple frequencies and hence, have higher value for relevance weights till around 4kHz. The gun-shot weight profile exhibits an exclusive peak at around 550Hz.

4.3.2.3 *Modulation Filtering Relevance Weights*

Fig. 4.17 shows the bubble plot for modulation filter relevance weights for different audio sounds (averaged over all respective files of each sound type). This plot is obtained by placing the relevance weight at the location of the center frequency (rate-scale) of the corresponding modulation filter and the size of the bubble is proportional to the magnitude of the corresponding relevance weight.



(a) (b)

Fig. 4.18: (a) Block schematic of senone embedding network used in the model, (b) t-SNE plot of the senone embeddings for TIMIT dataset.

For audio signals, it can be observed that sounds like air conditioner, dog-bark and engine idling show low-rate and low-scale characteristics. The car horn sounds show higher relevance weight in low rate regions. The children playing, drilling, jackhammer and street music show higher relevance weights in high rate regions. The jackhammer and street music show relevance weight values spread over multiple rates (low, band, high pass) with varied scale values, which can be attributed to the presence of spectral energy at intermittent time frames (as seen in Fig. 4.15).

4.3.2.4 *Brief Summary*

- Extending the 2-stage representation learning approach to urban sound classification task.
- Illustrating the benefits of the learnt representations on the sound classification task through improved performance.
- Analyzing the representations where the relevance weights reflect distinct audio characteristics learnt by the model.

4.4 Representation Learning With Feedback

While the proposed relevance weighting based model relies on the current input frame for generating weights, there have been works where embeddings of the previous output in the form of feedback have improved the weighting based approach. In this Section, we extend the model discussed in Section 4.2 to include the target embeddings into the learning framework.

The block schematic of the senone embedding network is shown in Figure 4.18(a). The entire acoustic model using the embeddings in the model is shown in Figure 4.19.

4.4.1 *Step-0: Embedding Network Pre-training*

Before the ASR training, an embedding network is trained to generate embeddings as representations. The embedding network (Figure 4.18(a)) is similar to the skip-gram network of word2vec models [70]. In our approach, the one-hot encoded senone (context dependent triphone hidden Markov model (HMM) states modeled in ASR [62]) target vector at frame t, denoted as $\boldsymbol{h}_t$, is fed to a network whose first layer outputs the embedding denoted as $\boldsymbol{e}_t$. The first layer is a linear layer with N output nodes (embedding dimension). This embedding then predicts the one-hot target vectors for the preceding and succeeding time frames $\boldsymbol{h}_{t-1}$ and $\boldsymbol{h}_{t+1}$ through affine layer and softmax operation. This model is trained using the ASR labels (senones) for each task before the acoustic model training for ASR.

Once the word2vec style model is trained, only the embedding extraction part (first layer outputs $\boldsymbol{e}_t$) is

used to generate embeddings for training data and is used in the ASR model training. We use embeddings of $N = 200$ dimensions. During the ASR testing, the embeddings are derived by feeding the softmax outputs from the acoustic model (similar to teacher forcing network [111]). The t-SNE visualization of the embeddings is shown in Fig. 4.18(b) for phonemes from TIMIT test set (details of TIMIT data are given in Sec. 2.4.6). As seen here, the embeddings, while being trained on one-hot senones, provide segregation of different phoneme types.

4.4.2 *Step-1: Acoustic Filterbank Representation with Relevance Weighting*

The initial step of acoustic filterbank learning is similar to the one discussed in Section 4.2.1. Each frame of size $S \times 1$ raw audio samples are processed with a 1-D convolution using F kernels modeled as cosine-modulated Gaussian function discussed in Eq. 4.1. The mean parameter μ_i is updated in a supervised manner for each dataset. The convolution followed by the operations discussed earlier generates $\boldsymbol{x}$ as F dimensional features for each of the T contextual frames, as shown in Figure 4.19. The $\boldsymbol{x}$ is interpreted as the "learned" time-frequency representation (learned spectrogram).

4.4.2.1 *Relevance weighting*

The relevance weighting paradigm for acoustic FB layer is implemented using a relevance sub-network fed with the $F \times T$ time-frequency representation $\boldsymbol{x}$ and embeddings $\boldsymbol{e}$ of the previous time step. Let $\boldsymbol{x}_t(f)$ denote the vector containing sub-band trajectory of band f for all T frames centered at t (shown in Figure 4.2(b)). Then, $\boldsymbol{x}_t(f)$ is concatenated with embeddings of the previous time step $\boldsymbol{e}_{t-1}$ with $tanh()$ non-linearity. This is fed to a two layer deep neural network (DNN) with a sigmoid non-linearity at the output. It generates a scalar relevance weight $\boldsymbol{w}_a(t, f)$ as the relevance weight corresponding to the input representation at time t for sub-band f. This operation is repeated for all the F sub-bands which gives a F dimensional weight vector $\boldsymbol{w}_a(t)$ for the input $\boldsymbol{x}_t$.

The F dimensional weights $\boldsymbol{w}_a(t)$ multiply each column of the "learned" spectrogram representation $\boldsymbol{x}_t$ to obtain the relevance weighted filterbank representation $\boldsymbol{y}_t$. Here also, the first layer outputs ($\boldsymbol{y}$) are processed using instance norm as discussed in Eq. 4.3 ([86, 106]). The value of c is 10^{-4}. The value of $T = 101$, number of sub-bands $F = 80$, acoustic filter length $L = 129$, and window length $S = 400$ is used as discussed earlier.

The soft relevance weighted time-frequency representation $\boldsymbol{z}$ obtained from our approach is shown in Figure 4.20(iii) for an utterance with airport noise from Aurora-4 dataset (the waveform is plotted in Figure 4.20(i)). The corresponding mel spectrogram (without relevance weighting) is plotted in Figure 4.20(ii). It can be observed that, in the learned spectrogram representation (Figure 4.20(iii)), the formant frequencies are shifted upwards because of the increased number of filters in the lower frequency region.

4.4.3 *Step-2: Modulation Filtered Representation with Relevance Weighting*

This step is same as the one discussed in Section 4.2.2 with a second convolutional layer as modulation filtering layer (shown in Figure 4.19) with non-parametric kernels. The modulation filtering layer generates K parallel streams, corresponding to K modulation filters. These are weighted using a second relevance weighting sub-network (expanded in Figure 4.19(c)).

The modulation relevance sub-network is fed with feature map $\boldsymbol{p}_k$; where $k = 1, 2, ..., K$, and embeddings e of the previous time step. The embedding $\boldsymbol{e}$ is linear transformed and concatenated with the input feature map. This is fed to a two-layer DNN with softmax non-linearity at the output. It generates a scalar relevance weight $w_m(k)$ corresponding to the input representation at time t (t as center frame) for sub-band k. The weights $\boldsymbol{w}_m$ are multiplied with the representation $\boldsymbol{p}$ to obtain weighted representation $\boldsymbol{q}$. This weighting acts as a selection of different modulation filtered representations. The resultant weighted representation $\boldsymbol{q}$ is fed to the batch normalization layer [43] followed by series of CNN and DNN layers with sigmoid nonlinearity.

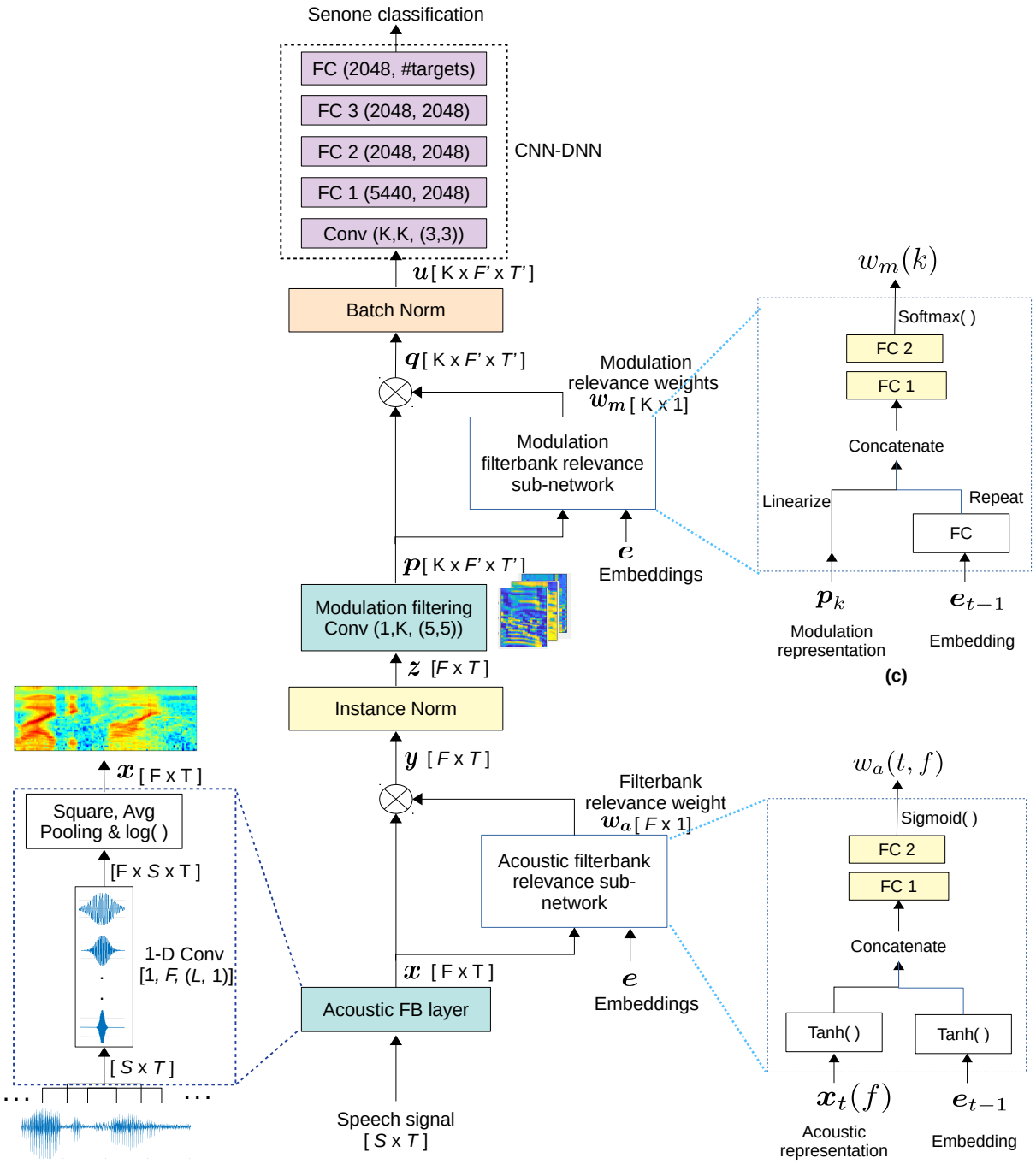


Fig. 4.19: (a) Block diagram of the representation learning from raw waveform using relevance weighting approach, (b) expanded acoustic FB relevance sub-network. Here, $\boldsymbol{x}_t(f)$ denotes the sub-band trajectory of band f for all frames centered at t, (c) expanded modulation filterbank relevance sub-network.

4.4.4 *Experiments and Results*

The speech recognition system is trained using PyTorch ([79]) while the Kaldi toolkit ([82]) is used for decoding and language modeling. The ASR results are reported with a tri-gram language model or using a recurrent neural network language model (RNN-LM).

For each dataset, we compare the ASR performance of the 2-stage representation learning with feedback approach (i.e., learning acoustic representation from raw waveform (A) with relevance weighting (A-R) and

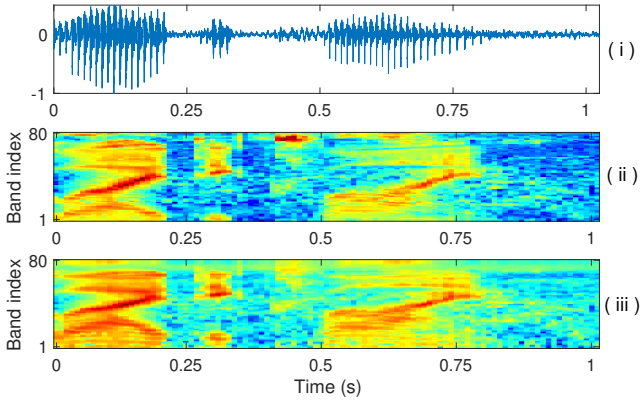


Fig. 4.20: (i) Speech signal from Aurora-4 dataset with airport noise, (ii) mel spectrogram representation (iii) acoustic FB representation with soft relevance weighting ($\boldsymbol{z}$ in Figure 4.19).

Table 4.13: Word error rate (%) for different configurations of the discussed model for the ASR task on Aurora-4 dataset using trigram LM.

Features	**ASR (WER in %)**				
	A	B	C	D	Avg.
A [Baseline Raw Waveform] (Section 4.2)	4.1	6.8	7.3	16.2	10.7
Acoustic Relevance					
A-R [Softmax, no embedding] (Section 4.2)	3.6	6.4	8.1	15.1	10.0
A-R [Sigmoid, no embedding]	3.4	6.4	6.7	15.5	9.9
A-R [Sigmoid, with senone embedding]	3.4	6.2	6.7	14.5	**9.6**
Acoustic + Mod. Relevance					
A-R,M-R [Softmax, no embedding] (Section 4.2)	3.6	6.1	**6.0**	14.8	9.6
A-R,M-R [Sigmoid, no embedding]	3.4	6.0	6.5	14.5	9.5
A-R,M-R [Sigmoid, with senone embedding in both relevance sub-networks]	**3.0**	**5.8**	6.2	**14.4**	**9.1**

feedback, followed by modulation filtering (M) with relevance weighting (M-R) denoted as (A-R,M-R)), with traditional log mel filterbank energy (MFB) features (80 dimension), other baseline features discussed in Section 2.1, results with excitation based (EB) features reported in [22], and SincNet method proposed in [84]. The neural network architecture shown in Figure 4.19 (except for the acoustic filterbank learning layer, the acoustic FB relevance sub-network and modulation filter relevance sub-network) is used for all the baseline features.

4.4.4.1 *Aurora-4 ASR*

The WSJ Aurora-4 corpus discussed in Section 2.4.1 is used for conducting ASR experiments. The ASR performance on the Aurora-4 dataset is shown in Table 4.13 for various configurations of the discussed approach and Table 4.14 for different baseline features. In order to observe the impact of different components of the model, we tease apart the components and measure the ASR performance (Table 4.13). We explore the variants of the discussed model such as use of softmax nonlinearity instead of sigmoid in both relevance weighting sub-networks, sigmoid in both relevance weighting sub-networks, without and with senone embedding, and the 2-stage approach (both relevance weighting sub-networks). The results with acoustic filtering alone (A) (discussed in Section 4.2) is similar to the approach proposed by [91]. Among the variants with A-R alone, the A-R (sigmoid with senone embeddings) improves over the softmax nonlinearity (was reported in Table 4.1 in Section 4.2). With joint A-R,M-R case, again the sigmoid with senone embeddings provides the best result.

While comparing with different baseline features in Table 4.14, it can be observed that most of the

Table 4.14: Word error rate (%) in Aurora-4 database for various feature extraction schemes decoded using trigram LM (and RNN-LM in paranthesis).

Cond	MFB	PFB	RAS	MHE	EB [22]	Sinc	MFB-R	A-R,M-R
A	4.2 (3.4)	4.0 (3.4)	5.3 (4.2)	3.8 (3.1)	3.7	4.0 (3.3)	3.9 (3.5)	**3.0 (2.9)**
B	7.2 (6.4)	7.5 (6.6)	8.5 (7.4)	7.4 (6.5)	6.0	7.3 (6.3)	7.2 (6.4)	**5.8 (5.3)**
C	7.2 (6.1)	7.3 (6.2)	9.0 (7.6)	7.3 (6.4)	**5.0**	7.3 (5.9)	7.1 (6.0)	6.2 (5.9)
D	15.9 (14.6)	17.9 (16.3)	17.7 (16.5)	17.3 (15.8)	15.8	16.2 (14.9)	16.1 (14.8)	**14.4 (13.7)**
Avg.	10.7 (9.7)	11.7 (10.5)	12.2 (11.1)	11.4 (10.2)	9.9	10.8 (9.7)	10.8 (9.8)	**9.1 (8.7)**

Table 4.15: Word error rate (%) in CHiME-3 Challenge database for multi-condition training with trigram LM.

Test Cond	MFB	PFB	RAS	MHE	Sinc	A-R(Soft)	A-R(Sigm)	A-R,M-R(Soft)	A-R,M-R(Sigm)
Sim_dev	12.9	13.3	14.7	13.0	13.4	12.5	12.4	12.0	**11.9**
Real_dev	9.9	10.7	11.4	10.2	10.1	9.9	9.9	9.6	**9.5**
Avg.	11.4	12.0	13.0	11.6	11.7	11.2	11.2	10.8	**10.7**
Sim_eval	19.8	19.4	22.7	19.7	22.2	19.2	19.0	**18.5**	18.7
Real_eval	18.3	19.2	20.5	18.5	18.7	17.3	17.2	**16.6**	17.0
Avg.	19.1	19.3	21.6	19.1	20.4	18.2	18.1	**17.5**	17.8

noise robust front-ends do not improve over the baseline mel filterbank (MFB) performance. The ASR system with MFB-R features, which denotes the application of the acoustic FB relevance weighting over the fixed mel filterbank features, also doesn't yield improvements over the system with baseline MFB features. We hypothesize that the learning of the relevance weighting with learnable filters allows more freedom in learning the model compared to learning with fixed mel filters. The discussed (A-R,M-R) representation learning (two-stage relevance weighting) with embeddings provide considerable improvements in ASR performance over the baseline system with average relative improvements of 15% over the baseline MFB features. Furthermore, the improvements in ASR performance are consistently seen across all the noisy test conditions and with a sophisticated RNN-LM. In addition, the performance achieved is also considerably better than the results reported for Aurora-4 database in works such as excitation based features (EB) reported by [22], [28], and [3].

For comparison with the SincNet method by [84], the proposed cosine modulated Gaussian filterbank is replaced with the sinc filterbank [1] as kernels in first convolutional layer (acoustic FB layer in Fig. 4.2). The ASR system with sinc FB (Sinc) is trained jointly without any relevance weighting keeping rest of the architecture same as shown in Fig. 4.2. From the results, it can be observed that the parametric sinc FB (without relevance weighting) performs similar to MFB.

4.4.4.2 *CHiME-3 ASR*

The CHiME-3 corpus discussed in Section 2.4.3 is used for building the noisy + reverberant ASR system. The results for the CHiME-3 dataset are reported in Table 4.15. The ASR system with SincNet performs similar to baseline MFB features. The initial approach of raw waveform filter learning with acoustic FB relevance weighting and feedback (A-R Sigm) improves over the baseline system as well as the other noise robust front-ends considered here. It also improves slightly over the corresponding softmax approach and without embedding (A-R Soft). The proposed approach of 2-stage relevance weighting with embeddings (A-R,M-R Sigm) over learned acoustic and modulation representations provides significant improvements over baseline features (average relative improvements of 10% over MFB features in the eval set). It also shows comparison with softmax nonlinearity and without embeddings in the relevance weighting sub-network (A-R,M-R Soft) discussed in Section 4.2. It can be observed that, while in dev data, the discussed approach

[1] https://github.com/mravanelli/SincNet/

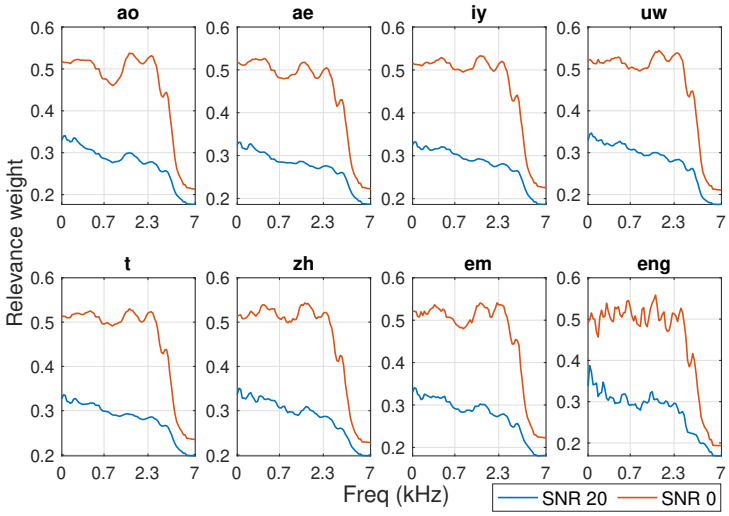


Fig. 4.21: The acoustic FB relevance weight profile ($\boldsymbol{w}_a$ in Figure 4.2) for each phoneme computed using the relevance weights for noisy TIMIT files with SNR 20 and SNR 0 dB, respectively.

with embedding performs slightly better than the previous approach, in eval data, the previous approach with softmax nonlinearity and without embeddings perform slightly better.

4.4.5 *Interpretability of the Representations*

The acoustic filterbank relevance weights $\boldsymbol{w_a}$ are analyzed using the TIMIT dataset (model trained using the Aurora-4 dataset, without any retraining) corrupted with noise as discussed in Section 2.4.6 [30]. The analysis is primarily to understand the weighting incorporated through the relevance sub-network. The relevance weights are averaged across the utterances for each phoneme and SNR level separately. Figure 4.21 shows the relevance weights averaged over all the 4 noise types. As can be observed for SNR of 20 dB (low noise) case, the relevance weight profile varies across phonemes with decreasing weights towards higher acoustic frequencies. The profile tend to become fluctuating for consonants. For the low SNR case (SNR = 0 dB), the relevance weights have more contrast between high and low frequency regions.

4.4.5.1 *Brief Summary*

- Adding the embeddings of the previous target in the acoustic FB relevance weighting framework.
- Illustrating performance gains in ASR with the use of senone embeddings and sigmoid nonlinearity in the 2-stage representation learning approach.
- Analyzing the relevance weights for different phonemes with the proposed approach.

4.5 Chapter Summary

In this chapter, a framework for relevance weighting based representation learning is explored from raw waveform. The key contributions of this chapter can be summarized as follows:

- Posing the representation learning problem in an interpretable filterbank learning framework using relevance weighting mechanism from raw waveform.
- Using a two-step process for learning representations: first step to obtain time-frequency (spectrogram) representation from raw waveform; second step to perform modulation filtering on first step output. Both steps have separate relevance weighting sub-networks, which are differentiable and can be easily integrated in the backpropagation learning.

Table 4.16: Summary of the proposed supervised 2-stage representation learning approach [A-R,M-R] in terms of ASR and USC performance.

Features	Task	WER/Acc.
Baseline	ASR	10.8
A-R,M-R		9.6
A-R,M-R - With feedback		**9.2**
Baseline	USC	52
A-R,M-R		**62**

- Implementing the acoustic filterbank in the first convolutional layer of the proposed model using a parametric cosine-modulated Gaussian filter bank whose parameters are learned. The relevance sub-network provides relevance weights to acoustic FB.
- Performing modulation filtering over the first layer output through the second convolution layer. Another relevance sub-network provides relevance weights to the modulation feature maps.
- Illustrating robustness in speech recognition and audio sound classification experiments on a variety of datasets containing noise and reverberation.
- Analysis of the relevance weights reveal that the weights capture underlying phonetic content for speech.
- Extending the representation learning framework to audio sounds for the task of urban sound classification (USC).
- Observing performance gains in the USC task with the learnt representations, and analysis of representation reveals distinctive audio characteristics.
- Incorporating the senone embeddings of the previous target to learn acoustic FB relevance weights with sigmoid nonlinearity.
- Illustrating ASR performance gains with the embedding based learning, and analysing the representations with the approach.

The ASR performance with supervised representation learning is summarized in Table 4.16. As can be observed from the summary, the proposed 2-stage relevance weighting approach [A-R,M-R] provides significant improvement over the ASR baseline. The incorporation of feedback further improves the performance. The extension of approach to urban sound classification gives significant improvement over baseline (by absolute 10% increase in accuracy).

Chapter 5

Summary and Future Extensions

5.1 Chapter Outline

In this chapter, we summarize the important contributions from this thesis. We highlight the benefits of the learnt representations for the task of speech recognition and urban sound classification. We also discuss the limitations of the proposed approaches and the scope of applicability. The chapter ends with a brief outline of various extensions of this thesis.

Section 5.2 highlights the main contributions from the thesis. The relation of the proposed work in the thesis with the prior works is discussed in Section 5.3. Section 5.4 discusses the limitations and scope of the thesis in speech and audio representation learning. Finally, we discuss various extensions of the thesis work in Section 5.5. We conclude the chapter in Section 5.6.

5.2 Summary of the Thesis Contributions

The thesis has focused on developing neural methods for representation learning of speech and audio signals, with the goal of improving downstream applications that rely on these representations.

For representation learning, we pursued two broad directions - supervised and unsupervised. In the case of speech/audio signals, we identified and explored two stages of representation learning. The first stage is the learning of a time-frequency representation (equivalent of spectrogram) from the raw audio waveform. The second stage is the learning of modulation representations (filtering the time-frequency representations along the temporal domain, called rate filtering and spectral domain, called scale filtering).

The novel contributions from this thesis can be summarized as -

(1) **Learning of irredundant modulation filters in unsupervised framework (Chap. 3, Sec. 3.2–3.6)**

- **Residual approach (Chap. 3, Sec. 3.2–3.4)**: To the best of our knowledge, the unsupervised learning of irredundant modulation filters is explored for the first time. We propose the learning of modulation filters using convolutional RBM. A residual approach is developed to learn non-overlapping and irredundant set of modulation filters followed by a data-driven filter selection criteria that enables us to select filters. The features are extracted by filtering spectrograms using the selected filters, which are then used for the task of ASR where the results show significant robustness to noise and reverberations. The filter learning is extended and proposed in 2-D fashion with rank constraints using CRBM. We extend the residual approach to two other models - autoencoder and GAN. The three models are analyzed and compared.

- **Modified cost function approach (Chap. 3, Sec. 3.5)**: We propose another approach of learning irredundant set of modulation filters through a modified cost function in variational framework. The variational autoencoder (VAE) trained with the modified cost function generates irredundant set of modulation filters in a single training (the residual approach learns one filter in a training and requires multiple model training). The obtained features show significant robustness in ASR.

Table 5.1: Summary of all unsupervised approaches of learning modulation filters and their comparison in terms of noisy speech recognition performance on Aurora-4 dataset.

Model	Cost function	Method of learning modulation filters	Type of filtering	ASR (WER %)	
				Clean	Multi
Baseline	-	-	-	24.7	12.1
CRBM 1-D	Boltzmann likelihood	Separate, Residual externally	Rate-scale	**19.4**	**10.8**
CRBM 2-D				19.9	11.1
CRBM 1-D	Boltzmann likelihood	Separate, Residual externally	Rate	23.0	**11.0**
CAE 1-D	Mean Square error			20.7	12.0
cGAN 1-D	Adversarial loss			**20.3**	11.3
CRBM 2-D	Boltzmann likelihood	Joint, Residual externally	Rate-scale	**19.9**	11.1
CVAE-ModC	All 4 terms (Eq. 3.17)	Joint, Modified cost function		22.1	11.2
CVAE-skip	Variational loss	Joint, Skip-connection		20.5	**11.0**

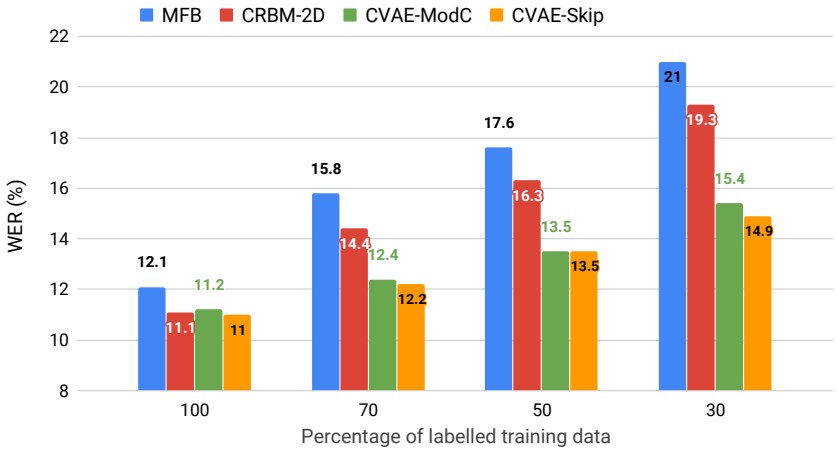


Fig. 5.1: Average ASR performance in terms of WER (%) in Aurora-4 database for multi condition training using lesser amount of labeled training data (70%, 50%, 30%).

- **Skip connection approach (Chap. 3, Sec. 3.6)**: Third approach of skip-connection based learning is proposed to learn modulation filters where the residual operation is incorporated inside the encoder of VAE itself through skip-connections. All the 3 approaches are analyzed and compared in terms of learnt filter parameters and ASR performance under different conditions.

The summary of the unsupervised representation learning approaches is given in Table 5.1 for modulation filtering and the ASR performance on Aurora-4 task is also highlighted. From the Table, it can be observed that the among 1-D and 2-D CRBM models for rate-scale filtering, CRBM 1-D model outperforms the baseline as well as being better than CRBM 2-D rank-1 model. Among the 3 different architectures - CRBM, CAE and cGAN for rate filtering, cGAN 1-D provides the best performance in clean training condition, while CRBM 1-D provides the best performance in multi-condition training. Among the three different joint filter learning techniques (residual computed externally, modified cost function and skip connection approach) in CRBM and CVAE, the CRBM 2-D rank-1 with residual computed externally approach outperforms other two in clean training condition, while in multi-condition, all the three approach are comparable in terms of ASR performance.

The modulation filtered features also show resilience to the reduced amounts of ASR training data. Figure 5.1 shows the summary for the semi-supervised case with 100, 70, 50 and 30% of the training

Table 5.2: Summary of unsupervised approaches of learning acoustic filterbank with modulation filters and their comparison in terms of noisy speech recognition performance on Aurora-4 dataset.

Model	Cost function	Method of learning Ac FB	Method of learning mod. FB	Type of mod. filtering	ASR	
					clean	multi
Baseline	-	-	-	-	24.7	12.1
CVAE-A	Variational (Eq. 3.20)	CVAE on raw	-	-	24.6	12.2
CVAE-A,M	Variational (Eq. 3.20)	CVAE on raw	Joint, skip-conn.	rate-scale	**20.6**	**11.4**

data. It can be observed that while all the modulation filtered representations provide improvements over baseline with reduced amounts of data, the CVAE-skip approach gives the best results among all in scarce labeled data conditions.

(2) **Representation learning from raw waveform in unsupervised framework (Chap. 3, Sec. 3.7)**: We propose the learning of time-frequency representation (spectrogram) from raw waveform in unsupervised variational framework. The parametric approach to learn time-domain acoustic filterbank as a set of cosine-modulated Gaussian filters is proposed. The learnt representations perform on par with the conventional mel based features.

The summary of the unsupervised representation learning approaches for acoustic filterbank and modulation filter learning is discussed in Table 5.2 on Aurora-4 database for noisy speech recognition. With the baseline as mel spectrogram features, the CVAE-A features from learning of acoustic filterbank using CVAE (without modulation filtering) is comparable to the baseline features. The two-stage approach [CVAE-A,M] of learning acoustic filterbank followed by modulation filters using skip-connection in CVAE gives considerable improvement in clean training as well as multi-condition training.

(3) **2-Stage Relevance Weighting Based Representation Learning in supervised framework (Chap. 4, Sec. 4.2)**: We propose interpretable representation learning from raw waveform as 2-stage approach in supervised fashion. The 2-stages perform the learning of acoustic filterbank and modulation filterbank in a sequential manner. A relevance weighting mechanism is developed to incorporate the weighting of the learnt features at each stage in forward propagation itself. The key difference between the proposed weighting and the self-attention method is that there is no linear combination of weighted representations involved. We analyze the learnt relevance weights for different speech sound types that reveal distinctive characteristics being captured. This has been presented for the first time to the best of our knowledge. The approach is then extended to audio sounds task.

- **Extension to Audio Sounds (Chap. 4, Sec. 4.3)**: The proposed 2-stage approach is extended for the task of audio sound classification with the motivation to realize and verify a common representation learning framework for any task with speech and audio data. The urban sound classification (USC) improves with the proposed approach and the analysis of relevance weights show that the proposed architecture is indeed able to capture distinct audio characteristics.

- **Representation Learning With Target Embedding Feedback (Chap. 4, Sec. 4.4)**: We propose the use of target embeddings in the learning of relevance weights as feedback to the relevance sub-networks. The senone (target) embeddings are learnt in word2vec style with one-hot senone targets as input and output. The ASR shows considerable improvements with the incorporation of feedback in relevance weighting.

The comparison of learnt center frequencies from the proposed 2-stage supervised framework is shown in Fig. 5.4 in contrast to mel center frequencies, where it can be observed that, while

Table 5.3: Summary of the proposed supervised 2-stage representation learning approach [A-R,M-R] in terms of ASR and USC performance.

Features	Task	WER/Acc.
Baseline	ASR	10.8
A		10.7
A-R,M-R		9.6
A-R,M-R - With feedback		**9.2**
Baseline	USC	52
A		58
A-R,M-R		**62**

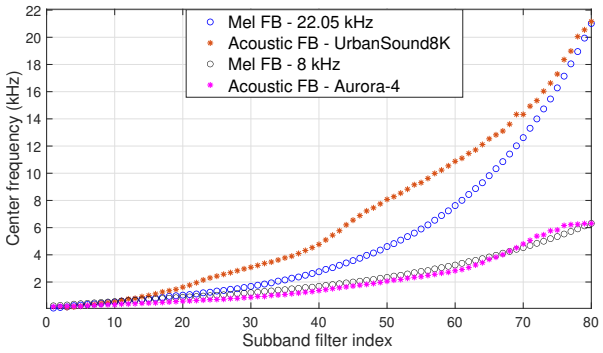


Table 5.4: Comparison of center frequency of acoustic FB learned using the discussed supervised 2-stage approach with those of mel FB.

the learnt means for ASR are close to mel center frequencies, the learnt means for USC have different profile compared to corresponding mel center frequencies. The effect is observed in the ASR and USC performance in Table 5.3. With acoustic FB features [A] alone (and no relevance weighting), while [A] is same as baseline performance in ASR, it improves over baseline in USC task. From the summary reported in Table 5.3, the proposed 2-stage relevance weighting approach [A-R,M-R] provides significant improvements over the ASR baseline. The incorporation of feedback further improves the performance. The extension of the approach to audio sound classification gives significant improvements over baseline (absolute 10% improvement in accuracy).

5.3 Relation with Prior Work

In this Section, we discuss the relation of the proposed work in the thesis with the prior works and the key differences are highlighted.

5.3.1 *Unsupervised Learning*

Sailor et. al., 2016 - Acoustic FB [88] - Here, the authors use convolutional restricted Boltzmann machine (CRBM) as a model for learning representations from raw speech signal. It learns all the filter taps leading to a large number of learnable parameters (for eg. for a filterbank with 80 filters, 128×80 parameters for 128 tap filter). While the non-parametric approach results in reduced interpretability of the learnt acoustic filterbank, in our unsupervised approach, learning filterbank through parametric approach leads to reduced learnable parameters and enhanced interpretability.

Schneider et. al., 2019 - Acoustic FB [97] - Here, the authors apply unsupervised pre-training to improve supervised speech recognition. Their wav2vec model is a multi-layer convolutional neural network that takes raw audio as input. The model is optimized to predict future samples from a given signal context with the objective of minimizing contrastive loss that requires distinguishing a true future audio sample from negatives. In our unsupervised approach, a variational autoencoder is used that operates on raw speech waveform. The first convolutional layer of encoder is designed as parametric filterbank layer and the network is trained with objective of minimizing variational loss (mean square error and latent loss).

Sailor and Patil, 2016 - Modulation FB [89] - In this work, the authors investigate unsupervised modulation filter learning using CRBM for ASR task. The temporal modulation representations are learned using log mel spectrogram as an input to CRBM. The work reports improvements in ASR performance over the baseline with the use of a system combination framework of mel filterbank features and modulation

features learned by CRBM. However, the number of modulation filters learnt are huge in number $(60-120)$ and the learnt representations alone do not improve over the baseline. In our unsupervised approach, we learn irredundant non-overlapping modulation filters (around 3 filters) and the learnt representations give significant improvements in ASR over the baseline mel features.

5.3.2 *Supervised Learning*

Tüske et. al., 2014 [105] - In this work, the authors use raw speech waveform as the input to DNN with first layer learning the filterbank. The filterbank (the first hidden layer weights) is initialized with auditory-inspired Gammatone filterbank and the DNN is trained for the task of ASR with cross-entropy loss. The analysis presented in the work suggests that even though the ASR did not improve with learning filterbank over the conventional mel features, the DNN is able to learn a set of band-pass filters in time domain purely from the raw waveform. In our approach, we use convolutional layer as the first layer of the acoustic model, with parametric kernels having only means as the learnable parameter. In addition, we do not initialize with any auditory-inspired filterbank parameters, and let the network learn them for the task with random initialization and unsupervised pre-training. This approach yields interpretable representations and the incorporation of sub-band relevance weighting gives significant improvements over the baseline mel features.

Ravanelli et. al., 2018 - SincNet FB [84] - The authors use Sinc filters as parametric acoustic filters in first convolutional layer, with only low and high cutoff frequencies of band-pass filters to be directly learned from data, and filters are learned in supervised manner for the task of speech recognition. The work shows that the filter learning is resilient to the presence of band-limited noise and gives improvements in ASR performance. In our supervised approach, we use cosine-modulated Gaussian filterbank in the first convolutional layer, with means as the learnable parameter. In addition, our relevance weighting mechanism allows the weighting of the learnt sub-band representations in the forward propagation itself. The proposed approach learns interpretable representations and provides significant improvements in ASR performance.

Shazeer et. al., 2017 [99], **Vaswani et. al, 2017** [108] - The motivation for relevance weighting comes from prior works on Mixture of Experts (MoE) models [45, 47]. For neural networks, the work proposed by authors in [99] explored multiple parallel neural networks followed by a gating network which performs a combination of the outputs from the networks. This MoE model showed significant promise for a language modeling task. The self attention module successfully inducted in transformer models [108] also incorporates information from multiple streams using a linear combination. In our work, the proposed relevance weighting based model does not add the weighted input again with the input but rather propagates the weighted feature stream directly to the subsequent layers. In addition, the weighting approach is applied on 2 stages (sub-bands weighting in acoustic FB layer and modulation weighting in modulation FB layer).

5.4 Limitations of the Proposed Work

While the proposed methods in the work are shown to provide improvements for the tasks and allow us to interpret, there are some limitations. Here we highlight some of the limitations of the work.

- **Empirical approach** - The proposed approaches are empirical. The models are proposed and analyzed based on performance metrics. The motivation behind the design of the proposed layer/model may have prior intuition, bio-inspired processing, motivation from signal processing, or works from the past. However, the final model with the network architecture and processing is justified based on performance.

- **Finding the optimal hyper-parameters** - As discussed in Section 3.5.4 and Section 4.2.5, a number of hyper-parameters are involved in the proposed models such as choice of non-linearity,

value of constant in normalization, number of sub-bands used in the analysis, filter length, and context length. Although these parameters offer flexibility in feature extraction, choosing the optimal parameter values for a given task can be cumbersome. Nevertheless, we note that reasonable parameter choices can be made from the results of the recognition experiments reported in respective sections.

- **Computational complexity** - The proposed approaches lead to a small increase in the amount of trainable parameters. In the first part of the work with unsupervised learning methods, there is requirement to train the generative models separately to obtain representations. Similarly, in supervised part with 2-stage approach, the initial 2 layers and relevance sub-networks lead to a small increase in percentage of trainable parameters (by approximately 0.2%).
  On similar note, the proposed approaches lead to an increase in computation time. The increase is variable for different approaches discussed in the thesis. For example, for the 2-stage relevance weighting approach, we performed an experiment with 100 test files from Aurora-4 corpus and measured the computation time for forward pass through the model for the baseline system as well as the proposed relevance weighted model. The computation time for the proposed model increased by 20% (171sec. of computational time for the proposed method versus 136sec. for the baseline mel-spec system).

- **No new architectures for unsupervised learning** - In this thesis, no new models are proposed for unsupervised learning of modulation filters or acoustic filterbank. The existing models such as RBM, AE, VAE are used and modified with required constraints, such as incorporating convolutional layer, change in number of kernels, design of kernels, use of skip-connections, etc. The main novelty in unsupervised learning were in developing judicious choices of cost functions that enabled learning of irredundant filters.

- **Bias towards time-frequency representation -** All the proposed methods in the thesis are biased in terms of using convolutional approaches and/or using parametric approaches that are initialised with mel/auditory filter-bank 'like' structure. Given both biological and system level evidences for their choice as feature representations, this thesis does not explore things beyond time-frequency representations, where some other architectures can be explored that can be more appropriate starting from raw waveform.

5.5 Future Extensions

The work reported in the thesis argues that data driven representations from the raw signal with minimal assumptions can yield task specific flexibility and interpretability while also providing superior performance. It is also worth noting that the thesis presents early research pursued in this emerging and exciting field. We list some of the possible extensions of the proposed techniques for speech and audio processing.

- **Incorporating the gross statistics of the signal**: The gross statistics of the signal maybe potentially useful in deriving the relevance weights for task-specific representations. The gross characteristics may capture the long-term information such as ambient environment characteristics, speaker characteristics (can be pitch, vocal tract characteristics), accent, channel information, etc. It may help the relevance sub-network learn and adapt the local data characteristics which is also hypothesized as the essence of data acquisition and learning in human beings. For example, gross statistics like i-vetors have been successfully applied for ASR tasks [93].

- **Novel model with new cost function for learning** - The work can be extended for developing new models with different cost functions in unsupervised learning frameworks. In addition, the

work done can be extended to other types of existing architectures such as RNN-LSTM, segmental RNN, RNN transducer, etc. in unsupervised framework.

- **Bio-inspired design of feature weighting mechanism** - The use of relevance weighting can be incorporated in different manner with motivations from bio-inspired studies of human auditory system and attention mechanism. While there have been previous findings that suggest that auditory attention causes not only enhancement in neural processing gain, but also sharpening in neural frequency tuning in human auditory cortex, the frequency sharpening can be incorporated by allowing center frequencies to be flexible based on input. There is another study that investigate that gain enhancement and frequency sharpening represent successive stages of a cooperative attentional modulation mechanism [19]. Hence, the relevance weighting mechanism that acts as feature selection/sharpening in our work can be modeled and used based on such findings in auditory modeling.

- **Role of Feedback in higher stages of processing** - Although the target embeddings are used only in the initial layers of the deep learning model, there have been studies on understanding of pitch perception which suggest a neuro-computational model for hierarchical generative process. It claims that higher (e.g., cortical) levels modulate the responses in lower (e.g., sub-cortical) levels via feedback connections [7]. Hence, the proposed feedback based approach in our work can be extended to higher level of representation layers.

5.6 Chapter Summary

In this chapter, we have summarized various contributions of this thesis (Sec. 5.2). The relation of the proposed work in the thesis with the prior works and the key differences have been discussed in Sec. 5.3. We have also described the limitations and scope of applicability of the proposed work in Sec. 5.4. Finally, we discuss some potential future extensions (Sec. 5.5).

5.7 Take Home Message from the Thesis

The proposed thesis conveys and directs towards the field to learn distinct and non-overlapping representations/features from the data in an interpretable manner than to use the engineered features. The learning frameworks explored are unsupervised and supervised, and the learnt representations being able to provide improved benefits over multiple tasks, with focus on automatic speech recognition (ASR) under various train and test conditions. The extension to tasks beyond ASR shows the ability of developing a generic framework to learn representations from raw audio/speech waveforms. Analysing the intermediate network layer representations and parameters in neural networks is an attempt and a step towards interpretable neural approaches and explainable deep learning.

Appendix A

Variational lower bound on marginal likelihood

The KL divergence between $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ and $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ for some arbitrary function q_ϕ is given as:

$$D_{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})] = \mathbb{E}_{\boldsymbol{z}|\boldsymbol{x}\sim q_\phi}[\log q_\phi(\boldsymbol{z}|\boldsymbol{x}) - \log p_\theta(\boldsymbol{z}|\boldsymbol{x})]$$

Applying Bayes rule to $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ gives,

$$\begin{aligned} D_{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})] = \mathbb{E}_{\boldsymbol{z}|\boldsymbol{x}\sim q_\phi}[&\log q_\phi(\boldsymbol{z}|\boldsymbol{x}) - \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) \\ &- \log p(\boldsymbol{z})] + \log p_\theta(\boldsymbol{x}) \end{aligned}$$

In the above equation, $p_\theta(\boldsymbol{x})$ comes out of expectation as it does not depend on $\boldsymbol{z}$. Rearranging, and contracting part of $\mathbb{E}_{\boldsymbol{z}|\boldsymbol{x}\sim q_\phi}$ yields,

$$\begin{aligned} \log p_\theta(\boldsymbol{x}) - D_{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})] = \mathbb{E}_{\boldsymbol{z}|\boldsymbol{x}\sim q_\phi}[&\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] \\ &-D_{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})] \end{aligned}$$

The right side term of the above equation can be written as,

$$\mathcal{L}(\theta, \phi; \boldsymbol{x}) = -D_{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})] + \mathbb{E}_{\boldsymbol{z}|\boldsymbol{x}\sim q_\phi}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})]$$

Hence,

$$\log p_\theta(\boldsymbol{x}) = D_{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})] + \mathcal{L}(\theta, \phi; \boldsymbol{x})$$

$$\log p_\theta(\boldsymbol{x}) \geq \mathcal{L}(\theta, \phi; \boldsymbol{x})$$

The last equation uses the non-negativity property of KL divergence. The term $\mathcal{L}(\theta, \phi; \boldsymbol{x})$ is called the variational lower bound of the marginal likelihood of datapoint $\boldsymbol{x}$. Thus, maximizing $\mathcal{L}$ inherently maximizes the data likelihood.

KL divergence between two Gaussian distributions

The KL divergence between the two Gaussian distributions: $q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}_\phi(\boldsymbol{x}), diag(\boldsymbol{\sigma}_\phi(\boldsymbol{x}))$ and $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ can be computed as,

$$D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) = \int \log\left(\frac{q_\phi(\boldsymbol{z}|\boldsymbol{x})}{p(\boldsymbol{z})}\right) q_\phi(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z}$$

Let z^i denote the i^{th} dimension of $\boldsymbol{z}$. Given the diagonal covariance assumption of q_ϕ,

$$D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}))$$

$$= \sum_i \int \Big[-\frac{1}{2}\log(2\pi) - \log(\sigma_\phi^i) - \frac{1}{2}\left(\frac{z^i - \mu_\phi^i}{\sigma_\phi^i}\right)^2$$

$$+ \frac{1}{2}\log(2\pi) + \frac{1}{2}(z^i)^2\Big] \frac{1}{\sqrt{2\pi}\sigma_\phi^i} exp\Big(\frac{z^i - \mu_\phi^i}{\sigma_\phi^i}\Big)^2 dz^i$$
$$= \frac{1}{2}\sum_i \Big(- \log(\sigma_\phi^i)^2 - 1 + (\mu_\phi^i)^2 + (\sigma_\phi^i)^2 \Big)$$

The above term is termed as 'latent loss' of VAE. Hence, Eq. 3.16 can be re-written as :

$$\mathcal{L}(\theta, \phi; \boldsymbol{x}) = \frac{1}{2}\sum_i \Big[\log(\sigma_\phi^i)^2 - (\sigma_\phi^i)^2 - (\mu_\phi^i)^2 + 1 \Big]$$
$$+ E_{\boldsymbol{z}|\boldsymbol{x}\sim q_\phi}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})]$$

As discussed in Sec. 3.5.1, the distribution $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ is also assumed to be Gaussian and the expectation (second term of the above equation) is the negative of mean square error (MSE) loss.

Bibliography

[1] Purvi Agrawal and Sriram Ganapathy. Unsupervised modulation filter learning for noise-robust speech recognition. *The Journal of the Acoustical Society of America*, 142(3):1686–1692, 2017.

[2] Purvi Agrawal and Sriram Ganapathy. Comparison of unsupervised modulation filter learning methods for ASR. *Proc. of Interspeech*, pages 2908–2912, 2018.

[3] Purvi Agrawal and Sriram Ganapathy. Modulation filter learning using deep variational networks for robust speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 13(2):244–253, 2019.

[4] Purvi Agrawal and Sriram Ganapathy. Unsupervised raw waveform representation learning for ASR. *Proc. of Interspeech 2019*, pages 3451–3455, 2019.

[5] Purvi Agrawal and Sriram Ganapathy. Deep variational filter learning models for speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5731–5735, 2019.

[6] Xavier Anguera, Chuck Wooters, and Javier Hernando. Acoustic beamforming for speaker diarization of meetings. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):2011–2022, 2007.

[7] Emili Balaguer-Ballester, Nicholas R Clark, Martin Coath, Katrin Krumbholz, and Susan L Denham. Understanding pitch perception as a hierarchical process with top-down modulation. *PLoS Comput Biol*, 5(3):e1000301, 2009.

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

[9] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[10] Maximilian Bisani and Hermann Ney. Bootstrap estimates for confidence intervals in asr performance evaluation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 409–412, 2004.

[11] Herve A Bourlard and Nelson Morgan. *Connectionist speech recognition: a hybrid approach*, volume 247. Springer Science & Business Media, 2012.

[12] Hugo Caselles-Dupré, Florian Lesaint, and Jimena Royo-Letelier. Word2vec applied to recommendation: Hyperparameters matter. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 352–356, 2018.

[13] Barry Chen, Qifeng Zhu, and Nelson Morgan. Long-term temporal features for conversational speech recognition. In *International Workshop on Machine Learning for Multimodal Interaction*, pages 232–242. Springer, 2004.

[14] Taishih Chi, Powen Ru, and Shihab A Shamma. Multiresolution spectrotemporal analysis of complex sounds. *The Journal of the Acoustical Society of America*, 118(2):887–906, 2005.

[15] Sandra Da Costa, Wietske van der Zwaag, Lee M Miller, Stephanie Clarke, and Melissa Saenz. Tuning in to sound: frequency-selective attentional filter in human primary auditory cortex. *Journal of Neuroscience*, 33(5):1858–1863, 2013.

[16] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural

networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1):30–42, 2011.

[17] Wei Dai, Chia Dai, Shuhui Qu, Juncheng Li, and Samarjit Das. Very deep convolutional neural networks for raw waveforms. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 421–425. IEEE, 2017.

[18] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, 1980.

[19] Jessica de Boer and Katrin Krumbholz. Auditory attention causes gain enhancement and frequency sharpening at successive stages of cortical processing—evidence from human electroencephalography. *Journal of cognitive neuroscience*, 30(6):785–798, 2018.

[20] Didier A Depireux, Jonathan Z Simon, David J Klein, and Shihab A Shamma. Spectro-temporal response field characterization with dynamic ripples in ferret primary auditory cortex. *Journal of neurophysiology*, 85(3):1220–1234, 2001.

[21] Xavier Domont, Martin Heckmann, Frank Joublin, and Christian Goerick. Hierarchical spectro-temporal features for robust speech recognition. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP).*, pages 4417–4420. IEEE, 2008.

[22] Thomas Drugman, Yannis Stylianou, Langzhou Chen, Xie Chen, and Mark JF Gales. Robust excitation-based features for automatic speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4664–4668, 2015.

[23] Rob Drullman, Joost M Festen, and Reinier Plomp. Effect of temporal envelope smearing on speech reception. *The Journal of the Acoustical Society of America*, 95(2):1053–1064, 1994.

[24] Rob Drullman, Joost M Festen, and Reinier Plomp. Effect of reducing slow temporal modulations on speech reception. *The Journal of the Acoustical Society of America*, 95(5):2670–2680, 1994.

[25] Taffeta M Elliott and Frédéric E Theunissen. The modulation transfer function for speech intelligibility. *PLoS computational biology*, 5(3):e1000302, 2009.

[26] ES ETSI. 202 050 v1. 1.1 stq; distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms. *ETSI ES*, 202(050):v1, 2002.

[27] Jeffrey A Fessler and Alfred O Hero. Space-alternating generalized expectation-maximization algorithm. *IEEE Transactions on signal processing*, 42(10):2664–2677, 1994.

[28] Josué Fredes, José Novoa, Simon King, Richard M Stern, and Nestor Becerra Yoma. Locally normalized filter banks applied to deep neural-network-based robust speech recognition. *IEEE Signal Processing Letters*, 24(4):377–381, 2017.

[29] Mark Gales and Steve Young. *The application of hidden Markov models in speech recognition.* Now Publishers Inc, 2008.

[30] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. DARPA TIMIT acoustic-phonetic continous speech corpus. *NASA STI/Recon technical report*, 93, 1993.

[31] 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.

[32] Donald D Greenwood. Auditory masking and the critical band. *The journal of the acoustical society of America*, 33(4):484–502, 1961.

[33] Hynek Hermansky and Nelson Morgan. RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing*, 2(4):578–589, 1994.

[34] Hynek Hermansky and Sangita Sharma. Temporal patterns (TRAPS) in asr of noisy speech. In *International Conference on Acoustics, Speech, and Signal Processing, Proceedings.*, volume 1, pages 289–292. IEEE, 1999.

[35] Gregory Hickok and David Poeppel. The cortical organization of speech processing. *Nature Reviews Neuroscience*, 8(5):393, 2007.

[36] Steven A Hillyard, Edward K Vogel, and Steven J Luck. Sensory gain control (amplification) as

a mechanism of selective attention: electrophysiological and neuroimaging evidence. *Philosophical Trans. of the Royal Society of London. Series B: Biological Sciences*, 353(1373):1257–1270, 1998.

[37] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[38] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[39] Yedid Hoshen, Ron J Weiss, and Kevin W Wilson. Speech acoustic modeling from raw multichannel waveforms. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4624–4628, 2015.

[40] Jui-Ting Huang, Jinyu Li, and Yifan Gong. An analysis of convolutional neural networks for speech recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4989–4993, 2015.

[41] Xuedong Huang and Li Deng. An overview of modern speech recognition. *Handbook of natural language processing*, 2:339–366, 2010.

[42] Jeih-Weih Hung and Lin-Shan Lee. Optimization of temporal filters for constructing robust features in speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(3):808–832, 2006.

[43] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proc. of International Conference on Machine Learning (ICML)*, pages 448–456, 2015.

[44] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[45] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[46] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[47] Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

[48] Jaakko Kauramäki, Iiro P Jääskeläinen, and Mikko Sams. Selective attention increases both gain and feature selectivity of the human auditory cortex. *PLoS One*, 2(9):e909, 2007.

[49] Chanwoo Kim and Richard M Stern. Power-normalized cepstral coefficients (PNCC) for robust speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4101–4104, 2012.

[50] Doh-Suk Kim, Jae-Hoon Jeong, Jae-Weon Kim, and Soo-Young Lee. Feature extraction based on zero-crossings with peak amplitudes for robust speech recognition in noisy environments. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 61–64. IEEE, 1996.

[51] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Proc. of International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1412.6980*, 2015.

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

[53] Keisuke Kinoshita, Marc Delcroix, Sharon Gannot, Emanuël AP Habets, Reinhold Haeb-Umbach, Walter Kellermann, Volker Leutnant, Roland Maas, Tomohiro Nakatani, Bhiksha Raj, et al. A summary of the REVERB challenge: state-of-the-art and remaining challenges in reverberant speech processing research. *EURASIP Journal on Advances in Signal Processing*, 2016(1):1–19, 2016.

[54] Keisuke Kinoshita et al. The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech. In *IEEE WASPAA*, pages 1–4, 2013.

[55] Michael Kleinschmidt. Localized spectro-temporal features for automatic speech recognition. In *Proc. of Eurospeech*, pages 2573–2576, 2003.

[56] György Kovács, László Tóth, and Dirk Van Compernolle. Selection and enhancement of gabor filters for automatic speech recognition. *International Journal of Speech Technology*, 18(1):1–16, 2015.

[57] Siddique Latif, Rajib Rana, Sara Khalifa, Raja Jurdak, Junaid Qadir, and Björn W Schuller. Deep representation learning in speech processing: Challenges, recent advances, and future trends. *arXiv preprint arXiv:2001.00378*, 2020.

[58] Rabiner Lawrence et al. *Fundamentals of speech recognition*. Pearson Education India, 2008.

[59] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[60] Jongpil Lee, Taejun Kim, Jiyoung Park, and Juhan Nam. Raw waveform-based audio classification using sample-level cnn architectures. *Proc. Neural Information Processing Systems (NeurIPS)*, 2017.

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

[62] Hui Lin, Li Deng, Dong Yu, Yi-fan Gong, Alex Acero, and Chin-Hui Lee. A study on multilingual acoustic modeling for large vocabulary asr. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4333–4336. IEEE, 2009.

[63] Andrej Ljolje. Speech recognition using fundamental frequency and voicing in acoustic modeling. In *Seventh International Conference on Spoken Language Processing*, 2002.

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

[65] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks*, pages 52–59. Springer, 2011.

[66] Nima Mesgarani and Shihab Shamma. Speech processing with a cortical representation of audio. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5872–5875. IEEE, 2011.

[67] Nima Mesgarani, Malcolm Slaney, and Shihab A Shamma. Discrimination of speech from nonspeech based on multiscale spectro-temporal modulations. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(3):920–930, 2006.

[68] Nima Mesgarani, Stephen V David, Jonathan B Fritz, and Shihab A Shamma. Phoneme representation and classification in primary auditory cortex. *The Journal of the Acoustical Society of America*, 123(2):899–909, 2008.

[69] Tomas Mikolov, Stefan Kombrink, Anoop Deoras, Lukar Burget, and Jan Cernocky. RNNLM-recurrent neural network language modeling toolkit. In *Proc. of the Automatic Speech Recognition and Understanding (ASRU) Workshop*, pages 196–201, 2011.

[70] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Proc. of International Conference on Learning Representations (ICLR), arXiv preprint arXiv:1301.3781*, 2013.

[71] Wiktor Młynarski and Josh H McDermott. Learning midlevel auditory codes from natural sound statistics. *Neural computation*, 30(3):631–669, 2018.

[72] Abdel-rahman Mohamed, George E Dahl, and Geoffrey Hinton. Acoustic modeling using deep belief networks. *IEEE transactions on audio, speech, and language processing*, 20(1):14–22, 2011.

[73] Hema A Murthy and Venkata Gadde. The modified group delay function and its application to phoneme recognition. In *Int. Conf. on Acoustics, Speech, and Signal Processing, 2003. Proceedings (ICASSP)*, volume 1, pages I–68. IEEE, 2003.

[74] Mahesh Kumar Nandwana, Julien Van Hout, Colleen Richey, Mitchell McLaren, Maria Alejandra Barrios, and Aaron Lawson. The VOiCES from a distance challenge 2019. *Proc. of Interspeech*, pages 2438–2442, 2019.

[75] Sridhar Krishna Nemala, Kailash Patil, and Mounya Elhilali. A multistream feature framework based on bandpass modulation filtering for robust speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 21(2):416–426, 2013.

[76] Mohammad Norouzi. *Convolutional restricted Boltzmann machines for feature learning*. PhD thesis, School of Computing Science-Simon Fraser University, 2009.

[77] Dimitri Palaz, Ronan Collobert, and Mathew Magimai Doss. Estimating phoneme class conditional

probabilities from raw speech signal using convolutional neural networks. *Proceedings of Interspeech*, pages 1766–1770, 2013.

[78] Santiago Pascual, Mirco Ravanelli, Joan Serrà, Antonio Bonafonte, and Yoshua Bengio. Learning problem-agnostic speech representations from multiple self-supervised tasks. *arXiv preprint arXiv:1904.03416*, 2019.

[79] Adam Paszke, Sam Gross, Soumith Chintala, and Gregory Chanan. PyTorch. *Computer software. Vers. 0.3*, 1, 2017.

[80] RD Patterson, Ian Nimmo-Smith, John Holdsworth, and Peter Rice. An efficient auditory filterbank based on the gammatone function. In *a meeting of the IOC Speech Group on Auditory Modelling at RSRE*, volume 2, 1987.

[81] David Pearce and J Picone. Aurora working group: Dsr front end lvcsr evaluation au/384/02. *Inst. for Signal & Inform. Process., Mississippi State Univ., Tech. Rep*, 2002.

[82] Daniel Povey et al. The KALDI speech recognition toolkit. In *IEEE ASRU*, number EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[83] N Vishnu Prasad and Srinivasan Umesh. Improved cepstral mean and variance normalization using bayesian framework. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 156–161. IEEE, 2013.

[84] Mirco Ravanelli and Yoshua Bengio. Interpretable convolutional filters with SincNet. *in Proc. of Neural Information Processing Systems (NIPS)*, 2018.

[85] Colleen Richey, Maria A Barrios, Zeb Armstrong, Chris Bartels, Horacio Franco, Martin Graciarena, Aaron Lawson, Mahesh Kumar Nandwana, Allen Stauffer, Julien van Hout, et al. Voices obscured in complex environmental settings (VOiCES) corpus. *Proc. of Interspeech*, pages 1566–1570, 2018.

[86] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533, 1986.

[87] Seyed Omid Sadjadi, Taufiq Hasan, and John HL Hansen. Mean Hilbert envelope coefficients (MHEC) for robust speaker recognition. In *Proc. of Interspeech*, 2012.

[88] Hardik B Sailor and Hemant A Patil. Filterbank learning using convolutional restricted boltzmann machine for speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5895–5899, 2016.

[89] Hardik B Sailor and Hemant A Patil. Unsupervised learning of temporal receptive fields using convolutional RBM for ASR task. In *European Signal Processing Conference (EUSIPCO)*, pages 873–877. IEEE, 2016.

[90] Tara N Sainath, Brian Kingsbury, Abdel Rahman Mohamed, and Bhuvana Ramabhadran. Learning filter banks within a deep neural network framework. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 297–302, 2013.

[91] Tara N Sainath, Ron J Weiss, Andrew Senior, Kevin W Wilson, and Oriol Vinyals. Learning the speech front-end with raw waveform CLDNNs. In *Proc. of Interspeech*, pages 1–5, 2015.

[92] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1041–1044, 2014.

[93] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny. Speaker adaptation of neural network acoustic models using i-vectors. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 55–59, 2013.

[94] Marc René Schädler and Birger Kollmeier. Separable spectro-temporal gabor filter bank features: Reducing the complexity of robust features for automatic speech recognition. *The Journal of the Acoustical Society of America*, 137(4):2047–2059, 2015.

[95] Ralf Schluter and Hermann Ney. Using phase spectrum information for improved speech recognition performance. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, volume 1, pages 133–136. IEEE, 2001.

[96] Ralf Schluter, Ilja Bezrukov, Hermann Wagner, and Hermann Ney. Gammatone features and feature

combination for large vocabulary speech recognition. In *International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–649. IEEE, 2007.

[97] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. *Proc. of Interspeech*, pages 3465–3469, 2019.

[98] Hiroshi Seki, Kazumasa Yamamoto, and Seiichi Nakagawa. A deep neural network integrated with filterbank learning for speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5480–5484, 2017.

[99] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *ICLR*, 2017. URL arXivpreprintarXiv:1701.06538.

[100] Nandini C Singh and Frédéric E Theunissen. Modulation spectra of natural sounds and ethological theories of auditory processing. *The Journal of the Acoustical Society of America*, 114(6):3394–3411, 2003.

[101] Stanley Smith Stevens and Hallowell Davis. Hearing: its psychology and physiology. 1938.

[102] Stanley Smith Stevens, John Volkmann, and Edwin B Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190, 1937.

[103] Etienne Thoret, Philippe Depalle, and Stephen McAdams. Perceptually salient regions of the modulation power spectrum for musical instrument identification. *Frontiers in psychology*, 8:587, 2017.

[104] Sangita Tibrewala and Hynek Hermansky. Multi-band and adaptation approaches to robust speech recognition. In *Fifth European Conference on Speech Communication and Technology*, 1997.

[105] Zoltán Tüske, Pavel Golik, Ralf Schlüter, and Hermann Ney. Acoustic modeling with deep neural networks using raw time signal for LVCSR. In *Proc. of Interspeech*, pages 890–894, 2014.

[106] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[107] Sarel Van Vuuren and Hynek Hermansky. Data-driven design of RASTA-like filters. In *Eurospeech*, volume 1, pages 1607–1610, 1997.

[108] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[109] Olli Viikki and Kari Laurila. Cepstral domain segmental feature vector normalization for noise robust speech recognition. *Speech Communication*, 25(1-3):133–147, 1998.

[110] Yu-Xiong Wang and Yu-Jin Zhang. Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering*, 25(6):1336–1353, 2013.

[111] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[112] Xiaowei Yang, Kuansan Wang, and Shihab A Shamma. Auditory representations of acoustic signals. *IEEE Transactions on Information Theory*, 38(2):824–839, 1992.

[113] B Yegnanarayana, P Satyanarayana Murthy, Carlos Avendaño, and Hynek Hermansky. Enhancement of reverberant speech using lp residual. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, volume 1, pages 405–408. IEEE, 1998.

[114] Liang-Chih Yu, Jin Wang, K Robert Lai, and Xuejie Zhang. Refining word embeddings for sentiment analysis. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 534–539, 2017.

[115] Neil Zeghidour, Nicolas Usunier, Iasonas Kokkinos, Thomas Schaiz, Gabriel Synnaeve, and Emmanuel Dupoux. Learning filterbanks from raw speech for phone recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5509–5513, 2018.

[116] Yu Zhang, Pengyuan Zhang, and Yonghong Yan. Attention-based LSTM with multi-task learning for distant speech recognition. *Proc. of Interspeech*, pages 3857–3861, 2017.

[117] Eberhard Zwicker and Ernst Terhardt. Analytical expressions for critical-band rate and critical

bandwidth as a function of frequency. *The Journal of the Acoustical Society of America*, 68(5): 1523–1525, 1980.

Vita

Purvi Agrawal is a PhD scholar in Learning and Extraction of Acoustic Patterns (LEAP) lab with Dr. Sriram Ganapathy, Dept. of Electrical Engineering, Indian Institute of Science (IISc), Bengaluru. Prior to joining IISc, she obtained her Master of Technology in Speech Communications from DA-IICT, Gandhinagar in 2015. She has also worked in Sony R & D Labs, Tokyo in 2017. She would be joining as an Applied Researcher-II at Microsoft India with speech research team in February 2021. Her research interests include interpretable deep learning, raw waveform modeling, low-resource data modeling, multimodal speech system building, unsupervised/self-supervised learning, biologically inspired deep learning, and source separation.

Representation learning is the branch of machine learning consisting of techniques that are capable of automatically discovering meaningful representations from raw data for efficient information extraction. In the speech processing field, representation learning has been a challenging task. Delving into this expanse of representation learning, the thesis makes the following contributions.

- Neural representation learning of the speech and audio data with focus on initial 2 stages
  - Stage-1: Acoustic representation from raw speech waveform
  - Stage-2: Modulation filtered representation from spectrogram features
- Unsupervised learning
  - Novelty in learning the irredundant modulation filters from the data using - residual approach, skip-connection and modified cost function
  - Directly compares unsupervised learning approaches in depth
- Supervised learning
  - Parametric approach to learn acoustic and modulation representations
  - Proposing to incorporate relevance weighting scheme to acoustic/ modulation filterbank
  - Analyze representations & network parameters at intermediate stages
  - Novelty in feedback based modelling with relevance weight
  - Extending the approach to audio signals - Generic learning framework for audio signals

Purvi Agrawal completed her Ph.D. from Learning and Extraction of Acoustic Patterns (LEAP) lab with Dr. Sriram Ganapathy, Dept. of Electrical Engineering, Indian Institute of Science (IISc), Bengaluru. Prior to joining IISc, she obtained her Master of Technology in Speech Communications from DA-IICT, Gandhinagar in 2015. She has also worked in Sony R & D Labs, Tokyo in 2017. Currently, she is an Applied Researcher-II at Microsoft with speech research team in India. Her research interests include interpretable deep learning, raw waveform modeling, low-resource data modeling, multi-modal speech system building, unsupervised/self-supervised learning, biologically inspired deep learning, and source separation.