

**AN INDIRECT SPEECH ENHANCEMENT FRAMEWORK THROUGH
INTERMEDIATE NOISY SPEECH TARGETS**

A Dissertation
Presented to
The Academic Faculty

By

Sicheng Wang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Engineering
Department of Electrical and Computer Engineering

Georgia Institute of Technology

Mar 2021

© Sicheng Wang 2021

AN INDIRECT SPEECH ENHANCEMENT FRAMEWORK THROUGH INTERMEDIATE NOISY SPEECH TARGETS

Thesis committee:

Prof. Chin-Hui Lee, Advisor
Electrical and Computer Engineering
Georgia Institute of Technology

Prof. Sabato Marco Siniscalchi, Co-
Advisor
Computer engineering
Università degli Studi di Enna “Kore”

Prof. David Anderson
Electrical and Computer Engineering
Georgia Institute of Technology

Prof. Alexander Lerch
School of Music
Georgia Institute of Technology

Prof. Elliot Moore II
Electrical and Computer Engineering
Georgia Institute of Technology

Date approved: Feb. 24, 2021

A thousand roads lead a man forever toward Rome.

Alain de Lille

For my parents

ACKNOWLEDGMENTS

I want to express my sincere gratitude to my advisor and mentor, Prof. Chin-Hui Lee, for offering me the opportunity to join the human language technology lab and leading me to research in the fast-evolving field of automatic speech processing. As my advisor, he lends his insights on many complex problems. As a mentor, he is patient and guides me through setbacks. I am deeply indebted to him for his advice and kindness.

I also want to thank my co-advisor, Prof. Sabato Marco Siniscalchi. I am immensely thankful for his guidance and encouragement during my study. As an experienced researcher, he helped me brainstorm many ideas, plan my experiments, and proof-read my writing. I am deeply affected by his diligence at work and attention to detail. I will always cherish the memories of many discussions and hang-outs we had together.

I am also grateful to my committee members, Prof. David Anderson, Prof. Elliot Moore II, and Prof. Alexander Lerch, for serving on my thesis reading committee. Prof. Anderson and Prof. Moore provided many helpful feedbacks after my proposal. Prof. Moore also kindly offered me to use his computing resources for my experiments.

I was fortunate to work with a team of talented lab mates at Georgia Tech. Dr. You-Chi Cheng shared his internship experience with me. Dr. I-Fan Chen helped me furnish the voice tag package. Dr. Zhen Huang provided many useful tips in my job searching. Dr. Kehuang Li coached me to maintain the server cluster. Dr. Wei Li, Qi Jun, Hu Hu, Chao-Han Huck Yang, and Yongliang He collaborated with me on several projects.

I had the opportunity to collaborate with some industrial researchers both during semesters and during the summers. Each project rewarded me with invaluable experience. I want to thank Dr. Pongtep Angkititrakul, Dr. Zhe Feng, Mr. Peter Eastty, Dr. Chao Weng, Dr. Joshua Atkins, Dr. Jason Wung, Dr. Ramin Pishehvar, Dr. Ming Lei, Dr. Zhijie Yan, Yongtao Jia, Linzhang Wang. My life at the lab is also enjoyable and memorable because of the visiting scholars, including Dr. Yong Xu, Dr. Ji Wu, Dr. Jing Zhang, Dr. Fengpei

Ge, Dr. Hu Chen, Dr. Haifeng Sun, Dr. Bo Wu, Dr. Yi Lin, Dr. Quandong Wang, Dr. Gang Chen, Leonard Loo Tiang Kuan, Meri Tan and friends at CSIP, Sam Li, Dr. Meng Zhong, Dr. Kyle Xu, Dr. Zhen Wang, Dr. Yuting Hu, Dr. Le Liang, Dr. Hao Ye, Ziyang He, and Helen Li. Finally, I want to thank Pat Dixon, Raquel Plaskett, Dr. Daniela Staiculescu, and Tasha Torrence for their great administrative support.

Lastly, I want to thank my parents for their support and encouragement during my study away from home. I'm grateful for my friends, Huaidong Yang, Ruxiu Liu, Congshan Wan, Li Wang, Yuan Gao, Yingdan Wu, Qiming Zhang, and Xiaoyao Liu for being part of my life in Atlanta.

TABLE OF CONTENTS

Acknowledgments	v
List of Tables	xi
List of Figures	xiii
List of Acronyms	xvi
Chapter 1: Introduction	1
1.1 Overview	1
1.2 Main contributions	4
Chapter 2: Background	5
2.1 Speech enhancement	5
2.1.1 Classical signal processing techniques	8
2.1.2 Deep learning methods	16
2.1.3 Progressive speech enhancement	20
2.2 Characterization of noise	22
2.3 Speech transformation	25
2.3.1 Mean-variance normalization	26
2.3.2 Exemplar-based methods	28

2.3.3	Multi-layer perceptrons	28
2.3.4	Generative models	29
2.4	A high-level description of the proposed progressive enhancement approach with intermediate noisy speech target	30
Chapter 3: Characterization of Additive Noises		37
3.1	Introduction	37
3.1.1	Noise in speech enhancement	37
3.1.2	Enhancement quality and improvement depending on noise types . .	37
3.2	On the criteria to select intermediate targets	39
3.2.1	Simple noise with high PESQ score	43
3.2.2	Noise shape	45
3.2.3	Bandwidth	47
3.2.4	Stationarity	47
3.3	Summary	49
Chapter 4: Indirect Speech Enhancement with Supervised Learning		50
4.1	Introduction	50
4.2	Matching feature statistics	51
4.2.1	Effects of noise in feature normalization in speech enhancement . .	51
4.2.2	Deviation of mean in normalizing speech in difficult noise types . .	54
4.2.3	Deviation of variance in normalizing speech in difficult noise types .	57
4.2.4	Mean-variance matching	59
4.2.5	Histogram equalization	61

4.2.6	Experiments and discussions	62
4.3	Speech conversion with DNN mapping	66
4.3.1	DNN training	67
4.3.2	Experiments and discussions	68
4.4	Interference of multiple noise sources	70
4.4.1	Framework of indirect enhancement with two noise sources	72
4.4.2	Experiments and discussions	73
4.5	Summary	77
Chapter 5: Indirect Speech Enhancement with Latent Space Learning		79
5.1	Introduction	79
5.2	Representational learning via auto-encoder	80
5.2.1	Latent space of speech features using PCA	82
5.2.2	Use of nonlinear auto-encoders to convert speech features	87
5.2.3	Experimental results	95
5.3	Dictionary-based indirect speech conversion and enhancement	102
5.3.1	Problem formulation	102
5.3.2	Experiments and discussions	105
5.4	Summary	109
Chapter 6: Conclusions		111
6.1	Summary of research	111
6.2	Contributions	111
6.2.1	Noise characterization	111

6.2.2	Indirect enhancement via supervised learning	112
6.2.3	Indirect enhancement via representational learning	113
6.3	Future work	114
6.3.1	Theoretical characterization of noise types	114
6.3.2	Noisy speech with multiple sources	115
6.3.3	Disentanglement of latent feature	115
6.3.4	Explorations of different deep architectures for speech transformation	115
Appendices		116
Appendix A: Derivation of mean deviation in the normalization of LPS feature .		117
Appendix B: Derivation of the variance deviation in the normalization of LPS feature		118
Appendix C: Definition of colored noise		121
Appendix D: Description of Nonspeech noise		125
Appendix E: Description of Noisex92 noise		126
References		131
Vita		144

LIST OF TABLES

3.1	PESQ on Noisex92 noise types	39
3.2	PESQ of conversion using simple noise as the intermediate targets	43
3.3	Comparison of MSE of direct and indirect methods on the 1st quantile of noisy speech	44
3.4	Effects of spectral shapes on the suitability as intermediate targets	45
3.5	MSE comparison between <i>volvo</i> and <i>ovlov</i> noise	46
3.6	Effects of bandwidth of conversion targets	47
3.7	Effects of stationarity of conversion targets	48
4.1	PESQ score with and without matching statistics	64
4.2	KLD between other noise and <i>volvo</i> in a hidden layer	66
4.3	Progressive indirect enhancement with <i>volvo</i> intermediate noise	71
4.4	MSE between various learning pairs	71
4.5	3dB babble noise mixed with various colored noise at 3dB. The intermedi- ate target is <i>babble</i> for <i>Path 1</i> and the corresponding colored noise for <i>Path</i> <i>2</i>	75
4.6	3dB factory noise mixed with various colored noise at 3dB. The interme- diate target is <i>factory1</i> for <i>Path 1</i> and the corresponding colored noise for <i>Path 2</i>	75
4.7	<i>Babble</i> and colored noise at various SNR	76
4.8	<i>Factory1</i> and colored noise at various SNR	76

5.1	Effects of stationarity of conversion targets	87
5.2	PESQ after enhancing using different depths of auto-encoders	96
5.3	PESQ after enhancing using different bottleneck width	97
5.4	PESQ of converted speech at various SNR levels	97
5.5	Results of noise aware training	99
5.6	Results of domain adversarial auto-encoder	100
5.7	Effect of spectral compression in exemplar conversion	106
5.8	Effect of codebook size in NMF conversion	108
5.9	NMF-based conversion on various SNR levels	109

LIST OF FIGURES

2.1	Suppression vs. <i>a priori</i> SNR, ξ_k , of a Wiener filter	12
2.2	Suppression vs. <i>a posteriori</i> SNR, γ_k , of an MLE, spectral subtraction, and a Wiener filter	13
2.3	Comparison of the suppression gain between an MMSE and a Wiener filter	14
2.4	Progressive speech enhancement by SNR	21
2.5	SNR affecting the results of automatic speech processing systems	22
2.6	Progressive learning in speech enhancement	31
2.7	Relocation of noise in the time domain	32
2.8	Relocation of noise in the frequency domain	33
2.9	Framework of indirect speech enhancement	33
2.10	The existence of intermediate targets	36
3.1	PESQ of 100 types of Nonspeech noise	38
3.2	Spectrograms of samples noises. Top row contains difficult noise types. Middle row contains simple noise types. Bottom row shows the outliers . . .	40
3.3	Spectral shape of clean speech and some easy noise. Long-term average speech spectrum shown in red.	41
3.4	k-means clustering with t-SNE projection on 115 noise types	42
3.5	Long term average spectra of speech, <i>volvo</i> , and <i>ovlov</i>	46
3.6	Non-stationary examples of <i>volvo</i> -like noise	48

4.1	Framework of indirect speech enhancement	50
4.2	PCA projections of normalized features for different noises	53
4.3	The observed mean deviation agrees with the estimated mean deviation . . .	56
4.4	The observed variance deviation agrees with the estimated variance deviation.	58
4.5	Deviation from zero mean (left) and unit variance (right) between simple and difficult noise samples. <i>White</i> is an example of difficult noise and <i>volvo</i> is an example of simple noise.	63
4.6	Comparison of feature distribution by SNR and by noise	64
4.7	Effect of histogram equalization on feature distribution	65
4.8	Effect of histogram equalization in hidden layers	66
4.9	DNN architecture for feature mapping and joint training	68
4.10	Conversion and enhancement of speech in babble noise	69
4.11	Conversion and enhancement of speech in pink noise	70
4.12	Conversion and enhancement of speech in white noise	70
4.13	Effect of conversion on feature distribution	71
4.14	Indirect enhancement of multiple interferences	73
4.15	Average power spectrum density of speech and various colored noise	77
5.1	First 9 latent speech bases extracted from clean and noisy speech	84
5.2	First latent <i>white</i> noise basis	85
5.3	Conversion of noisy speech from <i>white</i> noise into <i>pink</i> and <i>volvo</i> with PCA	86
5.4	PCA vs Deep AE in manifold learning.	88
5.5	Use auto-encoders to convert noisy speech into simpler noise	90
5.6	Architecture of noise aware speech conversion	91

5.7	Use domain adversarial auto-encoders to convert noisy speech into simpler noise	92
5.8	Architecture of vector quantized auto-encoder	94
5.9	Conversion quality with respect to data size in auto-encoders	98
5.10	Size of codebook and dimension of codebook features	101
5.11	Dynamic range compression with exponentiation factor, ρ	104
5.12	Comparison between AE and NMF converted speech	107
5.13	Comparison between different codebook composition	108
C.1	Spectrograms and PSDs of some colored noise	124
E.1	Spectrograms and PSDs of Noisex92 noise	129
E.1	Spectrograms and PSDs of Noisex92 noise (cont.)	130

LIST OF ACRONYMS

AE	autoencoder
ASR	automatic speech recognition
CASA	computational auditory scene analysis
CDF	cumulative distribution function
CMVN	cepstral mean variance normalization
CNN	convolutional neural network
DFT	discrete Fourier transform
DNN	deep neural network
FFT	fast Fourier transform
GAN	generative adversarial network
GMM	Gaussian mixture model
HMM	hidden Markov model
IRM	ideal relative mask
KLD	Kullback–Leibler divergence
LPS	log power spectrum
LSTM	long short-term memory
MLE	maximum likelihood estimators
MLP	multilayer perceptron
MMSE	minimum mean square error
MSE	mean squared error
NMF	non-negative matrix factorization
OLA	overlap-add

PCA principal component analysis
PESQ Perceptual Evaluation of Speech Quality
PLP perceptual linear prediction
PSD power spectrum density
RASTA representations relative spectrum
RBM restrictive Boltzmann machine
ReLU rectified linear units
SE speech enhancement
SNR signal-to-noise ratio
STFT short-time Fourier transform
SVD singular value decomposition
SVM support vector machine
t-SNE t-distributed stochastic neighbor embedding
TF time-frequency
VAD voice activity detector
VAE variational auto-encoder
VQ vector quantization
WSJ Wall Street Journal

SUMMARY

Noise presents a severe challenge in speech communication and processing systems. Speech enhancement aims at removing the inference and restoring speech quality. It is an essential step in a speech processing pipeline in many modern electronic devices, such as mobile phones and smart speakers. Traditionally, speech engineers have relied on signal processing techniques, such as spectral subtraction or Wiener filtering. Since the advent of deep learning, data-driven methods have offered an alternative solution to speech enhancement. Researchers and engineers have proposed various neural network architectures to map noisy speech features into clean ones. In this thesis, we refer to this class of mapping based data-driven techniques collectively as a direct method in speech enhancement. The output speech from direct mapping methods usually contains noise residue and unpleasant distortion if the speech power is low relative to the noise power or the background noise is very complex. The former adverse condition refers to low signal-to-noise-ratio (SNR). The latter condition implies difficult noise types. Researchers have proposed improving the SNR of speech signal incrementally during enhancement to overcome such difficulty, known as SNR-progressive speech enhancement. This design breaks down the problem of direct mapping into manageable sub-tasks. Inspired by the previous work, we propose to adopt a multi-stage indirect approach to speech enhancement in challenging noise conditions. Unlike SNR-progressive speech enhancement, we gradually transform noisy speech from difficult background noise to speech in simple noise types. The thesis's focus will include the characterization of background noise, speech transformation techniques, and integration of an indirect speech enhancement system.

CHAPTER 1

INTRODUCTION

1.1 Overview

Single-channel speech enhancement aims at recovering clean speech from a mixture of interfering speech, background noise, and channel distortions [1]. Many classic speech enhancement techniques, such as spectral subtraction [2, 3] and Wiener filtering [4, 5], rely on an accurate estimation of the noise spectrum, usually calculated as a smoothed average of past observations during speech pauses. However, when the noise spectrum is non-stationary, an accurate estimation cannot be reliably obtained. Adaptive methods [6, 7, 8] partially alleviates this issue by recursively averaging the noise’s short-time spectrum. More recently, data-driven methods based on deep neural network (DNN) have achieved an impressive improvement in terms of perceptual quality and intelligibility, especially in some non-stationary noise conditions [9, 10, 11]. The DNN-based approach finds a non-linear function to map noisy speech features into enhanced features. Unlike many classical methods, this approach does not assume that speech or noise follows a particular distribution or independent. Authors of [9, 10] showed that the DNN-based approach was more effective in handling non-stationary noise compared to statistical models, such as minimum mean square error (MMSE) short-time spectral amplitude estimator [12], and rendered enhanced speech with better quality.

Many conventional speech enhancement algorithms have noted that non-stationary noise is typically more challenging to handle than stationary noise [8]. The disparity in speech enhancement performance in different backgrounds is also observed in DNN-based speech enhancement systems [13]. A direct mapping, such as the DNN in [10], does not address the variability of noise types or different signal-to-noise ratio (SNR) conditions. On the

other hand, an indirect approach decomposes the process of regression-based mapping into smaller tasks. Notable examples include [14, 15] in which the authors designed a series of sub-tasks to improve the SNR in a noisy speech signal incrementally. In the SNR-progressive learning paradigm, each sub-network explicitly learns an intermediate target with slightly higher SNRs. The authors of [14, 15] first mapped speech in acoustically challenging environments to a partially de-noised signal. Next, it was refined to clean speech in subsequent stages. The authors showed that this approach consistently outperformed direct mapping, especially in low SNR environments.

Instead of gradually improving the SNR in speech, we could consider replacing the background noise types to remove the noise with greater ease. In this indirect framework of enhancement, we first convert speech in challenging acoustic conditions, such as loud machinery or interfering speakers, to speech in less destructive noise, such as an office or home. Since speech in an office or home environment is simpler to be handled than speech in machinery or babble, we could refine it to clean speech with better quality. When the original acoustic environment is complicated, for example, an environment with multiple noise sources, the indirect approach can be extended to establish multiple intermediate representations with different background noise in the process.

Furthermore, the indirect approach based on noise type conversion is different from SNR-based progressive learning in [14, 15]. In their work, the authors of [14, 15] assumed that higher SNR in the signal corresponded to better speech quality. Hence, they designed SNR-based progressive learning to improve the SNR in each stage, which naturally led to an incremental improvement of speech quality. In contrast, there is no obvious criterion to gauge the difficulty level of noise types, even though one could find some noise conditions more disruptive to speech communication in daily life. Previous work [2, 12] have often cited non-stationarity of the background noise as a key factor responsible for the quality degradation. However, the discussion of noise characteristics sensitive to DNN-based enhancement is somewhat limited in the literature. Hence, we need a detailed characterization

of additive noise and its interaction with speech. We will use such knowledge to calibrate noise types and determine suitable intermediate targets in the proposed framework of indirect speech enhancement.

After anchoring the intermediate targets for noisy speech in difficult noise types, we design a noisy speech transformer that converts difficult speech to simpler speech. It is followed by a refinement module that maps the simpler speech to clean speech. An ideal noisy speech converter in this indirect approach should replace the background noise while keeping speech unchanged. Nonetheless, speech suppression and artifacts are usually inevitable [16]. We need to consider a trade-off between speech distortion and noise transformation to achieve optimal conversion and enhancement. Another issue with converter design is the availability of converted samples as training targets. Data-driven methods such as DNN-based mapping [10] often require a large amount of aligned data. Techniques developed with stronger assumptions of signal properties, such as vector quantization (VQ) [17], Gaussian mixture model (GMM) [18], and non-negative matrix factorization (NMF) [19], are usually less parameterized, so less or no aligned data is required to train these models. We will evaluate some of the methods above, too.

The third issue in converter design is integrating each sub-task into an overall speech enhancement system. The authors of SNR-progressive learning [14] adopted smaller DNNs to perform each conversion stage, so the networks could be easily concatenated and jointly optimized. On the other hand, it is not straightforward to combine a sample-level converter with a frame-level refinement module. As a result, all these factors need to be considered when designing the conversion and refinement steps in our indirect approach.

Thanks to substantial interest in voice conversion, music morphing, and hearing-aid design, an extensive collection of waveform or spectral conversion techniques have been proposed, including the aforementioned VQ, GMM, NMF, DNN, unit selection [20], and frequency warping [21]. We will select and compare various conversion techniques that best address the various issues in indirect speech enhancement.

1.2 Main contributions

This thesis aims to investigate the feasibility of a multi-stage speech enhancement approach by gradually replacing background noise in noisy speech. The contributions of our work are summarized as follows:

Our first contribution is the characterization of additive noises in the context of speech enhancement. In Chapter 3, we consider the frequency and temporal properties of noise signals and empirically evaluate their effects on speech enhancement. We also show how adverse noise conditions cause feature mismatch as a result of improper normalization.

Our second contribution is the design of speech transformation techniques using supervised learning. Chapter 4 presents our first architecture of the indirect approach to speech enhancement. We transform source noisy speech into intermediate target speech by matching their feature distribution or frame-level details. Experimental studies also demonstrate that we can extend the proposed method to handle multiple noise sources.

Our third contribution is the design of indirect speech enhancement and speech transformation when no parallel utterances are available for supervised learning. In Chapter 5, we leverage upon representation learning to discover hidden structures of speech and noise in noisy speech mixtures. The latent representation allows us to manipulate speech and noise separately by replacing background noise in the latent space. This operation accomplishes speech transformation, a critical step in our indirect speech enhancement framework.

Lastly, we conducted thorough experiments to validate the proposed framework of indirect speech enhancement. We use the knowledge derived from Chapter 3 to select reasonable intermediate targets. The speech transformers in Chapter 4 and Chapter 5 are combined with refinement modules to perform indirect multi-stage speech enhancement. Our experimental results show that the indirect approach can yield performance gain over direct mapping in challenging acoustic conditions.

CHAPTER 2

BACKGROUND

2.1 Speech enhancement

When speech is corrupted by background noise, speech enhancement can recover clean speech for better quality and intelligibility. We consider the following additive noise model in this thesis. The additive noise model assumes that the noise-corrupted speech or noisy speech, $y[n]$, is the sum of the clean speech signal, $x[n]$, and the additive interference, $d[n]$. In Equation 2.1, we assume that the speech and noise are additive and uncorrelated.

$$y[n] = x[n] + d[n]. \quad (2.1)$$

Converting the signal into the frequency domain offers the following advantages:

- Filters at different frequencies or frequency bands can be designed and handled independently from one another. Therefore, there is significant flexibility in dealing with colored noise, which generally possesses prominent frequency characteristics.
- Most of our knowledge and understanding of speech production and perception are related to frequencies.
- Thanks to fast Fourier transform (FFT)s, the implementation of frequency-domain filters is generally very efficient.

Because speech is a non-stationary signal in general, its temporal and spectral characteristics could vary over time. We can nevertheless assume that speech is stationary within a short analysis window, typically 10-30ms. We define the short-time N -point discrete

Fourier transform (DFT) [22] of the noisy speech signal, $Y(m, k)$:

$$Y(m, k) \triangleq \sum_{l=-\infty}^{\infty} y[l]w[mR - l]e^{\frac{-2j\pi lk}{N}}. \quad (2.2)$$

In Equation 2.2, $y[l]$ is the speech signal, and $w[l]$ is a window function of length N , such as a Hamming window [23] or a Hann window [24]. The hop size is R samples. The frame index, m , is the location of the analysis window. The frequency index, k , corresponds to the frequency at $2\pi k/N$, $k = 0, 1, \dots, N - 1$. We can define the short-time DFT of clean speech and the noise in the same manner. Thus, Equation 2.1 in the short-time frequency domain is

$$Y(m, k) = X(m, k) + D(m, k). \quad (2.3)$$

In its polar form, the DFT coefficients can be expressed as

$$Y(m, k) = |Y(m, k)|e^{j\angle Y(m, k)}, \quad (2.4)$$

where $|Y(m, k)|$ is the magnitude and $\angle Y(m, k)$ is the phase. The power spectrum, $P_y(k)$, can be defined as

$$P_y(k) = \mathbb{E}[|Y(m, k)|^2], \quad (2.5)$$

where the expectation is taken over the observed signal for a unit duration. The power spectrum of clean speech, $P_x(k)$, and that of noise, $P_d(k)$, can be defined similarly. From the power spectra, we can define two SNR quantities frequently used in the derivation of spectrum estimators: the *a priori* SNR, ξ_k , and the *a posteriori* SNR, γ_k

$$\xi_k = \frac{P_x(k)}{P_d(k)}, \quad (2.6)$$

and

$$\gamma_k = \frac{P_y(k)}{P_d(k)}. \quad (2.7)$$

The *a priori* SNR, ξ_k , represents the oracle SNR at the frequency bin, k , whereas the *a posteriori* SNR, γ_k , is the observed SNR at bin, k , in the noisy speech.

Because the authors of [25, 26] have shown that the phase spectrum does not affect intelligibility and it is less critical for speech quality, most works focus only on the restoration of the magnitude spectrum. Speech enhancement can then be formulated as an estimation of the clean magnitude spectrum, $|\hat{X}(m, k)|$, from the noisy speech magnitude spectrum, $|Y(m, k)|$. To convert the DFT of $\hat{X}(m, k)$ back to waveforms, we perform the inverse DFT and overlap-add (OLA) algorithm [27].

The performance of a speech enhancement system can be evaluated subjectively and objectively. In a subjective test, human listeners are asked to rate the quality of enhanced speech or identify intelligible words. These tests do not generally yield reliable conclusions on their own. They need to be combined with appropriate statistical tests to assess if a speech enhancement system can improve speech quality [28]. Furthermore, they are time-consuming and costly. In contrast, objective metrics are efficient and reliable if the metrics maintain a high correlation with subjective listening. Some commonly seen metrics include the log spectral distortion [29], weighted-slope spectral distance [30], segmental SNR [31], and Perceptual Evaluation of Speech Quality (PESQ) [32]. Among these objective measures, PESQ yielded the highest correlation with subjective assessments [33]. It ranges from -0.5 to 4.5, with higher scores indicating better speech qualities. It will serve as the main evaluation metric in this thesis.

Due to the importance of speech enhancement, the topic has received much attention in the speech community. The classical methods fit into three main categories. Spectral subtraction algorithms, first proposed by Weiss in the time domain [34] and Boll in the frequency domain [3], are the most intuitive to understand. They assume noise is additive, and one can obtain an estimate of clean speech by subtracting the noise spectrum estimated during speech pauses. The second primary class includes the statistical model-based algorithms. These algorithms consider speech enhancement in a statistical estimation frame-

work. They assume that the DFT coefficients of noisy speech depend on the DFTs of clean speech. The task is to find an estimator of the DFT coefficients of the clean signal. Notable algorithms include Wiener filtering [5], speech/non-speech detection [35], and MMSE estimators [12]. The last class, subspace algorithms, is based on linear algebra theory. These algorithms assume that clean speech is confined to a subspace of noisy speech. Vector decomposition techniques, such as singular value decomposition (SVD), can be exploited to separate the speech and noise subspaces. This line of work was pioneered by Dendrinos [36] and later by Ephraim and Van Trees [37]. The speech community has also studied and developed other approaches based on multilayer perceptron (MLP) [38, 39], GMM [40], principal component analysis (PCA) [41], mask estimation [42], NMF [43], support vector machine (SVM) [44], and more recently, DNN [9, 10, 45]. In the rest of the section, we will briefly review the three major classes of noise enhancement algorithms' basic principles.

2.1.1 Classical signal processing techniques

Spectral subtraction

Spectral subtraction, first proposed in [3], is one of the most intuitive and heuristic methods. This class of algorithms exploits the assumption that background noise is additive and stationary. In its most basic form, one first obtains an estimate of the noise spectrum during speech pauses using a voice activity detector (VAD). Since noise is assumed to be stationary, its spectrum does not change at the next speech onset. We recover the clean speech spectrum by subtracting the noise spectrum from the noisy speech spectrum. We then update the estimate of the noise spectrum at the next speech pause. To recover speech waveform from the frequency domain, one performs inverse DFT and OLA in reconstruction. The following equation summarizes the principle of spectral subtraction

$$|\hat{X}(m, k)| = |Y(m, k)| - |\hat{D}(m, k)|. \quad (2.8)$$

In Equation 2.8, $|\hat{D}(m, k)|$ and $|\hat{X}(m, k)|$ denote the *estimated* noise spectrum and clean speech spectrum at frequency bin, k , respectively.

Alternatively, spectral subtraction can also be formulated with the power spectrum instead of the magnitude spectrum

$$|\hat{X}(m, k)|^2 = |Y(m, k)|^2 - |\hat{D}(m, k)|^2. \quad (2.9)$$

We can rearrange the terms in Equation 2.9 and make substitutions with *a priori* and *a posteriori* SNR

$$\begin{aligned} |\hat{X}(m, k)| &= \sqrt{|Y(m, k)|^2 - |\hat{D}(m, k)|^2} \\ &= \sqrt{\frac{P_y(k) - \hat{P}_d(k)}{P_y(k)}} |Y(m, k)| \\ &= \sqrt{\frac{\xi_k}{\xi_k + 1}} |Y(m, k)| \\ &= \sqrt{\frac{\gamma_k - 1}{\gamma_k}} |Y(m, k)|. \end{aligned} \quad (2.10)$$

In linear systems, the factor in front of $|Y(m, k)|$ is known as the system's transfer function. In the context of speech enhancement, it is also referred to as the suppression gain [16]. Hence, the suppression gain, $H(k)$, for spectral subtraction is

$$H(k) = \sqrt{\frac{\xi_k}{\xi_k + 1}} = \sqrt{\frac{\gamma_k - 1}{\gamma_k}}. \quad (2.11)$$

Spectral subtraction is straightforward to understand and implement. It is also an efficient algorithm as it only requires one forward computation in the subtraction [16]. Because the clean speech magnitude estimate, $|\hat{X}(m, k)|$, must stay positive, one must exercise caution in applying Equation 2.8. An easy solution is to apply a half-wave rectifier on the

difference spectrum

$$|\hat{X}(m, k)| = \begin{cases} \sqrt{|Y(m, k)|^2 - |\hat{D}(m, k)|^2}, & \text{if } |Y(m, k)|^2 \geq |\hat{D}(m, k)|^2 \\ 0. & \text{otherwise} \end{cases} \quad (2.12)$$

Equation 2.12 ensures that the estimated speech magnitude always stays non-negative. Nevertheless, the nonlinear truncation of negative values creates short and unrelated peaks in the speech spectrum. After converting the signal back to the time domain, these peaks translate to tones with frequencies varying from frame to frame. Such distortions are commonly referred to in the literature as *musical noise* [2]. They are particularly noticeable during an unvoiced speech where the speech power is relatively low. Some studies have reported that musical noise can be perceptively more disruptive to human listeners than the original background noise [16]. For this reason, much research has gone into finding ways to reduce musical noise.

A notable example of works in this area is spectral oversubtraction by Berouti [2]. It was motivated by the observation that some peaks in the difference spectrum, $|Y(m, k)| - |\hat{D}(m, k)|$, were broadband, whereas others were narrowband. By subtracting an amplified noise spectrum controlled by an augmentation factor, one could reduce the broadband peaks' magnitude. Oversubtraction also levels deep valleys in the spectrum by applying a spectral floor when speech was absent. Berouti conducted thorough empirical studies to evaluate the choice of the augmentation factor and spectral floor level.

The oversubtraction method was further extended by Lockwood *et al.* in [46]. They modified the augmentation factor so that it depended on the frequency. The modification was motivated by the observation that much real-world noise affected different frequency regions unevenly. Thus, larger values should be subtracted from frequency bands with low SNR; smaller values should be subtracted from bands with high SNR.

The effectiveness of spectral subtraction methods has been studied extensively. Most studies concurred that this class of algorithms improves speech quality but not speech in-

telligibility [47, 48]. Its adverse effect on speech intelligibility can be explained by the occasional elimination of low-power speech region due to inaccurate noise estimation. It remains an open question if more advanced noise estimation techniques can improve speech intelligibility in spectral subtraction methods.

Wiener filters

Another well-known class of speech enhancement algorithms is derived from the Wiener filtering [4] by minimizing the mean squared error (MSE) between the filtered output and the desired response. Recall the short-time DFT coefficients defined in Equation 2.2. The desired output is the clean signal, $X(m, k)$. The frequency response of the Wiener filter is denoted as $H(k)$. Hence, the filtered output is $H(k)Y(m, k)$. One can then define an error signal, $E(m, k)$ for each frequency bin, k , at each frame, m , as

$$E(m, k) = X(m, k) - H(k)Y(m, k). \quad (2.13)$$

The Wiener filter minimizes the energy of the error signal in Equation 2.13

$$\begin{aligned} \hat{H}(k) &= \arg \min_{H(k)} \mathbb{E}[|E(m, k)|^2] \\ &= \arg \min_{H(k)} \mathbb{E}[(X(m, k) - H(k)Y(m, k))^* (X(m, k) - H(k)Y(m, k))], \end{aligned} \quad (2.14)$$

In the equation above, $*$ is the complex conjugate.

We could determine the minimizer of the error signal's energy by taking its derivative with respect to $H(k)$ and set it to zero. The solution to the Wiener filter in the frequency domain is

$$\hat{H}(k) = \frac{\mathbb{E}[X(m, k)^2]}{\mathbb{E}[Y(m, k)^2]} = \frac{P_x(k)}{P_y(k)} = \frac{P_y(k) - P_d(k)}{P_y(k)}. \quad (2.15)$$

In Equation 2.15, the power spectrum density (PSD) of the clean signal, $P_x(k)$, is generally not available. It is approximated by $P_y(k) - P_d(k)$ by assuming the speech and the noise

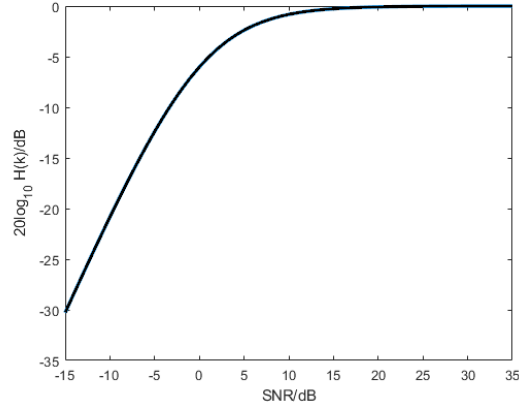


Figure 2.1: Suppression vs. *a priori* SNR, ξ_k , of a Wiener filter

are uncorrelated.

The Wiener filter in the frequency domain can also be written in terms of the *a priori* and *a posteriori* SNR as

$$\hat{H}(k) = \frac{\xi_k}{\xi_k + 1} = \frac{\gamma_k - 1}{\gamma_k} \quad (2.16)$$

It can be seen from Equation 2.16 that $0 \leq \hat{H}(k) \leq 1$. When $\xi_k \rightarrow \infty$, i.e., SNR is high, $\hat{H}(k) \approx 1$, which means there is no noise suppression. On the other hand, when $\xi_k \rightarrow 0$, i.e., SNR is low, $\hat{H}(k) \approx 0$, suggesting complete attenuation of the spectrum. In [16], the author plots the suppression gain of a Wiener filter with respect to the *a priori* SNR, ξ_k , replicated in Figure 2.1. There is almost no suppression at $\xi_k > 10dB$. For SNR below -5dB, attenuation becomes linear with respect to the SNR.

Compared to spectral subtraction, Wiener filtering is more aggressive in de-noising. The filtered clean signal's power is always lower than the oracle clean signal's power, which attributes to speech attenuation in a Wiener filter [49].

Statistical model-based methods

In this framework, the DFT of noisy speech serves as a set of measurement that depends on some unknown parameters, which are the DFTs of clean speech. We need to estimate these unknown parameters given the observation of noisy speech. Literature in estimation theory

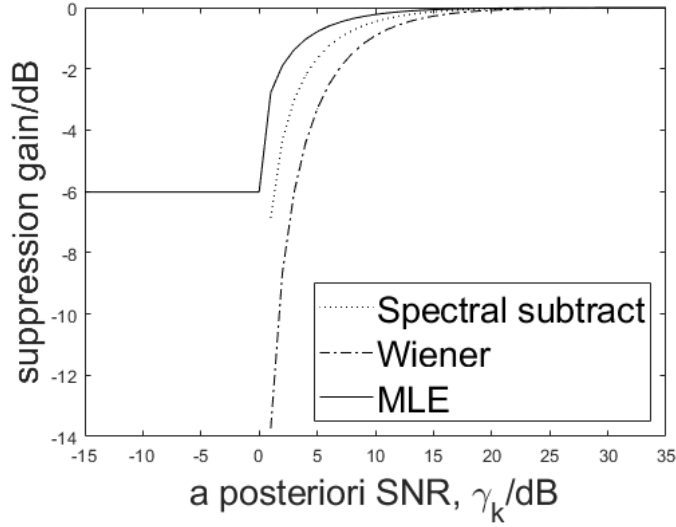


Figure 2.2: Suppression vs. *a posteriori* SNR, γ_k , of an MLE, spectral subtraction, and a Wiener filter

has provided us tools to derive these estimators, including maximum likelihood estimators (MLE) [35] and Bayesian estimators [16]. A major difference between these nonlinear estimators is that MLE assumes the parameters of interest are deterministic but unknown, whereas the Bayesian estimators assume that the parameters are random variables.

The MLE of the magnitude spectrum of clean speech is [16]

$$|\hat{X}(m, k)| = \frac{1}{2}|Y(m, k)| + \frac{1}{2}\sqrt{\frac{\gamma_k - 1}{\gamma_k}}|Y(m, k)|. \quad (2.17)$$

The author in [16] compared the suppression gain of spectral subtraction, Wiener filter, and MLE in Figure 2.2. The suppression gain of an MLE is plotted in solid line. Compared to spectral subtraction and the Wiener filter, it suffers from insufficient attenuation because of the noisy speech residue, $\frac{1}{2}|Y(m, k)|$. It is thus rarely used by itself [16].

The MMSE estimator, introduced by Ephraim and Malah, shares a similar motivation as the Wiener filter. When formulating the Wiener filter in Equation 2.13, we attempt to minimize the error signal of the complex spectrum. In order to derive the optimal magnitude estimator, Ephraim and Malah proposed to minimize the MSE, $E(k)$, between the

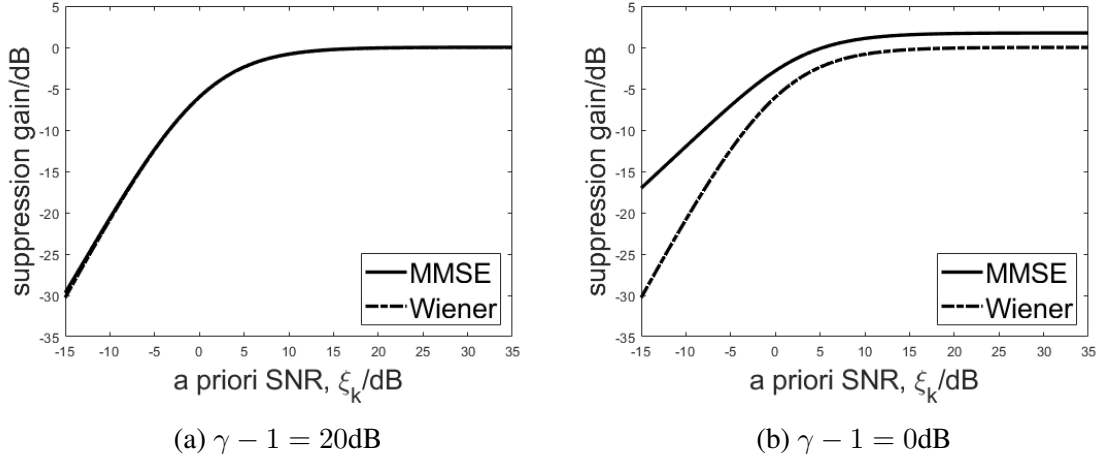


Figure 2.3: Comparison of the suppression gain between an MMSE and a Wiener filter

ground truth, $|X(m, k)|$, and the estimated magnitude, $|\hat{X}(m, k)|$:

$$E(k) = \mathbb{E}[(|X(m, k)| - |\hat{X}(m, k)|)^2]. \quad (2.18)$$

In their subsequent work, Ephraim and Malah further suggested minimizing the MSE between the log-magnitude spectra because they could be more subjectively meaningful [50]

$$E(k) = \mathbb{E}[(\log |X(m, k)| - \log |\hat{X}(m, k)|)^2]. \quad (2.19)$$

The derivation of the solutions to the optimal estimators in Equation 2.18 and Equation 2.19 was relatively involved. The closed-form solutions can be found in [12, 50]. The suppression gains depend on both ξ_k and γ_k . We can compare the suppression gain of an MMSE estimator with that of a Wiener filter. Figure 2.3 shows the suppression gain of an MMSE estimator (solid), and a Wiener filter (dotted) plotted against the *a priori* SNR, ξ_k . Figure 2.3a shows that when the *a posteriori* SNR is high, the MMSE gain is similar to that of the Wiener filter. When the *a posteriori* SNR is low in Figure 2.3b, we could tell that the MMSE estimator is not as aggressive as the Wiener filter. This behavior could help reduce speech distortion in adverse conditions.

The authors of [12, 51] compared the performance of an MMSE with spectral subtrac-

tion, Wiener filter, and MLE. They found that there was no perceptible musical noise if the *a priori* SNR was estimated correctly. It also resulted in less speech distortion compared to Wiener filters. The cause for the effective suppression of musical noise was discussed in detail by Cappe [52]. He discovered that the suppression relied on a reliable estimation of the *a priori* SNR, ξ_k , more so than the *a posteriori* SNR, γ_k . Consequently, the suppression in the MMSE estimator will not change abruptly from frame to frame. On the other hand, algorithms like spectral subtraction relied more heavily on the estimation of the *a posteriori* SNR, which might change rapidly between frames. Hence, MMSE yielded a smoother transition and avoided undesirable musical noise.

Subspace methods

Another major class of classical speech enhancement technique is derived from linear algebra theory. The subspace methods seek to decompose noisy speech into a signal subspace and a noise subspace. The signal subspace could be retrieved by nulling the noise subspace using algebraic tools, such as SVD or eigenvector-eigenvalue factorization [53].

After arranging speech samples into a matrix, \mathbf{Y} , one can estimate the speech matrix, \mathbf{X} , using either the least square approach with low-rank modeling or the minimum variance approach. The least-square approach [16] is formulated as

$$\hat{\mathbf{X}}_{LS} = \arg \min_{\hat{\mathbf{X}}} \|\hat{\mathbf{X}} - \mathbf{X}\|_F, \quad (2.20)$$

where $\|\cdot\|_F$ is the Frobenious norm of a matrix. The solution is given as the truncated SVD of \mathbf{Y} with presumed rank, r

$$\hat{\mathbf{X}}_{LS} = \mathbf{U}_y \Sigma_y \mathbf{V}_y^H = \sum_{j=1}^r \sigma_j \mathbf{u}_j \mathbf{v}_j^H. \quad (2.21)$$

In the equation above, the superscript, H , is the Hermitian transpose of a matrix.

The minimum variance approach determines transfer matrix, \mathbf{H} , such that reconstruc-

tion error is minimized [54, 55]

$$\hat{\mathbf{H}} = \arg \min_{\mathbf{H}} \|\mathbf{YH} - \mathbf{X}\|_F^2. \quad (2.22)$$

The estimator is given as

$$\hat{\mathbf{X}}_{MV} = \mathbf{U}_y \boldsymbol{\Sigma}_{MV} \mathbf{V}_y^H = \sum_{j=1}^r \frac{\sigma_j^2 - \sigma^2}{\sigma_j} \mathbf{u}_j \mathbf{v}_j^H, \quad (2.23)$$

where σ is the variance of noise. One can observe that both estimates share the same vectors but use different estimates for singular values.

The first use of SVD in speech enhancement appeared in [36]. The authors assumed that larger singular vectors with larger values corresponded to speech vectors. He demonstrated it to be a practical algorithm for noise reduction at high SNR. At low SNR (around 0dB), there was perceptible musical noise [56, 57]. Subsequent work showed that a higher rank, r , is required for unvoiced speech than for voiced speech [16]. The minimum variance approach was also less sensitive to the choice of r than the least square approach [56]. Authors of [51] also found that subspace methods generally did not improve speech qualities as much as statistical methods, such as the log-MMSE estimator, but some [37, 58] could outperform log-MMSE in terms of speech intelligibility.

2.1.2 Deep learning methods

The use of MLP as nonlinear filters to predict the clean spectrum dated back to the 1990s [38, 39, 59, 60]. An MLP is an extension of Rosenblatt's perceptron [61] by inserting hidden layers between the input and output layers. The MLP is parameterized by weights and biases. The weights, \mathbf{W} , are matrices connecting adjacent layers. The bias, \mathbf{b} , is added to each layer's output to model any linear shift in the data distribution. Forward propagation through a layer, j , is a matrix product between the layer's input, \mathbf{v}_j , and the weight matrix in that layer, \mathbf{W}_j . The output, \mathbf{v}_{j+1} , is then added to the corresponding bias, \mathbf{b}_j . Generally,

an element-wise nonlinear operation such as sigmoid or rectified linear units (ReLU) [62] function is also inserted between hidden layers to prevent the MLP from degenerating into a linear operation. Forward propagation in a layer with a ReLU activation is expressed as

$$\mathbf{v}_{j+1} = \text{ReLU}(\mathbf{W}_j \mathbf{v}_j + \mathbf{b}). \quad (2.24)$$

The weights and biases are initialized randomly. The predicted output can be computed layer by layer, according to Equation 2.24. At the last output layer of an M -layer MLP, the prediction \mathbf{v}_M can be compared with the ground truth, \mathbf{u} , with an appropriate loss function. In the case of speech enhancement, MSE loss is most widely used [10].

$$L_{MSE} = \frac{1}{N} \|\mathbf{v}_M - \mathbf{u}\|_2^2 = \frac{1}{N} \sum_{i=1}^N (v_i - u_i)^2. \quad (2.25)$$

Here the dimension of the output and the ground truth is assumed to be N . By computing the gradient of each parameter in the network with respect to the chosen loss function, we could perform iterative updates on the parameters to reduce the overall loss. The gradients of lower layers that are not directly connected to the output can be computed using the chain rule. This principle is known as back-propagation [63]. With stochastic gradient descent, the parameters are updated by a small amount in the negative direction of the gradient. This update lowers the loss after each iteration. This gradient descent step is the most basic procedure in the optimization of MLP [64]. After successive updates of the parameters, the loss would decrease and converge, when the network training is complete. Past research [65, 66] has shown that MLPs are universal approximators that can describe a wide variety of functions if they have sufficient width per hidden layer.

During the early stages of the application of MLPs in enhancement, the neural networks typically have relatively small sizes. Each layer has fewer than 200 neurons. There is also no consensus on the best features or targets for the mapping. Time-domain waveforms were used directly in [38, 39]. Log spectral features were adopted in [67]. In [68], the

author estimated the instantaneous SNR of spectrograms to suppress noise. However, the frequency resolution was low, and the system was unable to handle noise with sharp spectral peaks. In general, neural networks back then were usually shallow in terms of the number of layers and small in terms of the number of hidden neurons per layer. One of the major limitations of an MLP is its lack of closed-form solutions. The error surface of the loss function is generally not convex. Hence, there is no guarantee that a local minimum found by the gradient descent algorithms is a global minimum. Furthermore, MLPs with too many layers cannot be trained (in the sense of reducing the training loss) because of the *vanishing gradient* problem [69]. It refers to the phenomenon that the gradients become so small at layers close to inputs that the parameters could not be updated. As a result, most of the early work relied on the use of single-layer MLPs. Understandably, the complex nonlinear relationship was tough for a small network to approximate. The performance of speech enhancement with shallow networks was unsatisfactory.

In [70], Hinton, *et al.* first used restrictive Boltzmann machine (RBM)s to train MLPs layer-wise without labels greedily. This unsupervised pre-training step yielded better initialization for parameters in each layer. This procedure allowed DNNs with layers of pre-trained RBMs to be fine-tuned [70]. It also alleviated the vanishing gradient problem using other nonlinear activation, such as ReLU in place of sigmoid function [62]. Gradually even pre-training was no longer considered necessary if a large amount of training data is available.

Inspired by the break-through of MLPs in automatic speech recognition (ASR), speech enhancement based on DNN flourished in subsequent years. In [71, 72], DNNs were used to perform binary classification of sub-bands of noisy speech into speech or noise dominated bins. The classification results were used as ideal binary masks to recover clean speech, similar to masks in computational auditory scene analysis (CASA) [73]. The authors postulated that neural networks could learn more discriminative features than spectral features. Afterward, they concatenated the DNN's output to an SVM to estimate the mask.

Their design was a tandem system where DNNs were used mainly for feature extraction, and other classification algorithms were required to make the final classification decision. However, researchers soon experimented with stacking DNNs on top of other DNNs to create more “all-neural” models [74]. Authors of [75] used a deep de-noising auto-encoder in place of RBM to pre-train a DNN. They mapped noisy speech features directly to clean features.

Concurrently, authors of [9] framed speech enhancement as a regression task to map the log power spectrum (LPS) of noisy speech to clean speech. Unlike [75], the DNN in [9] was a standard MLP with RBM pre-training. Multiple frames were concatenated as inputs to include more temporal information, which significantly helped the enhancement quality. Experimental results showed improved speech quality in terms of both subjective and objective measures over traditional methods. Notably, the deep learning-based models could more effectively handle non-stationary noise and yielded enhanced waveforms with little music noise commonly found in traditional techniques. A primary reason for the systems’ effectiveness in [9] could be attributed to its use of a large volume of noise types and SNR conditions to simulate noisy speech during training.

Subsequent research directions included the use of other neural network models, such as convolutional neural network (CNN) [76], fully convolutional networks [77], and long short-term memory (LSTM) [78]. DNN with skip connections between non-consecutive layers were investigated to obtain better enhancement quality [79]. A myriad of work explored the suitability of other learning targets besides spectral features, such as ideal relative mask (IRM) [75, 76], phase-sensitive mask [80], and complex IRM [81]. More recently, direct mapping of speech waveforms in the time domain has also been attempted [11, 82]. The use of adversarial learning, such as speech enhancement generative adversarial network (GAN) [83, 84], also led to many new approaches to deep learning-based speech enhancement.

2.1.3 Progressive speech enhancement

Researchers have observed that speech quality improvement with a DNN-based speech enhancement system is not uniform across SNR levels [15] and noise types [13]. Noisy speech in lower SNR still contains many noticeable noise residues and artifacts after enhancement. This observation is in line with our intuition since lower SNR implies more severe distortion in the input signal. Hence more details have to be recovered. The performance gap between speech in different background noise further suggests that noise types, just as SNR, could be classified into difficult versus simple groups.

The work in [14, 15] pioneered a multi-stage indirect approach to speech enhancement. In their progressive learning framework [14], the authors divided direct mapping between noisy and clean speech with a DNN into multiple stages. The signal gained higher SNR as it propagated through the system. To enforce SNR gains in the neural network, the authors provided explicit learning targets at intermediate SNR levels as secondary labels in selected hidden layers. It effectively decomposed the neural network into a sequence of smaller networks, which only needed to handle simpler tasks in each stage due to smaller SNR differences. Smaller neural networks with constrained learning capability were stacked in [85] to approximate a larger teacher network’s performance. The stacked smaller networks also showed gradual improvement in speech quality as noisy speech propagated through the sequential model. This process is shown in Figure 2.4, where a noisy speech in *babble* noise at 0dB is gradually enhanced to 5dB, 10dB, and ultimately to an estimate of the clean speech.

The work in [14, 15] has inspired several more studies pursuing progressive learning in speech enhancement. In [86], the author divided the enhancement task into two sub-tasks: suppression of additive noise and dereverberation. They then combined the sub-tasks into an overall enhancement task. The system in [86] had three branches, one for each sub-task. Each task had three difficulty levels guided by different intermediate targets. For example, the dereverberation task had intermediate targets of gradually weaker reverberation. The

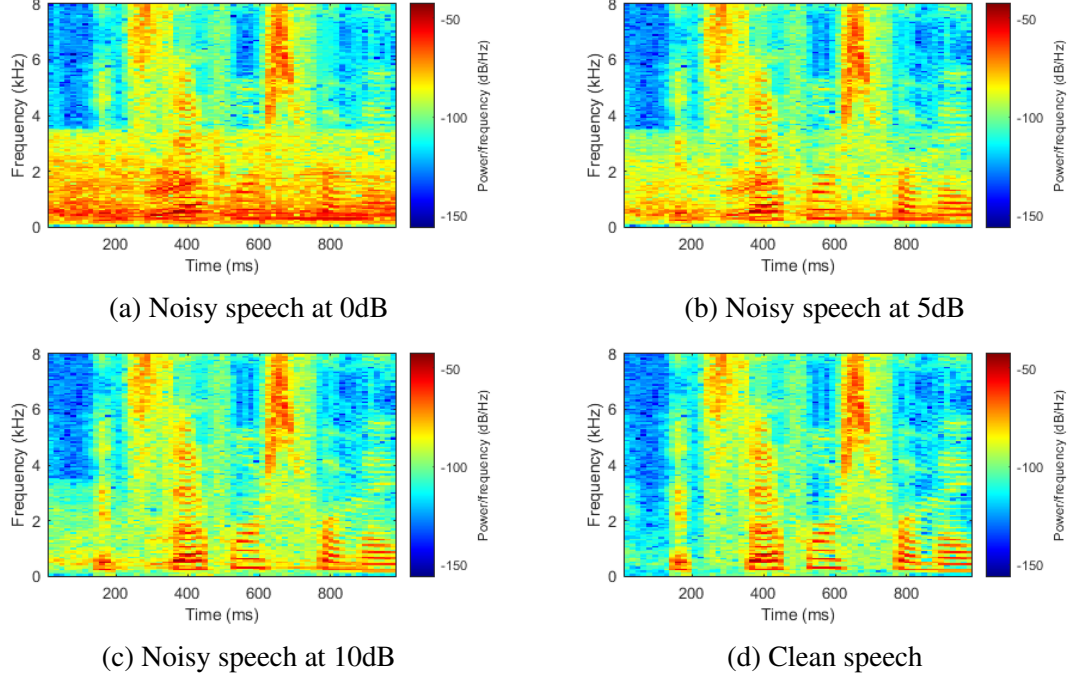


Figure 2.4: Progressive speech enhancement by SNR

suppression task’s intermediate targets were noisy speech in incremental SNR. The third task combined the intermediate outputs from the previous two sub-tasks and the original noisy speech. It used pre-enhanced features from simpler tasks to enhance the original difficult speech.

Following the progressive learning paradigm, authors in [87] examined the intermediate outputs and offered some analysis in each enhancement block’s behavior in the pipeline. They found that earlier blocks working with signals in lower SNR took care of the more noticeable distorted areas of the spectrum. The network did this by establishing a pattern of what was distortion and what was not. They also noted that the network mainly focused on the spectrum valleys where SNR was low. The later modules softened the spectrum in order to produce slow spectral magnitude changes. This operation avoided undesirable auto-generated distortions, such as annoying musical noise. However, it could also cause an over-softening effect in the final spectrum output.

The design of the progressive learning pipeline in [88] focused more on using efficient

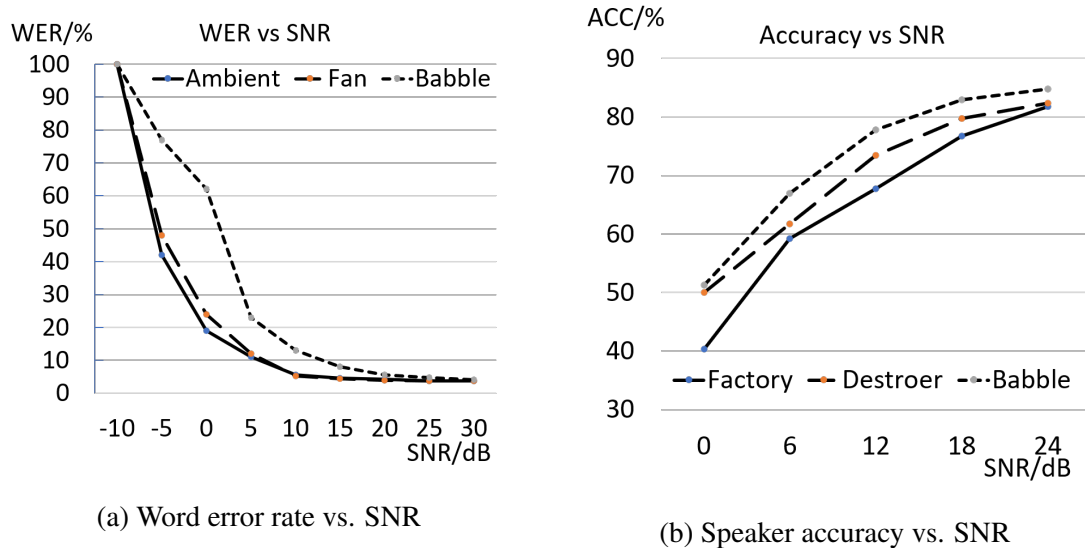


Figure 2.5: SNR affecting the results of automatic speech processing systems

models for parallelism and real-time learning. They promoted parameter sharing because of similar functionalities of enhancement blocks among different stages and the demand for small models. Moreover, a more parameter efficient model, convolutional-recurrent neural network, was used instead of LSTM in their work.

These methods fall under the umbrella of *curriculum learning* [89] in which a bigger or tougher task is dissected into simpler sub-tasks hoping that each smaller task can be better designed and trained for overall better performance. The experiments in [14, 15, 86] have confirmed that progressive learning is a useful technique in highly adverse conditions with low SNR or high reverberation.

2.2 Characterization of noise

Low SNR poses problems for speech processing systems. In Figure 2.5, one could tell low SNR drastically increases the word error rates on an ASR system [90]. Similarly, low SNR also adversely affects the accuracy of speaker identification [91]. Even though the effects are different across noise types, the overall trend showing degrading performance with respect to lower SNR is consistent.

The difficulty of noise types cannot be interpreted straightforwardly. The source of additive noise is ubiquitous in everyday acoustic environments, such as traffic noise outdoors, babble noise in a meeting room, and even electrical noise in microphones. Due to high variabilities, the discussion on the interaction between general noise types and speech in speech enhancement is limited.

Noisex92 [92] highlighted the drastic performance differences on a noise masking recognizer among noise types but did not extend the discussion into the properties of each noise contributing to their performance. A particular noise type, car environment, was discussed in [93] using a harmonic noise model. Even though it was designed for a specific noise type, the analysis-synthesis framework using harmonic noise models could be extended to other similar noise. Authors of [94] adopted a post-processing step after traditional speech enhancement (SE) methods to cope with factory-like noise with a high burst of energy in stationary noise. In [95], the authors noted that white likely reduced the dynamic range of cepstral coefficients within frames. The difficulty of recognizing each phoneme was assessed in [96]. The authors reported that consonants, including "s, sh, zh (as in vision)," were resistant to noise, including speech-shaped noise, babble noise, or white noise. These fricatives occupied the high-frequency bands that were less corrupted by the aforementioned noise. Other consonants did not have a steady classification accuracy due to different noise types reported in each work [97, 98, 99]. In [100], the author acknowledged the importance of designing noise-aware hearing aids for speech enhancement. It used energy-based features to first identify the presence of speech under the assumption that the background noise remained wide-sense stationary in a sustained noise environment. The deviation in the input speech signal's energy level was computed for stationary, non-stationary, and semi-stationary noise to perform classification.

The authors of [101] attempted noise classification in a limited scope and utilized this classification result in a DNN-based enhancement. First, a VAD isolated speech-absent frames. Then a GMM-based classifier determined the noise type. They found that the noise

specific enhancement model achieved better objective speech quality than noise agnostic systems. Most recently, authors of [102] applied different noise types from Noisex92 [103] corpus to analyze how the spectrum of each noise affected formant shifts. They found that wideband noise, such as *white noise*, consistently has a greater distortion on formant locations than narrowband noises, such as *volvo car noise* across between 5 to 15 dB.

In order to develop more noise resistant features, researchers have proposed various pre-processing steps throughout the decades. Unfortunately, those methods only work best when noise follows a presumed pattern. For example, researchers in [104] proposed the use of cepstral mean normalization to remove the mean value from all cepstral vectors. This technique is effective in counteracting the effect of channel distortion, but not additive noise. Representations relative spectrum (RASTA) proposed in [105] attempted to suppress constant additive noise in every log spectral component of the short term spectrum. This method has been extended to mel and perceptual linear prediction (PLP) features [106]. Its effectiveness was demonstrated in [107]. Nevertheless, it is impractical in sub-word models due to its high memory usage. In summary, these feature pre-processing steps are effective in filtering out steady noise but less useful in real-world non-stationary noise environments.

The separation of noise interference from the speech is also a topic of study in the field of *auditory scene analysis*. Human listeners tend to separate and group audio objects before identification [108]. The clues for clustering and separating include onset-offset time, temporal dynamics of amplitudes and frequencies, and spatial locations. More formally, auditory scene analysis parses auditory inputs into perceptual objects representing either physical sources or temporal sound patterns, such as melodies, which contribute to sound waves reaching the ears [109]. CASA is the study of auditory scene analysis by computational means [73]. Some advances in CASA include Bayesian principles [110], neural models [111], and temporal coherence models [112].

Although there have been limited efforts to quantify the effects of noise in speech enhancement systems, its effects on speech perception have received more attention. In [113],

the authors observed that people were less sensitive to acoustic stimuli, including noise or artifacts near high energy regions in speech, such as formant peaks. When listening to speech in a noisy environment, human listeners could reduce the noise effect by a masking mechanism. The phenomenon, called *noise masking*, has been exploited in the design of speech coding and enhancement systems [114, 115, 116]. Artifacts, such as quantization noise, were masked by formant peaks, hence became inaudible to human ears [117]. Speech coders were designed with perceptually weighted error criterion [115, 118], which placed more emphasis on spectral valleys where the noise was more noticeable than at spectral peaks. Noise floor normalization [113] was applied after filterbank analysis and log operation to modify the noisy speech spectra so that the system became more resistant against variations in background noise. The noise floor was chosen such that only bands with energy higher than the threshold were considered in the classification process. Subsequent work used a global noise threshold [114]. It has also been shown that the technique could be applied to a hidden Markov model (HMM) in speaker-dependent digit recognition, where improvement was achieved in low SNRs [119].

2.3 Speech transformation

As both enhancement and conversion require a transformation from source speech to target speech, many techniques applicable to speech enhancement are also suited for speech transformation. Voice conversion is a popular topic studied under the umbrella of speech transformation [120]. It generally attempts to modify a source speaker's speech signal to that of a target speaker while maintaining the linguistic contents intact. Even though voice conversion is not required or even desired in enhancement systems, some techniques that transform the speech spectrum could be modified to apply speech enhancement. The similarities stem from the fact that both speech enhancement and voice conversion traditionally require analysis and synthesis of the speech signal. Transformation is performed at a frame level by establishing a mapping between spectral features. The transformed magnitude

spectrum and phase spectrum are then synthesized to reconstruct the speech waveform. Indirect speech enhancement, which will be introduced on a high level in the next section, outputs speech in another background noise as intermediate outputs. This conversion process mimics some aspects of voice conversion, which also modify some speech signal characteristics. In the next section, we will survey some typical voice conversion methods that can be applied to speech enhancement and transformation.

2.3.1 Mean-variance normalization

Though not used as a speech transformation technique by itself, mean-variance normalization of speech vectors has found many uses as a feature pre-processing step in many tasks, including recognition, enhancement, and speaker identification [121]. The linear transform works with many types of speech features, including but not limited to power spectral density [122], the cepstral features [123], line spectral pairs [124], and perceptual linear prediction [125]. It shifts and scales speech features to an appropriate range for downstream processing. It is an effective technique to ensure the training and test data follow a similar distribution. This adjustment makes the overall speech recognition or enhancement more robust against changing noise conditions [49].

In its most basic form, mean normalization subtracts the mean statistic from each utterance. The resulting feature vector will have zero mean in each feature dimension. Variance normalization standardizes each feature dimension's variance to 1 by dividing the feature vector by the estimated standard deviation. If required, it is done after the mean normalization. The validity of mean normalization of cepstral features lies in the fact that the channel effect becomes linear in the cepstral domain. For example, distortion at the microphone can be modeled as linear filtering on the signal. The distortion varies depending on the transfer function of the electronics in the microphones, the distance between the speaker and the microphones, and the room acoustics. After removing the sampled mean from the feature, the effect of such channel distortion can be reduced. Unlike mean normalization, cepstral

variance normalization lacks a physical interpretation [49]. Still, many empirical studies have confirmed its usefulness in scaling the speech feature vectors to a better range [126].

Histogram equalization extends the idea of mean-variance normalization to higher moments [127]. Similar to mean and variance normalization, feature transformation is also performed in each dimension. However, a target histogram has to be determined beforehand. A unit Gaussian distribution can be selected when no prior information is available [49]. Several studies have found that though more complicated, histogram equalization does not yield a significant performance boost over simple mean-variance normalization [126].

Works in this field also include selections of different feature types, speech tasks, and, most importantly, methods to estimate the desired mean and variance statistics. Three cepstral mean variance normalization (CMVN) techniques are proposed and compared in [128]. The authors concluded that the long-term average is better than the short-term average and maximum likelihood estimate with respect to the model parameters. SNR dependent cepstral normalization was first introduced in [129]. It applied an additive correction dependent on the instantaneous SNR of the signal. The algorithm was simple and effective, but it required environment-specific training. Fixed codeword-dependent cepstral normalization was subsequently developed to provide greater recognition accuracy than the SNR dependent normalization [130]. It was further extended into multiple fixed codeword-dependent cepstral normalization. It exploited the simplicity and effectiveness of fixed codeword normalization, yet it did not need environment-specific training. Lastly, authors of [131] and [132] studied online and recursive normalization to enable feature normalization in real-time speech processing.

In most literature above, mean-variance normalization is performed on clean speech or noisy speech. Speech in different backgrounds does not receive special attention regarding applying different normalization statistics depending on the environment. This thesis will consider the effects of noise on feature normalization and adjust its use for indirect speech

enhancement.

2.3.2 Exemplar-based methods

This class of methods assumes converted speech, $\hat{\mathbf{X}}_{tgt}$, can be decomposed as a linear sum of a set of exemplars, \mathbf{t}_i

$$\hat{\mathbf{X}}_{tgt} = \sum_i w_i \mathbf{t}_i = \mathbf{w}^T \mathbf{T}. \quad (2.26)$$

Each exemplar, \mathbf{t}_i , is a row in the matrix, \mathbf{T} . The weight of each exemplar, w_i , form the weight vector, \mathbf{w} . The weights are computed to minimize the distance between the source and target features. It is desired for \mathbf{w} to be sparse as too many non-zero weights may cause the combined features to be over-smoothed. Since either the magnitude or power spectrum is guaranteed to be non-negative, assuming the weights are non-negative too, one can use NMF to solve the sparse weights iteratively [19, 133]. Its robustness in noisy environment might be of additional interest in a noisy speech transformation.

Exemplar-based voice conversion echoes the methods based on unit selection. Unit selection considers two costs: a target cost that measures the distance between converted vectors and a concatenation cost, representing the distortion after joining the sequence. It could synthesize converted speech with a more natural tone. The challenging job of choosing the optimal selection sequence is performed with dynamic programming, such as Viterbi decoding [134].

Exemplar-based methods are known for the high quality of reconstructed speech [19]. This property makes it attractive for indirect speech enhancement as speech quality is an important metric in assessing a speech enhancement system. We will explore the use of exemplars to perform speech transformation in this thesis.

2.3.3 Multi-layer perceptrons

The use of MLP in speech enhancement has been surveyed in subsection 2.1.2. The same tool has also seen its application in voice conversion. Both GMM and MLP can model

nonlinear transformations. A GMM models it with a weighted combination of class-based linear transformations, where the weights are the posterior probabilities. An MLP uses nonlinear activation functions in hidden layers to realize nonlinear mapping. Its first use in voice conversion was in [135] by only transforming the formants. More follow-ups involving MLPs include the work in [136, 137, 138].

MLPs and similar deep neural networks have seen growing interest in the community of speech enhancement in general. As discussed in section 2.1, it has demonstrated to be very effective in handling some noise that used to be difficult for traditional methods. The application of MLPs in speech transformation is very similar to their application in speech enhancement, as they can both be framed as mapping of spectral features. Hence, we can still employ MLPs in feature transformation in indirect speech enhancement.

2.3.4 Generative models

Recently, generative models, including variational auto-encoder (VAE) [139, 140] and GAN [141], have also been applied to speech conversion. Authors of [142] implemented non-parallel voice conversion with a VAE. Its benefit included unaligned corpora, which were usually more cost-effective to gather than parallel corpora. The encoder network learned speaker-independent phonetic representations, and the decoder learned to reconstruct speech from the target speaker. It relied on the assumption that a VAE could decouple the speaker and phonetic representations. It also assumed that the decoder could blend the two representations to synthesize a new frame. In order to better understand how a VAE can perform voice conversion, Hsu *et al.* explored feature disentanglement in [143]. They observed that vector arithmetic in latent spaces allowed speech attributes, such as a speaker or tonal information, to be manipulated. The modified latent representation could then be transformed into converted speech. This disentanglement property is relevant to speech enhancement and transformation as we only wish to transform the background noise while maintaining phonetic content unchanged.

Another class of generative models is based on GAN. The original GAN paper [141] let two neural networks, known as the generator and the discriminator, play a zero-sum game. The discriminator was trained to differentiate the generator’s outputs and ground truths, whereas the generator was trained to fool the discriminator by generating outputs similar to the ground truths. Cycle-GAN [144] built on this idea to use two pairs of generators and discriminators to transform features from a source domain to a target domain back and forth. The authors argued that an additional *cycle consistent loss* enforced the generated features to stay in a low dimensional manifold shared by desirable targets. It was first applied in voice conversion in [145, 146] to learn the forward and inverse mapping from a source to a target speaker. They found converted feature sequences to be near natural in terms of global variances and modulation spectra. A subjective evaluation showed that the converted speech quality was comparable to traditional methods that required parallel data. Star-GAN [147] was an extension of Cycle-GAN that enabled one-to-many mapping among a group of speakers. Even though the applications of the aforementioned generative models are relatively broad, we will be able to exploit techniques including feature disentanglement and adversarial loss in indirect speech enhancement, which will be introduced in the next section.

2.4 A high-level description of the proposed progressive enhancement approach with intermediate noisy speech target

When speech enhancement is too difficult due to low SNR or challenging noise, it is often not easy to obtain good enhancement results in a single step. The unsatisfactory quality is due to the highly non-linear relationship in high dimensional speech features. Prior works [14, 15] described in subsection 2.1.3 decomposed the problem of overcoming a large SNR gap with direct enhancement into a series of tasks with smaller SNR gaps. The assumption was that the smaller tasks were more manageable to learn for neural networks. When combined, the benefit of reduced difficulty outweighed distortions generated in each

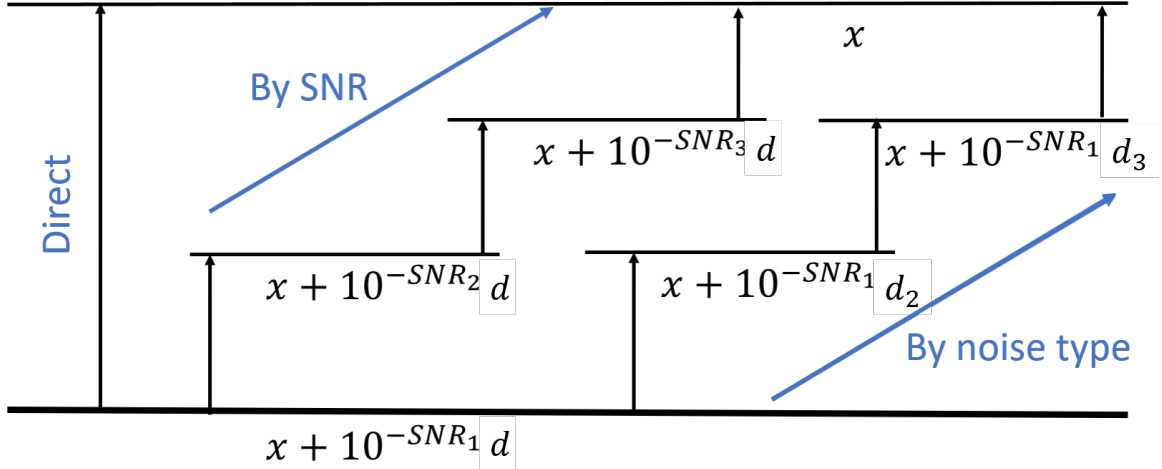


Figure 2.6: Progressive learning in speech enhancement

sub-task.

We could visualize the incremental improvement in SNR along the path in the center in Figure 2.6. The raw signal is denoted as the summation of the clean signal, x , and noise, n , scaled by an SNR factor. Instead of mapping from the noisy speech, $x + 10^{-SNR_1} n$, to the clean speech, x , directly, the first sub-task only learns to improve the signal to SNR_2 , where SNR_2 is higher than SNR_1 . The first stage's outputs are used as inputs to the next stage, again to learn a signal at even better SNR at SNR_3 . Eventually, the clean signal, x , can be recovered. Each stage's results can boost learning in subsequent stages since they receive pre-enhanced features at higher SNR.

When the authors of [14, 15] improves the speech SNR incrementally, the global average SNR in the signal improves. In other words, the improvement is better across all time samples and all frequency ranges after each sub-task. We can formulate progressive learning on a more local scale. Consider the example of the following contaminated speech signal in Figure 2.7. The noisy speech in the middle panel and the bottom panel both have the same average SNR, but it is evident that the bottom signal will be a simpler task for an enhancement system since it only needs to trim the noise dominated segment. The signal in the middle requires speech enhancement to separate speech from the overlapping noise, which is a harder task.

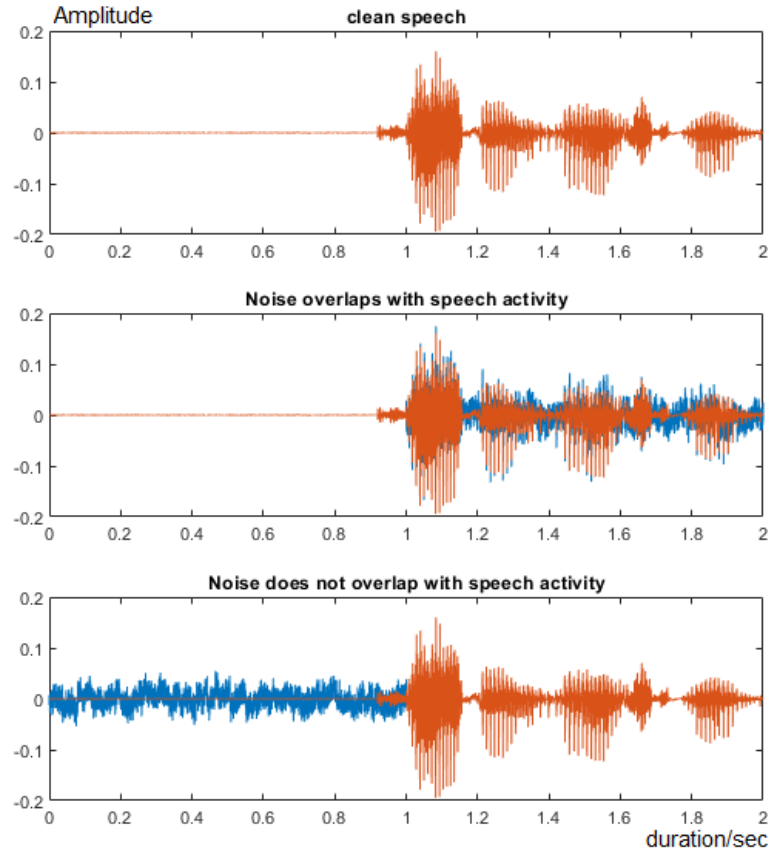


Figure 2.7: Relocation of noise in the time domain

A similar example can be illustrated in the frequency domain. In Figure 2.8, the signal on the left is contaminated by *white* noise at 0dB. The noise on the right is band-limited, but the overall SNR is kept at 0dB, too. A speech enhancement for the left signal must learn to find a good regressive mapping for all frequency bins. The signal on the right only requires the system to concentrate on making predictions in the corrupted bands. One could argue that even for speech signals at the same SNR, the tasks' difficulties are unequal depending on the relationship between speech and noise. While some noises are harder to handle, as in Figure 2.7 center and Figure 2.8 left, other noises are simpler, as in Figure 2.7 bottom and Figure 2.8 right.

These examples motivate us to design sub-tasks in progressive learning along a different

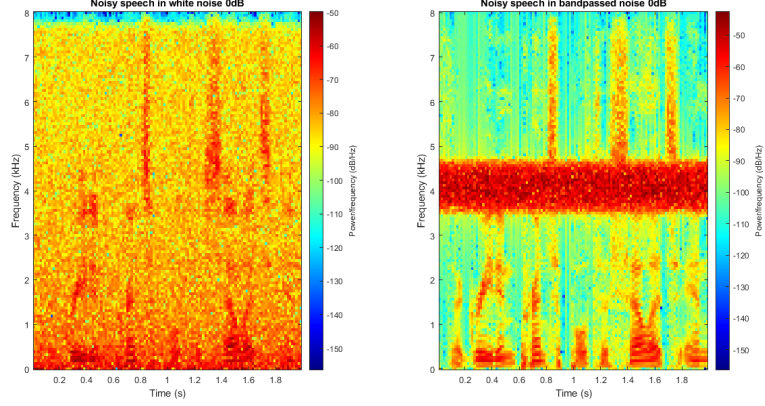


Figure 2.8: Relocation of noise in the frequency domain

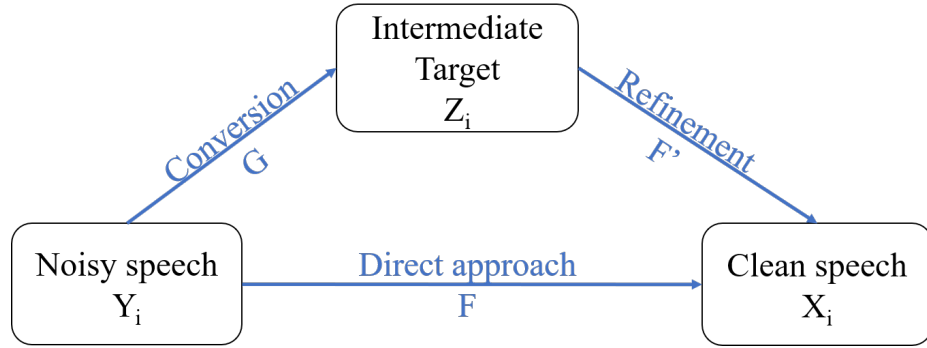


Figure 2.9: Framework of indirect speech enhancement

route. Specifically, we can design each sub-task to modify background noise into a simpler type. Schematically, along the right path in Figure 2.6, the original noise, n , in the noisy input speech, $x + 10^{-SNR_1}n$, is replaced by a simpler noise, n_2 , in the first sub-task. Its output is fed into subsequent tasks to obtain speech in even simpler conditions. Eventually, clean speech, x , is recovered after multiple intermediate stages. We refer to this flow as *indirect* speech enhancement via *conversion* to *intermediate targets*. The system is *indirect* as opposed to direct mapping using a black-box DNN for speech feature mapping. For each sub-task, the outputs are *intermediate* because subsequent tasks use them for enhancement. To accomplish each sub-task, we need to *convert* speech in one noise background into a simpler noise.

To better formulate indirect speech enhancement, we simplify the progressive path via

noise type conversion to Figure 2.9. We use X_i and Y_i to denote instances of clean and noisy speech features in the figure. The bottom path represents a direct mapping between noisy and clean speech, which is a DNN-based speech enhancement network, F . When trained with the MSE loss criterion, the task of direct enhancement is formulated as

$$\hat{F} = \arg \min_F \sum_i \|X_i - F(Y_i)\|_2^2. \quad (2.27)$$

The optimal parameter set, \hat{F} , is found by iteration updates with back-propagation.

The indirect method comprises at least a conversion step and a refinement step, shown with the upper path in Figure 2.9. The conversion step is accomplished by a converter, G . With an appropriate intermediate target, Z_i , the conversion step and the enhancement step seek to find \hat{G} and \hat{Z}_i such that

$$\begin{aligned} \hat{G} &= \arg \min_G \sum_i \|Z_i - G(Y_i)\|_2^2, \\ \hat{Z}_i &= \arg \min_{Z_i} \sum_i \|X_i - F'(Z_i)\|_2^2, \end{aligned} \quad (2.28)$$

assuming MSE is still chosen as the loss criterion. Here, the refinement network, F' , could be the same network as the direct enhancement, F , or it could be adapted to specific noisy speech, Z_i . The indirect approach jointly finds G , Z_i , and F' such that the combined loss $\sum_i \|Z_i - G(Y_i)\|_2^2 + \lambda \|X_i - F'(G(Y_i))\|_2^2$ is minimized. λ is a weight coefficient that reflects the ratio of errors from each step.

Progressive learning requires us to identify simple and difficult noise conditions. This step can either be done based on the noise signal's characteristics or some prior enhancement results. A detailed discussion on noise types will be discussed in Chapter 3. To illustrate the existence of intermediate targets, we use an example in Figure 2.10. The noisy input and its output from a pre-trained speech enhancement system are on the left. There is considerable residual noise in high-frequency bands. During the neural network

training, one uses MSE between the predicted output and the clean label as loss to back-propagate and update the network parameters. Instead of updating the network parameters, we can use the error gradients to modify the inputs to minimize the loss. In particular, we find an “optimized input,” \hat{Y} , such that

$$\hat{Y} = \arg \min_Y \|X_0 - F(Y)\|_2^2, \quad (2.29)$$

where Y_0 is the original input, and X_0 is the ground truth. Just as in regular network training, the error surface may not be convex, but one can still use gradient methods to iteratively find \hat{Y} such that $\|X_0 - F(\hat{Y})\|_2^2 < \|X_0 - F(Y_0)\|_2^2$. One such \hat{Y} after updating 20 iterations of updates is shown on the top right corner in Figure 2.10. Even though it is far from being a clean signal, its enhanced result, $F(\hat{Y})$, shown on the bottom right, is much better than the original output, $F(Y_0)$. Thus, the signal, \hat{Y} , could be a good example of an intermediate target for this input. The previous example suggests how to derive the intermediate targets, but it implies that intermediate targets exist, at least for a fixed neural network model. We will discuss how to obtain intermediate targets in Chapter 3.

We also need to ensure that there are practical ways of obtaining suitable intermediate targets. Furthermore, there exist good speech conversion techniques for noise type conversion. These targets are essential in supervised training for the neural networks in each sub-task. In offline training, we could synthesize these training targets using clean speech and the chosen noise type, just like how we simulated paired data for direct training [9]. Since the training is offline, we could create a large amount of paired data this way. Chapter 4 and Chapter 5 will further explore data simulation procedures. In Chapter 4, we will explore a more traditional simulation of paired training data, whereas Chapter 5 discusses speech transformation with unsupervised learning.

The design of the sub-tasks, i.e., the conversion steps, are essential too. In Chapter 4, we will explore the use of direct mapping between difficult and simple speech. These direct

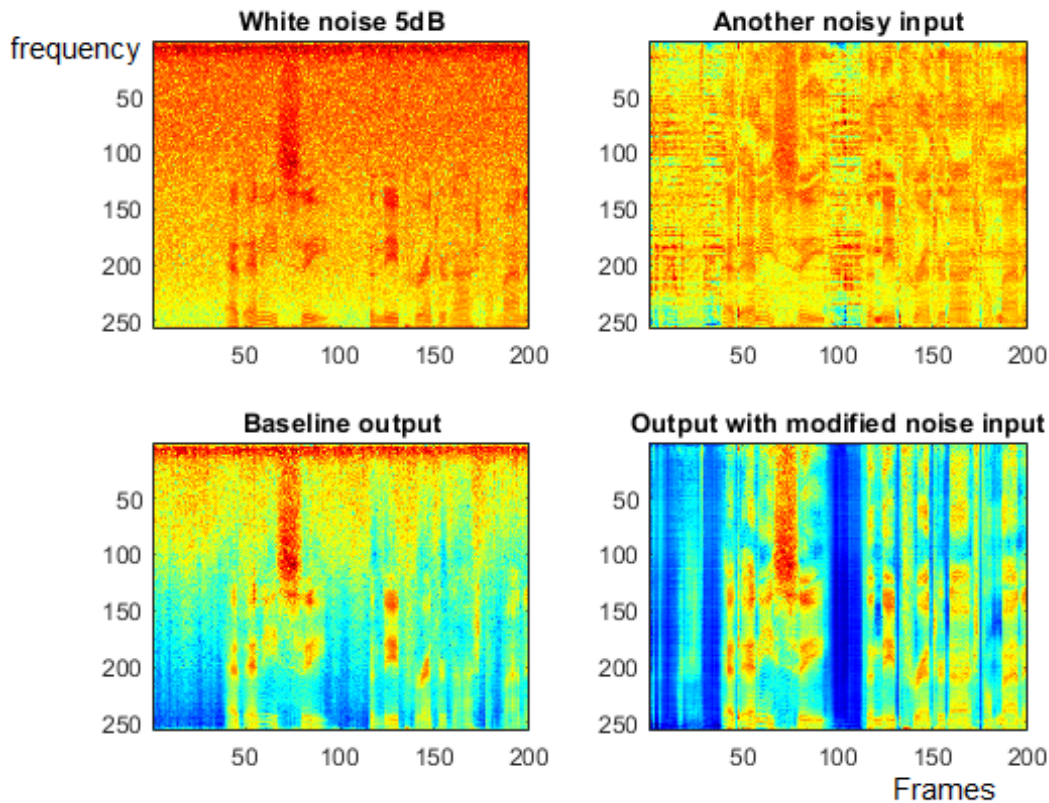


Figure 2.10: The existence of intermediate targets

mapping methods can be boosted by additional techniques such as multi-task training and noise-aware training. The conversion module is also concatenated with the refinement module for joint training. In Chapter 5, we consider speech transformation based on latent space methods. We consider the acoustic space made up of speech bases and noise bases. Noisy speech conversion can then be formulated as a change of basis in the acoustic space or the latent space. We could also leverage upon representation learning to find a structured representation of speech and noise. Such latent structures enable us to replace the difficult noise background with simpler ones. Lastly, the converters are integrated with refiners to create a complete indirect speech enhancement system.

CHAPTER 3

CHARACTERIZATION OF ADDITIVE NOISES

3.1 Introduction

3.1.1 Noise in speech enhancement

Noise is known to affect speech perception in human communication. Different types of noise have different impacts on the quality and intelligibility of speech. In [148], the authors noted that speech-shaped noise, such as *babble*, could mask out speech. This types of noise made speech less intelligible. The effects of non-stationary noise were discussed in [149, 150]. Noise can also degrade an ASR system’s performance, as it results in feature mismatch [151] or model mismatch [152]. The [153], the authors highlighted how *babble* and speech-shaped noise could obscure the F2 formant in vowel sounds. Intelligibility was also heavily compromised with additive noise, and many conventional speech enhancement methods have failed to improve it [51]. Moreover, most of the previous studies considered a handful of noise types, such as *babble* and *white* noise, or do not address the performance gap of ASR or SE systems due to different noise backgrounds. In the rest of the chapter, we will demonstrate large performance gaps among various types of noise. We will next discuss noise characteristics and how they affect feature pre-processing. Lastly, we conduct empirical studies to validate our proposed categorization of simple and difficult noise.

3.1.2 Enhancement quality and improvement depending on noise types

We use PESQ introduced in Chapter 2 to evaluate the quality before and after a DNN speech enhancement system. Based on the enhancement results, we demonstrate that differences in quality exist for different noise types. The differences include the final PESQ scores after enhancement and the extent of improvement. In other words, speech in different

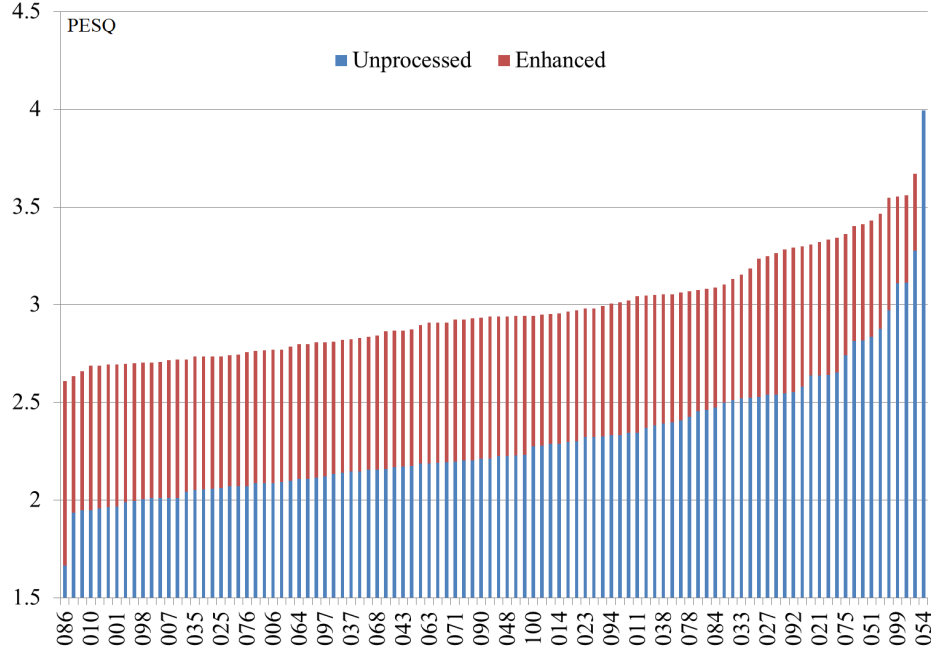


Figure 3.1: PESQ of 100 types of Nonspeech noise

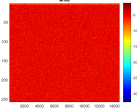
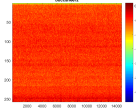
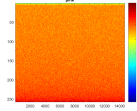
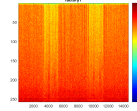
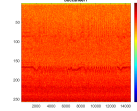
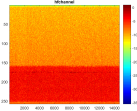
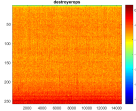
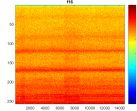
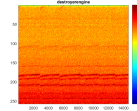
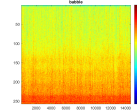
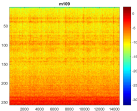
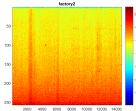
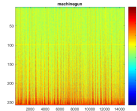
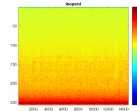
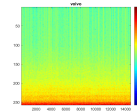
backgrounds reaches different qualities after enhancement. Moreover, the improvements in PESQ score from unprocessed speech to enhanced speech are also not the same across different noise types.

An enhancement system based on DNN is trained following the procedure in [10]. One hundred types of noise from the Nonspeech corpus [154] are used in training. The same 100 types of noise and 15 additional types from Noisex92¹ [103] are used to evaluate the DNN’s performance with the PESQ score. PESQ is a standard that automates the assessment of speech quality. It mimics a mean opinion score as if a human listener rates the enhanced speech. Its score range is from -0.5 to 4.5. Higher scores correlate to better perceived quality. More details on PESQ can be found in [32].

From Figure 3.1 and Table 3.1, the difference in enhancement quality is noticeable. Figure 3.1 shows the enhancement results on the 100 types of noise used in the training set. The greatest improvement is over 0.95 for noise *n086*, indicated by the longest red bar. On the other hand, noise *n054* is adversely affected by the enhancement network, showing a

¹Details of Nonspeech and Noisex92 noise can be found in Appendix D and E respectively.

Table 3.1: PESQ on Noisex92 noise types

Types	white	buccan2	pink	factory1	buccan1
Before	2.10	2.16	2.15	2.17	2.01
After	2.09	2.19	2.22	2.30	2.37
Spectrogram					
Types	hfchan	desops	f16	desengine	babble
Before	1.90	2.33	2.15	2.16	2.22
After	2.42	2.44	2.45	2.46	2.52
Spectrogram					
Types	m109	factory2	machinegun	leopard	volvo
Before	2.57	2.57	2.79	2.71	3.65
After	2.76	2.93	2.95	3.01	3.57
Spectrogram					

drop of 0.25 score. The noise conditions also do not achieve the same level of quality after enhancement. The best noise type, *n054*, achieves a PESQ score of 3.7, whereas the PESQ for the worst noise, *n086*, is only 2.6. There is a gap over 1.0 in perceptual quality. Similar differences exist for unseen noise in Table 3.1. Noise, such as *hfchan*, witnesses the greatest improvement (greater than 0.5), but noise, such as *white* and *volvo*, barely improves.

3.2 On the criteria to select intermediate targets

In section 2.4, the intermediate targets are chosen such that the distortion of either the conversion or the refinement stage is relatively small. When the intermediate target is selected to be close to the source noise type, the conversion is relatively easy, but the refinement needs to handle a difficult task. On the other hand, a simpler noise type as the intermediate target will ease the refinement task. In the rest of the section, we argue that a

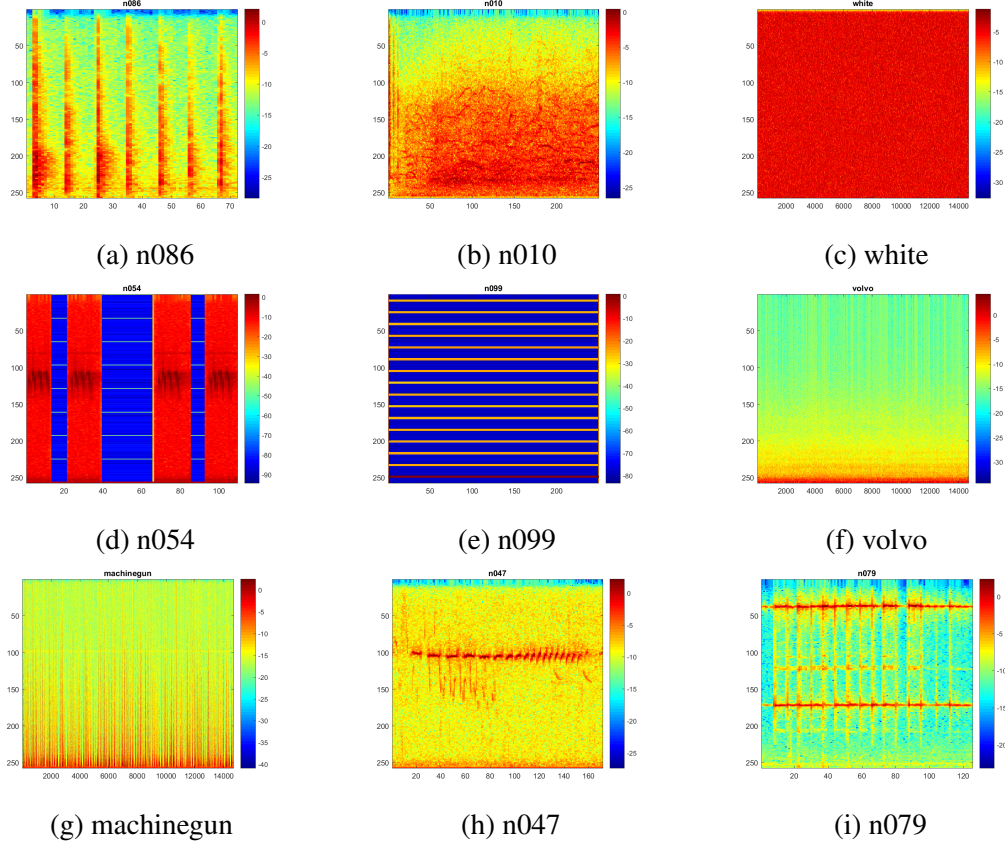


Figure 3.2: Spectrograms of samples noises. Top row contains difficult noise types. Middle row contains simple noise types. Bottom row shows the outliers

simpler noise type should be chosen instead of a hard noise type as an intermediate target.

Visual inspection of the spectrograms shown in Figure 3.2, partially re-affirms our prior understanding of the nature of noise. We expect steady and narrowband noise to be easier to be handled than non-stationary and wideband noises. Base on the PESQ score, we could infer that noise, including *n086*, *n010*, and *white*, are all relatively difficult. Their spectrograms show varying temporal characteristics and widebands in the signals. Simple noise types, such as *n054* and *volvo*, on the other hand, have stable temporal variations and are generally band-limited. Such observations are in agreement with past literature [8, 155]. However, we also notice that there are a few outliers. Non-stationary noise, such as *machinegun*, *n047*, and *n079*, are unexpectedly good both before and after enhancement. Despite their non-stationary temporal characteristics, these noise types have long-term av-

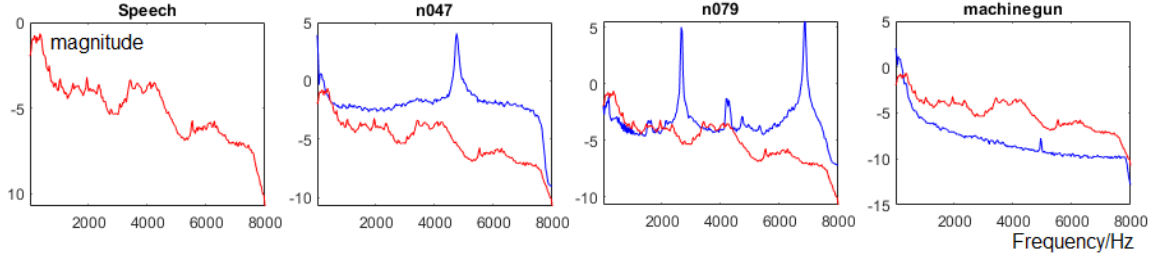


Figure 3.3: Spectral shape of clean speech and some easy noise. Long-term average speech spectrum shown in red.

erage spectra that only obscure part of the speech spectra, as shown in Figure 3.3. As a result, when input features of the speech enhancement network are short-time spectral features, only a small fraction of the frequency bins are dominated by noise, effectively simplifying the enhancement task.

Such a proposition is primarily based on our prior knowledge about noise-robustness from experience. We examine similarities and differences among noise types by leveraging upon clustering. Each waveform in our noise corpus is segmented into 5-second chunks. Each segment will be one observation sample. Welch’s method [156] is then used to compute the PSD estimate of the sample since averaged periodograms represent approximately uncorrelated estimates of the true PSD with reduced variability. Figure 3.4 shows the t-distributed stochastic neighbor embedding (t-SNE) visualization [157] of k-means with three clusters colored with red, green, and blue. Comparing the clustering result with the PESQ scores in Table 3.1, we could tell that the green cluster comprises the “best-performing” noise types, including *volvo*, *machinegun*, and *leopard*. The red cluster contains challenging noise types. The blue cluster represents noise with medium difficulty. The clustering result shows that simple noise shares some common traits as discovered by the k-means algorithm. Such traits may include their band-limitedness, stationarity, and other characteristics, such as the overall spectral shapes. We will verify each attribute’s effects by converting noisy speech to intermediate targets with or without these attributes.

We adopt an empirical approach to determine what characteristics make the noise simple and suitable as an intermediate target. The enhanced system is implemented with a

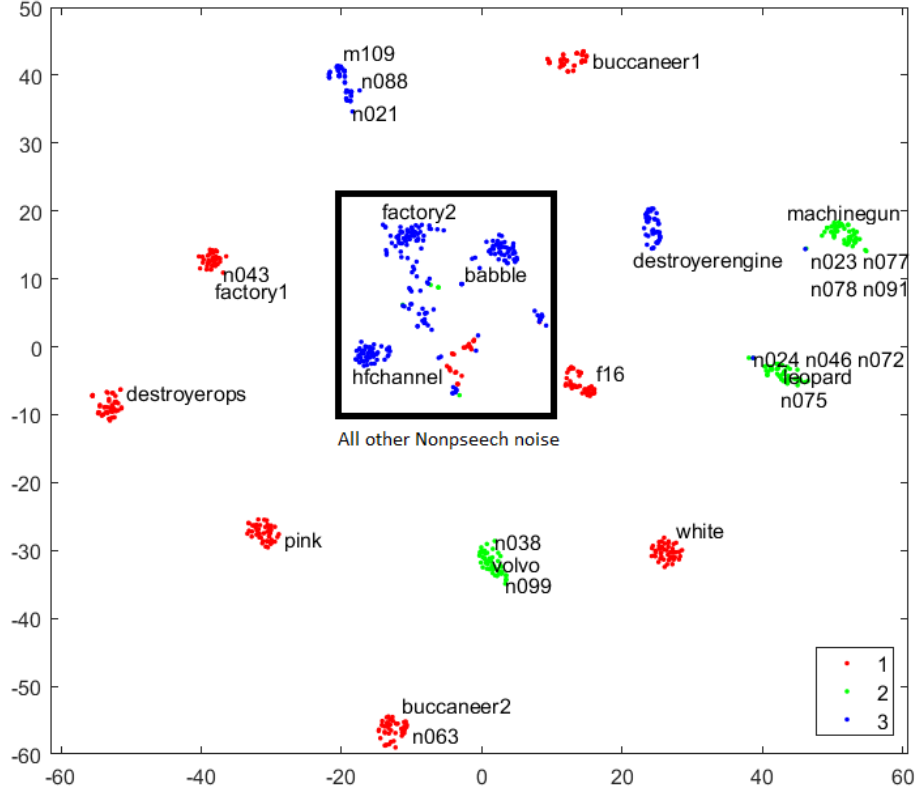


Figure 3.4: k-means clustering with t-SNE projection on 115 noise types

DNN in [9], which remains fixed for this study. It consists of 3 hidden layers of 2048 nodes with sigmoid activation. The input features are 11 consecutive frames of LPS features. The window length of framing is 512 at a frame rate of 256. We train the enhancement network to map noisy LPS features into a single frame of clean LPS features. Both the inputs and the targets are normalized by global mean and standard deviation per feature dimension. The noisy utterances are synthesized using *train_si84* part of the Wall Street Journal (WSJ) corpus [158] mixed with the Nonspeech noise corpus [154]. The total size of the training speech is 32 hours. The 333 testing utterances are from the *test_eval92* directory of the WSJ corpus. In testing, both Nonspeech and Noisex92 noises are used to create the matched and mismatched test set.

Table 3.2: PESQ of conversion using simple noise as the intermediate targets

Intermediate Target	Quality	Nonspeech			Noisex92		
		Avg	1st quant	4th quant	Avg	1st quant	4th quant
direct	2.99	2.99	2.80	3.22	2.57	2.26	3.10
n079	3.55	3.01	2.86	3.18	2.57	2.22	3.06
n099	3.55	3.00	2.84	3.17	2.52	2.18	3.05
volvo	3.57	3.05	2.90	3.21	2.58	2.28	3.09
n047	3.67	2.84	2.88	3.14	2.60	2.28	3.07
n054	3.75	2.89	2.76	3.06	2.50	2.20	2.98

3.2.1 Simple noise with high PESQ score

Since *n054* and *volvo* have high PESQ scores, they can be considered simple noise, and good candidates as intermediate targets. To confirm if the PESQ scores are a good way to select intermediate targets, we design experiments to perform indirect enhancement with speech in noise types with high PESQ scores as intermediate targets. In order to simulate the intermediate target speech, we add the target noise to clean speech to synthesize noisy speech in a desirable background. The conversion of noisy speech from the original noise background to the intermediate target is done with a DNN that maps the LPS features. A general enhancement system will then refine the converted speech. The DNN is trained to minimize MSE loss between original and converted speech.

Table 3.2 compares the PESQ of two test sets using the direct and indirect approaches with a list of intermediate targets considered as simple noise based on enhancement quality. When noisy speech in a noise type receives a PESQ score higher than 3.5, it is considered simple and used as an intermediate target. The original PESQ score is labeled as “Quality” in column 2 in Table 3.2. In addition to the overall average PESQ of each test set, two subsets are created in each test to evaluate the performance of best and worst noise conditions. They are labeled as the first and fourth quantiles in Table 3.2. Such grouping helps analyze the effects of conversion on both simple and challenging noise conditions. The first quantile consists of about a quarter of noise types with the worst performance, whereas

Table 3.3: Comparison of MSE of direct and indirect methods on the 1st quantile of noisy speech

Noise	Nonspeech				Noisex92			
	PESQ	Direct	Convert	Refine	PESQ	Direct	Convert	Refine
direct	2.80	380	-	-	2.26	550	-	-
n079	2.86		100	340	2.22		110	340
n099	2.84		190	10	2.18		200	10
volvo	2.90		140	90	2.28		140	90
n047	2.88		50	410	2.28		80	410
n054	2.76		80	170	2.20		150	170

the fourth quantile contains noise types with the best performance. One could argue that the first quantile is the difficult noise conditions, and the fourth quantile represents easy conditions.

We could draw several conclusions from Table 3.2. First, a good performance on the baseline DNN does not guarantee that a noise type can be used as an effective target for conversion. In the case of noise type *n054* and *n047*, despite their relatively high quality of 3.75 and 3.67 when evaluated with DNN, using them as intermediate targets does not improve the overall performance, indicated by the bottom two rows in Table 3.2. Second, even though the overall improvement from 2.99 to 3.05 is small in *volvo* noise, the benefits are more pronounced for noise types within the first quantile, where the improvement is from 2.80 to 2.90 when *volvo* is the intermediate target. A similar trend could be observed for other noise types. Conversion is generally not useful for simple noisy types in the fourth quantile, as the PESQ score all drops after conversion, and is more useful for difficult conditions in the first quantile. Third, the indirect approach with DNN-based conversion does not address the domain mismatch problem. Comparing the results in the Nonspeech test set and that in the Noisex92 test set, one could tell that the improvement is much smaller. This difference might be because the converter is also implemented with a similar DNN trained on the same features. Consequently, it also suffers a similar domain mismatch problem.

To explain why the indirect approach works for challenging noise, we can compute the

Table 3.4: Effects of spectral shapes on the suitability as intermediate targets

Noise	Nonspeech			Noisex92		
	Avg	1st quant	4th quant	Avg	1st quant	4th quant
direct	2.99	2.80	3.22	2.57	2.26	3.10
volvo	3.05	2.90	3.21	2.58	2.28	3.09
ovlov	2.82	2.69	2.98	2.44	2.20	2.87

distance between noisy features and clean targets. Likewise, we could compute the distance between conversion targets with respect to noisy and clean speech. MSE can be used as a distance metric. It is a rough measure of the difficulty of neural network-based mapping. In Table 3.3, we could see that the sum of the MSE of conversion and refinement stage is generally lower than the MSE of direct enhancement. For example, in the case of *volvo* noise on the Nonspeech test set, the average MSE is 380 between the original noisy speech and clean targets. However, converting to *volvo* only needs to overcome an average MSE of 140, followed by another 90 in the refinement. The sum of the two stages is lower than the MSE of direct enhancement. This smaller MSE could help explain the benefits of indirect enhancement.

3.2.2 Noise shape

Instead of selecting intermediate targets solely based on their performance on the baseline DNN system, one could choose targets based on some signal attributes. We invert the spectrum of *volvo* by modulating the signal by $e^{j\pi n}$ to create an artificial noise, named *ovlov* so that the two share the same bandwidth, stationarity, and instantaneous energy. The long term average spectra of clean speech, *volvo*, and *ovlov* noise are shown in Figure 3.5.

The experimental result of converting noisy speech to both *volvo* and *ovlov* is listed in Table 3.4. It is evident that *ovlov* is less an ideal target than *volvo* in terms of the quality after conversion, as its PESQ score of 2.82 and 2.44 on the Nonspeech and Noisex92 test set are lower than that of *volvo* noise. As Figure 3.5 suggests, the spectrum of *volvo* lies beneath that of speech in most frequency bins. This energy distribution reduces the

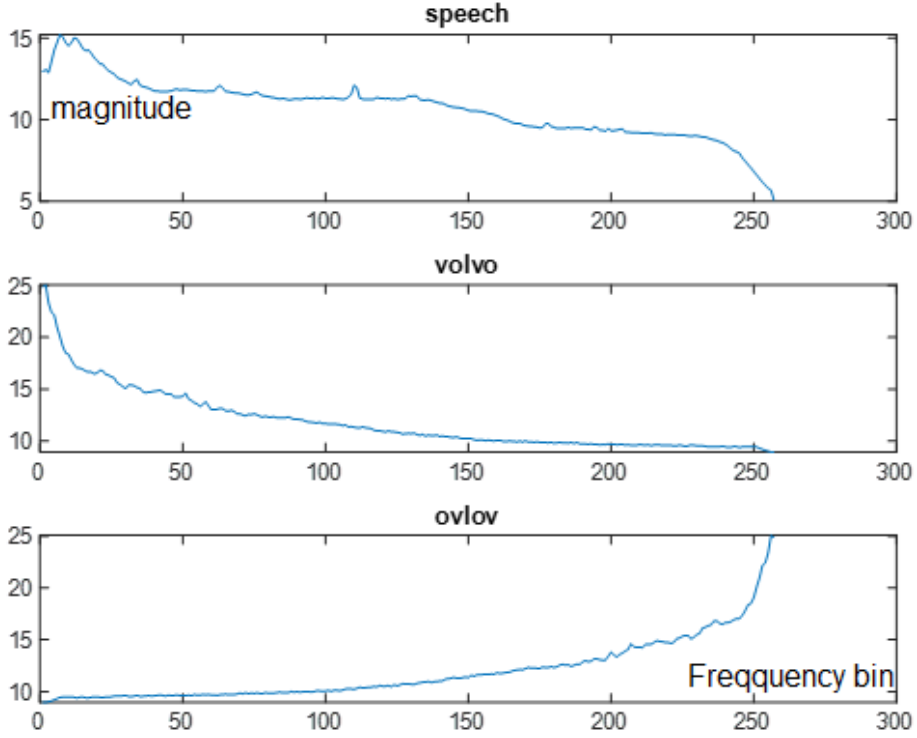


Figure 3.5: Long term average spectra of speech, *volvo*, and *ovlov*

Table 3.5: MSE comparison between *volvo* and *ovlov* noise

	Direct	Conversion	Refinement	Indirect
volvo	3300	2700	490	3200
ovlov	3300	4100	1300	5400

spectral mismatch between the target and clean speech. In terms of MSE loss, it is clear that *ovlov* noise is both harder to convert to and harder to refine due to its contrasting spectral shape from speech. In other words, a good intermediate target should exhibit spectral shapes that generally lie beneath the speech spectrum. This idea echoes the psycho-acoustic weighting used in conventional speech enhancement and speech coding [49, 117], where more emphasis is placed on spectral valleys to minimize distortions in these regions. In this case, the target noise should have low energy at frequencies where the speech is weak. Otherwise, the dominant noise in weak speech regions will make the subsequent refinement task difficult.

Table 3.6: Effects of bandwidth of conversion targets

Passbands /Hz	Nonspeech			Noisex92		
	Avg	1st quant	4th quant	Avg	1st quant	4th quant
-	2.99	2.80	3.22	2.57	2.26	3.10
0-50	3.04	2.89	3.20	2.58	2.27	3.09
0-100	3.02	2.88	3.17	2.62	2.38	3.09
0-200	2.91	2.79	3.03	2.56	2.33	2.99
0-500	2.80	2.71	2.90	2.49	2.27	2.86
100-200	2.96	2.81	3.10	2.58	2.34	3.01
200-400	2.80	2.70	2.92	2.48	2.26	2.86

3.2.3 Bandwidth

To study the effects of the bandwidth of intermediate noise type, we filter *white* noise with low-pass or band-pass filters with specific passbands to create artificial noise with desirable bandwidths. We also shift the narrowband noise's peak to investigate the effects on the location of the spectral peaks of various noise. In the first series of experiments, the passbands gradually increase from 50 Hz up to 500 Hz. The trend in Table 3.6 indicates that the performance degrades steadily above 100 Hz. This result confirms our early observation that wideband noise tends to be difficult for enhancement, and intermediate targets should be band-limited. In the second experiment, the peak of the noise spectrum is shifted. By comparing the rows of (0-100) vs. (100-200) and (0-200) vs. (200-400), we could conclude that the noise occupying lower frequency bands are more suitable for conversion targets. Such a result could be explained by the speech spectrum with higher energy in lower frequency bands, hence easier to mask the noise.

3.2.4 Stationarity

The last signal attribute to examine is the stationarity of the conversion targets. To maintain a fair comparison with another noise with similar bandwidth and spectral shape, we introduce non-stationarity to the *volvo* noise by creating interleaving patterns in its temporal envelop, as shown in the second and third spectrograms in Figure 3.6. The performance of

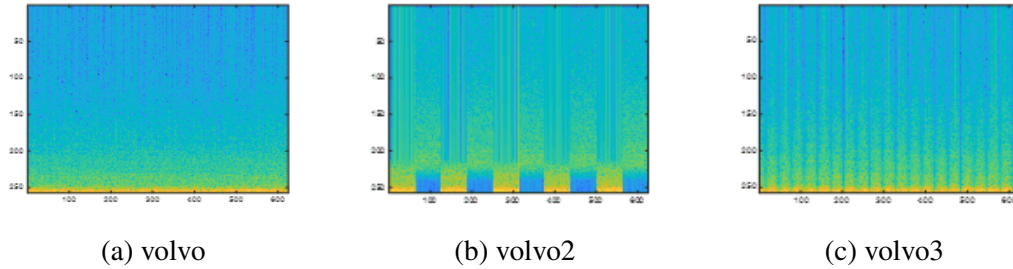


Figure 3.6: Non-stationary examples of *volvo*-like noise

Table 3.7: Effects of stationarity of conversion targets

Noise	Nonspeech			Noisex92		
	Avg	1st quant	4th quant	Avg	1st quant	4th quant
direct	2.99	2.80	3.22	2.57	2.26	3.10
volvo	3.05	2.90	3.21	2.58	2.28	3.09
volvo2	3.07	2.92	3.24	2.59	2.27	3.10
volvo3	3.05	2.90	3.21	2.58	2.29	3.08

using each variation of *volvo* noise is given in Table 3.7. Contrary to our prior assumption that non-stationary noise is typically harder than stationary ones, the results in Table 3.7 do not reveal a significant difference among the three noise types for either test set. This result agrees with the earlier observation of non-stationary noise types with good performance, such as *machinegun*, *n047*, and *n079* in Figure 3.2. It also suggests that stationarity is a less crucial factor when evaluating intermediate targets.

We can conclude through a series of experiments that a good performance on the baseline DNN is insufficient to guarantee that a noise type is suitable as an intermediate target. Simple noise should ideally have a long-term average spectrum that lies below clean speech spectrum. A band-limited signal that resides in low-frequency regions tends to have better performance over wide-band signals or signals occupying higher frequency bands. Its stationarity is less relevant as a conversion target.

3.3 Summary

In this chapter, we discuss the difficulty of processing speech input when background noise is present. Noise generally degrades the quality of perceived speech and hinders our ability to comprehend its content. The same issue exists for speech enhancement or ASR systems. We highlight the disparity in difficulties of enhancing speech in different background noise. The differences can be observed in terms of both the gain in the PESQ scores, as well as the final quality. We then attempt to cluster background noise into simple and difficult groups based on their spectral and temporal characteristics. Out of the many signals characteristics, we find the average spectrum shape and bandwidth most relevant when evaluating its difficulty in enhancement applications. Both of these factors are satisfied if the long-term average speech spectrum can effectively mask the noise. Such knowledge would aid us in finding suitable intermediate targets for indirect speech enhancement.

CHAPTER 4

INDIRECT SPEECH ENHANCEMENT WITH SUPERVISED LEARNING

4.1 Introduction

In Chapter 3, we have identified noise characteristics that determine if the speech in that noise can be easily enhanced. Such a simple noise could be a suitable intermediate target in our indirect approach to SE. Once we have selected an intermediate target, we need to design other components in the progressive enhancement framework. Specifically, we define a conversion step, during which original noisy speech is converted to speech in less difficult noise. It is followed by a refinement step responsible for recovering final clean speech using the intermediate speech as an input. This process is depicted in the upper path of Figure 4.1. In comparison, the direct approach is shown in the lower path.

In this chapter, we will first outline a speech conversion technique by matching statistics of speech features. This step is inspired by the observation in Chapter 3 that improper feature normalization leads to degrading enhancement performance. The next approach is based on a frame-level mapping. The motivation behind the frame-level mapping is to leverage the universal approximation theory of neural networks [159] to perform feature transformation. Lastly, we extend the indirect approach to handle noisy conditions with

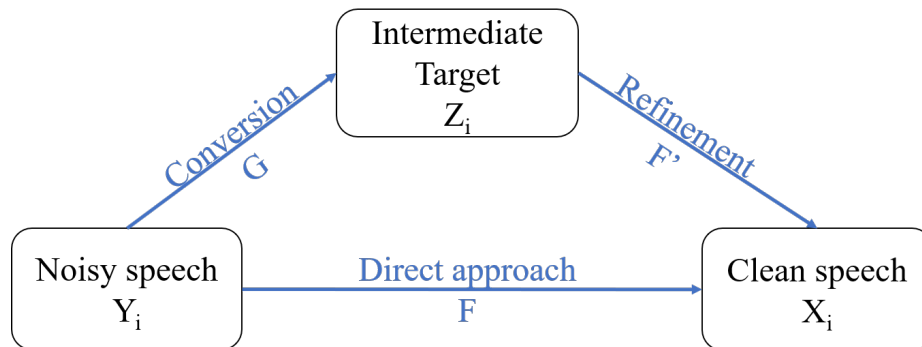


Figure 4.1: Framework of indirect speech enhancement

multiple noise sources. We show that the noise interferences can be removed sequentially to achieve indirect enhancement.

4.2 Matching feature statistics

This section first shows the effects of noise in feature normalization, which is crucial in DNN-based speech enhancement. An adverse condition arises when difficult noise at low SNR affects both mean and variance normalization. For the mean standardization, there will be a deviation from the global mean statistics. The deviation creates an offset from 0 in normalized features. The offset is greater at frequency bands with more dominant noise power. As a result, for speech in simple noise, mean normalization translates feature vectors to almost zero. However, the more difficult noisy speech will not be centered at 0 as its mean is far from the global mean. Similarly, for difficult noise, feature after variance normalization will also not have unit variance at bands with more noise power. Consequently, This results in a mismatch between input distribution in training and testing. To alleviate such deviation, we propose to apply a transformation to match the statistics of normalized features of a difficult noise to that of a simpler noise. To achieve this goal, we will investigate the use of mean-variance matching and histogram equalization algorithms.

4.2.1 Effects of noise in feature normalization in speech enhancement

Feature normalization, or feature standardization, refers to the practice of scaling input features to the same range so that they have similar magnitudes. Min-max scaling could be used if lower and upper bounds of the feature values are known [160]. The normalized feature will be constrained in the range of $[-1, 1]$ or $[0, 1]$. This is commonly seen in image processing where pixel values are finite after digitization, as in $[0, 255]$ for a 8-bit gray-scale image. LPS is a commonly used feature in SE [16]. It is defined as

$$X_{LPS} = \log (X(m, k)^* X(m, k)) = 2 \log |X(m, k)|, \quad (4.1)$$

where $X(m, k)$ is the short-time DFT of the m -th frame at frequency bin, k . One can see that such features are not bounded. This property makes min-max scaling difficult. Features like LPS can be normalized with z-score normalization instead [161]. Z-score normalization can be defined as the following linear transform

$$\bar{X}_{LPS} = \frac{X_{LPS} - \mu_{LPS}}{\sigma_{LPS}}, \quad (4.2)$$

where μ_{LPS} and σ_{LPS} are the mean and standard deviation of feature, X_{LPS} , accumulated over the frames in each dimension.

As a result of z-score normalization, each feature dimension's mean will be 0, and the standard deviation will be 1. Theoretically, this linear transform is not essential as the linear operation can be captured by the input layer in a DNN. However, practical reasons exist for the benefits of feature normalization. Optimizers such as stochastic gradient descent could converge faster. Without feature normalization, the error surface could become elongated, and a global learning rate will make learning in some dimensions very slow [159]. In speech enhancement, both input and target features are standardized to possess zero-mean and unit-variance [10]. An inverse linear transform of Equation 4.2 is performed after DNN prediction to reconstruct LPS in the original scale.

We wish to understand the effects of normalization on the LPS features depending on the background noise. The LPS features used in our enhancement experiments have 257 dimensions. We rely on dimension reduction techniques to find their projections for visualization. The two-dimensional projections of the normalized LPS features are obtained with PCA. Figure 4.2 shows the distribution of these projected speech vectors in a few noise types, including *white*, *pink*, and *volvo* noise after normalization. One could observe that the normalized noisy speech vectors in different noise have dissimilar distributions. Specifically, noisy speech vectors in *white* and *pink* noise drift away from the center. Their clusters are also more compressed compared to that of *volvo*.

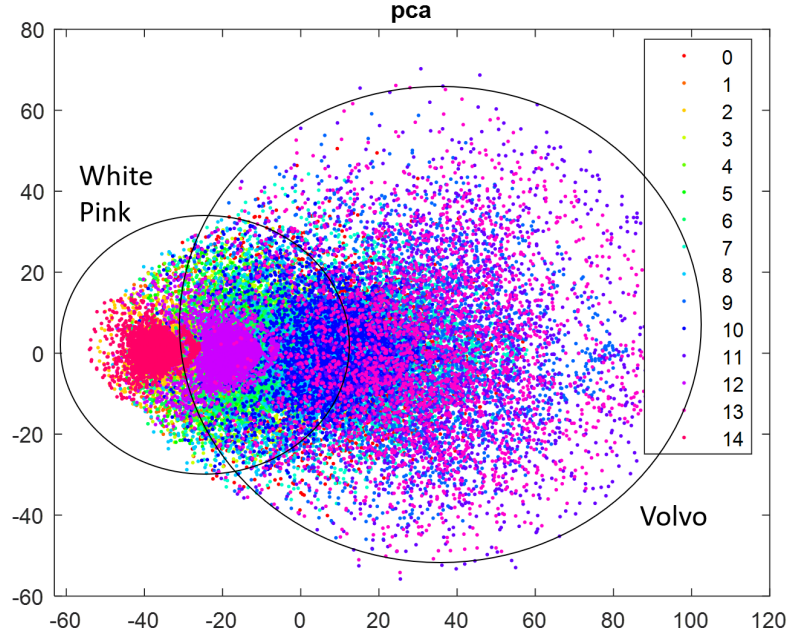


Figure 4.2: PCA projections of normalized features for different noises

Intuitively, clean speech can be considered a simple noise since it can be reconstructed with high fidelity. We understand that different speech sounds, including vowels and consonants, have very different spectra. Vowels possess formant peaks and mostly occupy lower frequency bands. Consonants, such as stops and fricatives, are noise-like and occupy high-frequency bands [162]. Thus, it is instinctive to expect clean speech features to spread apart because different vowels and consonant sounds have very different spectra. By comparing *volvo* and *white*, we can infer that *volvo* is a simpler noise type because it is more spread out as clean speech does. On the other hand, speech in *white* or *pink* noise always has a noise background. The noise background can be considered as a marker that makes noisy speech features alike. Hence, the projections speech features in *pink* or *white* noise are more tightly clustered. To better understand such a difference, we will discuss how different noise affects the normalization of the LPS feature in the next section.

4.2.2 Deviation of mean in normalizing speech in difficult noise types

To understand why speech in different noise backgrounds follows different distributions after the z-score normalization defined in Equation 4.2, we will show that the normalization of LPS features suffers from mean and variance deviation if the overall SNR level is low or the noise type is difficult.

Recall the additive noise model after short-time Fourier transforms. The complex spectrum of noisy speech at frame index, m , and frequency bin, k , is

$$Y(m, k) = X(m, k) + D(m, k) \quad (4.3)$$

$$= |X(m, k)|e^{j\angle X(m, k)} + |D(m, k)|e^{j\angle D(m, k)}. \quad (4.4)$$

$|X(m, k)|$ and $|D(m, k)|$ are the clean speech and noise magnitudes. $\angle X(m, k)$ and $\angle D(m, k)$ are their phases. Since z-score normalization is applied to each dimension independently, we will drop the frequency indicator, k , in the discussion below. We denote $|Y(m, k)|$ as Y_m . The same for X_m and D_m .

The instant power spectrum is computed as

$$Y_m^2 = Y_m^* Y_m \quad (4.5)$$

$$= (X_m e^{j\angle X(m)} + D_m e^{j\angle D(m)})(X_m e^{-j\angle X(m)} + D_m e^{-j\angle D(m)}) \quad (4.6)$$

$$= X_m^2 + D_m^2 + 2X_m D_m \cos(\angle X(m) - \angle D(m)). \quad (4.7)$$

Let $\phi_{XD} = \angle X(m) - \angle D(m)$ denote the difference of phase angle, and let $\xi_m = \frac{X_m}{D_m}$ be the instantaneous SNR. We can take the logarithm of both sides of Equation 4.5 to derive the noisy LPS,

$$\log Y_m^2 = \log (X_m^2 + D_m^2 + 2X_m D_m \cos(\angle X(m) - \angle D(m))) \quad (4.8)$$

$$= \log X_m^2 + W(\xi_m, \phi_{XD}), \quad (4.9)$$

where

$$W(\xi_m, \phi_{XD}) = \log \left(1 + \frac{D_m^2}{X_m^2} + \frac{2D_m \cos \phi_{XD}}{X_m} \right). \quad (4.10)$$

To perform feature normalization, we need to compute the global mean and variance of clean LPS features. The mean of the clean LPS in each dimension is defined as

$$\mu_{LPS} = \mathbb{E}[\log X_m^2]. \quad (4.11)$$

Similarly, the variance of the LPS in each dimension is defined as

$$\sigma_{LPS}^2 = \text{Var}(\log X_m^2) = \mathbb{E}[(\log X_m^2 - \mu_{LPS})^2]. \quad (4.12)$$

We want to show that the normalized feature will not be centered at 0 or have unit variance after normalization by Equation 4.2 for difficult noise types, which could help us understand the translation and compression of noisy speech features in *pink* and *white* noise in Figure 4.2.

If the normalized features are zero-centered, the expectation of the noisy LPS features must equal to the global mean. However, if they are not equal, there will be a deviation. We define the deviation in mean as

$$\Delta_\mu = \mathbb{E}[\log Y_m^2] - \mu_{LPS}. \quad (4.13)$$

By expanding $W(\xi_m, \phi_{XD})$ in Equation 4.10 by Taylor expansion and assuming ϕ_{XD} follows a uniform distribution in $-\pi$ to π , we can show that

$$\Delta_\mu \approx \begin{cases} \mathbb{E}\left[\frac{2 \cos \phi_{XD}}{\xi_m}\right], & \text{if } \xi_m \rightarrow \infty \\ -2\mathbb{E}[\log \xi_m]. & \text{if } \xi_m \rightarrow 0 \end{cases} \quad (4.14)$$

The derivation of Equation 4.14 can be found in Appendix A. We can interpret the

result in Equation 4.14 depending on ξ_m . When the signal is in high SNR, there are a lot of time-frequency (TF) bins with high ξ_m . Similarly, if noise is effectively masked by speech, i.e., it fits the criterion of a simple noise, most TF bins have high ξ_m , the deviation after mean normalization can be approximated as $\mathbb{E}\left[\frac{2 \cos \phi_{XD}}{\xi_m}\right]$. As $\xi_m \rightarrow \infty$, the deviation, Δ_μ approaches 0. Hence, for high SNR or simple noise, there is little deviation in the mean. This is the case of speech in *volvo* noise in Figure 4.2.

When the signal is in low SNR, there are many TF bins with low ξ_m . Speech in difficult noise has many TF bins with low ξ_m , too. In these scenarios, Δ_μ can be approximated as $-2\mathbb{E}[\log \xi_m]$. This implies that the deviation depends on ξ_m . When there are many TF bins with low SNR, or when some instantaneous SNRs are very low, the deviation will be significant.

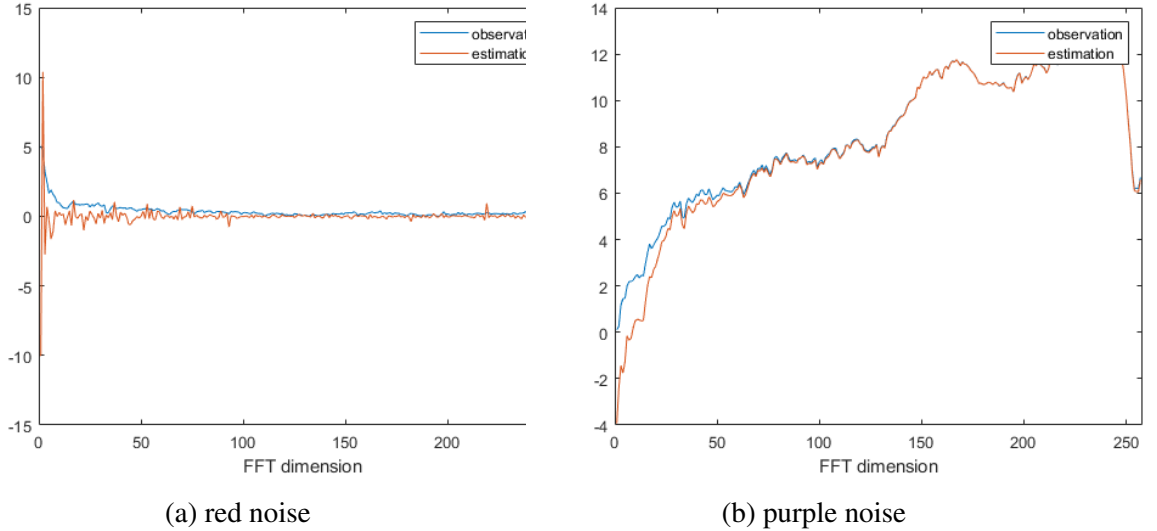


Figure 4.3: The observed mean deviation agrees with the estimated mean deviation

Next, we rely on simulation to verify the results in Equation 4.14. An utterance of clean speech is separately mixed with *red* noise and *purple* noise to create two segments of recordings. The details of *red* and *purple* noise can be found in Appendix C. Because the spectrum of *red* noise is more similar to that of speech, it can be better masked by speech than *purple* noise. On the other hand, *purple* noise contains energies at speech spectral

valleys. Our analysis in Chapter 3 suggests that speech in *red* noise is simpler than speech in *purple* noise. We would expect speech in *red* noise to suffer less from mean deviation, whereas speech in *purple* noise will experience significant deviation.

We compute the estimated mean deviation of speech in *red* noise using the first condition in Equation 4.14. The estimated deviation of speech in *purple* noise is estimated using the second condition in Equation 4.14. We also measure the actual deviation and denote it as “observed.” The estimated and observed deviation for each noise are plotted in orange and blue, respectively in Figure 4.3.

For *red* noise shown in Figure 4.3a, both the observed and the estimated deviation are very close to 0, suggesting little mean deviation for this simple noise. As a result, normalized features will be effectively centered around 0. This behavior is desirable as it matches the distribution of centered features in training the SE DNN.

For *purple* noise shown in Figure 4.3b, both the theoretical prediction and the empirical observation show that the deviation increases at higher frequency bins. This agreement is expected as *purple* noise has a larger power density in higher frequency ranges. The result confirms that the mean deviation depends on $-2\mathbb{E}[\log(\xi_m)]$, where ξ_m is generally very low in high-frequency bins for *purple* noise.

4.2.3 Deviation of variance in normalizing speech in difficult noise types

We consider the difference between the variance of noisy speech feature, $\text{Var}(\log Y_m^2)$, and the global variance, $\sigma_{LPS}^2 = \text{Var}(\log X_m^2)$

$$\Delta_{\sigma^2} = \text{Var}(\log Y_m^2) - \text{Var}(\log X_m^2) \quad (4.15)$$

If Δ_{σ^2} is around 0, then the noisy speech feature will have unit variance after normalization. Otherwise, the feature will not be scaled to a desirable range. If $\Delta_{\sigma^2} < 0$, normalizing the noisy speech feature with σ_{LPS}^2 will over-compress the features, resulting in tightly

clustered features, such as speech in *pink* and *white* noise in Figure 4.2.

We can simplify the expression in Equation 4.15 depending on ξ_m . We leave the details to Appendix B and present the final results directly

$$\Delta_{\sigma^2} \approx \begin{cases} \mathbb{E}\left[\frac{2 \cos \phi_{XD}^2}{\xi_m}\right], & \text{if } \xi_m \rightarrow \infty \\ \text{Var}(\log D_m^2) - \text{Var}(\log X_m^2). & \text{if } \xi_m \rightarrow 0 \end{cases} \quad (4.16)$$

The results in Equation 4.16 can be interpreted as follows. At high SNR or in simple noise, $\xi_m \rightarrow \infty$. Then the deviation of variance is negligible as $\mathbb{E}\left[\frac{2 \cos \phi_{XD}^2}{\xi_m}\right] \rightarrow 0$. This corresponds to *volvo* noise in Figure 4.2. In contrast, for speech in difficult noise or at low SNR, the deviation depends on how different the noise spectrum is from the speech spectrum. This corresponds to *pink* and *white* noise in Figure 4.2.

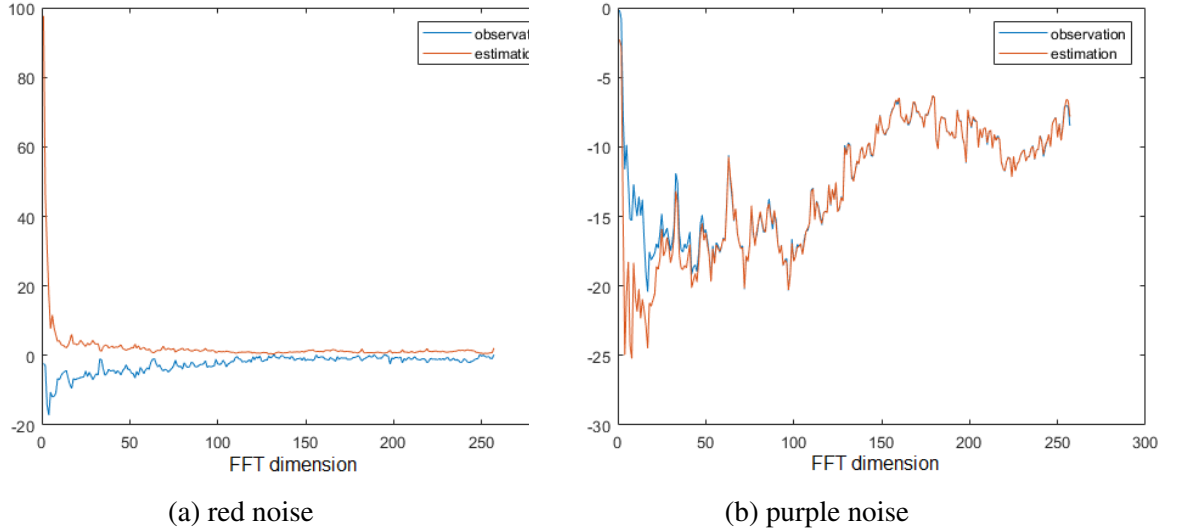


Figure 4.4: The observed variance deviation agrees with the estimated variance deviation.

We also examine the agreement of the estimated deviation and the observed values. The same speech in *red* and *purple* noise from the previous section is reused for our measurement. We plot the results in Figure 4.4. Both the estimated and observed deviation is very small for *red* noise in Figure 4.4a. This small deviation suggests that the normalized feature will have unit variance and be scaled properly. For *purple* noise in Figure 4.4b,

the deviation is significant in both our prediction and measurement. Furthermore, the deviation is negative, suggesting that the noisy speech LPS has a smaller variance than the global variance. As a result, the normalized feature will have a variance of less than 1. This explains the tight clusters for difficult noise types in Figure 4.2.

Over-compression of normalized features makes the features more tightly clustered and less separable. In a classification task such as recognition, the densely clustered feature vectors leave less margin for a DNN to find decision boundaries. In a regression task such as enhancement, the features are less distinguishable, generating a smeared spectrogram in reconstruction.

The analysis above shows that noise can affect feature normalization in speech enhancement. Difficult noise makes the normalized features not centered at zero and over-compressed. The offset and the shift introduce a mismatch between the normalized features during testing time. In the next section, we will discuss some techniques to address such a mismatch in the indirect enhancement pipeline.

4.2.4 Mean-variance matching

Mean-variance normalization is a linear transformation of input features so that the shifted and rescaled features possess desirable mean and variance. Mean normalization in the cepstral domain, as reviewed in Chapter 2, removes the convolutional effects of the channel if the channel is assumed to be stationary. Though lacking a physical interpretation, variance normalization is still widely adopted in DNNs to allow more efficient convergence of back-propagation [159]. The training process converges faster due to improved numerical conditions of the optimization. Normalization also ensures that the default initialization of network layer weights is appropriate [160].

The rationale behind matching the normalized statistics is to enforce the training and testing features to follow a similar distribution [163]. When training a DNN, normalization statistics can be adjusted to account for variations in features. The adjustment ensures that

training features are all normalized properly to zero mean and unit variance. When testing or enhancing an utterance in difficult noise types, we cannot perfectly normalize the input features due to the factors analyzed in the previous section. Hence, we wish to change the distribution of input features into another distribution that can be normalized properly, such as speech in simple noise. We name this process mean-variance matching. Even though speech features are converted, there is no need to transform features frame by frame. We are only interested in matching the mean and the variance of the transformed features.

In order to match the mean and variance statistics of noisy speech, we first estimate these statistics from simple and difficult speech, respectively. Speech in difficult noise is denoted as the source domain with subscript S . Speech in simple noise is denoted as the target domain with subscript T . We could estimate the mean and variance statistics in either domain reliably by accumulating sufficient frames of speech. Let the dimension-wise mean and variance of speech features in the target domain be μ_T and σ_T^2 , respectively. The mean statistics in each feature dimension is estimated by averaging all frames, X_{T_i}

$$\mu_T = \frac{1}{M} \sum_i^M X_{T_i}. \quad (4.17)$$

The variance is estimated over observed frames in the target domain

$$\sigma_T^2 = \frac{1}{M-1} \sum_i^M (X_{T_i} - \mu_T)^2. \quad (4.18)$$

The mean, μ_S , and variance, σ_S^2 , of speech in the source domain are estimated in a similar manner. The standard deviation, σ_S and σ_T , are the square roots of their respective variances. With these mean and standard statistics available, two simple affine transformations will translate and scale speech feature vectors in the source domain, X_S , to match the distribution of that in the target domain, X_T , in each feature dimension.

$$\hat{X}_T = \frac{X_S - \mu_S}{\sigma_S} \sigma_T + \mu_T. \quad (4.19)$$

The transformed speech \hat{X}_T shares a distribution more similar to that of a simple noise. Hence, it is reasonable to expect the transformed speech to yield better enhancement results over the directly enhanced speech.

4.2.5 Histogram equalization

Clean speech features are more Laplacian than Gaussian [164]. Hence, matching the low order statistics, including mean and variance, does not necessarily match the overall distribution. If more data is available, we could obtain reliable estimates of higher-order statistics, allowing us to match higher moments of the speech vectors. Histogram equalization is one such technique [127] by matching all moments. Like mean-variance normalization, a one-to-one transformation is also created for each utterance in the difficult noisy speech domain to the simpler domain. The target distribution is the distribution of speech feature vectors from an appropriate intermediate target environment. The distribution in each dimension is estimated by accruing sufficient features in the simple noise. For LPS feature, X_T , from the target domain, we approximate the distribution by its probability mass function, $f_T(i)$, which represents the fraction of LPS values at level i

$$f_T(i) = \text{Prob}(X = i) = \frac{n_i}{n}, \quad (4.20)$$

where n_i is the count of values at level i out of n total observations. Since the range of LPS features is unbounded, we have to estimate its extreme values. We can then specify a proper range for i . The cumulative distribution function (CDF), $F_T(i)$, can be obtained by summing $f_T(i)$

$$F_T(i) = \sum_{j=-\infty}^i \text{Prob}(X = j). \quad (4.21)$$

The CDF for speech in the source domain, $F_S(i)$, can be obtained in the same way for the source feature, X_S . It is shown that we can first apply F_S to X_S to obtain a uniform distribution [49]. Subsequently, the inverse CDF, F_T^{-1} , will transform the original feature,

X_S , to possess the same distribution as X_T . Hence, the overall transformation process is

$$\hat{X}_T = F_T^{-1}(F_S(X_S)). \quad (4.22)$$

The refinement module processes the transformed speech next for final enhancement. It is nevertheless important to note that since more observations are required to obtain accurate estimates of the exact distribution, many utterances must be collected for histogram equalization to be accurate. We will evaluate the data size requirement in the next section.

4.2.6 Experiments and discussions

In the following experiments, clean speech from the WSJ0 corpus [158] is mixed with babble, pink, and white noise from Noisex92 [103] to synthesize noisy speech. We first show that speech in difficult noise, such as *white* noise, suffers from mismatched normalization. Chapter 3 discussed that the average power spectrum density of *volvo* noise is masked by speech, but white noise is not. Thus, we could consider *volvo* a simple noise and *white* a difficult noise. Figure 4.5 compares the mean and variance of noisy speech in *white* noise (blue) and *volvo* noise (red). The yellow curves are the references: 0 for the mean and 1 for the variance. While the statistics of speech in *volvo* noise (red) hover above and below the reference, the statistics of *white* noise (blue) are consistently off from the reference. It suggests that the speech in *white* noise is likely sampled from a different distribution from the data used to compute the reference. Hence, its enhancement result is likely to be unsatisfactory.

Besides comparing the mean and variance, we also examine the overall distribution. To better visualize the shift in distribution due to noise, we first compare the distribution of input features subject to different SNR levels. Unlike noise, SNR has a natural interpretation in relation to the noisy environment’s adversity, so it is easier to lend us insights into its influence. A random dimension in the clean input feature vector is selected, and its distribu-

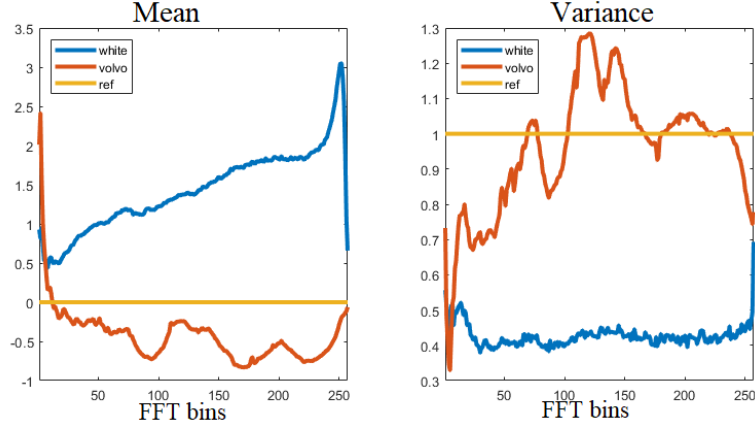


Figure 4.5: Deviation from zero mean (left) and unit variance (right) between simple and difficult noise samples. *White* is an example of difficult noise and *volvo* is an example of simple noise.

tion is shown in purple in Figure 4.6a. The clean feature, denoted as *ref*, is shown in purple. At 5dB white noise, the feature shifts to the yellow distribution. As the SNR level further decreases, the distributions at 0dB and -5dB are shown in red and blue, respectively. It is straightforward to notice that the more adverse the condition is, i.e., the lower the SNR, the further apart the distribution is from the reference. It confirms our assumption that lower SNR results in a larger deviation in normalized features.

With this intuition, we next examine the shift of the distribution of noisy speech in different background noise at 0dB, shown in Figure 4.6b. Speech in *volvo* (blue histogram) is chosen as the reference since it is considered a simple noise. The yellow histogram is speech in *babble* noise. The purple and orange histograms belong to speech in *pink* and *white* noise, respectively. Since *white* noise shows the greatest deviation from the simple noise, it should be the most challenging noise type, just as speech in -5dB is the most challenging SNR condition in Figure 4.6a. This observation is consistent with the quality assessment with PESQ. The first column in Table 4.1 tabulates the PESQ score of speech in *volvo*, *babble*, *pink*, and *white* noise at 0dB. Since speech in *white* noise shows larger variation from the reference than *babble*, its PESQ score is also the lowest, as expected.

The improvement of direct enhancement is limited in *babble* noise. There is no gain

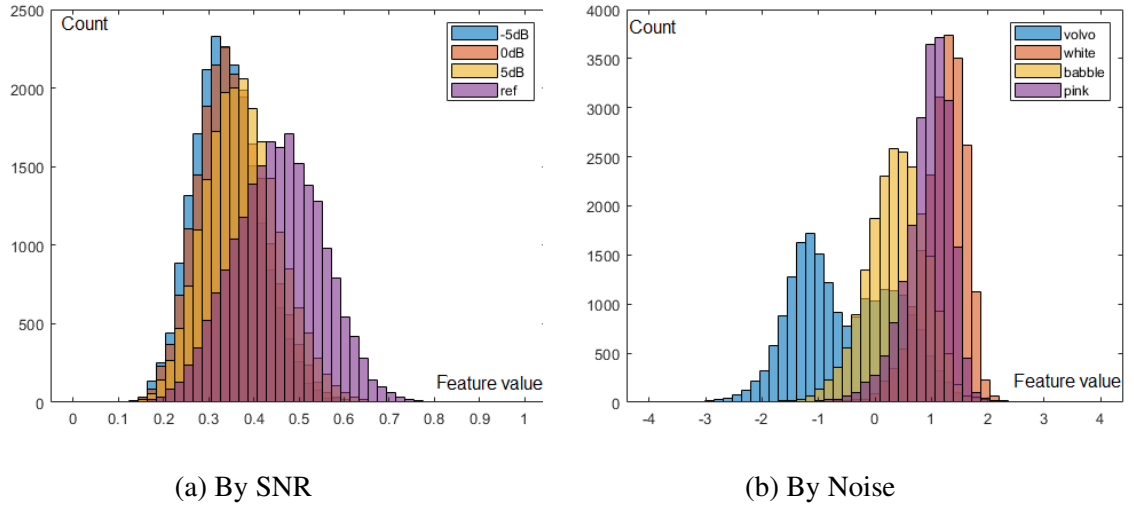


Figure 4.6: Comparison of feature distribution by SNR and by noise

PESQ	raw	direct	mean-variance normalization	histogram equalization
volvo	3.20	3.25	-	-
babble	1.81	2.12	1.99	2.03
pink	1.63	1.70	2.07	2.16
white	1.56	1.48	2.07	2.16

Table 4.1: PESQ score with and without matching statistics

for speech in *white* noise. We apply mean-variance normalization as described in subsection 4.2.4 to *babble*, *pink*, and *white* speech. The PESQ results in Table 4.1 show that it is a good solution to very challenging noisy speech, such as speech in *pink* and *white*. After applying mean-variance normalization, the feature mismatch decreases. Enhancement of the transformed speech yields a PESQ score of 2.07 for speech in both noise types, a large improvement from 1.70 and 1.48, respectively. Mean-variance normalization is ineffective with *babble* noise. The reason could be that *babble* is only moderately difficult, as its mismatch is not as serious as *pink* and *white* noise, as indicated in Figure 4.6b. Hence, if the mean and variance are not estimated reliably, the distortion is likely to outweigh matching statistics' benefits.

Next, we examine the results with histogram equalization. Figure 4.7 displays the effect of histogram equalization. The source domain is speech in *white* noise (blue). The

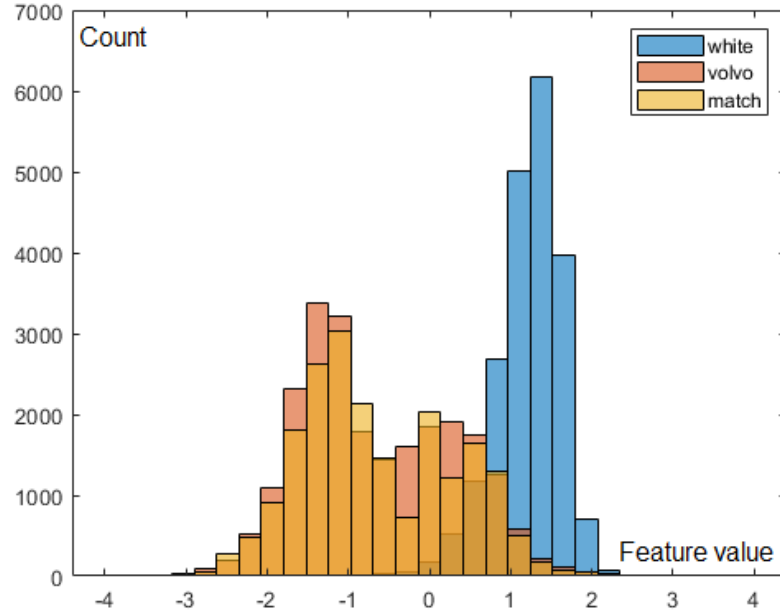


Figure 4.7: Effect of histogram equalization on feature distribution

target domain is speech in *volvo* noise (orange). Even though the two have very different distributions, we can apply the transformation described in Equation 4.22. The result of histogram equalization is shown in yellow in Figure 4.7. The transformed features follow a similar distribution as speech in *volvo* noise. Applying speech enhancement on the transformed feature further improves speech quality after mean-variance normalization. The last column in Table 4.1 shows that the PESQ score of speech in *pink* and *white* noise can be further enhanced to 2.16. Hence, indirect speech enhancement with matching feature statistics is an effective strategy for difficult noise environments.

The effect of histogram equalization can be better understood by analyzing the distribution of latent variables inside the neural network before and after the feature transformation. We sample a random node at the middle layer of the enhancement DNN for *volvo*, *babble*, *white*, and *pink* noisy speech. Figure 4.8a displays the distribution of each noise type. Similar to Figure 4.6b, the distributions in the latent layer also show a recognizable trend reminding us of that in input features. Specifically, speech in *pink* and *white* noise (*pink*

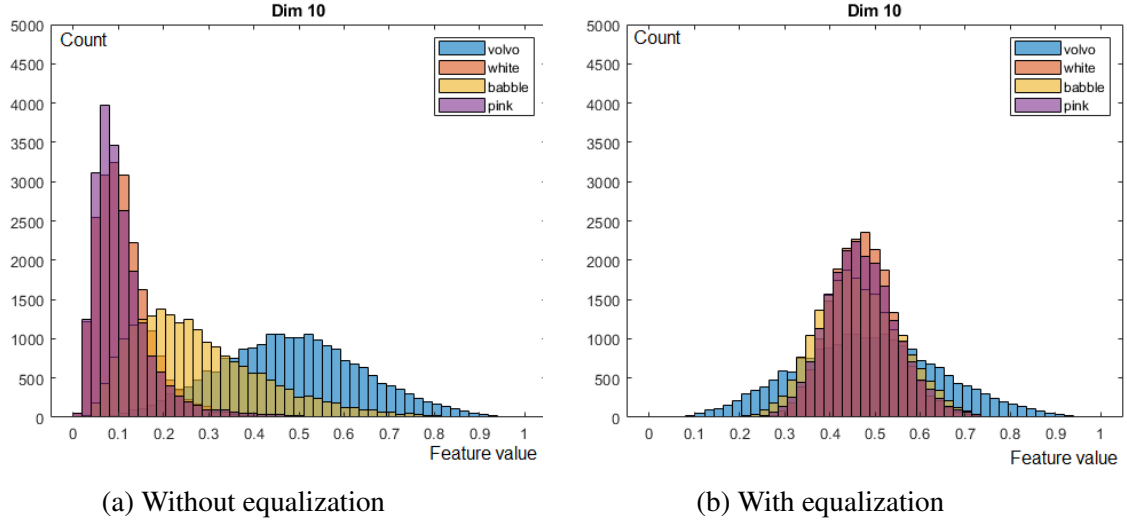


Figure 4.8: Effect of histogram equalization in hidden layers

	babble	pink	white
No equalization	1.23	4.75	6.85
With equalization	0.05	0.13	0.20

Table 4.2: KLD between other noise and *volvo* in a hidden layer

and red color) are more mismatched from *volvo* (blue color) than *babble* noise (yellow color) because they are more challenging. After histogram equalization, the distributions become more alike in Figure 4.8b. The Kullback–Leibler divergence (KLD) between the hidden representation of noisy speech and *volvo* speech has also been greatly reduced, as shown in Table 4.2. As a result, the PESQ scores of enhanced speech are more comparable in the last column of Table 4.1, regardless of the original noise type.

4.3 Speech conversion with DNN mapping

In the previous section, we only match the overall distribution of the noisy speech features to that of a simpler noisy speech. It is effective when the overall distribution can be accurately estimated. We use simple linear transformations with mean-variance normalization because we only need to estimate a few statistics. When we can simulate more data, or use a more complicated model, we could perform indirect enhancement by transforming

speech in difficult noise to speech in simple noise frame by frame. In this section, we focus on this frame-based conversion technique using DNNs.

4.3.1 DNN training

We implement the conversion stage in Figure 4.1 with a DNN. Let the original noisy speech in difficult noise condition be $x_1[n] + d_1[n]$, where $x_1[n]$ is clean speech and $d_1[n]$ is background noise. $x_1[n]$ could be from a new speaker's speech not included in the training corpus. The intermediate target noise, $d_2[n]$, is chosen based on the factors discussed in Chapter 3. Segments of silence are selected from $x_1[n] + d_1[n]$ to filter out just the noise segments. Next, clean speech from a speech corpus, $x_2[n]$, is mixed with both the original noise, $d_1[n]$, and the target noise, $d_2[n]$. This creates parallel training pairs, $x_2[n] + d_1[n]$ and $x_2[n] + d_2[n]$. They are parallel because the underlying clean speech is matched sample by sample, hence frame by frame after short-time Fourier transform (STFT).

During the training of the converter DNN, the input to the DNN is the LPS feature of speech in difficult noise, $X_2 + D_1$. The label is the LPS of speech in intermediate target noise, $X_2 + D_2$. Then we train the parameters of the neural network using stochastic gradient descent to minimize the MSE loss. During conversion, intermediate speech is predicted by feed-forward speech features of original noisy speech features through the DNN. We then expand the converted features to include multiple context windows by concatenating adjacent frames. The concatenated features are fed into the refinement network for complete the process of indirect enhancement. This process is depicted pictorially on the left side in Figure 4.9.

The second stage, denoted as refinement, can be further fine-tuned. When used with converted features, it only needs to enhance speech in the specific noise environment. Hence, the refinement network can be a specific purpose, not a general-purpose enhancer. It is possible because the intermediate target is selected beforehand. We can be fine-tune the refinement DNN d to map $X_2 + N_2$ to X_2 following the same DNN training procedure.

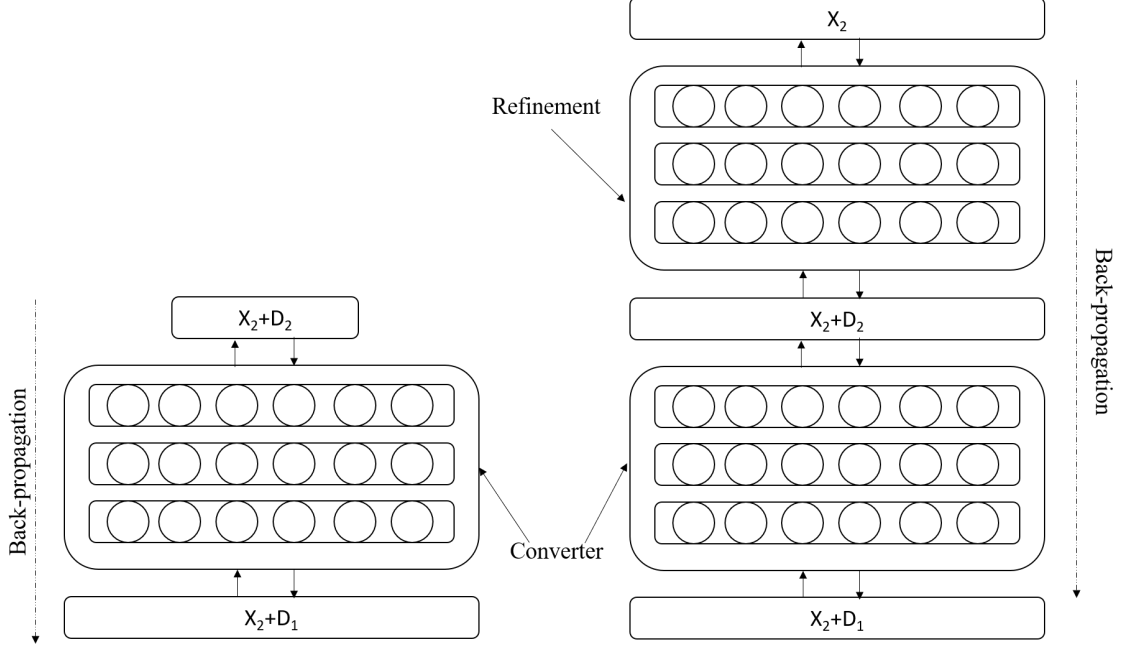


Figure 4.9: DNN architecture for feature mapping and joint training

The top-right block in Figure 4.9 shows this process.

Lastly, the two stages in indirect enhancement can be jointly optimized. The refinement DNN is stacked on top of the converter. A context-expansion layer that extends a frame to neighboring frames is inserted between those two networks. The joint system’s input is noisy speech feature, $X_2 + N_1$, and the target label is the clean speech feature, X_2 . The whole system is fine-tuned to minimize the MSE between predicted output and clean targets with a gradient descent optimizer.

4.3.2 Experiments and discussions

We evaluate the same noisy speech, i.e., speech in *babble*, *pink*, and *white noise*, as discussed in section 4.2. We select speech in *volvo* noise as the intermediate target for each environment in the indirect approach. The converter and the refinement DNNs are separately trained using the techniques described in the previous section. Both networks are 3-layer DNN with a width of 2048 in each layer. The nonlinear activations between layers are sigmoid functions. We also perform joint training by concatenating the converter

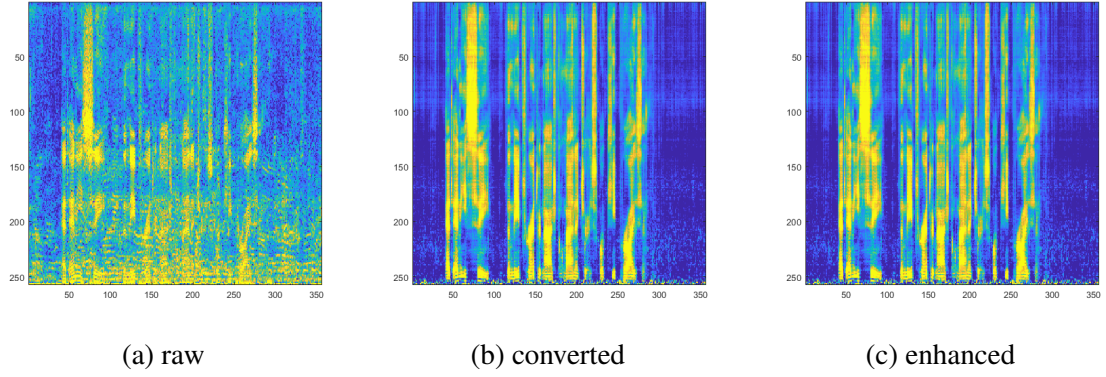


Figure 4.10: Conversion and enhancement of speech in babble noise

and the refinement and training the combined system at a lower learning rate. The PESQ scores of various direct and indirect systems in different noise conditions are tabulated in Table 4.3. The 95% confidence levels are appended after each score value. The column labeled as “direct” shows that direct enhancement is difficult for speech in *pink* and *white* noise, as the PESQ scores are still below 2.0 after direct enhancement. The proposed DNN mapping method, labeled as “indirect,” is better than the mean-variance normalization introduced in section 4.2. It is because the feature mapping method matches the features frame by frame. On a detailed level, noise is substantially removed, and speech distortion is minimized. More importantly, it is also effective with speech in moderately challenging noise, such as *babble*, as the PESQ score all improve to above 2.4.

Figure 4.10, Figure 4.11, and Figure 4.12 present speech examples of indirect enhancement with a DNN in *babble*, *pink*, and *white* noise. In each figure, the three spectrograms correspond to the noisy speech without processing (raw), converted with a DNN (converted), and post-enhanced (enhanced). First, most of the original noise has been removed after the conversion stage, as the original noise is no longer visible in the converted spectrograms at the center. The converted features facilitate the refinement stage, so that the enhanced spectrograms show no visible residue noise.

The last two columns in Table 4.3 illustrate the effect of joint training vs. direct adaptation. For all three noise types, the indirect method with joint training, labeled as “indirect

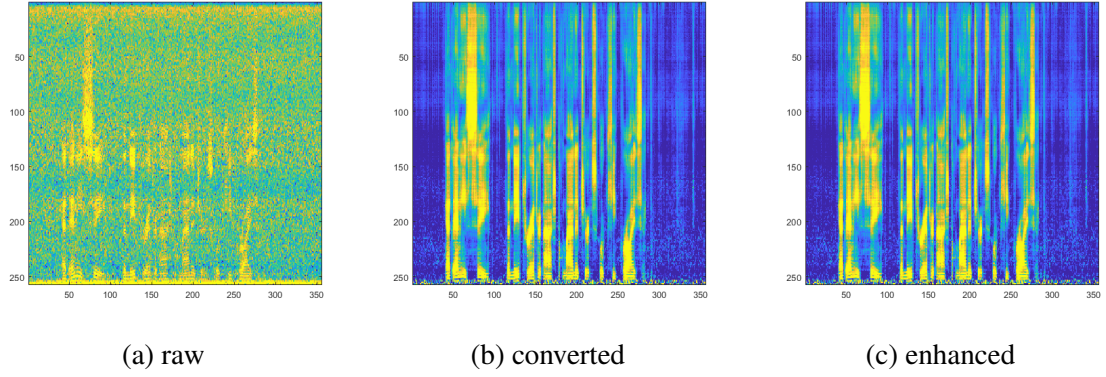


Figure 4.11: Conversion and enhancement of speech in pink noise

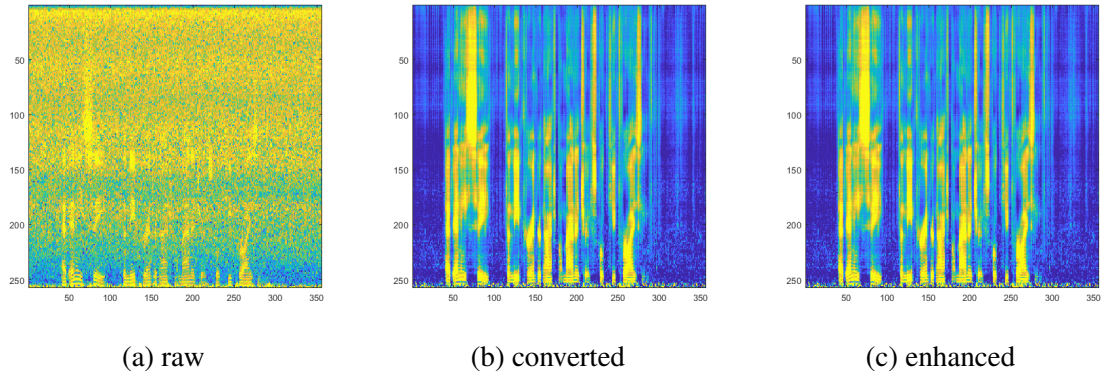


Figure 4.12: Conversion and enhancement of speech in white noise

refined” outperforms direct adaptation, labeled as “direct refined.” We could explain the improvement by comparing the MSE between various learning targets in Table 4.4. For example, direct enhancement from *white* noise to clean speech needs to reduce an MSE gap of 4.53. If *volvo* is selected as the intermediate target, the first stage needs to close an MSE gap of 4.40, and the second stage only has a gap of 1.29. That makes either stage a simpler task for a DNN to learn. We could draw the same conclusions for *babble* and *pink* noise in Table 4.4.

4.4 Interference of multiple noise sources

In many real-world situations, multiple noise sources may exist during a conversation. In the simple additive noise model introduced in Chapter 3, the corrupted speech, y , can be

	raw	direct	MV matching	indirect	direct refined	indirect refined
babble	1.81 ± 0.05	2.11 ± 0.05	1.99 ± 0.05	2.47 ± 0.05	2.56 ± 0.05	2.62 ± 0.04
pink	1.63 ± 0.05	1.69 ± 0.09	2.07 ± 0.05	2.43 ± 0.05	2.50 ± 0.04	2.54 ± 0.04
white	1.56 ± 0.06	1.48 ± 0.09	2.07 ± 0.05	2.45 ± 0.05	2.49 ± 0.05	2.60 ± 0.05

Table 4.3: Progressive indirect enhancement with *volvo* intermediate noise

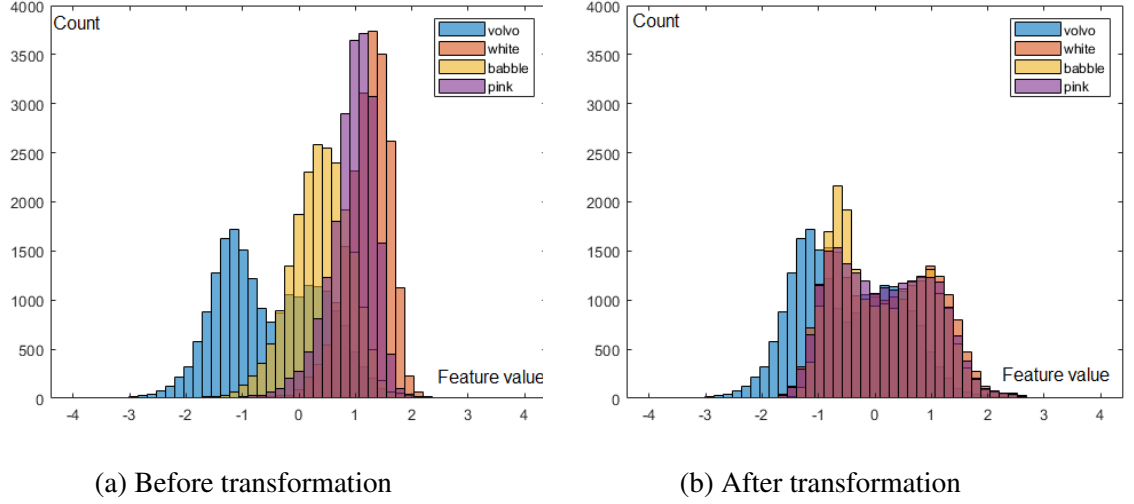


Figure 4.13: Effect of conversion on feature distribution

modeled as a linear combination of clean speech, x , and noise, d_i , scaled by its SNR factor, γ_i . The respective SNR is $SNR_i = 10\log_{10}\frac{1}{\gamma_i} = -10\log_{10}\gamma_i$

$$y = x + \sum_i \gamma_i d_i. \quad (4.23)$$

If noise sources are independent, the overall SNR is approximately

$$SNR = 10\log_{10}\frac{1}{\sum_i \gamma_i} = -10\log_{10}\sum_i \gamma_i. \quad (4.24)$$

	babble	pink	white
noisy-clean	2.54	3.94	4.53
noisy-intermediate	2.16	3.71	4.40
intermediate-clean	1.29	1.29	1.29

Table 4.4: MSE between various learning pairs

In direct DNN-based speech enhancement, we could generally consider the mixture of noise a single source of interference, i.e., $d = \sum_i \gamma_i n_i$, so noisy speech, $y = x + d$, is mapped to clean speech, x , directly. To achieve better performance of DNN-based enhancement, we want the noise acoustic space to be broad [10]. It implies that the training data must contain superpositions of noise, too. However, the combination of noise types increases exponentially as the noise database grows. It is not easy to enumerate and simulate all such combinations during direct training. The indirect approach to enhancement offers an alternative solution by removing only one interference every time, thus simplifying the task for each stage.

4.4.1 Framework of indirect enhancement with two noise sources

In this section, we consider speech mixed with two noise sources. It serves as a starting point to discuss speech in multiple noise sources. When two loud noise interferences exist in speech, the overall SNR is low if noise sources are independent. The overall low SNR level makes direct mapping difficult. Eliminating only one noise source each time is simpler because the SNR gap is smaller than removing all noise at once. Furthermore, a single noise often only corrupts part of the speech spectrum, so a neural network can still rely on the rest of the signal spectrum in prediction.

After each stage, one noise source is removed from the mixture. This process is repeated until no noise source exists or clean speech has been recovered. An example process involving two noise sources, d_1 and d_2 , is presented in Figure 4.14. The original noisy speech consists of speech in two interferences, n_1 and n_2 . Direct mapping attempts to transform the noisy speech, $x + n_1 + n_2$, to x in a single step, as shown on the left. There are two options for the indirect path. The first path, shown in the middle in Figure 4.14, uses $x + n_1$ as the intermediate target. In this case, noise n_2 is removed first. The intermediate speech is subsequently enhanced to clean speech, x . The other option is to remove n_1 first. We could visualize this path on the right in Figure 4.14. The order of removal of the two

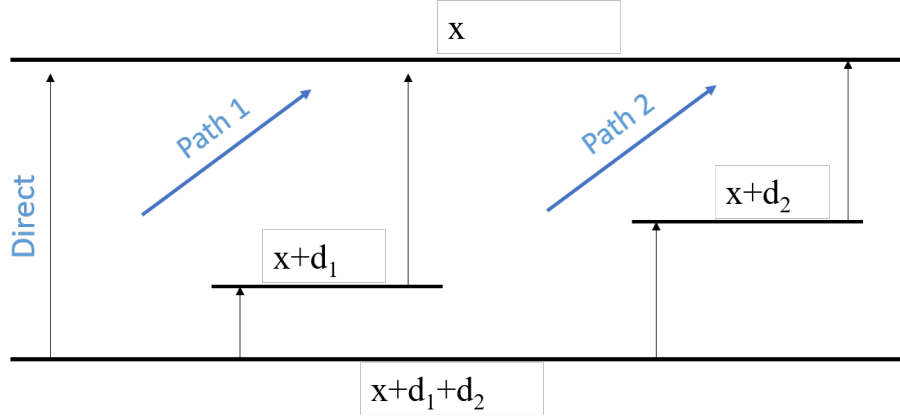


Figure 4.14: Indirect enhancement of multiple interferences

interferences has been switched.

In general, either path should yield an improvement over direct enhancement, as shown in section 4.3. Nevertheless, one path is preferable to the other based on the noise characteristics of n_1 and n_2 . We could apply the same analysis in Chapter 3 to select the appropriate intermediate stage. Figure 4.14 assumes that n_1 is a more difficult noise, since the mapping from $x + n_1$ to x incur a larger loss. In contrast, n_2 is a simpler noise. If *Path 1* is chosen, the second transformation from $x + n_1$ to x will remain difficult. The distortion in DNN mapping might overshadow the benefit of incremental improvement. We split the task with commensurate difficulties along *Path 2*, which ensures that each sub-task becomes easier to learn.

4.4.2 Experiments and discussions

In the following experiments, we assume that there are only two noise sources in the noisy speech. Furthermore, we assume that we can obtain isolated recordings of both noise types. This scenario could arise when we detect a new noise source in an environment where we have already collected some audios before. For instance, we have previously recorded the background noise in a mechanic shop. We also have babble noise in our noise database. When we need to enhance speech mixed with both babble and machine noise in

a mechanic shop, we could apply the indirect approach proposed in the previous section. In our experiment, two different noise sources are added to clean speech to simulate noisy speech in such an environment. Clean speech is sampled from the WSJ corpus [158]. The first interfering noise is either *babble* or *factory1* noise from Noisex92 [119] corpus. The second interference is a colored noise, including *brown*, *gray*, *blue*, *pink*, *purple*, and *white*. A detailed description of each noise can be found in the appendices. For each noisy-clean pair, we compute the mean squared difference of their LPS features. The difference allows us to gauge the distance between the conversion or enhancement pairs. We conduct and compare indirect enhancement along both paths in Figure 4.14.

In Table 4.5, the noise sources are *babble* noise and a colored noise. Both noise sources are scaled to 3dB, so the overall SNR between the clean speech and the interference is 0dB. As the SNR is relatively low, the raw audio has PESQ scores below 2.0, which is in the row labeled as “raw PESQ” in Table 4.5 and Table 4.6. The direct enhancement achieves some improvement. On average, audio quality can be improved to 2.52, shown in the row labeled “direct PESQ” in the same table.

In comparison, the indirect approach along either path shows further improvement over the direct approach. The results are labeled “path1 PESQ” and “path2 PESQ” in Table 4.5 and Table 4.6. Along *Path 1*, the noise from Noisex92 is always the intermediate target. That means speech in *babble* is the intermediate target in Table 4.5, and *factory1* is the intermediate target in Table 4.6. Along *Path 2*, speech in the corresponding colored noise listed in the table is the intermediate target. On average, either path achieves an improvement of about 0.1 in PESQ score over direct enhancement, which is perceptually significant. The last column of Table 4.5 shows that direct enhancement achieves a score of 2.52 on average. The indirect approach along either path achieves a score of 2.61 or 2.59. The improvement could be explained by simpler sub-tasks. In general, each stage in the proposed indirect path has a lower MSE compared to the direct task. Nonetheless, we have argued that it is better to choose sub-tasks with comparable difficulty. We could

	Brown		Gray		Blue		Pink		Purple		White		Average	
direct MSE	8.0		8.9		22.0		16.1		22.3		22.1		16.6	
path1 MSE	0.4	8.2	0.8	8.2	8.2	8.2	4.1	8.2	8.7	8.2	8.0	8.2	5.0	8.2
path2 MSE	2.6	3.6	2.8	4.6	0.8	21.7	0.5	16.2	1.5	21.4	0.5	22.6	1.4	15.0
raw PESQ	1.99		1.99		1.72		1.74		1.80		1.59		1.81	
direct PESQ	2.70		2.68		2.47		2.39		2.48		2.37		2.52	
path1 PESQ	2.75		2.77		2.55		2.50		2.62		2.44		2.61	
path2 PESQ	2.80		2.74		2.51		2.45		2.61		2.41		2.59	

Table 4.5: 3dB babble noise mixed with various colored noise at 3dB. The intermediate target is *babble* for *Path 1* and the corresponding colored noise for *Path 2*.

	Brown		Gray		Blue		Pink		Purple		White		Average	
direct MSE	13.7		13.9		22.9		17.8		23.4		22.4		19.0	
path1 MSE	0.2	14.7	0.2	14.7	3.1	14.7	1.4	14.7	3.4	14.7	3.0	14.7	1.9	14.7
path2 MSE	6.1	3.6	5.2	4.6	1.1	21.7	1.1	16.2	1.8	21.4	0.7	22.6	2.7	15.0
raw PESQ	1.88		1.89		1.68		1.68		1.74		1.56		1.74	
direct PESQ	2.64		2.66		2.44		2.4		2.49		2.35		2.50	
path1 PESQ	2.68		2.77		2.54		2.49		2.60		2.50		2.61	
path2 PESQ	2.79		2.81		2.52		2.53		2.58		2.40		2.61	

Table 4.6: 3dB factory noise mixed with various colored noise at 3dB. The intermediate target is *factory1* for *Path 1* and the corresponding colored noise for *Path 2*.

confirm this claim in Table 4.5. For example, in the first column under *brown* noise, *Path 2* contains two sub-tasks with similar MSE, 2.6 and 3.6. The two sub-tasks along *Path 1* are more dissimilar in terms of difficulty, with MSE values of 0.4 and 8.2, respectively. The PESQ score along *Path 2* (2.80) is higher than that along *Path 1* (2.75). For *white* noise, the MSE difference indicates that *Path 1* is preferred. The PESQ scores show that *Path 1* has higher scores of 2.44 over 2.41 along *Path 2*. Thus, when we need to design an indirect path that allows us to choose the order of intermediate targets, it will be more favorable to select intermediate targets that result in sub-tasks with comparable difficulties.

The same experiment is repeated by replacing the first noise source from *babble* to *factory1*, which has more high-frequency components than *babble*. In Table 4.6, the results show a similar trend as the previous experiment. For simpler noise types such as *brown* and *gray*, it is better to choose *Path 2* as *brown* and *gray* are simpler than *babble*. *Brown*

PESQ	Babble/Brown						Babble/White					
SNR/dB	1.25	6	3	3	6	1.25	1.25	6	3	3	6	1.25
raw	1.89		1.99		2.16		1.67		1.59		1.54	
direct	2.62		2.71		2.87		2.37		2.37		2.44	
indirect	2.72		2.80		2.94		2.46		2.44		2.54	
improvement	0.83		0.81		0.78		0.79		0.85		1.0	

Table 4.7: *Babble* and colored noise at various SNR

PESQ	Factory1/Brown						Factory1/White					
SNR/dB	1.25	6	3	3	6	1.25	1.25	6	3	3	6	1.25
raw	1.77		1.88		2.08		1.61		1.56		1.54	
direct	2.53		2.64		2.81		2.34		2.35		2.40	
indirect	2.68		2.79		2.92		2.43		2.50		2.50	
improvement	0.91		0.91		0.84		0.82		0.94		0.96	

Table 4.8: *Factory1* and colored noise at various SNR

and *gray* noise are easily masked by speech, as evident from the power spectral estimate shown in Figure 4.15. The red curves show the average PSD of speech, and the blue curves represent those of various colored noise. The average spectrum of *brown* and *gray* are effectively masked by speech, whereas other colored noise is not. For the rest of the more difficult types, speech in *factory1* noise becomes relatively easy. It is thus a more appropriate intermediate target. In conclusion, for indirect enhancement of multiple noise interference, it is better to convert to speech in simpler noise that can be effectively masked by speech, which is consistent with the analysis in Chapter 3.

In the next experiment, we adjusted the two noise source mixing ratio while maintaining the overall SNR level at 0dB. We want to know if intermediate targets' choice would still be the same as a result of changes in relative SNRs of each interference. Table 4.7 shows the result of noisy speech when *babble* is mixed with *brown* or *white* noise. Table 4.8 repeats the experiment by replacing *babble* with *factory1* noise. The second rows in both tables specify the mixing ratio of two interferences: 1.25dB/6dB, 3dB/3dB, and 6dB/1.25dB. In terms of relative energy, the two noise are mixed at 3:1, 1:1, and 1:3, respectively. The overall SNR remains still at 0dB.

By inspecting the first three columns in Table 4.7 and Table 4.8, we could see that the

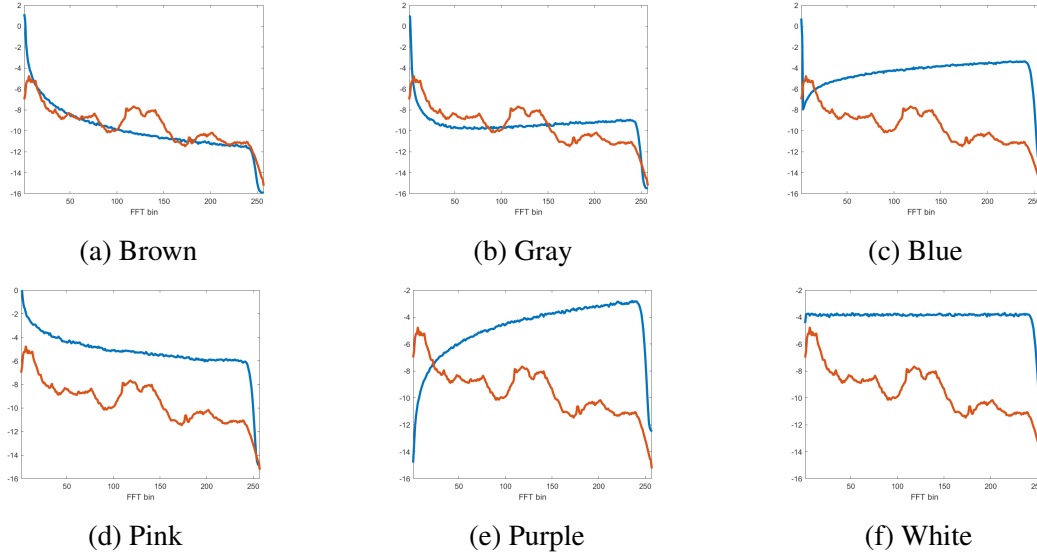


Figure 4.15: Average power spectrum density of speech and various colored noise

improvement is greater when *babble* or *factory1* has lower SNR than *brown* noise because *babble* or *factory1* is harder than *brown*. A lower SNR in a harder noise makes the noisy speech more difficult overall. Consistent with our results in earlier sections, we find that the indirect approach is more effective when it is corrupted by difficult noise. Following the same argument, we can see the reversed trend for *white* noise. Namely, larger improvement is seen when *babble* or *factory1* noise is at higher SNR. That is because *white* noise is a more difficult noise type than *babble* or *factory1*. Hence, noise power from each source should be considered together with noise types to select intermediate targets in indirect speech enhancement.

4.5 Summary

In this chapter, we discussed several approaches to realize indirect speech enhancement by focusing on designing the conversion step. A crucial factor for the degrading performance of speech in some difficult environments is the different feature distribution. It creates a mismatch between the feature used in training and testing. We demonstrate that difficult noise types cause the features to be normalized improperly, which results in a mismatch

between the features in training and testing. The first method in this chapter uses mean-variance normalization or histogram equalization to reduce the mismatch. Reducing the mismatch in the feature space also translates to a reduced mismatch in hidden activations in DNNs, which explains the improvement with indirect speech enhancement.

If more paired data is available for supervised learning, one can convert original noisy features into speech features in a simple noise frame by frame using regression-based mapping, such as a DNN. Compared to only matching the distribution, the mapping method reduces distortion in spectral details and yields better enhancement quality. It outperforms the normalization method in terms of perceptual quality scores.

Lastly, indirect speech enhancement can be effectively applied when multiple noise interferences are present. In this scenario, noise can be progressively removed from the mixture one by one. It is better than direct enhancement, which removes all noise at once. We performed an analysis of task difficulties by measuring the MSE gap between input and output features. The MSE gap confirms that it is useful to decompose difficult learning tasks into simpler sub-tasks. Empirical evidence also shows that it is more useful to convert speech in simple noise as intermediate targets.

CHAPTER 5

INDIRECT SPEECH ENHANCEMENT WITH LATENT SPACE LEARNING

5.1 Introduction

In Chapter 3, we developed a few guidelines to find noise types suitable as intermediate targets. Nonetheless, it is possible that speech in that intermediate target is difficult to collect or synthesize. For instance, the amount of target noise collected is too little to synthesize any meaningful training data set, or an online adaptation system does not offer enough time to collect data and train a converter separately. Hence, in this chapter, we explore ways to perform noisy type conversion with unsupervised learning when there are no direct learning targets of speech in the desirable noise background.

Representational learning is one such tool we can utilize. Given enough unlabeled training data, a good representation learning could discover a structured representation of features in a latent space. Features in the latent space lie on a manifold, a continuous non-intersecting surface [159]. Manifolds have some interesting properties, such as *feature disentanglement* and *latent vector arithmetic*. Feature disentanglement aims to extract different aspects of the latent representation features. We wish to decompose the noisy speech into speech features, noise features, and SNR features in our application. Subsequently, latent vector arithmetic enables us to manipulate the latent space components by replacing some attributes while keeping the rest fixed, thus accomplishing conversions. Afterward, the re-synthesis step would combine the new latent features and create a converted output.

In the rest of the chapter, we will introduce various forms of autoencoder (AE) that are very popular in *representational learning* [159]. Since nonlinearities in deep AE make them harder to analyze, we start discussing linear models, allowing us to understand the latent structure better. Then we will see linear models fall short of extracting structured

latent features. Modifications are required by building deeper architecture and imposing latent constraints. We will follow the analysis by a series of experiments to show that we can convert noisy speech from difficult noise into simpler ones for indirect enhancement using unsupervised learning.

5.2 Representational learning via auto-encoder

AEs have been popular in unsupervised learning as a tool to perform feature extraction or dimension reduction [159]. More recently, its use has extended beyond deterministic mappings to describe probabilistic distributions, such as generating modeling. A basic AE consists of an encoding block and a decoding block. Let \mathbf{x} stand for an input feature vector. In speech enhancement, it is usually an LPS vector. The encoding block, F_{enc} , transforms \mathbf{x} to a latent representation, \mathbf{h} , which is generally much more compact and structured

$$\mathbf{h} = F_{enc}(\mathbf{x}). \quad (5.1)$$

In its most common form, an AE is *under-complete*. The latent layer, \mathbf{h} , also commonly referred to as the bottleneck layer, has a smaller dimension than the input. It enforces data compression in the encoding process. The decoding block, G_{dec} , has to learn to reconstruct the original feature at the output layer from the compact latent representation, \mathbf{h} . The following equation describes the whole process of auto-encoding

$$\hat{\mathbf{x}} = G_{dec}(\mathbf{h}) = G_{dec}(F_{enc}(\mathbf{x})). \quad (5.2)$$

An AE is often trained with MSE loss to minimize the difference between original inputs and reconstructed outputs

$$F_{enc}, G_{dec} = \arg \min_{F, G} \sum_i L_{MSE}(\mathbf{x}_i, \hat{\mathbf{x}}_i) \quad (5.3)$$

where the reconstruction loss, L_{MSE} , is the standard MSE

$$L_{MSE}(\mathbf{x}_i, \hat{\mathbf{x}}_i) = \|\mathbf{x}_i - G_{dec}(F_{enc}(\mathbf{x}_i))\|_2^2 \quad (5.4)$$

Compared to reconstructed outputs, we are usually more interested in the latent representation as features are more saliently organized in this low dimensional subspace.

The most basic AE could be modified to include additional constraints in the bottleneck layer, \mathbf{h} . For example, a regularization loss of weight, λ , could be added to Equation 5.3 to enforce sparsity in the latent space

$$F_{enc}, G_{dec} = \arg \min_{F, G} \sum_i (L_{MSE}(\mathbf{x}_i, \hat{\mathbf{x}}_i) + \lambda \|F_{enc}(\mathbf{x}_i)\|_1^2). \quad (5.5)$$

De-noising AE is another variation that minimizes the loss in Equation 5.6, where $\delta\mathbf{x}_i$ is a small perturbation in input features to promote more robust feature extraction and reconstruction. This property makes it a handy tool in feature selection. Its training finds two networks, F_{enc} and G_{dec} , that optimize the following loss

$$F_{enc}, G_{dec} = \arg \min_{F, G} \sum_i L_{MSE}(\mathbf{x}_i, G_{dec}(F_{enc}(\mathbf{x}_i + \delta\mathbf{x}_i))). \quad (5.6)$$

The exact realization of F_{enc} and G_{dec} can take many forms depending on the applications. Usually, some nonlinear transformations are included in F_{enc} such that $G_{dec} \circ F_{enc}$ does not degenerate into a linear multiplication. In the case the whole AE is linear, it is similar to PCA. Both can be used in dimension reduction and feature extraction. In the next section, we will first use a linear model to analyze the issue of unsupervised noisy speech conversion.

5.2.1 Latent space of speech features using PCA

In a linear AE, F_{enc} and G_{dec} are simply two matrices in Equation 5.3. Then the reconstructed output could be written as

$$\hat{\mathbf{x}} = G_{dec}(F_{enc}(\mathbf{x})) = \mathbf{G}_{dec}\mathbf{F}_{enc}\mathbf{x}, \quad (5.7)$$

where \mathbf{G}_{dec} and \mathbf{F}_{enc} are the matrices instantiating the decoder and the encoder, respectively. It draws a close parallel with PCA. By collecting many speech feature, \mathbf{x} , we could create an observation matrix, \mathbf{X} , where each row corresponds to a feature vector. The feature matrix, \mathbf{X} , is first centered by removing its mean in each dimension, $\bar{\mathbf{X}}$. Then we can perform PCA on the centered matrix, $\tilde{\mathbf{X}} = \mathbf{X} - \bar{\mathbf{X}}$, to obtain an orthonormal loading matrix, \mathbf{V} . It is the same matrix, \mathbf{V} , if SVD is performed on $\tilde{\mathbf{X}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ ¹. It is straightforward to see that the hidden representation, \mathbf{H} , is

$$\mathbf{H} = \tilde{\mathbf{X}}\mathbf{V} = \mathbf{U}\mathbf{\Sigma}, \quad (5.8)$$

and the reconstruction, $\hat{\mathbf{X}}$, is

$$\hat{\mathbf{X}} = \tilde{\mathbf{X}}\mathbf{V}\mathbf{V}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \quad (5.9)$$

If a lower dimension, R , is desired, some columns of \mathbf{V} could be discarded. Alternatively, define a column-trimming matrix, \mathbf{D} , such that

$$\mathbf{D} = \begin{bmatrix} \mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_R, \mathbf{0}, \dots, \mathbf{0} \end{bmatrix}, \quad (5.10)$$

¹ $\mathbf{V}^T = \mathbf{V}^{-1}$ is an orthogonal matrix.

where \mathbf{e}_i is an indicator vector with only the i th position as 1 and 0 everywhere else. A total of R dimensions will be selected. In this case, the lossy reconstruction would be

$$\hat{\mathbf{X}} = \tilde{\mathbf{X}}(\mathbf{VD})(\mathbf{VD})^{-1}, \quad (5.11)$$

where $(\mathbf{VD})^{-1}$ is the pseudo-inverse of \mathbf{VD} since $\text{rank}(\mathbf{VD}) < \text{rank}(\mathbf{V})$.

The ideal binary mask, \mathbf{M} , discussed in Chapter 2 indicates whether a time-frequency bin is dominated by speech or noise energy. It is usually multiplied element-wise to the feature matrix, \mathbf{X} , to filter out noise dominated bins. In a linear AE such as PCA, the binary classification of speech and noise bins leads to masking in latent representations. In particular, the latent speech and noise are respectively

$$\mathbf{H}_{speech} = \mathbf{M} \otimes \tilde{\mathbf{X}}\mathbf{VD}, \quad (5.12)$$

and

$$\mathbf{H}_{noise} = (\mathbf{1} - \mathbf{M}) \otimes \tilde{\mathbf{X}}\mathbf{VD}, \quad (5.13)$$

where \otimes is the element-wise product.

To better understand this result, we use the following toy example: Let $\mathbf{m} = [1, 1, 0, 0]$ be a four-dimensional vector. The mask vector means the first two dimensions are dominated by speech and the last two by noise. The feature vector, $\tilde{\mathbf{x}} = [x_1, x_2, x_3, x_4]$, is a frame of centered feature vector in \mathbb{R}^4 . Each x_i is a frequency bin. Assume the AE has a bottleneck width of 3, so the truncation matrix, \mathbf{D} can be written as $\mathbf{D} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, 0]$. \mathbf{V} is the encoding/loading matrix with column vectors, $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$, and \mathbf{v}_4 . Combining \mathbf{VD} creates $[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$. Then noisy speech in the latent space is $\tilde{\mathbf{x}}\mathbf{VD}$

$$\mathbf{H}_{noisy} = \tilde{\mathbf{x}}\mathbf{VD} = \left[\sum_{i=1}^4 x_1 v_{i1}, \sum_{i=1}^4 x_2 v_{i2}, \sum_{i=1}^4 x_3 v_{i3} \right].$$

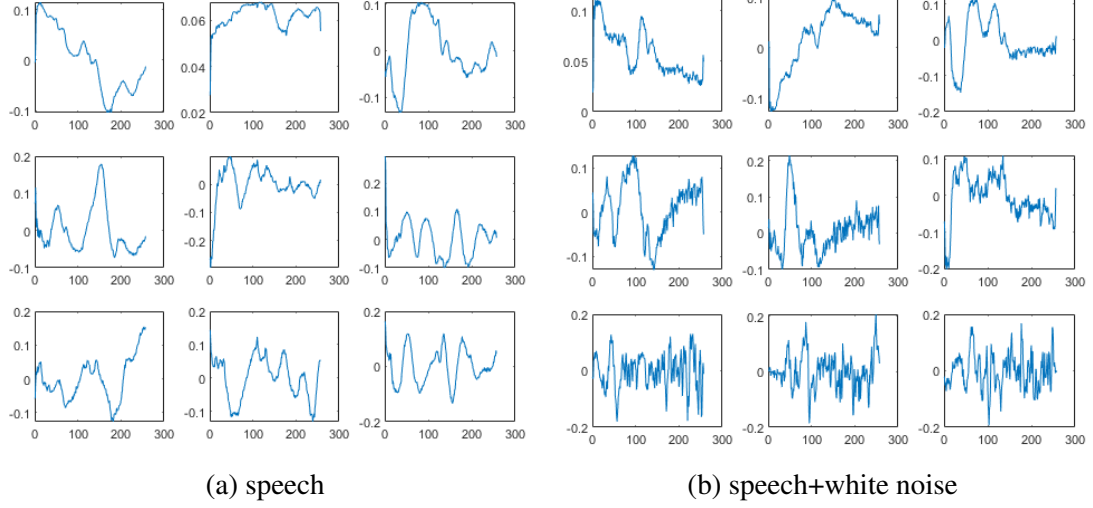


Figure 5.1: First 9 latent speech bases extracted from clean and noisy speech

Since the mask, \mathbf{m} , only keeps the lower two dimensions of \mathbf{x} as speech

$$\mathbf{H}_{speech} = \mathbf{m} \otimes \tilde{\mathbf{x}}\mathbf{V}\mathbf{D} = \left[\sum_{i=1}^2 x_1 v_{i1}, \sum_{i=1}^2 x_2 v_{i2}, \sum_{i=1}^2 x_3 v_{i3} \right],$$

and noise is the upper two dimensions

$$\mathbf{H}_{noise} = (\mathbf{1} - \mathbf{m}) \otimes \tilde{\mathbf{x}}\mathbf{V}\mathbf{D} = \left[\sum_{i=3}^4 x_1 v_{i1}, \sum_{i=3}^4 x_2 v_{i2}, \sum_{i=3}^4 x_3 v_{i3} \right].$$

The equations above show that encoding with a linear AE is similar to selecting and combining speech and noise bases in the loading matrix, \mathbf{V} .

We perform SVD on the centered feature matrix of speech to get \mathbf{V} . The first nine columns corresponding to the first nine singular values are selected and plotted in Figure 5.1a. We only select nine because other singular values are considerably smaller. First, we can tell these nine bases resemble the PSD of various speech activities. For example, the first basis resembles the long-term average PSD. The second base contains stronger high-frequency components, reminding us of the spectrum of some consonant sounds. Bases from four to nine seem to suggest some formant patterns.

Next, we plot the first nine bases of speech in *white* noise in Figure 5.5b. Comparing

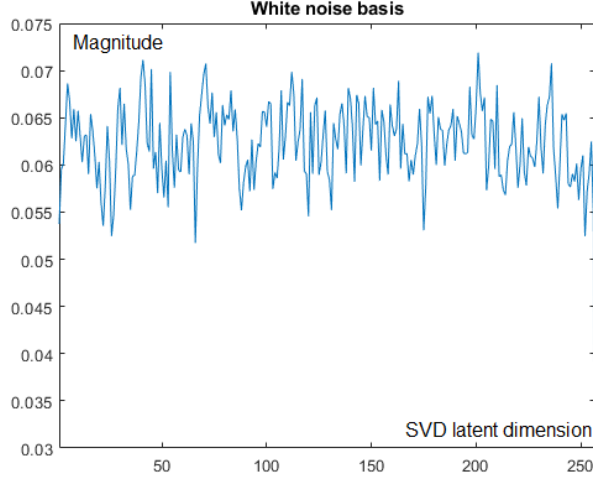


Figure 5.2: First latent *white* noise basis

Figure 5.1a and Figure 5.5b, we are almost able to see a one-to-one correspondence between the bases extracted from noisy speech and those extracted from the clean speech. Their overall shapes of each basis are very similar. A notable difference between those two is the bases in Figure 5.5b have stronger high-frequency perturbations, which is due to the contribution of *white* noise. We visualize this contribution by plotting the most significant latent vector of *white* noise in Figure 5.2. Unsurprisingly, this basis resembles the flat spectrum of white noise. It also contains high-frequency components visible in Figure 5.5b. This visualization helps us better understand our earlier argument that speech bases and noise bases span the latent space in a linear AE.

The analysis above hinted to us on performing noisy speech conversion with an AE. Since the latent representation is a superposition of the speech bases and noise bases in the latent space, one can replace the bases to obtain different noisy speech reconstruction. In particular, let $\tilde{\mathbf{Y}}$ be the centered observation matrix of noisy speech in another noise and $\tilde{\mathbf{Y}} = \mathbf{U}_{\mathbf{Y}} \Sigma_{\mathbf{Y}} \mathbf{V}_{\mathbf{Y}}^T$, we can then perform a basic noise type conversion

$$\hat{\mathbf{Y}} = \tilde{\mathbf{X}} \mathbf{V} \mathbf{V}_{\mathbf{Y}}. \quad (5.14)$$

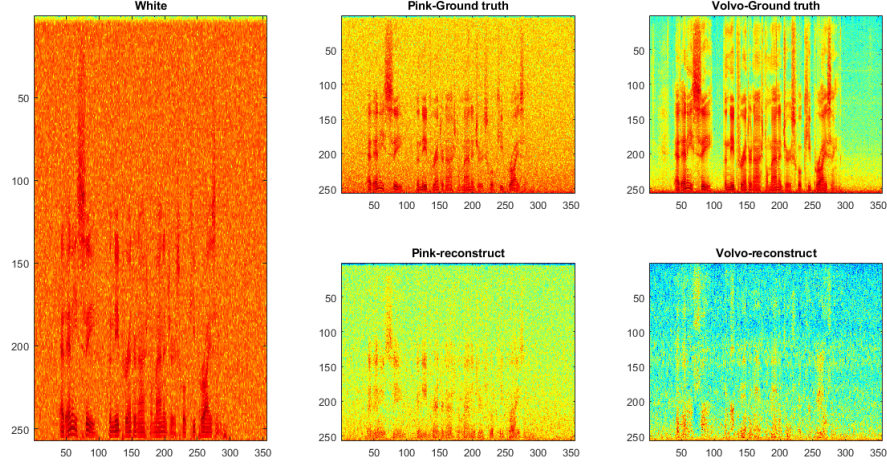


Figure 5.3: Conversion of noisy speech from *white* noise into *pink* and *volvo* with PCA

First, the encoder, \mathbf{V} , transforms noisy speech in the source domain into a latent representation. This latent representation can be decomposed into speech and noise subspaces. The decoder, \mathbf{V}_Y , is calculated from the noisy speech in a target noise domain. It corresponds to speech and another noise subspace. It will transform the latent weights into speech in another noise. Figure 5.3 presents a sample result of conversion using this technique.

A potential issue with this approach is the permutation of bases in different loading matrices. We cannot guarantee that the bases are ordered in the same way after performing SVD. Due to the different effects of noise on speech, this order may change from noise to noise, which could create misaligned weights and bases. As a result, this technique would not work well when the noise types are too different, as dissimilar noise types are likely to shuffle the order of bases. For example, in Figure 5.3, since both *white* noise and *pink* noise are wide-band noise, the conversion of noisy speech from *white* noise (left) to noisy speech in *pink* noise (bottom center) results in less distortion compared to conversion from *white* noise to *volvo* (bottom right). By measuring the MSE between the converted speech and their ground truth in Table 5.1, we can confirm that conversion to very different noise is indeed very tough with this approach.

This leads us to explore deep and nonlinear architectures in subsequent sections for

Table 5.1: Effects of stationarity of conversion targets

source noise	white	white
target noise	pink	volvo
MSE to ground truth	3.7	16.7

noisy speech conversion.

5.2.2 Use of nonlinear auto-encoders to convert speech features

A deep AE also consists of an encoder and a decoder. Nonlinear activation layers are usually inserted after each linear layer to prevent linear transformations from collapsing into a single matrix multiplication. Deep networks have a lot more representational power than shallow networks. As pointed out in [165, 166], some functions can be expressed by deep networks that cannot be approximated by shallower ones, unless the shallow networks are impractically wide. Thus, depth exponentially reduces the number of parameters required to represent a function and lowers the demand for training data. From a compression point of view, deeper auto-encoders can achieve better compression results than their shallow counterparts [167]. In practice, one could pre-train a deep network with layers of shallow networks, and stack them to create a deeper AE.

Due to its stronger representational power, a deep AE is more general than PCA to discover latent structures in data. As a linear transform, PCA projects features onto a hyper-plane in lower dimensions. For many problems, the input data may not have a linear representation. For example, in Figure 5.4, the latent data dwell on a nonlinear manifold that cannot be described by a hyper-plane. It shows the need for nonlinear transformations in AEs.

Noisy speech conversion based on deep AEs can be formulated as discussed below. For two domains, *src* and *tgt*, we assume that the encoders, F_{enc} , and decoders, G_{dec} , in the source and target domain have been well trained. Hence,

$$\mathbf{X}_{src} \approx G_{src}(F_{src}(\mathbf{X}_{src})), \quad (5.15)$$

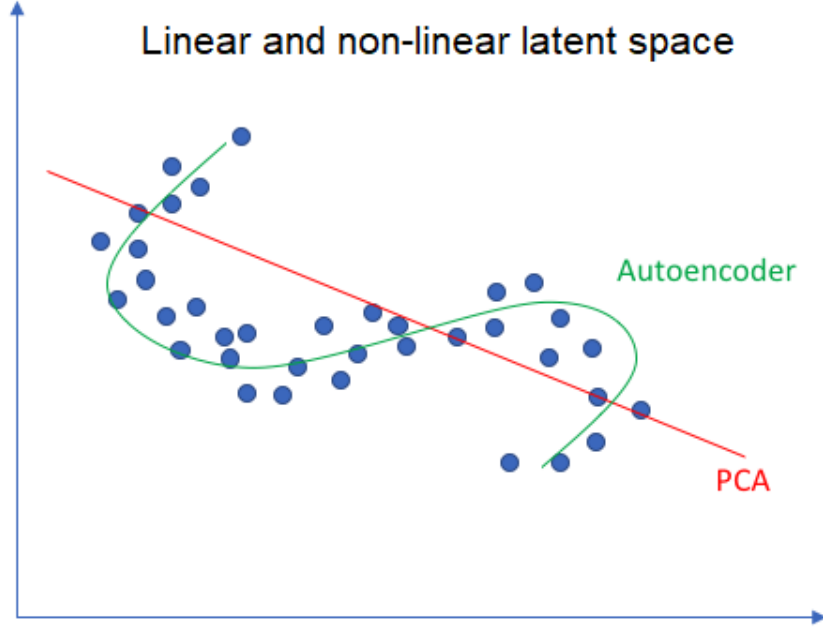


Figure 5.4: PCA vs Deep AE in manifold learning.

and

$$\mathbf{X}_{tgt} \approx G_{tgt}(F_{tgt}(\mathbf{X}_{tgt})). \quad (5.16)$$

By assuming $F_{src}(\mathbf{X}_{src}) = F_{tgt}(\mathbf{X}_{tgt})$, we can generate converted feature, $\hat{\mathbf{X}}_{tgt}$, with

$$\hat{\mathbf{X}}_{tgt} = G_{tgt}(F_{src}(\mathbf{X}_{src})). \quad (5.17)$$

There is much freedom in designing the architecture of a deep AE as one could explore encoder and decoder of different depth and width. Depth-wise, both the encoder and decoder can be a concatenation of several nonlinear transformations. The layers can also consist of dense, convolutional, or recurrent layers. Width-wise, even though the encoder and the decoder are usually connected by a bottleneck layer, which is much thinner than the rest of the network, one could design *over-complete* auto-encoders with a wide bottleneck layer [159]. Regardless of the specific architecture, the network parameters are still obtained via back-propagation by the same rule in Equation 5.3.

However, the number of parameters in the auto-encoder could increase if the model

gets large. It could be problematic if there are relatively little data but many parameters to estimate. In such cases, the model may only “memorize” the specific inputs seen in the data set and fails to generalize to new features. We could address the problem with regularization, such as using L1 loss to enforce sparsity in Equation 5.5. With the additional L1 loss, the decoder must rely on a small fraction of neurons in the bottleneck layer to reconstruct its output, effectively limiting the bottleneck width.

An alternative way to promote sparsity and discourage memorization is to include a term of KLD. It requires us to pre-select a sparsity constant, ρ , which incorporates our prior belief of how often a neuron in the latent layer should activate. ρ follows a Bernoulli distribution. The computation of ρ is discussed in detail in [159]. By minimizing the KLD between the prior distribution, ρ , and empirical observation, $\hat{\rho}$, we accomplish constraining the neurons in the bottleneck layer to activate only occasionally. The modified loss is

$$F_{enc}, G_{dec} = \arg \min_{F, G} \sum_i (L_{MSE}(\mathbf{x}_i, G_{dec}(F_{enc}(\mathbf{x}_i))) + \sum_j KLD(\rho || \hat{\rho}_j)). \quad (5.18)$$

where j is a neuron in the latent layer.

We could use AEs to perform representational learning on speech features. Afterward, we can convert speech in difficult noise to speech in intermediate targets to achieve indirect speech enhancement. Figure 5.5 shows an example converter. We use the unlabeled audio features to train the source AE. The audio features are from a source domain, which typically should be speech in difficult noise conditions. A similar set-up is used to train an AE in the target domain, which corresponds to speech in simpler noise. After two separate sets of AE are trained, we could perform conversion by replacing the source decoder with the target decoder. The conversion is possible if both the source and target encoders encode noisy speech into the same latent space. Additionally, both decoders must decode from the same latent space. It is a strong assumption. We will validate or challenge this assumption by examining a series of factors in the experimental section.

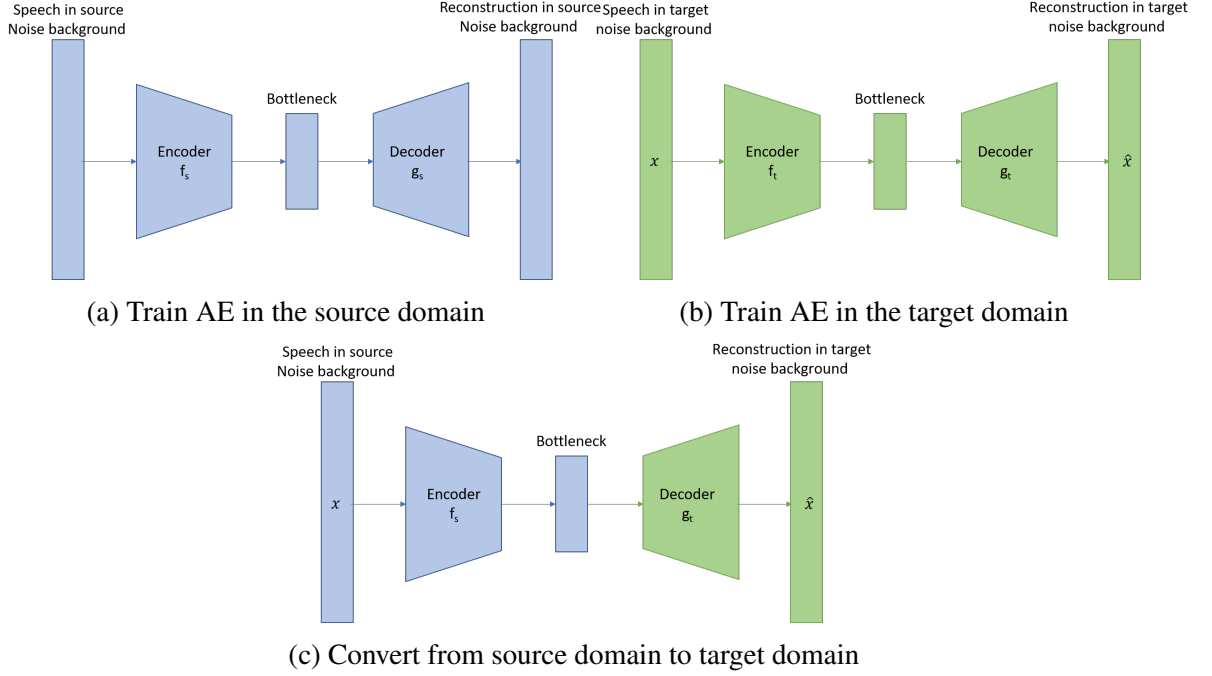


Figure 5.5: Use auto-encoders to convert noisy speech into simpler noise

Noise aware conversion

In the first part of this chapter, we have shown that conversion with a linear AE is similar to changing bases with a decoder. In a nonlinear AE, we can include target domain noise vectors to give the decoder more explicit information to perform the conversion. It allows the encoder to concentrate on learning latent representations of speech since the decoder can rely on additional input of noise vector in its reconstruction. A set-up of “noise-aware conversion” following this design is presented in Figure 5.6.

The first encoder, F_s , takes noisy speech, $\mathbf{x} + \mathbf{d}$, as input. The encoder then maps the noisy speech to the bottleneck, $F_s(\mathbf{x} + \mathbf{d})$. The second encoder H_t only encodes noise information, which outputs the noise bottleneck, $H_t(\mathbf{d})$. By giving the decoder, G_t , the noise information, $H_t(\mathbf{d})$, more explicitly, the first encoder F_s do not need to encode the noise, thus creating a speech bottleneck with better speech representation. We will examine if better speech representation in the latent layer would facilitate better reconstruction and conversion, which could lead to better enhancement results.

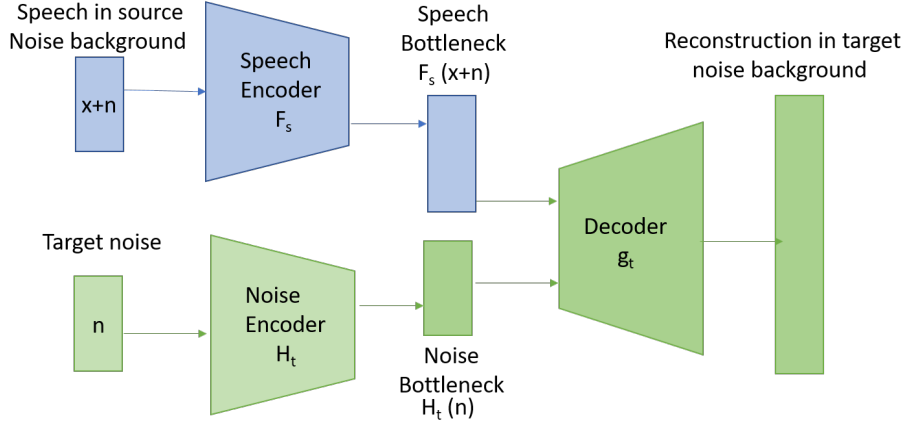


Figure 5.6: Architecture of noise aware speech conversion

Domain adversarial auto-encoder

The set-up in noise-aware AEs allows a speech encoder only to encode speech, but it may not impose enough constraints to enforce such behavior. Since the task during training is only to re-synthesize noisy speech, a speech encoder may still encode speech and noise together. We can impose additional constraints on speech encoding with a *domain adversarial loss*.

Introduced in [168], domain adversarial training extracts latent features indiscriminative to domain knowledge, such as background noise information. A domain classifier is appended after a latent layer to classify from which domain the features come. To achieve better classification accuracy, the domain classifier encourages latent features to be more discriminative. In other words, speech features from the source noise domain would be more separable from the features in the target noise domain. It is exactly opposite from what we need for background noise conversion since we do not wish to encode domain information after the speech encoder. We could reverse the phenomenon by adding a gradient reversal layer [168] between a latent bottleneck layer and the domain classifier. When training a domain adversarial auto-encoder (DAAE) in the forward pass in Figure 5.7a, the gradient reversal layer functions as an identity. The domain classifier predicts whether noisy speech is from the source or the target domain. Its loss function is the binary cross-

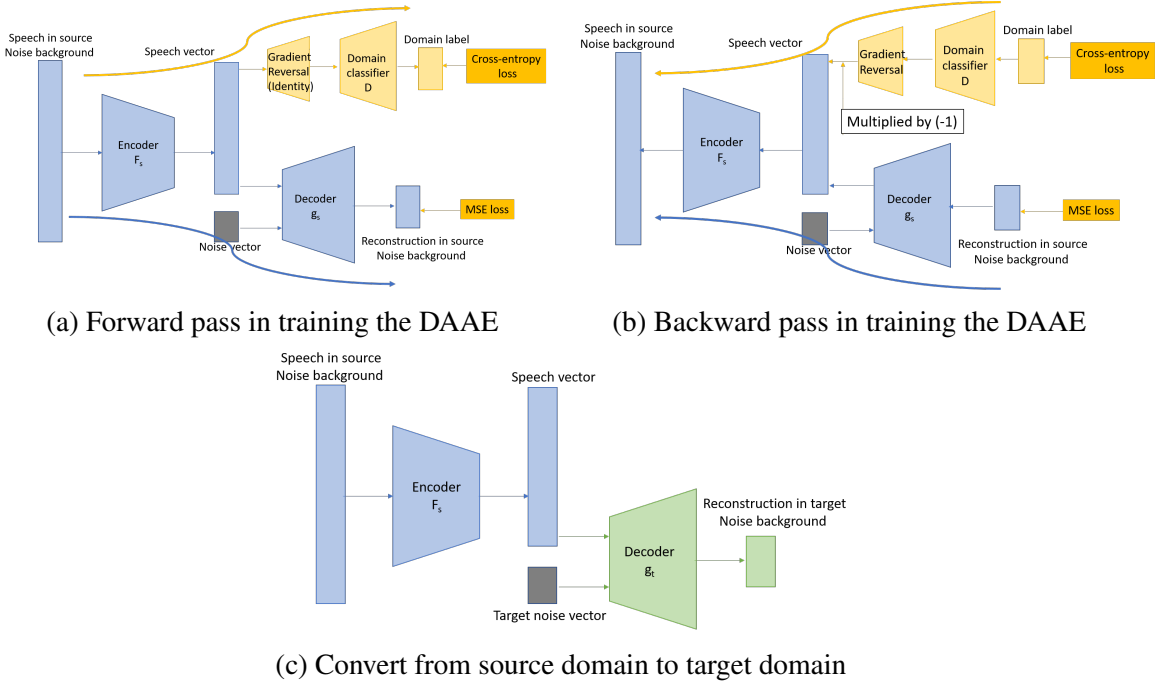


Figure 5.7: Use domain adversarial auto-encoders to convert noisy speech into simpler noise

entropy function that is typically used in classification tasks [168]

$$L_{xent} = -q_s \log(\hat{q}_s) - (1 - q_s) \log(1 - \hat{q}_s). \quad (5.19)$$

In the above equation, $q_s = 1$ when the feature is from the source domain, and $q_s = 0$ when the feature is from the target domain. \hat{q}_s is the output from the domain classifier. During back-propagation, the domain classifier is updated with standard DNN training. The gradient after the gradient reversal unit will be multiplied by a negative constant before entering the bottleneck layer, as shown in Figure 5.7b. Since the gradient is reversed, optimization with gradient descent becomes gradient ascent. Hence, subsequent updates in lower layers of DAAE will maximize the domain confusion. The adversarial mechanism makes the bottleneck features indiscriminate of domain information.

Let D_{cls} be the domain classifier. The modified objective function of this system is

$$F_{enc}, G_{dec}, D_{cls} = \arg \min_{F, G, D} \sum_i (L_{MSE}(\mathbf{x}_i, G_{dec}(F_{enc}(\mathbf{x}_i))) + \lambda L_{xent}(q_i, D_{cls}(F_{enc}(\mathbf{x}_i)))), \quad (5.20)$$

where L_{MSE} and L_{xent} are the reconstruction loss and the classification loss defined in Equation 5.4 and Equation 5.19, respectively. q_i is the ground truth domain label. Since the gradient reversal unit is parameter-free, and it automatically adjusts the gradient in back-propagation, the adversarial loss is not apparent in the loss function in Equation 5.20. During conversion in Figure 5.7c, we still replace the source decoder with the target decoder. Unlike a vanilla AE, the latent features are less indiscriminative, so it contains less information from the source domain, which helps the target decoder reconstruct noisy speech in the target domain.

Vector quantized auto-encoder

Inspired by VQ [169], we could use a set of fixed bases so that the AEs in source and target domains share the same span. Specifically, the encoder's output, $\mathbf{Z} = F_{enc}(\mathbf{X})$, will be quantized to a fixed code, \mathbf{Z}' , in the codebook, \mathbf{C} , given some distance metric, such as the euclidean distance. The decoder will then map the quantized code, \mathbf{Z}' , back to \mathbf{X}

$$\mathbf{Z}' = \arg \min_{\mathbf{c} \in \mathbf{C}} \|\mathbf{F}_{enc}(\mathbf{X}) - \mathbf{c}\|_2^2. \quad (5.21)$$

We present a frame-based VQAE system in Figure 5.8. Each frame of acoustic feature, \mathbf{X} , in the source noise background, will be quantized after the VQ block. It will be decoded with the target decoder to reconstruct speech features in the target noise background. The benefit of the usage of a fixed codebook is clear. Since the auto-encoders from both source and target domains are trained with the same codebook, the encoded activations now reside in the same latent space. Furthermore, there is no more issue with permutation as the codebook is fixed. Compared to conventional AE, VQAE also has some drawbacks. As the VQ

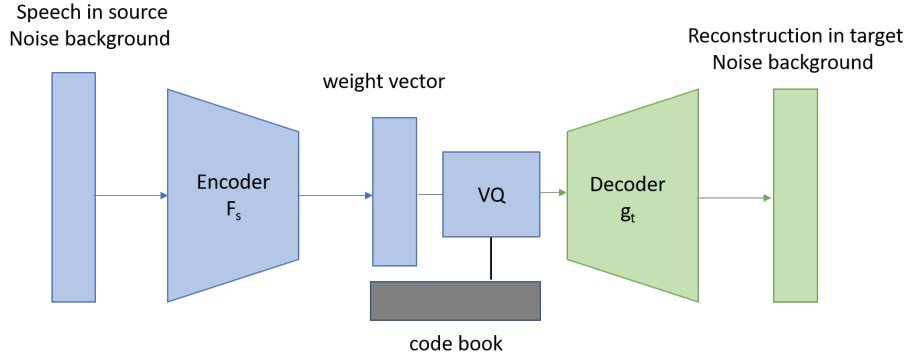


Figure 5.8: Architecture of vector quantized auto-encoder

block discretizes latent codes in Figure 5.8, the decoding output will be less smooth. The non-smoothness will contribute to distortions in reconstruction, making the reconstructed speech less natural. Secondly, the quantization step involves $\arg \min$, which is not differentiable. It breaks down back-propagation when updating the parameter weights in an AE. Lastly, the codebook must be carefully designed to facilitate encoding in both the source and target domain.

To address the first issue, we choose a relatively large codebook to reduce quantization loss. At the extreme, there could be as many codebook entries as the number of acoustic frames. It is equivalent to no quantization, effectively eliminating quantization loss. For a speaker-dependent system, the codebook size can be greatly reduced, as we do not need to consider speaker variability. As there are few practical guidelines for setting the codebook size, this quantity can be determined empirically.

There are several workarounds about the issue of differentiability of $\arg \min$'s operation. Sub-gradient or finite difference methods could be used to approximate the gradient flow. Another way to overcome the difficulty is to replace $\arg \min$, a hard assignment, with soft decisions, for example, the softmax function [159]. Compared to the original VQ, modified VQ with soft decisions allows the latent representations to reside in a continuous subspace. It will help reconstructed features to be more smooth.

5.2.3 Experimental results

This section evaluates speech transformation with methods proposed in section 5.2. The original noisy speech is the source domain, and the intermediate target speech is the target domain. AEs in the source and the target domain are first separately trained, after which we obtain a pair of encoder and decoder in the source domain and the other pair in the target domain. During conversion, the source encoder first generates latent codes of the original noisy speech. The latent codes are then fed through the decoder in the target domain to achieve speech transformation. The transformed speech is subsequently used as intermediate inputs to a general-purpose enhancement model described in Chapter 2. We perform speech transformation on a low-level spectral feature, i.e., LPS. In the following experiments, 257-dimensional LPS features of noisy speech are encoded into latent codes by encoders. The decoder output is also 257-dimensional LPS. We evaluate the quality of the final output with the PESQ.

We sample three noise types from the Noisex92 database: *white*, *pink*, and *babble* noise. *White* noise is wide-band, whereas *pink* noise is more band-limited. *Babble* noise is non-stationary. For all three types of noise in the source domain, we choose *volvo* noise as the target domain due to its characteristics, as discussed in Chapter 3.

In the rest of the section, we will evaluate and analyze factors that affect the proposed unsupervised conversion technique’s performance, including neural network architecture, SNR dependency, size of the data set, noise-aware training, domain adversarial training, and vector quantized training.

Network architecture

In subsection 5.2.2, we have argued that deep auto-encoders can learn the underlying structure in noisy speech better than linear models due to their greater depth and width. To investigate the effects of auto-encoders’ different depths, we gradually increase the number of nonlinear layers. Table 5.2 presents the results for all three types of noisy speech at

	Direct	depth=0	depth=1	depth=2	depth=3
white	1.48	1.83	1.94	1.95	1.91
babble	2.12	2.04	2.09	2.15	2.17
pink	1.70	1.92	2.03	2.05	2.03
average	1.77	1.93	2.02	2.05	2.04

Table 5.2: PESQ after enhancing using different depths of auto-encoders

0dB. The noise types, *white* and *pink*, are very difficult for the enhancement network, so the PESQ score of direct enhancement is only 1.48 and 1.7, respectively. Enhancing the converted features with a linear auto-encoder (d=0) improves the performance of these two challenging noise types considerably (1.48 to 1.83 for *white* noise and 1.70 to 1.92 for *pink* noise) but lowers the quality of speech in *babble* noise (2.12 to 2.04). As we include more nonlinear layers in the AEs, the quality improves for all three noise types with one or two nonlinear layers (d=1 and d=2). For instance, PESQ of noisy speech in *white* improves from 1.48 to 1.95. It suggests that nonlinear networks can extract features better than linear networks, as we expected. Adding more nonlinearity beyond two layers is not beneficial, as the quality drops slightly at d=3 (from 1.95 to 1.91 for *white* noise). It could attribute to the difficulty in training deeper models due to vanishing gradient and the vulnerability of over-fitting of over-parameterized models.

Transitional AEs employ narrow bottleneck layers to compress features into more succinct representations in a latent space. However, our focus is on the speech and noise component’s disentanglement in the latent space. Hence, a wider bottleneck layer, such as an *over-complete* AE, could allow the encoders to find a better structure for speech and noise separately. For an AE with only one nonlinear layer, we vary the nonlinear layer’s width from 64 to 1024. When the width is larger than the feature dimension (257), the auto-encoder is over-complete. The results are presented in Table 5.3. For each of the three noise types, PESQ of enhanced speech improves as the bottleneck width grows up until 512. For example, PESQ of speech in *white* noise improves from 1.48 up to 2.13. When the latent layer is too narrow, compression in the latent space results in high reconstruction loss, thus

	Direct	64	128	256	512	1024
white	1.48	1.76	1.94	2.09	2.13	2.13
babble	2.12	1.95	2.09	2.12	2.16	2.13
pink	1.70	1.84	2.03	2.15	2.20	2.18
average	1.77	1.85	2.02	2.12	2.17	2.15

Table 5.3: PESQ after enhancing using different bottleneck width

	Non-conversion				Conversion			
	babble	pink	white	average	babble	pink	white	average
-5	1.7	1.24	1.08	1.34	1.62	1.70	1.71	1.68
0	2.12	1.7	1.48	1.76	2.12	2.15	2.09	2.12
5	2.51	2.16	1.99	2.22	2.54	2.50	2.42	2.49

Table 5.4: PESQ of converted speech at various SNR levels

lowering the reconstruction quality. However, if the latent layer is too wide, such as 1024 nodes, there is a risk for an AE to memorize input features instead of extracting meaningful representations. Hence, an intermediate width of 256 or 512 is more appropriate for our application.

SNR dependency

The analysis in Chapter 3 and the results in Chapter 4 show that enhancement through intermediate conversion is more beneficial at low SNR conditions. We are also interested in learning its performance at various SNR levels for unsupervised speech conversion. For the following experiments, a simple AE with one nonlinear layer is used for the three noise environments mentioned before. The results are shown in Table 5.4. Comparing the columns labeled as “average” between non-conversion and conversion, we could observe the improvement of speech quality across all SNR levels. For example, PESQ improves from 1.76 to 2.12 when speech is at 0dB. The trend is also in line with previous results in Chapter 4 as the improvement is more noticeable for the most difficult noise type (*white*) compared to the simpler noise type (*babble*). The gain is also more pronounced when noise is at a lower SNR. For instance, PESQ improves from 1.34 to 1.68 at -5dB with an improvement of 0.34. In contrast, the gain reduces to 0.25 for environments at 5dB, shown

in the last row in Table 5.4.

Training data size

A primary reason for the degradation of DNN-based enhancement is domain mismatch in an unseen deployment environment. The proposed method adopts unsupervised learning to perform speech conversion, making it a reasonable candidate for system adaptation in mismatched conditions. In the following experiments, the target domain AE in *volvo* noise is trained with 10 minutes of speech by a speaker. The source domain AE, which depends on the environment at the deployment stage, is trained with varying number of utterance from 1 to 40. Each utterance is 10 seconds long on average. Figure 5.9 shows that very

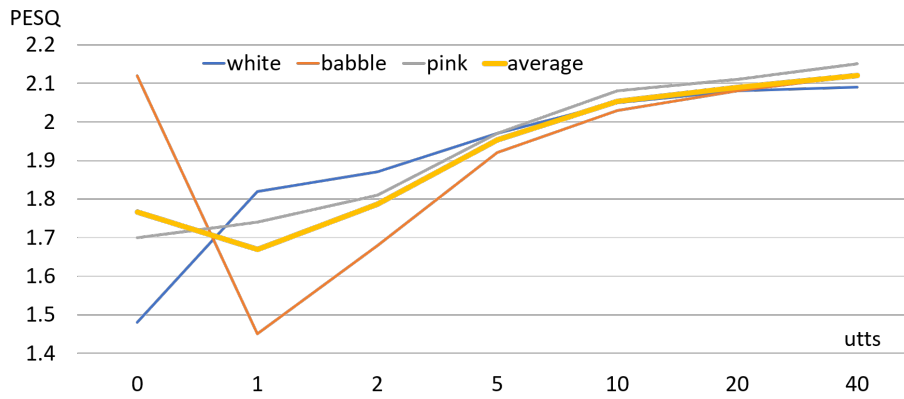


Figure 5.9: Conversion quality with respect to data size in auto-encoders

few utterances are required to achieve conversion with decent improvement for challenging noise, such as *white* and *pink*. The improvement is quite noticeable with even just one utterance for *white* noise. The performance steadily grows as more utterances are available. Figure 5.9 shows that a less challenging environment, such as *babble* noise, requires many more utterances to be effective. It is better not to perform conversion if there are insufficient utterances to train the AE as it will lead to performance degradation. When 40 utterances are used to train the source encoder, all three noise environments converge to similar performances.

	Direct	No noise information	Noise aware
white	1.48	2.09	2.17
babble	2.12	2.12	2.12
pink	1.70	2.15	2.20
average	1.77	2.12	2.16

Table 5.5: Results of noise aware training

Constraints on the latent space

Previous experiments concluded the effectiveness of using deep AEs to perform speech conversion in indirect speech enhancement, particularly in challenging noise and SNR conditions. Next, we seek to analyze the addition of constraints on the latent space that helps an encoder better separate speech and noise representation.

Noise aware training, as proposed in Figure 5.6, aims at providing explicit domain information for the AE to better disentangle speech from noise in the latent space. An encoder receives noise LPS features in addition to noisy speech features. It may help the encoder identify the noise components in noisy speech. For fairness, we keep the latent layer’s dimension fixed to compare results from the previous section. Because speech is generally more non-stationary with greater spectral variation, more neurons are required for its projections than background noise. In the following experiment, we use 246 neurons for speech and 10 for noise.

Table 5.5 compares the conversion results with and without noise aware training. There are incremental improvements for *white* (2.09 to 2.17) and *pink* noise (2.15 to 2.20), but the improvement is less significant for *babble* noise. We think that *white* and *pink* noise have stable long-term average spectra, whereas *babble* as a non-stationary noise possesses a varying spectrum. Hence, it is harder for the encoder to capture its varying spectral characteristics.

Domain adversarial auto-encoder (DAAE), as described in Figure 5.7, is investigated next. By encouraging the latent representation to be indiscriminative of the source and target domain, the latent space in both AEs becomes similar. It would allow the target

	Direct	AE	DAAE
white	1.48	2.09	2.17
babble	2.12	2.12	2.13
pink	1.70	2.15	2.17
average	1.77	2.12	2.14

Table 5.6: Results of domain adversarial auto-encoder

domain decoder to replace the source decoder without creating a mismatch between the latent bases. Table 5.6 discusses the result of DAAE. Compared to simple AE, DAAE achieves additional gain in enhancement quality (from 2.12 to 2.14), thanks to the more shared latent space.

VQAE is similar to DAAE in using a common set of bases in the latent space. DAAE promotes shared bases by penalizing latent features with domain-dependent information. This specification is more explicit in VQAE, as the encoded representation is restricted to be one of the entries in the codebook in the case of hard VQ or a linear combination of them in soft VQ. Since we are converting noisy speech into speech in target noise domain by using a decoder trained in target domain, the decoder must be able to recognize the quantized codes after encoding. Thus, we design a codebook by quantizing a basic AE’s latent codes. K-means algorithm is a straightforward method to group the latent vectors into clusters. A problem with K-means is that it requires a pre-defined number of clusters. Furthermore, Euclidean metric used in K-means tends not to perform well in high dimensions [170, 171]. It is not easy to obtain good cluster centroids that can be used as codebook entries in high dimensions. As discussed in the narrow bottlenecks experiments in Table 5.3, reconstruction is also not good if the dimension is too low. Thus, we vary the latent feature dimension (8,32, and 128) and the number of codebook entries (10,50, and 500) for each noise type. We present the results in Figure 5.10.

For each noise condition in Figure 5.10, the results are group by the size of codebooks (10,50, and 500). Within each group, the blue, orange, and the gray bar correspond to latent vectors of dimension 8, 32, and 128, respectively. We first note that the trend is uniform

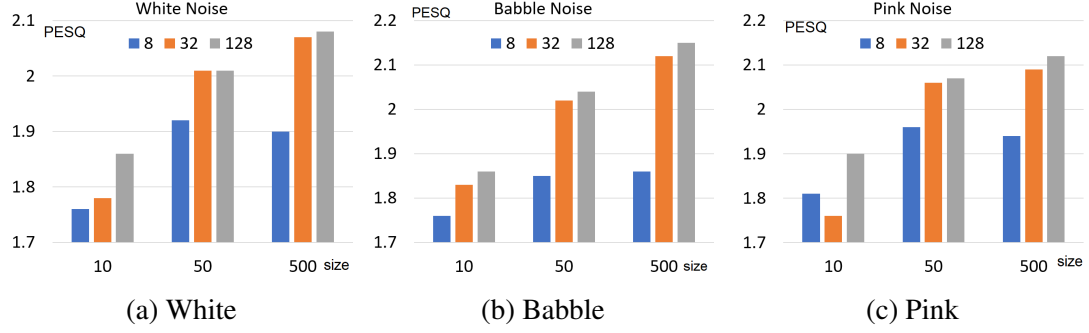


Figure 5.10: Size of codebook and dimension of codebook features

across all three noise types. K-means works well on low-dimensional vectors, but the reconstruction of 257-dimensional LPS from 8-dimensional latent code is difficult. Hence, the blue bars corresponding to using only eight codebook entries show the worst conversion in Figure 5.10. As the latent dimension grows to 32 or 128, the conversion quality improves, indicated by the higher gray and orange bars over blue bars in Figure 5.10. Hence, the benefit of better feature reconstruction outweighs the challenges of K-mean clustering for VQAE in high dimensional space.

Another trade-off to consider is the size of the codebook. With a small codebook, an encoder has an easier job fitting an acoustic feature to a code vector. However, the quantization error could be too large, resulting in unsatisfactory reconstruction. The results in Figure 5.10 suggests that a larger codebook size is more favorable than a smaller codebook. A codebook size of 50 or 500 outperforms that of size 10. It corresponds to a wide output layer of the encoder. A risk of using a wide softmax layer in the encoder is low activation in all dimensions without significant peaks. It could be rectified by adding a multiplier β larger than 1 to softmax. A sharper softmax of degree β is defined as

$$\xi(\beta \mathbf{v}) = \frac{\exp(\beta \mathbf{v})}{\sum_i \exp(\beta \mathbf{v})}. \quad (5.22)$$

The modification above allows us to use wide softmax layers, hence a large codebook for more faithful reconstruction. In summary, the above experiments show that VQAE can

be an effective technique to perform unsupervised speech conversion for indirect speech enhancement.

5.3 Dictionary-based indirect speech conversion and enhancement

In section 5.2, we encode noisy speech from the source domain into a latent representation, implicitly decomposed into speech and noise components. The target noise component replaces the source noise component to perform background noise replacement. By keeping the speech component unchanged, the reconstructed speech is expected to maintain the same speech content but in the target noise environment.

There are no constraints on the latent layer that enforce the disentanglement of speech from noise in a vanilla AE. Three techniques, namely noise-aware training, domain adversarial loss, and vector quantized auto-encoder, have been proposed in section 5.2 to impose some constraints on the latent layer to promote separation. This section will further develop a technique akin to VQAE by using an explicit codebook to represent speech and noise activation in a latent space.

5.3.1 Problem formulation

The additive noise model of speech introduced in Chapter 2 assumes that speech mixture is the sum of the speech subspace and the noise subspace. In the frequency domain, the speech subspace can be written as XW_X , where X is a codebook of speech bases, and W_X are the activation weights. Similarly, the noise subspace for noise type, A , is $D_AW_{D_A}$. The codebooks, X and D_A , are collections of basis vectors that span speech and noise acoustic spaces. The weights vectors W_X and W_{D_A} represent the activation in the latent space. Then the mixed noisy speech, Y , is the sum of these two subspaces

$$Y = \begin{bmatrix} X & D_A \end{bmatrix} \begin{bmatrix} W_X \\ W_{D_A} \end{bmatrix} \ni W_X, W_{D_A} \geq 0. \quad (5.23)$$

By constraining the weights, W_X and W_{D_A} , to be non-negative, we can use NMF to solve for the activation weights since the magnitude spectra are always non-negative. Solving Equation 5.23 for the weights, W_X and W_{D_A} , can be interpreted as encoding the noisy speech, Y , into latent activations. The codebooks are equivalent to encoder weights.

The converted speech, \hat{Y} , can be reconstructed as

$$\hat{Y} = \begin{bmatrix} X & D_B \end{bmatrix} \begin{bmatrix} W_X \\ W_{D_A} \end{bmatrix}. \quad (5.24)$$

To replace background noise type while keeping speech contents intact, we must keep the speech subspace unchanged. Hence in Equation 5.24, both X and W_X stay the same. We further assume that the noise activation is the same, so the same weights, W_{D_A} , is used. Only the noise basis is replaced by the target noise space, D_B , to convert the noise subspace. It is analogous to employing a decoder from the target noise domain when using AEs to perform noisy speech conversion.

As the codebook is usually over-complete to ensure as much acoustic variation could be captured as possible, the activation weights are usually constrained to be sparse. Let P be the combined codebook, $\begin{bmatrix} X & D_A \end{bmatrix}$, and W be the combined weights. The activation weights, W , are computed by minimizing the following loss, L_{NMF} [19]

$$L_{NMF} = KLD(Y, PW) + \lambda ||W||_1. \quad (5.25)$$

L1 norm is used in place of L0 norm since it is computationally tractable and promotes sparsity. It is shown in [172] that the weight vectors, W_X and W_{D_A} , can be computed iteratively as

$$W \leftarrow W \otimes \frac{P^T Y}{P^T W + \lambda}. \quad (5.26)$$

where \otimes and $-$ are element-wise product and division. At deployment, spectral features are converted according to Equation 5.24. The spectral features are then taken with logarithm

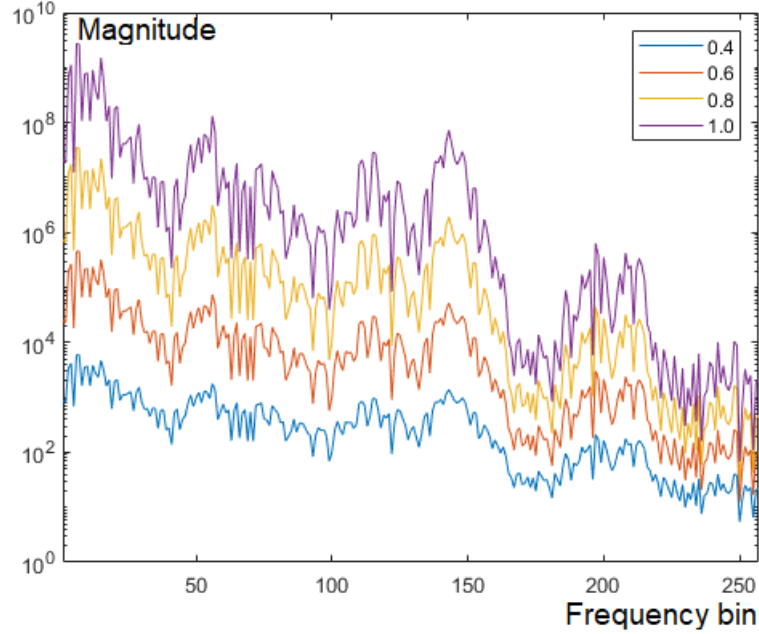


Figure 5.11: Dynamic range compression with exponentiation factor, ρ

to convert into LPS. It can then be normalized and enhanced by a downstream enhancer.

The raw magnitude features exhibit a large dynamic range between high-intensity and low-intensity frames. LPS features use the logarithm to compress the dynamic range but forgo non-negativeness. The large dynamic range may be problematic as exemplars of high intensity could easily overshadow those with less energy. In this application, we introduce a spectral exponentiation factor, ρ , to compress the range. The magnitude spectrum is compressed as Y^ρ . The effect of this exponentiation factor can be visualized in Figure 5.11. By selecting a value of ρ less than 1, high energy and low energy regions will be more comparable, hence have a more even contribution in reconstruction. To incorporate the exponentiation factor, Equation 5.23 and Equation 5.24 can be updated as follows

$$Y^\rho = \begin{bmatrix} X & D_A \end{bmatrix}^\rho \begin{bmatrix} W_X \\ W_{D_A} \end{bmatrix}, \quad (5.27)$$

and

$$\hat{Y} = \left(\begin{bmatrix} X & D_B \end{bmatrix}^\rho \begin{bmatrix} W_X \\ W_{D_A} \end{bmatrix} \right)^{1/\rho}. \quad (5.28)$$

5.3.2 Experiments and discussions

codebook construction

Three codebooks need to be constructed: the speech exemplars, X , the noise exemplars from the source domain, D_A , and the noise exemplars from the target domain D_B . The speech codebook, X , is a collection of feature frames sampled from a specific speaker in a clean environment. The target domain is an intermediate noise type selected according to the criteria in Chapter 3. Its exemplars, D_B , could be collected as a training step. The noise exemplars from the source domain, D_A , can be sampled from speech silence in a recording using a voice activity detector. The number of noise exemplars from the source and target domain must be the same. There are no other steps in the training phase of the converter. Codebooks and hyper-parameters, such as ρ , have been pre-defined. During the testing time, the activation weights are first randomly initialized. We perform iterative updates to minimizing the loss in Equation 5.25, according to Equation 5.26. Lastly, we achieve conversion to the target domain by multiplying the activation weights with the combined codebook in the target noise domain, according to Equation 5.28. The converted speech is used as an intermediate noisy speech to be further processed by a speech enhancer.

Spectral compression factor, ρ

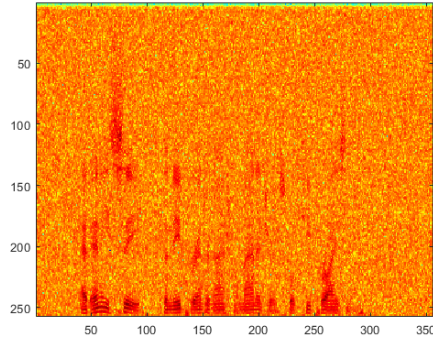
Figure 5.11 shows that the exponentiation of the spectral magnitude by $\rho < 1$ can reduce the dynamic range, hence allowing exemplars with low energy to be included. We examine how compression affects the conversion quality. PESQ scores with respect to the compression factor, ρ , in the range of 0.2 to 1.0, are tabulated in Table 5.7.

The default value of 1.0 means no spectral compression. It presents a significant im-

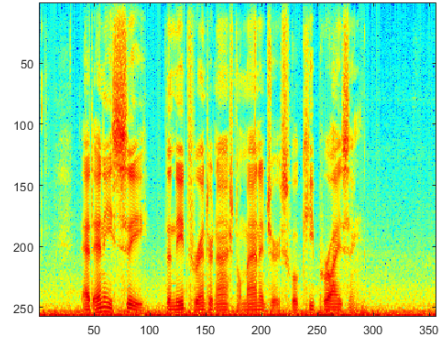
ρ	Direct	AE	NMF				
			1.0	0.8	0.6	0.4	0.2
white	1.48	2.09	2.23	2.28	2.34	2.30	1.00
babble	2.12	2.12	2.35	2.41	2.45	2.37	1.01
pink	1.70	2.15	2.27	2.17	2.27	2.29	0.97
average	1.77	2.12	2.28	2.29	2.35	2.32	0.98

Table 5.7: Effect of spectral compression in exemplar conversion

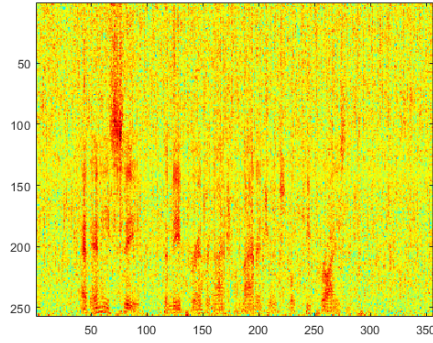
provement from direct enhancement. On average, PESQ improves from 1.70 to 2.27. It is also better than unsupervised conversion with an AE (2.15). A major reason for the better quality of NMF-based reconstruction over AE-based reconstruction is the use of speech exemplars in the feature domain and large codebook size, which significantly improves the quality of reconstructed speech. Evident from Figure 5.12, the source domain’s speech (*white* noise) is shown on the top left. Speech in *volvo* noise is the intermediate target. The oracle converted speech in *volvo* noise is shown on the top right for reference. The AE converted speech on the bottom left still has significant noise residue from the source domain because the encoder still encodes noise from the source domain. As a result, the converted speech still contains *white*-like background noise. The exemplar-based approach, shown on the bottom right, uses a completely different set of exemplars from the target domain for reconstruction. Consequently, the reconstructed speech will only be spanned by feature vectors in the domain of *volvo* noise. Its output will be much closer to the oracle output in Figure 5.12b. Furthermore, we also observe that dynamic range compression helps achieve better conversion. Table 5.7 shows that moderate compression at $\rho = 0.6$ achieves the overall best conversion quality. We compute the standard deviation of the activation weights of W in Equation 5.26. When there is no spectral compression, i.e., $\rho = 1.0$, the standard deviation of the activation is 0.0173, compared to 0.0132 for $\rho = 0.6$. It implies that the activation is less uniform at large values of ρ . Only a few high energy exemplars are chosen for reconstruction. A more uniform selection at low ρ allows more exemplars to be selected and creates a smoother reconstruction.



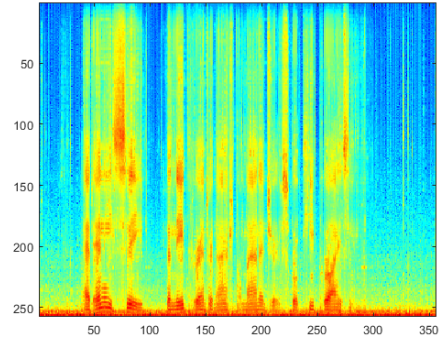
(a) Speech in the source domain



(b) Speech in the target domain



(c) AE-converted speech



(d) NMF-converted speech

Figure 5.12: Comparison between AE and NMF converted speech

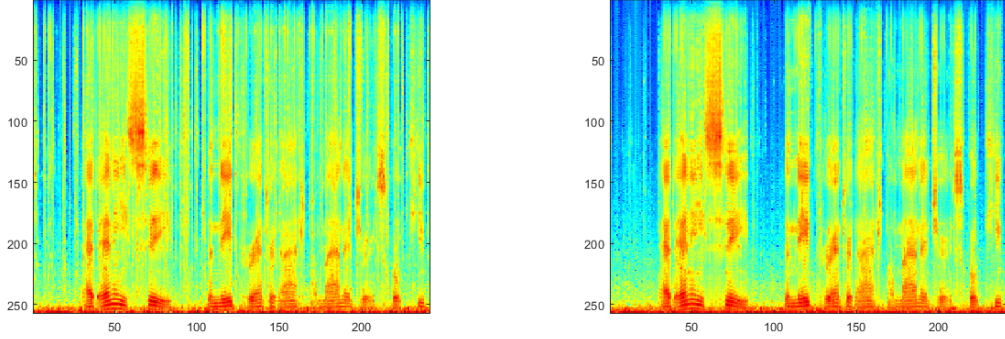
Exemplar dimensions

Since there is no compression or quantization in exemplar-based conversion, the codebook’s size must be much greater than the codebook used in AE-based conversion. In this section, we examine the effect of the size of the codebook of exemplars.

In the first experiment, we focus on the first three columns listed under NMF in Table 5.8. The combined codebook has a fixed size of 5000 entries, and we adjust the proportion of speech and noise exemplars. Intuitively, speech exhibits greater variability, requiring more dimensions in the latent space. Surprisingly, we find the opposite to be true. The first three columns under NMF shows that as speech dimensions decrease and noise dimensions increase, the conversion quality improves. We could understand this unexpected observation by comparing the converted speech using different compositions of speech and noise

		AE	NMF						
Speech	Direct	256	4000	2500	1000	500	1000	1000	1000
Noise			1000	2500	4000	2000	2000	4000	8000
white	1.48	2.09	2.17	2.34	2.55	2.38	2.42	2.55	2.17
babble	2.12	2.12	2.36	2.49	2.51	2.40	2.48	2.51	2.54
pink	1.70	2.15	1.98	2.27	2.48	2.34	2.31	2.48	2.56
average	1.77	2.12	2.17	2.37	2.51	2.37	2.40	2.51	2.42

Table 5.8: Effect of codebook size in NMF conversion



(a) Speech exemplar = 1k, noise exemplar = 4k (b) Speech exemplar = 4k, noise exemplar = 1k

Figure 5.13: Comparison between different codebook composition

exemplars. The converted speech using 4000 speech exemplars and 1000 noise exemplars are shown in Figure 5.13a. Compared to the result obtained using 1000 speech exemplars and 4000 noise exemplars in Figure 5.13b, the former spectrogram still contains much residue noise. The residue noise exist as vertical stripes during speech silence. A relatively large number of noise exemplars are required to decompose the background noise in the source domain for challenging noise conditions. A large number of noise exemplars also help the decoder render more natural background noise. On the other hand, speech components are unchanged in the conversion process. Hence, less resolution may be needed. As a result, the NMF-based speech transformation favors larger dimensions for noise exemplars.

In the second experiment, we attempt to investigate a reasonable range for the size of both the speech and noise codebook for optimal conversion quality. By comparing the fourth (500/2000) and the fifth (1000/2000) column under NMF in Table 5.8, we could tell

	Non-conversion				Conversion			
	babble	pink	white	average	babble	pink	white	average
-5	1.70	1.24	1.08	1.34	2.00	2.18	2.23	2.14
0	2.12	1.70	1.48	1.76	2.36	2.48	2.55	2.46
5	2.51	2.16	1.99	2.22	2.64	2.69	2.76	2.70
average	2.11	1.70	1.52	1.77	2.33	2.45	2.51	2.43

Table 5.9: NMF-based conversion on various SNR levels

that a reasonably large codebook size is required to represent speech well, as 500 speech exemplars are not rich enough to decompose the speech subspace. The last three columns in Table 5.8 show that an overly large codebook could also adversely affect the conversion quality. As the codebook grows, many similar exemplars are included in the codebook. These repeated or similar exemplars do not bring more improvement in conversion quality. On the other hand, it slows down the iterative optimization significantly. In our experiment, the configuration of 1000 speech exemplars and 4000 noise exemplars provides the best and most efficient conversion and subsequent enhancement quality.

SNR dependency

Lastly, we demonstrate the NMF-based conversion is effective at many SNR levels. Table 5.9 shows that the proposed conversion scheme achieved noticeable improvement at many SNR levels for all three noise types. It is also important to note the improvement is larger on *pink* and *white* noise than *babble* noise since the first two are more difficult with lower PESQ scores.

5.4 Summary

This chapter focuses on indirect enhancement by converting noisy speech from a source noise domain to a target noise domain without explicit mapping targets used in Chapter 4. This situation will be handy when noisy-clean speech pair is not available. The principle behind unsupervised speech conversion is to decompose noisy speech into speech and

noise subspaces, known as representational learning. We applied two classes of techniques to solve this problem, auto-encoders, and matrix decomposition. With AEs, we find latent representations of speech and noise in a bottleneck layer. With NMF, we construct weight matrices representing the activity of speech or noise exemplars. In either case, the speech and noise subspaces are assumed to be separable in the latent space. We could then replace the noise subspace of the source noise with that of the target noise to accomplish speech conversion and indirect speech enhancement. Various techniques designed to promote greater disentanglement of speech and noise subspace are discussed, including noise-aware training, domain-adversarial training, and vector-quantized training. Conversion based on these AE models has been shown to improve the overall enhancement quality for the noise types investigated. Several factors that affect the performance, including width, depth, and SNR levels, are also discussed. The AE models cannot completely disentangle the speech and noise subspace. They leave considerable noise residue in converted speech and degrade the enhancement quality. NMF addresses this concern by explicitly using different noise bases for source and target domains. Consequently, it achieves conversion with much less residue noise. A downside of NMF-based conversion is its long latency due to its iterative optimization. It is less ideal for applications such as online adaptation. A possible solution could incorporate exemplars in VQAE to achieve both good conversions with high-resolution exemplars and fast inference in neural networks.

CHAPTER 6

CONCLUSIONS

6.1 Summary of research

This thesis proposes an indirect approach to speech enhancement by leveraging upon the framework of curriculum learning [89] described in Chapter 2. Conventional DNN-based enhancement systems [9] trained using data-driven techniques generally do not distinguish different noise environments. As a result, the performance is unsatisfactory in adverse acoustic conditions with mismatched noise at low SNR. We recognize the difficulty in enhancing such noisy speech directly and propose to divide the process into simpler sub-tasks. In the indirect set-up, we first transform noisy speech features into speech in another background noise that is easier to be processed. This step does not require all background noise to be removed at once. The residue noise will be eliminated in subsequent refinement stages. Since each stage is only responsible for partially enhancing the speech, the sub-tasks can be designed with greater flexibility to address the issues at every step. We empirically demonstrate that the indirect method yields substantial performance gains over direct methods in traditionally adverse acoustic environments for each of the proposed methods.

6.2 Contributions

6.2.1 Noise characterization

We introduce the indirect approach to speech enhancement in section 2.3. Compared to direct approaches, our indirect approach offers more benefits in challenging noise conditions due to simpler sub-tasks. It requires us to first identify applicable scenarios with adverse noise conditions. Next, we could select an intermediate speech target after calibrating the

difficulty levels of noise types. Our first contribution is the clustering and classification of additive noise into simple or difficult noise types in the context of enhancement with empirical validation. Consistent with previous studies, we confirm that wideband noise is more difficult to enhance than narrowband noise. However, we also discover that non-stationarity does not pose a significant challenge since dynamic noise can be approximated with pseudo-stationary noise at a frame level, given the frame rate is relatively high. This observation also corroborates the advantage of DNN-based speech enhancement over traditional statistical methods in handling non-stationary noise. We identify that the long term average PSD of noise to be a reasonable indicator of the difficulty of additive noise. In general, simple noise can be masked by the average speech spectrum. On the other hand, difficult noise has high energy at spectral valleys. We also show that difficult noise adversely affects speech enhancement is improper normalization because of its mismatched feature statistics. Enhancement experiments are conducted on simulated noisy speech in various noise conditions to evaluate how the aforementioned factors affect enhancement quality.

6.2.2 Indirect enhancement via supervised learning

After identifying the intermediate stages, we proposed several techniques of speech transformation. The first proposed technique transforms difficult noisy speech by normalizing its feature statistics to an easier noise type. It is motivated by the observation that speech features in more challenging noise conditions follow a different distribution from clean speech. This difference also translates to different activations in hidden and output layers of a DNN. Feature normalization and histogram matching could reduce such mismatches. We conducted experiments to show that it is an effective technique to handle very difficult noise. For moderately challenging noise, the improvement is not as noticeable.

When parallel training pairs can be synthesized or recorded, we may perform mapping on a frame-level to achieve speech transformation with greater effectiveness. We use DNNs

to map speech features to a target domain frame by frame while minimizing the MSE loss. This supervised learning approach can mostly alleviate the problem of domain mismatch. Experimental findings confirm that it is also effective for moderately challenging noisy conditions, as it reduces speech distortion in the conversion process. We further extend this technique to handle interferences from several sources. When multiple noise sources are present, the indirect approach can be leveraged to remove the disturbance progressively. In this multi-stage set-up, the noise mixture is considered a difficult noise condition, as it possesses complex temporal and frequency structures. After each stage, one noise type is removed from the noisy speech. We remove the noise sources progressively, until the clean speech is recovered. By comparing and evaluating the operating order, we determine that it is generally better to remove the more challenging noise upfront. Such knowledge would help speech engineers develop the speech enhancement pipeline in practical situations.

6.2.3 Indirect enhancement via representational learning

This thesis’s third contribution includes various speech transformation techniques without requiring parallel data to train speech converters. It is achieved by discovering underlying structures in noisy speech features in a latent space. Even though speech and noise acoustic features are not linearly separable, we take advantage of auto-encoders and dictionary learning to transform speech features into a latent space where they become separable. We could then replace the noise sub-space from the original domain with that from a target domain while keeping the speech sub-space unchanged. Subsequently we synthesize converted speech by combining the speech sub-space and the target noise sub-space. We explore latent structures using auto-encoders and dictionary-based learning. With auto-encoders, source encoders transform input speech features into latent vectors. We impose constraints on the latent space to promote greater separability of the speech and noise. With the dictionary-based method, we utilize NMF to determine the activation weights of a set of over-complete speech and noise exemplars. Speech conversion could then be conducted

by changing the set of noise bases. We evaluate the validity of the proposed techniques with a series of experiments on simulated data. We observe that the dictionary-based methods can convert noisy speech with greater fidelity because of the use of over-complete exemplars. However, its iterative procedure may make it unsuitable for some online or resource-constraint applications.

6.3 Future work

This thesis attempts to develop a noise-aware strategy in deep learning-based speech enhancement models. By recognizing that the acoustic environment is challenging, we resort to the proposed indirect approaches to decompose speech enhancement into multiple stages. Unlike SNR, which has a clear relationship with the enhancement difficulty, noise types do not possess such natural interpretation and have not received widespread study. As an initial step, much analysis and validation are performed empirically. To this end, some potential directions for future investigation can be suggested.

6.3.1 Theoretical characterization of noise types

We identified several factors that make some noise environments more difficult to enhance than the rest in Chapter 3 using an experimental approach. However, the acoustic conditions can be very diverse, and it is impractical to enumerate and archive all of them. Hence, it is desirable to develop a more theoretical understanding of the interaction of noise and speech in the context of speech enhancement. Our attempts in subsection 4.2.1 is a starting point to explain how difficult noise affect feature normalization, but we expect the interaction between noise and speech goes beyond the input space. As the community gradually gains more insights into the inner workings of deep models, it is reasonable to study how deep models treat the noise types differently in the context of speech enhancement.

6.3.2 Noisy speech with multiple sources

This thesis only addresses the interference of additive noise in a single channel. In reality, disturbance in speech could occur due to echo, reverberation, competing speakers on top of additive noise. In some commercial smart speakers, the audio is generally processed in multiple stages [173], where each stage only handles one aspect of the interferences. However, there is still room to explore which interference should be processed first. It would be useful for future researchers to relate the degree of degradation of each interference to the order of its removal in the enhancement process.

6.3.3 Disentanglement of latent feature

Disentanglement of the latent features helps us achieve unsupervised speech transformation in this application, and helps speech researchers understand structures of speech in deep learning in general. By developing methods to separate speech features by some desirable traits, such as phonemes, gender, speaker, tone, and emotion, we can apply this technique to many other speech-related tasks, including speech recognition, speaker identification, and emotion classification.

6.3.4 Explorations of different deep architectures for speech transformation

Deep model-based speech enhancement has received much attention in the speech community in the last decade. More advanced models and features have been proposed in the literature, such as the use of complex ratio mask to include phase prediction [174], raw waveform enhancement [175], and models combining beamforming techniques [176]. Since few of these models focus on noise-aware training, we expect the proposed indirect approach to benefit these advanced models. It is nevertheless non-trivial to scale the proposed work to multi-channel or complex models. It warrants further investigations to extend our proposed framework to these recent developments.

Appendices

APPENDIX A

DERIVATION OF MEAN DEVIATION IN THE NORMALIZATION OF LPS

FEATURE

The mean deviation, Δ_μ , is defined in Equation 4.13. Substitute the definition of μ_{LPS} in Equation 4.11 into Equation 4.13

$$\begin{aligned}\Delta_\mu &= \mathbb{E}[\log Y_m^2] - \mathbb{E}[\log X_m^2] \\ &= \mathbb{E}[W(\xi_m, \phi_{XD})].\end{aligned}\tag{A.1}$$

When SNR is high, i.e., $\xi_m \rightarrow \infty$, we can further show that the first order Taylor series expansion of $W(\xi_m, \phi_{XD})$ is

$$W(\xi_m, \phi_{XD}) \approx \frac{2 \cos \phi_{XD}}{\xi_m}.\tag{A.2}$$

Hence,

$$\mathbb{E}[W(\xi_m, \phi_{XD})] = \mathbb{E}\left[\frac{2 \cos \phi_{XD}}{\xi_m}\right] \approx 0.\tag{A.3}$$

When SNR is low, i.e., $\xi_m \rightarrow 0$,

$$W(\xi_m, \phi_{XD}) \approx -2 \log(\xi_m) + 2 \xi_m \cos \phi_{XD}.\tag{A.4}$$

Furthermore, we assume the phase difference follows a uniform distribution, i.e., $\phi_{XD} \sim \mathcal{U}(-\pi, \pi]$. Hence for low SNR,

$$\mathbb{E}[W(\xi_m, \phi_{XD})] = \mathbb{E}[-2 \log(\xi_m) + 2 \xi_m \cos \phi_{XD}] \approx -2 \mathbb{E}[\log(\xi_m)].\tag{A.5}$$

APPENDIX B

DERIVATION OF THE VARIANCE DEVIATION IN THE NORMALIZATION OF

LPS FEATURE

The variance deviation, ϕ_{σ^2} , is defined in Equation 4.15. We attempt to simplify it as

$$\begin{aligned}
\phi_{\sigma^2} &= \text{Var}(\log Y_m^2) - \text{Var}(\log X_m^2) \\
&= \text{Cov}\left(\log Y_m^2 - \log X_m^2, \log Y_m^2 + \log X_m^2\right) \\
&= \text{Cov}\left(W(\xi_m, \phi), W(\xi_m, \phi) + 2 \log X_m^2\right) \\
&= 2\text{Cov}\left(\log X_m^2, W(\xi_m, \phi)\right) + \text{Var}(W(\xi_m, \phi)) \\
&= 2\mathbb{E}\left[\log X_m^2 W(\xi_m, \phi)\right] - 2\mathbb{E}\left[\log X_m^2\right]\mathbb{E}\left[W(\xi_m, \phi)\right] \\
&\quad \dots + \mathbb{E}[W(\xi_m, \phi)^2] - (\mathbb{E}[W(\xi_m, \phi)])^2,
\end{aligned} \tag{B.1}$$

where $\phi = \angle X(m, k) - \angle D(m, k)$ represents the phase difference. The results in Equation B.1 can be simplified depending on if ξ_m is high or low.

High SNR ($\xi_m \rightarrow \infty$)

When the SNR is high, we know that $\mathbb{E}[W(\xi_m, \phi)] \approx \mathbb{E}\left[\frac{2 \cos \phi \xi_m}{\xi_m}\right] \approx 0$ from Equation A.3.

We assume that $\log X_m^2$ and ϕ are independent. This allows us to simplify the first term in Equation B.1:

$$2\mathbb{E}\left[\log X_m^2 W(\xi_m, \phi)\right] \approx 2\mathbb{E}\left[\log X_m^2 \frac{2 \cos \phi}{\xi_m}\right] \tag{B.2}$$

$$= 4\mathbb{E}\left[\cos \phi\right]\mathbb{E}\left[\frac{\log X_m^2}{\xi_m}\right] \tag{B.3}$$

$$= 0 \tag{B.4}$$

The second term, $-2\mathbb{E}[\log X_m^2]\mathbb{E}[W(\xi_m, \phi)]$, and the last term, $(\mathbb{E}[W(\xi_m, \phi)])^2$ are 0 too, because we can make the substitution in Equation A.3.

For the third term,

$$\begin{aligned}
\mathbb{E}[W(\xi_m, \phi)^2] &= \int_0^\infty \int_{-\pi}^\pi f(\xi_m, \phi) W(\xi_m, \phi)^2 d\phi d\xi \\
&\approx \int_0^\infty \int_{-\pi}^\pi f(\xi_m, \phi) \left(\frac{2 \cos \phi}{\xi} \right)^2 d\phi d\xi \\
&= \int_0^\infty f(\xi) \left(\frac{1}{2\pi} \int_{-\pi}^\pi \frac{4 \cos^2 \phi}{\xi^2} d\phi \right) d\xi \\
&= \mathbb{E} \left[\frac{2}{\xi^2} \right].
\end{aligned} \tag{B.5}$$

Substituting these results back to Equation B.1, we can find that

$$\Delta_{\sigma^2} \approx 0 \tag{B.6}$$

when most ξ_m are high SNRs.

Low SNR ($\xi_m \rightarrow 0$)

When most ξ_m are in low SNR, $W(\xi_m, \phi) \approx -2 \log \xi_m = \log D_m^2 - \log X_m^2$ by Equation A.5. We can simplify $\mathbb{E}[W(\xi_m, \phi)^2]$ by making the substitution above,

$$\begin{aligned}
\mathbb{E}[W(\xi_m, \phi)^2] &= \int_0^\infty \int_{-\pi}^\pi f(\xi_m, \phi) W(\xi_m, \phi)^2 d\phi d\xi_m \\
&\approx \int_0^\infty \int_{-\pi}^\pi f(\xi_m, \phi) \left(-2 \log \xi_m + 2 \xi_m \cos \phi \right)^2 d\phi d\xi \\
&= \int_0^\infty f(\xi) \left(\frac{1}{2\pi} \int_{-\pi}^\pi 4 \log^2 \xi_m - 8 \xi_m \log \xi_m \cos \phi + 4 \xi_m^2 \cos^2 \phi d\phi \right) d\xi_m \\
&= \mathbb{E} \left[-4 \log^2 \xi_m + 2 \xi^2 \right]
\end{aligned} \tag{B.7}$$

We can then simplify Equation B.1 when SNR is low

$$\begin{aligned}
\Delta_{\sigma^2} &= 2\mathbb{E}\left[\log X_m^2 W(\xi_m, \phi)\right] - 2\mathbb{E}\left[\log X_m^2\right]\mathbb{E}\left[W(\xi_m, \phi)\right] \\
&\quad \dots + \mathbb{E}[W(\xi_m, \phi)^2] - \left(\mathbb{E}[W(\xi_m, \phi)]\right)^2 \\
&\approx 2\mathbb{E}\left[\log X_m^2 (\log D_m^2 - \log X_m^2)\right] - 2\mathbb{E}\left[\log X_m^2\right]\mathbb{E}\left[\log D_m^2 - \log X_m^2\right] \\
&\quad - 4\mathbb{E}\left[\log^2(X_m/D_m)\right] + 2\mathbb{E}\left[\frac{X_m^2}{D_m^2}\right] - \mathbb{E}^2\left[\log D_m^2 - \log X_m^2\right] \tag{B.8} \\
&= \mathbb{E}[\log^2 D_m^2] - \mathbb{E}^2[\log D_m^2] - \left(\mathbb{E}[\log^2 X_m^2] - \mathbb{E}^2[\log X_m^2]\right) + 2\mathbb{E}[\xi^2] \\
&= \text{Var}(\log D_m^2) - \text{Var}(\log X_m^2) + 2\mathbb{E}[\xi^2] \\
&\approx \text{Var}(\log D_m^2) - \text{Var}(\log X_m^2).
\end{aligned}$$

If many TF bins have low SNR, $2\mathbb{E}[\xi^2] \approx 0$. The deviation in variance is approximated as the difference in the variance of the noise spectrum and that of the speech spectrum.

APPENDIX C

DEFINITION OF COLORED NOISE

The naming convention of noise originated from white noise, which has a flat power spectrum in linear frequency axis. It is called white as an analogy to the white light which is assumed to have a flat power spectrum of the electromagnetic waves in the visible light range. The other colors are named to loosely reflect a similarity with the visible light spectrum of the corresponding color. In other words, the spectrum of a pink noise would translate into pink light if the audio frequency axis were to change into electromagnetic frequency axis of appropriate frequency ranges. Their spectrogram and PSDs are included at the end of this appendix.

C.1 White noise

White noise has a flat spectrum over linear frequency in Hz. Consequently, the signal has uniform PSD in the linear frequency axis. Digital white noise can be generated by randomly and independently selecting samples.

C.2 Gray noise

Gray noise is a perceptually weighted white noise as human's hearing is not linear due to biases in equal loudness contour. Hence, the spectrum in each frequency range is modified to give the listener the perception of equal loudness across all frequencies.

C.3 Pink noise

Pink noise has a PSD inversely proportional to frequency. Its power density falls off at 10dB per decade or 3dB per octave [177]. It is also known as $1/f$ noise as its PSD, $S(f)$,

follows

$$S(f) \propto \frac{1}{f}. \quad (\text{C.1})$$

It is commonly detected in flicker noise in electronics, astronomical physics and neurobiology [178].

C.4 Red noise

Red noise has a PSD inversely proportional to the square of frequency. Its power density falls off faster than that of pink noise at 6dB per octave

$$S(f) \propto \frac{1}{f^2}. \quad (\text{C.2})$$

Red noise can be generated with temporal integration of white noise. It is also known as Brownian noise, as it is the type of noise generated in a Brownian motion or random walk.

C.5 Blue noise

Blue noise, a.k.a. azure noise, can be considered as the complement of pink noise. Its PSD rolls up 3dB per octave as its PSD is proportional to frequency, provided the frequency range is finite

$$S(f) \propto f. \quad (\text{C.3})$$

It has been observed in Cherenkov radiation and used in dithering.

C.6 Purple noise

Purple noise, a.k.a. violet noise, is the counterpart of red noise. Its PSD rolls up 6dB per octave since its PSD is proportional to f^2

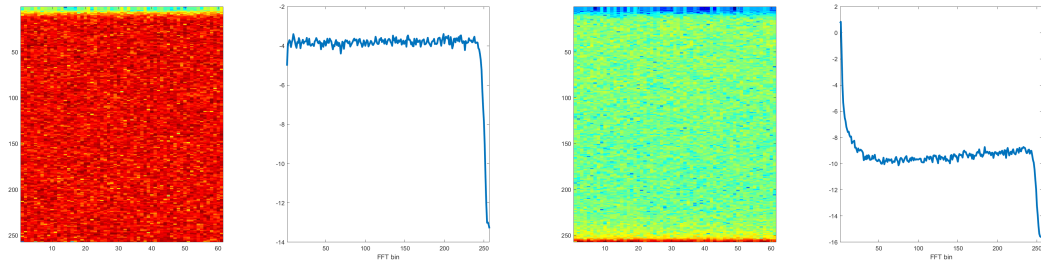
$$S(f) \propto f^2. \quad (\text{C.4})$$

Just as red noise can be generated from integration of white noise, purple noise can be obtained by differentiating the white noise. It has been observed in acoustic thermal noise of ocean water [179] and applied to dithering in digital audio.

C.7 Black noise

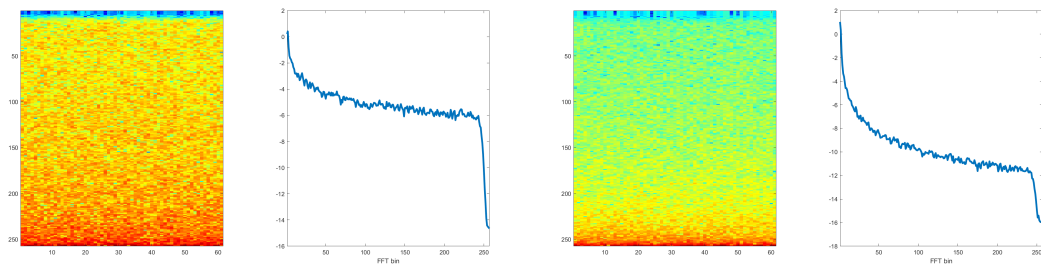
Sometimes it is used to denote the absence of any frequency, hence black.

Figure C.1: Spectrograms and PSDs of some colored noise



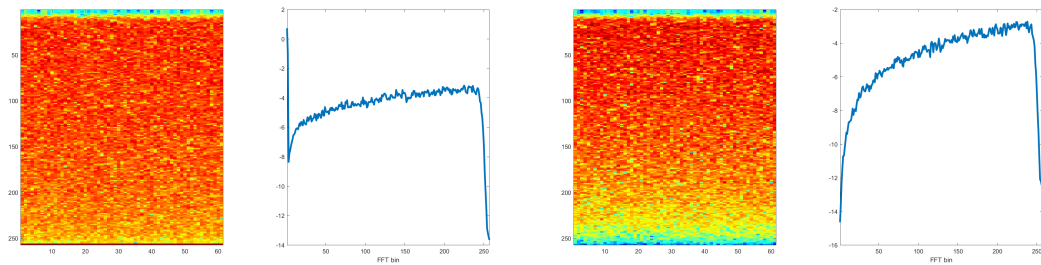
(a) White

(b) Gray



(c) Pink

(d) Red



(e) Blue

(f) Purple

APPENDIX D

DESCRIPTION OF NONSPEECH NOISE

The noise types in the Nonspeech were collected by Hu and Wang in [154]. It includes 100 types of commonly seen noise. They are classified into the following categories by their noise name from n001 to n100.

- n001-n017: Crowd noise
- n018-n029: Machine noise
- n030-n043: Alarm and siren
- n044-n046: Traffic and car noise
- n047-n055: Animal sound
- n056-n069: Water sound
- n070-n078: Wind
- n079-n082: Bell
- n083-n085: Cough
- n086: Clap
- n087: Snore
- n088: Click
- n088-n090: Laugh
- n091-n092: Yawn
- n093: Cry
- n094: Shower
- n095: Tooth brushing
- n096-n097: Footsteps
- n098: Door moving
- n099-n100: Phone dialing

APPENDIX E

DESCRIPTION OF NOISEX92 NOISE

This page catalogues the Noisex92 noise used in the study. They were typically noise measured in field by the speech research unit at Institute for Perception-TNO, Netherlands, United Kindom in Feb., 1990. Except high frequency channel, white, and pink, other audio was recorded by 1/2" B&K condensor microphone on digital audio tapes with anti-aliasing filter but no pre-emphasis at a sampling rate of 19.98 kHz with a bit depth of 16 bits [180]. The spectrograms and PSD of each noise type can be viewed at the end.

E.1 Babble

The source of this babble was 100 people talking in a canteen environment. The room is over 2m wide. Individual voices is barely intelligible. The sound level was 88 dBA.

E.2 Buccaneer1,Buccaneer2

They are also referred to as cockpit noise in some other literature. Buccaneer1 was recorded when a Buccaneer jet was traveling at 190 knots at an altitude of 1000 ft without airbrakes. The sound level was 109 dBA. Buccaneer2 was recorded when the jet was traveling at 450 knots at 300 ft. The sound level was 116 dBA.

E.3 Destroyer Engine Room, Destroyer Operation Room

They were recorded on a destroyer. The sound level was 101 dBA and 70 dBA respectively.

E.4 F16

It is another noise recorded in the cockpit of a fighter jet. The microphone was placed at the co-pilot's seat in a two-seat F-16 traveling at a speed of 500 knots at an altitude between 300 and 600 ft. The sound level was 103 dBA.

E.5 Factory1, Factory2

Factory1 was recorded near plate-cutting and electrical welding equipment and Factory2 was recorded in a car production hall. Factory2 has a narrow energy band.

E.6 Hfchan

The noise is extracted from a high frequency radio channel after demodulation.

E.7 Leopard, M109

The Leopard noise was created by a Canadian Leopard 1 tank moving at 70 mph. The sound level was 114dBA at recording. M109 was another tank noise. An M109 self-propelled howitzer traveling at 19 mph was recorded. The sound level was 100dBA.

E.8 Machinegun

It was a non-stationary burst noise from firing repeatedly from a 0.5 calibre machine gun.

E.9 Pink

The definition follows the same in Appendix A. The recording was acquired by sampling a high-quality analog noise generator.

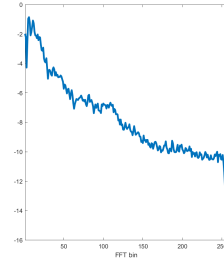
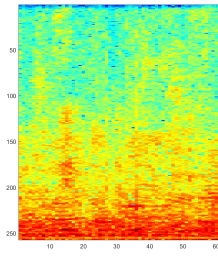
E.10 Volvo

It was narrowband signal. The sound of a Volvo 340 automobile traveling at 70 mph in the fourth gear on a tarmacked road in rainy weather was recorded.

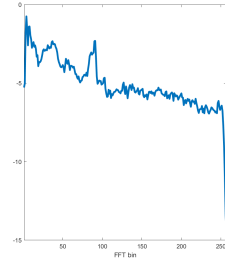
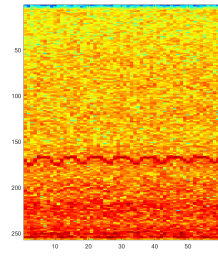
E.11 White

The definition follows the same in Appendix A. The recording was acquired by sampling a high-quality analog noise generator.

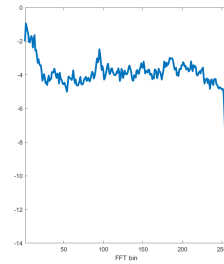
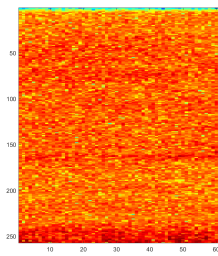
Figure E.1: Spectrograms and PSDs of Noisex92 noise



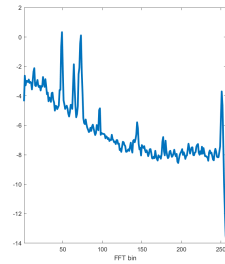
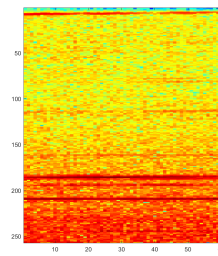
(a) Babble



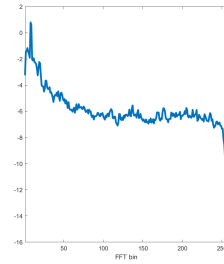
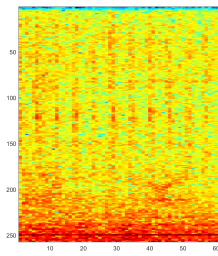
(b) Buccaneer1



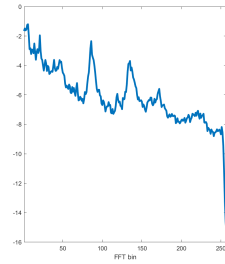
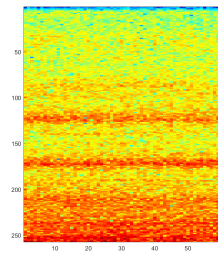
(c) Buccaneer2



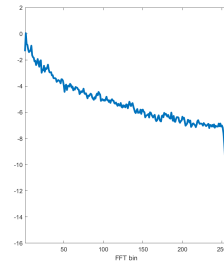
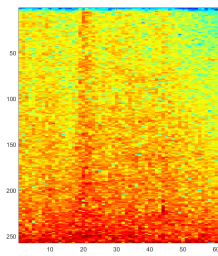
(d) Destroyer engine



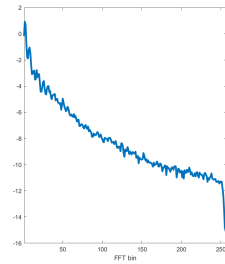
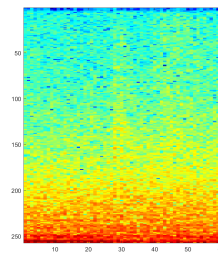
(e) Destroyer room



(f) F16

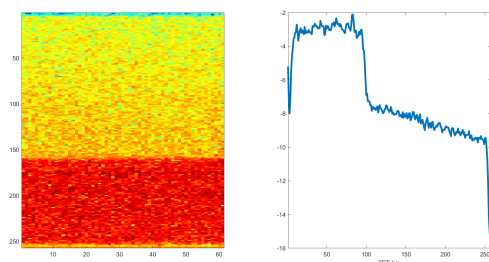


(g) Factory1

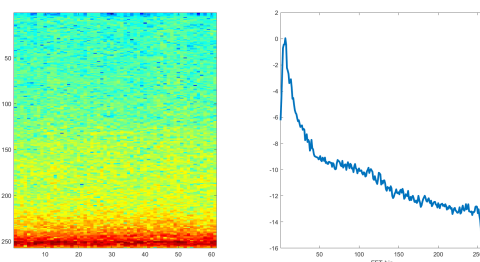


(h) Factory2

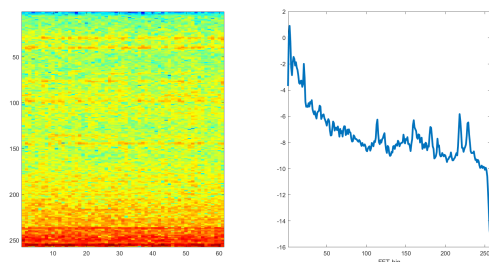
Figure E.1: Spectrograms and PSDs of Noisex92 noise (cont.)



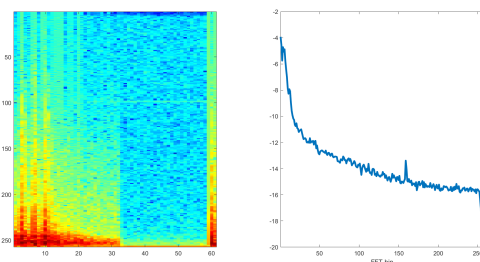
(i) High frequency channel



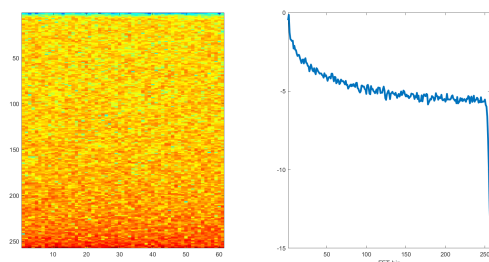
(j) Leopard tank



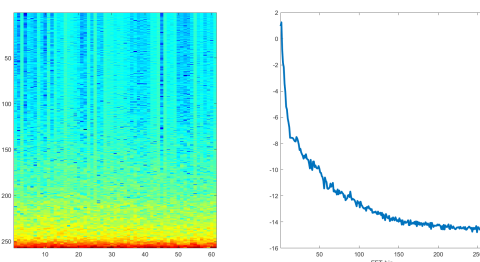
(k) M109 tank



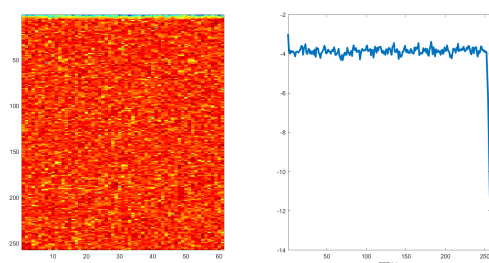
(l) Machinegun



(m) Pink



(n) Volvo



(o) White

REFERENCES

- [1] L. R. Rabiner, “Digital processing of speech signal,” *Digital Process. of Speech Signal*, 1978.
- [2] M. Berouti *et al.*, “Enhancement of speech corrupted by acoustic noise,” in *ICASSP*, IEEE, vol. 4, 1979, pp. 208–211.
- [3] S. Boll, “Suppression of acoustic noise in speech using spectral subtraction,” *TASLP*, vol. 27, no. 2, pp. 113–120, 1979.
- [4] N. Wiener, *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. MIT Press, 1950.
- [5] J. S. Lim and A. V. Oppenheim, “Enhancement and bandwidth compression of noisy speech,” *Proc. of the IEEE*, vol. 67, no. 12, pp. 1586–1604, 1979.
- [6] R. Frazier *et al.*, “Enhancement of speech by adaptive filtering,” in *ICASSP*, IEEE, vol. 1, 1976, pp. 251–253.
- [7] D. O’Shaughnessy, “Linear predictive coding,” *IEEE Potentials*, vol. 7, no. 1, pp. 29–32, 1988.
- [8] I. Cohen and B. Berdugo, “Speech enhancement for non-stationary noise environments,” *Signal process.*, vol. 81, no. 11, pp. 2403–2418, 2001.
- [9] Y. Xu *et al.*, “An experimental study on speech enhancement based on deep neural networks,” *IEEE Signal Process. Letters*, vol. 21, no. 1, pp. 65–68, 2013.
- [10] —, “A regression approach to speech enhancement based on deep neural networks,” *TASLP*, vol. 23, no. 1, pp. 7–19, 2015.
- [11] S.-W. Fu *et al.*, “Raw waveform-based speech enhancement by fully convolutional networks,” in *APSIPA*, IEEE, 2017, pp. 006–012.
- [12] Y. Ephraim and D. Malah, “Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator,” *TASLP*, vol. 32, no. 6, pp. 1109–1121, 1984.
- [13] S. Wang *et al.*, “A cross-task transfer learning approach to adapting deep speech enhancement models to unseen background noise using paired senone classifiers,” in *ICASSP*, IEEE, 2020, pp. 6219–6223.

- [14] T. Gao *et al.*, “Snr-based progressive learning of deep neural network for speech enhancement,” in *Interspeech*, 2016, pp. 3713–3717.
- [15] Y.-H. Tu *et al.*, “A multi-target snr-progressive learning approach to regression based speech enhancement,” *TASLP*, 2020.
- [16] P. C. Loizou, *Speech enhancement: theory and practice*. CRC press, 2013.
- [17] M. Abe *et al.*, “Voice conversion through vector quantization,” *Journal of the Acoustical Society of Japan (E)*, vol. 11, no. 2, pp. 71–76, 1990.
- [18] T. Toda *et al.*, “Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory,” *TASLP*, vol. 15, no. 8, pp. 2222–2235, 2007.
- [19] Z. Wu *et al.*, “Exemplar-based voice conversion using non-negative spectrogram deconvolution,” in *8th ISCA Workshop on Speech Synthesis*, 2013.
- [20] A. J. Hunt and A. W. Black, “Unit selection in a concatenative speech synthesis system using a large speech database,” in *ICASSP*, IEEE, vol. 1, 1996, pp. 373–376.
- [21] D. Erro *et al.*, “Voice conversion based on weighted frequency warping,” *TASLP*, vol. 18, no. 5, pp. 922–931, 2009.
- [22] J. B. Allen and L. R. Rabiner, “A unified approach to short-time fourier analysis and synthesis,” *Proc. of the IEEE*, vol. 65, no. 11, pp. 1558–1564, 1977.
- [23] R. W. Hamming, *Digital filters*. Courier Corporation, 1998.
- [24] A. V. Oppenheim, *Discrete-time signal processing*. Pearson Education India, 1999.
- [25] K. K. Paliwal and L. D. Alsteris, “On the usefulness of stft phase spectrum in human listening tests,” *Speech commun.*, vol. 45, no. 2, pp. 153–170, 2005.
- [26] D. Wang and J. Lim, “The unimportance of phase in speech enhancement,” *TASLP*, vol. 30, no. 4, pp. 679–681, 1982.
- [27] R. Crochiere, “A weighted overlap-add method of short-time fourier analysis/synthesis,” *TASLP*, vol. 28, no. 1, pp. 99–102, 1980.
- [28] P. C. Loizou, “Speech quality assessment,” in *Multimedia analysis, processing and communications*, Springer, 2011, pp. 623–654.
- [29] K. Li and C.-H. Lee, “A deep neural network approach to speech bandwidth expansion,” in *ICASSP*, IEEE, 2015, pp. 4395–4399.

- [30] D. Klatt, "Prediction of perceived phonetic distance from critical-band spectra: A first step," in *ICASSP*, IEEE, vol. 7, 1982, pp. 1278–1281.
- [31] J. H. Hansen and B. L. Pellom, "An effective quality evaluation protocol for speech enhancement algorithms," in *5th Int. Conf. on Spoken Language Process.*, 1998.
- [32] A. W. Rix *et al.*, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *ICASSP*, IEEE, vol. 2, 2001, pp. 749–752.
- [33] Y. Hu and P. C. Loizou, "Evaluation of objective quality measures for speech enhancement," *TASLP*, vol. 16, no. 1, pp. 229–238, 2007.
- [34] M. R. Weiss *et al.*, "Study and development of the intel technique for improving speech intelligibility," Nicolet Scientific Corp Northvale NJ, Tech. Rep., 1975.
- [35] R. McAulay and M. Malpass, "Speech enhancement using a soft-decision noise suppression filter," *TASLP*, vol. 28, no. 2, pp. 137–145, 1980.
- [36] M. Dendrinou, S. Bakamidis, and G. Carayannis, "Speech enhancement from noise: A regenerative approach," *Speech Commun.*, vol. 10, no. 1, pp. 45–57, 1991.
- [37] Y. Ephraim and H. L. Van Trees, "A signal subspace approach for speech enhancement," *TASLP*, vol. 3, no. 4, pp. 251–266, 1995.
- [38] S. Tamura and A. Waibel, "Noise reduction using connectionist models," in *ICASSP*, 1988, pp. 553–554.
- [39] S. Tamura, "An analysis of a noise reduction neural network," in *ICASSP*, IEEE, 1989, pp. 2001–2004.
- [40] S. Roweis, "One microphone source separation," *Advances in Neural Inf. Process. Syst.*, vol. 13, pp. 793–799, 2000.
- [41] G.-J. Jang and T.-W. Lee, "A maximum likelihood approach to single-channel source separation," *J. of Mach. Learning Research*, vol. 4, pp. 1365–1392, 2003.
- [42] M. L. Seltzer *et al.*, "A bayesian classifier for spectrographic mask estimation for missing feature speech recognition," *Speech Commun.*, vol. 43, no. 4, pp. 379–393, 2004.
- [43] K. W. Wilson *et al.*, "Speech denoising using nonnegative matrix factorization with priors," in *ICASSP*, IEEE, 2008, pp. 4029–4032.

- [44] K. Han and D. Wang, "A classification based approach to speech segregation," *J. of the Acoust. Soc. of America*, vol. 132, no. 5, pp. 3475–3483, 2012.
- [45] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *TASLP*, vol. 21, no. 7, pp. 1381–1390, 2013.
- [46] P. Lockwood and J. Boudy, "Experiments with a nonlinear spectral subtractor (nss), hidden markov models and the projection, for robust speech recognition in cars," *Speech commun.*, vol. 11, no. 2-3, pp. 215–228, 1992.
- [47] J. Lim, "Evaluation of a correlation subtraction method for enhancing speech degraded by additive white noise," *IEEE TASLP*, vol. 26, no. 5, pp. 471–472, 1978.
- [48] Y. Hu and P. C. Loizou, "Subjective comparison of speech enhancement algorithms," in *ICASSP*, IEEE, vol. 1, 2006, pp. I–I.
- [49] J. Benesty *et al.*, *Springer Handbook of Speech Processing*. Springer, 2007.
- [50] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *TASLP*, vol. 33, no. 2, pp. 443–445, 1985.
- [51] Y. Hu and P. C. Loizou, "A comparative intelligibility study of single-microphone noise reduction algorithms," *J. of the Acoust. Soc. of America*, vol. 122, no. 3, pp. 1777–1786, 2007.
- [52] O. Cappé, "Elimination of the musical noise phenomenon with the ephraim and malah noise suppressor," *TASLP*, vol. 2, no. 2, pp. 345–349, 1994.
- [53] K. Hermus *et al.*, "A review of signal subspace speech enhancement and its application to noise robust speech recognition," *J. on Advances in Signal Process.*, vol. 2007, no. 1, p. 45 821, 2006.
- [54] S. Van Huffel, "Enhanced resolution based on minimum variance estimation and exponential data modeling," *Signal process.*, vol. 33, no. 3, pp. 333–355, 1993.
- [55] B. De Moor, "The singular value decomposition and long and short spaces of noisy matrices," *Trans. on signal process.*, vol. 41, no. 9, pp. 2826–2838, 1993.
- [56] S. H. Jensen *et al.*, "Reduction of broad-band noise in speech by truncated qsvd," *TASLP*, vol. 3, no. 6, pp. 439–448, 1995.
- [57] P. S. K. Hansen *et al.*, "Experimental comparison of signal subspace based noise reduction methods," in *ICASSP*, IEEE, vol. 1, 1999, pp. 101–104.

- [58] U. Mittal and N. Phamdo, "Signal/noise klt based approach for enhancing speech degraded by colored noise," *TASLP*, vol. 8, no. 2, pp. 159–167, 2000.
- [59] F. Xie and D. Van Compernelle, "A family of mlp based nonlinear spectral estimators for noise reduction," in *ICASSP*, IEEE, vol. 2, 1994, pp. II–53.
- [60] E. A. Wan *et al.*, "Networks for speech enhancement," *Handbook of Neural Networks for Speech Process.*, vol. 139, no. 1, p. 7, 1999.
- [61] F. Rosenblatt, "Principles of neurodynamics. perceptrons and the theory of brain mechanisms," Cornell Aeronautical Lab Inc Buffalo NY, Tech. Rep., 1961.
- [62] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010.
- [63] J. Leonard and M. Kramer, "Improvement of the backpropagation algorithm for training neural networks," *Comput. & Chem. Eng.*, vol. 14, no. 3, pp. 337–341, 1990.
- [64] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv:1609.04747*, 2016.
- [65] K. Hornik *et al.*, "Multilayer feedforward networks are universal approximators.," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [66] G. Cybenko, "Approximations by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Syst.*, vol. 2, pp. 183–192, 1989.
- [67] F. Xie and D. V. Compernelle, "Speech enhancement by nonlinear spectral estimation—a unifying approach," in *3rd European Conf. on Speech Commun. and Technol.*, 1993.
- [68] J. Tchorz and B. Kollmeier, "Snr estimation based on amplitude modulation analysis with applications to noise suppression," *TASLP*, vol. 11, no. 3, pp. 184–192, 2003.
- [69] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Syst.*, vol. 6, no. 02, pp. 107–116, 1998.
- [70] G. Hinton *et al.*, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

- [71] Y. Wang and D. Wang, “Boosting classification based speech separation using temporal dynamics,” in *13th Annual Conf. of the Int. Speech Commun. Association*, 2012.
- [72] ———, “Cocktail party processing via structured prediction,” in *Advances in Neural Inf. Process. Syst.*, 2012, pp. 224–232.
- [73] D. Wang and G. J. Brown, *Computational auditory scene analysis: Principles, algorithms, and applications*. Wiley-IEEE press, 2006.
- [74] E. W. Healy *et al.*, “An algorithm to improve speech recognition in noise for hearing-impaired listeners,” *J. of the Acoust. Soc. of America*, vol. 134, no. 4, pp. 3029–3038, 2013.
- [75] X. Lu *et al.*, “Speech enhancement based on deep denoising autoencoder,” in *Interspeech*, vol. 2013, 2013, pp. 436–440.
- [76] S.-W. Fu *et al.*, “Snr-aware convolutional neural network modeling for speech enhancement,” in *Interspeech*, 2016, pp. 3768–3772.
- [77] S. R. Park and J. Lee, “A fully convolutional neural network for speech enhancement,” *arXiv preprint arXiv:1609.07132*, 2016.
- [78] F. Weninger *et al.*, “Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr,” in *Int. Conf. on Latent Variable Analysis and Signal Separation*, Springer, 2015, pp. 91–99.
- [79] M. Tu and X. Zhang, “Speech enhancement based on deep neural networks with skip connections,” in *ICASSP*, IEEE, 2017, pp. 5565–5569.
- [80] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, “Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks,” in *ICASSP*, IEEE, 2015, pp. 708–712.
- [81] D. S. Williamson *et al.*, “Complex ratio masking for monaural speech separation,” *TASLP*, vol. 24, no. 3, pp. 483–492, 2015.
- [82] Y. Luo and N. Mesgarani, “Tasnet: Time-domain audio separation network for real-time, single-channel speech separation,” in *ICASSP*, IEEE, 2018, pp. 696–700.
- [83] S. Pascual *et al.*, “Segan: Speech enhancement generative adversarial network,” *arXiv preprint arXiv:1703.09452*, 2017.
- [84] C. Donahue *et al.*, “Exploring speech enhancement with generative adversarial networks for robust speech recognition,” in *ICASSP*, IEEE, 2018, pp. 5024–5028.

- [85] S. Wang *et al.*, “A transfer learning and progressive stacking approach to reducing deep model sizes with an application to speech enhancement,” in *ICASSP*, IEEE, 2017, pp. 5575–5579.
- [86] X. Shu *et al.*, “A progressive enhancement method for noisy and reverberant speech,” in *23rd Int. Conf. on Digital Signal Process.*, IEEE, 2018, pp. 1–5.
- [87] J. Llobert *et al.*, “Progressive speech enhancement with residual connections,” *arXiv preprint arXiv:1904.04511*, 2019.
- [88] A. Li *et al.*, “Speech enhancement using progressive learning-based convolutional recurrent neural network,” *Applied Acoustics*, vol. 166, p. 107 347, 2020.
- [89] Y. Bengio *et al.*, “Curriculum learning,” in *26th ICML*, ACM, 2009, pp. 41–48.
- [90] A. H. Moore *et al.*, “Speech enhancement for robust automatic speech recognition: Evaluation using a baseline system and instrumental measures,” *Computer Speech & Language*, vol. 46, pp. 574–584, 2017.
- [91] X. Zhao *et al.*, “Robust speaker identification in noisy and reverberant conditions,” *IEEE TALSP*, vol. 22, no. 4, pp. 836–845, 2014.
- [92] A. Varga and H. Steeneken, “Assessment for automatic speech recognition: Ii. noisex-92,” *Speech Commun.*, vol. 12, no. 3, pp. 247–251, 1993.
- [93] R. Chen *et al.*, “Speech enhancement in car noise environment based on an analysis-synthesis approach using harmonic noise model,” in *ICASSP*, 2009, pp. 4413–4416.
- [94] K. Manohar and P. Rao, “Speech enhancement in nonstationary noise environments using noise properties,” *Speech Commun.*, vol. 48, no. 1, pp. 96–109, 2006.
- [95] D. Bateman *et al.*, “Spectral contrast normalization and other techniques for speech recognition in noise,” in *ICASSP*, IEEE, vol. 1, 1992, pp. 241–244.
- [96] J. Meyer *et al.*, “Speech recognition in natural background noise,” *PloS one*, vol. 8, no. 11, e79279, 2013.
- [97] L. Varnet *et al.*, “Phoneme resistance during speech-in-speech comprehension,” in *13th Annual Conf. of the Int. Speech Commun. Association*, 2012.
- [98] B. T. Meyer *et al.*, “Human phoneme recognition depending on speech-intrinsic variability,” *J. of the Acoust. Soc. of America*, vol. 128, no. 5, pp. 3126–3141, 2010.
- [99] S. A. Phatak and J. B. Allen, “Consonant and vowel confusions in speech-weighted noise,” *J. of the Acoust. Soc. of America*, vol. 121, no. 4, pp. 2312–2326, 2007.

- [100] F. Saki and N. Kehtarnavaz, "Automatic switching between noise classification and speech enhancement for hearing aid devices," in *38th Annual Int. Conf. of the Eng. in Med. and Biol. Soc.*, IEEE, 2016, pp. 736–739.
- [101] W. Shi *et al.*, "Deep neural network and noise classification-based speech enhancement," *Modern Physics Letters B*, vol. 31, no. 19-21, p. 1740096, 2017.
- [102] M. Sadeghi *et al.*, "The effect of different acoustic noise on speech signal formant frequency location," *Int. J. of Speech Technol.*, vol. 21, no. 3, pp. 741–752, 2018.
- [103] A. Varga *et al.*, "The noisex-92 study on the effect of additive noise on automatic speech recognition system," *Reports of NATO Research Study Group*, 1992.
- [104] B. S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *J. of the Acoust. Soc. of America*, vol. 55, no. 6, pp. 1304–1312, 1974.
- [105] H. Hermansky *et al.*, "Compensation for the effect of the communication channel in auditory-like analysis of speech (rasta-plp)," in *2nd European Conf. on Speech Commun. and Technol.*, 1991.
- [106] H.-G. Hirsch *et al.*, "Improved speech recognition using high-pass filtering of sub-band envelopes," in *2nd European Conf. on Speech Commun. and Technol.*, 1991.
- [107] H. Hermansky and N. Morgan, "Rasta processing of speech," *TASLP*, vol. 2, no. 4, pp. 578–589, 1994.
- [108] D. F. Rosenthal and H. G. Okuno, *Computational auditory scene analysis*. Lawrence Erlbaum Associates Publishers, 1998.
- [109] B. T. Szabó *et al.*, "Computational models of auditory scene analysis: A review," *Frontiers in Neuroscience*, vol. 10, p. 524, 2016.
- [110] D. Kersten *et al.*, "Object perception as bayesian inference," *Annual Reviews Psychol.*, vol. 55, pp. 271–304, 2004.
- [111] A. Gutschalk *et al.*, "Neuromagnetic correlates of streaming in human auditory cortex," *J. of Neuroscience*, vol. 25, no. 22, pp. 5382–5388, 2005.
- [112] J. A. O'Sullivan *et al.*, "Evidence for neural computations of temporal coherence in an auditory scene and their enhancement during active listening," *Journal of Neuroscience*, vol. 35, no. 18, pp. 7256–7263, 2015.
- [113] D. Klatt, "A digital filter bank for spectral matching," in *ICASSP*, vol. 1, IEEE, 1976, pp. 573–576.

- [114] J. Holmes and N. Sedgwick, "Noise compensation for speech recognition using probabilistic models," in *ICASSP*, IEEE, vol. 11, 1986, pp. 741–744.
- [115] B. Atal and M. Schroeder, "Predictive coding of speech signals and subjective error criteria," *TASLP*, vol. 27, no. 3, pp. 247–254, 1979.
- [116] M. Hasegawa-Johnson and A. Alwan, "Speech coding: Fundamentals and applications," *Wiley encyclopedia of telecommun.*, 2003.
- [117] P. C. Loizou, "Speech enhancement based on perceptually motivated bayesian estimators of the magnitude spectrum," *TASLP*, vol. 13, no. 5, pp. 857–869, 2005.
- [118] M. Schroeder *et al.*, "Optimizing digital speech coders by exploiting masking properties of the human ear," *J. of the Acoust. Soc. of America*, vol. 66, no. 6, pp. 1647–1652, 1979.
- [119] A. Varga and K. Ponting, "Control experiments on noise compensation in hidden markov model based continuous word recognition," in *1st European Conf. on Speech Commun. and Technol.*, 1989.
- [120] S. H. Mohammadi and A. Kain, "An overview of voice conversion systems," *Speech Commun.*, vol. 88, pp. 65–82, 2017.
- [121] C.-P. Chen and J. A. Bilmes, "Mva processing of speech features," *IEEE TALSP*, vol. 15, no. 1, pp. 257–270, 2006.
- [122] X. Xiao *et al.*, "Temporal structure normalization of speech feature for robust speech recognition," *IEEE Signal Process. Letters*, vol. 14, no. 7, pp. 500–503, 2007.
- [123] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," *Speech Commun.*, vol. 25, no. 1-3, pp. 133–147, 1998.
- [124] K. Paliwal, "A study of lsf representation for speaker-dependent and speaker-independent hmm-based speech recognition systems," in *ICASSP*, IEEE, 1990, pp. 801–804.
- [125] H. Hermansky, "Perceptual linear predictive (plp) analysis of speech," *J. of the Acoust. Soc. of America*, vol. 87, no. 4, pp. 1738–1752, 1990.
- [126] C.-P. Chen *et al.*, "Frontend post-processing and backend model enhancement on the aurora 2.0/3.0 databases," in *7th Int. Conf. on Spoken Language Process.*, 2002.
- [127] A. De La Torre *et al.*, "Histogram equalization of speech representation for robust speech recognition," *IEEE TALSP*, vol. 13, no. 3, pp. 355–366, 2005.

- [128] A. E. Rosenberg *et al.*, “Cepstral channel normalization techniques for hmm-based speaker verification,” in *3rd Int. Conf. on Spoken Language Process.*, 1994.
- [129] A. Acero and R. M. Stern, “Environmental robustness in automatic speech recognition,” in *ICASSP*, IEEE, 1990, pp. 849–852.
- [130] A. Acero and R. M. Stern, “Robust speech recognition by normalization of the acoustic space,” in *ICASSP*, 1991.
- [131] J.-W. Hung, “Cepstral statistics compensation using online pseudo stereo code-books for robust speech recognition in additive noise environments,” in *ICASSP*, IEEE, vol. 1, 2006, pp. I–I.
- [132] O. Viikki *et al.*, “A recursive feature vector normalization approach for robust speech recognition in noise,” in *ICASSP*, IEEE, vol. 2, 1998, pp. 733–736.
- [133] C.-C. Hsu *et al.*, “Dictionary update for nmf-based voice conversion using an encoder-decoder network,” in *ISCSLP*, IEEE, 2016, pp. 1–5.
- [134] Ö. Salor and M. Demirekler, “Dynamic programming approach to voice transformation,” *Speech Commun.*, vol. 48, no. 10, pp. 1262–1272, 2006.
- [135] M. Narendranath *et al.*, “Transformation of formants for voice conversion using artificial neural networks,” *Speech Commun.*, vol. 16, no. 2, pp. 207–216, 1995.
- [136] S. Desai *et al.*, “Spectral mapping using artificial neural networks for voice conversion,” *TASLP*, vol. 18, no. 5, pp. 954–964, 2010.
- [137] E. Azarov *et al.*, “Real-time voice conversion using artificial neural networks with rectified linear units,” in *Interspeech*, 2013, pp. 1032–1036.
- [138] L.-J. Liu *et al.*, “Using bidirectional associative memories for joint spectral envelope modeling in voice conversion,” in *ICASSP*, IEEE, 2014, pp. 7884–7888.
- [139] D. J. Rezende *et al.*, “Stochastic backpropagation and approximate inference in deep generative models,” *arXiv:1401.4082*, 2014.
- [140] D. Kingma and M. Welling, “Auto-encoding variational bayes,” in *2nd ICLR*, 2013.
- [141] I. Goodfellow *et al.*, “Generative adversarial nets,” in *Advances in Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [142] C.-C. Hsu *et al.*, “Voice conversion from non-parallel corpora using variational auto-encoder,” in *APSIPA*, IEEE, 2016, pp. 1–6.

- [143] W.-N. Hsu *et al.*, “Learning latent representations for speech generation and transformation,” *arXiv:1704.04222*, 2017.
- [144] J.-Y. Zhu *et al.*, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *ICCV*, IEEE, 2017, pp. 2223–2232.
- [145] T. Kaneko and H. Kameoka, “Parallel-data-free voice conversion using cycle-consistent adversarial networks,” *arXiv preprint arXiv:1711.11293*, 2017.
- [146] T. Kaneko *et al.*, “CycleGAN-vc2: Improved cycleGAN-based non-parallel voice conversion,” in *ICASSP*, IEEE, 2019, pp. 6820–6824.
- [147] H. Kameoka *et al.*, “Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks,” in *Spoken Language Technol. Workshop*, IEEE, 2018, pp. 266–273.
- [148] D. Byrne *et al.*, “An international comparison of long-term average speech spectra,” *J of the Acoust. Soc. of America*, vol. 96, no. 4, pp. 2108–2120, 1994.
- [149] H. E. Cullington and F.-G. Zeng, “Speech recognition with varying numbers and types of competing talkers by normal-hearing, cochlear-implant, and implant simulation subjects,” *J. of the Acoust. Soc. of America*, vol. 123, no. 1, pp. 450–461, 2008.
- [150] J. R. Dubno *et al.*, “Recognition of filtered words in noise at higher-than-normal levels: Decreases in scores with and without increases in masking,” *J. of the Acoust. Soc. of America*, vol. 118, no. 2, pp. 923–933, 2005.
- [151] J. P. Openshaw and J. Masan, “On the limitations of cepstral features in noise,” in *ICASSP*, IEEE, vol. 2, 1994, pp. II–49.
- [152] A. P. Varga and R. K. Moore, “Hidden markov model decomposition of speech and noise,” in *ICASSP*, IEEE, 1990, pp. 845–848.
- [153] G. Parikh and P. C. Loizou, “The influence of noise on vowel and consonant cues,” *J. of the Acoust. Soc. of America*, vol. 118, no. 6, pp. 3874–3888, 2005.
- [154] G. Hu and D. Wang, “A tandem algorithm for pitch estimation and voiced speech segregation,” *TASLP*, vol. 18, no. 8, pp. 2067–2079, 2010.
- [155] J. Li *et al.*, “An overview of noise-robust automatic speech recognition,” *TASLP*, vol. 22, no. 4, pp. 745–777, 2014.

- [156] P. Welch, “The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms,” *Trans. on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.
- [157] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” *J. of Mach. Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
- [158] J. Garofolo *et al.*, “CSR-I (WSJ0) LDC93S6B,” *Linguistic Data Consortium*, 1993.
- [159] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [160] Y. A. LeCun *et al.*, “Efficient backprop,” in *Neural Networks: Tricks of the Trade*, Springer, 2012, pp. 9–48.
- [161] C. Barras and J.-L. Gauvain, “Feature and score normalization for speaker verification of cellular data,” in *ICASSP*, IEEE, vol. 2, 2003, pp. II–49.
- [162] L. Rabiner, “Fundamentals of speech recognition,” *Fundamentals of speech recognition*, 1993.
- [163] E. Vincent *et al.*, “An analysis of environment, microphone and data simulation mismatches in robust speech recognition,” *Computer Speech & Language*, vol. 46, pp. 535–557, 2017.
- [164] B. Chen and P. C. Loizou, “A laplacian-based mmse estimator for speech enhancement,” *Speech Commun.*, vol. 49, no. 2, pp. 134–143, 2007.
- [165] R. Eldan and O. Shamir, “The power of depth for feedforward neural networks,” in *Conf. on Learning Theory*, 2016, pp. 907–940.
- [166] M. Telgarsky, “Benefits of depth in neural networks,” *arXiv preprint arXiv:1602.04485*, 2016.
- [167] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [168] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in *32nd ICML*, vol. 37, 2015.
- [169] A. Van Den Oord *et al.*, “Neural discrete representation learning,” in *Advances in Neural Inf. Process. Syst.*, 2017, pp. 6306–6315.
- [170] P. Domingos, “A few useful things to know about machine learning,” *Commun. ACM*, vol. 55, no. 10, pp. 78–87, 2012.

- [171] C. C. Aggarwal *et al.*, “On the surprising behavior of distance metrics in high dimensional space,” in *Int. Conf. on Database Theory*, Springer, 2001, pp. 420–434.
- [172] J. F. Gemmeke *et al.*, “Exemplar-based sparse representations for noise robust automatic speech recognition,” *IEEE TALSP*, vol. 19, no. 7, pp. 2067–2080, 2011.
- [173] *Optimizing siri on homepod in far-field settings*, <https://machinelearning.apple.com/research/optimizing-siri-on-homepod-in-far-field-settings>, Accessed: 2020-09-02.
- [174] K. Tan and D. Wang, “Learning complex spectral mapping with gated convolutional recurrent networks for monaural speech enhancement,” *IEEE TALSP*, vol. 28, pp. 380–390, 2019.
- [175] Y. Luo and N. Mesgarani, “Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation,” *IEEE TALSP*, vol. 27, no. 8, pp. 1256–1266, 2019.
- [176] N. Tawara *et al.*, “Multi-channel speech enhancement using time-domain convolutional denoising autoencoder,” in *Interspeech*, 2019, pp. 86–90.
- [177] *Institute for telecommunication sciences*.
- [178] W. H. Press, “Flicker noises in astronomy and elsewhere,” *Comments on Astrophysics*, vol. 7, pp. 103–119, 1978.
- [179] J. A. Hildebrand, “Anthropogenic and natural sources of ambient noise in the ocean,” *Marine Ecology Progress Series*, vol. 395, pp. 5–20, 2009.
- [180] R. Lewis, “Noise data,” *Signal Process. Inf. Base*, 2013.

VITA

Sicheng Wang was born on June 28, 1989, and grew up in Mianyang City in Southwest China. He finished his high school education at Hwa Chong Institution in Singapore. He obtained a degree of Bachelor of Science at Lafayette College in Easton, Pennsylvania. Upon graduation, he joined Georgia Institute of Technology to pursue Master and Ph.D. in electrical and computer engineering.