University of Sheffield

Perceptually Motivated Speech Enhancement

George Close

Primary Supervisor: Prof. Stefan Goetze

Secondary Supervisor: Prof. Thomas Hain

A thesis submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the
School of Computer Science
March 27, 2025

Dedication

But this machine can only swallow money
You can't lay a patch by computer design
It's just a lot of stupid, stupid signs

R.E.M - 'The Sidewinder Sleeps Tonite'

I first express my thanks to my supervisors, Professor Stefan Goetze and Professor Thomas Hain for their support, feedback and advice throughout the past 4 years. I would also like to thank Dr Rama Doddipatla and Dr Cong Thanh Do of Toshiba Research Europe for their continued interest and guidance of the project.

I would like to thank everyone at the UKRI CDT for Speech and Language Technologies and say that I am proud to have been part of such an exceptional cohort of students. In particular I would like to highlight my student co-authors Samuel Hollands, Dr William Ravenscroft, Robbie Sutherland and Dr Rhiannon Mogridge.

I would like to thank my family for their unerring support and encouragement over my entire academic career. Finally, I would like to thank my wife, Yetzi Cortes Flores Close, for her infinite patience, understanding and love.

Abstract

Speech Enhancement (SE) is a vital technology for online human communication. Applications of Deep Neural Network (DNN) technologies in concert with traditional signal processing approaches to the task have revolutionised both the research and implementation of SE in recent years. However, the training objective of these Neural Network Speech Enhancement (NNSE) systems generally do not consider the psychoacoustic processing which occurs in the human auditory system. As a result, enhanced audio can often contain auditory artefacts which degrade the perceptual quality or intelligibility of the speech. To overcome this, systems which directly incorporate psychoacoustically motivated measures into the training objectives of NNSE systems have been proposed.

A key development in speech audio processing in recent years is the emergence of Self Supervised Speech Representation (SSSR) models. These are powerful foundational DNN models which can be utilised for a number of more specific speech processing tasks, such as speech recognition, emotion detection as well as SE. Finally, the methods of evaluation of SE systems have been revolutionised by DNN technology, that is to say the creation of systems which are able to directly predict Mean Option Score (MOS) ratings of Speech Quality (SQ) or Speech Intelligibility (SI) derived from human listening tests.

This thesis aims to investigate these three areas; psychoacoustic training objectives of NNSE, the incorporation of SSSR features and the prediction of human derived labels of speech directly from audio signals. Further, the intersection of these areas and combined use of techniques from these areas will be investigated.

A widely adopted approach for psychoacoustically motivated NNSE training is the MetricGAN framework. Here, a NNSE network is trained as generator adversarially (pitted against in competition) with a metric prediction discriminator. The discriminator is tasked with predicting the score assigned to the input audio by a (typically non-differentiable and thus unable to be used as a loss function directly) metric function, while the generator uses inference of the discriminator to obtain a loss value for its outputs. While MetricGAN has proved effective and is becoming a widely adopted technique, there is scope to improve it in several areas. Several of the contributions of this thesis are related to these improvements including the introduction of an additional DNN tasked with improving the range of inputs to the metric prediction Discriminator, changes to the Neural Network (NN) structure of both components and the prediction of non-intrusive measures among others. A key finding of this work is that perceptually motivated NNSE systems tend to *overfit* towards the target perceptual metric, resulting in degraded "real world" enhancement performance. The concept of the metric prediction is further developed into systems proposed for the related task of DNN based human MOS prediction. This can be done intrusively meaning that the system has access to a non-distorted version of the signal under test as a reference or non-intrusively meaning that only the signal under test is available. Here, human labels of SQ or SI are directly predicted from the audio signal stimulus. SI prediction is mainly investigated in the domain of hearing aid SE

system evaluation in this work. State of the art performance is achieved by SQ prediction systems developed and presented in this work.
Two novel applications of SSSR are presented. Firstly, as feature space representations in the loss function of NNSE systems. In particular, it is found that using earlier intermediate DNN layer outputs in this application is particularly effective, and a strong correlation between the SSSR distance measure and psychoacoustic metrics and MOS labels is shown. Secondly, SSSR representations are proposed for use as feature extractors for the discriminator DNN components of the MetricGAN framework, as well as for MOS estimators.

Contents

Dedication i

Abstract ii

List of Symbols xviii

List of Acronyms xxi

I Introduction and Background 1

1 Introduction 2

1.1 The Speech Enhancement Task . . . 2

1.2 Motivation, Research Questions and Contributions . . . 3

1.2.1 Towards A Unified View of Loss Functions and Metrics . . . 3

1.2.2 Research Objectives and Questions . . . 5

1.2.3 Contributions . . . 5

1.3 Thesis Structure . . . 7

2 Background 8

2.1 Notation and Signal Model . . . 8

2.1.1 Signal Model . . . 8

2.1.2 Single Channel Speech Enhancement . . . 8

2.1.3 DNN Metric Prediction . . . 9

2.1.4 STFT Features . . . 9

2.2 Speech in Noisy Environments . . . 10

2.2.1 Digital Audio Representation . . . 11

2.3 Speech Enhancement Algorithms . . . 12

2.3.1 Signal Processing Based Speech Enhancement 12
2.3.2 DNN-based Speech Enhancement 13
2.3.3 Speech Enhancement Generative Adversarial Network (SEGAN) 15
2.3.4 Metric Derived Objective Function 16
2.4 Deep Neural Networks 16
2.4.1 Neural Network Model Structure 17
2.4.2 Neural Network Activation Functions 27
2.4.3 Neural Network Loss Functions 29
2.4.4 Neural Network Training 30
2.5 Pretrained Foundational Speech Models 31
2.5.1 Pre-training and Fine-tuning 31
2.5.2 Self Supervised Speech Representation (SSSR) 31
2.5.3 Whisper 33
2.6 Generative Adversarial Networks (GANs) 35
2.7 Metrics 36
2.7.1 Mean Opinion Score 38
2.7.2 Signal to Noise Ratio 38
2.7.3 Scale Invariant Signal Distortion Ratio 38
2.7.4 STOI 39
2.7.5 PESQ 39
2.7.6 Composite Measure 40
2.7.7 DNSMOS 40
2.7.8 HASPI 40
2.8 Neural Metric Prediction 41
2.9 Datasets , Challenges and Corpora for Speech Enhancement 41
2.9.1 VoiceBank-DEMAND 41
2.9.2 CHiME3 Data 44
2.9.3 CHiME7 - UDASE Data 45
2.9.4 CommonVoice Dataset 45
2.10 Datasets for Speech Intelligibility (SI) Prediction 45
2.10.1 Clarity Prediction Challenge 1 45
2.11 Datasets for Speech Quality (SQ) MOS Prediction 46
2.11.1 NISQA Dataset 47
2.11.2 Tencent Dataset 47

2.11.3 IUB Dataset . . . 47
2.11.4 PSTN Dataset . . . 47
2.11.5 Overall MOS Distribution of SQ Datasets . . . 48
2.12 Baseline NNSE System - MetricGAN+ . . . 49
2.12.1 Generator Network for Signal Enhancement . . . 49
2.12.2 Discriminator Network for Metric Prediction . . . 49
2.12.3 MetricGAN+ Training . . . 50
2.12.4 Discriminator Model Structure . . . 50
2.12.5 Generator Model Structure . . . 51
2.13 CMGAN SE DNN . . . 51
2.14 DPT-FSNet SE DNN . . . 54

II Expanding the MetricGAN Framework 56

Preface 57

3 MetricGAN+/- 58
3.1 Introduction . . . 58
3.2 Metric Score Distribution in Training Data . . . 58
3.3 MetricGAN+/- Framework . . . 60
3.4 MetricGAN+/- Experiments . . . 60
3.4.1 Experiment Setup . . . 60
3.4.2 Experiment Results . . . 61
3.4.3 Spectogram Analysis . . . 62
3.4.4 Validation Performance . . . 62
3.4.5 Generalisation To Unseen Data . . . 62
3.5 Summary . . . 64

4 Further MetricGAN Variations 65
4.1 Introduction . . . 65
4.2 System Overview . . . 65
4.2.1 SE Generator . . . 65
4.2.2 Metric Prediction Discriminator . . . 66
4.2.3 Degenerator . . . 66
4.2.4 Training Details . . . 67

4.3 Experiments . . . 68
4.3.1 Datasets Used . . . 68
4.3.2 Experiment 1: Investigating the Effect of Hyperparameter w . . . 69
4.3.3 Experiment 2: Historical Set Reduction Techniques . . . 73
4.3.4 Experiment 3: Phase Aware Enhancement . . . 77
4.3.5 Experiment 4: ASR based enhancement objective . . . 78
4.4 Summary . . . 79

5 CMGAN+/+ 80
5.1 Speech Enhancement System Description . . . 80
5.1.1 Conformer-based Generator . . . 80
5.1.2 Metric Estimation Discriminator . . . 81
5.1.3 Metric Data Augmentation Pseudo-Generator . . . 81
5.2 Experiment Setup . . . 82
5.3 Results . . . 82
5.3.1 Spectrogram Analysis . . . 83
5.3.2 Challenge Results . . . 85
5.4 Summary . . . 85

6 Multi-CMGAN+/+ 88
6.1 Introduction . . . 88
6.2 Speech Enhancement System . . . 88
6.2.1 Conformer-based Speech Enhancement Generator . . . 88
6.2.2 Metric Estimation Discriminator . . . 89
6.2.3 Discriminator Network Structure . . . 89
6.3 Experiments . . . 90
6.3.1 Training Setup . . . 90
6.4 Results . . . 90
6.5 Summary . . . 91

III Self Supervised Speech Representation Based Loss Functions 92

Preface 93

7 SSSR loss for Speech Enhancement **94**
7.1 Introduction . . . 94
7.2 SSSR derived distances in relation to speech assessment metrics . . . 94
7.2.1 Datasets Used . . . 95
7.2.2 SSSR distances and SE motivated metrics . . . 95
7.2.3 SSSR distances and human quality assessment . . . 96
7.3 SSSR Based Signal Enhancment Experiment . . . 96
7.3.1 Experiment setup . . . 96
7.3.2 Loss Functions . . . 97
7.3.3 Enhancement Model Structure . . . 97
7.3.4 Signal Enhancement Performance . . . 98
7.3.5 Analysis . . . 98
7.4 Summary . . . 98

8 Language influence in SSSR Losses **100**
8.1 Introduction . . . 100
8.2 CommonVoice-DEMAND: A Multilingual Speech Enhancement Dataset . . . 100
8.2.1 CommonVoice Dataset . . . 100
8.2.2 Candidate Selection . . . 100
8.2.3 Dataset Creation . . . 101
8.3 Speeech Enhancement Experiments . . . 102
8.3.1 Experiment Setup . . . 102
8.3.2 Datasets . . . 102
8.3.3 SSSR Signal Enhancement Loss Function . . . 102
8.3.4 Results . . . 103
8.4 Summary . . . 105

IV Human Audio Label Prediction **106**

Preface **107**

9 Neural Network Speech Intelligibility Prediction 108
9.1 Introduction . . . 108
9.2 Neural Intelligibility Prediction . . . 108
9.2.1 Feature Extraction . . . 110
9.2.2 Model Structure for Non-Intrusive Prediction . . . 110
9.2.3 Model Struture for Intrusive Prediction . . . 110
9.3 Experiments . . . 111
9.3.1 Tools and Software . . . 111
9.3.2 Data Description . . . 111
9.4 Clarity Prediction Challenge (CPC) Metric Distributions . . . 111
9.5 Experiment Setup . . . 112
9.5.1 Results . . . 113
9.6 Summary . . . 114

10 Self Supervised Representations for SI Prediction 115
10.1 Introduction . . . 115
10.2 SSSRs for Metric Prediction . . . 115
10.3 Analysing Relationships between SSSRs and Human SI . . . 115
10.4 SSSR-based Intelligibility Prediction . . . 117
10.4.1 Model Structure and Experiment Setup . . . 117
10.5 Results . . . 119
10.5.1 Results on CPC1 Closed set . . . 119
10.5.2 Results of CPC1 Open set . . . 120
10.5.3 System and Listener-wise Analysis . . . 120
10.6 Summary . . . 123

11 WhisSQA - Speech Quality Prediction 125
11.1 Speech Quality (SQ) Prediction Models . . . 126
11.2 Mamba . . . 126
11.3 Experiment 1 - Feature Selection . . . 127
11.3.1 Experiment Setup . . . 128
11.3.2 Results . . . 128
11.4 Experiment 2 - Training Data Selection . . . 129
11.4.1 Experiment Setup . . . 129

11.4.2 Results . . . 130
11.5 Experiment 3 - Task Variant Exploration . . . 130
11.5.1 Experiment Setup . . . 131
11.5.2 Results . . . 131
11.6 Experiment 4 - Model Variations . . . 132
11.6.1 Experiment Setup . . . 132
11.6.2 Results . . . 132
11.7 Analysis . . . 133
11.7.1 Layer Weights . . . 133
11.8 Summary . . . 134

12 Hallucinations in Neural Network SE Systems 135
12.1 Introduction . . . 135
12.2 Non-Intrusive Speech Quality Predictor . . . 136
12.3 Speech Enhancement System . . . 137
12.3.1 Model Structure . . . 137
12.3.2 Loss Function . . . 137
12.4 Experiment 1 - Scaling the Quality Estimator's Influence . . . 138
12.4.1 Experiment Setup . . . 138
12.4.2 Results . . . 138
12.4.3 Spectrogram Comparison . . . 139
12.5 Experiment 2 - Listening Test . . . 139
12.5.1 Setup . . . 139
12.5.2 Results . . . 141
12.6 Summary . . . 141

V Conclusions 143

13 Concluding Remarks 144

14 Future Research 145

List of Figures

1.1 Basic front-end pipeline for Speech Enhancement in a noisy environment with a single microphone. For more detail on the signal model used in this work, see Section 2.1. . . . 3

1.2 Overview of SE metrics and NNSE loss functions . . . 4

2.1 Illustration of Short Time Fourier Transform (STFT) feature computation. The variable $p[n]$ is a placeholder for any signal, i.e $p[n] \in \{x[n], s[n], \hat{s}[n]\}$ . . . 9

2.2 Waveform (top) and spectrogram (bottom) representations of clean (left) and noisy (right) speech, sourced from the VoiceBank-DEMAND (Valentini-Botinhao et al., 2016) dataset. . . . 11

2.3 Speech spectrograms in four different noise environments, all with a Signal-to-Noise-Ratio (SNR) of 2.5dB . . . 12

2.4 Speech spectrograms in four different SNRs, all noise is from a cafe environment sourced from the VoiceBank-DEMAND (Valentini-Botinhao et al., 2016) dataset . 13

2.5 Noisy speech signal enhanced via traditional signal processing method . . . 14

2.6 NNSE using a mapping approach . . . 14

2.7 NNSE using a masking approach . . . 15

2.8 Supervised DNN Training. . . . 17

2.9 Unsupervised DNN Training. . . . 17

2.10 A simple DNN consisting of four sequential Linear Layers with no activation functions. . . . 18

2.11 A 1D Convolutional Neural Network (CNN) filter of length 3 over an input sequence of length 9 with a stride of 1 (top),2 (middle) and 3 (bottom) respectively. . . . 20

2.12 A 1D CNN filter of length 3 and stide 3 over an input sequence of length 9 with a dilation of 1 (top), 2 (bottom) respectively. . . . 21

2.13 A 1D CNN filter of length 3 and stride 1 with left side padding of 2 . . . 21

2.14 A 2D CNN filter of size 2×2 and stride $(2, 2)$ with right side padding of $(0, 1)$ and 2D CNN filter of size 2 and stride 1 with right side padding of $(0, 1)$. . . . 22

2.15 Average (top) and Max (bottom) pooling on the output of a 1D CNN filter . . . 23

2.16 Forward pass of an Recurrent Neural Network (RNN) layer over three time steps . 23

2.17 Overview of a Conformer DNN block. 25

2.18 Plots of various non-linearities used as DNN layer activation functions. 27

2.19 Representations extracted from SSSR model with time-domain input signal $\mathbf{s}[n]$. Feature channels are sorted (Ravenscroft et al., 2022) and values normalised for clarity. 32

2.20 An overview of the WavLM architecture. 34

2.21 An overview of the Whisper DNN model architecture. 35

2.22 Typical GAN structure overview 36

2.23 An intrusive metric computed for the signal $p(t)$ on a time domain signal given the reference signal $s(t)$ 37

2.24 A non-intrusive metric computed on a time domain signal $p(t)$. 37

2.25 Block diagram of Short-Time Objective Intelligibility (STOI) score calculation. . . 39

2.26 Block diagram of Perceptual Evaluation of Speech Quality (PESQ) score calculation. 39

2.27 Training of intrusive and non-intrusive metric predictors of an intrusive metric. . . 42

2.28 Training of MOS predictor. 42

2.29 PESQ and STOI distributions in the VoiceBank-DEMAND training and test sets. . 43

2.30 Signal generation for Clarity Prediction Challenge. 46

2.31 Normalised MOS score distribution across SQ Datasets (lines indicate minimum, mean and maximum MOS in each dataset). 48

2.32 Training and inference of MetricGAN+ Generator. 49

2.33 Training and inference of MetricGAN+ Discriminator. 50

2.34 MetricGAN+ $\mathcal{D}$ DNN structure 51

2.35 MetricGAN+ $\mathcal{G}$ DNN structure 52

2.36 CMGAN $\mathcal{G}$ DNN structure 53

2.37 DPT-FSNet DNN structure 55

3.1 PESQ and STOI distribution of the VoiceBank-DEMAND Training Set with Noise Type labels. 59

3.2 STOI scores of the replay buffer of STOI objective MetricGAN+. 59

3.3 Spectograms of: (a) clean reference features $\mathbf{S}_f$, (b) noisy features $\mathbf{X}_f$, (c) Mask $\mathbf{M}_\mathcal{G}$ and (d) enhanced output $\hat{S}_f$ for MetricGAN+ baseline PESQ objective model, (e) Mask $\mathbf{M}_\mathcal{G}$ and (f) enhanced output $\hat{S}_f$ for MetricGAN+/- PESQ objective model. Source audio file is `p232_014.wav` of VoiceBank-DEMAND testset. 63

3.4 Graph showing PESQ score on validation set during training for PESQ objective MetricGAN+ and MetricGAN+/- models 64

4.1 Diagram of VoiceBank-DEMAND-Rerecorded Recording Environment 69

4.2 PESQ and STOI metric correlation between original VoiceBank-DEMAND and VoiceBank-DEMAND-Rerecorded 70

4.3 Distribution of PESQ scores of $\mathcal{N}$'s outputs at final training epoch with $s[n]$ as input and $w = 0.65$ for training on the original VoiceBank-DEMAND. 71

4.4 Distribution of PESQ scores in original VoiceBank-DEMAND training set 72

4.5 Distribution of PESQ scores in Rerecorded VoiceBank-DEMAND training set . . . 73

4.6 Distribution of PESQ scores of $\mathcal{N}$'s outputs at final training epoch with $x[n]$ as input and $w = 0.8$ for training on the original VoiceBank-DEMAND. 75

4.7 Training time versus epoch counter for the historical set reduction techniques. . . . 76

4.8 Training time versus validation PESQ score for the historical set reduction techniques. 77

5.1 Noisy and enhanced spectrograms of audio file `S01_P01_0.wav` from the CHiME-5 evaluation set. 87

7.1 Scatter plots showing the relationship between the PESQ metric and Mean Squared Error (MSE) Spectrogram distance d_{SG} as well as, HuBERT d_{FE}, HuBERT d_{OL} distances for the VoiceBank-DEMAND testset 96

7.2 Scatter plots showing the relationship between human MOS scores and MSE Spectrogram distance d_{SG}, HuBERT d_{FE} and HuBERT d_{OL} with in the NISQA Challenge testset 97

7.3 Visualisation of inputs representations of $s[n]$, $x[n]$ to d_{SG}, HuBERT d_{EF} and XLSR d_{EF}. SSSR features are sorted according to depthwise euclidean distance following Algorithm 1 in (Ravenscroft et al., 2022) and a sigmoid function is applied to increase clarity. 99

9.1 Diagram of general non-intrusive SI metric prediction training 109

9.2 SI metrics versus ground truth correctness percentage in CPC 1 Training Set 111

9.3 Histogram showing the distribution of ground truth correctness i in CPC1 training set (top) and a bar chart showing average correctness i per listener in the CPC1 training set (bottom). Dotted lines are respective overall average values. 112

10.1 Scatter plots showing the correlation between distances d_{FE} (Eq. 10.1) for XLSR features and d_{OL} (Eq. 10.2) Correctness i for the CPC1 training set (upper panels). Scatter plots showing the relationship between the $\hat{\mathbf{s}}'[n]$ and $\hat{\mathbf{s}}[n]$ distances (lower) . 117

10.2 Scatter plots showing the correlation between HuBERT d_{FE} and d_{OL} and Correctness i in the CPC1 training set (upper).Scatter plots showing the relationship between the $\hat{\mathbf{s}}'[n]$ and $\hat{\mathbf{s}}[n]$ distances (lower) 118

10.3 Scatter plot showing the correlation between d_{SG} and Correctness i in the CPC1 training set 119

10.4 System (top) and listener-wise (bottom) correctness prediction $\hat{i}$ (l./green) vs. true i (r./brown) using HuBERT output for $\hat{\mathbf{S}}_{\mathrm{OL}}$ model on CPC1 closed set. 121

10.5 System (top) and listener-wise (bottom) correctness prediction $\hat{i}$ (l./green) vs. true i (r./brown) using HuBERT outpur for $\hat{\mathbf{S}}'_{\mathrm{OL}}$ model on CPC1 closed set. 122

10.6 System (top) and listener-wise (bottom) correctness prediction $\hat{i}$ (l./green) vs. true i (r./brown) using HuBERT output for $\hat{\mathbf{S}}_{\mathrm{OL}}$ model on CPC1 open set. Listeners and Systems unseen during training are bold. 123

10.7 System (top) and listener-wise (bottom) correctness prediction $\hat{i}$ (l./green) vs. true i (r./brown) using HuBERT output for $\hat{\mathbf{S}}'_{\mathrm{OL}}$ model on CPC1 open set. Listeners and Systems unseen during training are bold. 124

11.1 General structure of proposed SQ prediction neural network(s) and feature extraction. Note that each 'Weighted Sum' block contains model parameters, i.e. layer weights $\{\alpha^{(0)}, .., \alpha^{(12)}\}$ which are updated during prediction model training. 127

11.2 Scatter plots for NISQA testset performance of single headed baseline NISQA (left) and best performing proposed $\mathcal{D}_1$ model using $\bar{\mathbf{X}}_{\mathrm{E}}$ (right). 128

11.3 Unmasked $\bar{\mathbf{X}}_{\mathrm{E}}$ features (top left), $\bar{\mathbf{X}}_{\mathrm{E}}$ with padding region masked (top right) and $\bar{\mathbf{X}}_{\mathrm{E}}$ with signal region masked (bottom) 133

11.4 Layer weights for models trained on NISQA only and NISQA, Tencent and PSTN for different input features: WavLM (top), Whisper Encoder (middles) and Decoder (bottom). 134

12.1 (Magnitude) spectrogram comparison for differing values of α. 140

List of Tables

2.1 Breakdown of SE metrics by mixing SNR value in the VoiceBank-DEMAND training set. 44

2.2 Breakdown of SE metrics by distortion noise type in the VoiceBank-DEMAND training set. 44

2.3 Breakdown of SE metrics by mixing SNR value in the VoiceBank-DEMAND testset. 44

2.4 Breakdown of SE metrics by distortion noise type in the VoiceBank-DEMAND testset. 45

2.5 Comparison of SQ Datasets (EN: English; DE: German, CH: Chinese). 46

3.1 Performance of MetricGAN+ (MG+) and MetricGAN+/- (MG+/-) on VoiceBank-DEMAND test set for objective PESQ (P) or STOI (S), * denotes the simulation where β is made learnable 61

3.2 Performance on real component of CHiME3 test set 62

3.3 Performance on simulated component of CHiME3 test set 62

4.1 Comparison between MetricGAN (MG) derived frameworks 65

4.2 Performance of MetricGAN+/- with x as input and w values for $\mathcal{N}$ on the original VoiceBank-DEMAND dataset. 72

4.3 Performance of MetricGAN+/- with s as input and w values for $\mathcal{N}$ on the original VoiceBank-DEMAND dataset. 73

4.4 Performance of MetricGAN+/- with x as input and w values for $\mathcal{N}$ on the rerecorded VoiceBank-DEMAND dataset. 74

4.5 Performance of MetricGAN+/- with s as input and w values for $\mathcal{N}$ on the rerecorded VoiceBank-DEMAND dataset. 74

4.6 Performance of MetricGAN+/- with historical set size reduction techniques on original VoiceBank-DEMAND 76

4.7 Performance of Cutoff historical set reduction technique with differing values for O_{cutoff} on original VoiceBank-DEMAND 77

4.8 Performance of CMGAN+/- β ditto with CMGAN network structure for $\mathcal{G}$ and $\mathcal{N}$. 78

4.9 Performance of MetricGAN+/- with phase aware Generators on original VoiceBank-DEMAND 78

4.10 WER objective on original VoiceBank-DEMAND 79

4.11 WER objective on rerecorded VoiceBank-DEMAND 79

5.1 SI-SDR results on the reverberant LibriCHiME eval set. 83

5.2 DNSMOS results on CHiME5 eval set. 84

5.3 Comparison with other challenge entries ranked by DNSMOS OVR score. 85

5.4 Comparison of top-performing challenge entries on listening tests with human participants, ranked by OVRL MOS. 86

6.1 DNSMOS results on CHiME5 eval set. 91

6.2 SI-SDR results on the reverberant LibriCHiME eval set. 91

7.1 Spearman r and Pearson ρ correlation between distance measures and speech quality and intelligibility metrics in the VoiceBank-DEMAND testset, as well as MOS in the NISQA Challenge testset 95

7.2 Signal Enhancement performance on the VoiceBank-DEMAND testset. 98

8.1 Performance of models trained on CommonVoice-DEMAND *English*; tested on English, Spanish and Welsh testsets. 103

8.2 Performance of models trained on CommonVoice-DEMAND *Spanish*; tested on English, Spanish and Welsh testsets. 104

8.3 Performance of L_{FE} Loss models trained on CommonVoice-DEMAND English and tested on English testset. 105

9.1 Spearman r and Pearson ρ Correlation between SI metrics and correctness label i in CPC1 Training set 112

9.2 Non Intrusive Performance on the Clarity Prediction Challenge Training Set 113

9.3 Intrusive Performance on the Clarity Prediction Challenge Training Set 114

9.4 Non Intrusive Performance on the Clarity Prediction Challenge Test Set 114

10.1 Spearman and Pearson correlations between distance measures and correctness values i in the CPC1 training set, strongest correlations in bold. 116

10.2 Non-Intrusive Prediction Performance on the Clarity Prediction Challenge 1 (CPC1) closed set. Best performances for baselines and proposed methods in boldface font. 120

10.3 Non Intrusive Prediction Performance on the CPC1 open set. 121

11.1 Experiment model structure and training data overview. 125

11.2 Predictor performance of $\mathcal{D}_1$ for best epoch (Ep.) in terms of Spearman Correlation r and Root Mean Squared Error (RMSE) e for different input features on the NISQA dataset. **Best** and second best shown in **Bold** and underline respectively. 129

11.3 Training Data Ablation Study for best performing proposed $\mathcal{D}_1$ model. **Best** and second best shown in **Bold** and underline respectively. 130

11.4 Comparison of $\mathcal{D}_1$ with SOTA systems **Best** and second best shown in **Bold** and underline respectively. 131

11.5 Results for Multi Headed $\mathcal{D}_1$ Models versus Single Head (MOS Only) Prediction . 131

11.6 Results for Intrusive (I) versus Non-Intrusive (NI) Prediction 132

11.7 SQ Model Variation performance using simple Linear model base. 133

11.8 Masked $\bar{\mathbf{X}}_{\mathrm{E}}$ performance for model $\mathcal{D}_3$ 133

12.1 Proposed SQ Predictor compared with baseline NISQA model. 136

12.2 Performance of Speech Enhancement for different α in ((12.2)) for the VoiceBank-DEMAND testset. 137

12.3 Listening Test Results 141

List of Symbols

$\mathbf{A}$ State Space learnable parameter hidden state weight matrix. 25

$\mathcal{A}_{\mathrm{D}}$ Whisper Decoder. 34

$\mathcal{A}_{\mathrm{E}}$ Whisper Encoder. 34

$\alpha_{\mathrm{STOI-loss}}$ STOI loss weighting. 16

B Batch size. 51

b NN layer bias parameter value. 18

$c[n]$ CNN output hidden state. 19

D_k Attention Key dimension. 22

D_q Attention Query dimension. 22

$\mathcal{D}$ Metric Prediction system. 9

d_{FE} Feature Encoder distance. 95

d_{OL} Output Representation distance. 95

e Euler's Number. 28

F_{Hz} Frequency. 9

$\mathcal{F}$ NN layer. 17

f Technical frequency in STFT. 9

$h(t)$ continuous hidden state. 25

h Window function in STFT. 10

H MetricGAN History Portion percentage hyperparameter. 50

h_n RNN hidden state. 20

I MetricGAN History Portion segments. 50

j_t Quantized vector in wav2vec pre-training objective. 32

$\mathbf{K}$ Attention Key matrix. 22

$\bar{\mathbf{K}}$ State space model convolutional kernel. 26

$\mathcal{L}$ Neural Network training loss function. 3, 16

L_k Attention Key length. 22

L_q Attention Query length. 22

LC IBM local criterion value. 15

$\mathcal{Q}$ Speech Enhancement metric function. 3

l DNN loss value. 3

lstm_t LSTM cell state. 21

λ_{SEGAN} SEGAN loss weighting. 15

m Discrete time index of windowed signal in STFT. 9

$\mathcal{N}$ Degenerator. 60

n Discrete time index. 8

$O_{\mathcal{N}}$ Objective metric value of $\mathcal{N}$ outputs. 70

$o(t)$ continuous output state. 25

$\mathbf{P}_{\mathrm{Im}}$ Imaginary Spectrogram component of any discrete time domain signal. 9

$\mathbf{P}_{\mathrm{Mag}}$ Magnitude Spectrogram of any discrete time domain signal. 9

$\mathbf{P}_{\mathrm{P}}$ Phase component of any discrete time domain signal. 10

$\mathbf{P}_{\mathrm{Re}}$ Real Spectrogram component of any discrete time domain signal. 9

$p(t)$ Any continuous time domain signal. 25

$p[n]$ Any discrete time domain signal. 9

$\mathbf{Q}$ Attention Query matrix. 22

$\hat{q}$ Estimated Speech Enhancement metric value. 9

q Speech Enhancement metric value. 3, 9

r_t output vector in wav2vec pre-training objective. 32

ρ Pearson Correlation. 95

r Spearman Correlation. 95

$\mathbf{S}_{\mathrm{FE}}$ Feature Encoder output of an SSSR model. 31

$\mathbf{S}_{\mathrm{OL}}$ Final layer output of an SSSR model. 31

$\mathcal{S}$ Clarity hearing loss simulation. 116

$\hat{s}[n]$ Discrete time domain enhanced speech signal. 3, 8

$s[n]$ Discrete time domain clean speech signal. 3, 8

T Time index of 2D Frequency/Feature domain representation. 9

t continuous time domain index. 25

τ_{wav2vec} wav2vec pre-training objective scaling constant. 32

$\mathbf{V}$ Attention Value matrix. 23

$v[n]$ Discrete time domain additive noise signal. 8

$\mathbf{W}$ NN layer weight parameter matrix. 18

w Degenerator objective metric value. 60

ξ MetricGAN output mask floor. 60

$x[n]$ Discrete time domain noisy speech signal. 4, 8

List of Acronyms

ASR	Automatic Speech Recognition
BLSTM	Bidirectional Long Short-Term Memory
CD	Contrastive Divergence
CHiME3	Computational Hearing in Multisource Environments 3
CHiME7	Computational Hearing in Multisource Environments 7
CMGAN	Conformer Metric Generative Adversarial Network
CNN	Convolutional Neural Network
CPC	Clarity Prediction Challenge
CPC1	Clarity Prediction Challenge 1
CTC	Connectionist Temporal Classification
DFT	Discrete Fourier Transform
DNN	Deep Neural Network
DNSMOS	Deep Noise Suppression Mean Opinion Score
DPT	Dual Path Transformer
ESTOI	Extended Short-Time Objective Intelligibility
GAN	Generative Adversarial Network
GELU	Gaussian Error Linear Unit
GLU	Gated Linear Unit
GPU	Graphics Processing Unit
HA	Hearing Aid
HASPI	Hearing Aid Speech Perception Index
HL	Hearing Loss
HLS	Hearing Loss Simulation
HSR	Human Speech Recognition
IBM	Ideal Binary Mask
ISTFT	Inverse Short Time Fourier Transform
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error

MBSTOI Modified Binaural Short-Time Objective Intelligibility
MFCC Mel-Frequency Cepstral Coefficient
MHA Multi-Head Attention
MOS Mean Option Score
MSE Mean Squared Error
MUSHRA Multiple Anchor, Hidden Reference Assessment
NISQA Non-Intrusive Speech Quality Assessment
NN Neural Network
NNSE Neural Network Speech Enhancement
OLA Overlap-Add
PCM Pulse Code Modulation
PEMO-Q Perception Model-Based Quality
PESQ Perceptual Evaluation of Speech Quality
POLQA Perceptual Objective Listening Quality Analysis
PSD Power Spectral Density
RBM Restricted Boltzman Machine
RMSE Root Mean Squared Error
RNN Recurrent Neural Network
ReLU Rectified Linear Unit
SE Speech Enhancement
SEGAN Speech Enhancement Generative Adversarial Network
SI Speech Intelligibility
SI-SDR Scale Invariant Speech Distortion Ratio
SNR Signal-to-Noise-Ratio
SPIN SPeech In Noise
SQ Speech Quality
SSM Stuctured State Models
SSSR Self Supervised Speech Representation
STFT Short Time Fourier Transform
STOI Short-Time Objective Intelligibility
UDASE Unsupervised Domain Adaptation Speech Enhancement
VAD Voice Activity Detector
VB-D VoiceBank-DEMAND
WAV Waveform Audio File Format
WSJ0 Wall Street Journal
WER Word Error Rate

Part I

Introduction and Background

Chapter 1

Introduction

1.1 The Speech Enhancement Task

Since the advent of computerised speech processing in the 1960s (David & McDonald, 1956) many different techniques and use cases have emerged. These have had transformational effects on global society, expanding and changing the way we communicate with each other, and increasingly with our tools and machines. Typically these techniques are divided in two categories - *front-end* and *back-end* systems (Haeb-Umbach et al., 2021). The former deals with the initial processing of the input audio signal; applications which fall into this category include beamforming (Li et al., 2021), de-reverberation (Fu, Yu, Hung, et al., 2021), source separation (Ravenscroft et al., 2022; T. Sun et al., 2021), acoustic echo cancellation (Xiong et al., 2012), and Speech Enhancement (SE) (de Oliveira, Grinstein, et al., 2024; Fu et al., 2019; Richter, Welker, Lemercier, Lay, Peer, et al., 2024). The latter category (sometimes called the *downstream* task) encompasses the reasons why input speech is being processed, for example Automatic Speech Recognition (ASR), video conferencing or simply storage for later playback. It is important to design a pipeline of front-end and back-end systems which is appropriate to the recording environment and speakers being captured. The aim of the front-end should be to improve the performance of the back-end task. Figure 1.1 shows an example pipeline of an SE system in an scenario typical of those discussed in this work, wherein speech is recorded in a noisy environment by a single microphone channel. In this work, the class of system being targeted are front-end systems for the task of Speech Enhancement (SE) any for which *human perception* of the output is of critical interest. This includes *real-time* (causal) applications where the processing time of the audio must be short to minimise system delay such as online video or voice calls as well as non real-time (non-causal) applications such the post-recording processing of lecture recording audio. To further narrow the scope of interest, this work is concerned primarily with front-end systems where a single recording channel is available and where the distortion to the speech signal can be characterised as *additive* environmental noise. This class of front-end system has a long lasting history of research (Hendriks et al., 2013; Lim & Oppenheim, 1978) and has seen a significant increase in research interest and development in recent years (Babaev et al., 2024; Bulut & Koishida, 2020; Defossez et al., 2020; Kounovsky & Malek, 2017; Yen et al., 2023), as the world-wide changes in work patterns due to the COVID-19 pandemic continue. Remote working has become increasingly commonplace which necessities meeting conference software tools with robust SE capabilities to handle diverse working environments.

In the last decade, use of data driven approaches, namely Deep Neural Networks (DNNs) have exploded in popularity across the entire field of computer science (Bengio, 2009; Rumelhart et

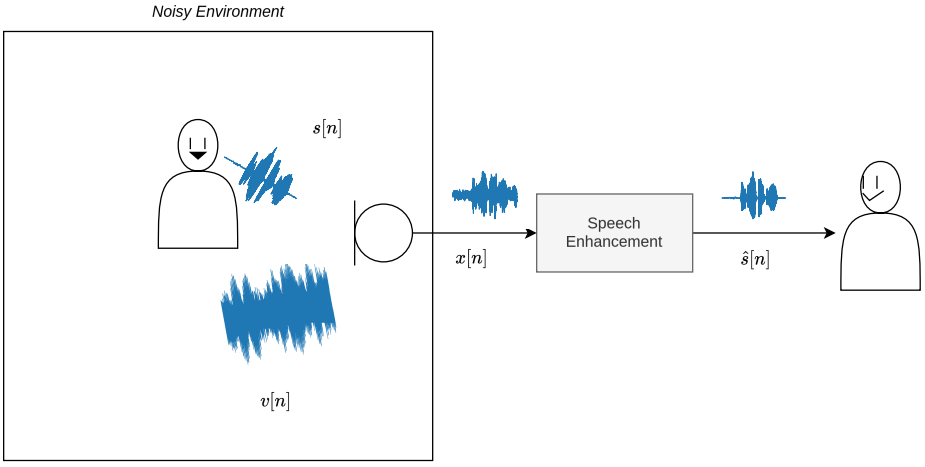

Figure 1.1: *Basic front-end pipeline for Speech Enhancement in a noisy environment with a single microphone.*
For more detail on the signal model used in this work, see Section 2.1.

al., 1986; Vaswani et al., 2017). Speech processing has not been untouched by this development, with the advent of Neural Network Speech Enhancement (NNSE) systems which significantly outperform traditional signal processing based approaches. In parallel, the metrics used to assess SE systems have similarly undergone drastic development thanks to DNNs (Kumar et al., 2023; Mittag et al., 2021; Reddy et al., 2022), however at a somewhat less breakneck pace; typically performance assessment of NNSE systems still rely on signal processing based metrics.

1.2 Motivation, Research Questions and Contributions

1.2.1 Towards A Unified View of Loss Functions and Metrics

Typically in NNSE literature, two classes of function for assessing audio signals are presented, the loss (or cost) function $\mathcal{L}(\cdot)$, and the evaluation metric $\mathcal{Q}(\cdot)$. In the standard supervised training setup, $\mathcal{L}(\cdot)$ takes as two inputs: a representation of the NNSE output audio signal $\hat{s}[n]$ and the corresponding clean reference signal $s[n]$. The purpose of the loss function $\mathcal{L}(\cdot)$ is to compare these two inputs in a mathematically differentiable way, such that the output loss value l can be *back-propagated* to the neural network, updating its parameters. For more information on NNs in general, see Section 2.4.

The purpose of the metric function $\mathcal{Q}(\cdot)$ on the other hand is to assess some aspect of the enhanced audio $\hat{s}[n]$ in order to evaluate the performance of the NNSE system. *Intrusive* metric functions typically take as input pairs $(s[n], \hat{s}[n])$, and return some metric value q which represents an assessment of the particular quantity of the speech signal that the metric function is designed to

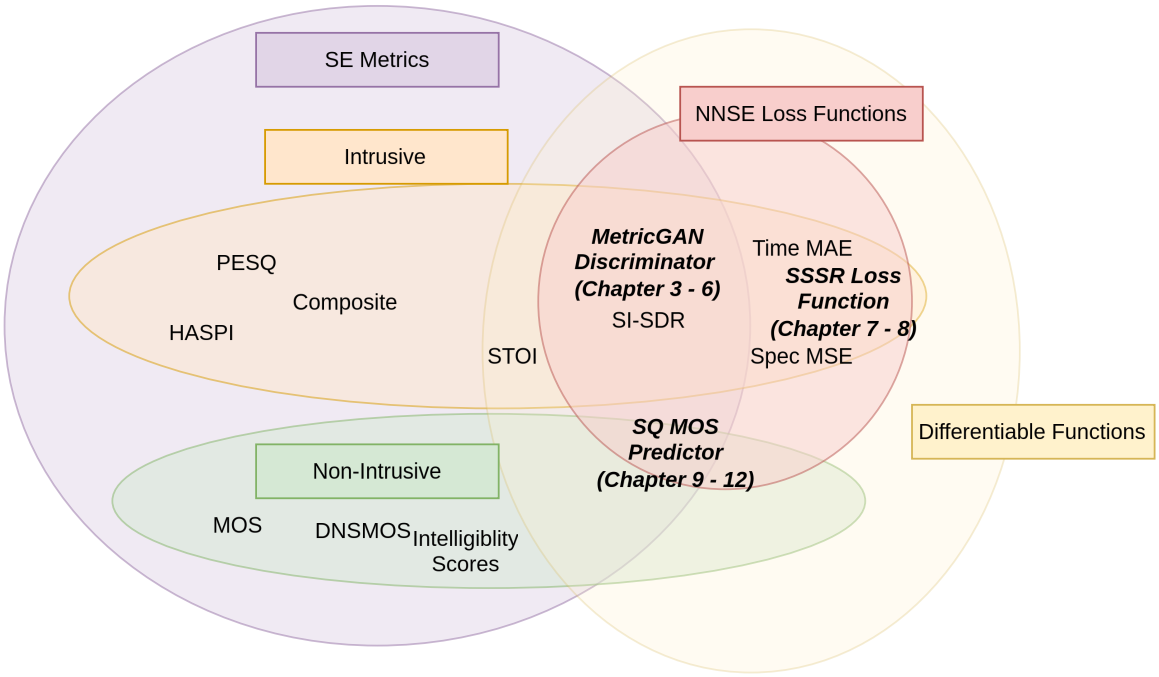

Figure 1.2: *Overview of SE metrics and NNSE loss functions*

assess. By comparing the metric score assigned to the noisy input $x[n]$ to that assigned to $\hat{s}[n]$, it is possible express how well the particular quantity of the speech audio has been improved by the enhancement system. For more detail on NNSE metrics see Section 2.7.

Taking a high level view of these two classes, some striking similarities can be observed. Both $\mathcal{L}(\cdot)$ and $\mathcal{Q}(\cdot)$ take some audio signal as input and return a single value, with that value representing the ability of the system which has processed that signal to enhance it. In fact, some functions such as the Scale Invariant Speech Distortion Ratio (SI-SDR) are commonly used as both loss function and evaluation metric. The major difference between the classes is the hard requirement that $\mathcal{L}(\cdot)$ has to be differentiable.

Figure 1.2 gives an overview of NNSE loss functions $\mathcal{L}(\cdot)$ and SE performance metrics $\mathcal{Q}(\cdot)$ used in this work in regards of their intrusiveness and differentiability. The purple circle contains functions of the class $\mathcal{Q}(\cdot)$, which fall into one of two types; intrusive metrics in orange or non-intrusive metrics in green. All loss functions $\mathcal{L}(\cdot)$ used in this work are shown in red which by definition also fall within the yellow area which denotes differentiability.

The unification of NNSE training objective and SE metric is not without problems. In the case of intrusive (i.e with reference) metric optimisation, there is the problem of 'overfitting' (becoming overly familiar with the training data) towards the score assigned by the metric (or a proxy of the metric). In this case, the score of the target metric of the output audio is high but is low in other, non-target metrics; in other words, the NNSE system has learnt to exploit flaws inherent to the computation of the metric. When the NNSE system is being optimised towards a non-intrusive MOS predictor, a similar issue can occur where the NNSE system learns to produce audio outside of the space of audio which the MOS predictor has observed during it's own training, rendering the predictor unable to properly assess it. Shown in bold in Figure 1.2 are the core interests of this work, along with the chapters in which they are explored. Firstly the creation of intrusive, differentiable predictors of intrusive SE metrics for use in NNSE training within the MetricGAN

framework. Secondly, the use of features derived from SSSR in intrusive loss functions for NNSE training. Finally, the creation of non-intrusive DNN based MOS predictors for both evaluation and training of NNSE systems.

1.2.2 Research Objectives and Questions

This thesis aims to satisfy three main research objectives, which correspond to the areas highlighted in Figure 1.2:

- **Objective 1** Develop extensions and improvements for an NNSE framework which involves an 'in the loop' metric prediction component. See Part II.
  - How can normally non-differentiable SE metrics can be incorporated into loss functions for the training of NNSE systems?
  - What methods to improve the ability of the metric prediction component of such systems to adapt to metric scores which do not appear in the training data can be devised?
  - How does use of DNN structures for NNSE which incorporate encoding of phase information effect performance of such systems?
- **Objective 2:** Investigate the use of Self Supervised Speech Representations (SSSRs) in NNSE training objectives. See Part III.
  - How can audio representations derived from pre-trained SSSR models be incorporated into loss functions for training NNSE systems?
  - To what degree do existing and proposed loss functions correlate with SE metrics and human MOS scores?
  - What is the the effect of the training data in terms of quantity and language of the audio used as well as the pre-training objective used in the creation of the SSSR models for this use?
- **Objective 3:** Explore the design and applications of DNN-based predictors of human assigned/derived labels of Speech Quality (SQ) and Speech Intelligibility (SI). See Part IV.
  - To what extent can traditional metrics be used as pre-training objectives for this task?
  - How does the nature of the features extracted from audio effect prediction performance?
  - What are the training data related issues associated with these tasks and how can different corpora be combined effectively?
  - How can inference of prediction models be incorporated into NNSE training?

1.2.3 Contributions

The contributions of this thesis have been published as a series of conference papers (Close, Hain, et al., 2022; Close, Hain, et al., 2023a, 2024, 2023b, 2023c; Close, Hollands, et al., 2022; Close, Ravenscroft, et al., 2023a, 2024, 2023b; Mogridge et al., 2024; Ravenscroft et al., 2024; Sutherland et al., 2024) and am (under review) journal paper, (Close et al., 2025). The following lists in brief the nature of these contributions, and the subsequent section of this thesis in which they are detailed.

1. In (Close, Hain, et al., 2022) an extension to the MetricGAN+ (Fu, Yu, Hsieh, et al., 2021) which introduces an additional 'de-generator' structure to the framework. The purpose of this extension is to widen the range of metric scores observed by the metric prediction component of the framework. The proposed system MetricGAN+/- outperforms the baseline MetricGAN+ on common test sets. See **Chapter 3** for further details.

2. In (Close, Hain, et al., 2023c) experiments involving advanced NNSE structures which are able to implicitly encode phase information are carried out, incorporating the findings of (Fu, Yu, Hsieh, et al., 2021) and (Close, Hain, et al., 2022). This work is detailed and expanded upon in **Chapter 4**; this chapter also explores techniques to reduce the training time overhead of the proposed systems, training and testing on a more realistic version of the training dataset and optimising towards an ASR system. The idea of ASR optimised NNSE training was further developed in (Ravenscroft et al., 2024).

3. In (Close, Ravenscroft, et al., 2023a) and (Close, Ravenscroft, et al., 2024) variations of the MetricGAN-U framework (Fu, Yu, Hung, et al., 2021) which incorporate the non-intrusive SQ prediction metric Deep Noise Suppression Mean Opinion Score (DNSMOS) are proposed, initially as an entry to the Computational Hearing in Multisource Environments 7 (CHiME7) Unsupervised Domain Adaptation Speech Enhancement (UDASE) challenge. **Chapter 5** and **Chapter 6** detail the proposed CMGAN+/+ and Multi-CMGAN+/+ systems respectively.

4. In (Close, Ravenscroft, et al., 2023b) the use of intermediate representations derived from SSSRs in NNSE loss functions is proposed which outperforms traditional spectrogram based losses. Further, the correlation between the proposed loss functions and intrusive SQ metrics is analysed as well as with human MOS labels. This work is detailed in **Chapter 7**.

5. As a follow-up to (Close, Ravenscroft, et al., 2023b), (Close, Hain, et al., 2023a) the nature of the SSSR used in the previously proposed SSSR loss functions is considered. In particular the language of the audio used to train the SSSR is investigated; to enable this, a framework for the generation of training, validation and test sets for NNSE systems in a number of languages is proposed. Results for NNSE systems trained and tested on these proposed datasets is detailed in **Chapter 8**. A related technique for the speech source separation task was proposed in (Ravenscroft et al., 2024).

6. In (Close, Hollands, et al., 2022) an entry to the Clarity Prediction Challenge 1 (CPC1) (Graetzer et al., 2020) is proposed. The proposed approach involves pre-training a DNN by predicting Speech Intelligibility (SI) metrics before fine-tuning on real human intelligibility values. Further, the correlation between real human intelligibility scores and intelligibility metrics is explored. The proposed system is detailed in **Chapter 9**.

7. A follow up work to (Close, Hollands, et al., 2022) which incorporates the use of SSSR feature representations for the SI prediction task was published as (Close, Hain, et al., 2023b) and is detailed in **Chapter 10**. An entry (Mogridge et al., 2024) to the Clarity Prediction Challenge 2 (Barker et al., 2024) which builds on this approach ranked second overall in the challenge. Further, a method for training hearing aid NNSE systems was proposed in (Sutherland et al., 2024).

8. In (Close et al., 2025) several models for the SQ prediction task are proposed which make use of input features derived from an SSSR and the Whisper ASR model (Radford et al., 2022). State-of-the-art performance on a common testset is achieved by use of such features.

Chapter 11 details the proposed models, along with several experiments investigating the training corpora combination, task variations and model structure.

9. Finally, in (Close, Hain, et al., 2024) a pre-trained SQ predictor like that proposed in Chapter 11 is incorporated into the loss function for an NNSE system. Potential problems with this approach are noted, and a small human listening test is performed. This contribution is detailed in **Chapter 12**.

1.3 Thesis Structure

This thesis is structured in 5 parts. In the remaining portion of Part I, the core tasks, concepts, datasets and baselines used in the subsequent parts are detailed. Each of the three subsequent parts aims to contribute to one of the research objectives detailed above. In Part II, variations and improvements on a baseline psychoacoustic metric motivated NNSE system are proposed. In Part III, the use of features derived from pre-trained foundational speech models in the training objective of NNSE systems is explored. Then, in Part IV approaches to the tasks of metric and human MOS/Intelligibility prediction are proposed, and used in the training objective of an NNSE system. Finally, Part V concludes the thesis with a brief summary and discussion of avenues for future work.

Chapter 2

Background

In this chapter, the background for the later experimental work is detailed. This comprises a description of the signal model for the core SE problem addressed in this work, an explanation of the acSE problem generally and the means by which is it assessed, an introduction to the DNN concepts used, data description and finally details of the core baseline systems .

2.1 Notation and Signal Model

This section will introduce the signal model and important notation which will be used throughout the remainder of this work.

2.1.1 Signal Model

The discrete speech signal $x[n]$ recorded by a single microphone in a noisy environment is given as

$$x[n] = s[n] + v[n] \tag{2.1}$$

where $s[n]$ is the desired speech signal, $v[n]$ is additive noise and n is the discrete time index $-\infty \leq n \leq \infty$

2.1.2 Single Channel Speech Enhancement

The goal of single channel SE system $\mathcal{G}(\cdot)$ is, given the microphone signal defined in (2.1), to return an estimation of the clean speech signal $s[n]$ denoted as $\hat{s[n]}$. This enhanced output of $\mathcal{G}(\cdot)$, $\hat{s}[n]$ is an a estimation of $s[n]$ is given as

$$\hat{s}[n] = \mathcal{G}(x[n]) \approx s[n] \tag{2.2}$$

2.1.3 DNN Metric Prediction

The predicted speech aspect value q as predicted by a non-intrusive DNN speech metric prediction network $\mathcal{D}$ is given as

$$\hat{q} = \mathcal{D}(x[n]) \tag{2.3}$$

with intrusive DNN prediction given similarly as

$$\hat{q} = \mathcal{D}(s[n], x[n]) \tag{2.4}$$

.

2.1.4 STFT Features

Throughout this thesis, a number of speech signal representations derived from the Short Time Fourier Transform (STFT) are used as depicted in Figure 2.1. From a time domain speech audio

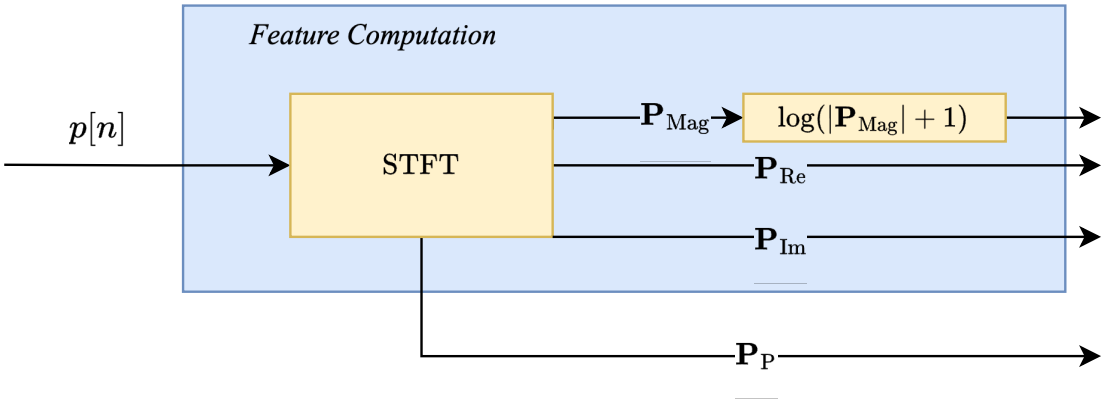

Figure 2.1: *Illustration of STFT feature computation. The variable $p[n]$ is a placeholder for any signal, i.e $p[n] \in \{x[n], s[n], \hat{s}[n]\}$*

signal $p[n]$, complex time-spectral features $\mathbf{P} \in \mathbb{C}^{F_{\text{Hz}} \times T}$ are calculated using the STFT of length (i.e the number of samples being transformed in each window) F_{Hz} for each of T frames of a placeholder time domain signal $p[n]$, to obtain real part $\mathbf{P}_{\text{Re}}$ and imaginary part $\mathbf{P}_{\text{Im}}$. From these, a magnitude spectrogram $\mathbf{P}_{\text{Mag}}$ and phase information $\mathbf{P}_{\text{P}}$ can be computed. Further, an additional processing step applied depending on the system either $\log(\mathbf{P}_{\text{Mag}} + 1)$ or a power law compression. Specifically, each feature is obtained as follows:

$$\mathbf{P} = \sum_{n=-\infty}^{\infty} p[n] \cdot h[n-m] \cdot e^{-j2\pi f n} \tag{2.5}$$

where:

- $\mathbf{P}$ is the STFT spectrogram matrix of the signal $p[n]$
- m is the discrete time index of the windowed signal.
- f is the frequency variable.

- h is a (Hanning) window function centred around time n.
- $e^{-j2\pi fn}$ is the complex exponential function representing the Fourier kernel.
- j is the imaginary unit, $j = \sqrt{-1}$.

$\mathbf{P}$ is complex valued, from which the real component $\mathbf{P}_{\mathrm{Re}}$ and imaginary component $\mathbf{P}_{\mathrm{Im}}$ can be obtained. The magnitude spectrogram $\mathbf{P}_{\mathrm{Mag}}$ can be obtained by:

$$\mathbf{P}_{\mathrm{Mag}} = \sqrt{\mathbf{P}_{\mathrm{Re}}^2 + \mathbf{P}_{\mathrm{Im}}^2} \tag{2.6}$$

The phase representation $\mathbf{P}_{\mathrm{P}}$ is calculated as the argument (or angle) of the complex number:

$$\mathbf{P}_{\mathrm{P}} = \arctan(\mathbf{P}_{\mathrm{Im}}, \mathbf{P}_{\mathrm{Re}}) \tag{2.7}$$

where $\arctan(y, x)$ is the four-quadrant inverse tangent function that computes the angle θ from the conversion of rectangular coordinates (x, y) to polar coordinates (r, θ).

2.2 Speech in Noisy Environments

When speech recordings are made, in addition to the desired speech signal, various corrupting aspects (noise) of the recording environment are also captured. A number of factors including the distance between the speaker and the microphone, and the nature of the recording environment affects the degree of this corruption. *This corrupting noise can have a detrimental effect on the downstream task.*

Figure 2.2 shows visual representations of a 'clean' speech signal $s[n]$ (left panel) recorded in a controlled environment with minimal background noise alongside a 'noisy' version $x[n]$ (right panel) of that same speech signal which has had background noise $v[n]$ artificially added. The upper representations are time domain waveforms which show the amplitude of the signals over time; from these the corrupting effect of the noise in the speech signal can be observed as additional 'spikes' in signal content e.g in the first 0.5 seconds. The lower representations show the frequency domain magnitude spectrograms computed via a STFT. To obtain this frequency domain representation, first the time domain signal is divided into overlapping segments, usually between 10ms and 30ms in duration. Then an analysis window function (typically a Hanning or Hamming window) is applied to each segment, and the frequency representation of each is computed using the Fast Fourier Transform (FFT) algorithm. The magnitude of the energy in each frequency *bin* (as encoded vertically on the spectrogram representation) is expressed on a decibel dB scale. These spectogram show clearly where the noise is present in the non speech regions surrounding the speech and also where the noise obfuscates the speech spectrally.

As shown in Figure 2.3 different environments have their own unique noise characteristics and thus effect the speech signals in different ways. The amount of noise present in a signal can be expressed as the Signal-to-Noise-Ratio (SNR) (cf. Section 2.7.2). The higher the SNR value, the lesser the effect of the noise on the speech signal. As shown in Figure 2.4, the lower the SNR value, the less distinct the speech signal is from the the noise in the spectrogram.

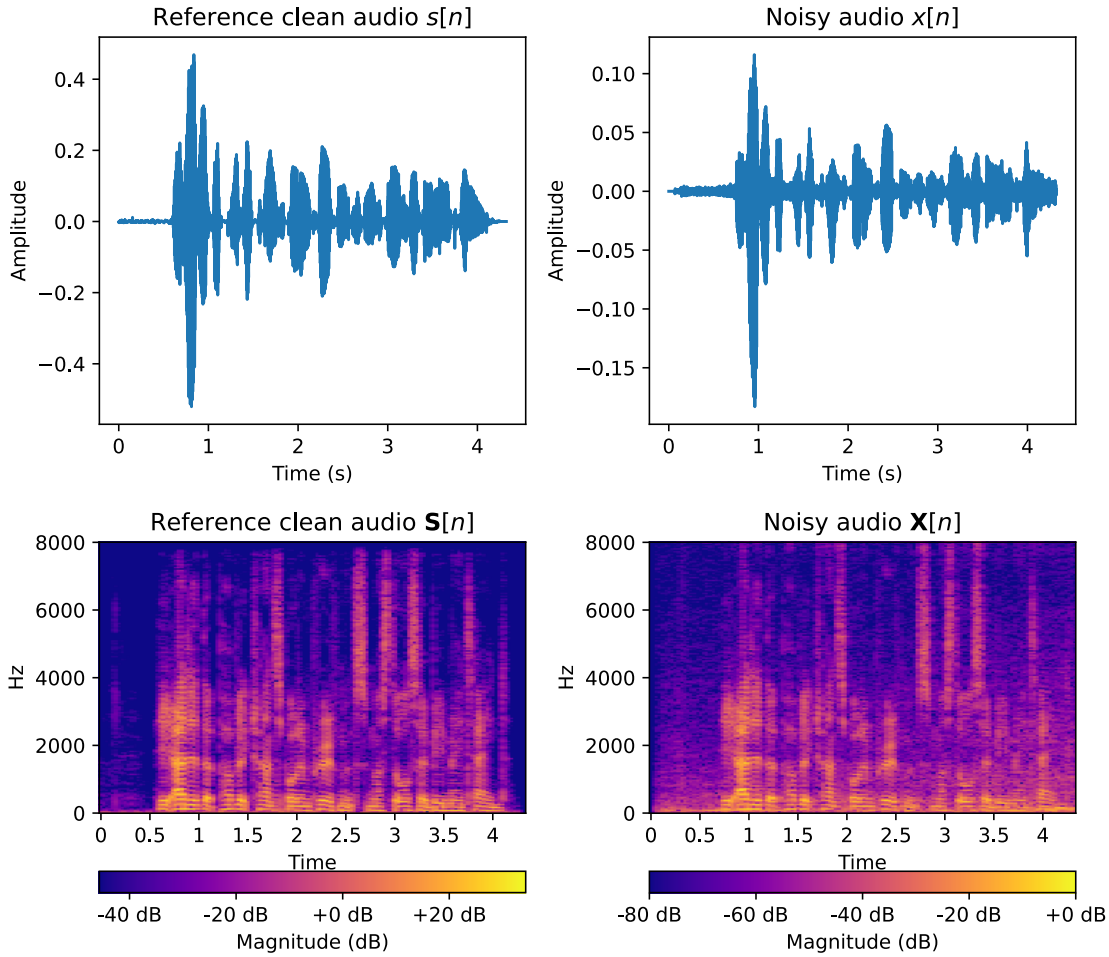

Figure 2.2: *Waveform (top) and spectrogram (bottom) representations of clean (left) and noisy (right) speech, sourced from the VoiceBank-DEMAND (Valentini-Botinhao et al., 2016) dataset.*

2.2.1 Digital Audio Representation

Typically in digital systems, audio is captured or stored in a linear Pulse Code Modulation (PCM) (Oliver et al., 1948) format as an array of *amplitude* values along with a sample rate which describes the rate of playback in terms of number of samples to be processed in one second. For example, an audio file consisting of 48000 samples with a sample rate of 16000Hz would last $48000/16000 = 3$ seconds. The higher the sample rate the higher the effective *resolution* of the audio recording, at the cost of higher storage requirement. Typically, 16000Hz (16kHz) is considered a reasonable rate capturing the nuance of speech audio. Audio at some given sample rate can be *downsampled* to some lower sample rate or (less commonly and more problematically) *upsampled* to some higher rate. In this work, the uncompressed Waveform Audio File Format (WAV) (Fleischman, 1998) is used throughout.

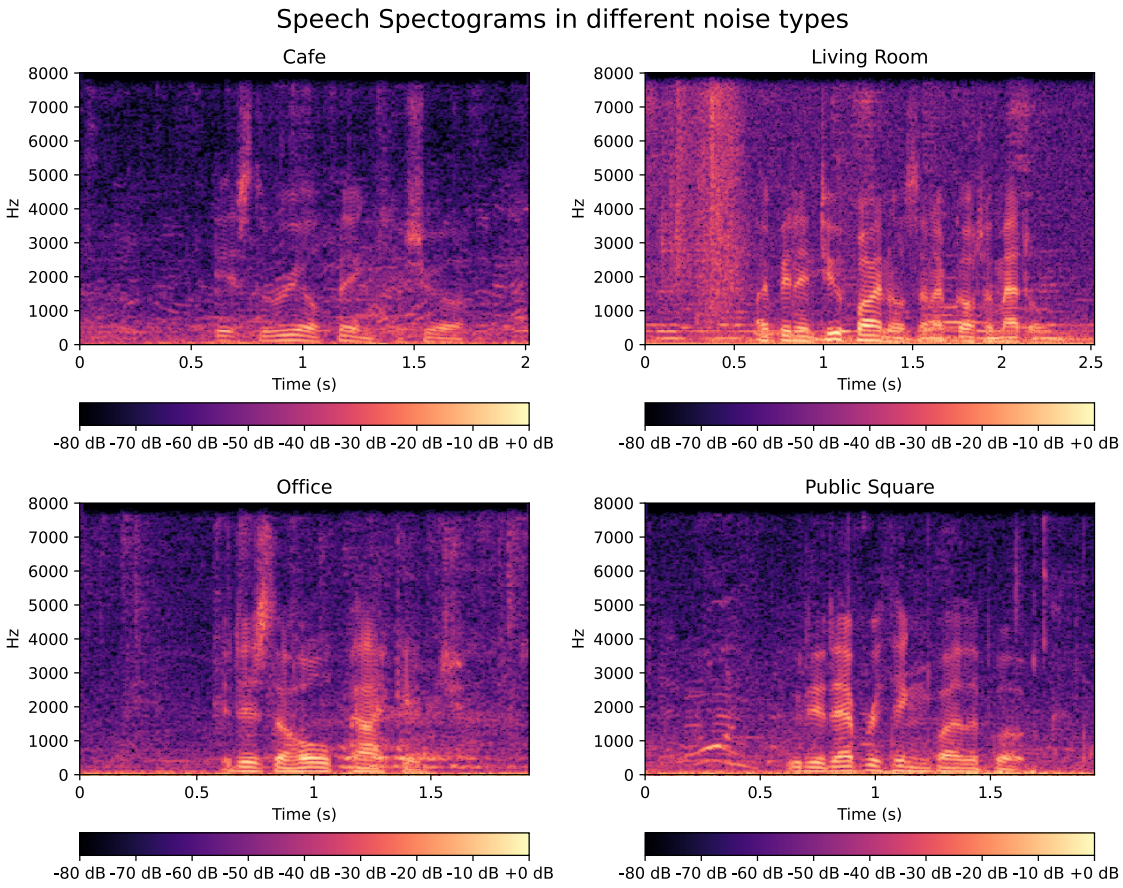

Figure 2.3: *Speech spectrograms in four different noise environments, all with a SNR of* $2.5dB$

2.3 Speech Enhancement Algorithms

The aim of speech enhancement algorithms is to process the speech in such a way that the portion of the microphone signal which contains target speech is retained and enhanced, while all other noisy portions of the signal are reduced or removed. This can be achieved with the aim of improving the human perception of the quality or intelligibility of the speech signal or to reduce the errors in the automatic computer processing, e.g. transcription.

2.3.1 Signal Processing Based Speech Enhancement

Traditionally, this task has been approached using a signal processing based solution wherein the noise portion of the input signal is probabilistically estimated and attenuated from the signal. Such approaches often assume that the speech and noise parts of the signal are uncorrelated. However, there are several situations where the parts of the signal which are being treated as noise are in reality highly correlated with the target speech i.e when the input signal contains competing speech signals. Figure 2.5 shows clean, noisy, and enhanced speech using a traditional noise reduction algorithm (Hendriks et al., 2013). This algorithm has three component steps; first a frequency

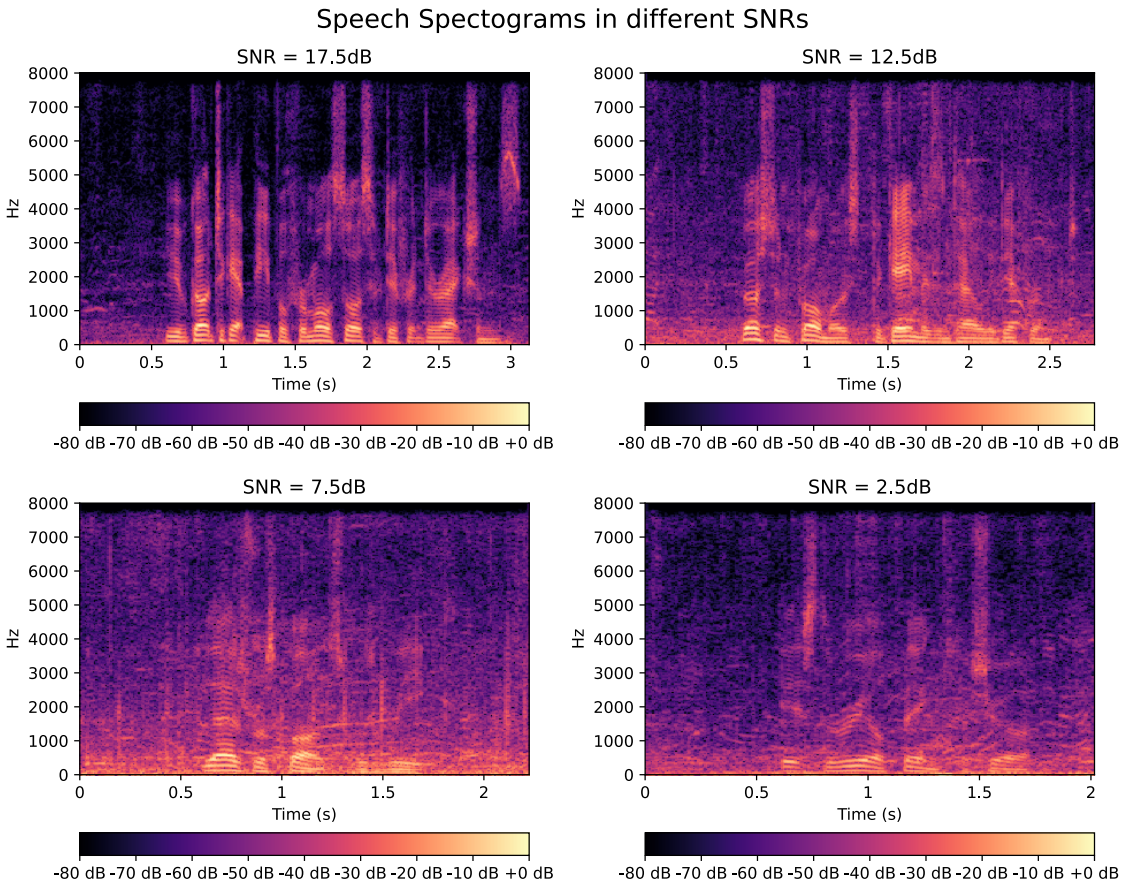

Figure 2.4: *Speech spectrograms in four different SNRs, all noise is from a cafe environment sourced from the VoiceBank-DEMAND (Valentini-Botinhao et al., 2016) dataset*

domain representation is computed from the input time domain signal using a magnitude discrete Fourier Transform (MDFT). This is followed by a estimation of the noise Power Spectral Density (PSD) using a per frame prediction of the presence of speech. Finally the PSD of the speech is computed using the noise PSD.

While the noise has clearly been reduced, as shown in the dark regions of the enhanced spectrogram, the algorithm has created *spots* of distortion which are easily visible in the non-speech regions. These manifest as audible 'musical tones' when the enhanced signal is played back. Such can distortions have a detrimental effect on the downstream task; for example, speech which has been enhanced in such a way can sometimes significantly reduce performance of ASR, as important parts of the signal can be destroyed by the enhancement. Further, such distortions can also degrade the human perception of the enhanced audio.

2.3.2 DNN-based Speech Enhancement

In recent years, Neural Network Speech Enhancement (NNSE) has become increasingly popular and has shown increased performance compared to the traditional methods. These approaches are 'data driven', meaning that their creation requires access to a large amount of noisy speech data.

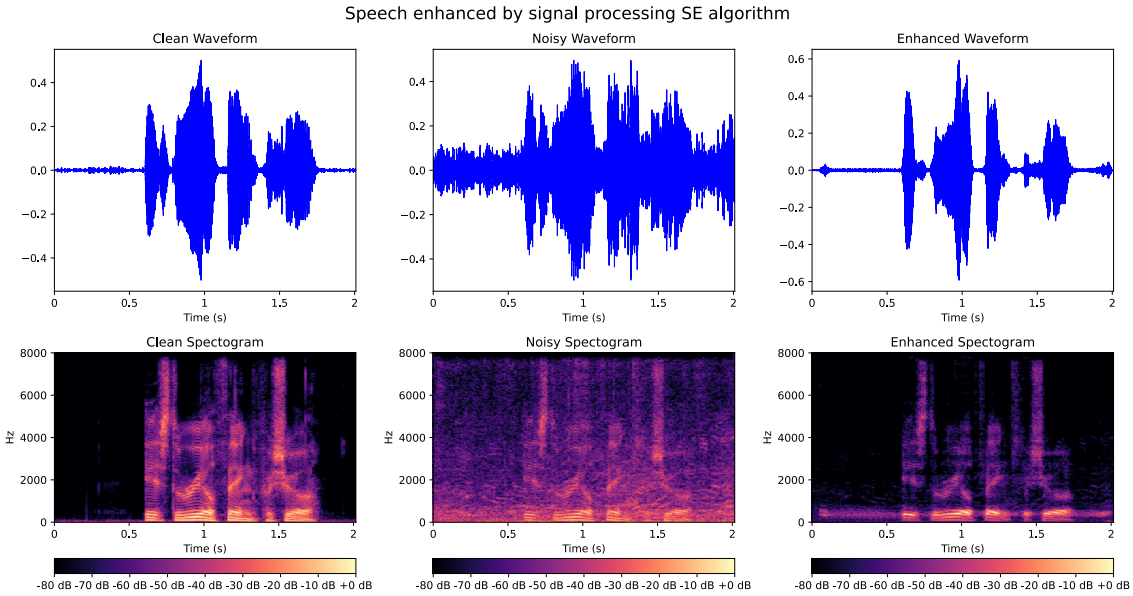

Figure 2.5: *Noisy speech signal enhanced via traditional signal processing method*

Additionally, DNN speech enhancement systems are typically trained (their parameters fitted) in a 'supervised' manner which requires access to a clean 'reference' audio of the noisy audio. As such, the training data usually requires artificially simulated noisy audio. There are two primary techniques for NNSE networks, mapping and masking (L. Sun et al., 2017), shown in Figure 2.6 and Figure 2.7, respectively. In the former, the output of the network is itself an enhanced version of the input representation which can be directly transformed into the enhanced audio signal. In the latter, the output is a so called 'mask' which, when multiplied with the noisy input representation,results in the enhanced representation.

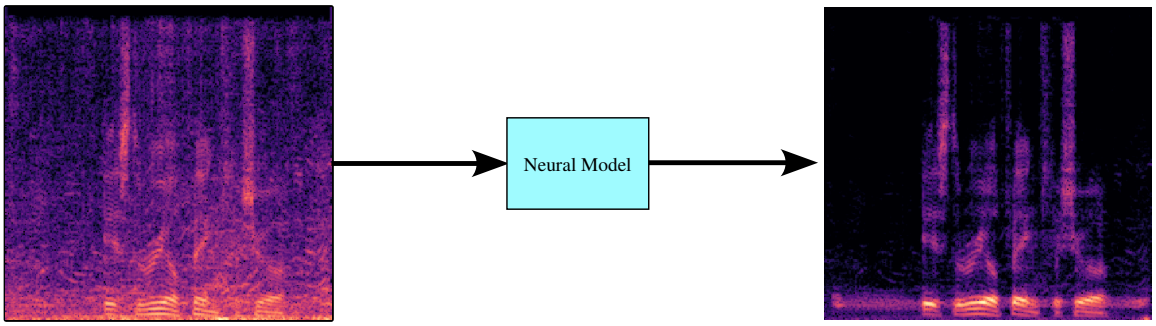

Figure 2.6: *NNSE using a mapping approach*

In early forays into mapping based NNSE (Y. Xu et al., 2014) a basic form of DNN, (a stack of Restricted Boltzman Machines (RBMs)) is pre-trained in an unsupervised approach using Contrastive Divergence (CD) (Bengio, 2009) over noisy data, before supervised fine-tuning using a STFT domain loss function (cf.. (2.39)). A common objective in early NNSE masking based systems (Y. Wang et al., 2014) was the prediction of an Ideal Binary Mask (IBM), defined for a

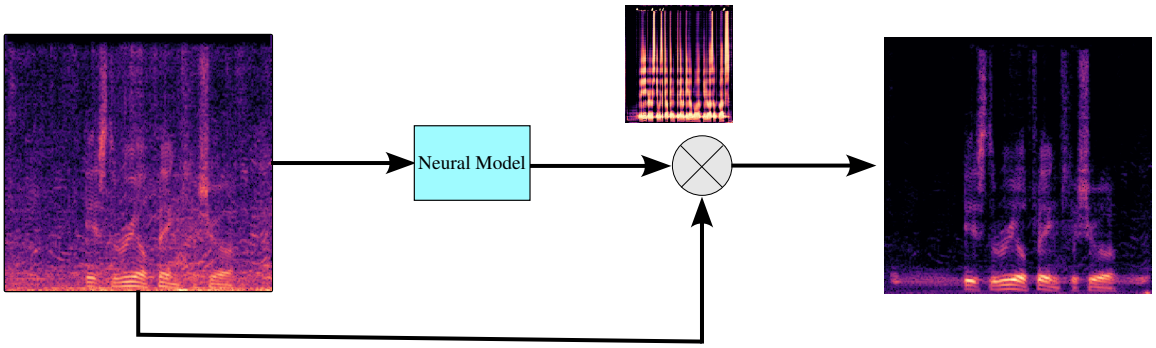

Figure 2.7: *NNSE using a masking approach*

simulated noisy mixture magnitude STFT representation $\mathbf{X}_{\text{Mag}}$

$$\text{IBM}(\mathbf{X}_{\text{Mag}}) = \begin{cases} 1, & \text{if } \text{SNR}_{\text{dB}}(\mathbf{X}_{\text{Mag}}) > \text{LC} \\ 0, & \text{otherwise,} \end{cases} \tag{2.8}$$

where LC is a 'local criterion' usually set to be 5dB smaller than the SNR of the mixture.
Mapping and masking approaches were compared in (Kounovsky & Malek, 2017), using then novel CNN layers in an NNSE network; it was found here that mapping based approaches outperform masking; however contemporaneous work (Y. Wang et al., 2014) concluded that the opposite is true. Generally speaking, mapping approaches have proved to be the most widespread, but are typically trained with loss functions directly involving the enhanced synthesised audio $\hat{s}[n]$ rather than being directly trained using a target mask. Some recent methods involve the combination of both techniques (Cao et al., 2022; Dang et al., 2022), such that the noisy magnitude input is masked, while the enhanced real and imaginary components are mapped.

2.3.3 Speech Enhancement Generative Adversarial Network (SEGAN)

The standard Generative Adversarial Network (GAN) (cf. Section 2.6) structure has been applied to the SE task, where the Discriminator is tasked with distinguishing between time domain outputs of the speech enhancement Generator and clean time domain reference signals (Pascual et al., 2017). A Least Square GAN (LSGAN) (Mao et al., 2016) approach is taken, such that the loss of the Discriminator $\mathcal{D}$ is

$$L_{\mathcal{D}_{\text{SEGAN}}} = \frac{1}{2}[(\mathcal{D}(s[n]) - 1)^2] + \frac{1}{2}[\mathcal{D}((\mathcal{G}(x[n]))^2] \tag{2.9}$$

, while that of the Generator G is

$$L_{\mathcal{G}_{\text{SEGAN}}} = \frac{1}{2}[(\mathcal{D}(\mathcal{G}(x[n]) - 1)^2)] + \lambda_{\text{SEGAN}}[||\mathcal{G}(x[n]) - s[n]||_1] \tag{2.10}$$

with λ_{SEGAN} being a hyper-parameter controlling the weighting of the L1 norm term. However this loss formulation has no relation to human perception; to overcome this limitation, a modification to the GAN structure has been developed which is designed to incorporate measures of human perception or any other conceivable signal measure - MetricGAN (Fu et al., 2019) (cf. Section 2.12).

2.3.4 Metric Derived Objective Function

In both mapping and masking approaches explored in the papers described above and many others (X. Lu et al., 2013; Y. Xu et al., 2015) the objective function of the learning component being minimised is usually a MSE between the output of the network $\hat{s}[n]$ and some oracle mask or the clean reference $s[n]$. In (Fu, Wang, et al., 2018) it is noted that a small MSE distance does not always correlate with surrogate measures of human perception such as PESQ and STOI or with machine perception in the form of ASR performance. This suggests that it is a poor objective function for this task. Thus the paper (Fu, Wang, et al., 2018) proposes a NNSE system that is optimised with STOI as its objective:

$$\mathcal{L}_{\text{STOI}} = -\text{STOI}(\hat{s}[n], s[n]) - 1 \tag{2.11}$$

The STOI function is a specially designed version of the function that allows it to be back-propagated as it is implemented in a differentiable way. This NNSE system trained with the STOI objective outperformed both utterance based MSE NNSE system in terms of STOI on the TIMIT (Garofolo et al., 1993) dataset. However, this resulted in a degraded PESQ score compared to the MSE objective models. The paper further proposes a mixture objective of time domain MSE and STOI together:

$$L_{\text{STOI+MSE}} = [\text{STOI}(\hat{s}[n], s[n]) - 1] + \alpha_{\text{STOI-loss}} \sum_{n}^{N} (s[n] - \hat{s}[n])^2] \tag{2.12}$$

where $\alpha_{\text{STOI-loss}}$ is a weighing factor between the targets. This seems to balance out the PESQ and STOI scores, as well as giving better ASR performance. While each of the steps in the computation of the STOI score can be differentiated, allowing it to be directly used as an objective function, it is a complex calculation. Furthermore, optimising for objectives more complex than STOI in this way is a challenge.

2.4 Deep Neural Networks

Despite their recent ubiquity, Deep Neural Network (DNN) are not a new idea with some of the first proposed as early as the 1950s (Kleene, 1956). The resurgence of interest and application in recent years is due in part to hardware and software advances in Graphics Processing Unit (GPU) systems which can be used to efficiently train (i.e fit the parameters of) DNNs increased availability of the large amount of data required thanks to increasingly faster and cheaper storage technology and the data collection efforts of the research community. Training of DNNs can fall into two broad categories,*supervised* and *unsupervised*, based on the nature of the loss function/training objective. In the supervised setting, the DNN is provided with paired data of input and label; for example the training data for ASR systems typically consists of speech audio paired with a text transcription or for SE pairs of matched clean and noisy signals $(s[n], x[n])$. The task of the DNN in this setting is to map from the input to the label. An overview of supervised DNN training is depicted in Figure 2.8.

In the unsupervised setting the network is provided with unlabelled data, and the objective of the network is to find underlying structure and patterns within the data. In the training of unsupervised DNNs the training objective is derived from some secondary statistic or representation of the data itself. An overview of unsupervised DNN training is depicted in Figure 2.9. Note that all DNN systems used in this work should more correctly be termed "Artificial Deep Neural Networks"

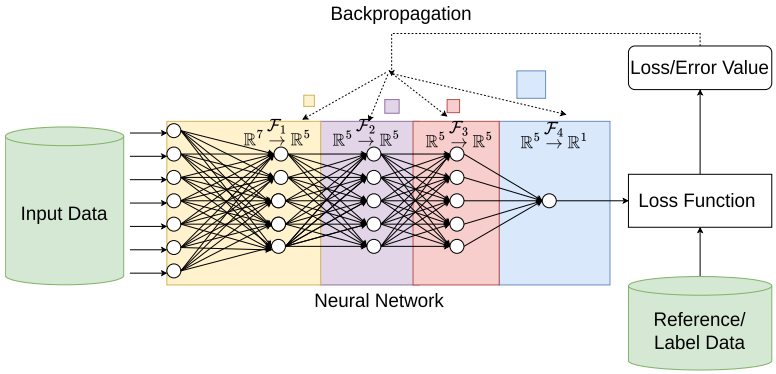

Figure 2.8: *Supervised DNN Training.*

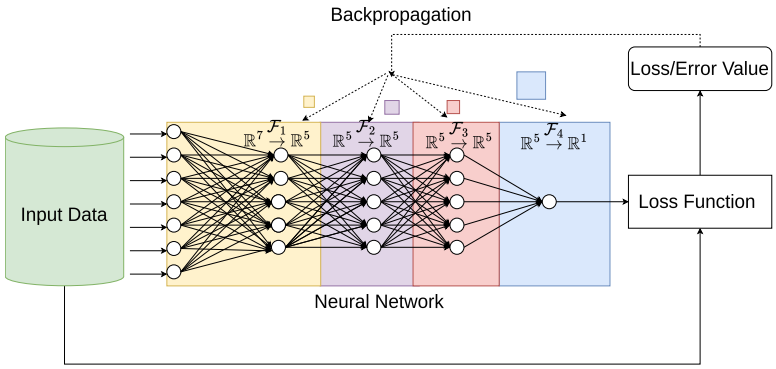

Figure 2.9: *Unsupervised DNN Training.*

to distinguish them from vastly more complex biological neural networks (e.g. animal cognitive functions). This qualifier is omitted for the sake of clarity.

There are three core aspects to the design and creation of all DNN systems:

- Network structure
- Loss function and training objective
- Training, validation and testing setup

This section will aim to describe each of these in the following, such that the specific details for proposed DNN systems for SE and related tasks can better be understood.

2.4.1 Neural Network Model Structure

Neural networks generally consist of a number of *layers* of parameters, often ordered sequentially, such that the output of one layer is the input to the next. These take many forms, with the simplest being a *linear* (or *fully-connected*) layer. In a linear layer $\mathcal{F}$, the input representation y_{n-1} is multiplied by one set of weights and then a 'bias' value added to each:

$$y_n = \mathcal{F}_n(y_{n-1}) = \mathbf{W}y_{n-1} + b \tag{2.13}$$

where y_n is the output of the layer n, y_{n-1} is the output of the previous layer and $\mathbf{W}$ and b are the weight matrix and bias value of the layer $\mathcal{F}$ respectively. The values of $\mathbf{W}$ and b are called *parameters* and are updated during the training of the model, following a (random) initialisation. Linear layers can map to output dimensions of different sizes to their input, such that an input $y_{n-1} \in \mathbb{R}^I$ can be mapped to an output $y_n \in \mathbb{R}^O$ by a linear layer with weight matrix $\mathbf{W} \in \mathbb{R}^{I \times O}$. Figure 2.10 depicts a simple DNN consisting of 4 Linear Layers. The first of these $\mathcal{F}_1$ maps from an

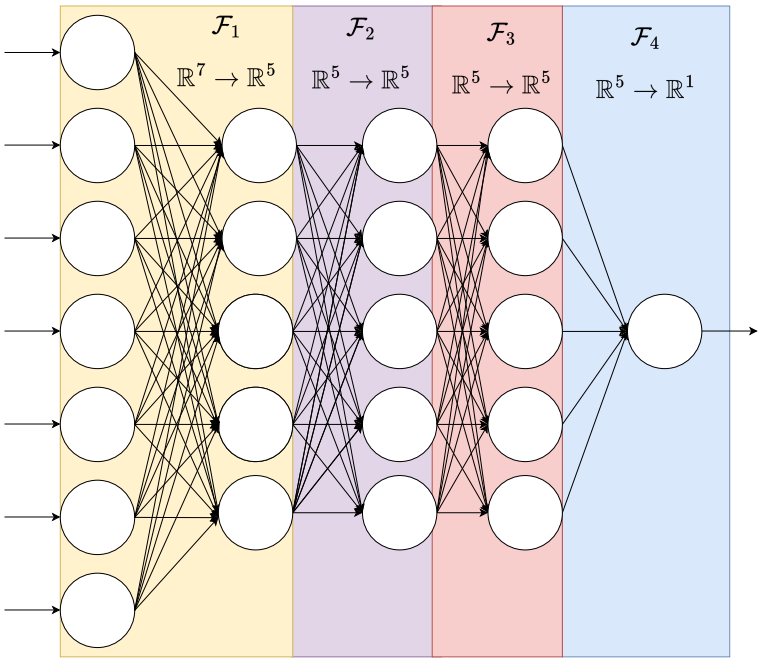

Figure 2.10: *A simple DNN consisting of four sequential Linear Layers with no activation functions.*

input representation $y_0 \in \mathbb{R}^7$ to $y_1 \in \mathbb{R}^5$. The whole DNN depicted in Figure 2.10 can be expressed as a chain of layers:

$$\mathcal{F}_{\text{DNN}}(y_0) = \mathcal{F}_4(\mathcal{F}_3(\mathcal{F}_2(\mathcal{F}_1(y_0)))) \tag{2.14}$$

such that $\mathcal{F}_{\text{DNN}} : \mathbb{R}^7 \to \mathbb{R}^1$. The final layer is referred to as the *output layer* and in supervised training its output typically represents the models prediction of a ground truth label which regards to the input data. The input to the first layer are called the *input features*. For example, if the task was predicting the probability that it might rain on a given day, the 7 dimensional input feature might encode information about the previous day's precipitation, the current cloud cover, temperature, date, etc. The single output *neuron* will contain the predicted probability of rain, given the specific input features.

The layers between the input and output are called *hidden layers* and their outputs *hidden units* as unlike the input and output layers their behaviour is not directly controlled by the designer of the network. In the example DNN in Figure 2.10, layers $\mathcal{F}_2$ and $\mathcal{F}_3$ are hidden layers. During training, the values of the parameters of each layer are updated based on the loss value computed by the loss (objective) function using a *backpropagation* algorithm (Deisenroth et al., 2020). Each instance of inference of a DNN during training is called a *forward pass*, while the parameter update step is call the *backward pass*.

The *parameter count* of a DNN refers total number of learnable parameters in the DNN model.

For example, layer $\mathcal{F}_1$ of the simple DNN in Figure 2.10 will have a weight matrix $\mathbf{W}$ of size $7 \cdot 5 = 35$ plus 5 bias values b giving a total of 40 parameters. Similarly, hidden layers $\mathcal{F}_2$ and $\mathcal{F}_3$ have $5 \cdot 5 + 5 = 30$ parameters each, while the output layer $\mathcal{F}_4$ contributes $5 \cdot 1 + 1 = 6$ learnable parameters. Overall the parameter count of $\mathcal{F}_{\text{DNN}}$ is $40 + 30 + 30 + 6 = 106$. The *size* of a model as represented by its total parameter count is of crucial importance in both training and inference of a DNN. A model which is too small might be unable to learn the complex relationships between the input data and the target. However, a model which is too large might be prone to overfiting by learning exactly the content of the training data. Of particular importance for most NNSE systems is that inference of the DNN be quick enough to handle real-time (i.e be able to process a input sequence in less time than the length of the input in time) processing of the input audio for applications such as video conferencing where the delay introduced by the system is critical.

2.4.1.1 CNN

Convolutional Neural Network (CNN) (LeCun et al., 1989; O'Shea & Nash, 2015), or Convolutional layers are an specialised form of linear layer. Each CNN layer is composed of a number of filters or kernels. In the case of 1-dimensional Convolution, each filter 'slides' across the width of the input vector, computing a scalar dot product value between the weights of the kernel and the input then adding the bias term. The *stride* of a 1D Convolutional layer refers to the number of points the kernel slides to obtain the next value in the output. Figure 2.11 depicts a single 1D CNN filter of length 3 with differing stride values over an input $x[n]$ of length 9 to obtain convolutional output $c[n]$; the greater the stride value, the greater the level of sub-sampling of the input. The *dilation* of a 1D Convolutional layer refers to how much the filter 'skips' elements of the input to produce the next output sequence element; Figure 2.12 shows a 1D CNN filter of length 3 and stride 3 with dilation of 1 (top) and 2 (bottom).

It is often desirable to preserve the size of an input through a CNN layer; to do this *zero padding* is used. As the name suggests, this is the practise of appending values of 0 to the beginning or end of the input sequence such that the length of the output is the same as the input when using a stride of 1. Figure 2.13 shows an example of zero-padding to ensure that the output of the filter $c[n]$ with length 3 and stride 1 has the same length as the input $x[n]$

In the case of a 2-dimensional Convolution, each 2-dimensional filter slides across both the width and height of the input matrix. All of the above properties of a 1D CNN also apply to the 2D case. The size of a 2D CNN filter is expressed by a width and a height, and its stride by a movement across the width and height of the input. The dilation and padding of a 2D CNN filter are defined similarly. Figure 2.14 depicts two 2D CNN filters with differing sizes and strides. When several CNN layers are chained together sequentially, a *pooling layer* is typically inserted between the layers. The purpose of this pooling layer is to down-sample the filter output representations to reduce the number of parameters in order to reduce overfitting or improve inference latency. One form of pooling is *average pooling*; here a layer slides across the output of a filter, returning an average value of the region under the pooling layer. Another similar approach is *max pooling* where only the maximum value of the region under the filter is output. Figure 2.15 shows average and max pooling with a pooling length of 2, halving the size of the filter output.

In relation to audio processing and SE, 1D CNN layers are more commonly used when the input to the DNN is 1-dimensional i.e when it is time domain audio (Luo & Mesgarani, 2019). 2D CNN

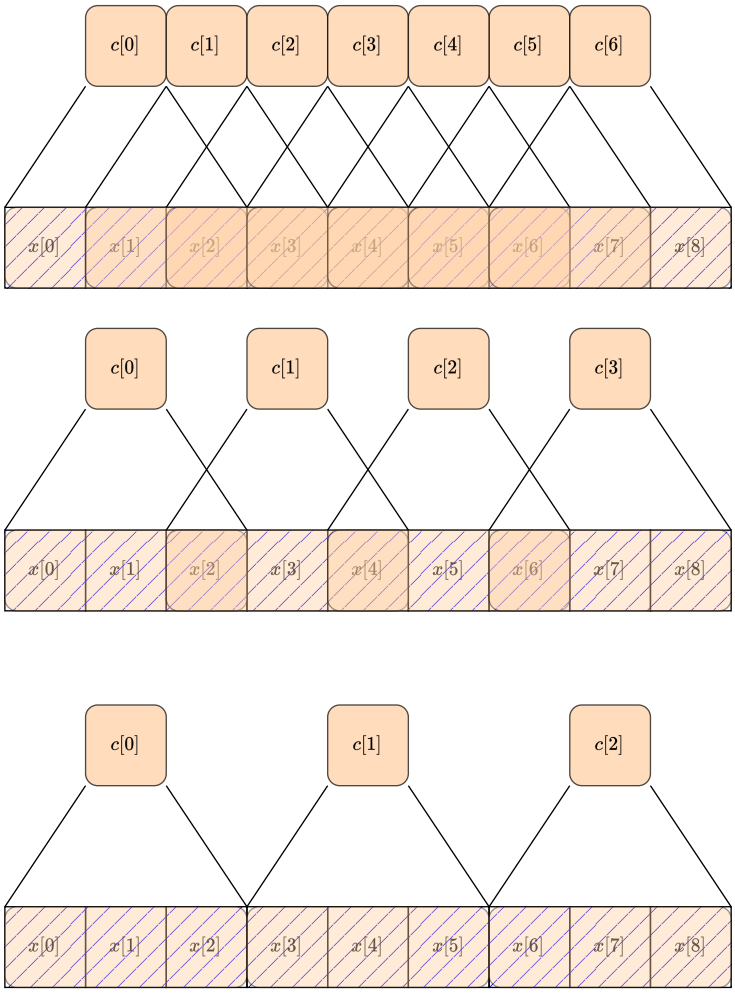

Figure 2.11: *A 1D CNN filter of length* 3 *over an input sequence of length* 9 *with a stride of* 1 *(top),*2 *(middle) and* 3 *(bottom) respectively.*

layers are used when the input is a STFT domain representations with a time (length) and frequency (height) dimension.

2.4.1.2 RNN and Long Short-Term Memory (LSTM)

Recurrent Neural Network (RNN) (Rumelhart & McClelland, 1987) layers have feedback connections which allow them to retain a *memory* of past inputs. As input, an RNN takes in the data and the previous *hidden state* which are processed by the layer. A simple RNN layer or cell can be expressed as

$$h_n = \mathcal{F}(y_{n-1}, h_{n-1}) = \tanh(y_{n-1}\mathbf{W}_y + b_y + h_{n-1}\mathbf{W}_h + b_h) \tag{2.15}$$

such that the layer output y_n of the layer is a weighted and biased sum of the current input y_{n-1} and the past output of the layer h_{n-1}. Typically, a tanh non-linearity is applied to the output. Figure 2.16 shows a single RNN layer over three time steps n. A Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) unit is an variant of the RNN structure which introduces gates which control the flow of information within the network layer. The introduction of these gates is intended to

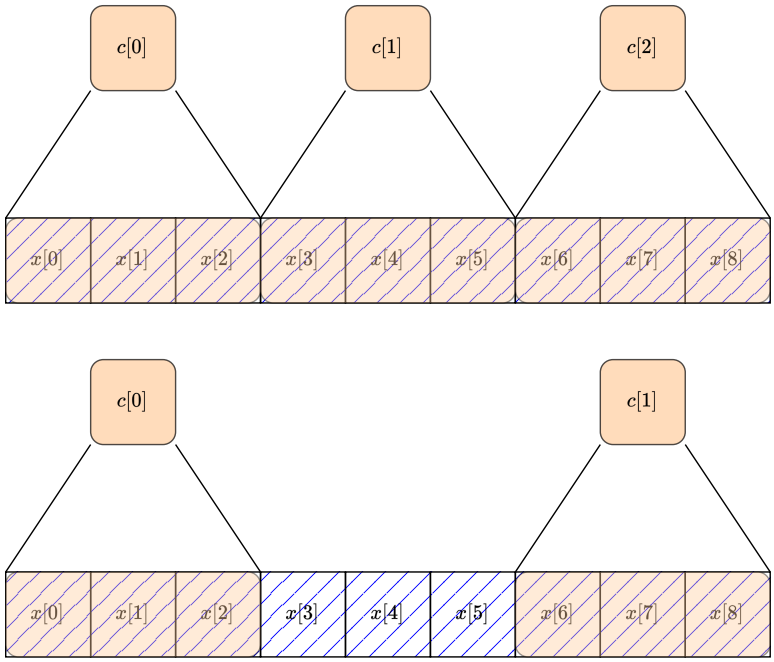

Figure 2.12: *A 1D CNN filter of length* 3 *and stide* 3 *over an input sequence of length* 9 *with a dilation of* 1 *(top),* 2 *(bottom) respectively.*

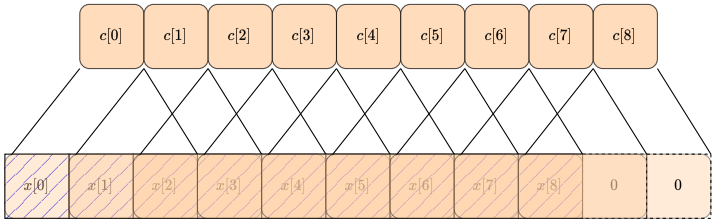

Figure 2.13: *A 1D CNN filter of length* 3 *and stride* 1 *with left side padding of* 2

address the *vanishing gradient problem* where long-term gradients can vanish as the number of computations increases. Specifically, an input, output and forget gate are introduced. The forget gate is responsible for deciding what information from the previous hidden state to discard, while the input and output gates control which parts of the input and output data to store within the new last hidden state. These gates are implemented similarly to the core RNN function (2.15), except with a sigmoid non-linearity. The LSTM cell state lstm_t is the sum of two element wise products of the forget gate with the prior cell state and the input gate with the output of the standard RNN cell. The output of the LSTM is finally the element-wise product of the output gate with the tanh non-linearity of the cell state.

The Bidirectional Long Short-Term Memory (BLSTM) (Thireou & Reczko, 2007) is a further variant on the LSTM which processes the input data in both directions (forward and backward), combining the information from both the past and future inputs. The BLSTM consits of two LSTM

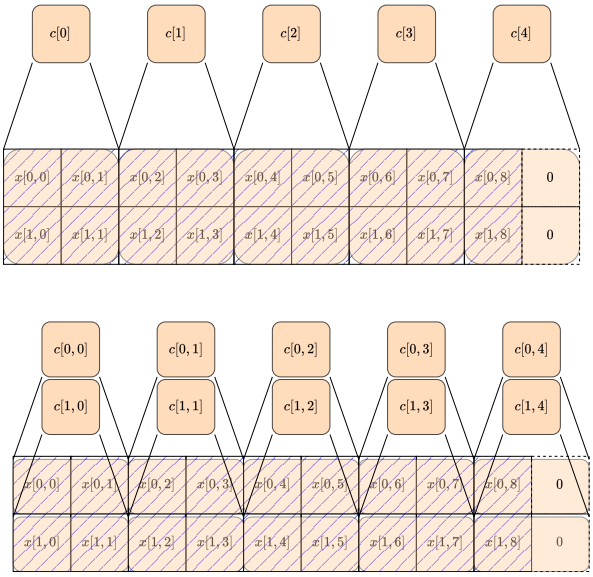

Figure 2.14: *A 2D CNN filter of size* 2×2 *and stride* $(2, 2)$ *with right side padding of* $(0, 1)$ *and 2D CNN filter of size* 2 *and stride* 1 *with right side padding of* $(0, 1)$.

units, one of which processes the input forward and the other backwards. The outputs of these are then concatenated to form the output of the BLSTM. The core utility of RNNs in SE is their ability to capture temporal relationships in sequential input data.

2.4.1.3 Attention and Transformer

The *attention* mechanism was originally developed for RNN-based language modelling tasks (Bahdanau et al., 2015) which required the computation of a *context* representation of the input data. This context vector is computed with an attention mechanism such that it can apply more weight to the most relevant features for the current RNN time step.

The Transformer (Vaswani et al., 2017) structure is a DNN which implements a mechanism of *scaled dot product attention*. At it's core, this involves the dot product of a the query matrix $\mathbf{Q}$ of shape $L_q \times D_q$ and a key matrix $\mathbf{K}$ of shape $L_k \times D_k$, such that $D_k = D_q$. The dot product is then simply

$$\mathbf{Q}\mathbf{K}^T. \tag{2.16}$$

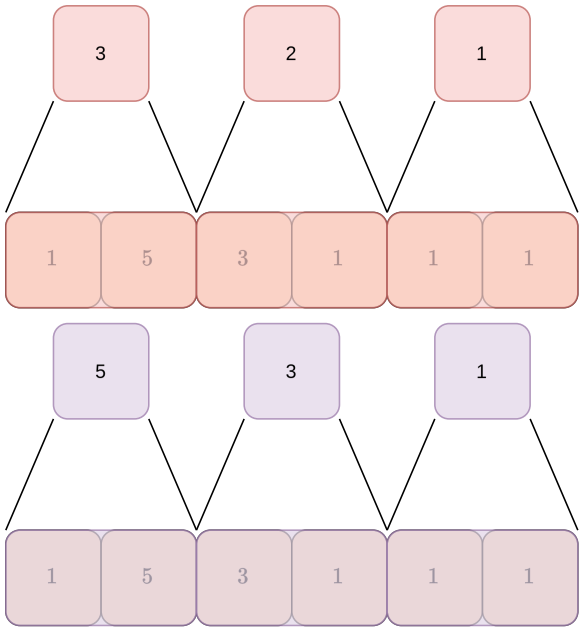

Figure 2.15: *Average (top) and Max (bottom) pooling on the output of a 1D CNN filter*

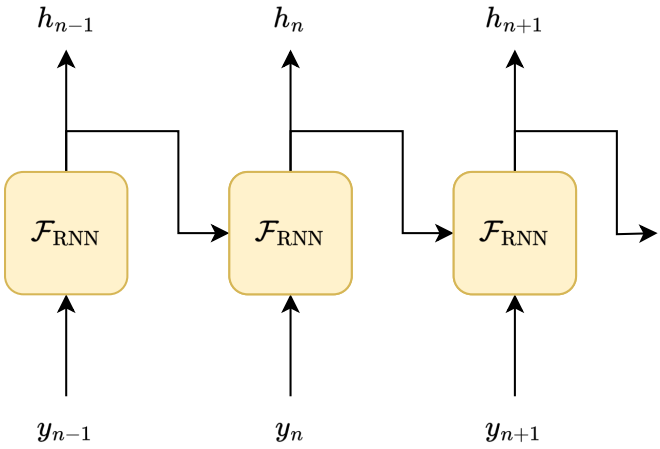

Figure 2.16: *Forward pass of an RNN layer over three time steps*

The scaled attention over some value matrix $\mathbf{V}$ is then defined as

$$\mathcal{A}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_q}}\right)\mathbf{V}. \tag{2.17}$$

In *self-attention*, the query, key and value matrices are created from the same input representation $\mathbf{X}$:

$$\begin{aligned} \mathbf{Q} &= \mathbf{W}_q\mathbf{X} \\ \mathbf{K} &= \mathbf{W}_k\mathbf{X} \\ \mathbf{V} &= \mathbf{W}_v\mathbf{X} \end{aligned} \tag{2.18}$$

where $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v$ are matrices of learnable DNN parameters. By comparing the value at a given

position in the input data sequence (the query) with each other position (the keys) in the data, an expression of the relationship between each value and every other value in the data can be learnt. In *cross-attention*, the query and keys are from a different sequence to the values, enabling an expression of a relationship between the representations $\mathbf{X}$ and $\mathbf{Y}$:

$$\begin{aligned} \mathbf{Q} &= \mathbf{W}_q \mathbf{Y} \\ \mathbf{K} &= \mathbf{W}_k \mathbf{Y} \\ \mathbf{V} &= \mathbf{W}_v \mathbf{X} \end{aligned} \tag{2.19}$$

Each Transformer block contains multiple attention layers (or *heads*) in parallel, (similar to how each CNN layer contains multiple filters). The output of the Multi-Head Attention (MHA) mechanism is concatenated and normalised before being input to a series of linear layers to give the output of the Transformer block. *Skip connection* summations are typically placed after the MHA and linear layers, adding the input representation of the layer to the output representation. The input to a Transformer has a *positional encoding* applied to it in order to provide positional (in the case of audio representations, temporal) information to the MHA, which otherwise has no way of receiving that information.

A Transformer based architecture might include both self-attention and cross-attention mechanisms; for an example of such an architecture see Section 2.5.3 introducing the Whisper ASR model. For an example architecture utilising only self-attention Transformer blocks see Section 2.5.2. The Transformer is particularly good at learning the long term dependencies within the input data. Unlike RNN- based approaches, the entire input sequence can be processed at once, in parallel in this manner. However, computing the scaled dot-product and softmax can be expensive computationally and memory wise, especially for longer sequences. Typically, the input and output dimensions of a Transformer block are equivalent, allowing for the chaining together sequentially of several identical Transformer blocks.

2.4.1.4 Conformer

The Conformer (Gulati et al., 2020) is a variant of the Transformer which introduces a CNN component following the MHA stage. Figure 2.17 shows the overall structure of a Conformer; it consists of a self-MHA module followed by a Convolutional module between two identical feed-forward Linear modules. The model utilises several *dropout* (Srivastava et al., 2014) layers throughout; dropout is a widely used technique in DNN which helps to prevent over-fitting during model training. Summation skip connections sum the input and output of each component block; in the case of the Linear module, these are so called 'half-step'(Y. Lu et al., 2019) connections such that the values within the output representation are halved before summation with the input. Each Linear Module consists of two Linear layers, the first of which projects to 4 times the size of the input and has an Swish (Ramachandran et al., 2017) activation, while the second projects back down to the size of the input, such that is can be summed with the input to the block. The self-MHA Module follows a standard configuration described previously, with a relative positional embedding. The Convolutional Module contains two distinct forms of CNN layer. The first is the pointwise convolution (Hua et al., 2018) which uses filters of size $(1, 1)$ such that each filter is applied to each point of the input tensor, with no shifting across the input representation. The other kind of CNN used in this block is a single depth-wise (across the filters) 1D layer. The first pointwise convolution projects to twice the size of the input and has a Gated Linear Unit (GLU) activation which encourages the second half of the output representation to act as a 'gate' over the first, halving the dimentionality back to that of the input.

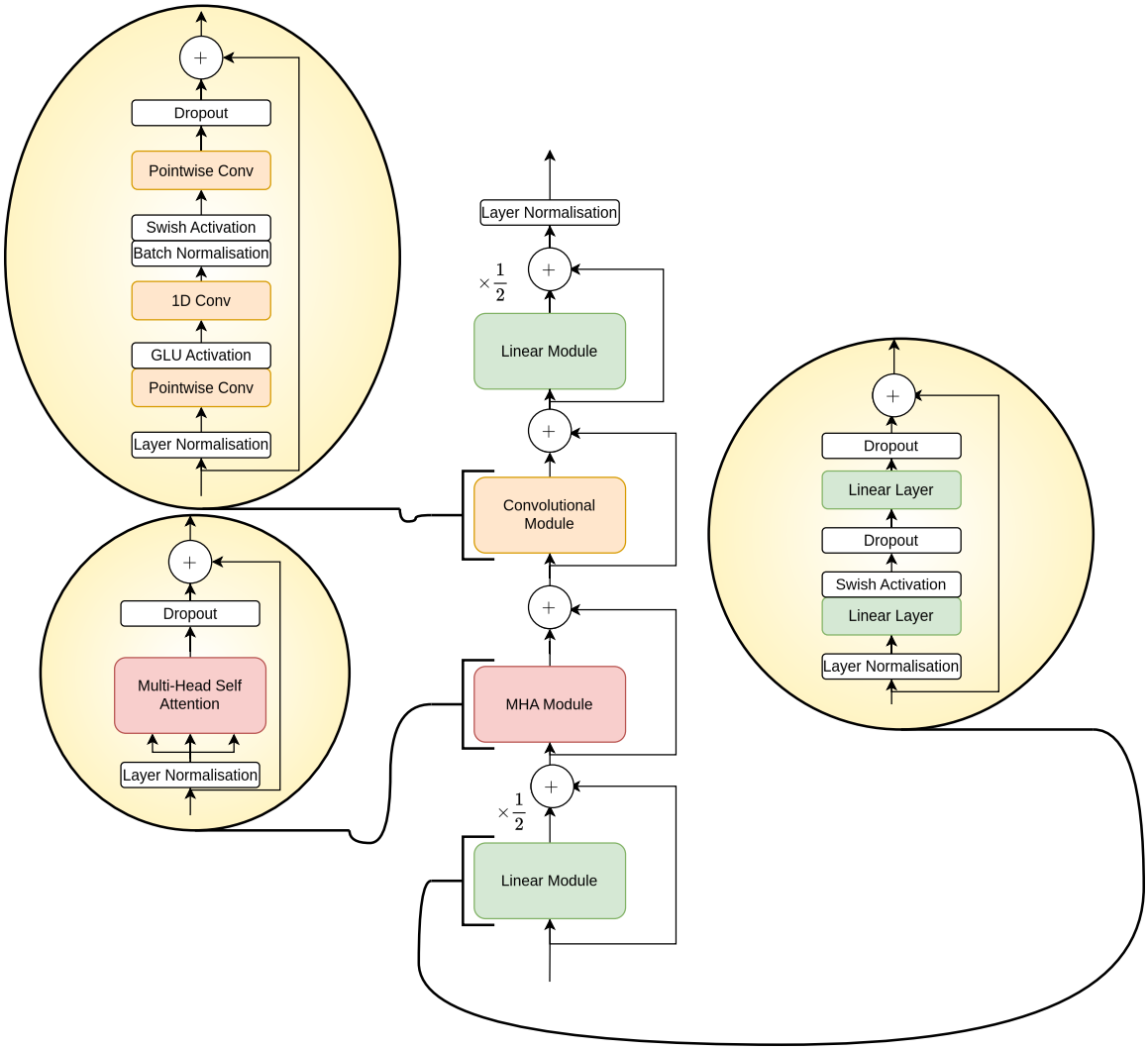

Figure 2.17: *Overview of a Conformer DNN block.*

2.4.1.5 Structured State Space models and MAMBA

Stuctured State Models (SSM) models (Gu et al., 2022) are an emerging area of interest in DNN design. A state space model can be defined as follows. A continuous 1D input signal $p(t)$ is mapped to an N dimensional continuous hidden latent state $h(t)$ before projection to a continuous 1D output signal $o(t)$.

$$\begin{aligned} h'(t) &= \mathbf{A}h(t) + \mathbf{B}p(t) \\ o(t) &= \mathbf{C}h(t) + \mathbf{D}p(t) \end{aligned} \tag{2.20}$$

$h'(t)$ here defines the change in $h(t)$ over time. In the context of DNNs, the matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ consist of learnable parameters. Computation of $\mathbf{D}$ is trivial as it resolves to a weighted residual connection with the input $p(t)$; in the following (and much of the literature) it is assumed that $\mathbf{D} = 0$. To transform the system from continuous to discrete such that an SSM can be formulated as a DNN requires p_t and computation of the value of the hidden state h_t at a given time step t, which

can then be used to get the output value o_t:

$$\begin{aligned} h_t &= \bar{\mathbf{A}} h_{t-1} + \bar{\mathbf{B}} x_t \\ o_t &= \bar{\mathbf{C}} h_t \end{aligned} \tag{2.21}$$

One method to discretize $\mathbf{A}$ and $\mathbf{B}$ to obtain $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ is the *bilinear* method (Tustin, 1947):

$$\begin{aligned} \bar{\mathbf{A}} &= (\mathbf{I} - \Delta_{\text{ssm}}/2 \cdot \mathbf{A})^{-1}(\mathbf{I} + \Delta_{\text{ssm}}/2 \cdot \mathbf{A}) \\ \bar{\mathbf{B}} &= (\mathbf{I} - \Delta_{\text{ssm}}/2 \cdot \mathbf{A})^{-1}\Delta_{\text{ssm}}\mathbf{B} \\ \bar{\mathbf{C}} &= \mathbf{C} \end{aligned} \tag{2.22}$$

where $\mathbf{I}$ is the identity matrix and Δ_{ssm} is a 'step size' representing the resolution of the input. Given that the value of h_t is dependent on h_{t-1} this discrete form of the SSM can be modelled as as a form of RNN. In order for the model to be trained efficiently, this RNN like formulation can be converted into a CNN like one. Starting at time step $t = 0$, given that $h_{-1} = 0$, the hidden states for $t = 0, 1, 2$ are:

$$\begin{aligned} h_0 &= \bar{\mathbf{B}} p_0 \\ h_1 &= \bar{\mathbf{A}} h_0 + \bar{\mathbf{B}} p_1 = \bar{\mathbf{A}}\bar{\mathbf{B}} p_0 + \bar{\mathbf{B}} p_1 \\ h_2 &= \bar{\mathbf{A}} h_1 + \bar{\mathbf{B}} p_2 = \bar{\mathbf{A}}^2\bar{\mathbf{B}} p_0 + \bar{\mathbf{A}}\bar{\mathbf{B}} p_1 + \bar{\mathbf{B}} p_2 \end{aligned} \tag{2.23}$$

and so on.The discrete output $o[t]$ can be expressed similarly:

$$\begin{aligned} o_0 &= \bar{\mathbf{C}}\bar{\mathbf{B}} p_0 \\ o_1 &= \bar{\mathbf{C}}\bar{\mathbf{A}} h_0 + \bar{\mathbf{B}} p_1 = \bar{\mathbf{C}}\bar{\mathbf{A}}\bar{\mathbf{B}} p_0 + \bar{\mathbf{C}}\bar{\mathbf{B}} p_1 \\ o_2 &= \bar{\mathbf{C}}\bar{\mathbf{A}} h_1 + \bar{\mathbf{C}}\bar{\mathbf{B}} p_2 = \bar{\mathbf{C}}\bar{\mathbf{A}}^2\bar{\mathbf{B}} p_0 + \bar{\mathbf{C}}\bar{\mathbf{A}}\bar{\mathbf{B}} p_1 + \bar{\mathbf{C}}\bar{\mathbf{B}} p_2 \end{aligned} \tag{2.24}$$

and so on. This can be expressed as a summation of T terms where T is the length of the discrete input sequence $p[t]$:

$$o_T = \bar{\mathbf{C}}\bar{\mathbf{A}}^T\bar{\mathbf{B}} p_0 + \bar{\mathbf{C}}\bar{\mathbf{A}}^{T-1}\bar{\mathbf{B}} p_1 + ... + \bar{\mathbf{C}}\bar{\mathbf{A}}\bar{\mathbf{B}} p_{T-1} + \bar{\mathbf{C}}\bar{\mathbf{B}} p_T \tag{2.25}$$

such that a convolutional kernel $\bar{\mathbf{K}}$ can be defined as

$$\bar{\mathbf{K}} = (\bar{\mathbf{C}}\bar{\mathbf{B}}, \bar{\mathbf{C}}\bar{\mathbf{A}}\bar{\mathbf{B}}, ..., \bar{\mathbf{C}}\bar{\mathbf{A}}^T\bar{\mathbf{B}}) \tag{2.26}$$

such that the entire SSM mapping from discrete input $u[t]$ to discrete output $o[t]$ can be computed by

$$o = \bar{\mathbf{K}} * u \tag{2.27}$$

MAMBA (Gu & Dao, 2023) is a SSM model which differs from the standard design in two major ways. Firstly it implements a selection mechanism which is dependent on the input sequence, allowing for the effective filtering of the information encoded in that input. In practise, this means that the computation of matrices $\mathbf{B}$ and $\mathbf{C}$ and the value Δ_{ssm} are *selective* and dependent on the content of the input $u[t]$. This is implemented by the learning of linear projections for each component over the input sequence, similar to the computation of the key,query pairs in attention. Further, $\mathbf{A}$ is fixed and structured as a high-order polynomial projection operators(HiPPO) (Gu et al., 2020) matrix:

$$\mathbf{A}_{nk} = \begin{cases} (2n+1)^{1/2}(2k+1)^{1/2} & \text{if } n > k, \\ n+1 & \text{if } n = k, \\ 0 & \text{if } n < k, \end{cases} \tag{2.28}$$

By learning a value of Δ_{ssm}, the MAMBA model is able to control how much influence the current input p_t has over the hidden state h_t; a large Δ_{ssm} means that the model is focusing on that input for 'longer', while a small Δ_{ssm} value means that that input is ignored.By making $\mathbf{B}$ and $\mathbf{C}$ selective allows for the control of information into the hidden state and into the output state respectively. The second difference in MAMBA is that it introduces a hardware efficient 'scan' (rather than a convolution) algorithm which scales in complexity linearly with the length of the input sequence. Compared to the now ubiquitous Transformer structure, MAMBA has shown equivalent performance in a number of tasks, and is significantly more efficient computationally both during training and inference. It has been applied to speech audio in the speech enhancement task (Chao et al., 2024) and ASR (X. Zhang et al., 2024) where it demonstrated state-of-the-art performance.

2.4.2 Neural Network Activation Functions

Typically the output of each layer of DNN is processed by an non-linear activation function, which allows it to learn more intricate patterns. Without these, DNNs are only useful for mapping linear relationships between data (Haykin, 2009). Activation functions are used both within a DNN between hidden layers as well as at the final output layer to enforce some target (i.e a Sigmoid activation function on the example binary classification model for rainfall introduced in Section 2.4.1). Figure 2.18 visualises the characteristic of some of the activation functions used in this work and described here.

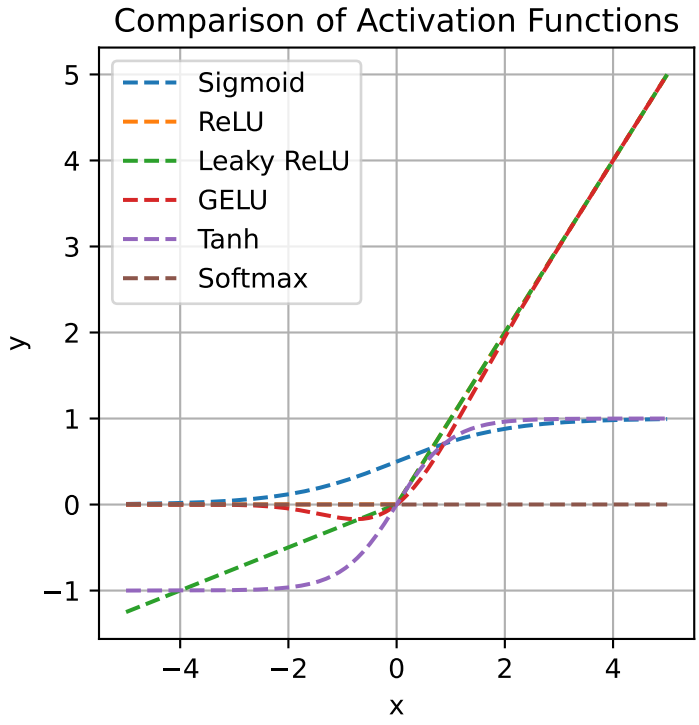

Figure 2.18: *Plots of various non-linearities used as DNN layer activation functions.*

2.4.2.1 Sigmoid

The Sigmoid function maps the input to a value in the range between 0 and 1:

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \tag{2.29}$$

where e is the mathematical constant Euler's number (Euler, 1770).

2.4.2.2 Swish

The Swish activation (Ramachandran et al., 2017) is a variant of sigmoid which multiplies the input by the output of the sigmoid function:

$$\text{Swish}(x) = x * \frac{1}{1 + e^{-x}} \tag{2.30}$$

2.4.2.3 Tanh

The Tanh function applies the hyperbolic tangent to the input:

$$\text{Tanh}(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.31}$$

2.4.2.4 ReLU

The Rectified Linear Unit (ReLU) (Fukushima, 1969) function maps negative values in the input to 0:

$$\text{ReLU}(x) = \max(0, x) \tag{2.32}$$

2.4.2.5 Leaky ReLU

Leaky ReLU (Maas et al., 2013) is a variant of ReLU which allows some small hyperparameter α positive gradient scale for negative inputs:

$$\text{LeakyReLU}(x) = max(0, x) + \alpha * min(0, x) \tag{2.33}$$

2.4.2.6 Parametric ReLU

The Parametric ReLU (He et al., 2015) is a variant on Leaky ReLU where the scale of the positive gradient is learnt along with the rest of the network, rather than being a hyperparameter.

$$\text{PReLU}(x) = max(0, x) + a_{\text{PreLU}} * min(0, x) \tag{2.34}$$

where a_{PreLU} is a learnable DNN parameter.

2.4.2.7 GELU

The Gaussian Error Linear Unit (GELU) (Hendrycks & Gimpel, 2016) activation function multiplies the input by the Gaussian Cumulative Distribution Function, which is typically approximated:

$$\text{GELU}(x) = 0.5 * x * (1 + \tanh(\sqrt{2/\pi} * (x + 0.044715 * x^3))) \tag{2.35}$$

2.4.2.8 Softmax

The Softmax function rescales the input such that each element lies between 0 and 1 with the sum of the output being 1. For each element x_i in the input:

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{2.36}$$

2.4.2.9 GLU

The Gated Linear Unit (GLU) (Dauphin et al., 2017) function splits the input matrix in halves a and b and element-wise multiplies the first half by the sigmoid of the second:

$$\text{GLU}(x_a, x_b) = x_a \otimes \frac{1}{1 + e^{-x_b}} \tag{2.37}$$

where $\otimes$ is an element wise multiplication.

2.4.3 Neural Network Loss Functions

A key aspect of the creation of supervised neural noise reduction networks is the loss term or objective function. Broadly speaking, this is a function which returns a value that during training the network attempts to minimise, and describes the difference between the model's prediction and the label. At each training step (over a 'batch' of inputs) this value is computed and used to update the parameters of the network using a *back-propagation* algorithm.

Typically in NNSE training, the loss function is some distance computed between a representation of the clean label ('reference') audio and the enhanced audio output by the model. The simplest of these is the Mean Absolute Error (MAE) (l1) loss between the time domain reference audio $s[n]$ and the enhanced audio $\hat{s}[n]$:

$$\mathcal{L}_{\text{Time}} = |\sum_{n}^{N} \frac{1}{N}(s[n] - \hat{s}[n])| \tag{2.38}$$

Another commonly used loss function is the MSE between magnitude Short Time Fourier Transform (STFT) representations:

$$\mathcal{L}_{\text{Spec}}(\mathbf{S}_{\text{Mag}}, \hat{\mathbf{S}}_{\text{Mag}}) = \frac{1}{T \cdot F_{\text{Hz}}} \sum_{t}^{T} \sum_{f_{\text{Hz}}}^{F_{\text{Hz}}} (\mathbf{S}_{\text{Mag}}[t, f_{\text{Hz}}] - \hat{\mathbf{S}}_{\text{Mag}}[t, f_{\text{Hz}}])^2 \tag{2.39}$$

These losses have been found to introduce artefacts in the enhanced speech and show a low correlation with measures of human perception (Bagchi et al., 2018; Chai et al., 2018; Goetze et al., 2014). Thus, models trained solely using a clean speech distance loss function may introduce unwanted artefacts.

2.4.4 Neural Network Training

In supervised DNN training, the data and label pairs are typically split intro three partitions, training, validation and testing sets or *splits*. The model is trained using the training set; one full iteration through the training set is referred to as an *epoch*. At the end of each epoch (or otherwise at some interval during the training), the performance of the model is evaluated (without updating it's parameters) on the validation set. After training is complete, the model is then evaluated over the testset. The design of the data partition is of critical importance. The validation split must be different enough from the training data that a degradation in validation performance across epochs can be used to reveal overfitting (learning the training data too well). Similarly, the training split must also be distinct from the training data in order to assess the generalisation of the model to unseen data, which is crucial for real-world uses. In datasets for the NNSE task, the testset audio data contains distortion types and speakers which are not present in the training set. It is possible to control if certain layers of parameters are updated during training; those which are set to not be updated are called *frozen* parameters.

As an example, let's return to the DNN depicted in Figure 2.10 tasked with predicting the chance of rainfall on a given day. The training data for this task would consist of relevant input features (rainfall on the days prior, cloud cover, temperature, date etc.) paired with a binary label of if it did or did not rain under the conditions described in those features in the past. From this, a simple binary classification model can be trained. In the design of the training/validation/testing split for this task, it might be prudent to test the model on data from times of the year which it did not observe during training i.e. to ensure that a model trained on data from winter can generalise to predicting rainfall in the summer.

The training of DNN systems is affected by a number of *hyperparameter* values. These typically include the learning rate which controls the influence of the loss value in the back-propagation stage, and the *batch* size which is the number of inputs processed by the model in one update step. Other hyperparameter values are used depending on the specific setup, for example feature transformation and loss function; the parameters of an STFT transformation or the weighting between two loss terms. More generally, a hyperparameter can refer to any value which is set by the designer of the model and for which a value is not learnt during training.

2.4.4.1 Normalisation Layers

There are several common techniques for reducing training time and improving model generalisation. Two commonly used techniques are *batch normalisation* (Ziaee & Çano, 2023) and *layer normalisation* (Ba et al., 2016). In batch normalisation, each element in the batch is normalised by the mean μ_{batch} and variance σ_{batch} of the entire batch:

$$\text{BatchNorm}(x) = \gamma \left(\frac{x - \mu_{\text{batch}}}{\sqrt{\sigma_{\text{batch}} + \epsilon}} \right) + \beta \tag{2.40}$$

where γ and β are learnable parameters and ϵ is some very small value to prevent potential division by 0. Layer normalisation is similar, but applies normalisation relative to the mean μ_{unit} and variance σ_{unit} of the values in the hidden unit representation:

$$\text{LayerNorm}(x) = \gamma \left(\frac{x - \mu_{\text{unit}}}{\sqrt{\sigma_{\text{unit}} + \epsilon}} \right) + \beta \tag{2.41}$$

2.5 Pretrained Foundational Speech Models

2.5.1 Pre-training and Fine-tuning

A common strategy for DNN training is that of *pre-training* followed by *fine-tuning*. In pre-training, the model is typically trained towards a unsupervised general objective using a large training set. Then in fine-tuning, the model is trained on some smaller dataset with supervised objective towards some specific task. In this work, a number of pre-trained models are used as feature extractors in both the NNSE task and speech metric/MOS prediction, and are presented in the next section.

2.5.2 Self Supervised Speech Representation (SSSR)

Self Supervised Speech Representation (SSSR) models are DNN models of speech which are trained in a self supervised way using large corpora of speech data (Baevski et al., 2020). This is typically done by 'masking' a portion of the input and then tasking the model with recreating the masked portion, in a manner similar to an auto-encoder network. At inference time, the network layers responsible for the recreation step are removed and the model instead returns a deep 'context' representation of the input time domain audio. At this point, additional task-specific layers can be appended to the network, with the self supervised representation model either being fine-tuned or frozen as the task specific layers are trained. Generally speaking, SSSRs can be said to first *perceive* the input audio in a feature encoder step, and then *predict* the context of the content of the audio in the deeper layers.

SSSR models output a *context* representation of the input speech audio waveform. Structurally, they consist of two main stages. The first, denoted by the operator $\mathcal{G}_{\mathrm{FE}}$ in the following with subscript $_{\mathrm{FE}}$ standing for *feature encoder*, is built from a number of 1D convolutional layers which convert the input time-domain speech signal $s[n]$ to a two-dimensional feature representation:

$$\mathbf{S}_{\mathrm{FE}} = \mathcal{G}_{\mathrm{FE}}(s[n]), \tag{2.42}$$

with a feature dimension F (typically of size 512) and a time dimension T, i.e. the number of frames, which is dependent on the length of the input audio signal. The strides and kernel widths of the 1D Convolutional layers result in a output frequency of 49Hz i.e 1 second of audio at 16000Hz sample rate is represented by 49 time dimension T indexes in $\mathbf{S}_{\mathrm{FE}}$.

The second stage, denoted by $\mathcal{G}_{\mathrm{OL}}$, consists of a number of self-attention Transformer (cf. Section 2.4.1.3) layers and operates over a linear projection of the feature encoder output

$$\mathbf{S}_{\mathrm{OL}} = \mathcal{G}_{\mathrm{OL}}(\mathcal{G}_{\mathrm{FE}}(\mathbf{s}[n])). \tag{2.43}$$

The output representation $\mathbf{S}_{\mathrm{OL}}$ shares the time dimension T with $\mathbf{S}_{\mathrm{FE}}$ but has a different, usually larger feature dimension F. Subscript $_{\mathrm{OL}}$ stands for *output layer* of the SSSR. Structurally, the networks consist of two distinct stages as shown in Figure 2.19.

2.5.2.1 Pre-training objectives for SSSRs

There exist several schemes for the pre-training of SSSRs.
The **wav2vec2.0** (Baevski et al., 2020) self-supervised pre-training objective is as follows. First,

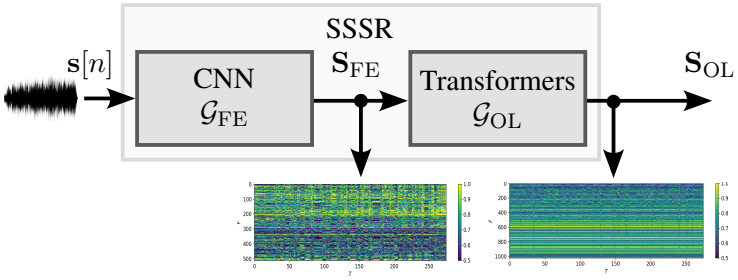

Figure 2.19: *Representations extracted from SSSR model with time-domain input signal* $\mathbf{s}[n]$. *Feature channels are sorted (Ravenscroft et al., 2022) and values normalised for clarity.*

the output of the CNN feature encoder $\mathbf{X}_{\text{FE}}$ is multiplied by a learnt projection to project each 512 length vector to length 640 *logits*. Each of these logits are divided into 2 groups G of vectors, representing *codebooks* of 320 discrete vectors. A Grumbel-Softmax (Gumbel, 1954) is used to sample a one-hot vectors for each group, giving 2 one hot vectors which are concatenated. From this, a quantized vector j_t for each $t \in T$ in $\mathbf{X}_{\text{FE}}$ is obtained, which can be projected by another linear projection to the size of the feature dimension of $\mathcal{G}_{\text{OL}}$. The idea behind this is to build these codebooks which are able to encode general common speech features across the inputs during training. Then, 50% of the frames are *masked*, before being input to the Transformer stage $\mathcal{G}_{\text{OL}}$. The output of $\mathcal{G}_{\text{OL}}$ can be expressed as T vectors r_t of length 1024. The task of the model is to solve a *contrastive* identification problem by identifying the quantized vector j_t which corresponds to the each masked $\mathbf{X}_{\text{FE}}$ frame

$$\mathcal{L}_{\text{wav2vec}} = -\log \frac{\exp(\text{sim}(\text{r}_\text{t}, \text{j}_\text{t})/\tau_{\text{wav2vec}})}{\sum_{\bar{j}\sim J_t} \exp(\text{sim}(\text{c}_\text{t}, \bar{\text{j}}\tau_{\text{wav2vec}}))} \tag{2.44}$$

where τ_{wav2vec} is a scaling constant hyperparameter and $\text{sim}(a, b)$ is the cosine similarity

$$\text{sim}(a, b) = a^T b/||a||||b||. \tag{2.45}$$

Overall this loss function maximises the similarity between the r_t and the quantized vector j_t of the masked frame while minimising the similarity between r_t and all the other masked frames. This loss function is supplemented with a diversity loss (Dieleman et al., 2018) which encourages the full scope of the codebook to be used.

The Hidden Unit BERT (HuBERT) (Hsu et al., 2021) pre-training objective differs significantly from that of wav2vec2.0. Instead of a contrastive objective, an approach inspired by masked language modeling (Devlin et al., 2019) is used. First, k-means clustering (Lloyd, 1982) is performed over Mel-Frequency Cepstral Coefficient (MFCC) (Davis & Mermelstein, 1980) features of the training dataset. Each MFCC feature vector of length 39 is assigned to one of 100 clusters. From this, a *hidden unit embedding* vector of length T can be built, encoding to which cluster each MFCC frame was assigned. Then, as in wav2vec2.0, 50% of the frames in $\mathbf{X}_{\text{FE}}$ are masked and input to $\mathcal{G}_{\text{OL}}$. Each frame in the output of $\mathcal{G}_{\text{OL}}$ is projected to the same feature dimension as e (100 during the initial iteration) and a cross-entropy loss is is used to compute the similarity between the masked frames and the hidden unit embedding. Following the initial iteration, in subsequent iterations the k-means clustering is instead computed with 500 clusters over the output of one of the intermediate transformer layers, clustering with 768 rather than 39 features.

2.5.2.2 SSSR Representations

The **Cross-Lingual Speech Representation (XLSR)** (Babu et al., 2022) is an SSSR with the main distinguishing feature being that it is trained using audio containing a large number of languages. It is intended to act as a 'universal' model of speech, encoding latent speech representations which are shared across languages. It is trained on 436k hours of speech from 128 different languages from datasets including VoxPopuli, CommonVoice and BABEL[1], with the Wav2Vec2 (Baevski et al., 2020) contrastive masking objective. Note that unlike the other two SSSRs used in this work, it is trained on potentially noisy data, notably CommonVoice (described below in Section 2.9.4) and BABEL which contains conversational telephone recordings. Its $\mathcal{G}_{\mathrm{FE}}$ representations have a feature dimension F of 512 while its $\mathcal{G}_{\mathrm{OL}}$ representations have an F of 1024.

Hidden Unit BERT (HuBERT) (Hsu et al., 2021) is an SSSR model; during training it makes use of a BERT (Devlin et al., 2019) inspired loss function. The output of its feature encoder $\mathcal{G}_{\mathrm{FE}}$ has a dimension of $F = 512$, while its final layer output after $\mathcal{G}_{\mathrm{OL}}$ has a feature dimension of 758. In this work, the HuBERT model used is trained on the 960 hour Librispeech (Panayotov et al., 2015) training set and is sourced from the fairseq GitHub repository[2]. It is important to note that this dataset consists of English read speech only, so the model has only ever been exposed to English speech.

Multilingual HuBERT (mHuBERT) (Lee et al., 2022) is a variation on HuBERT which has been trained on multilingual speech data, specifically the English, French and Spanish language parts of the VoxPopuli (C. Wang et al., 2021) dataset, each containing 4.5k hours totalling 13.5k hours of speech. It has the same feature dimensions as HuBERT. It can be considered as a *middle point* between the monolingual HuBERT and the massively multilingual XLSR.

WavLM (Chen et al., 2022) is a varient on HuBERT, which introduces a secondary SE task, wherein k-mean clusters of clean audio are predicted from potentially noisy inputs. In this work, the WavLM Base[3] model trained on the Librispeech 960 hour (Panayotov et al., 2015) dataset is used which has $L = 12$ Transformer layers. It has the same feature dimensions as HuBERT. Figure 2.20 shows an overview of the WavLM architecture. The numbers in brackets of the 1D Convolutional layers denote the stride and kernel width of that layer.

In (Hsieh et al., 2020) the relationship between the SSSR distances and perceptual measures is noted. In (Tal et al., 2022), a number of techniques to incorporate SSSRs (namely HuBERT) into a single channel speech enhancement system are proposed. One of these techniques, called 'supervision' in (Tal et al., 2022) involves the use of the distance between the SSSR output representations of clean reference speech and the enhanced noisy speech output by the enhancement model as an additional loss term to train the model. This is in turn inspired by a prior work (Hsieh et al., 2020) in which the Wasserstein distance between clean and enhanced SSSR representations of the audio is used as a loss term.

2.5.3 Whisper

Whisper is a 'weakly supervised' encoder-decoder Transformer-based ASR system. It has shown state-of-the-art performance on a number of monolingual ASR benchmark datasets, as well as multilingual transcription and translation tasks (Radford et al., 2022). In this work, it is used as a foundational model feature extraction system. Figure 2.21 shows an overview of the Whisper model

[1] https://catalog.ldc.upenn.edu/byyear
[2] https://github.com/facebookresearch/fairseq
[3] https://huggingface.co/microsoft/wavlm-base

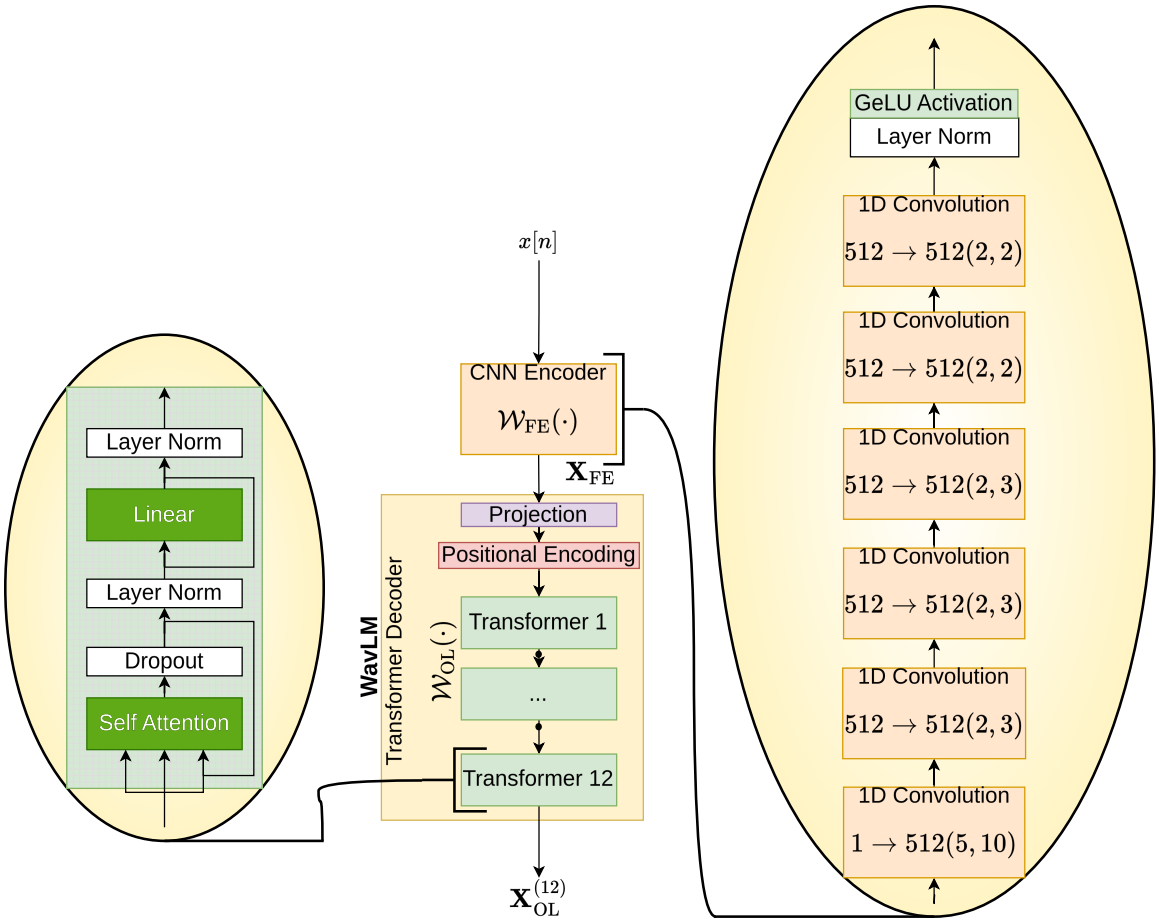

Figure 2.20: *An overview of the WavLM architecture.*

architecture, specifically, that of the `whisper-small` variant. It consists of several sequential Transformer-based self-attention encoder blocks $\mathcal{A}_{\text{E}}(\cdot)$ followed by the same number of sequential Transformer-based self-attention and cross attention decoder blocks $\mathcal{A}_{\text{D}}(\cdot)$. The input to the encoder $\mathcal{A}_{\text{E}}(\cdot)$ is a log-Mel spectrogram matrix representation of the input audio $x[n]$ (padded to 30 seconds in length) denoted as $\mathbf{X}_{\text{MEL}}$, which is windowed by a 1 dimensional CNN layer with a GELU activation function, followed by a sinusoidal positional encoding. Each Transformer block in $\mathcal{A}_{\text{E}}(\cdot)$ consists of a self-MHA layer, followed by a linear layer with summation residual connections. The output of the encoder is $\mathbf{X}_{\text{E}}^{(12)}$ a two-dimensional representation of dimension F of 768 by T of 1500 (Radford et al., 2022).

The Whisper decoder takes the form of a language model; the first Transformer block of the decoder takes as input a sequence of tokens which encode the language, task, timestamp in seconds and the previously transcribed words of the utterance. Each Transformer block in the decoder has access to the output of the encoder via the cross-attention mechanism such that $\mathbf{X}_{\text{E}}^{(12)}$ is used as to compute the key and query matrices. The final output of the decoder (not used in this work) is a prediction of the next token (i.e. the next word) in the input sequence. The T dimension of the output of each Whisper decoder layer is significantly smaller than any other feature used in this work.

Whisper is trained using a Connectionist Temporal Classification (CTC) like loss function which allows for an alignment-free training over audio-transcript pairs. In this work the

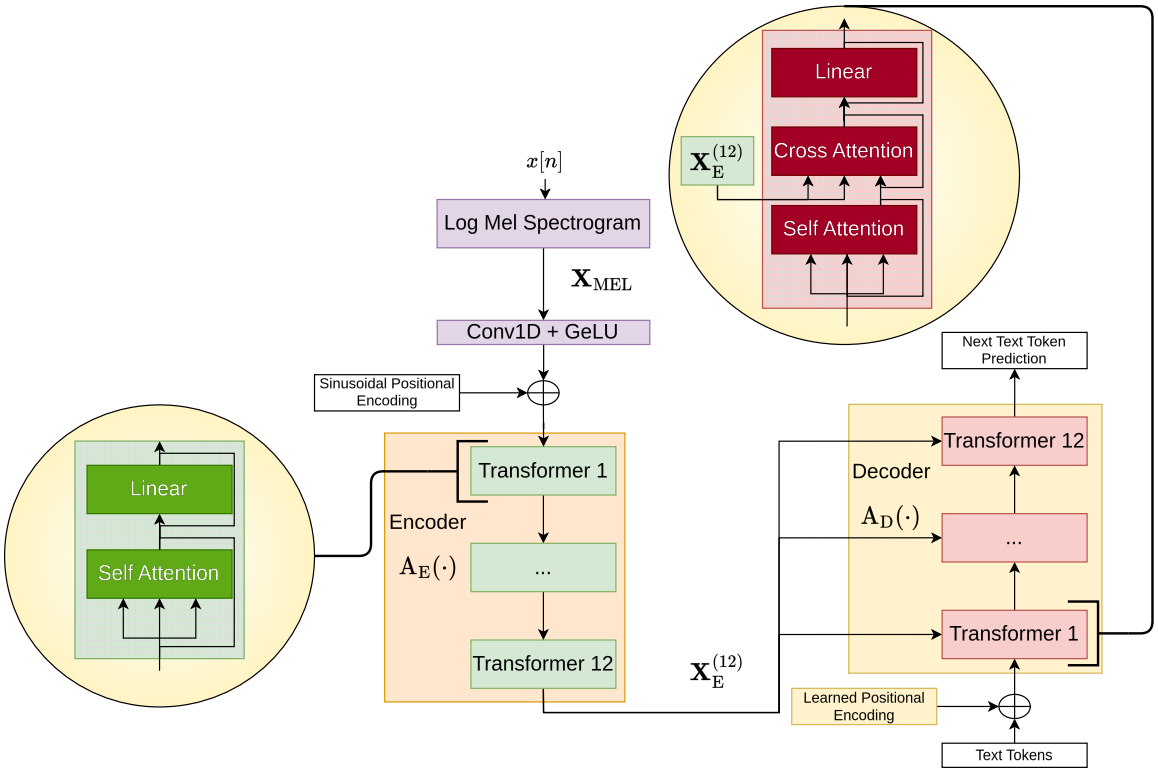

Figure 2.21: *An overview of the Whisper DNN model architecture.*

`whisper-small`[4] model, trained on 680k hours of weakly-labelled speech data is used. Recent work has found that features extracted from both the encoder (Santiago Cuervo, Ricard Marxer, 2024) and decoder (Mogridge et al., 2024) layers of Whisper are useful for capturing intelligibly-related information.

The encoder $\mathcal{A}_{\mathrm{E}}(\cdot)$ and decoder $\mathcal{A}_{\mathrm{D}}(\cdot)$ of this model each have 12 transformer blocks; the set of outputs of each of the constituent transformer blocks are denoted as $\{\mathbf{X}_{\mathrm{E}}^{0}..\mathbf{X}_{\mathrm{E}}^{(12)}\}$ and $\{\mathbf{X}_{\mathrm{D}}^{0}..\mathbf{X}_{\mathrm{D}}^{(11)}\}$ respectively.

2.6 Generative Adversarial Networks (GANs)

A key concept which will be used throughout this work is that of the Generative Adversarial Network (GAN) framework (Goodfellow et al., 2014). This structure involves the training of two district DNNs, a Generator network and a Discriminator network. The Generator is the network trained to produce some output specific to the target task, for example, production of an image given a text prompt as input. The task of the Discriminator is to distinguish between the outputs of the Generator and 'real' examples of samples in the target task domain. These two networks are trained in tandem, with inference of the Discriminator network used to form the loss function of the Generator.

A useful way to think about this structure is to think of the Generator as an art forger and the Discriminator as an art expert. The art forger produces paintings with the goal of 'fooling' the

[4] https://huggingface.co/openai/whisper-small

art expert into labelling their work as real, while the art expert attempts to correctly label the real paintings as real and the forger's forgeries as such. As the two networks are trained together, they learn *adversarially* to beat each other in this 'game'.

Figure 2.22 shows the general approach to GAN training. The 'Real/Fake Probability' output of

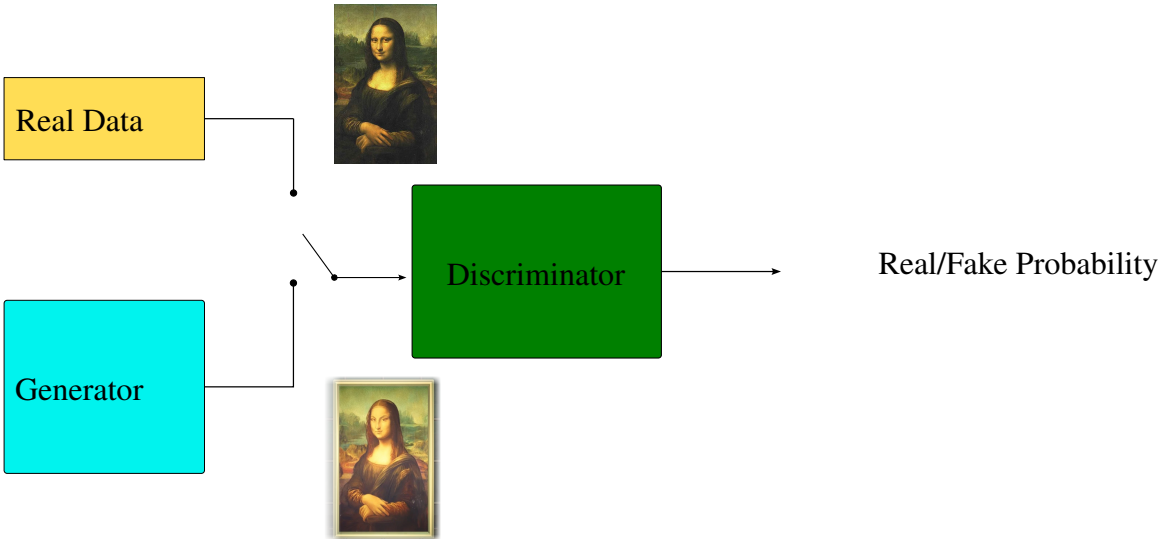

Figure 2.22: *Typical GAN structure overview*

the Discriminator is used to formulate loss functions for both networks. Note that in this example no specific input representation is given to the Generator, but this is not the case in the GAN applications studied in this work.

GANs have been found to be useful in a number of areas, particularly in the image processing domain for applications such as image generation (B. Zhang et al., 2022), style transfer (Azadi et al., 2018), and many others (Ramesh et al., 2021).

2.7 Assessing Performance of Speech Enhancement

The two quantities of assess the speech audio are the *quality* and *intelligibility*:

- Speech Quality (SQ) relates to the aspects of the speech signal which are independent of linguistic/semantic meaning of the speech. High quality speech audio dose not contain any audible non-speech environmental sound and the speech signal is un-distorted.
- Speech Intelligibility (SI) relates to the clarity of the semantic content of the speech such that the meaning of what is said is preserved.

These concepts are related, such that high quality audio is typically also highly intelligible. Both signal processing and DNN approaches to speech enhancement require means of assessing and comparing the performance of the proposed system. In SE, opinion scores regarding enhanced and noisy speech can be gathered through listening tests; however, this can be expensive and time consuming. To avoid this, computational estimators (metrics) of the quality (Rix et al., 2001) and

intelligibility (Taal et al., 2011) have been developed. Intelligibility of speech is usually defined as the number of speech units (words or phones) which can be correctly identified in the signal by human assessors. Intelligibility of speech degrades mainly at very low SNR values. Quality of speech is less strongly defined and is highly subjective; different human assessors can have wildly different criteria for what constitutes high or low quality, leading to a high variance in quality assessments. As such, human evaluations of quality are often averaged over the ratings for each signal. Some metrics also incorporate features other than time domain signals to their input, such as representations of a specific individual's hearing loss (Kates & Arehart, 2014). A number of metrics are psycho-acoustically motivated meaning that they attempt to incorporate a model of the physical and mental characteristics of human hearing. Metrics which are not explicitly related to human perception but which do correlate with them such as the SNR also exist, as well as those derived from weighted combinations of a number of component measures (Lin et al., 2019). Metrics can be either *intrusive* (cf. Figure 2.23), meaning that they require the existence of a 'clean' (free from noise) reference version of the signal under test for comparison or *non-intrusive* (cf. Figure 2.24), meaning that only the signal under test is required. Non-intrusive metrics often use an 'internal' reference to assess the input signal. An issue with intrusive measures is that they are less useful to assess the aspects of 'real' data for which no reference signal can be easily obtained. When

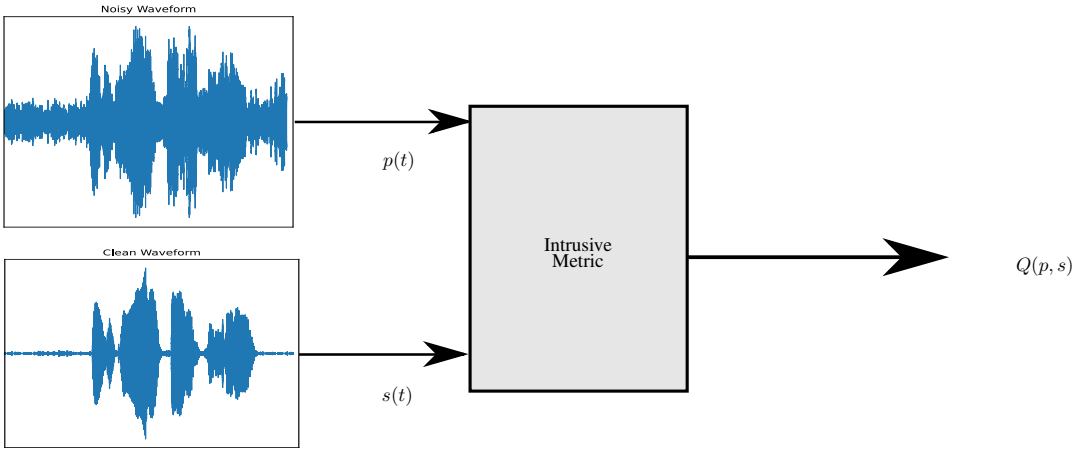

Figure 2.23: *An intrusive metric computed for the signal $p(t)$ on a time domain signal given the reference signal $s(t)$*

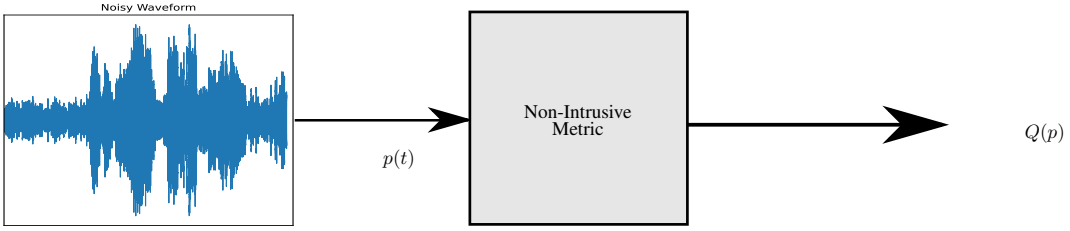

Figure 2.24: *A non-intrusive metric computed on a time domain signal $p(t)$.*

datasets for speech enhancement tasks are created, a common technique is to mix the clean speech signals with the noise at discrete SNR values. The general approach for the assessment of speech enhancement systems is to compare the mean value of a number of these metrics on a shared 'test set' of data.

This section will introduce some of the signal assessment metrics which will be used in the following chapters.

2.7.1 Mean Opinion Score

Mean Option Score (MOS) (*P.10 : Vocabulary for performance, quality of service and quality of experience*, 2017) is way of averaging human assessment of an audio signal. Typically, human evaluators are asked to listen to audio and assign a score between 1 and 5 (Bad, Poor, Fair, Good, Excellent) higher being better. Then for a signal p the MOS can be obtained:

$$\mathrm{MOS_p} = \frac{\sum_{n-1}^{N} R_n}{N} \tag{2.46}$$

where R_n is the score assigned by human assessor n and N is the total number of assessors for the signal p. There exist many standards for the gathering of MOS scores. Some, such as (*Subjective test methodology for evaluating speech communication systems that include noise suppression algorithm.* 2003) task the human participants with evaluating the audio over a number of dimensions, while others such as Multiple Anchor, Hidden Reference Assessment (MUSHRA) (International Telecommunication Union, 2015) involve an intrusive, direct comparison between a number of signals simultaneously.

2.7.2 Signal to Noise Ratio

One of the fundamental signal assessment measures is the Signal-to-Noise-Ratio (SNR). This can be broadly defined as the power of the desired signal s compared to the power of the undesired noise v:

$$\mathrm{SNR_{dB}}(x) = 10\log_{10}(\frac{s_P}{v_P}) \tag{2.47}$$

where the power of the speech signal $s[n]$, s_P can be estimated by $\frac{1}{N}\sum_{n=0}^{N-1} s[n]^2$. Calculation of SNR requires either the true value of $v[n]$ and $s[n]$ or an estimation of them.

2.7.3 Scale Invariant Signal Distortion Ratio

The SI-SDR (Roux et al., 2018) is a widely used intrusive metric for a number of audio tasks including SE. It is defined for a given noisy input signal $x[n]$ as

$$\mathrm{SI-SDR}(s[n], x[n]) = 10\log_{10}\frac{\left\|\frac{\langle x[n], s[n]\rangle s[n]}{\|s[n]\|^2}\right\|^2}{\left\|x[n] - \frac{\langle x[n], s[n]\rangle s[n]}{\|s[n]\|^2}\right\|^2} \tag{2.48}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product between the two signals. As a ratio, the value is unbound; higher values indicate that the input $x[n]$ is closer to the reference signal $s[n]$. SI-SDR can be used directly as a loss function in the training of NNSE systems.

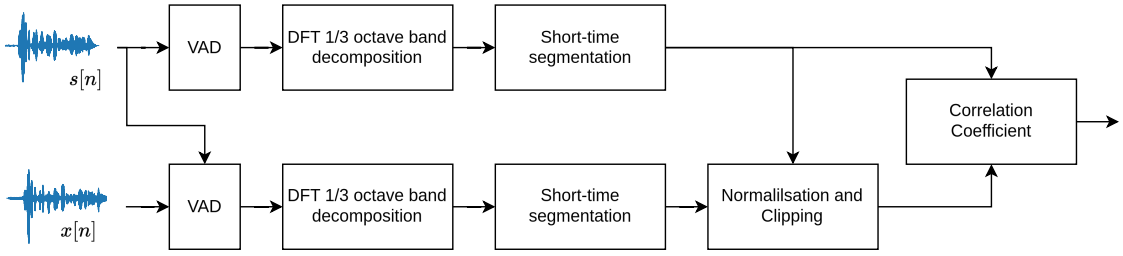

Figure 2.25: *Block diagram of STOI score calculation.*

2.7.4 STOI

Short-Time Objective Intelligibility (STOI) (Taal et al., 2011) is a intrusive measure of speech intelligibility, calculated via a correlation between the reference and degraded signal. It ranges from 0 and 1 (although it is sometimes expressed as a percentage between 0 % and 100 %). The score represents the predicted percentage of words a listener will identify correctly in a listening test. The block diagram for STOI is shown in Figure 2.25. The first computational step is a simple Voice Activity Detector (VAD) stage to remove those parts of the signal (non-speech) which do not contribute to the intelligibility. This is followed by the calculation of 1/3 octave filter-bank representations of the clean and degraded signals. These are then segmented into discrete blocks, and the correlation between these blocks for the reference and degraded signal is computed. The final STOI score is then derived from an average over the duration in time of the inputs.
Formally, the STOI score of some audio signal $x[n]$ is given by:

$$q^{x}_{\mathrm{STOI}} = \mathrm{STOI}(s[n], x[n]) \tag{2.49}$$

Several variants and improvements on STOI have been developed, such as Extended Short-Time Objective Intelligibility (ESTOI) (Jensen & Taal, 2016) which also incorporates spectral correlation, and Modified Binaural Short-Time Objective Intelligibility (MBSTOI) (Andersen et al., 2018) which computes the intelligibility of binaural (2-channel) signals. Further, STOI can be implemented in a way which allows it to be used directly as a loss function for the training of NNSE systems (Fu, Wang, et al., 2018).

2.7.5 PESQ

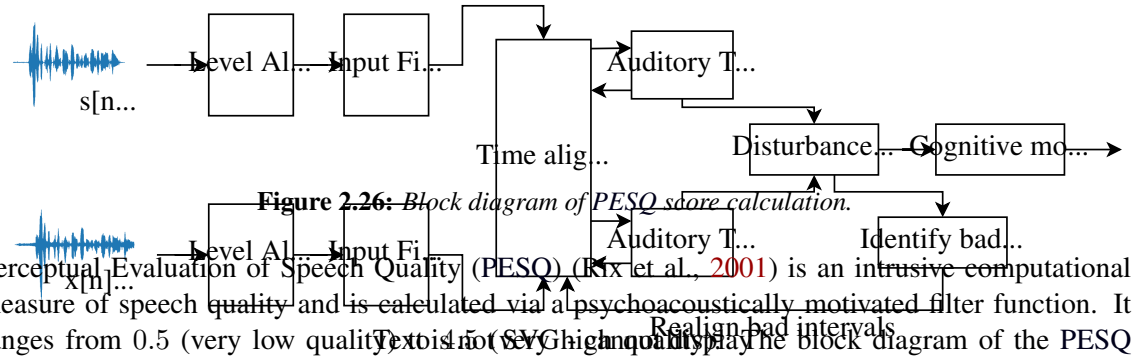

Figure 2.26: *Block diagram of PESQ score calculation.*

Perceptual Evaluation of Speech Quality (PESQ) (Rix et al., 2001) is an intrusive computational measure of speech quality and is calculated via a psychoacoustically motivated filter function. It ranges from 0.5 (very low quality) to 4.5 (very high quality). The block diagram of the PESQ function in shown in Figure 2.26. Unlike STOI which assumes that the two signals are aligned in

time, here the degraded and reference signals are aligned in blocks before being processed by the cognitive modelling step.
Many variants and successors to PESQ have been proposed, such as Perceptual Objective Listening Quality Analysis (POLQA) (Beerends et al., 2013) and Perception Model-Based Quality (PEMO-Q) (Huber & Kollmeier, 2006). However, PESQ remains a widely used metric for the SE task.

Formally, the PESQ score for a degraded audio signal $x[n]$ is given by:

$$q^x_{\text{PESQ}} = \text{PESQ}(s[n], x[n]) \tag{2.50}$$

The formulation of PESQ is non-differentiable, so direct use of it as a loss function for training NNSE models is not possible. This non-differentiablity is caused by a non-deterministic computation which happens with the disturbance processing stage. A reformulation of PESQ which allows it to be directly differentiable has been proposed (Martín-Doñas et al., 2018).

2.7.6 Composite Measure

Composite Measure (Lin et al., 2019) is a metric derived in part from a weighting of PESQ and SNR where CSIG, CBAK and COVL represent predictions of MOS for signal distortion, background noise interference and overall speech quality respectively. These measures are valued between 0 and 5, higher being better. Formally, the Composite scores for a degraded audio signal $x[n]$ is given by:

$$[q^x_{\text{CSIG}}, q^x_{\text{CBAK}}, q^x_{\text{COVL}}] = \text{Composite}(s[n], x[n]) \tag{2.51}$$

2.7.7 DNSMOS

Deep Noise Suppression Mean Opinion Score (DNSMOS) (K. A. Reddy et al., 2020) is a *non-intrusive* speech quality metric. It consists of a neural network which was trained to predict human MOS ratings for speech signals. As it is non-intrusive, it is particularly useful for assessing the quality of real-world recordings such as in the CHiME-7 UDASE challenge testset (Leglaive et al., 2023), and was one of the evaluation metrics used in assessing the entries to the challenge.
For a input time domain speech signal $x[n]$ DNSMOS estimates three values, being estimates of the well-known composite measure (Lin et al., 2019):

$$[q^x_{\text{SIG}}, q^x_{\text{BAK}}, q^x_{\text{OVR}}] = \text{DNSMOS}(x[n]), \tag{2.52}$$

where $q^x_{\text{SIG}}, q^x_{\text{BAK}}, q^x_{\text{OVR}}$ are each values between 1 and 5 which represent the estimated speech quality, background noise quality and overall quality, respectively (higher values indicating better quality). In this work the non-differentiable implementation of DNSMOS provided in the CHIME-7 (Leglaive et al., 2023) baseline system is used.

2.7.8 HASPI

The Hearing Aid Speech Perception Index (HASPI) (Kates & Arehart, 2014) metric is a intrusive, conditional metric for speech intelligibility designed specifically to assess the performance of

hearing aid speech enhancement systems. HASPI scores fall within the range of 0 to 1 representing the predicted percentage of intelligible words in the input. The metric is defined as:

$$q_{\text{HASPI}}^{\hat{s}[n]} = \text{HASPI}(s[n], \hat{s}[n], \mathbf{a}) \tag{2.53}$$

where $\mathbf{a}$ is the audiogram (vector encoding the hearing loss) for a given individual in a given ear (i.e $\mathbf{a}^l$ and $\mathbf{a}^r$) for the left and right ear respectively. Note that HASPI contains its own internal hearing loss simulation, applying the effect of the hearing loss encoded in $\mathbf{a}$ to the input signal $\hat{s}[n]$.

2.8 Neural Network Metric and MOS Prediction

One of the core application of DNN in this work is in the design of speech quality prediction networks (Fu, Tsao, et al., 2018). These fall into two general categories, metric predictors and MOS predictors. In the former, a neural network is trained to approximate the behaviour of a signal processing based SE metric; both the predictor and the target metric can be intrusive (Fu, Tsao, et al., 2018; Z. Xu et al., 2021) or non-intrusive (Close et al., 2025; Fu, Yu, Hung, et al., 2021). An example training diagram for intrusive and non intrusive prediction of an intrusive metric and of an MOS predictor is shown in Figure 2.28.

In the non-intrusive case, a neural network is trained to directly predict human quality MOS (or intelligibility) scores from audio (K. A. Reddy et al., 2020; Kumar et al., 2023; Mittag et al., 2021; Tamm et al., 2022); here the predictor can be intrusive or non-intrusive depending on the available data or use case. The main difference between the two categories as used in this work is the manner in which they can be trained relative to the training of a NNSE system. Metric predictor networks can be trained *in the loop* with NNSE systems (Close, Hain, et al., 2022; Fu et al., 2019; Fu, Yu, Hsieh, et al., 2021) as new values of the target metric can be computed using the training time enhanced audio outputs of the NNSE system. As it is not realistically feasible to gather true MOS labels for these training time outputs, neural MOS predictors must be trained prior to their use in NNSE training. More generally, the potential training of a metric predictor is limited only by the availability of distorted audio (with corresponding reference), while training data generation of MOS predictors requires costly human listening tests.

2.9 Datasets , Challenges and Corpora for Speech Enhancement

2.9.1 VoiceBank-DEMAND

VoiceBank-DEMAND (VB-D) (Valentini-Botinhao et al., 2016) is a widely used dataset for speech enhancement neural network training. It consists of clean English read speech, artificially corrupted with environmental noise from the DEMAND (Thiemann et al., 2013) noise dataset, as well as two additional noise types: speech-shaped noise (SSN) and babble. Note that the babble noise was created by randomly overlapping the clean speech audio. The clean speech files vary in length from around 3 to 10 seconds, while the DEMAND noise recordings are all 10 minutes long. The noisy signal $x[n]$ is created by adding a random part of the noise recording $v[n]$ of the same length as the clean speech $s[n]$.

$$x[n] = s[n] + c \cdot v[n] \tag{2.54}$$

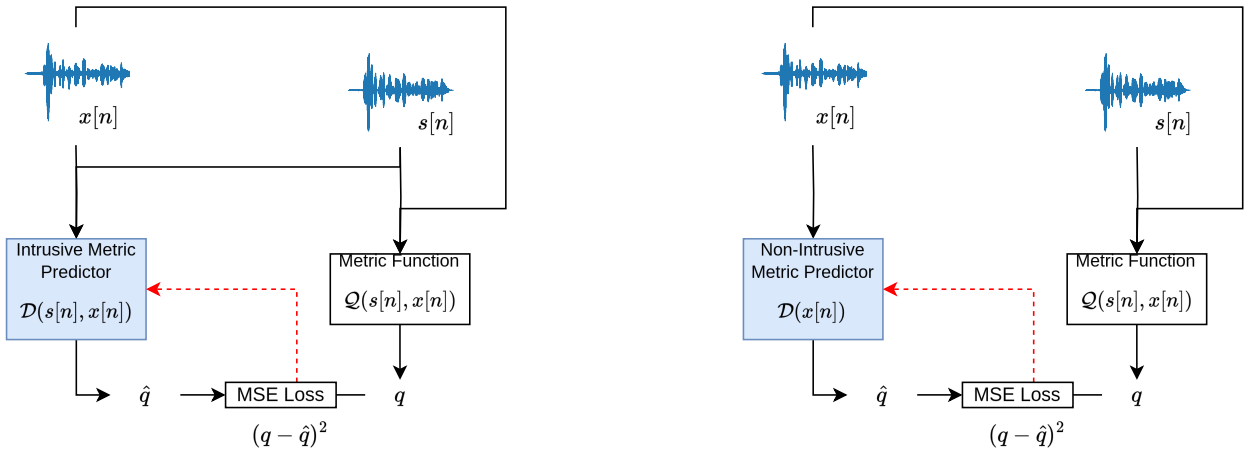

Figure 2.27: *Training of intrusive and non-intrusive metric predictors of an intrusive metric.*

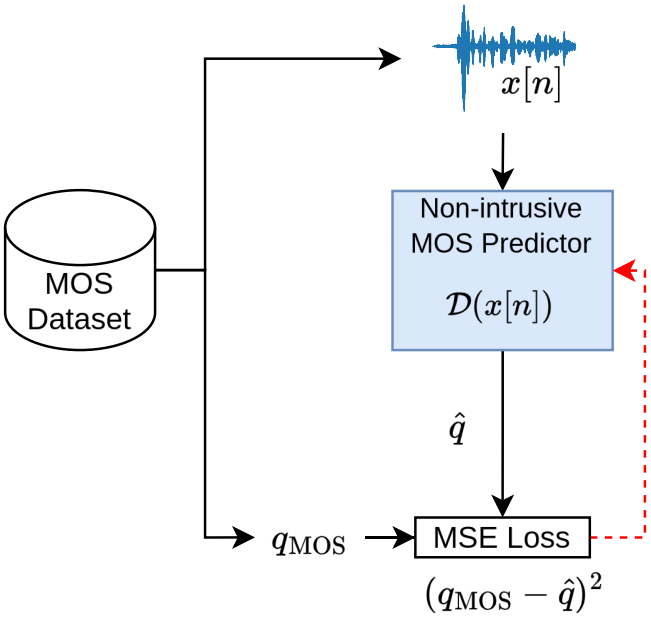

Figure 2.28: *Training of MOS predictor.*

This scaling factor

$$c = \sqrt{\frac{P_s}{P_v \cdot 10^{\frac{\text{SNR}}{10}}}} \tag{2.55}$$

is computed using a given target mixing SNR, and using the ITU-T P.56 method (Sector, 2011) for computing the active speaker power of the clean (speech) reference audio P_s and of the noise P_v. The training set consists of 11572 pairs of clean and noisy speech, $(s[n], x[n])$, from 28 different speakers (14 male, 14 female) with native British accents speaking English. The clean speech is mixed at 0, 5, 10 and 15 dB SNR with cafeteria, car, kitchen, meeting, metro, restaurant, station, and traffic noise from DEMAND as well as babble and speech-shaped noise. The audio from two speakers are held-out from training and used as a validation set whenever a model is trained using VoiceBank-DEMAND in this work. The test set consists of 824 $(s[n], x[n])$ pairs from two additional speakers (one male, one female) who do not appear in the training set, mixed at 2.5, 7.5, 12.5 and 17.5 dB SNR with bus, cafe, living room, office and public square noise from DEMAND. All audio has a sample rate of 48000 Hz and is in WAV format, but the dataset is typically down-sampled to 16000 Hz for speech enhancement training.

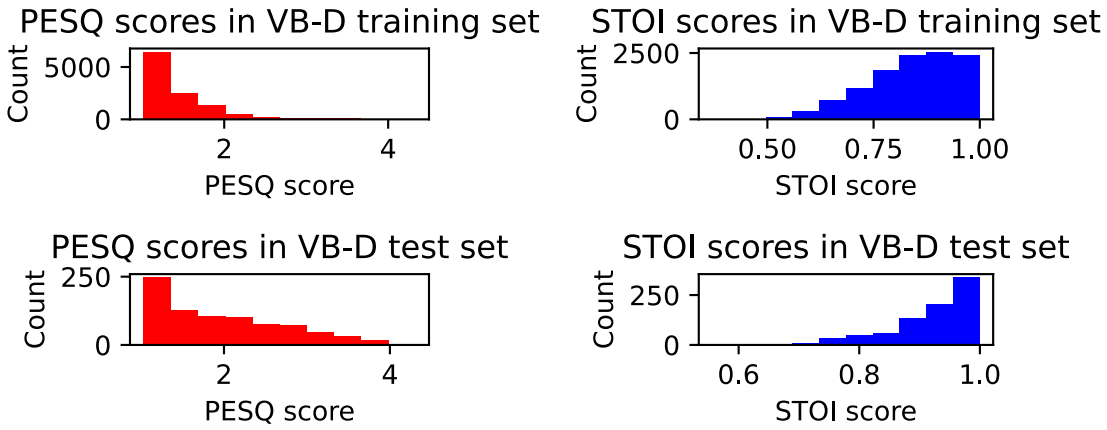

Figure 2.29: *PESQ and STOI distributions in the VoiceBank-DEMAND training and test sets.*

Figure 2.29 shows the distribution of PESQ and STOI scores in the VoiceBank-DEMAND training and test sets. From these, it can be observed that the test set contains audio of a higher quality and intelligibility compared to the training set; the average PESQ and STOI scores of the training set are 1.47 and 0.84 respectively, while that of the test set are 1.97 and 0.92.

Table 2.1 and Table 2.2 break down SE metrics in the VoiceBank-DEMAND training set in terms of the mixing SNR and noise type respectively. From the former, it can be observed that the mixing SNR has a very strong relationship with the SE metrics, where higher mixing SNR values result in less distorted noisy signals. From the latter, it is clear that the more similar the noise is to speech, the lower the SE metric scores, with the most distorting noise types being babble and speech shaped noise (ssn). The station noise type is also destructive as it tends to contain lound,impulsive sounds which completely envelop the speech, especially at low SNR values. Table 2.3 and Table 2.4 break down the SE metrics for the VoiceBank-DEMAND testset. Similar patterns to those in the training set are apparent here, however the values are overall higher owing to the higher mixing SNR values. VoiceBank-DEMAND is a widely used benchmark dataset for single channel NNSE task. However, in recent years it has become too 'easy' in particular due to the fact that the test set is significantly less challenging to enhance than the training set, as it has less disruptive noise types and higher mixing SNR values. One of the aims of this work is to introduce more challenging alternatives to VoiceBank-DEMAND which are detailed in Section 4.3.1.2 where the dataset is rerecorded in

Table 2.1: *Breakdown of SE metrics by mixing SNR value in the VoiceBank-DEMAND training set.*

Mixing SNR (dB)	PESQ	STOI	CSIG	CBAK	COVL
15	**1.85**	**0.91**	**3.09**	**2.61**	**2.45**
10	1.53	0.87	2.62	2.17	2.03
5	1.30	0.83	2.21	1.79	1.69
0	1.16	0.76	1.85	1.48	1.43

Table 2.2: *Breakdown of SE metrics by distortion noise type in the VoiceBank-DEMAND training set.*

Noise Type	PESQ	STOI	CSIG	CBAK	COVL
metro	1.55	0.86	2.87	2.09	2.16
car	**2.40**	**0.96**	**4.19**	**2.68**	**3.31**
ssn	1.26	0.79	1.51	1.80	1.33
traffic	1.38	0.87	2.78	2.00	2.04
kitchen	1.55	0.92	2.08	2.26	1.80
babble	1.26	0.77	2.10	1.77	1.58
cafeteria	1.33	0.81	2.38	1.89	1.79
station	1.27	0.82	2.53	1.86	1.84
meeting	1.38	0.83	2.46	2.01	1.85
restaurant	1.24	0.77	1.53	1.79	1.32

a real environment and Section 8.2 which proposes a simulation framework wherein VoiceBank-DEMAND like datasets in a number of languages and varying mixing SNR values can be produced.

2.9.2 CHiME3 Data

The Computational Hearing in Multisource Environments 3 (CHiME3) (Barker et al., 2015) challenge test set consists of multi channel real and simulated noisy speech (1320 clean/noisy pairs of each). The read speech text in both cases is sourced from the Wall Street Journal (WSJ0) (Paul & Baker, 1992) corpus. Of particular interest is the real component of the data which was recorded in real noisy environments (a bus, cafe, pedestrian area, and street junction) by real speakers, with the reference audio coming from a close-talking headset microphone. Of the 6 recording channels, the channel closest to the speaker is selected as the noisy input. Testing a trained NNSE system on this test set gives a good insight in to how it generalises to real world data.

Table 2.3: *Breakdown of SE metrics by mixing SNR value in the VoiceBank-DEMAND testset.*

Mixing SNR (dB)	PESQ	STOI	CSIG	CBAK	COVL
17.5	**2.60**	**0.96**	**4.05**	**3.17**	**3.33**
12.5	2.10	0.94	3.59	2.63	2.83
7.5	1.76	0.92	3.14	2.21	2.42
2.5	1.42	0.87	2.62	1.77	1.96

Table 2.4: *Breakdown of SE metrics by distortion noise type in the VoiceBank-DEMAND testset.*

Noise Type	PESQ	STOI	CSIG	CBAK	COVL
cafe	1.49	0.88	2.72	2.14	2.06
bus	2.48	0.95	3.96	2.74	3.22
living	1.61	0.90	2.78	2.17	2.15
psquare	1.74	0.91	3.26	2.33	2.47
office	**2.53**	**0.96**	**4.01**	**2.83**	**3.27**

2.9.3 CHiME7 - UDASE Data

The CHiME7 UDASE (Leglaive et al., 2023; Leglaive et al., 2024) challenge is focused on the removal of additive noise from reverberant overlapping speech. In particular, the challenge task is concerned with adapting models trained on labelled 'out-of-domain' data to unlabelled 'in-domain' data. Challenge participants were provided with three datasets: an in-domain unlabelled training set (Barker et al., 2018), and out-of-domain labelled set (Cosentino et al., 2020) and a close to in-domain labelled development/evaluation set. Of particular relevance to this work was the choice of evaluation metric; in the first round of evaluation, the neural MOS predictor metric DNSMOS (see Section 2.7.7) was used to select the best submissions to be evaluated in the second round. In the second round, human listening tests were carried out.

2.9.4 CommonVoice Dataset

The CommonVoice (Ardila et al., 2020) corpus consists of recordings of read speech in 108 languages, with corresponding text prompt sentences. The recordings are crowd-sourced using the CommonVoice website[5]. Validation that the recordings properly represent the prompt sentence is also crowd-sourced. In addition to the audio recording and prompt sentence text, some additional metadata is for a subset of the recordings available such as gender and accent of the speaker. While the primary intended use of the CommonVoice data is the training / fine-tuning of ASR systems, it is a useful source for speech audio generally.

2.10 Datasets for Speech Intelligibility (SI) Prediction

2.10.1 Clarity Prediction Challenge 1

The dataset for the first CPC1 (Barker et al., 2022) as used in this work can be expressed as a series of sequences: $(\hat{\mathbf{s}}[n], \{\mathbf{a}_l, \mathbf{a}_r\}, i)$, which is generated as visualised in Figure 2.30. $\hat{\mathbf{s}}[n]$ represents the binaural output of a hearing aid system for some noisy speech input $\mathbf{x}[n]$, containing some clean speech $\mathbf{s}[n]$. $\{\mathbf{a}_l, \mathbf{a}_r\}$ are the left (l) and right (r) audiogram representations of a particular listener's hearing loss. Blue and red box plots in Figure 2.30 illustrate the Hearing Loss (HL) distribution in the CPC1 dataset from which the individual audiograms are sampled. Finally, i represents the intelligibility of the audio $\hat{\mathbf{s}}[n]$ for that listener, defined as the percentage of words they were able to reproduce by speaking aloud immediately after hearing the audio, compared to a

[5]https://commonvoice.mozilla.org

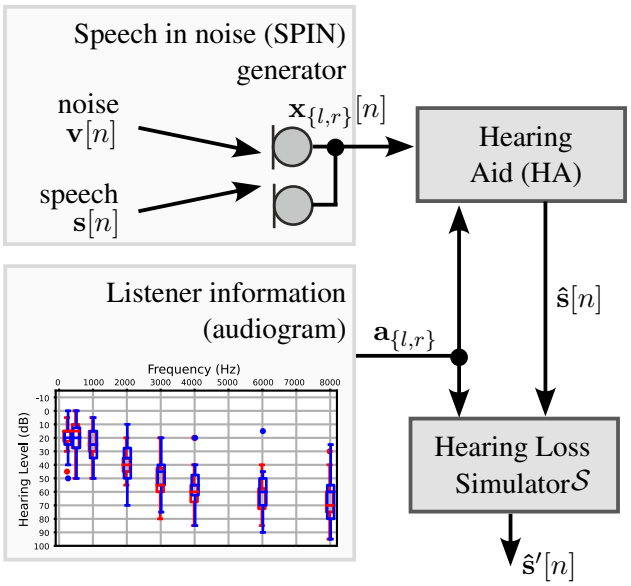

Figure 2.30: *Signal generation for Clarity Prediction Challenge.*

ground truth transcription of the speech also denoted as the *correctness* of the listener's response. Additionally, binaural audio $\hat{\mathbf{s}}'[n]$ is defined as the output of the baseline Cambridge MSBG Hearing Loss Simulation (HLS), denoted here by operator $\mathcal{S}$, cf. (Stone & Moore, 1999) for additional details on the Clarity system.

$$\hat{\mathbf{s}}'[n] = \mathcal{S}(\hat{\mathbf{s}}[n], \{\mathbf{a}_l, \mathbf{a}_r\}) \tag{2.56}$$

The signal $\hat{\mathbf{s}}'[n]$ is an approximation of the audio that is perceived by the hearing-impaired listener. This can be thought of as encoding the hearing characteristics of the specific listener (audiogram) within the signal.

2.11 Datasets for Speech Quality (SQ) MOS Prediction

Table 2.5: *Comparison of SQ Datasets (EN: English; DE: German, CH: Chinese).*

Dataset	Subset	Distortions	Lang.	Distortion Types	Samples	Average MOS
NISQA (Mittag et al., 2021)	TRAIN_SIM	Simulated	EN	white Gaussian noise, MNRU noise, background noise clips, codecs, packet loss, amplitude clipping	10000	2.91
	TRAIN_LIVE	Real	EN	live telephone/VoIP, white gaussian noise, MNRU noise, background noise clips, codecs, packet loss, amplitude clipping	1020	2.23
	VAL_SIM	Simulated	EN	codecs, packet loss, speech level, frequency filters, amplitude clipping	2500	2.97
	VAL_LIVE	Real	EN	live telephone/VoIP	200	
	TEST_FOR	Simulated	EN	audio codecs, background noise, packet-loss, amplitude clipping, live condidtion VoIP	240	2.40
	TEST_P501	Simulated	EN	live condition VoIP/mobile network recordings	240	2.60
	TEST_LIVETALK	Real	DE	natural environment conditions	232	2.76
Tencent (Yi et al., 2022)	w/ reveb	Simulated	CN	reverberation	3297	2.90
	w/o reverb	Simulated		white noise, background noise, codecs, frequency filtering, amplitude clipping	8366	
IUB (Dong & Williamson, 2020)	COSINE	Real	EN	background noise	18000	3.12
	VOiCES	Semi-Simulated		background noise, reverberation	18000	
PSTN (Mittag et al., 2020)	–	Simulated	EN	background noise	58709	3.12

Several SQ datasets are used in this work to train non-intrusive SQ MOS predictors. It is important to consider a large number of datasets in order to ensure that the MOS SQ predictor has been

exposed to a large variety of audio conditions during it's training. The nature of distortions, language, dataset size in terms of number of samples and average signal quality are summarised in Table 2.5. Generally, these datasets consist of sets of tuples $(x[n], q)$ where $x[n]$ is some degraded speech audio signal and q is a corresponding MOS value which has been calculated from human listening tests. For some datasets and subsets within datasets, other information is available such as a reference signal $x[n]$, the standard deviation of the MOS score, the raw score assigned by each human evaluator or the number of human evaluators.

2.11.1 NISQA Dataset

The Non-Intrusive Speech Quality Assessment (NISQA) (Mittag et al., 2021) dataset is an SQ assessment dataset, comprising of pre-defined train, validation and test sets. Each of these are further divided into subsets, characterised by if the nature of the distortion in the speech signal is artificially simulated or occurring 'in the wild' as a real distortion. In addition to a MOS scores of overall audio quality, The NISQA dataset also provides labels for other speech 'dimensions'(Wältermann, 2013) namely Noisiness, Coloration, Discontinuity and Loudness. With the exception of the LIVETALK testset, clean reference signals $x[n]$ are available.

2.11.2 Tencent Dataset

The Tencent audio SQ dataset was released as part of the ConferencingSpeech 2022 challenge (Yi et al., 2022). It consists of two artificially simulated training subsets, one with artificial reverberation added and one without.

2.11.3 IUB Dataset

The IU Bloomington (IUB) (Dong & Williamson, 2020) SQ dataset consists of two subsets. The first uses distorted audio sourced from the COnversational Speech In Noisy Environments (COSINE) (Hashmi, 2021) dataset, real multi party conversations captured using multi-channel wearable microphones recorded in noisy everyday environments. The second subset uses audio from the Voices Obscured in Complex Environmental Settings (VOiCES) (Richey et al., 2018) corpus where speech and noise were played aloud and recorded in two rooms of different sizes.
Unlike the other datasets used in this work, the MOS scores for this dataset were gathered using a MUSHRA (International Telecommunication Union, 2015) protocol, which is then transformed to a MOS scale between 0 and 10, rather than the 1 to 5 scale commonly used. The 1 - 5 MOS label is obtained via a fitting operation over the gathered MUSHRA ratings.

2.11.4 PSTN Dataset

The Public Switched Telephone Network (PSTN) SQ dataset (Mittag et al., 2020) consists of simulated 'real' phone calls, some with simulated background noise added to the transmitted signal. It was generated by making real phone calls over Skype. It follows a similar design to that of NISQA, but is significantly larger.

2.11.5 Overall MOS Distribution of SQ Datasets

The distributions of MOS scores in the training and validation subsets of the datasets (normalised between 0.2 and 1) are shown in Figure 2.31. The mean MOS value across the datasets is similar, falling somewhere around 0.6. However, the datasets differ significantly in the shape of their distributions. Both NISQA and Tencent show a roughly uniform distribution of scores from 0.2 to 1, with the 'tail' at the lower end of the Tencent distribution showing that that dataset contains a large number of low scores. Conversely, the tapering in at the highest end in both NISQA and Tencent indicate that these datasets contain relatively few instances of very highly rated audio.

In contrast, the distribution of the PSTN dataset scores is generally normal, tailing off a the extreme low and high end. Slightly more scores are above 0.5 than below, indicating that the audio in this dataset is generally high quality.

The distribution of the MOS score in the IUB dataset is entirely unlike that of the others, with extremely few points falling at the highest and lowest values. The distribution is significantly more erratic than the other datasets, with an extreme dearth in scores valued around 0.65. This can possibly be explained by the non-standard method that the MOS scores were gathered, as well as the differing range of the unnormalised scores.

The combined distribution across all the datasets is shown in purple at the top of Figure 2.31. It displays a similar normal-like distribution to that of the PSTN dataset, likely due to that dataset contributing roughly half of all samples. There are somewhat more samples of extreme low quality compared to extreme high quality.

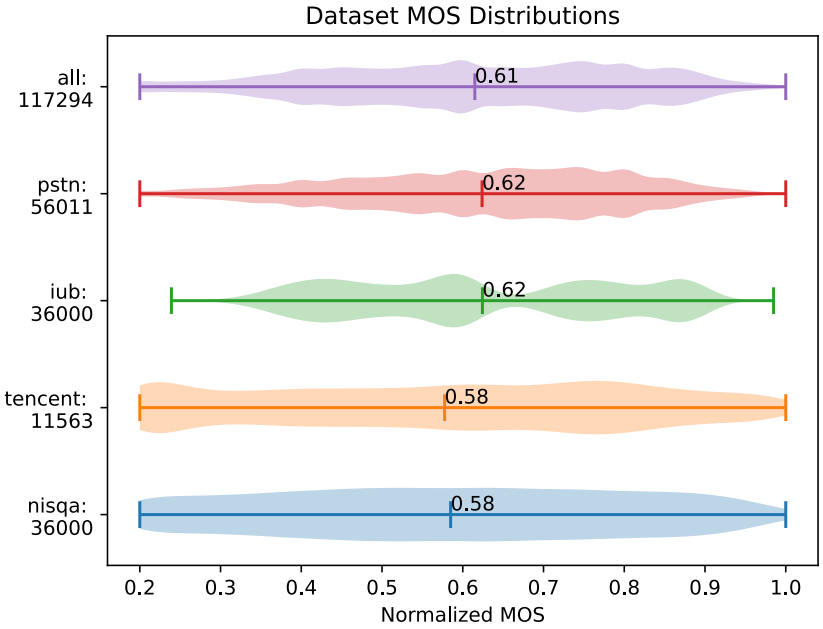

Figure 2.31: *Normalised MOS score distribution across SQ Datasets (lines indicate minimum, mean and maximum MOS in each dataset).*

2.12 Baseline NNSE System - MetricGAN+

The MetricGAN+ NNSE framework (Fu, Yu, Hsieh, et al., 2021) consists of two networks: a speech enhancement model $\mathcal{G}$, which aims to remove the undesired signal parts, i.e the noise $v[n]$ from a noisy signal $x[n]$ to produce an estimate of a clean signal $s[n]$, and an intrusive metric prediction discriminator (more correctly an evaluator) $\mathcal{D}$, which predicts the intrusive SE performance metrics providing a target to optimise the signal enhancement. The phase of the spectral bins $\angle p_{k,\ell}$ will be used later to resythesize the time domain signal using the Overlap-Add (OLA) method.

2.12.1 Generator Network for Signal Enhancement

Figure 2.32 shows the training of the NNSE Generator $\mathcal{G}$. The dotted blue arrows and processes show the objective function and loss calculation back-propagated to the model. In order to obtain the enhanced signal $\hat{s}[n]$ from the noisy features $\mathbf{X}_f$ in the generator $\mathcal{G}$'s training and inference, the magnitude compression is reversed by subtracting 1 from each element and taking the exponential of each element in the feature representation. The output of $\mathcal{G}$ is a time-frequency (T-F) mask

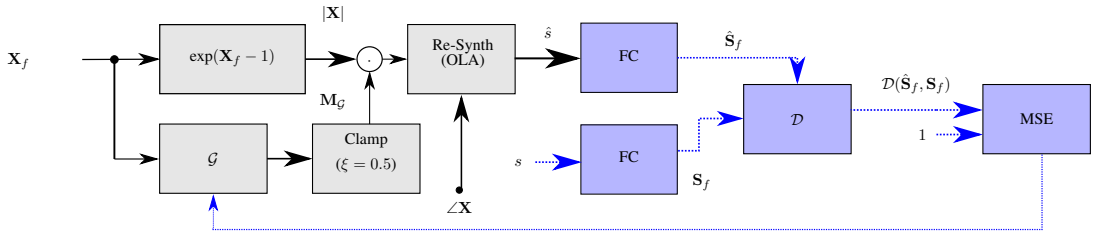

Figure 2.32: *Training and inference of MetricGAN+ Generator.*

matrix $\mathbf{M}_\mathcal{G}$, which is then multiplied with the noisy magnitude spectogram $|\mathbf{X}|$ to result in the enhanced signal spectrogram $|\hat{\mathbf{S}}|$. The enhanced time domain audio signal $\hat{s}[n]$ is calculated using OLA resythesis, using the noisy phase information $\angle\mathbf{X}$. Note that each element in the mask $\mathbf{M}_\mathcal{G}$ is 'clamped' in order to reduce residual musical tones caused by the mask, i.e. it is limited to element wise values $\xi \leq \mathbf{M}_\mathcal{G} \leq 1$. The loss function of the speech enhancement network $\mathcal{G}$ is dependent entirely on the metric score of its output $\hat{s}[n]$ (in its feature space representation $\hat{\mathbf{S}}_f$) as predicted by discriminator $\mathcal{D}$.

$$L_{\mathcal{G},\text{MG+}} = \mathbb{E}[(\mathcal{D}(\hat{\mathbf{S}}_f, \mathbf{S}_f) - 1)^2] \tag{2.57}$$

where 1 represents a 'perfect' score in the normalised metric $Q'(\cdot)$.

2.12.2 Discriminator Network for Metric Prediction

The discriminator $\mathcal{D}$ is trained to reproduce the normalised target metric $Q'(\cdot)$ minimising the distance from its output and the 'true' normalised metric score used as its loss function, as visualised in Figure 2.33. Arrows and processes marked blue denote those which occur only during training. The loss of the discriminator comprises three MSE terms depending on the clean reference signal s, or $\mathbf{S}_f$, the degraded noisy signal x, or $\mathbf{X}_f$, and the enhanced signal $\hat{s}$, or $\hat{\mathbf{S}}_f$. More specifically, its objective function is given as:

$$L_{D,MG+} = \mathbb{E}[(\mathcal{D}(\mathbf{S}_f, \mathbf{S}_f) - 1)^2 + (\mathcal{D}(\hat{\mathbf{S}}_\mathbf{f}, \mathbf{S}_f) - \mathcal{Q}'(\hat{s}[n], s[n]))^2 + (\mathcal{D}(\mathbf{X}_f, \mathbf{S}_f) - \mathcal{Q}'(x[n], s[n]))^2] \tag{2.58}$$

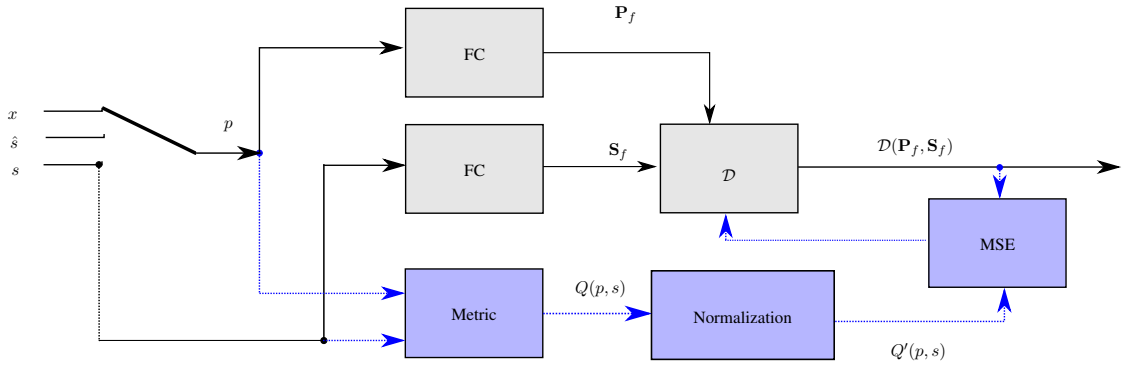

Figure 2.33: *Training and inference of MetricGAN+ Discriminator.*

The 1 in the first term of (2.58) represents the fact that $\mathcal{Q}'(s[n], s[n]) = 1$. In the second term, the scores of signals enhanced by $\mathcal{G}$, $\hat{s}$ are considered and compared to the ground truth score for the enhanced signal. In the final term, the scores of noisy signals x are considered and compared to the true score for the noisy signal. Note that in the case of the metrics investigated in this work the input to the function that defines the metric are the time domain signals $x[n]$, $\hat{s}[n]$ and $s[n]$, but this may not always be the case.

2.12.3 MetricGAN+ Training

Each epoch of training consists of four steps, the first three representing the training of $\mathcal{D}$ and the final step the training of $\mathcal{G}$. At the start of each epoch, I audio segments are randomly picked out from the training set. Firstly $\mathcal{D}$ is trained as given in (2.58) on these I random audio segments. The audio segments are time domain signals of varying length. Then, in the second step, $\mathcal{D}$ is trained using a 'replay buffer' where saved enhanced outputs of the generator $\mathcal{G}$ from past epochs are used to train $\mathcal{D}$. The size of this replay buffer is decided by a '`history_portion`' hyper-parameter H, which corresponds to the replay buffer growing by a set percentage of the audio segments observed each epoch. This is done to prevent $\mathcal{D}$ from 'forgetting' too much about the behaviour of $\mathcal{Q}'(\cdot)$ on previously enhanced speech.
Then the first step is repeated with $\mathcal{D}$ again being trained using the t random samples. Finally, $\mathcal{G}$ is trained also using these t samples as in (2.57). During training of the discriminator $\mathcal{D}$, the NNSE generator $\mathcal{G}$ is 'frozen' and its parameters are not updated; the opposite is true during $\mathcal{G}$'s training. Note that samples are added to the replay buffer during the first step of $\mathcal{D}$'s training, meaning that 20% of the 'current' epoch data are always present in the replay buffer. As $\mathcal{D}$ is trained before $\mathcal{G}$, the $\hat{s}$ in (2.58) actually represents the output of the previous epoch's $\mathcal{G}$. This is especially relevant during the first epoch training, as the $\hat{s[n]}$ in (2.58) is the output of a newly initialised, un-trained $\mathcal{G}$.

2.12.4 Discriminator Model Structure

The discriminator $\mathcal{D}$'s structure is shown in Figure 2.34. The input to the network is the magnitude spectrogram of the reference audio $\mathbf{S}_f$ and that of the signal under test $\mathbf{P}_f \in \mathbf{X}_f, \mathbf{S}_f, \hat{\mathbf{S}}_f$.The input is of shape $B \times T \times F \times 2$. The initial block is a Convolutional Neural Network (CNN) with four 2D convolutional layers with 15 filters of a kernel size of $(5, 5)$. After the convolutional layers, a mean is taken over the 2nd and 3rd dimensions (i.e the convolved time and frequency dimensions), and

this vector of length 15 is fed into three sequential linear layers, with 50, 10 and 1 output neurons, respectively. All of the network layers have a LeakyReLU activation while the final layer has no activation.

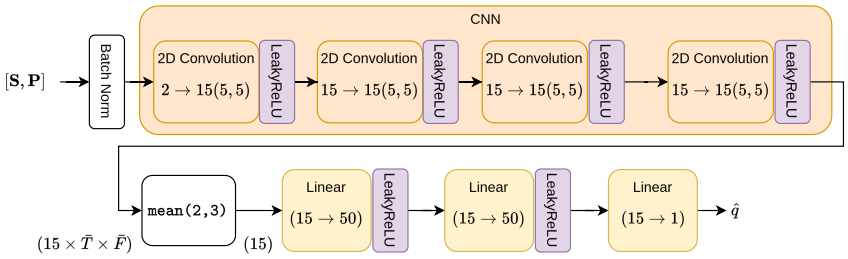

Figure 2.34: *MetricGAN+ $\mathcal{D}$ DNN structure*

2.12.5 Generator Model Structure

The generator $\mathcal{G}$'s network, shown in Figure 2.35 ,takes as input a magnitude spectrogram $\mathbf{X}_f$. The input is of shame $B \times T \times F$. The time dimension T of the input is preserved throughout the network, such that the output mask $\mathbf{M}_\mathcal{G}$ can be multiplied with the input. It's structure consists of a Bidirectional Long Short-Term Memory (BLSTM) (Weninger et al., 2015) unit with two LSTM layers with 200 neurons each. This is followed by two linear layers, the first with 300 output neurons and a LeakyReLU (Maas et al., 2013) activation and the second 257 output neurons with a 'Learnable' Sigmoid activation function. This Learnable Sigmoid is given as:

$$y_{\text{learnable-sigmoid}} = \frac{\beta}{1 + e^{-\alpha x}} \tag{2.59}$$

where β is a hyper-parameter (default set to 1.2) and α is a learnable parameter.

2.13 CMGAN SE DNN

The Conformer Metric Generative Adversarial Network (CMGAN) (Cao et al., 2022) is a variant of the MetricGAN framework within which the main change is a significantly more complex SE DNN $\mathcal{G}$. Figure 2.36 shows an overview of the CMGAN $\mathcal{G}$ NNSE structure; it consists of 4 blocks, an encoder, a Conformer based bottleneck, a mask decoder and a complex (mapping) decoder. The encoder takes as input the noisy magnitude, real and imaginary STFT components $\mathbf{X}_{\text{Mag}}, \mathbf{X}_{\text{Re}}, \mathbf{X}_{\text{Im}}$, stacked on a common dimension such that the input to the network is of shape $B \times T \times F \times 3$ where B, T and F are the batch size, time dimension and frequency dimensions, respectively. These are processed by a dilated DenseNet (Huang et al., 2017) consisting of 4 2D CNN layers with increasing dilation d. The final CNN layer halves the feature dimension from F to F' to reduce the complexity of the network. The encoder output is then processed by N TS-Conformer blocks. Each of these consists of two sequential Conformer blocks, the first of which operates over the time dimension of the input and the second over the frequency dimension. Each Conformer block has additive skip connections. The model has two output branches which share the output of the final TS Conformer as input. In the Mask Decoder branch a second dilated DenseNet further processes the output, followed by a so called *SubPixel* 2D Convolutional layer, which up-samples the feature dimension back to F from F'. This is followed by two final CNN layers which

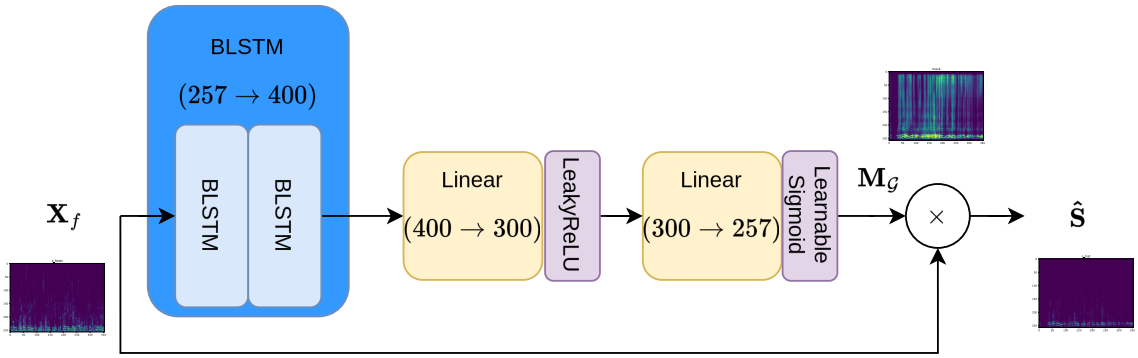

Figure 2.35: *MetricGAN+ $\mathcal{G}$ DNN structure*

project back from the 64 filters to 1 magnitude mask. The Complex decoder is structured similarly, except the filter dimension is reduced to 2, such that the output is the enhanced real and imaginary components.

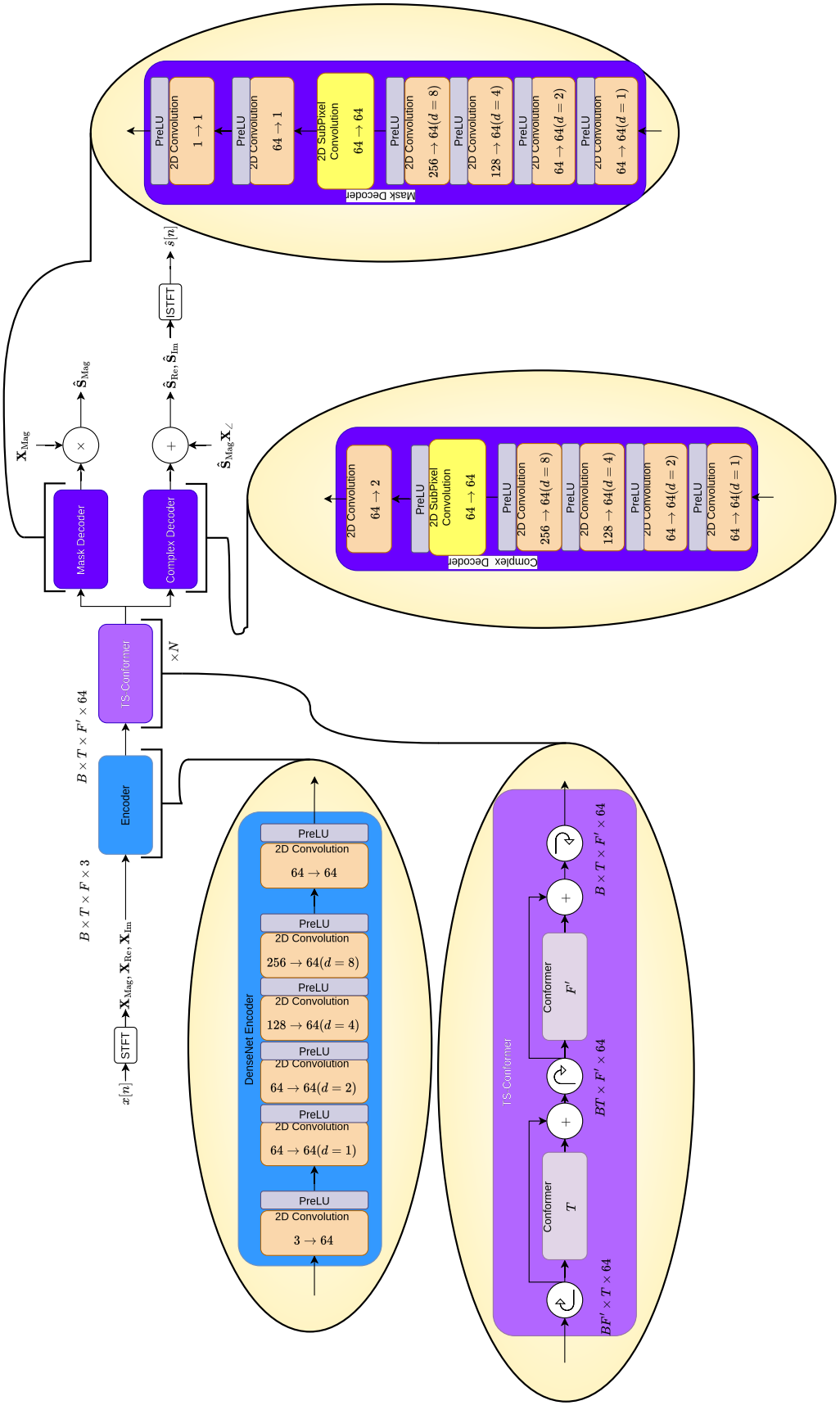

Figure 2.36: *CMGAN $\mathcal{G}$ DNN structure*

There are three component outputs, a magnitude mask $\mathbf{M}_{\mathcal{G}}$ and a real and imaginary component, $\hat{\mathbf{S}}'_{\mathrm{Re}}$ and $\hat{\mathbf{S}}'_{\mathrm{Im}}$. The magnitude mask is multiplied with the noisy magnitude $\mathbf{X}_{\mathrm{Mag}}$ to produce $\hat{\mathbf{S}}_{\mathrm{Mag}}$. Then, the combination of the enhanced magnitude $\hat{\mathbf{S}}_{\mathrm{Mag}}$ with the original noisy phase $\mathbf{X}_{\angle}$ are added with the other two outputs $\hat{\mathbf{S}}'_{\mathrm{Re}}$ and $\hat{\mathbf{S}}'_{\mathrm{Im}}$:

$$\hat{\mathbf{S}}_{\mathrm{Re}} = \hat{\mathbf{S}}_{\mathrm{Mag}}\cos(\mathbf{X}_{\angle}) + \hat{\mathbf{S}}'_{\mathrm{Re}}; \quad \hat{\mathbf{S}}_{\mathrm{Im}} = \hat{\mathbf{S}}_{\mathrm{Mag}}\sin(\mathbf{X}_{\angle}) + \hat{\mathbf{S}}'_{\mathrm{Im}}. \tag{2.60}$$

An Inverse Short Time Fourier Transform (ISTFT) is taken over $\hat{\mathbf{S}}_{\mathrm{Re}}, \hat{\mathbf{S}}_{\mathrm{Im}}$ to obtain $\hat{s}[n]$.

2.14 DPT-FSNet SE DNN

The Dual-Path-Transformer Full-band and Sub-band Fusion Network (DPT-FSNet) (Dang et al., 2022) is another SE DNN used in this work. Figure 2.37 shows an overview of DNN structure; it is generally similar to that of CMGAN $\mathcal{G}$, consisting of an Encoder, 'Dual Path' Transformer bottleneck and a Decoder. The Encoder takes as input the real and imaginary components of the noisy input audio (with a shape of $B \times T \times F \times 2$), and consists of a series of Dense CNN layers with skip connections. Unlike in CMGAN where the feature dimension is halved at input to the bottleneck, here it is the filter dimension of 64 which is reduced to 32. Transformer layers are used to process the time dimension T and the feature dimension F in sequence. At the output of the bottleneck after the features have been projected back to 64, two layers process the output of the TS Conformer in parallel with the resulting representations multiplied together. This is designed to act as a gating mechanism. The Decoder is structured similarly to the Encoder and outputs to 2 representing the predicted masks for the real and imaginary components.

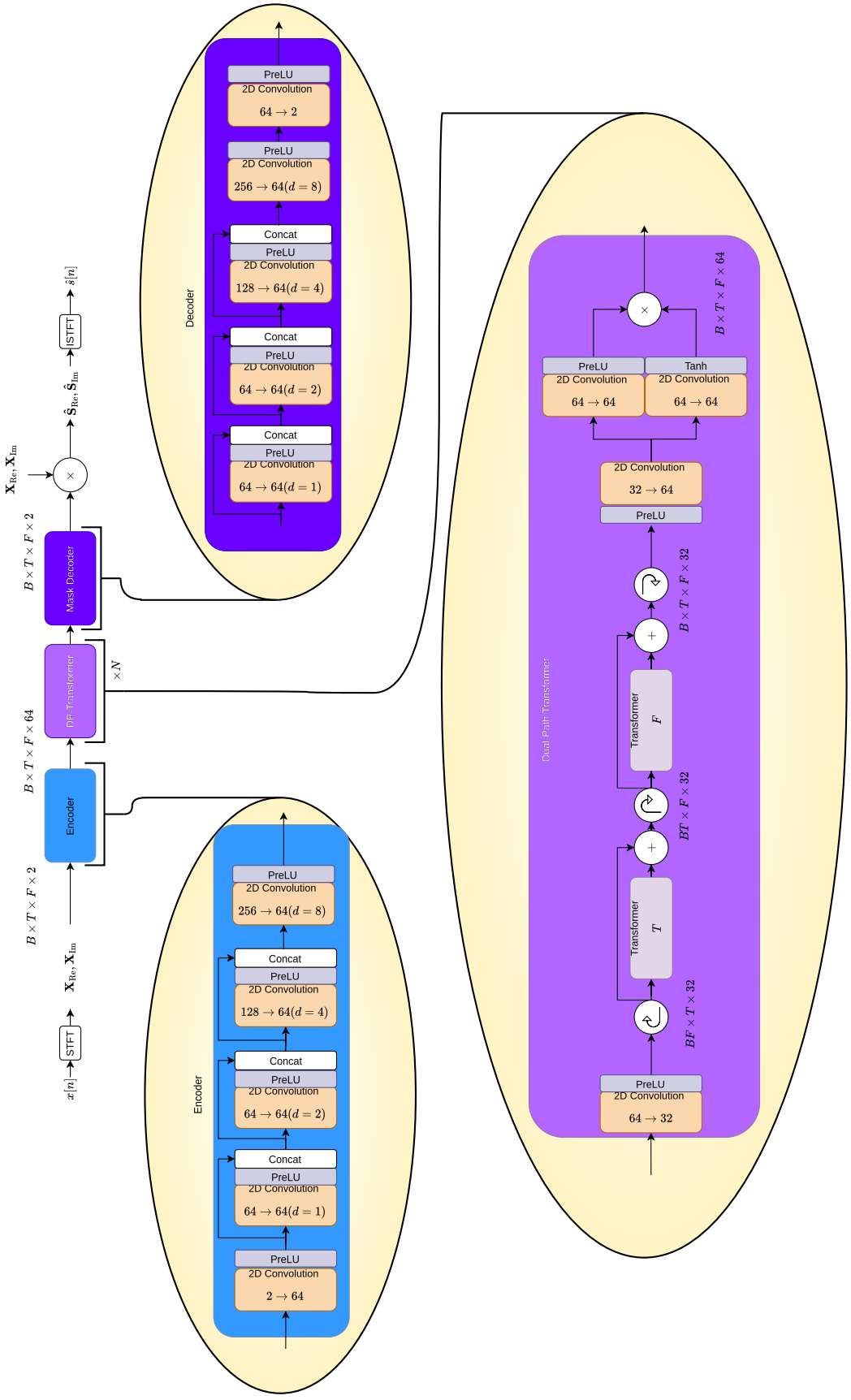

Figure 2.37: *DPT-FSNet DNN structure*

Part II

Expanding the MetricGAN Framework

Preface

In this part, variations and expansions to the MetricGAN (cf. Section 2.12) NNSE framework are proposed. In Chapter 3, an extension which is designed to improve the ability of the metric prediction discriminator to accurately predict the target metric is proposed. In Chapter 4, further experiments and variations involving this extension are detailed, as well as experiments involving the CMGAN (cf. Section 2.13) NNSE network. In Chapter 5 and Chapter 6 MetricGAN variants which incorporate the prediction of non-intrusive MOS estimators are proposed, as well as a novel input feature for the metric prediction discriminator derived from SSSRs.

Chapter 3

MetricGAN+/-: Improving Speech Enhancement Performance by Expanded Discriminator Training

3.1 Introduction

In this chapter MetricGAN+/- is detailed (an extension of MetricGAN+,cf. Section 2.12) which introduces an additional network - a *de-generator* to improve the robustness of the discriminator metric prediction network (and of the generator) by ensuring observation of a wider range of metric scores in training. Experimental results on the VoiceBank-DEMAND dataset show relative improvement in PESQ score of 3.8% (3.05 vs. 3.22 PESQ score), as well as better generalisation to unseen noise and speech signals from the CHiME3 testset.

3.2 Metric Score Distribution in Training Data

The central idea behind MetricGAN is that the performance of the NNSE Generator $\mathcal{G}$ is dependent entirely on how well well the metric prediction network $\mathcal{D}$ is able to predict the (normalised) objective metric score $\mathcal{Q}'$, via $\mathcal{G}$'s loss function, (2.57). It follows that in order for $\mathcal{D}$ to do this, it needs to be able to observe a the full range of values $0 < \mathcal{Q}' < 1$. However, this is not the case for the training data used in (Fu, Yu, Hsieh, et al., 2021) and other publications in this domain which rely on VoiceBank-DEMAND (Valentini-Botinhao et al., 2016) (cf. Section 2.9.1). As shown in Figure 3.1, the PESQ scores of the noisy data in VoiceBank-DEMAND training set are skewed significantly towards scores in the range of 1 to 2 PESQ (mean score is 1.47). The STOI distribution is even more biased, with no values below around 0.4, and a mean STOI score of 0.84. While it might be assumed that in early epochs the value $\mathcal{Q}'(\hat{s}[n], s[n])$ in (2.58) will be low, even in the initial epochs, the value of $\mathcal{Q}'(\hat{s}[n], s[n])$ is high for $\hat{s}[n]$. This can be observed in Figure 3.2, where, even in the first epoch, the historical set/replay buffer of $\mathcal{G}$ contains only enhanced audio $\hat{s}[n]$ audio with high STOI scores.

Due to the nature of the training of $\mathcal{G}$, it is unlikely that $\mathcal{D}$ will ever observe $\mathcal{Q}'$ values lower than those present in the training set. It can be theorised that using MetricGAN+, the discriminator $\mathcal{D}$

only learns values of a local version of the target metric $\mathcal{Q}'$ in the range $q'_{\min} < \mathcal{Q}' < 1$, where $q'_{\min}$ is the minimum $\mathcal{Q}'$ score present in the training set.

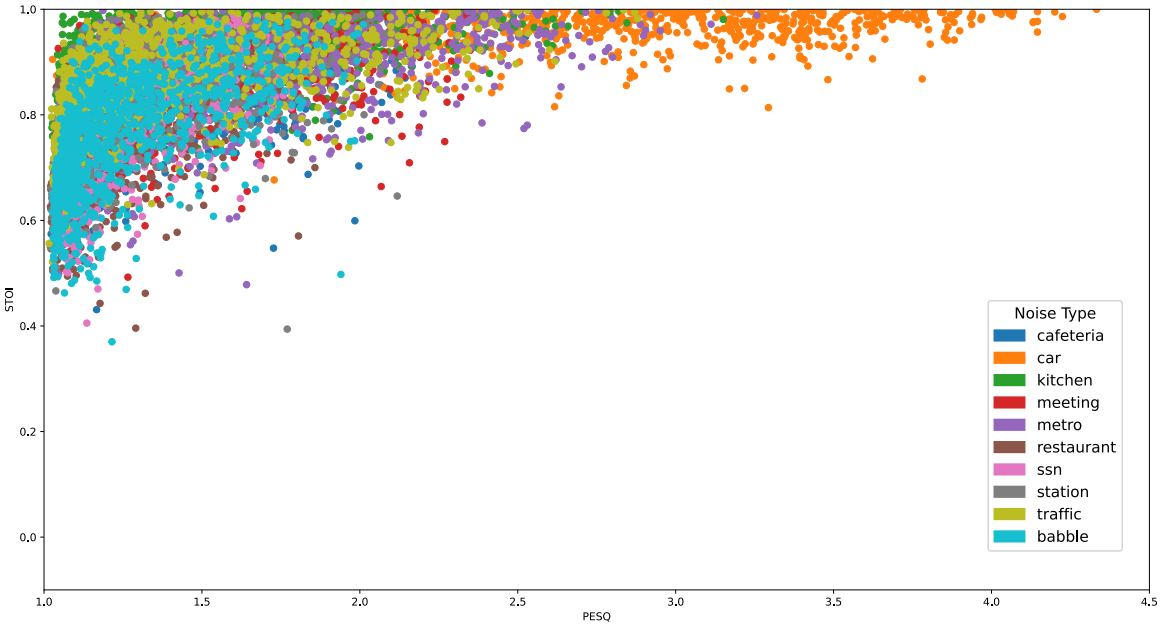

Figure 3.1: *PESQ and STOI distribution of the VoiceBank-DEMAND Training Set with Noise Type labels.*

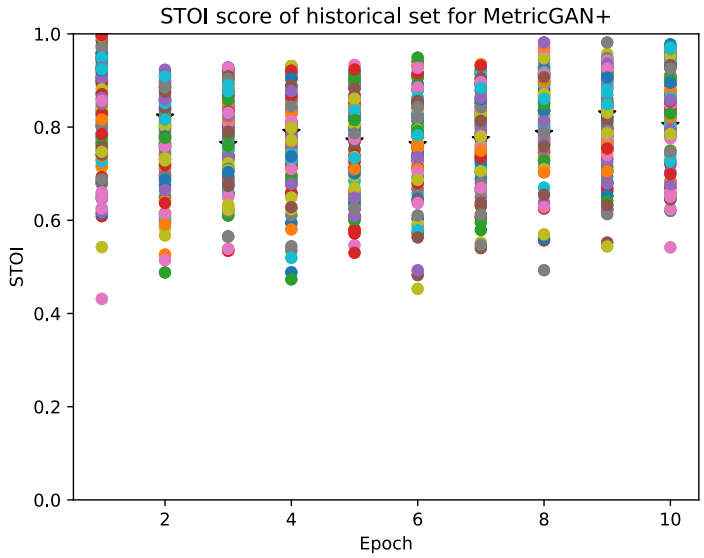

Figure 3.2: *STOI scores of the replay buffer of STOI objective MetricGAN+.*

This limits the ability of $\mathcal{D}$ to actually predict $\mathcal{Q}'$, and thus $\mathcal{G}$'s ability to enhance signals relative to $\mathcal{Q}'$ using $\mathcal{D}$ as a surrogate. To eliminate this limitation on $\mathcal{D}$'s training it is necessary to find a way to guarantee that it will observe a wider range of $\mathcal{Q}'$ values.

3.3 MetricGAN+/- Framework

The framework proposed in this chapter, MetricGAN+/-, expands on MetricGAN+ in one major way - the introduction of an additional network, a 'de-generator' $\mathcal{N}$ which, given an input signal $x[n]$, will attempt to output a signal with a *non-perfect* target score of metric $\mathcal{Q}'$. The key idea of this extension is to allow $\mathcal{D}$ to observe a wider range of metrics scores outside of those present in the training data. The output audio of $\mathcal{N}$'s mask $\mathbf{M}_{\mathcal{N}}$ applied to noisy magnitude spectogram $\mathbf{X}_{\text{Mag}}$ is defined as $y[n]$ with its feature space representation as $\mathbf{Y}_f$. An extra term is appended to the objective function of $\mathcal{D}$ that accounts for the prediction of the $\mathcal{Q}'$ scores of these 'de-enhanced' signals:

$$\begin{aligned}\mathcal{L}_{D,\text{MG+/-}} = [(\mathcal{D}(\mathbf{S}_f, \mathbf{S}_f) - 1)^2 + (\mathcal{D}(\hat{\mathbf{S}}_f, \mathbf{S}_f) - \mathcal{Q}'(\hat{s}, s))^2 \\ + (\mathcal{D}(\mathbf{X}_f, \mathbf{S}_f) - \mathcal{Q}'(x, s))^2 + (\mathcal{D}(\mathbf{Y}_f, \mathbf{S}_f) - \mathcal{Q}'(y, s))^2] \quad (3.1)\end{aligned}$$

where y represents the output of the de-generator network on the noisy signal $x[n]$. The objective function of $\mathcal{N}$ is given as

$$\mathcal{L}_{\mathcal{N},\text{MG+/-}} = [(\mathcal{D}(\mathbf{Y}_f, \mathbf{S}_f) - w)^2], \quad \text{for } 0 < w < 1, \quad (3.2)$$

where w is a hyper-parameter corresponding to the value of $\mathcal{Q}'$ that $\mathcal{N}$ is trained to output signals with. The loss function of $\mathcal{G}$ is the same as for MetricGAN+, as given in (2.57), thus the training of $\mathcal{N}$ is influenced entirely by its performance as assessed by $\mathcal{D}$, in the same manner as $\mathcal{G}$ using MetricGAN+. The training of $\mathcal{N}$ is the same as the training of $\mathcal{G}$ depicted in Figure 2.32 except that $\mathcal{G}$ is replaced by $\mathcal{N}$, $\hat{s}$, $\hat{\mathbf{S}}_f$ by y, $\mathbf{Y}_f$. An identical network structure to $\mathcal{G}$ is used for $\mathcal{N}$.

The training of MetricGAN+/- is similar to that that of MetricGAN+ described in Section 2.12.3 with slight differences. $\mathcal{D}$ is trained using (3.1); as a result the replay buffer now contains both enhanced and de-enhanced data, effectively doubling its size. After $\mathcal{D}$'s training, $\mathcal{N}$ is trained using (3.2). Then $\mathcal{G}$ is trained as usual using (2.57).

3.4 MetricGAN+/- Experiments

3.4.1 Experiment Setup

The aim of the following experiments is to compare the performance of the baseline system MetricGAN+ which is available as part of the SpeechBrain (Ravanelli et al., 2021) toolkit with our extension, MetricGAN+/-. The Adam optimiser (Kingma & Ba, 2014) with a learning rate of 0.0005 is used. The STFT is used with a Discrete Fourier Transform (DFT) length of L_{DFT} =512, a window length of 512 (32 ms) at sampling frequency of $f_s = 16$ kHz and a hop (overlap) length 256 (16 ms), resulting in a 50% overlap between frames. The minimum value in the time frequency masks $\mathbf{M}_{\mathcal{G}}$ and $\mathbf{M}_{\mathcal{N}}$ is set to $\xi = 0.05$.

Both PESQ and STOI as objective Q and different values of w are experimented with. The values of w are selected such that they correspond to sparely populated values of $\mathcal{Q}'$ in the dataset.Also, one experiment (denoted by * in Table 3.1) where the value of β in $\mathcal{N}$'s Learnable Sigmoid activation as given in (2.59) to also learned (in addition to α) is performed. Additionally, experiments reducing the size of the replay buffer training step for $\mathcal{D}$, via modifying H are performed. In order to ensure that the performance gain does not come entirely from the larger H in MetricGAN+/-, the baseline MetricGAN+ performance with H set to 0.4 is also reported.

3.4.2 Experiment Results

Table 3.1 shows the performance of MetricGAN+/- relative to the MetricGAN+ baseline and the unprocessed noisy audio on the VoiceBank-DEMAND testset. Performance is also compared with a second baseline system SEGAN (Pascual et al., 2017) (cf. Section 2.3.3), a state-of-the-art speech enhancement system. For more comparison baseline performances the interested reader is referred to Table 3 in (Fu, Yu, Hsieh, et al., 2021), which shows that MetricGAN+ with a PESQ objective outperforms all systems listed in terms of PESQ score.Performance is assessed using PESQ and STOI and also the Composite (Lin et al., 2019) Measure.

Model Name	Obj.	w	H	P	S	Csig	Cbak	Covl
Noisy	-	-	-	*1.97*	*92*	*3.35*	*2.44*	*2.63*
MG+ (P) (Fu, Yu, Hsieh, et al., 2021)	P	-	0.2	3.05	93	4.03	2.87	3.52
MG+ (S)	S	-	0.2	2.42	**93.4**	3.56	2.58	2.97
SEGAN (Pascual et al., 2017)	-	-	-	2.42	92.5	3.61	2.61	3.01
MG+	P	-	0.4	3.17	92.3	4.05	2.91	3.59
MG+/-	P	1.0	0.2	3.20	93.0	4.08	2.94	3.62
MG+/-	P	0.50	0.2	**3.22**	91.3	4.05	2.94	3.62
MG+/-	P	0.45	0.2	3.21	91.9	4.09	**2.95**	3.64
MG+/-*	P	0.45	0.2	3.17	93.0	**4.16**	2.93	**3.65**
MG+/-	P	0.45	0.1	3.13	92.1	4.05	2.91	3.58
MG+/-	P	0.30	0.2	3.04	93.0	4.07	2.88	3.55
MG+/-	S	0.45	0.1	2.13	93.2	3.04	2.42	2.56
MG+/-	S	0.30	0.2	2.31	93.3	3.19	2.49	2.72

Table 3.1: *Performance of MetricGAN+ (MG+) and MetricGAN+/- (MG+/-) on VoiceBank-DEMAND test set for objective PESQ (P) or STOI (S), * denotes the simulation where β is made learnable*

The first four rows in Table 3.1 present the results the un-enhanced noisy data and of different baselines. The results for the baseline MetricGAN+ models shown here are obtained using the implementation in SpeechBrain (Ravanelli et al., 2021). Further simulations are conducted for various values of hyperparameter w used in the training of $\mathcal{N}$. Table 3.1 shows a clear improvement in PESQ score for PESQ objective MetricGAN+/- models over the baseline MetricGAN+ (3.05 vs 3.22 PESQ), and also versus the PESQ value reported in (Fu, Yu, Hsieh, et al., 2021) of 3.15. An increase is also observed in the composite measure scores. Interestingly, there is an improvement even when $w = 1$, which is the case where $\mathcal{N}$ and $\mathcal{G}$ have the same objective, and thus $\mathcal{N}$ also learns to enhance. Hypothetically, this is due to slight variations in the outputs of $\mathcal{N}$ and $\mathcal{G}$ during training, as well as the increased replay buffer size compared to the baseline. Highest performance in terms of PESQ score is obtained with a w value set to 0.5, which means that $\mathcal{N}$ attempts to produce signals with a PESQ score of 3. Speculatively, this performance increase is due to there being few clean/noisy pairs in the training set with a PESQ score around this value.

By making the β parameter in $\mathcal{N}$'s activation function learnable, a slight improvement against the baseline is observed, as well as increased Csig and Covl scores versus all other simulations. It is found that increasing H in the baseline MetricGAN+ from 0.2 to 0.4 such that its size is comparable to MetricGAN+/-'s does slightly improve PESQ score. This is contrary to the findings in (Fu, Yu, Hsieh, et al., 2021) where the authors report no improvement for values larger than 0.2. However, larger values of H will drastically increase the training time requirement of the system. A better

understanding of what $\mathcal{D}$ leans from the replay buffer training and better curation of its contents is the key to further performance gains, as well as reduced training time required.

3.4.3 Spectogram Analysis

Figure 3.3 shows the spectrograms of the clean reference $\mathbf{S}_{\text{Mag}}$, noisy input $\mathbf{X}_{\text{Mag}}$, generator output mask $\mathbf{M}_{\mathcal{G}}$ and this mask applied $\hat{\mathbf{S}}_{\text{Mag}}$ for baseline MetricGAN+, MetricGAN+/- ($w = 0.45$) PESQ objective models. The mask in Figure 3.3 (e) attempts to remove low frequency signal content while boosting the area corresponding to the frequency curve of the fundamental speech frequencies. Furthermore, the baseline MetricGAN+ PESQ model in Figure 3.3 (c, d) attenuates the signal in the initial non speech region, while the MetricGAN+/- model in Figure 3.3 (e, f) suppresses less energy around 400 Hz over the whole utterance. This artefact can already be observed in the baseline MetricGAN+ but is more prominent for the proposed method, which could explain the relatively low Cbak score for this method. This is potentially a results of $\mathcal{D}$ not learning to properly penalise errors in this region, perhaps due to the additional influence of $\mathcal{N}$'s outputs on it's training.

3.4.4 Validation Performance

Figure 3.4 shows PESQ score performance on the validation set during training for PESQ objective MetricGAN+, MetricGAN+ with H set to 0.4 and MetricGAN+/- ($w = 0.5$). Both models that include larger replay buffers perform significantly better and their score increases at a higher rate in early epochs. This suggests that $\mathcal{D}$'s performance (and consequently $\mathcal{G}$ as shown in the graph) is improved by an increased size of the replay buffer, and further improved by access the data produced by $\mathcal{N}$.

3.4.5 Generalisation To Unseen Data

Model Type	PESQ	STOI	Csig	Cbak	Covl
Noisy	*1.37*	*44.0*	***2.96***	*1.42*	*2.09*
MG+ PESQ	1.54	45.8	2.67	2.09	2.00
MG+ STOI	1.24	44.7	2.45	1.84	1.76
MG+/- PESQ	**1.76**	44.3	2.86	**2.03**	**2.20**
MG+/- STOI	1.22	**45.3**	2.31	1.81	1.67

Table 3.2: *Performance on real component of CHiME3 test set*

Model Type	PESQ	STOI	Csig	Cbak	Covl
Noisy	*1.27*	*87.0*	*2.61*	*1.39*	*1.88*
MG+ PESQ	2.14	87.4	3.05	2.31	2.53
MG+ STOI	1.52	**88.9**	2.75	2.07	2.08
MG+/- PESQ	**2.38**	86.1	**3.17**	**2.41**	**2.70**
MG+/- STOI	1.47	88.5	2.62	2.02	1.99

Table 3.3: *Performance on simulated component of CHiME3 test set*

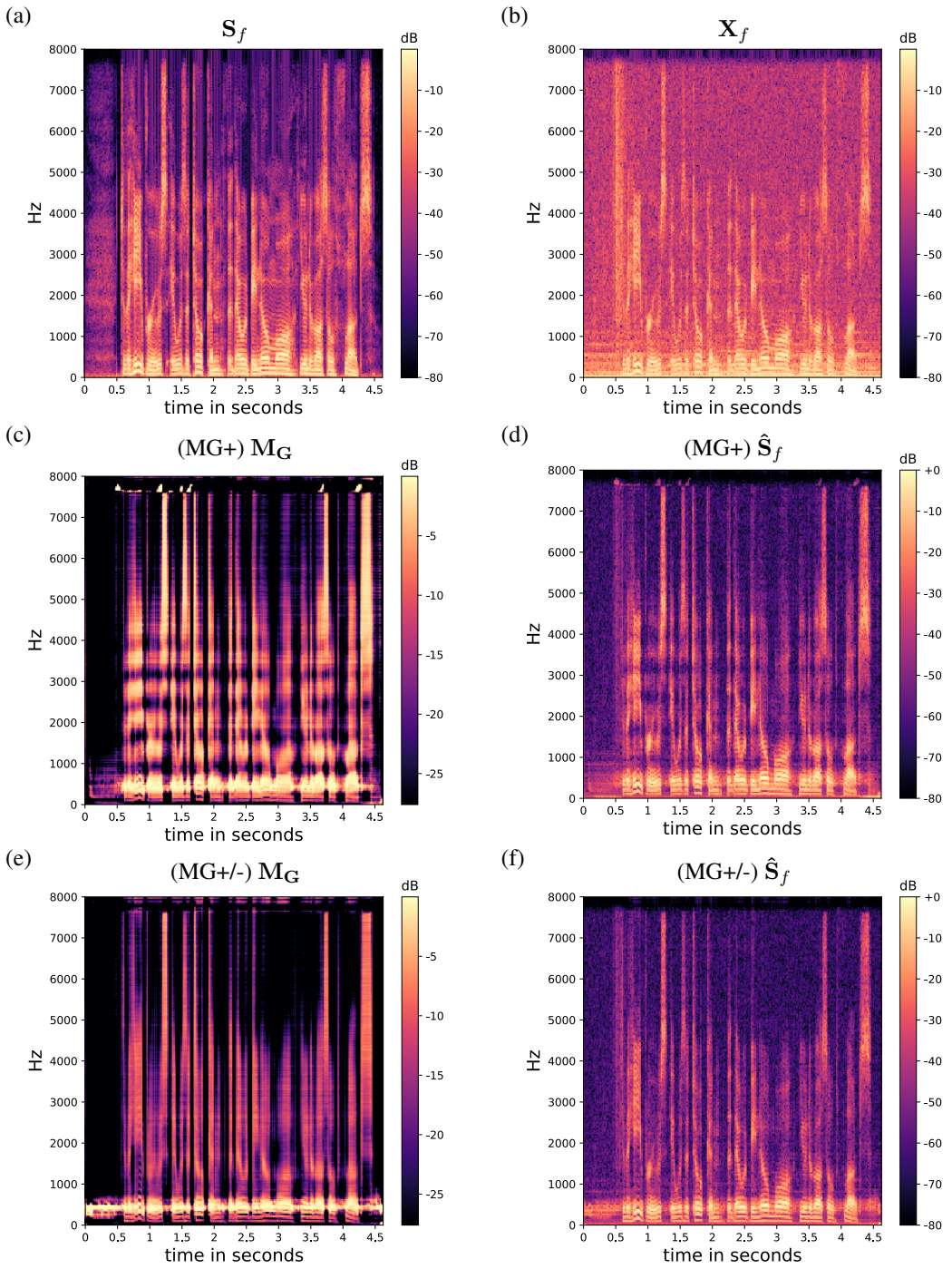

Figure 3.3: *Spectograms of: (a) clean reference features $\mathbf{S}_f$, (b) noisy features $\mathbf{X}_f$, (c) Mask $\mathbf{M}_\mathcal{G}$ and (d) enhanced output $\hat{S}_f$ for MetricGAN+ baseline PESQ objective model, (e) Mask $\mathbf{M}_\mathcal{G}$ and (f) enhanced output $\hat{S}_f$ for MetricGAN+/- PESQ objective model. Source audio file is* `p232_014.wav` *of VoiceBank-DEMAND testset.*

Table 3.2 and Table 3.3 shows the performance of the baseline MetricGAN+ and the best performing proposed MetricGAN+/- systems on the CHiME3 (c.f Section 2.9.2) test set. An increased performance in terms of PESQ, Csig, Cbak and Covl between PESQ objective MetricGAN+/- and

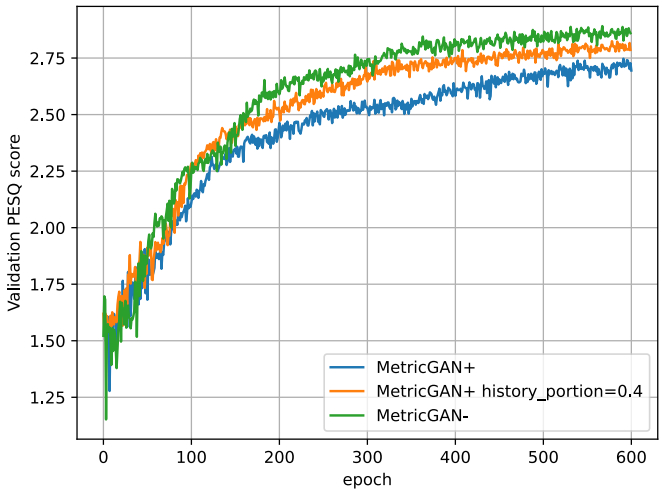

Figure 3.4: *Graph showing PESQ score on validation set during training for PESQ objective MetricGAN+ and MetricGAN+/- models*

the baseline, as well as a slight improvement in STOI score for STOI objective MetricGAN+/- is shown. This suggests that $\mathcal{D}$'s access to the de-generated signals produced by $\mathcal{N}$ allows $\mathcal{G}$ in MetricGAN+/- systems to generalise better to unseen environments.

3.5 Summary

In this chapter an extension to the MetricGAN+ baseline framework is proposed, which improves its performance in terms of PESQ score and related measures, as well as improving its generalisation to unseen data. It is found that training the discriminator network on a wider range of metric scores and with a larger replay buffer achieves greater performance than the baseline system.

Chapter 4

Further MetricGAN Variations

4.1 Introduction

In this chapter, experiments involving further variations of the MetricGAN+ (cf. Section 2.12) and it's previously proposed extension MetricGAN+/- (cf. Chapter 3) are carried out. The effect of the target metric value w used in the training of the de-generator $\mathcal{N}$ is investigated. Additionally, the CMGAN (Cao et al., 2022) (cf. Section 2.13) NNSE DNN structure is incorporated into the MetricGAN+/- framework.

4.2 System Overview

This section gives an overview of the three main components of the MetricGAN+/- framework; the NNSE Generator $\mathcal{G}$, the metric prediction discriminator $\mathcal{D}$ and the de-generator $\mathcal{N}$.

4.2.1 SE Generator

4.2.1.1 CMGAN $\mathcal{G}$ Loss

Introduced in (Cao et al., 2022), two additional losses can be used to train $\mathcal{G}$ alongside inference of $\mathcal{D}$ in (2.57). These loss terms help to further improve enhancement performance on the basis of a distance from clean reference representations, while the GAN loss helps shape the enhancement towards producing outputs of high metric scores. Firstly a time domain loss L_{time} (Abdulatif et al., 2021) which directly compares the enhanced time domain signal $\hat{s}$ with the clean reference signal

		Generator $\mathcal{G}$				Discriminator $\mathcal{D}$		De-Generator $\mathcal{N}$
Framework	**Training**	**Name**	**Features**	**Strucuture**	**Loss**	**Name**	**Input**	**Used**
MG (Fu et al., 2019)	Sampled	$\mathcal{G}_{\text{MG}}$	$\mathbf{X}_{\text{Mag}}$	BLSTM	((2.57))	$\mathcal{D}_{\text{MG}}$	s,$\hat{s}$	
MG+ (Fu, Yu, Hsieh, et al., 2021)	Sampled + Hist	$\mathcal{G}_{\text{MG+}}$	$\mathbf{X}_{\text{Mag}}$	BLSTM	((2.57))	$\mathcal{D}_{\text{MG}}$	s,$\hat{s}$,x	
MG+/- (Chapter 3)	Sampled + Hist	$\mathcal{G}_{\text{MG+}}$	$\mathbf{X}_{\text{Mag}}$	BLSTM	((2.57))	$\mathcal{D}_{\text{MG}}$	s,$\hat{s}$,x,y	✓
CMGAN (Cao et al., 2022)	Simple	$\mathcal{G}_{\text{CMGAN}}$	$\mathbf{X}_{\text{Mag}}$,$\mathbf{X}_{\text{Im}}$,$\mathbf{X}_{\text{Re}}$	Conformer	((4.3))	$\mathcal{D}_{\text{CMGAN}}$	s,$\hat{s}$	
CMGAN+ α (prop)	Simple	$\mathcal{G}_{\text{CMGAN}}$	$\mathbf{X}_{\text{Mag}}$,$\mathbf{X}_{\text{Im}}$,$\mathbf{X}_{\text{Re}}$	Conformer	((4.3))	$\mathcal{D}_{\text{CMGAN}}$	s,$\hat{s}$,x	
CMGAN+/- α (prop)	Simple	$\mathcal{G}_{\text{CMGAN}}$	$\mathbf{X}_{\text{Mag}}$,$\mathbf{X}_{\text{Im}}$,$\mathbf{X}_{\text{Re}}$	Conformer	((4.3))	$\mathcal{D}_{\text{CMGAN}}$	s,$\hat{s}$,x,y	✓
CMGAN+/- β (prop)	Sampled+ Hist	$\mathcal{G}_{\text{CMGAN}}$	$\mathbf{X}_{\text{Mag}}$,$\mathbf{X}_{\text{Im}}$,$\mathbf{X}_{\text{Re}}$	Conformer	((4.3))	$\mathcal{D}_{\text{MG}}$	s,$\hat{s}$,x,y	✓

Table 4.1: *Comparison between MetricGAN (MG) derived frameworks*

s,(2.38). Secondly, a time-frequency (TF) domain loss (Braun & Tashev, 2020), which makes explicit use of the component outputs of $\mathcal{G}_{\text{CMGAN}}$: $\hat{\mathbf{S}}_{\text{Mag}}, \hat{\mathbf{S}}_{\text{Im}}$ and $\hat{\mathbf{S}}_{\text{Re}}$. The distance between the enhanced and the reference magnitude is computed byL_{Mag}, (2.39). The loss for the real and imaginary components L_{RI} is defined similarly:

$$L_{\text{RI}} = \frac{1}{T \cdot F_{\text{Hz}}} \sum_{t}^{T} \sum_{f_{\text{Hz}}}^{F_{\text{Hz}}} (\mathbf{S}_{\text{Ri}}[t, f_{\text{Hz}}] - \hat{\mathbf{S}}_{\text{Ri}}[t, f_{\text{Hz}}])^2 + \frac{1}{T \cdot F_{\text{Hz}}} \sum_{t}^{T} \sum_{f_{\text{Hz}}}^{F_{\text{Hz}}} (\mathbf{S}_{\text{Im}}[t, f_{\text{Hz}}] - \hat{\mathbf{S}}_{\text{Im}}[t, f_{\text{Hz}}])^2 \tag{4.1}$$

These two terms are combined with a hyperparameter weighing α to result in the time frequency loss L_{TF}

$$L_{TF} = \alpha L_{\text{Mag}} + (1 - \alpha) L_{\text{RI}}. \tag{4.2}$$

The final loss for $\mathcal{G}$ under CMGAN is then given as in (Cao et al., 2022)

$$L_{\mathcal{G}} = \gamma_1 L_{\text{GAN}} + \gamma_2 L_{\text{Time}} + \gamma_3 L_{\text{TF}} \tag{4.3}$$

where $\gamma_1, \gamma_2, \gamma_3$ are hyperparameter weights to control the influence of each loss term.

4.2.1.2 A Note on the Generator Structures

The network structures of the speech enhancement generators were selected to investigate the effect of the explicit consideration of complex valued components (and thus phase information) in their inputs and outputs; the baseline SE DNN MetricGAN+ $\mathcal{G}$ (cf. Section 2.12) operates only over magnitude spectrograms, which do not consider the phase; in this case the phase information of the noisy speech $\mathbf{X}_P$ is used in re-synthesis. To contrast this the CMGAN $\mathcal{G}$(cf. Section 2.13) takes as input the component $\mathbf{X}_{\text{Re}}$ and $\mathbf{X}_{\text{Re}}$ outputs of the STFT.

4.2.2 Metric Prediction Discriminator

4.2.2.1 CMGAN Discriminator Network Structure

The CMGAN Discriminator $\mathcal{D}_{\text{CMGAN}}$ is used in this chapter. It is structured similarly to that of MetricGAN+ (cf. Section 2.12.4) called here $\mathcal{D}_{\text{MG+}}$) with some differences. Firstly each convolutional layer have a PreLU (rather than ReLU) activation and have $16, 32, 64, 128$ output filters respectively. There are two linear layers rather than three following the average pooling with 64 and 1 output neurons respectively. Finally, the first of these linear layers has a PreLU activation, and the 'learnable' Sigmoid used originally in the MetricGAN+/- $\mathcal{G}$ is repurposed as the activation on the final layer.

4.2.3 Degenerator

4.2.3.1 Network Stucture

In most experiments in this chapter, the Degenerator $\mathcal{N}$ is structured identically to the MetricGAN+ Generator (cf. Section 2.12.5) as in Chapter 3. Additionally, experiments are carried out where the CMGAN Generator structure (cf. Section 2.13) is used for $\mathcal{N}$.

4.2.3.2 Degenerator Input

An experiment where the input to $\mathcal{N}$ is the clean reference magnitude $\mathbf{S}_{\text{Mag}}$ is carried out. It is trained using a modified version of (3.2):

$$L_{\mathcal{N}_{\text{Clean}}} = \mathbb{E}\{(\mathcal{D}(\bar{\mathbf{S}}_{\text{Mag}}, \mathbf{S}_{\text{Mag}}) - w)^2\} \tag{4.4}$$

where $\bar{\mathbf{S}}_{\text{Mag}}$ is the magnitude of the de-enhanced time domain signal $\bar{s}[n]$.

4.2.4 Training Details

4.2.4.1 Historical Training of Discriminator $\mathcal{D}$

First proposed in (Fu, Yu, Hsieh, et al., 2021), this is $\mathcal{D}$ being trained using a 'replay buffer' where saved outputs of the generator $\mathcal{G}$ and degenerator $\mathcal{N}$ from past epochs are used to train $\mathcal{D}$. The goal of this is to prevent $\mathcal{D}$ from *forgetting* about it's assessment of the outputs of past epochs of $\mathcal{G}$'s training. The size of this replay buffer is decided by a '`history_portion`' H hyper-parameter, which corresponds to a percentage sampling of these saved outputs each epoch. For example in a MetricGAN+/- system, after the first epoch of $\mathcal{D}$'s training the replay buffer contains 100 outputs of $\mathcal{G}$ and 100 outputs of $\mathcal{N}$; with `history_portion = 0.2` 20% of each buffer's samples i.e 40 samples will be used to train $\mathcal{D}$ as it's historical training.
Due to the growth of this buffer, training of MetricGAN+/- systems tends to slow down at later epochs, where most of the time spent each epoch is consumed by the historical training. In this chapter, a number of a number of techniques are proposed to limit the size of this replay buffer:

- **Cutoff** - saved outputs which are older than O_{cutoff} epochs relative to the current epoch are removed from the historical set, where O_{cutoff} is a hyperparameter.
- **Disable** - the historical training is disabled once a certain training epoch E is reached where E is a hyperparameter.
- **Random** - the saved outputs have a $O_{\text{random}}\%$ chance of being removed from the historical set, where O_{random} is a hyperparameter.
- **Flatten** - the saved outputs are retained in the historical set only if the associated $Q'(\cdot)$ falls within one standard deviation of the mean $Q'(\cdot)$ score of the entire historical set, i.e if:

$$\mathbb{E}\left(\sum_{h=0}^{H} Q'(\cdot)\right) - \sigma_{Q'(\cdot)} \leq Q'(\cdot) \leq \mathbb{E}\left(\sum_{h=0}^{H} Q'(\cdot)\right) + \sigma_{Q'(\cdot)} \tag{4.5}$$

  is true for the given sample. This flattening starts once a epoch E is reached, where E is a hyperparameter value.

The goal of all of these techniques is to reduce the overhead in terms of time and compute of training MetricGAN+/- systems, while avoiding a significant drop in performance.

4.2.4.2 Sampled β Training

In the Sampled (β) training scheme, the frameworks are trained using a sampled approach, with a replay buffer used to train $\mathcal{D}$ as detailed in Section 2.12.3

4.2.4.3 Simple α Training

In the Simple (α) training a more traditional training scheme is used, where in each epoch the model(s) observers the entire training set. Each training point is segmented into 2 second clips. For each batch, first the Generator $\mathcal{G}_{\text{CMGAN}}$ is trained as described in Section 4.2.1.1 followed by $\mathcal{N}$ (if applicable) and finally $\mathcal{D}$.
In this chapter, the same validation set is used as in the Sampled training, with the best performing epoch in terms of PESQ score on this validation set loaded at test time.

4.3 Experiments

4.3.1 Datasets Used

4.3.1.1 VoiceBank-DEMAND

The VoiceBank-DEMAND (Valentini-Botinhao et al., 2016) is used in this chapter, for details see Section 2.9.1.

4.3.1.2 VoiceBank-DEMAND-Rerecorded

In this chapter a new variation on the VoiceBank-DEMAND dataset is introduced, which is called 'VoiceBank-DEMAND-Rerecorded'. There are two main reasons why this dataset was created. Firstly, the testset of the original VoiceBank-DEMAND dataset contains mixtures with relatively high SNR values (higher overall than those of the training set). This makes the testset too 'easy' for the enhancement systems being trained. Secondly, the original dataset is an artificial mixture of clean speech and noise with no simulation of recording environment such as reverberation. This is unlike real spaces where such factors can greatly effect the recording of speech.
'VoiceBank-DEMAND-Rerecorded' is a recreation of the original dataset but rather than the noise being scaled to the desired SNR and then mixed with the clean speech, the noise is scaled and then played aloud alongside the clean speech from two loudspeakers, with the mixture of the two signals being captured by a 16 channel microphone array. The pairings of a each speech file with a noise recording at specific SNRs are the same as that of the original dataset, however the exact segment of the noise file is not the same as this was randomly selected in the original simulation. The two loudspeakers and the microphone array were positioned 1 meter equidistant from each other on the floor of a soundproofed room. The dimensions of the room are 7 by 9.2 by 2.8 meters and the recording setup was positioned in a roughly 1 meters from a corner. Fig Figure 4.1 depicts the recording setup; Spk_s and Spk_v represent the loudspeakers which play aloud the clean speech s and noise v respectively, while $\text{m}_1, \text{m}_2, ..., \text{m}_{16}$ represent the recording microphones. In this work, the recordings from the microphone nearest to the loudspeakers m_1 are used as $x[n]$. Due to some failure in the recording process, the testset portion of VoiceBank-DEMAND-Rerecorded is slightly smaller than that of the original, containing 757 utterances. Figure 4.2 shows the correlations in the PESQ and STOI metric values between the original and rerecorded training sets; from this it can be observed that the rerecorded training set has lower metric scores overall, and so is more difficult to enhance.

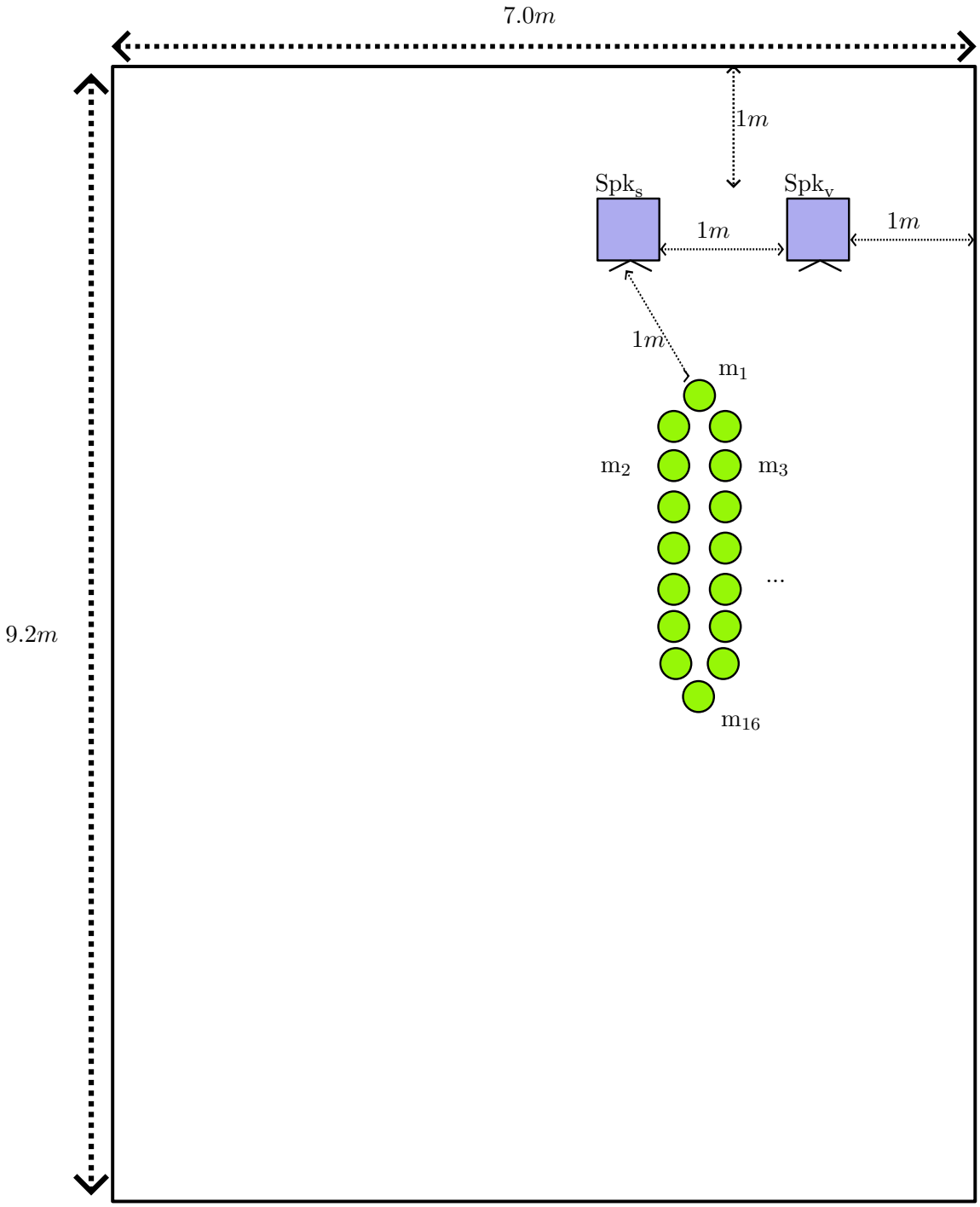

Figure 4.1: *Diagram of VoiceBank-DEMAND-Rerecorded Recording Environment*

4.3.2 Experiment 1: Investigating the Effect of Hyperparameter w

In Chapter 3 when the Degenerator $\mathcal{N}$ concept is introduced, only a single value for the hyperparameter w is used. This value controls the target metric score of which $\mathcal{N}$ aims to produce audio with, via $\mathcal{N}$'s loss function either ((3.2)) or ((4.4)). This experiment aims to better understand the effect of this hyperparameter, and its relationship with the distribution of target metrics scores

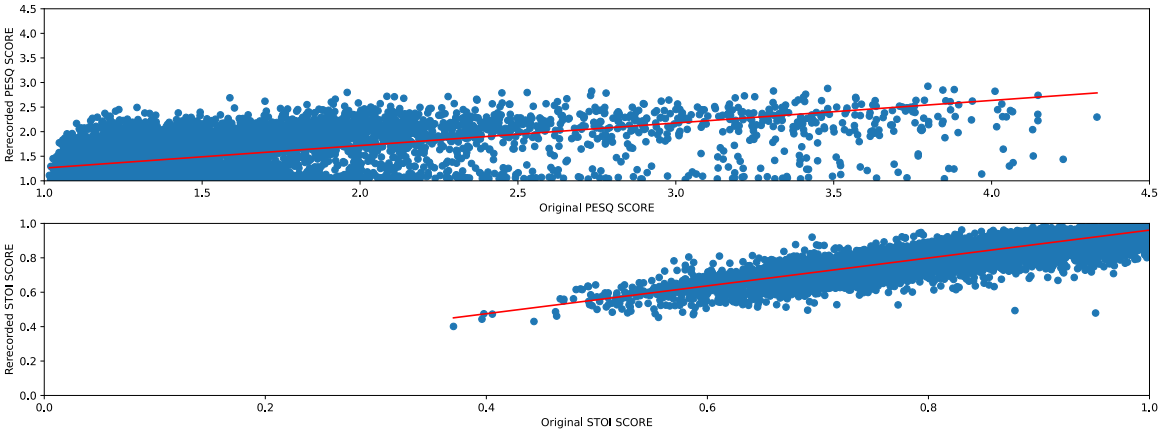

Figure 4.2: *PESQ and STOI metric correlation between original VoiceBank-DEMAND and VoiceBank-DEMAND-Rerecorded*

in the training data.

4.3.2.1 Experiment 1 Setup

Several MetricGAN+/- models were trained with both noisy audio x and clean audio s as inputs to Dengenerator network $\mathcal{N}$ using ((3.2)) and ((4.4)) respectively. Values of the hyperparameter w in the range of 0.1 to 1 were used, in intervals of 0.05. This represents $\mathcal{N}$ being trained to produce audio with a PESQ score defined as:

$$O_{\mathcal{N}} = (w \cdot 3.5) + 1 \tag{4.6}$$

resulting in a range of PESQ objectives from 1.35 to 4.5. Note that in the case that $w = 1$, $\mathcal{N}$ is being trained to the same objective as $\mathcal{G}$. The other hyperparameter values are the same as those in the experiments in (Close, Hain, et al., 2022); the models were trained for 600 epochs with the Adam (Kingma & Ba, 2014) optimiser a learning rate of 0.0005 for all three networks. Each model is evaluated using the STOI (Taal et al., 2011), PESQ (Rix et al., 2001), Composite (Lin et al., 2019) and SI-SDR (Roux et al., 2018) metrics.

4.3.2.2 Experiment 1 Results: VoiceBank-DEMAND

Table 4.2 and Table 4.3 show the results for using x and y as input to $\mathcal{N}$ respectively on the orignal VoiceBank-DEMAND dataset. In Table 4.2 the best performing model in terms of PESQ and the Csig and Covl components of the Composite Measure has a w value of 0.8 ($O_{\mathcal{N}} = 3.8$). This beats the best PESQ score reported in Chapter 3 of 3.22. From Table 4.3 the best performing MetricGAN+/- system in terms of PESQ score has a value of 0.65 ($O_{\mathcal{N}} = 3.275$). However, the best performing model in terms of all three components of the Composite measure has a w value of 0.15 ($O_{\mathcal{N}} = 1.525$). Overall, the frameworks which used s as input to $\mathcal{N}$ did not perform as well as those which use x; this is likely due to the difficulty of the task of reducing the PESQ score of clean speech versus the task of reducing the PESQ score of already noisy speech.

4.3.2.3 Experiment 1 Results: VoiceBank-DEMAND-Rerecorded

Table 4.4 and Table 4.5 show the results for using x and y as input to $\mathcal{N}$ respectively on the rerecorded VoiceBank-DEMAND dataset. In Table 4.4, the best performing model in terms of PESQ was that where the value of w is 0.9 ($\mathrm{O}_{\mathcal{N}} = 4.15$). Table 4.5 shows that the best performing model in terms of PESQ score where y is input to $\mathcal{N}$ has a w value of 0.65 ($\mathrm{O}_{\mathcal{N}} = 3.275$). In both, the SI-SDR metric is not improved by the enhancement; this is likely due to the effect of the reverberation caused by the recording environment.

4.3.2.4 Experiment 1 Analysis

Figure 4.4 and Figure 4.5 show the distribution of PESQ scores for the original and rerecorded VoiceBank-DEMAND training sets respectively. Both also display the optimal $\mathrm{O}_{\mathcal{N}}$ which corresponds to the best performing value of w for when both x and s are the input to $\mathcal{N}$. It can be observed from these figures that the best value for w is one which results in $\mathcal{N}$ being trained to produce outputs with PESQ scores which are not well represented in the training set. All four optimal $\mathrm{O}_{\mathcal{N}}$ values are high, meaning that $\mathcal{N}$ is trained to produce relatively high quality outputs. For both the original and rerecorded datasets, the optimal value for w is lower when $s[n]$ is the input to $\mathcal{N}$ than when the input is $x[n]$.

4.3.2.5 Experiment 1 Degenerator Performance

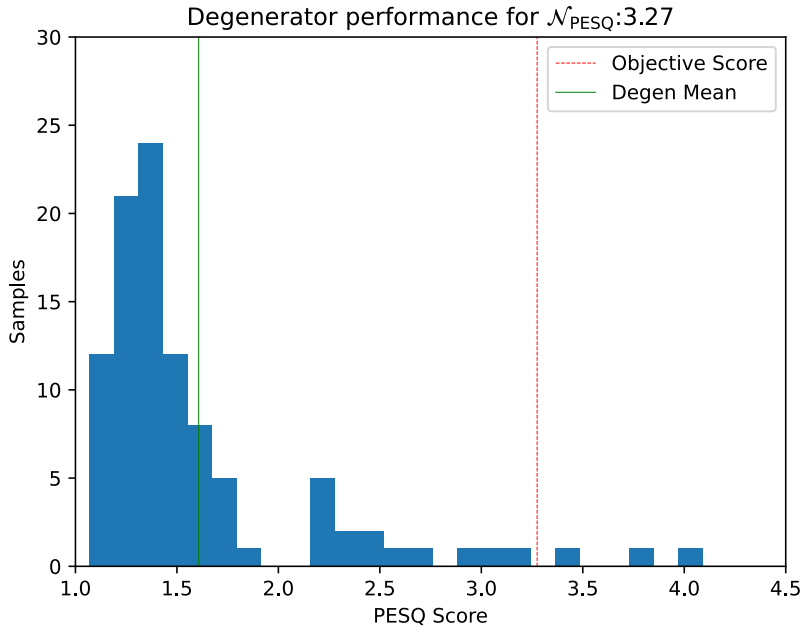

Figure 4.3: *Distribution of PESQ scores of $\mathcal{N}$'s outputs at final training epoch with $s[n]$ as input and $w = 0.65$ for training on the original VoiceBank-DEMAND.*

Figure 4.6 shows the distribution of PESQ scores for the output of the degenerator $\mathcal{N}$ in the final epoch of training for the best performing MetricGAN+/- framework in Table 4.2 i.e $w = 0.8$ with

nosiy signal x as input. The dotted red line represents the target score for which $\mathcal{N}$ is being trained to produce outputs with while the solid green line represents the mean score of the actual outputs in this epoch. This figure indicates that $\mathcal{N}$ is unable to archive it's training objective even at the last training epoch of the framework. A similar result can be observed for the best performing model with s as input in Figure Figure 4.3.

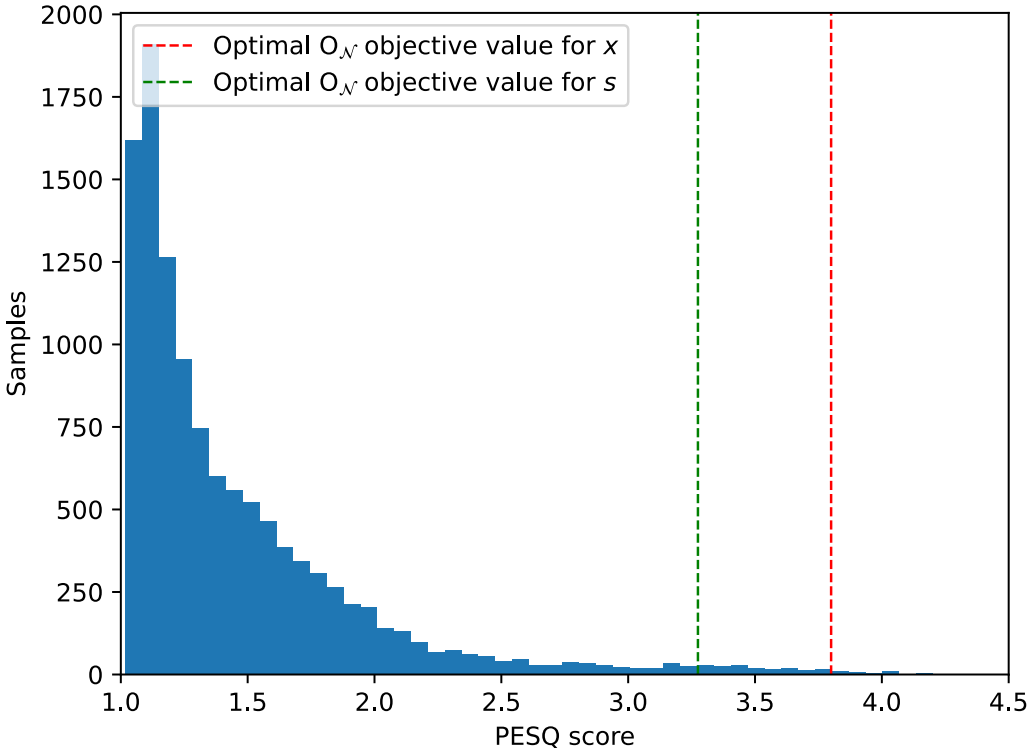

Figure 4.4: *Distribution of PESQ scores in original VoiceBank-DEMAND training set*

w	**PESQ**	**STOI**	**Csig**	**Cbak**	**Covl**	**SI-SDR**
Noisy	*1.97*	*0.92*	*3.35*	*2.44*	*2.63*	*8.98*
0.1	3.01	0.93	4.06	3.06	3.52	**10.32**
0.2	3.06	0.93	4.07	3.10	3.57	7.98
0.3	3.06	0.93	4.05	3.05	3.54	7.23
0.4	3.17	0.92	3.95	3.06	3.53	5.99
0.5	3.19	0.92	4.00	**3.13**	3.58	7.09
0.6	3.20	0.92	4.04	3.08	3.60	5.39
0.7	3.18	0.92	3.98	3.07	3.55	6.41
0.8	**3.25**	**0.93**	**4.18**	3.12	**3.70**	6.21
0.9	3.13	0.93	4.09	3.08	3.6	7.92
1.0	3.14	0.92	4.07	3.07	3.59	7.45

Table 4.2: *Performance of MetricGAN+/- with x as input and w values for $\mathcal{N}$ on the original VoiceBank-DEMAND dataset.*

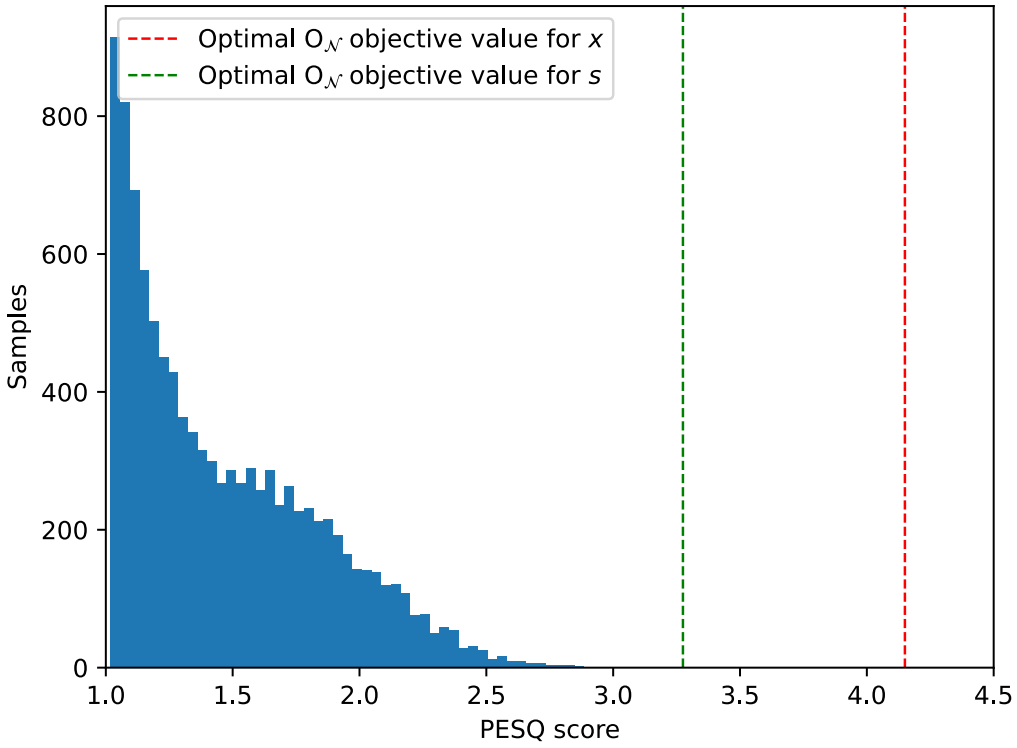

Figure 4.5: *Distribution of PESQ scores in Rerecorded VoiceBank-DEMAND training set*

w	**PESQ**	**STOI**	**Csig**	**Cbak**	**Covl**	**SI-SDR**
Noisy	*1.97*	*0.92*	*3.35*	*2.44*	*2.63*	*8.98*
0.1	3.12	0.92	4.06	3.10	3.58	6.96
0.2	2.98	0.93	3.94	3.10	3.45	10.22
0.3	3.12	0.92	3.94	3.10	3.50	7.99
0.4	3.10	0.92	4.03	3.12	3.55	7.79
0.5	3.12	0.92	4.06	3.12	3.58	7.83
0.6	3.03	0.92	4.03	3.06	3.51	7.19
0.7	3.01	0.92	3.99	3.12	3.53	6.60
0.8	3.01	0.92	3.99	3.07	3.52	6.48
0.9	3.11	0.92	3.94	3.06	3.50	5.59
1.0	3.07	0.92	3.99	3.04	3.51	7.10

Table 4.3: *Performance of MetricGAN+/- with s as input and w values for $\mathcal{N}$ on the original VoiceBank-DEMAND dataset.*

4.3.3 Experiment 2: Historical Set Reduction Techniques

In this experiement, the historical set reduction techniques introduced in Section 4.2.4.1 are implemented and compared.

w	**PESQ**	**STOI**	**Csig**	**Cbak**	**Covl**	**SI-SDR**
Noisy	*1.67*	*0.88*	*2.81*	*2.04*	*2.18*	***1.08***
0.1	2.64	**0.90**	**3.50**	2.58	3.03	-0.27
0.2	2.36	0.84	2.76	2.22	2.44	-5.13
0.3	2.71	0.88	3.37	2.60	2.99	-0.61
0.4	2.74	0.89	3.45	2.61	3.04	-0.32
0.5	2.72	0.88	3.32	2.59	2.97	-1.01
0.6	2.73	0.87	3.20	2.54	2.89	-1.23
0.7	1.45	0.75	2.23	1.72	1.68	-7.89
0.8	2.69	0.89	2.93	2.56	2.75	-0.99
0.9	**2.92**	0.88	3.30	**2.64**	**3.04**	0.05
1.0	2.62	0.88	2.90	2.55	2.71	-1.59

Table 4.4: *Performance of MetricGAN+/- with x as input and w values for $\mathcal{N}$ on the rerecorded VoiceBank-DEMAND dataset.*

w	**PESQ**	**STOI**	**Csig**	**Cbak**	**Covl**	**SI-SDR**
Noisy	*1.67*	*0.88*	*2.81*	*2.04*	*2.18*	***1.08***
0.1	2.63	0.89	2.85	2.53	2.68	-0.53
0.2	**2.71**	0.88	**3.26**	**2.59**	**2.93**	-0.99
0.3	2.55	0.88	3.08	2.53	2.77	-1.28
0.4	2.64	0.89	3.05	2.56	2.79	-1.13
0.5	2.64	0.89	3.21	2.56	2.87	-0.26
0.6	2.58	0.89	3.27	2.55	2.88	-0.77
0.7	2.45	**0.90**	3.24	2.50	2.81	-0.66
0.8	2.56	0.89	2.78	2.52	2.62	-0.82
0.9	2.53	0.88	2.90	2.50	2.66	-1.55
1.0	1.89	0.86	1.63	2.16	1.66	-2.65

Table 4.5: *Performance of MetricGAN+/- with s as input and w values for $\mathcal{N}$ on the rerecorded VoiceBank-DEMAND dataset.*

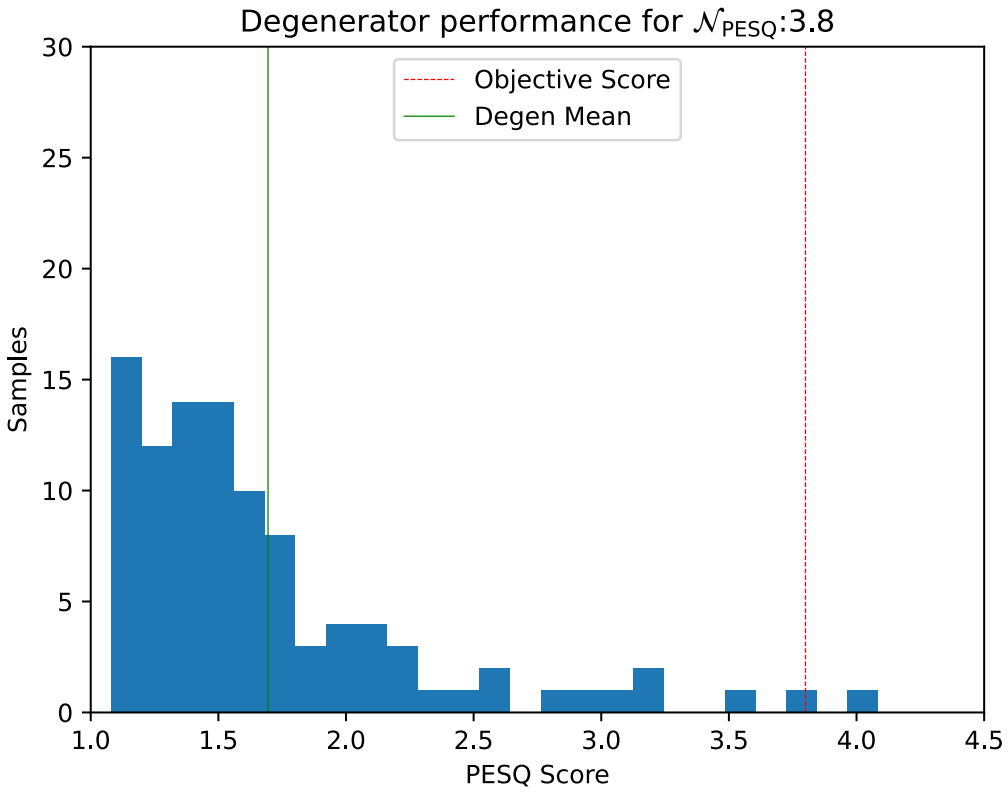

Figure 4.6: *Distribution of PESQ scores of $\mathcal{N}$'s outputs at final training epoch with $x[n]$ as input and $w = 0.8$ for training on the original VoiceBank-DEMAND.*

4.3.3.1 Experiment Setup

Several MetricGAN+/- models were trained on the original VoiceBank-DEMAND dataset with the techniques detailed in Section 4.2.4.1. The best performing hyperparameter values from Table 4.2 were selected, i.e x as input to $\mathcal{N}$ and w at 0.65 for all the models. O_{cutoff} for the Cutoff technique is set to 10 and epoch E for both the Disable and Flatten technique is set to 300, i.e the halfway point in training. O_{random} is set to 50 for the Random technique. For all frameworks, H is set to 0.2.

4.3.3.2 Performance of Techniques

Table 4.6 shows the results for the different historical set reduction techniques. Along with the signal quality metrics, the table also shows the training time for each model in hours t as well as the trade off between gain in PESQ score over the noisy data and training time, Δ_{PESQ}/t. The model which took the shortest time to train was that which used the Cutoff technique, which achieved performance somewhat comparable to the baseline while taking a significantly shorter time to train. All of the historical set reduction techniques were able to achieve higher Δ_{PESQ}/t scores than the baseline, which suggests that the large history buffer has only a small effect on the overall performance of the framework. Figure 4.7 visualises the training time and Figure 4.8 visualises the

relative gain in validation time PESQ score. Since the cutoff technique showed the best results, a further experiment was carried out, where frameworks using the Cutoff technique with differing values for O_{cutoff} were trained; Table 4.7 shows the results. A pattern emerges in these results wherein a higher values of O_{cutoff} produce better performing enhancement models at the cost of increased training time. All Cutoff frameworks have similar Δ_{PESQ}/t scores, which suggests that the relative gain in PESQ score between the frameworks is consistent. Near baseline performance is achieved by all of the frameworks here.

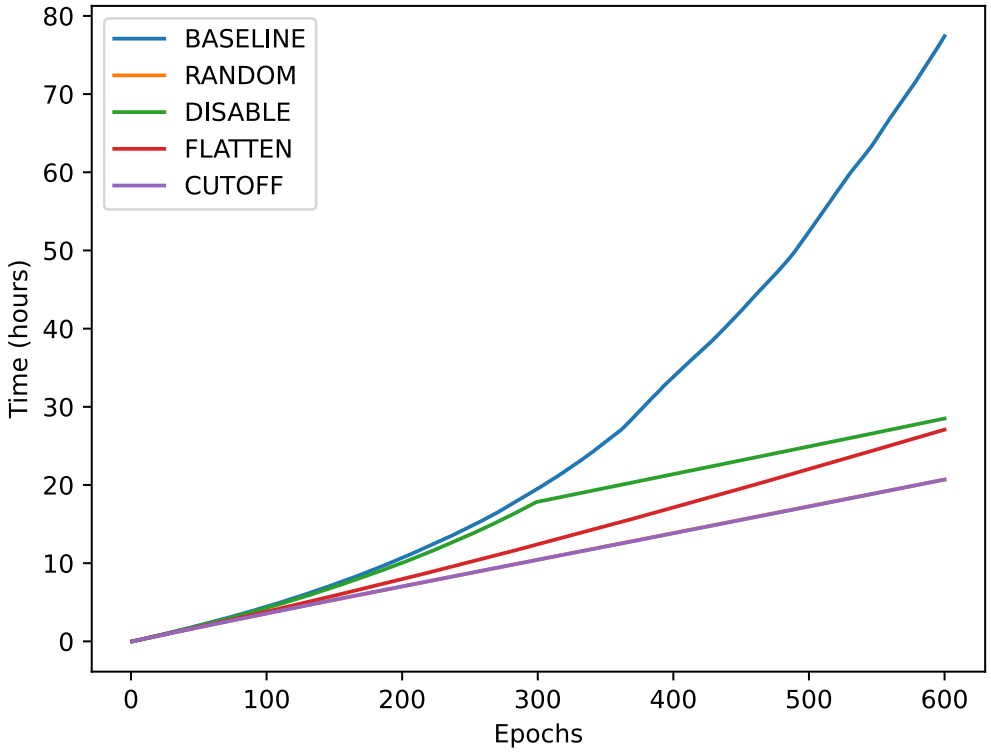

Figure 4.7: *Training time versus epoch counter for the historical set reduction techniques.*

Technique	**Train Hours** t	**PESQ**	$\boldsymbol{\Delta}_{\text{PESQ}}/\mathbf{t}$	**STOI**	**Csig**	**Cbak**	**Covl**	**SI-SDR**
Noisy	–	*1.97*	–	*0.92*	*3.35*	*2.44*	*2.63*	*8.98*
Baseline	77.44	**3.25**	0.017	**0.93**	**4.18**	**3.12**	**3.70**	6.21
Cutoff	**20.70**	3.05	**0.052**	0.92	4.01	2.99	3.52	5.70
Disable	28.56	3.05	0.038	0.93	3.99	3.12	3.51	8.79
Random	20.75	2.87	0.043	0.94	4.06	2.99	3.47	**15.94**
Flatten	27.13	3.02	0.039	0.93	4.05	3.04	3.53	7.58

Table 4.6: *Performance of MetricGAN+/- with historical set size reduction techniques on original VoiceBank-DEMAND*

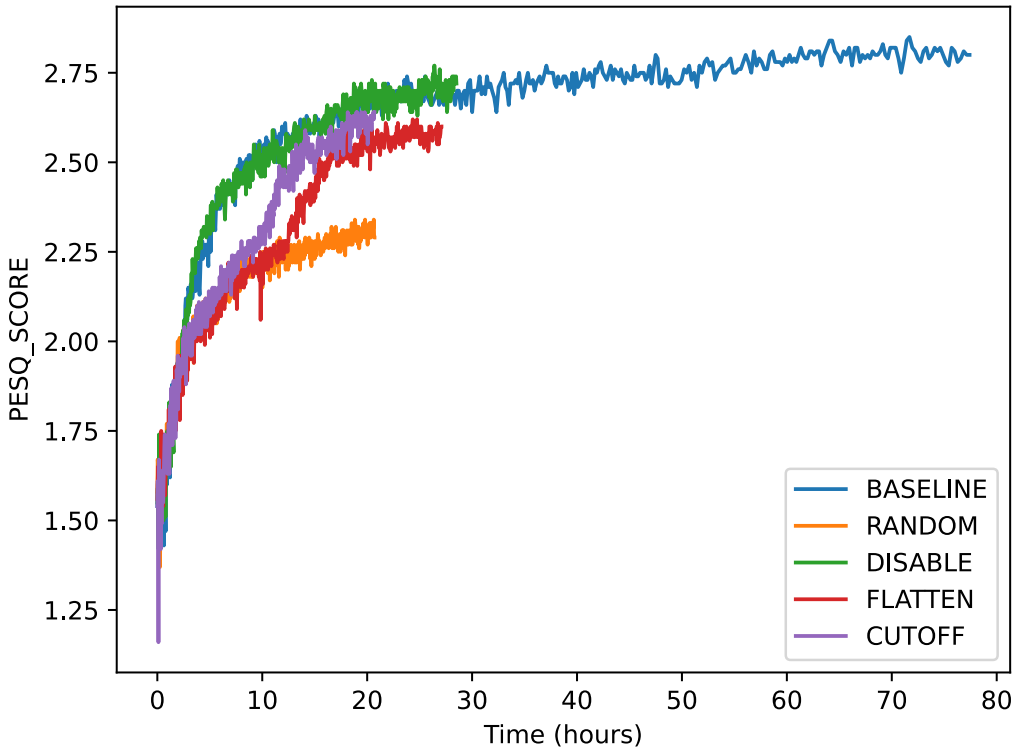

Figure 4.8: *Training time versus validation PESQ score for the historical set reduction techniques.*

O_{cutoff}	**Train Hours** t	**PESQ**	$\Delta_{\text{PESQ}}/\mathbf{t}$	**STOI**	**Csig**	**Cbak**	**Covl**	**SI-SDR**
Noisy	–	*1.97*	–	*0.92*	*3.35*	*2.44*	*2.63*	*8.98*
Baseline (∞)	77.44	**3.25**	0.017	**0.93**	**4.18**	**3.12**	**3.70**	6.21
5	**19.65**	2.98	0.051	0.92	3.92	2.97	3.43	**7.35**
10	20.60	3.06	0.053	0.93	4.06	3.00	3.55	7.06
15	20.59	3.14	**0.057**	0.92	4.09	3.06	3.60	5.62
20	20.70	3.13	0.056	0.93	4.07	3.04	3.59	5.25

Table 4.7: *Performance of Cutoff historical set reduction technique with differing values for* O_{cutoff} *on original VoiceBank-DEMAND*

4.3.4 Experiment 3: Phase Aware Enhancement

4.3.4.1 Experiment 3 Setup

In this experiment, MetricGAN+/- frameworks incorporating the phase aware Generator structure $\mathcal{G}_{\text{CMGAN}}$. In the case of $\mathcal{G}_{\text{CMGAN}}$, frameworks were trained both using the Simple and Sampled training scheduling denoted as α and β respectively. Note that unlike in (Cao et al., 2022), here a validation set is utilised, which means that the training set is roughly 10% smaller than for the models trained in that work; those results are provided as a baseline here. As a comparison,a

version of CMGAN is trained where $\mathcal{D}$ is exposed to the noisy audio x but $\mathcal{N}$ is not used (denoted as CMGAN+ α). For an overview of all the proposed frameworks and baselines, see Table 4.1.

4.3.4.2 Experiment 3 Results

Table 4.9 shows the results for the experiments described above on the original VoiceBank-DEMAND testset. The CMGAN framework with a validation stage under-performs the baseline values reported in the original paper due to having access to less training data. However, with the introduction of the additional loss term for $\mathcal{D}$ introduced in (Fu et al., 2019), comparable performance to this baseline is achieved by CMGAN+ α.
The two frameworks which incorperate the Degenerator $\mathcal{N}$ concept, CMGAN+/- α and CMGAN+/- β do not perform well. In the case of CMGAN+/- α, the overall performance is worse than that of CMGAN+ α, and only slight outperforms CMGAN. CMGAN+/- β is worse again, under-performing even the MetricGAN+/- baseline in all measures but Cbak. This suggests that having different network structures for $\mathcal{G}$ and $\mathcal{N}$ is detrimental to performance, as the simpler network (in this case $\mathcal{N}$) is able to learn it's optimal parameters more quickly than $\mathcal{G}$.

To test, this another version of the CMGAN+/- β framework was trained, using the CMGAN network structure for both $\mathcal{G}$ and $\mathcal{N}$; the results for this (denoted as CMGAN+/- β ditto) are shown in Table Table 4.8. By matching the structure of $\mathcal{G}$ and $\mathcal{N}$, performance is somewhat improved.

Model	PESQ	STOI	Csig	Cbak	Covl	SI-SDR
Noisy	*1.97*	*0.92*	*3.35*	*2.44*	*2.63*	*8.98*
CMGAN+/- β	3.13	0.95	4.38	3.42	3.77	17.43
CMGAN+/- β ditto	3.20	0.95	4.49	3.57	3.87	18.54
CMGAN+/- β ditto $w = 0.45$	**3.24**	**0.95**	**4.50**	**3.58**	**3.90**	**18.12**
CMGAN+/- β $L_{G_{\text{GAN}}}$ only	3.17	0.94	4.27	3.11	3.72	10.46
CMGAN+/- β no history	3.05	0.94	4.36	3.39	3.72	18.13

Table 4.8: *Performance of CMGAN+/- β ditto with CMGAN network structure for $\mathcal{G}$ and $\mathcal{N}$*

Model	PESQ	STOI	Csig	Cbak	Covl	SI-SDR
Noisy	*1.97*	*0.92*	*3.35*	*2.44*	*2.63*	*8.98*
CMGAN (Cao et al., 2022) (reported, no valid)	*3.41*	*0.96*	*4.54*	*3.82*	*4.02*	*20.66*
MetricGAN+/-(Chapter 3)	3.25	0.93	4.18	3.12	3.70	6.21
CMGAN α w/ valid	3.39	0.96	4.49	3.80	3.99	20.56
CMGAN+ α	**3.40**	**0.96**	**4.53**	**3.82**	**4.02**	**20.64**
CMGAN+/- α	3.39	0.96	4.52	3.79	4.00	20.40
CMGAN+/- β	3.13	0.95	4.38	3.42	3.77	17.43

Table 4.9: *Performance of MetricGAN+/- with phase aware Generators on original VoiceBank-DEMAND*

4.3.5 Experiment 4: ASR based enhancement objective

4.3.5.1 Experiment 4 Setup

In this experiment MeticGAN+/- frameworks are trained with minimising the Word Error Rate (WER) of a particular ASR system as the objective metric. Both the original and rerecorded

VoiceBank-DEMAND datasets are used. The value of w is 0.45 in both cases. The ASR system used is based on HuBERT (Hsu et al., 2021) (cf. Section 2.5.2) and trained on LibriSpeech-960 (Panayotov et al., 2015), sourced from HuggingFace[1].

4.3.5.2 Experiment 4 Results

Table 4.10 and Table 4.11 show the results for the WER objective for the original and rerecorded VoiceBank-DEMAND datasets.The baseline system here is the best performing PESQ objective model from the previous experiments. On the original dataset, the WER objective performed extremely poorly, degrading all performance measures, including WER even over the input noisy signals. However, for the rerecorded dataset, a slight improvement in WER is found over the noisy input, as well as over the baseline PESQ objective system. These results show that while WER objective enhancement is technically possible with a MetricGAN system, the performance gain is not large.

Model	**PESQ**	**STOI**	**Csig**	**Cbak**	**Covl**	**SI-SDR**	**WER**
Clean	–	–	–	–	–	–	*8.0*
Noisy	*1.97*	*0.92*	*3.35*	*2.44*	*2.63*	***8.98***	***10.3***
Baseline	**3.25**	**0.93**	**4.18**	**3.12**	**3.70**	6.21	10.9
WER Objective	1.49	0.89	2.90	2.02	2.14	1.36	11.2

Table 4.10: *WER objective on original VoiceBank-DEMAND*

Model	**PESQ**	**STOI**	**Csig**	**Cbak**	**Covl**	**SI-SDR**	**WER**
Clean	–	–	–	–	–	–	*7.7*
Noisy	*1.67*	*0.88*	*2.81*	*2.04*	*2.18*	***1.08***	*15.4*
Baseline	**2.92**	0.88	**3.30**	**2.64**	**3.04**	0.05	32.7
WER Objective	1.62	**0.89**	1.98	2.01	1.72	0.60	**15.2**

Table 4.11: *WER objective on rerecorded VoiceBank-DEMAND*

4.4 Summary

In this chapter, a number of additional experiments further exploring the de-generator extension and it's training objective are carried out. Experiments relating to the network structure of the NNSE Generator DNN are carried out as well as related to the training scheme and replay buffer. Finally, an experiment investigating optimisation towards WER of an ASR system are performed.

[1] https://huggingface.co/facebook/hubert-large-ls960-ft

Chapter 5

CMGAN+/+: Optimising Speech Enhancement towards non-intrusive MOS Predictors

The CHiME-7 UDASE challenge (cf. Section 2.9.3) targets domain adaptation to unlabelled speech data. This chapter describes a proposed NNSE system submitted to the challenge. A CMGAN (cf. Section 2.13) based framwork is used; the discriminator of the GAN is trained to predict the output score of a DNSMOS metric. Additional data augmentation strategies are employed which provide the discriminator with historical training data outputs as well as more diverse training examples from an additional pseudo-generator. The proposed approach, denoted as CMGAN+/+, achieves significant improvement in DNSMOS evaluation metrics with the best proposed system achieving 3.51 OVR-MOS, a 24% improvement over the baseline.

5.1 Speech Enhancement System Description

The overall architecture of the proposed system in this chapter is largely based on the CMGAN (Cao et al., 2022) framework described in Section 2.13, but with two extensions proposed in (Fu, Yu, Hsieh, et al., 2021) (cf. Section 2.12) and Chapter 3. The first extension is to train the discriminator $\mathcal{D}$ on a historical set of past generator outputs every epoch. The second extension is to train $\mathcal{D}$ to predict the metric score of noisy, clean and enhanced audio, as well as the output of a secondary pseudo-generator network $\mathcal{N}$ which is designed to increase the range of metric values observed by $\mathcal{D}$. This chapter introduces a new structure for $\mathcal{D}$, as well as a new input feature which is derived from a pre-trained SSSR (cf. Section 2.5.2).

5.1.1 Conformer-based Generator

The Conformer model generator $\mathcal{G}$ is based on the best performing CMGAN configuration in (Cao et al., 2022). The network itself combines mapping and masking approaches for spectral speech enhancement, utilising a conformer (Gulati et al., 2020) based bottleneck; see Section 2.13 for details. The model is trained with the multi-term loss function detailed in Section 4.2.1.1. Note that

as the metric being predicted is non-intrusive, $L_{\mathcal{G}_{\text{GAN}}}$ is defined here as

$$L_{\mathcal{G}_{\text{GAN}}} = (\mathcal{D}(\hat{s}[n]) - 1)^2 \tag{5.1}$$

which represents an assessment of the enhanced signal by the metric Discriminator $\mathcal{D}$. The 1 in (5.1) represents the highest possible DNSMOS value of 5 after being normalised between 0 and 1.

5.1.2 Metric Estimation Discriminator

The discriminator $\mathcal{D}$ part of the GAN structure is trained to predict a normalised DNSMOS (Reddy et al., 2022) score for a given input signal. Inference of $\mathcal{D}$ is used as in (2.57) as one of the loss terms of $\mathcal{G}$ and as the sole loss function of $\mathcal{N}$ in (3.2), enforcing an optimisation towards the target metric.
Experiments with training $\mathcal{D}$ to predict one of the outputs of Deep Noise Suppression Mean Opinion Score (DNSMOS) (i.e $Q_{\text{SIG}}, Q_{\text{BAK}}$ or Q_{OVR}) (cf. Section 2.7.7) are conducted.

5.1.2.1 Discriminator Network Structure

The discriminator network structure consists of 2 BLSTM layers followed by a single attention feed-forward layer with a sigmoid activation, similar to the network proposed in (Cooper et al., 2022). The input to $\mathcal{D}$ is the output of the HuBERT feature encoder $\mathcal{H}_{\text{FE}}(\cdot)$. Unlike in previous chapters, $\mathcal{D}$ here takes in only the representations of the distorted signal, i.e it is *non-intrusive* (Fu, Yu, Hung, et al., 2021) as the metric being predicted Deep Noise Suppression Mean Opinion Score (DNSMOS) is also non-intrusive.

5.1.2.2 Discriminator Loss Function

Within each epoch, first the Discriminator $\mathcal{D}$ is trained on the current training elements:

$$\begin{aligned} L_{\mathcal{D},\text{MG+}} = \{&(\mathcal{D}(s[n]) - Q'(s))^2 \\ &+ (\mathcal{D}(\hat{s}[n]) - Q'(\hat{s}[n]))^2 \\ &+ (\mathcal{D}(x[n]) - Q'(x[n]))^2 \\ &+ \mathcal{D}(y[n]) - Q'(y[n]))^2\} \end{aligned} \tag{5.2}$$

where $s[n]$ is the clean audio, the noisy mixture $x[n]$, the mixture as enhanced by $\mathcal{G}$, $\hat{s}[n]$, and the mixture as enhanced by $\mathcal{N}$, $y[n]$. This is followed by a historical training stage, where $\mathcal{D}$ is trained to predict the metric scores from past outputs of the generative networks $\mathcal{G}$ and $\mathcal{N}$. $Q'(\cdot)$ is the *true* DNSMOS score of the input audio, normalised between 0 and 1.

5.1.3 Metric Data Augmentation Pseudo-Generator

As first proposed in Chapter 3, an additional speech enhancement network $\mathcal{N}$ is trained, and its outputs y used to train the metric prediction discriminator $\mathcal{D}$ (last term in (3.1)) . This model is trained solely using the GAN loss in (3.2). Its network structure is that of the original MetricGAN, detailed in Section 2.12.

5.2 Experiment Setup

The framework is trained on the simulated LibriMix dataset (Cosentino et al., 2020), using the same data loading configuration as the teacher network in the baseline system (Leglaive et al., 2023). The labelled LibriMix training set consists of 33900 clean/noisy audio pairs, with the clean speech sourced from the LibriSpeech (Panayotov et al., 2015) dataset and the added noise from WHAM! (Wichern et al., 2019) dataset. The framework is trained for 200 epochs, on a random sample of 100 training elements from the train set in each epoch. The Adam optimizer is used for all three networks, with learning rates of $0.005, 0.005$ and 0.001 for $\mathcal{G}, \mathcal{N}$ and $\mathcal{D}$ respectively. Frameworks are trained where $\mathcal{D}$ is trained to predict target metric DNSMOS terms $Q_{\mathrm{SIG}}, Q_{\mathrm{BAK}}$ and Q_{OVR} individually.
Following the configuration in the original CMGAN system, $\gamma_1, \gamma_2, \gamma_3$ in (4.3) are set to $1, 0.2$ and 0.05 respectively, while α in (4.2) is set to 0.9. An additional simulation completely disabling the GAN component of the framework, i.e. setting γ_3 to 0, as well as training solely using the GAN loss by setting γ_1 and γ_2 to 0 and γ_3 to 1 are performed. We further experiment with the addition of the SI-SDR loss (cf. (2.48)) (Roux et al., 2018):

$$L_{\mathcal{G}L+\mathrm{SI-SDR}} = L_{\mathcal{G}} + L_{\mathrm{SI-SDR}}(s, \hat{s}) \tag{5.3}$$

Additionally, we experiment with setting w, the hyperparameter which controls the objective of $\mathcal{N}$ in (3.2), to 1.0, 0.8 and 0.45.
At evaluation time, the best-performing epoch in terms of the target metric on the LibriMix validation set is loaded. Note that only the labelled portion of the challenge training data is used in taining, unlike the baseline system. Additionally,results are reported for the best-performing epoch after further fine-tuning for 20 epochs on the labelled LibriCHiME dev set which consists is similar to LibriMix but with the noise sourced from the real CHiME recordings.

5.3 Results

Table 6.2 shows the results of the baseline systems and the proposed systems (for different w in ((4.4)) and different target metrics Q from ((2.52)) on the simulated Reverberant LibriCHiME evaluation set in terms of Scale Invariant Speech Distortion Ratio (SI-SDR) score. Here, the proposed system shows generally lower performance than the baselines, with the exception of the models which are trained with Q_{BAK} as their target metric. The model trained with a w value of 0.8 with Q_{BAK} as the objective when fine-tuned in the LibriCHiME dev set was able to achieve an average SI-SDR score of 7.41 dB. Similarly, the model trained with a w value of 1 and Q_{OVR} achieves an average SI-SDR score of 7.41 dB. The relatively poor overall performance by the proposed systems in terms of SI-SDR as evaluation metric can perhaps be explained by the fact that the baseline systems all explicitly use SI-SDR as a loss function during training; our system which incorporates SI-SDR loss directly outperforms the baseline in this measure as shown in the following.
Table 5.2 show results of the baseline systems and the proposed systems on the real CHiME evaluation set in terms of DNSMOS scores. Here, the proposed systems all show a marked improvement over the baseline systems, with an improvement in terms of the target metric after fine-tuning in most cases. Furthermore, the inclusion of the GAN term in (4.3) also has a significant effect on this measure, as shown by the performance of the proposed system without the GAN term. Unlike Q_{SIG} and Q_{BAK} fine-tuning on the LibriCHiME dev set degrades performance on the models

Model	w	Q	**SI-SDR (dB)**
unprocessed	–	–	*6.59*
Sudo rm -rf (Tzinis et al., 2020)	–	–	7.8
RemixIT (Tzinis et al., 2022)	–	–	9.44
RemixIT (Tzinis et al., 2022) w/ VAD	–	–	**10.05**
CMGAN+/+	1.00	SIG	4.71
fine-tuned			3.55
CMGAN+/+	0.80	SIG	4.53
fine-tuned			3.55
CMGAN+/+	0.45	SIG	5.98
fine-tuned			4.30
CMGAN+/+	1.00	BAK	6.95
fine-tuned			6.89
CMGAN+/+	0.80	BAK	6.31
fine-tuned			7.39
CMGAN+/+	0.45	BAK	6.42
fine-tuned			5.84
CMGAN+/+	1.00	OVR	7.41
fine-tuned			4.29
CMGAN+/+	0.80	OVR	1.19
fine-tuned			5.15
CMGAN+/+	0.45	OVR	4.75
fine-tuned			6.78
no GAN term	–	–	6.61
GAN only	1.00	SIG	-30.97
GAN only	1.00	BAK	-67.28
GAN only	1.00	OVR	-41.60
CMGAN+/+ + SD-SDR	1.0	SIG	**10.13**

Table 5.1: *SI-SDR results on the reverberant LibriCHiME eval set.*

trained towards Q_{OVR}. Generally, the models trained with a w value of 1 perform better than the other values; this may be caused by the difficulty of the task of $\mathcal{N}$ to enhance or 'de-enhance' the input audio representation.
The results for the model trained solely using the GAN term towards Q_{SIG} are shown in the last row of Table 5.2. While this model shows good performance on its target metric, it scores rather poorly on the other two DNSMOS components. Furthermore, when played back, audio enhanced by this system is *significantly* distorted, with barely any of the original signal retained. The models trained only using the GAN term towards Q_{BAK} and Q_{SIG} are similarly distorted.

5.3.1 Spectrogram Analysis

Figure 5.1 shows spectrograms for noisy (upper panel in Figure 5.1) and enhanced audio by the system with Q_{SIG} as target metric and a w of 1 (second panel), the system with no GAN term (3rd panel) and the system using the GAN term only (also with Q_{SIG}, w of 1, lower panel in Figure 5.1). In the lower panel of Figure 5.1, the significant distortion of the signal by the GAN-only model is visible, despite it achieving a similar DNSMOS SIG improvement relative to the noisy input as the

Model	w	Q	**OVR**	**BAK**	**SIG**
unprocessed	-		*2.84*	*2.92*	*3.48*
Sudo rm -rf (Tzinis et al., 2020)	-		2.88	3.59	3.33
RemixIT (Tzinis et al., 2022)	-		2.82	3.64	3.26
RemixIT (Tzinis et al., 2022) w/ VAD	-		2.84	3.62	3.28
CMGAN+/+	1.00	SIG	3.29	3.85	3.76
fine-tuned			3.45	3.90	3.98
CMGAN+/+	0.80	SIG	3.20	3.70	3.68
fine-tuned			3.37	3.46	3.86
CMGAN+/+	0.45	SIG	3.33	3.81	3.80
fine-tuned			3.49	3.90	**3.98**
CMGAN+/+	1.00	BAK	3.12	3.90	3.39
fine-tuned			3.28	**4.08**	3.29
CMGAN+/+	0.80	BAK	3.06	3.82	3.32
fine-tuned			3.15	3.95	3.07
CMGAN+/+	0.45	BAK	2.87	3.74	3.18
fine-tuned			3.08	3.87	3.23
CMGAN+/+	1.00	OVR	**3.51**	3.99	3.78
fine-tuned			2.60	3.25	3.14
CMGAN+/+	0.80	OVR	3.37	3.87	3.56
fine-tuned			2.75	3.27	3.27
CMGAN+/+	0.45	OVR	3.23	3.94	3.33
fine-tuned			2.84	3.24	3.26
no GAN term	–	–	2.87	3.54	3.34
GAN only	1.00	SIG	2.66	1.58	3.72
GAN only	1.00	BAK	2.67	3.78	2.41
GAN only	1.00	OVR	2.70	3.68	3.00
CMGAN+/+ + SI-SDR	1.00	SIG	3.04	3.70	3.42

Table 5.2: *DNSMOS results on CHiME5 eval set.*

other enhancement models. This suggests that the model has learned to 'enhance' the input audio in a way to trick the DNSMOS SIG metric into awarding it high scores. The reason as to why DNSMOS awards such high scores to significantly distorted audio remains unknown; it is possible that as DNSMOS is a data-driven system itself, the problem arises from its neural network not ever observing audio which has been distorted in such a way during its own training, resulting in it assigning an effectively meaningless score.

5.3.2 Challenge Results

Table 5.3 compares the challenge entries in terms of DNSMOS and SI-SDR on the sim challenge evaluation sets.

		CHiME-5 (DNSMOS)			**Reverb Libri-CHiME-5**
Rank	**System**	**OVRL**	**BAK**	**SIG**	**SI-SDR (dB)**
1	CMGAN+/+ fine	**3.55**	3.93	**3.92**	4.7
2	CMGAN+/+	3.40	**3.97**	3.76	7.8
3	NWPU/ByteAudio (Z. Zhang et al., 2023)	3.07	3.93	3.39	**13.0**
4	Sogang ISDS1 (Jang & Koo, 2023)	2.90	3.60	3.39	12.4
5	Sogang ISDS2 (Jang & Koo, 2023)	2.88	3.70	3.32	12.4
6	*OOD teacher* (Leglaive et al., 2023)	*2.88*	*3.59*	*3.33*	*7.8*
7	*RemixIT-VAD* (Tzinis et al., 2022)	*2.84*	*3.62*	*3.28*	*10.1*
8	*Unprocessed*	*2.84*	*2.92*	*3.48*	*6.6*
9	*RemixIT* (Tzinis et al., 2022)	*2.82*	*3.64*	*3.26*	*9.4*

Table 5.3: *Comparison with other challenge entries ranked by DNSMOS OVR score.*

The submitted system uses DNSMOS SIG as its target metric with a w value of 1. Note that the results shown here for our submitted systems differ slightly from those in the previous section, as they come from different runs of the model on a different random seed. Both our base and fine-tuned models significantly outperform all other entries in terms of DNSMOS on the real CHiME-5 evaluation set, but show lower performance for SI-SDR as target metric. After evaluation by the challenge organisers in terms of DNSMOS and SI-SDR as shown in Table 5.3, the two best-performing systems for each of the two target metrics (including the proposed system) were evaluated in listening tests.Table 5.4 shows the results listening-tests of audio enhanced by the top-performing systems, as well as the unprocessed audio. Interestingly, the proposed system shows lower performance in the listening tests than expected from the high scores in terms of DNSMOS in Table 5.4.

5.4 Summary

In this chapter, the CMGAN+/+ speech enhancement system for the CHiME-7 UDASE challenge is described. The system uses a GAN-based model with discriminator input data augmentation strategies to improve metric prediction performance. Results on the unlabelled CHiME-5 evaluation set demonstrate improvements in DNSMOS evaluation metrics, significantly outperforming the

		CHiME-5 (Listening Tests)		
Rank	**System**	**OVRL**	**BAK**	**SIG**
1	NWPU/ByteAudio	**3.11**	**4.30**	3.41
2	Sogang ISDS1	2.75	3.08	3.43
3	*Unprocessed*	*2.68*	*2.20*	***3.97***
4	*RemixIT-VAD*	*2.45*	*2.97*	*3.02*
5	CMGAN+/+ fine	2.14	2.75	2.63

Table 5.4: *Comparison of top-performing challenge entries on listening tests with human participants, ranked by OVRL MOS.*

baseline system in OVR, BAK and SIG measures. However, this does not directly translate to high ratings in listening tests with humans. By training solely using a metric optimisation loss, possible flaws in the metric being optimised towards have to be considered.

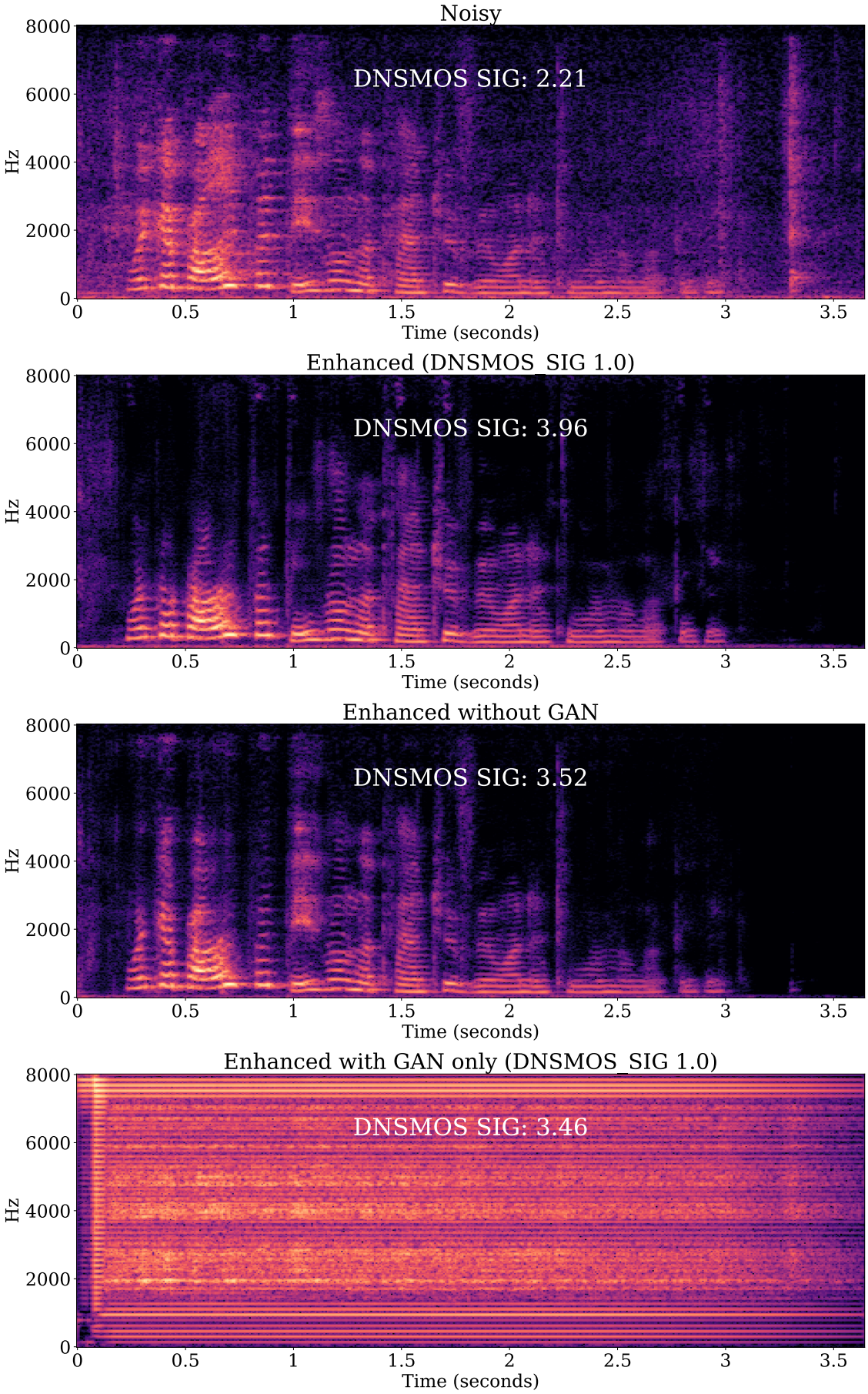

Figure 5.1: *Noisy and enhanced spectrograms of audio file* `S01_P01_0.wav` *from the CHiME-5 evaluation set.*

Chapter 6

Multi-CMGAN+/+: Optimizing Speech Enhancement towards Multiple Metrics

6.1 Introduction

This chapter expands on the DNSMOS prediction within the MetricGAN framework introduced in Chapter 5 by introducing a multi-headed predictor which is capable of predicting the scores of multiple metrics simultaneously.

6.2 Speech Enhancement System

The overall architecture of the proposed system is based on the CMGAN framework proposed in (Cao et al., 2022), but with two extensions based on (Fu, Yu, Hsieh, et al., 2021) and (Close, Hain, et al., 2022). The first extension is to train the discriminator $\mathcal{D}$ on a historical set of past generator outputs every epoch. The second extension is to train $\mathcal{D}$ to predict the metric score of noisy, clean and enhanced audio, as well as the output of a secondary pseudo-generator network $\mathcal{N}$ which is designed to increase the range of metric values observed by $\mathcal{D}$. This work introduces a new structure for $\mathcal{D}$ allowing it to predict multiple metrics at once, as well as a new input feature which is derived from a pre-trained SSSR model.

6.2.1 Conformer-based Speech Enhancement Generator

6.2.1.1 Conformer-based Generator Network Structure

The conformer model generator $\mathcal{G}$ is based on the best performing CMGAN configuration in (Cao et al., 2022) (cf. Section 2.13). The network itself combines mapping and masking approaches for spectral speech enhancement, utilising a conformer (cf. Section 2.4.1.4 based bottleneck. The model's input are STFT components of the complex-valued noisy audio, $\mathbf{X}_{\mathrm{Re}}, \mathbf{X}_{\mathrm{Im}}$, with a reasonably high temporal resolution (hop size of 6 ms with a 50% overlap, and a fast Fourier transform (FFT) length of 400 samples). The output of the model are the enhanced real and imaginary STFT components $\hat{\mathbf{S}}_{\mathrm{Re}}$ and $\hat{\mathbf{S}}_{\mathrm{Im}}$ from which the enhanced time domain audio $\hat{s}[n]$ is obtained by ISTFT. For more detail see Section 2.13.

6.2.1.2 Generator Loss Function

The generator model $\mathcal{G}$ is trained with a multi-term loss function:

$$L_{\mathcal{G}} = L_{\mathcal{G}_{\text{GAN}}} + L_{\mathcal{G}_{\text{Time}}} + L_{\mathcal{G}_{\text{SI-SDR}}} \tag{6.1}$$

$L_{\mathcal{G}_{\text{GAN}}}$ is (2.57) which represents an assessment of the enhanced signal by the metric Discriminator $\mathcal{D}$. $\mathcal{D}(\hat{s}[n])$ is the inference of the metric prediction discriminator $\mathcal{D}$,given the enhanced signal as input, which has an output of dimension $N_Q \times 1$ representing the N_Q predicted normalised Q' values of the target metrics, i.e. N_Q equals 3 when using (2.52). The $\mathbf{1}$ vector in (2.57), also of length N_Q, represents the highest possible target metric values normalized between 0 and 1. Thus, the net effect of this loss term is to encourage $\mathcal{G}$ to maximise the predicted scores assigned to its outputs by $\mathcal{D}$.
$L_{\mathcal{G}_{\text{Time}}}$ is a mean absolute error between the enhanced and clean time domain mixtures, (2.38). Finally, $L_{\mathcal{G}_{\text{SISDR}}}$ is the SI-SDR (Roux et al., 2018) loss, (2.48). With the exception of $L_{\mathcal{G}_{\text{GAN}}}$, all terms of $L_{\mathcal{G}}$ require access to clean label/reference audio $s[n]$.

6.2.1.3 Block Processing for Longer Inputs

Due to the quadratic time-complexity of the transformer layers in the conformer models, processing long sequences can be unfeasible due to high memory requirements. Transformers are also typically unsuitable for continuous processing as the entire sequence is required to compute self-attention. To address these issues input signals are processed in overlapping blocks of 4s for evaluation and inference as this has been shown to be in an optimal signal length for attention-based enhancement models (Ravenscroft et al., 2023). A 50% overlap with a Hann window is used to cross-fade each block with one an another. Models are trained with 4s signal length limits (Ravenscroft et al., 2023).

6.2.2 Metric Estimation Discriminator

The discriminator $\mathcal{D}$ part of the GAN structure is trained to predict three normalised speech quality metrics for a given input signal. Inference of $\mathcal{D}$ is used in (2.57) as one of the loss terms of $\mathcal{G}$ and as the sole loss function of $\mathcal{N}$ in (3.2), enforcing an optimisation towards the target metrics.
We experiment with training $\mathcal{D}$ to predict each outputs of DNSMOS (i.e $Q_{\text{SIG}}, Q_{\text{BAK}}$ or Q_{OVR}), as well as PESQ (Q_{PESQ}).

6.2.3 Discriminator Network Structure

The discriminator network structure consists of 2 BLSTM layers followed by three parallel attention feed-forward layers with sigmoid activations, similar to the network proposed in (Cooper et al., 2022). Each attention feed-forward layer outputs a single neuron which represents the prediction value of one of the three target metrics. The input feature of $\mathcal{D}$ is the output of the HuBERT feature encoder $\mathcal{H}_{\text{FE}}(\cdot)$; for more detail on this feature see, Section 2.5.2. The output of $\mathcal{D}$ has dimension $B \times N_Q$ where B is the batch size and each of N_Q values represents a normalised predicted metric value. Note that inference of $\mathcal{D}$ is always non-intrusive, even if one of its target metrics such as PESQ is intrusive.

6.2.3.1 Discriminator Loss Function

Within each epoch, first the Discriminator $\mathcal{D}$ is trained on the current training elements:

$$\begin{aligned} L_{\mathcal{D},\text{MG+}} = \mathbb{E}\{&(\mathcal{D}(s[n]) - [Q'_1(s), ..., Q'_{N_Q}(s)])^2 \\ &+ (\mathcal{D}(\hat{s}[n]) - [Q'_1(\hat{s}), ..., Q'_{N_Q}(\hat{s})])^2 \\ &+ (\mathcal{D}(x[n]) - [Q'_1(x), ..., Q'_{N_Q}(x)])^2 \\ &+ (\mathcal{D}(y[n]) - [Q'_1(y), ..., Q'_{N_Q}(y)])^2\} \end{aligned} \tag{6.2}$$

. $Q'_1(\cdot), Q'_2(\cdot)$ and $Q'_3(\cdot)$ are the true target metric scores of the input audio, normalized between 0 and 1. The Q' vectors in (6.2) can be shorter than 3 if less than $N_Q = 3$ metrics are considered. This is followed by a historical training stage, where $\mathcal{D}$ is trained to predict the metric scores from past outputs of the generative networks $\mathcal{G}$ and $\mathcal{N}$.

6.3 Experiments

6.3.1 Training Setup

The framework is trained on simulated labelled data from the LibriMix (Cosentino et al., 2020) for 200 epochs, following a similar dataloading system as in (Leglaive et al., 2023) generating mixtures of a single speaker with noise. The labelled LibriMix training set consists of 33900 clean/noisy audio pairs, with the clean speech sourced from the LibriSpeech (Panayotov et al., 2015) dataset and the added noise from WHAM! (Wichern et al., 2019) dataset.
Each epoch, 300 samples from the training set are randomly selected. These are first used to train the metric prediction Discriminator $\mathcal{D}$ using (5.2). This is followed by the training of $\mathcal{D}$ on the historical set. Then the 300 random samples are used to train $\mathcal{N}$ using inference of $\mathcal{D}$ with (5.2), followed finally by the training of $\mathcal{G}$ using (4.3) which also uses inference of $\mathcal{D}$.
Different combinations of the DNSMOS terms and PESQ are experimented with as the three target metrics for $\mathcal{D}$ by setting each of Q_1, Q_2, Q_3 in (5.2) to be $Q_{\text{PESQ}}, Q_{\text{SIG}}, Q_{\text{BAK}}$ or Q_{OVR}.
The proposed models are evaluated on the CHiME7 UDASE task (Leglaive et al., 2023) evaluation sets. These are a real world unlabelled set consisting of CHIME5 recordings which are evaluated using DNSMOS and a simulated labelled set consisting of reverberant LibriMix audio which are evaluated using SI-SDR. The proposed system is compared to CMGAN+/+ (cf. Chapter 5) as well as the challenge baselines (Leglaive et al., 2023).

6.4 Results

Table 6.1 shows the results of the proposed framework in terms of DNSMOS on the CHiME-7 UDASE task real evaluation set. The proposed systems significantly outperform the baseline systems in all measures, while also outperforming the author's prior work CMGAN+/+ in terms of OVR and BAK. However, CMGAN+/+ still outperforms the proposed system in terms of SIG, which is the only metric it is optimized towards.

Table 6.1: *DNSMOS results on CHiME5 eval set.*

Model	Q_1, Q_2, Q_3	**OVR**	**BAK**	**SIG**
unprocessed	–	*2.84*	*2.92*	*3.48*
Sudo rm -rf (Tzinis et al., 2020)	–	2.88	3.59	3.33
RemixIT (Tzinis et al., 2022) w/VAD	–	2.84	3.62	3.28
CMGAN+/+ (cf. Chapter 5)	SIG	3.29	3.85	**3.76**
Multi-CMGAN+/+	SIG/BAK/OVR	**3.42**	3.86	3.56
Multi-CMGAN+/+	SIG/BAK/PESQ	3.08	3.78	3.41
Multi-CMGAN+/+	SIG/OVR/PESQ	2.80	3.62	3.19
Multi-CMGAN+/+	BAK/OVR/PESQ	3.12	**3.86**	3.49

Table 6.2: *SI-SDR results on the reverberant LibriCHiME eval set.*

Model	Q_1, Q_2, Q_3	**SI-SDR (dB)**
unprocessed	–	*6.59*
Sudo rm -rf (Tzinis et al., 2020)	–	7.8
RemixIT (Tzinis et al., 2022) w/ VAD	–	**10.05**
CMGAN+/+	SIG	4.71
Multi-CMGAN+/+	SIG/BAK/OVR	3.36
Multi-CMGAN+/+	SIG/BAK/PESQ	4.47
Multi-CMGAN+/+	SIG/OVR/PESQ	0.09
Multi-CMGAN+/+	BAK/OVR/PESQ	6.95

Table 6.2 shows the results of the proposed framework in terms of SI-SDR on the CHiME-7 UDASE task simulated evaluation set. Here, the weaknesses of the proposed system relative to the CHiME-7 baseline systems is apparent, with our proposed framework significantly degrading the input with the exception of the model which does *not* optimise the SIG component of DNSMOS.

6.5 Summary

In this chapter a MetricGAN framework utilising a multi-metric prediction discriminator is introduced. A number of combinations of target metric for this prediction network are experimented with, and improved performance on test set consisting of real data is shown. However a degradation in performance on a simulated testset is also shown, suggesting a significant distortion in the enhanced outputs of the proposed system. This idea is futher developed in Chapter 12.

Part III

Self Supervised Speech Representation Based Loss Functions

Preface

In this part, novel perceptually motivated loss functions for NNSE systems are introduced. These proposed loss functions are computed from intrusive comparison between representations of the clean reference signal and the output of the NNSE system which are derived from Self Supervised Speech Representations (SSSRs) (cf. Section 2.5.2). In Chapter 7 the correlation between a standard STFT based loss function and perceptual metrics and that of the proposed SSSR based loss functions is explored. Then in Chapter 8 the relationship between the data used to train the SSSR representation and that used to train the NNSE system is investigated, with a focus on the language of the speech audio.

Chapter 7

Self Supervised Speech Representation Loss Functions for Speech Enhancement

7.1 Introduction

Recent work in the domain of speech enhancement has explored the use of self-supervised speech representations (cf. Section 2.5.2) to aid in the training of neural speech enhancement models. However, much of this work focuses on using the deepest or final outputs of self supervised speech representation models, rather than the earlier feature encodings. In this chapter it is shown that the distance between the feature encodings of clean and noisy speech correlate strongly with psychoacoustically motivated measures of speech quality and intelligibility, as well as with human MOS ratings. Experiments using this distance as a loss function are performed and improved performance over the use of STFT spectrogram distance based loss as well as other common loss functions from speech enhancement literature is demonstrated using objective measures such as PESQ and STOI.

7.2 SSSR derived distances in relation to speech assessment metrics

In this work, first the Mean Squared Error (MSE) distance between representations of some clean speech $s[n]$ and a corresponding noisy version of $s[n]$, $x[n]$ is analysed (cf. (2.1)). Specifically, we define these using either $\mathbf{S}_{\mathrm{FE}}$, $\mathbf{X}_{\mathrm{FE}}$ or $\mathbf{S}_{\mathrm{OL}}$, $\mathbf{X}_{\mathrm{OL}}$ where FE and OL denote the SSSR encoder representation and final output layer respectively, as detailed in Section 2.5.2 and shown in Figure 2.19. The MSE distances between these SSSR representations are defined as:

$$d_{\mathrm{FE}}(\mathbf{S}_{\mathrm{FE}}, \mathbf{X}_{\mathrm{FE}}) = \frac{1}{T \cdot F} \sum_{t}^{T} \sum_{f}^{F} (\mathbf{S}_{\mathrm{FE}}[t, f] - \mathbf{X}_{\mathrm{FE}}[t, f])^2 \tag{7.1}$$

$$d_{\mathrm{OL}}(\mathbf{S}_{\mathrm{OL}}, \mathbf{X}_{\mathrm{OL}}) = \frac{1}{T \cdot F} \sum_{t}^{T} \sum_{f}^{F} (\mathbf{S}_{\mathrm{OL}}[t, f] - \mathbf{X}_{\mathrm{OL}}[t, f])^2 \tag{7.2}$$

where T and F denote time and feature dimensions of the representation.

	PESQ		STOI		Csig		Cbak		Covl		MOS	
	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ	r	ρ
d_{SG}	-0.66	-0.53	-0.60	-0.68	-0.75	-0.70	-0.84	-0.69	-0.74	-0.64	0.35	-0.27
XLSR d_{FE}	-0.82	-0.78	**-0.80**	**-0.81**	-0.93	-0.92	-0.88	-0.85	-0.90	-0.87	-0.47	-0.43
XLSR d_{OL}	-0.66	-0.61	-0.69	-0.68	-0.76	-0.75	-0.74	-0.72	-0.74	-0.71	-0.44	-0.40
HuBERT d_{FE}	**-0.83**	**-0.79**	-0.75	-0.76	**-0.95**	**-0.93**	**-0.90**	**-0.87**	**-0.91**	**-0.89**	**-0.48**	**-0.46**
HuBERT d_{OL}	-0.44	-0.43	-0.40	-0.42	-0.52	-0.52	-0.45	-0.45	-0.50	-0.49	-0.42	-0.37

Table 7.1: *Spearman r and Pearson ρ correlation between distance measures and speech quality and intelligibility metrics in the VoiceBank-DEMAND testset, as well as MOS in the NISQA Challenge testset*

q These SSSR derived distances are compared with a distance which is commonly used as a loss function in speech enhancement tasks the magnitude spectrogram MSE distance, (2.39), here called d_{SG}. In the following, d_{FE} and d_{OL} are computed using the XLSR and HuBERT models. As mentioned in the previous section, $\mathbf{S}_{\mathrm{FE}}$, $\mathbf{X}_{\mathrm{FE}}$ and $\mathbf{S}_{\mathrm{OL}}$, $\mathbf{X}_{\mathrm{OL}}$ of the XLSR model have feature dimensions F of 512 and 1024 respectively, while those of HuBERT have feature dimensions of 512 and 768, sharing a time dimension T, the size of which is dependent on the length in samples of the input time domain audio.
$\mathbf{S}_{\mathrm{SG}}$ and $\mathbf{X}_{\mathrm{SG}}$ are computed using a Fourier Transform with an FFT size of 512, window length of 32 ms, and a hop length of 16 ms (resulting in a 50% frame overlap) using a hamming window. This results in a spectogram with a frequency dimension F_{Hz} of 257, and a time dimension T which is dependent on the length in samples of the input time domain audio.

7.2.1 Datasets Used

To express the relationship between the distance measures and psychoacoustically motivated metrics the VoiceBank-DEMAND (Valentini-Botinhao et al., 2016)(cf. Section 2.9.1) is used.
In order to assess the relationship between the distance measures and human MOS ratings, the NISQA (Mittag et al., 2021)(cf. Section 2.11) dataset is used. This is a dataset of variable length clean and noisy speech audio file pairs $(s[n], x[n])$ with real human-annotated MOS labels, designed for the training and testing of neural SQ MOS predictors.The two testsets (P501 and FOR) used here contain 440 clean/noisy pairs in total.
The audio files in both datasets have a sample rate of 48 kHz and are down-sampled to 16 kHz such that $\mathcal{G}(\cdot)$ in ((2.42)), ((2.43)) can computed.

7.2.2 SSSR distances and SE motivated metrics

Fig. 7.1 shows the relationships between the distance measures and PESQ (Rix et al., 2001) scores computed using the $s[n]$, $x[n]$ pairs in the VoiceBank-DEMAND (Valentini-Botinhao et al., 2016) testset using d_{SG}, HuBERT d_{FE} and HuBERT d_{OL} . From this, it can be observed that HuBERT d_{FE} correlates significantly more strongly than d_{SG} with PESQ. Furthermore, the distance computed using the output of the 1D convolutional encoder d_{FE} correlates more strongly than the distance computed using the SSSR output d_{OL}. This suggests that the phonetic and linguistic processing which occurs in the deeper parts of the model are less sensitive to the noise in $x[n]$. The first 5 columns of Table 7.1 shows the Spearman r and Pearson correlations ρ between PESQ, STOI and the components of the Composite (Lin et al., 2019) measure. Like with PESQ and STOI, the

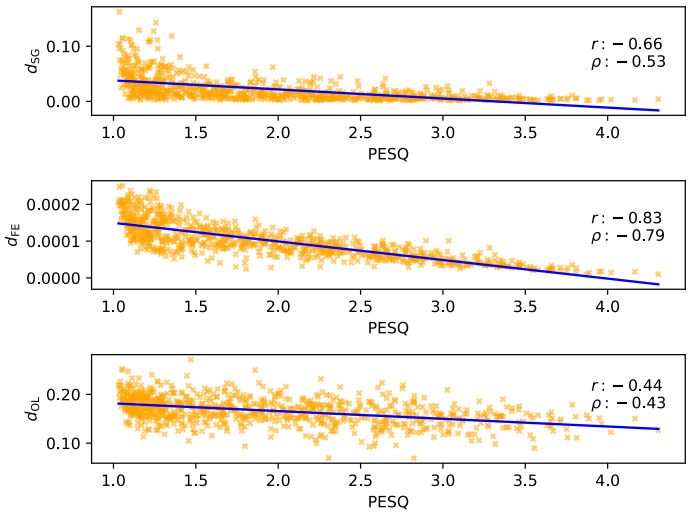

Figure 7.1: *Scatter plots showing the relationship between the PESQ metric and MSE Spectrogram distance d_{SG} as well as, HuBERT d_{FE}, HuBERT d_{OL} distances for the VoiceBank-DEMAND testset*

Composite measure scores all correlate more strongly with the proposed SSSR distances than with d_{SG}, in particular the feature encoder derived distance d_{FE}.

7.2.3 SSSR distances and human quality assessment

Fig. 7.2 and the last column of Table 7.1 show the relationship between the MSE distances and human MOSs in the 'FOR' and 'P501' testset $s[n]$, $x[n]$ pairs of the NISQA (cf. Section 2.11.1) dataset. While the overall correlations are lower here than those of the metrics analysed in the first 5 colunms, the same pattern emerges with d_{FE} and d_{OL} correlating more strongly with the MOS scores than d_{SG}. The HuBERT based distances again correlate more strongly than XLSR; this is possibly due to the language match between the training data of HuBERT and that of the data, both being English only.

7.3 SSSR Based Signal Enhancment Experiment

An experiment is carried out in order to assess the effectiveness of the SSSR derived distance measures as loss functions for speech enhancement tasks.

7.3.1 Experiment setup

Simple masking based speech enhancement models were trained using a number of different loss functions; d_{SG}, d_{FE}, d_{OL} as described in the previous sections, as well as Si-SDR loss (cf. (2.48) and STOI loss (2.11)) (Fu, Wang, et al., 2018). Each model was trained for 50 epochs on the VoiceBank-DEMAND (Valentini-Botinhao et al., 2016) training set. d_{FE}, d_{OL} are computed for XLSR or HuBERT feature encoder and output representations. The Adam (Kingma & Ba, 2014) optimiser is used with a learning rate of 0.001. At test time, the epoch obtaining the highest PESQ

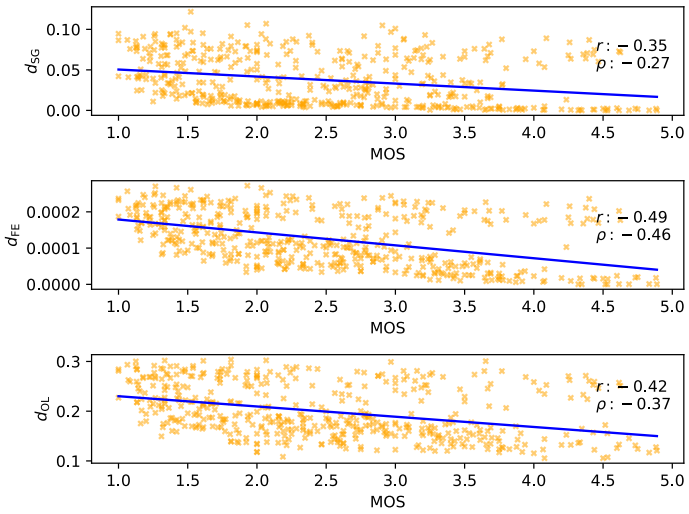

Figure 7.2: *Scatter plots showing the relationship between human MOS scores and MSE Spectrogram distance d_{SG}, HuBERT d_{FE} and HuBERT d_{OL} with in the NISQA Challenge testset*

score on the validation set is loaded. The SpeechBrain (Ravanelli et al., 2021) toolkit is used to implement the experiment.

7.3.2 Loss Functions

The distance measures defined in (7.1), (7.2) and (2.39) are modified to be used as loss terms for a speech enhancement neural model:

$$L_{\mathrm{FE}}(\mathbf{S}_{\mathrm{FE}}, \hat{\mathbf{S}}_{\mathrm{FE}}) = \frac{1}{T \cdot F} \sum_{t}^{T} \sum_{f}^{F} (\mathbf{S}_{\mathrm{FE}}[t, f] - \hat{\mathbf{S}}_{\mathrm{FE}}[t, f])^2 \tag{7.3}$$

$$L_{\mathrm{OL}}(\mathbf{S}_{\mathrm{OL}}, \hat{\mathbf{S}}_{\mathrm{OL}}) = \frac{1}{T \cdot F} \sum_{t}^{T} \sum_{f}^{F} (\mathbf{S}_{\mathrm{OL}}[t, f] - \hat{\mathbf{S}}_{\mathrm{OL}}[t, f])^2 \tag{7.4}$$

$$L_{\mathrm{SG}}(\mathbf{S}_{\mathrm{SG}}, \hat{\mathbf{S}}_{\mathrm{SG}}) = \frac{1}{T \cdot F_{\mathrm{Hz}}} \sum_{t}^{T} \sum_{f_{\mathrm{Hz}}}^{F_{\mathrm{Hz}}} (\mathbf{S}_{\mathrm{SG}}[t, f_{\mathrm{Hz}}] - \hat{\mathbf{S}}_{\mathrm{SG}}[t, f_{\mathrm{Hz}}])^2 \tag{7.5}$$

where $\hat{s}[n]$ is the enhanced time domain audio signal output by the neural model when $x[n]$ is input and $\hat{\mathbf{S}}_{\mathrm{FE}}$, $\hat{\mathbf{S}}_{\mathrm{OL}}$ and $\hat{\mathbf{S}}_{\mathrm{SG}}$ are the feature encoder output, output layer and spectrogram representations of $\hat{s}[n]$ respectively.

7.3.3 Enhancement Model Structure

The MetricGAN+ $\mathcal{G}$ DNN structure is used as the NNSE system in this chapter. See Section 2.12.5 for more details on this network structure.

7.3.4 Signal Enhancement Performance

Table 7.2 shows the experiment results. The proposed model using HuBERT L_{FE} as its loss function outperforms the baseline using the spectrogram distance L_{SG} in terms of PESQ and the Composite measure Csig, Cbak and Covl. Additionally, the best performing model by a significant margin in terms of Cbak uses the XLSR encoder distance loss function L_{FE}, and most SSSR based losses outperform the baseline systems in this measure. Those models which use SSSR encoder distance L_{FE} outperform those which use SSSR output layer distance L_{OL}; this is consistent with the correlation values in Table 7.1 where d_{FE} distances correlate more strongly with the metrics than d_{OL} distances.

Table 7.2: *Signal Enhancement performance on the VoiceBank-DEMAND testset.*

Loss Function	PESQ	STOI	Csig	Cbak	Covl	Si-SDR
noisy	*1.97*	*0.92*	*3.35*	*2.44*	*2.63*	*8.98*
L_{SG}	2.70	**0.94**	4.00	2.62	3.35	18.62
L_{SISDR} (Luo & Mesgarani, 2018)	2.28	0.92	3.51	2.44	2.88	**18.66**
L_{STOI} (Fu, Wang, et al., 2018)	2.12	0.93	3.46	2.16	2.77	13.31
HuBERT L_{FE}	**2.79**	0.94	**4.10**	2.68	**3.44**	18.47
HuBERT L_{OL}	2.55	0.92	3.66	2.42	3.08	14.92
XLSR L_{FE}	2.69	0.92	3.77	**3.05**	3.21	9.72
XLSR L_{OL}	2.43	0.91	3.21	2.64	2.79	13.00

7.3.5 Analysis

Fig. 7.3 shows an example of the feature representations of $s[n]$ and $x[n]$ used as inputs to d_{SG} and d_{FE} for both HuBERT and XLSR. Tonal noise introduced in $x[n]$, visible as a line spanning approximately the first 50 time frames in $\mathbf{X}_{\mathrm{SG}}$, is well represented in the XLSR $\mathbf{X}_{\mathrm{FE}}$ but not in HuBERT $\mathbf{X}_{\mathrm{FE}}$. This is a possible explanation for the increased Cbak score for the XLSR L_{FE} loss over the HubERT based loss L_{FE} as XLSR $\mathbf{X}_{\mathrm{FE}}$ representations appear to be more sensitive to noise in non speech regions of the representation. The fact that XLSR is trained in part on noisy data is a potential explanation for this behaviour.

7.4 Summary

In this chapter it is demonstrated that the earlier 'perceive' feature encoder layers of SSSRs preserve aspects of noise and distortion in speech to a greater degree than the deeper 'predict' layers. Moreover, we find that a simple distance measure between the encoder representations of clean and noisy speech correlates strongly with perceptually motivated metrics of speech quality, as well as with human speech quality assessment. This correlation is affected by the attributes of the data used to train the SSSR. This finding is validated by the use of these distance measures as loss functions for a speech enhancement task, where feature encoder distance outperforms both the deeper output layer and a standard spectrogram based loss. Future work will investigate the effect of the training data on the SSSR encoder representations.

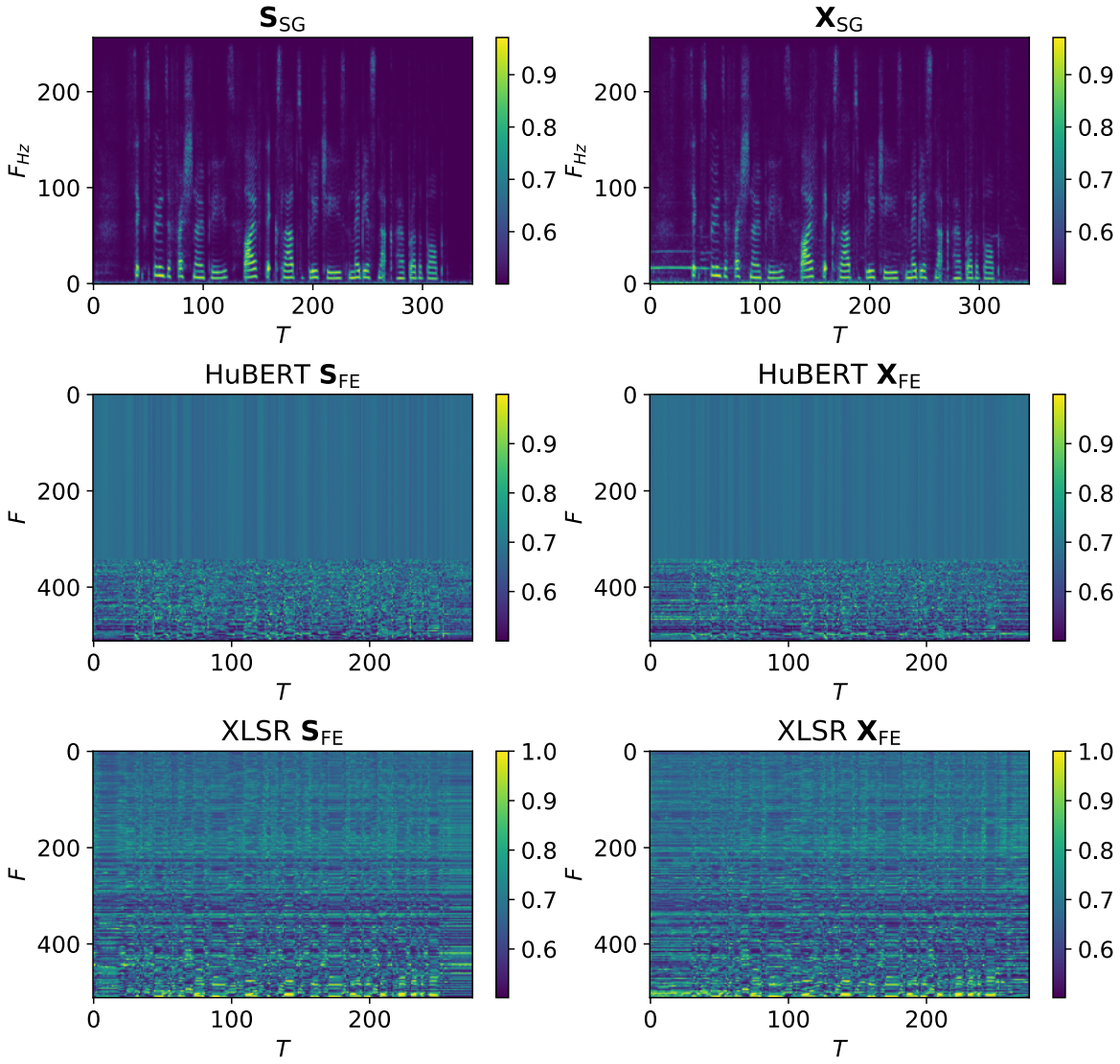

Figure 7.3: *Visualisation of inputs representations of $s[n]$, $x[n]$ to d_{SG}, HuBERT d_{EF} and XLSR d_{EF}. SSSR features are sorted according to depthwise euclidean distance following Algorithm 1 in (Ravenscroft et al., 2022) and a sigmoid function is applied to increase clarity.*

Chapter 8

Investigating the Effect of Language on SSSR based Loss Functions for Neural Network Speech Enhancement (NNSE) Systems

8.1 Introduction

This chapter further develops on the SSSR based loss functions for NNSE systems proposed in the previous chapter. In particular, the focus of this chapter is on the effect of the *language* of the speech being enhanced in relation to that of which was used to train the SSSR.

8.2 CommonVoice-DEMAND: A Multilingual Speech Enhancement Dataset

This section details the creation process of the proposed *CommonVoice-DEMAND* speech enhancement dataset, which is intended to be a multilingual variation on the popular monolingual VoiceBank-DEMAND (VB-D) dataset.

8.2.1 CommonVoice Dataset

To create a multilingual dataset which is as similar as possible to the VB-D dataset described in Section 2.9.1, speech is sourced from the Mozilla CommonVoice (Ardila et al., 2020) dataset (cf. Section 2.9.4) In this work, we make use of the English, Spanish, and Welsh portions of the dataset. These portions contain 3209, 2152 and 152 hours of audio, respectively.

8.2.2 Candidate Selection

Due to their crowd-sourced nature, the quality of the recordings in the CommonVoice dataset varies considerably. For the creation of the multilingual CommonVoice-DEMAND dataset, the aim is

to select the cleanest possible audio for use as reference signals. Additionally, certain signal enhancement metrics require the input audio to have a minimum length. The process for the selection of candidate reference signals for a given language to create the CommonVoice-DEMAND datasets can be summarised as follows:
Firstly, only recordings which have been validated by the crowd-sourced validation process are selected. This is to ensure that the audio does contain the prompt sentence and is not a failed recording or too noisy for the speech to be intelligible.
Secondly, recordings of less than 2 seconds length and those which contain a single-word utterance are excluded since it was found that some speech enhancement metrics have difficulties assessing such recordings.
Finally, the quality of the remaining audio recordings is assessed. A VAD is used to segregate frames of length L of the signal $x[n]$ into disjoint sets $\mathcal{A}$ and $\mathcal{B}$, for which the signal fulfils either the hypothesis that speech is present $\mathcal{H}_1$, or that speech is absent $\mathcal{H}_0$, respectively. An SNR estimate is obtained by

$$\widehat{\mathrm{SNR}}(x[n]) = 10\log_{10}\left(\frac{\frac{1}{|\mathcal{A}|}\sum_{\ell\in\mathcal{A}}\sum_{n=0}^{K-1}x^2[\ell K+n]}{\frac{1}{|\mathcal{B}|}\sum_{\ell\in\mathcal{B}}\sum_{n=0}^{K-1}x^2[\ell K+n]}\right) \tag{8.1}$$

for frame index ℓ with $|\mathcal{A}|$ and $|\mathcal{B}|$ denoting the respective cardinalities of the sets $\mathcal{A}$ and $\mathcal{B}$. For simplicity, the Google WebRTC-VAD[1] is used. A minimum threshold of 50 dB estimated SNR by the above formulation is used to select the candidate recordings. Note that while this is a somewhat crude estimator in that it does not account for the noise which is present in the speech-active frames in $\mathcal{A}$, it was found empirically that this approach works sufficiently well to select high-quality recordings containing little to no background noise, with a low computation overhead. For each language, 20000 of such candidate recordings are selected in this way. During this process, the candidate recordings are converted from the MP3 format to the WAV file format and are up-sampled from 32 kHz sample rate to 48 kHz. This is done for parity with the VoiceBank-DEMAND datset which is created using 48kHz audio; as the resultant mixed audio is downsampled to 16kHz for signal enhancement experiments this sampling rate limit should usually not be a problem.

8.2.3 Dataset Creation

In order to ensure that the proposed CommonVoice-DEMAND datasets are comparable to the original VoiceBank-DEMAND dataset, the log files describing VoiceBank-DEMAND are used. These consist of two lists (one for the training set, one for the test set) for clean audio file $s[n]$, the name of the noise file for $v[n]$, from which a random section of the same length as $s[n]$ is obtained and the desired mixing SNR value. The process for the creation of the CommonVoice-DEMAND datasets is as follows:

- A candidate CommonVoice recording is selected which is closest to the length in seconds to the clean audio recording from the original VoiceBank dataset. This candidate recording is either padded with zero values or truncated such that it is the same number of samples exactly as the original VoiceBank recording.
- The mixing process as described in Section 2.9.1 is carried out, using the selected CommonVoice recording as $s[n]$, and creating a corresponding noisy version $x[n]$ according to (2.54) and (2.55). Since CommomVoice data has a sampling rate of 32 kHz, the resulting

[1] https://github.com/wiseman/py-webrtcvad

sampling rate of CommonVoice-DEMAND is also 32 kHz and thus lower than VoiceBank-DEMAND's 48 kHz. Note, that for SE, data is usually anyhow downsampled to 16 kHz.

- The selected CommonVoice recording is then removed from the list of available candidate recordings, ensuring that uniqueness for each $s[n], x[n]$ pair in the resultant dataset.

The goal of this process is to create a dataset which is as similar as possible to the original VoiceBank-DEMAND dataset, but using clean speech with a different language or source. These datasets differ from the original VoiceBank-DEMAND in the (usually) greater number of speakers in the training and test sets and the exact portion of the noise file from which $v[n]$ is created due to the random sampling of the noise file. The created CommonVoice-DEMAND data ensures reproducibility by fixed seeds in the random number generator.

8.3 Speeech Enhancement Experiments

8.3.1 Experiment Setup

Masking-based speech enhancement networks are trained using SSSR derived loss functions using the proposed CommonVoice-DEMAND datasets. The SpeechBrain (Ravanelli et al., 2021) toolkit is used to facilitate the training and testing of the models. The models are trained for 50 epochs with the Adam (Kingma & Ba, 2014) optimiser, with a learning rate of 0.001. At test time, the model with the highest PESQ (Rix et al., 2001) score on the validation set is loaded and evaluated.
The MetricGAN+ $\mathcal{G}$ DNN structure is used as the NNSE system in this chapter. See Section 2.12.5 for more details on this network structure.

8.3.2 Datasets

CommonVoice-DEMAND training and test sets were generated using the process described above for English and Spanish. These languages were chosen as they match languages used to train the mHuBERT model, and the CommonVoice corpus component for each is sufficiently large. A testset for Welsh was also created as a language which was not used to train HuBERT or mHuBERT but which was used as one of XLSRs 128 languages. The CommonVoice-DEMAND datasets have the same size training and test sets as the original VoiceBank-DEMAND with 11572 and 824 $s[n], x[n]$ pairs, respectively. A validation set of size 770 is created from each CommonVoice-DEMAND training set. All audio is at 16 kHz sample rate.

8.3.3 SSSR Signal Enhancement Loss Function

The SSSR loss function as defined in Chapter 7 is used, which is based on the MSE distance between the output feature encoder representations of the enhanced signal $\hat{s}[n]$ and the reference signal $s[n]$. The SSSR loss function is given by

$$L_{\mathrm{FE}}(\mathbf{S}_{\mathrm{FE}}, \hat{\mathbf{S}}_{\mathrm{FE}}) = \frac{1}{TF} \sum_{t,f} (\mathbf{S}_{\mathrm{FE}}[t,f] - \hat{\mathbf{S}}_{\mathrm{FE}}[t,f])^2, \tag{8.2}$$

where $\mathbf{S}_{\mathrm{FE}}$ and $\hat{\mathbf{S}}_{\mathrm{FE}}$ are the feature encoder output representations of $s[n]$ and $\hat{s}[n]$, respectively. F and T denote the feature and time dimensions of the representations, with F being 512 for all

Table 8.1: *Performance of models trained on CommonVoice-DEMAND* English*; tested on English, Spanish and Welsh testsets.*

	Model	PESQ	STOI	CSIG	CBAK	COVL
English	*Noisy*	*2.19*	*0.95*	*3.27*	*2.40*	*2.67*
	Spec Loss	2.64	0.96	3.66	2.64	3.14
	STOI Loss	2.46	**0.96**	3.47	2.35	2.93
	SISDR Loss	2.73	0.96	3.45	2.74	3.07
	HuBERT L_{FE}	2.75	0.95	**3.78**	2.71	**3.25**
	mHuBERT L_{FE}	**2.79**	0.96	3.70	2.76	3.23
	XLSR L_{FE}	2.48	0.93	3.30	**2.92**	2.86
Spanish	*Noisy*	*2.12*	*0.95*	*2.98*	*2.23*	*2.46*
	Spec Loss	2.57	0.96	3.37	2.62	2.94
	STOI Loss	2.39	**0.96**	3.19	2.31	2.73
	SISDR Loss	2.68	0.96	3.15	2.74	2.88
	HuBERT L_{FE}	2.72	0.95	**3.57**	2.71	**3.11**
	mHuBERT L_{FE}	**2.75**	0.96	3.48	2.75	3.08
	XLSR L_{FE}	2.51	0.93	2.93	**2.81**	2.65
Welsh	*Noisy*	*2.12*	*0.96*	*3.06*	*2.18*	*2.50*
	Spec Loss	2.61	0.96	3.39	2.56	2.97
	STOI Loss	2.45	**0.97**	3.33	2.27	2.83
	SISDR Loss	2.72	0.97	3.15	2.69	2.89
	HuBERT L_{FE}	2.71	0.96	**3.56**	2.61	3.09
	mHuBERT L_{FE}	**2.78**	0.96	3.48	2.68	**3.09**
	XLSR L_{FE}	2.44	0.93	2.93	**2.73**	2.62

models used in this work and T depending on the length in samples of the time domain audio. Models are trained using HuBERT, mHuBERT, and XLSR to obtain the representations in (8.2). In addition, the spectrogram MSE loss, STOI (Taal et al., 2011) loss (Fu, Wang, et al., 2018), and SI-SDR (Roux et al., 2018) loss are used as baselines. These baselines are popular loss functions for speech enhancement training and are language-independent. Spectrograms are created using a STFT with a FFT length of 512, a window length of 32 ms, a hop length of 16 ms, and a hamming window.

8.3.4 Results

Tables 8.1 and 8.2 display results on CommonVoice-DEMAND testsets for models trained on English and Spanish, respectively. PESQ (Rix et al., 2001), STOI (Taal et al., 2011), and the components of the Composite (Lin et al., 2019) intrusive metrics are reported, where higher values are better. The scores for the best performing model on each testset are highlighted in bold. The models perform better on the respective testset matching their language of training; the drop in performance on the non-matching testsets is consistent across all the models trained, including the baseline systems. Performance on the proposed English CommonVoice-DEMAND dataset is similar to that of models trained on the original VoiceBank-DEMAND in (Close, Ravenscroft, et al., 2023b). The best performing models, in terms of CBAK score are those trained with XLSR L_{FE} loss, except for one case. This is again consistent with the findings in (Close, Ravenscroft, et al., 2023b). HuBERT and mHuBERT perform similarly, with mHuBERT slightly outperforming in most cases.

Table 8.2: *Performance of models trained on CommonVoice-DEMAND* Spanish*; tested on English, Spanish and Welsh testsets.*

	Model	PESQ	STOI	CSIG	CBAK	COVL
English	*Noisy*	*2.19*	*0.95*	*3.27*	*2.40*	*2.67*
	Spec Loss	2.50	0.95	3.54	2.59	3.00
	STOI Loss	2.37	**0.96**	3.32	2.35	2.80
	SISDR Loss	2.44	0.95	3.25	2.58	2.82
	HuBERT L_{FE}	2.60	0.95	3.58	2.62	3.06
	mHuBERT L_{FE}	**2.70**	0.95	**3.62**	2.69	**3.14**
	XLSR L_{FE}	2.44	0.93	3.13	**2.75**	2.72
Spanish	*Noisy*	*2.12*	*0.95*	*2.98*	*2.23*	*2.46*
	Spec Loss	2.64	0.96	3.50	2.68	3.04
	STOI Loss	2.40	**0.96**	3.11	2.36	2.70
	SISDR Loss	2.61	0.96	3.18	2.72	2.86
	HuBERT L_{FE}	2.81	0.96	**3.75**	2.76	3.25
	mHuBERT L_{FE}	**2.89**	0.96	3.73	2.81	**3.29**
	XLSR L_{FE}	2.63	0.95	3.06	**2.83**	2.77
Welsh	*Noisy*	*2.12*	*0.96*	*3.06*	*2.18*	*2.50*
	Spec Loss	2.63	0.96	3.49	2.59	3.03
	STOI Loss	2.42	**0.97**	3.22	2.29	2.77
	SISDR Loss	2.60	0.97	3.18	2.63	2.85
	HuBERT L_{FE}	2.72	0.96	3.62	2.62	3.13
	mHuBERT L_{FE}	**2.83**	0.96	**3.64**	**2.69**	**3.21**
	XLSR L_{FE}	2.52	0.94	2.94	2.64	2.65

Interestingly, all SSSR loss function models trained using Spanish audio perform better on the Welsh testset than those trained on the English audio. This is despite the fact that Welsh is lexically more similar to English than Spanish (Bella et al., 2021).

The quantity of data used for training the SSSR appears to be more important than language, as mHuBERT is trained with more English audio than HuBERT, however XLSR, trained with the most English speech data, performs worse. To further investigate this, an additional model was trained utilising the WavLM Base+ (Chen et al., 2022) SSSR(cf. Section 2.5.2.2), training and testing on the English CommonVoice-DEMAND dataset. WavLM Base+ has a parameter count comparable to HuBERT and mHuBERT and is trained with a similar objective and but with an additional speech denoising task. It is trained on 96k hours of English only audio. These results are shown in Table 8.3; the new model performs similarly in terms of PESQ score but somewhat better than all others in terms of CSIG.

Overall, these results suggest that the BERT style training objective HuBERT, mHuBERT and WavLM might make them better suited as loss function feature representations when signal quality is the main concern as shown by the high PESQ and CSIG scores while the contrastive feature encoding masking objective of XLSR makes it more suitable if the objective of the enhancement is background noise reduction at the cost of additional speech distortion as the higher CBAK scores of the XLSR models demonstrates.

Table 8.3: *Performance of L_{FE} Loss models trained on CommonVoice-DEMAND English and tested on English testset.*

Model	PESQ	STOI	CSIG	CBAK	COVL
Noisy	*2.19*	*0.95*	*3.27*	*2.40*	*2.67*
HuBERT L_{FE}	2.75	0.95	3.78	2.71	3.25
mHuBERT L_{FE}	**2.79**	0.96	3.70	2.76	3.23
XLSR L_{FE}	2.48	0.93	3.30	**2.92**	2.86
WavLM L_{FE}	2.76	**0.96**	**3.84**	2.71	**3.28**

8.4 Summary

In this chapter, a system to create noisy speech datasets for a number of languages are proposed. These noisy speech datasets are used to train and test neural speech enhancement models using SSSR based loss functions. It is found that the language of the audio used to train the representations has a minimal impact on their performance when used in this manner, and that training objective and amount of training data has a greater effect.

Part IV

Human Audio Label Prediction

Preface

This part is concerned with DNN models to predict human derived labels of speech quality and intelligibility. In Chapter 9 human SI and intelligibility metric prediction DNNs are proposed, in the context of an entry to Clarity Prediction Challenge 1 (CPC1) (cf. Section 2.10.1). In Chapter 10 features derived from SSSR (cf. Section 2.5.2) models are used as input feature representations for human intelligibility prediction. In Chapter 11 both SSSR features and those derived from the Whisper ASR system (cf. Section 2.5.3) are applied as input features to Speech Quality (SQ) prediction DNNs. Finally in Chapter 12 inference of a SQ predictor is used in the loss function of an NNSE system.

Chapter 9

Deep Neural Network based Intelligibility Prediction for Clarity Prediction Challenge 1 (CPC1)

9.1 Introduction

In the context of the CPC1 (Graetzer et al., 2021) (cf. Section 2.10.1) Speech Intelligibility (SI) is defined as the percentage of words that a listener correctly identifies after listening to a sequence of words.
SI prediction metrics are either *intrusive*, i.e. rely on access to the clean reference signal, or *non-intrusive*, i.e. rather than facilitating a comparative function non-intrusive metrics analyse only the degraded signal under test to identify key areas of potential distortion (Falk et al., 2015). There are 3 key domains of non-intrusive SI; feature-based approaches using key acoustic features and potentially other linguistic information for prediction, statistical data-driven methods such as machine learning, and neurophysiological measures that integrate neuroimaging or oculometric techniques (Feng & Chen, 2022). This chapter aims to use both non-intrusive and intrusive methods for predicting intrusive SI metrics as outlined in Section 2.7.

9.2 Neural Intelligibility Prediction

Inspired by the metric predictor Discriminator DNN in MetricGAN (cf. Section 2.12, Part II) which use a neural network to mimic the performance of an intrusive metric for speech quality and intelligibility, this contribution uses a similar network structure to predict the metric score that will be assigned to the input audio. Note that here networks that are provided with representations of both the degraded and reference signal (intrusive) and also with those that are only provided with the degraded (non-intrusive) are investigated.
The focus is on a metric prediction objective over simply using the ground truth 'correctness' information in the training data as this was found to be distributed in a way that was difficult for our non-intrusive models to find any discernible patterns in. Intuition is that if these metrics have been found to correlate with human intelligibility, then non-intrusive predictors of said metrics should

also. Additionally, the performance of each of our non-intrusive metric predictors after being fine-tuned on the ground truth intelligibility is reported.
Figure 9.1 provides a generalised overview demonstrating the training of such a neural network.

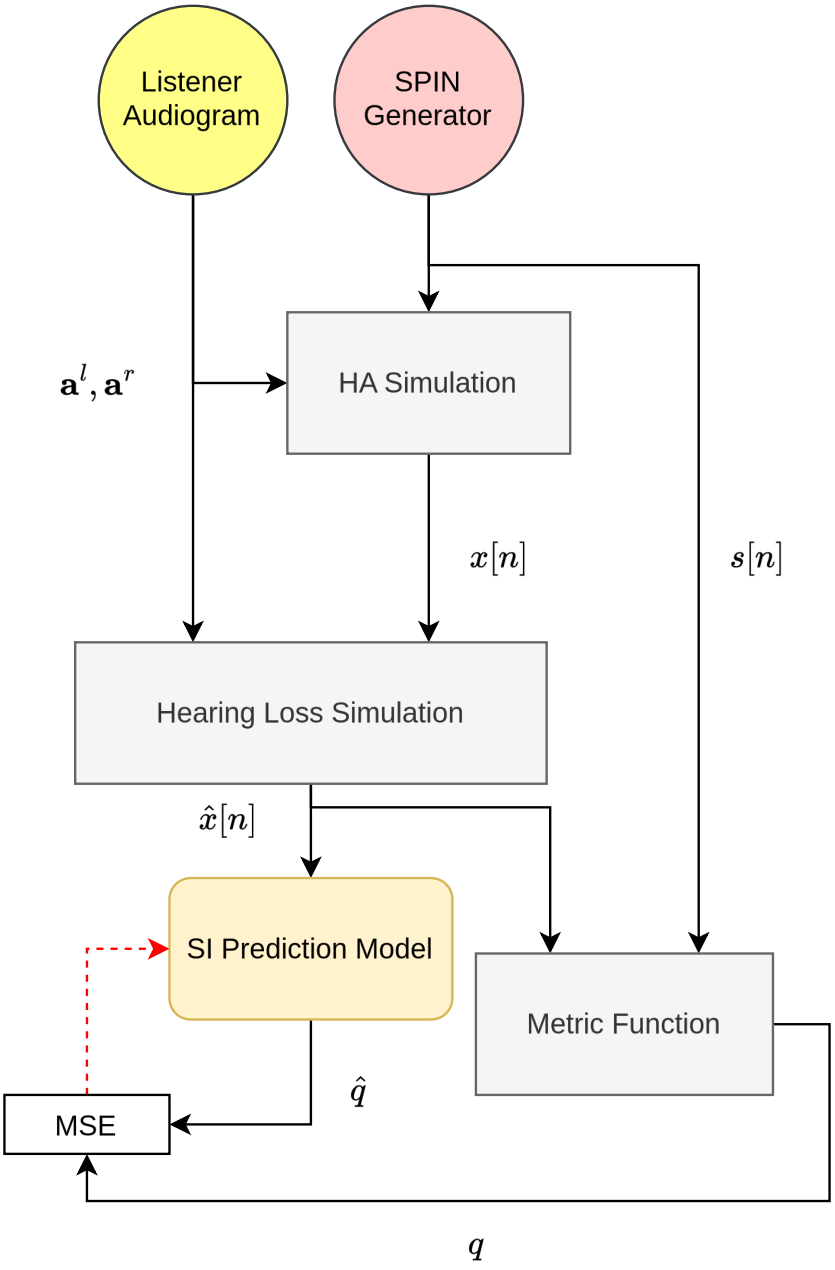

Figure 9.1: *Diagram of general non-intrusive SI metric prediction training*

Here, noisy audio $x[n]$ is generated by a SPeech In Noise (SPIN) generator and processed by a hearing aid (HA) simulation then a HL simulation which both take a representation of the specific listener's HL as input. This takes the form of an audiogram pair $\{\mathbf{a}^l, \mathbf{a}^r\}$ which represent the specific characteristics of their HL for the left and right ears respectively. Details on the HL model used in the CPC1 baseline can be found in (Nejime & Moore, 1997). The output of this $\hat{x}[n]$ is input

to a SI prediction model, along with the clean reference audio $s[n]$. The output of this prediction model $\hat{Q}$ is compared to the true value of the SI Q i.e the Human Speech Recognition (HSR), and the model is updated.

9.2.1 Feature Extraction

The same feature extraction as described in (Close, Hain, et al., 2022) is used here with the discrete time domain input audio being transformed to normalised log features. Note that in the following $\mathbf{X}_f^l$, $\mathbf{X}_f^r$ denotes the feature representation of the hearing aid output while $\hat{\mathbf{X}}_f^l$,$\hat{\mathbf{X}}_f^r$ is the feature representation of the hearing aid output x with the baseline HL applied $\hat{x}[n]$.

9.2.2 Model Structure for Non-Intrusive Prediction

For each of the 3 metrics investigated, the same basic model structure is adapted for the specific requirements of the metric. The basic structure is based on that of the discriminator network depicted in Figure 2.34 - 4 2D convolutional layers with 15 filters of a kernel size of (5, 5), followed by a mean over the 2nd and 3rd dimensions, and this representation is fed into 3 sequential linear layers, with 50, 10, and 1 output neuron(s) respectively. The first 2 of these layers have a LeakyReLU activation while the final layer has no activation. For STOI, the score for each channel of the HA output audio $\hat{x}$ is predicted separately, with the input to the prediction network being the feature space representation of the given channel $\hat{\mathbf{X}}_f^c$ where c is a channel index. As such, the input dimension to the average pooling and first 2D convolutional layer is set to 1. For MBSTOI, the score is predicted for the HA output stereo audio together, with the input to the network being the feature space representations of both channels $\{\hat{\mathbf{X}}_f^l, \hat{\mathbf{X}}_f^r\}$. The input dimension of the average pooling layer and the initial convolutional layer is 2 to account for these stacked channel representations. Finally for HASPI which like STOI is defined per audio channel, the $\mathbf{X}_f^l$ and $\mathbf{X}_f^r$ representation of the audio, but also use $\mathbf{a}^l$ and $\mathbf{a}^r$ the audiogram representations of the listener's HL is used as input. This 6 element representation is passed through a linear layer with 10 output neurons then another with 50; this representation is then concatenated along the feature dimension with the representation of the audio of the same size. This 100 element representation is then fed through a further 3 linear layers with 50, 10, and 1 output node(s) respectively, all but the last layer having a LeakyReLU activation. Additionally,a model is trained with the same structure as that for the HASPI prediction described above, using $\{\hat{\mathbf{X}}_f^l, \hat{\mathbf{X}}_f^r\}$ as input and train it to predict the ground truth Correctness scores in the training data.

9.2.3 Model Struture for Intrusive Prediction

Additionally intrusive versions of the metric prediction models are trained and fine-tuned. These are similar to those above except we also input the clean reference features $\mathbf{S}_f^c$ to the model. For the STOI and HASPI prediction models the clean and degraded features are stacked per channel, $\{\mathbf{S}_f^l, \hat{\mathbf{X}}_f^l\}$, $\{\mathbf{S}_f^r, \hat{\mathbf{X}}_f^r\}$ for STOI and $\{\mathbf{S}_f^l, \mathbf{X}_f^l\}$,$\{\mathbf{S}_f^r, \mathbf{X}_f^r\}$ along with $\mathbf{a}^l$, $\mathbf{a}^r$ for HASPI. For MBSTOI we use both channels of the clean and degraded features, $\{\hat{\mathbf{X}}_f^l, \mathbf{S}_f^l, \hat{\mathbf{X}}_f^r, \mathbf{S}_f^r\}$.

9.3 Experiments

9.3.1 Tools and Software

Experiments are implemented via modifications to the CPC1 baseline system, replacing the simple fitting model with the neural models described above using PyTorch (Paszke et al., 2019). The SpeechBrain (Ravanelli et al., 2021) framework is used for audio loading and dataloader creation. Existing Python and MATLAB implementations are used for STOI MBSTOI (taken from CPC1 baseline) and HASPI

9.3.2 Data Description

Audio data provided by the CPC1 is used for the hearing aid outputs x, the hearing aid outputs processed by the baseline HL simulation $\hat{x}$ and the anechoic clean reference signal s , accompanied by ground truth correctness scores Q_h and listeners' audiograms $\{\mathbf{a}^l, \mathbf{a}^r\}$ for left and right ear, respectively. In total the challenge corpus provides 4863 training examples expressed as combinations of 'scenes' ($s[n]$, $x[n]$), listener HL characteristics ($\mathbf{a}^l, \mathbf{a}^r$), HL simulations $\hat{x}[n]$ and correctness scores Q_h. The spoken sentence are taken from the Clarity speech corpus (Graetzer et al., 2022). For more detail on the CPC1 corpus see Section 2.10.1.

9.4 CPC Metric Distributions

The upper plot in Figure 9.3 shows the distribution of correctness i in the CPC1 training set. From this, it can be observed that in the majority of cases, the listener was able to fully reproduce the speech in the audio they heard, i.e. $i = 100$ for $\approx 50\%$ of the assessed files. The next largest class is where $i = 0$, meaning that the listener was not able to reproduce any words in the audio. This distribution is due to the more realistic *in-the-wild* SI measurement strategy for the Clarity dataset (Barker et al., 2022) which is in contrast to lab-based SI matrix tests (Kollmeier et al., 2015). The lower panel of Figure 9.3 shows the average correctness i for each listener in the CPC1 training set. With the exception of listener L0227, all of the listeners achieve similar performance.

The Spearman r and Pearson ρ correlations between SI metrics and the ground truth correctness i

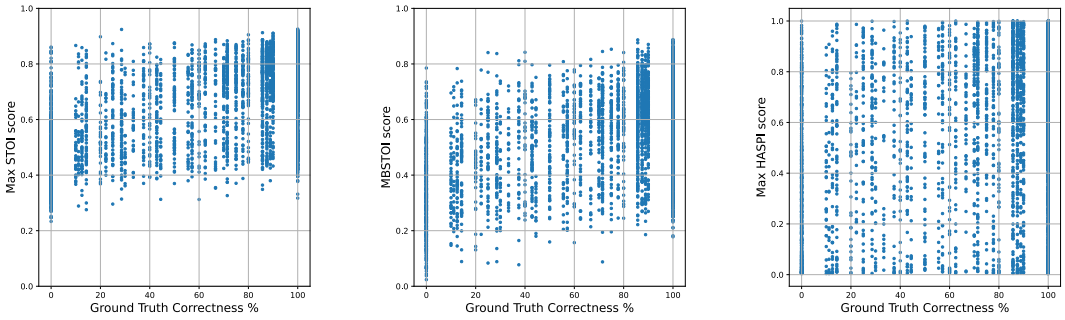

Figure 9.2: *SI metrics versus ground truth correctness percentage in CPC 1 Training Set*

are presented in Table 9.1 and the relationships are visualised in Figure Figure 9.2.

Metric	STOI	MBSTOI	HASPI
r	0.65	0.61	0.34
ρ	0.57	0.54	0.31

Table 9.1: *Spearman r and Pearson ρ Correlation between SI metrics and correctness label i in CPC1 Training set*

These both show that, while correlation is low for all three metrics, STOI and MBSTOI correlate somewhat more strongly with the data compared with HASPI - this is interesting, especially given that HASPI is the one metric of the 3 which has explicit access to the audiogram information. One possible explanation is that STOI and MBSTOI are computed using $\hat{s}'[n]$ while HASPI uses $s[n]$ as it contains its own internal HL simulation; it is possible that this internal model produces outputs which differ greatly from that of the baseline system.

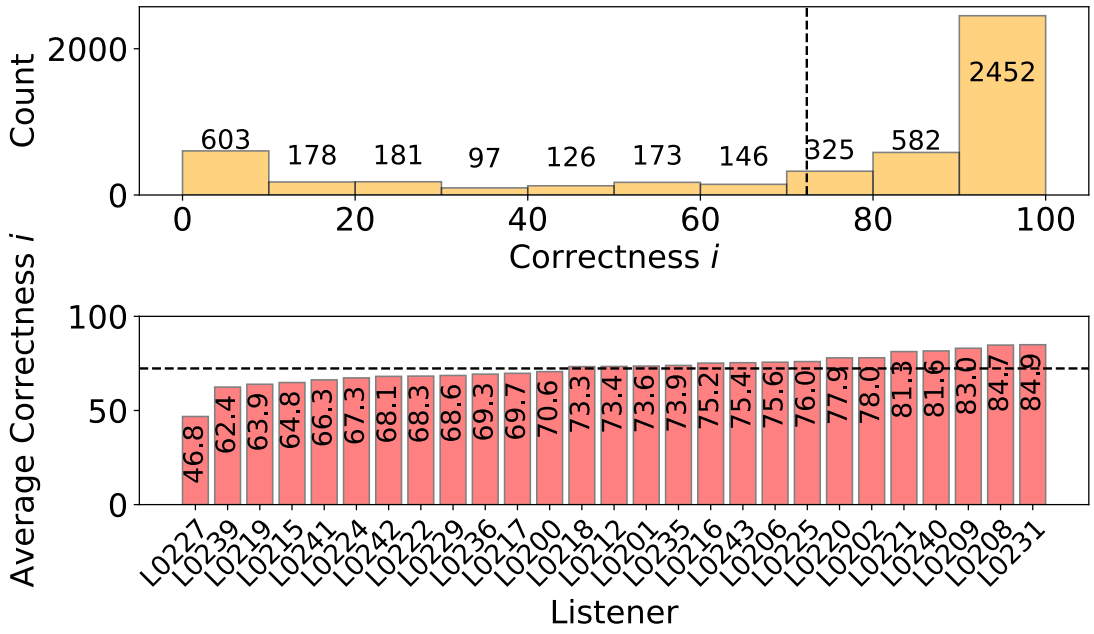

Figure 9.3: *Histogram showing the distribution of ground truth correctness i in CPC1 training set (top) and a bar chart showing average correctness i per listener in the CPC1 training set (bottom). Dotted lines are respective overall average values.*

9.5 Experiment Setup

STOI, MBSTOI, and HASPI scores for the entire train set are pre-computed. Then the models are trained as described above, to reproduce the score. The feature extraction use a STFT with a window length of 20ms, a hop length of 10ms and an FFT size of 1024. The hearing aid outputs x have a sampling rate of 32kHz, while the hearing aid outputs with the baseline HL simulation applied $\hat{x}$ have a sampling rate of 44kHz. Following on from the baseline system we train with a 5 fold validation technique, partitioning the folds on the scene ID. We use the Adam (Kingma & Ba,

2014) Optimiser with a learning rate of 0.001 for all models. All models are trained with a batch size of 1 with the exception of the model that directly predicts Correctness which uses a batch size of 20. The metric prediction models are additionally fine-tuned using the ground truth HSR 'Correctness' (intelligibility) scores; in the case of the metrics that are defined per channel (STOI and HASPI) the channel that returned the highest predicted score between the 2 is used, as a simplified simulation of the 'better ear effect'. This fine-tuning process consists of exposing the model to the entire training set in the same way as in the pre-training, but having it's outputs compared to the ground truth rather than the metric. The same technique is used to evaluate the model performance.

9.5.1 Results

Table 9.2 shows the results for non-intrusive prediction over the entire training set for the challenge. The upper half shows the RMSE betweenmodel outputand ground truth 'correctness' values, i.e. HSR. The lower half shows the RMSE between target metric and prediction of the model. r and ρ are the Spearman and Pearson Correlations, respectively.

In terms of prediction error, the model showing best non-intrusive target metric prediction is

Table 9.2: *Non Intrusive Performance on the Clarity Prediction Challenge Training Set*

Model Objective	**Correctness Error**	r	ρ
STOI	35.63	0.30	0.21
STOI (fine)	34.55	0.32	0.25
MBSTOI	39.30	0.26	0.18
MBSTOI (fine)	34.72	0.32	0.23
HASPI	38.80	0.23	0.22
HASPI (fine)	**31.55**	**0.53**	**0.46**
Correctness	33.44	0.45	0.42
	Prediction Error	r	ρ
STOI	**13.88**	0.43	0.3
STOI (fine)	16.44	0.43	0.3
MBSTOI	15.50	0.44	0.33
MBSTOI (fine)	21.81	0.47	0.32
HASPI	25.10	**0.59**	**0.59**
HASPI (fine)	37.09	0.29	0.29

the STOI prediction model, while the the HASPI model shows lowest performance. This is likely because the calculation of STOI is considerably simpler than that for HASPI. As expected, fine-tuning to the ground truth correctness increases prediction error while decreasing correctness error for all models.

Best model in terms of prediction of ground truth correctness is the fine-tuned HASPI predictor. This is interesting given that HASPI itself has the lowest correlation with the ground truth correctness in the data - it is possible that access to the audiogram information is what enables this. The slight performance improvement versus the model that was only trained to predict the correctness shows that the HASPI objective pre-training did improve performance.

Table 9.3 shows the results of the intrusive prediction models, along with that of the challenge baseline system. The prediction error results follow the same pattern as those of the non-intrusive models, but with lower overall error rates and significantly higher correlations.

Both the fine-tuned STOI and MBSTOI models slightly outperformed the baseline system in

Table 9.3: *Intrusive Performance on the Clarity Prediction Challenge Training Set*

Model Objective	**Correctness Error**	r	ρ
baseline	*28.5*	*0.62*	*0.54*
STOI	32.45	0.58	0.52
STOI (fine)	27.59	0.66	0.56
MBSTOI	29.67	0.65	0.54
MBSTOI (fine)	**27.20**	**0.67**	**0.58**
HASPI	41.04	0.27	0.25
HASPI (fine)	29.67	0.65	0.54
Correctness	35.62	0.31	0.27
	Prediction Error	r	ρ
STOI	**9.05**	**0.86**	**0.83**
STOI (fine)	16.24	0.75	0.70
MBSTOI	10.79	0.79	0.80
MBSTOI (fine)	22.64	0.73	0.7
HASPI	23.06	0.68	0.68
HASPI (fine)	29.11	0.43	0.43

Table 9.4: *Non Intrusive Performance on the Clarity Prediction Challenge Test Set*

Model Objective	**Correctness Error**	r	ρ
HASPI (fine)	**31.99**	**0.43**	**0.50**
Correctness	33.42	0.42	0.39

terms of correctness error and correlations.Interestingly, of the two models that directly predict the Correctness values Q, the non-intrusive model slightly outperforms the intrusive one.
Table 9.4 shows the performance on the test set of the two non-intrusive models submitted to the challenge. The pretrained HASPI model performs slightly better overall compared to the direct Correctness model.

9.6 Summary

In this chapter, the use of DNN for the task of SI prediction was investigated. Of the models trained, it was found that intrusive models outperform non intrusive models for both metric prediction and for real intelligibility prediction. An intrusive neural model outperforms the intrusive baseline system for the challenge. Furthermore, pre-training models to predict an intelligibility metric, and then fine-tuning on the true intelligibility improves performance. Additionally, the relationship between the real intelligibility scores in the data and signal processing based intrusive metrics was examined, and it was found that these are only weakly correlated.

Chapter 10

Non-Intrusive SI Prediction using SSSR Features

10.1 Introduction

In this chapter non-intrusive SI prediction models which make use of SSSR features are proposed. The proposed models show improved performance over those which use a traditional STFT based feature. Further, the relationship between SSSR based distance measures and real human SI scores is explored. It is found that non-intrusive SI predictors tend to learn the overall characteristics of the hearing aid enhancement system used to process the input audio prior to evaluation.

10.2 SSSRs for Metric Prediction

SSSRs have been applied to metric prediction tasks, typically to quality prediction (Cooper et al., 2022; Mittag et al., 2021). In (Tamm et al., 2022), XLSR representations are used as feature extraction in a non-intrusive human MOS prediction network.

Similarly, in (Becerra et al., 2022) SSSRs are used for the same quality prediction task, but they are fine-tuned with a mean pooling layer rather than being used simply as feature extraction. SSSRs were also applied to the CPC1 challenge in (Edo Zezario et al., 2022), where they were used as feature extractors alongside spectrograms and learnable filter banks.
In all these cases, only the final SSSR output $\mathcal{G}_{\mathrm{OL}}$ was considered. However, findings in Chapter 7 suggest that the output of the initial encoding stage $\mathcal{G}_{\mathrm{FE}}$ better captures quality-related information. As such, in this chapter, both representations stages are considered and compared as feature transformations for SI prediction.

10.3 Analysing Relationships between SSSRs and Human SI

In order to express the relationship between SSSRs and correctness i in the dataset, two distance measures are defined in a MSE sense:

$$d_{\mathrm{FE}} = \frac{1}{TF} \sum_{t}^{T} \sum_{f}^{F} (\mathbf{S}_{\mathrm{FE}}[t, f] - \mathbf{P}_{\mathrm{FE}}[t, f])^2 \tag{10.1}$$

$$d_{\mathrm{OL}} = \frac{1}{TF}\sum_{t}^{T}\sum_{f}^{F}(\mathbf{S}_{\mathrm{OL}}[t,f] - \mathbf{P}_{\mathrm{OL}}[t,f])^2 \tag{10.2}$$

The distance d_{FE} in (Eq. 10.1) expresses the MSE distance between the SSSR *feature encoding* layer representations $\mathbf{S}_{\mathrm{FE}}[t,f]$ of the clean reference audio $\mathbf{s}[n]$ and the representations $\mathbf{P}_{\mathrm{FE}}[t,f]$ of the test signal $\mathbf{p}[n]$, while (Eq. 10.2) expresses the MSE distance between the SSSR *output layer* representations $\mathbf{S}_{\mathrm{OL}}[t,f]$ and $\mathbf{P}_{\mathrm{OL}}[t,f]$, with t and f denoting block time and feature index, respectively. Note that $\mathbf{p}[n]$ and is a placeholder for either the speech signal after Hearing Aid (HA) enhancement $\hat{\mathbf{s}}[n]$ or this signal after HLS processing $\hat{\mathbf{s}}'[n]$ as shown in Figure 2.30. Distances in (Eq. 10.1), (Eq. 10.2) are designed to express the distortion captured by the SSSR due to the transformations which have been applied to $\mathbf{s}[n]$ to produce e.g. $\hat{\mathbf{s}}'[n]$, i.e. the artificial distortion/reverb added to create $\mathbf{x}[n]$, enhancement by the hearing aid system (in $\hat{\mathbf{s}}[n]$) and finally the HLS. In addition to distances (Eq. 10.1) and (Eq. 10.2) the MSE distance between spectrogram representations of $\mathbf{s}[n]$ and $\mathbf{p}[n]$ Eq. 2.39 will be analysed, with f_{Hz} and F_{Hz} denoting the technical frequency and the highest frequency analysed, respectively. In the following, the left (first) channel of the audio is used to compute the distance measures (Eq. 10.1), (Eq. 10.2) and (Eq. 2.39).

Representation, Distance	$\mathbf{p}[n]$	**Spearman**	**Pearson**
SPEC, d_{SG},	$\hat{\mathbf{s}}[n]$	-0.10	-0.18
SPEC, d_{SG},	$\hat{\mathbf{s}}'[n]$	-0.09	-0.07
XLSR, d_{FE}	$\hat{\mathbf{s}}[n]$	-0.13	-0.16
XLSR, d_{FE}	$\hat{\mathbf{s}}'[n]$	-0.24	-0.28
XLSR, d_{OL}	$\hat{\mathbf{s}}[n]$	-0.26	-0.27
XLSR, d_{OL}	$\hat{\mathbf{s}}'[n]$	-0.24	-0.24
HuBERT, d_{FE}	$\hat{\mathbf{s}}[n]$	$\mathbf{-0.38}$	$\mathbf{-0.47}$
HuBERT, d_{FE}	$\hat{\mathbf{s}}'[n]$	-0.23	-0.29
HuBERT, d_{OL}	$\hat{\mathbf{s}}[n]$	-0.10	-0.17
HuBERT, d_{OL}	$\hat{\mathbf{s}}'[n]$	-0.28	-0.32

Table 10.1: *Spearman and Pearson correlations between distance measures and correctness values i in the CPC1 training set, strongest correlations in bold.*

Table 10.1 shows the Spearman and Pearson correlations of the MSE distances with the correctness values i for the CPC1 training set. Absolute correlations are low, but this is expected for the Clarity dataset (cf. (Barker et al., 2022) and Section 2.10.1). Comparing the distances between the feature representations and the intelligibility scores i allows for an expression of how distortion in the signal, which might affect intelligibility, is captured by that feature representation. Interestingly, applying the hearing loss simulation $\mathcal{S}$ in (Eq. 2.56) does not uniformly improve the correlation with i across all distances in Table 10.1; only for the XLSR encoder output representation distance d_{FE} and the HuBERT final output representation distance d_{OL} does using $\hat{\mathbf{s}}'[n]$ lead to higher correlation than using $\hat{\mathbf{s}}[n]$. This might suggest that the hearing loss of the listeners did not significantly affect their ability to reproduce the prompt in the audio., or that the hearing loss simulation is not effective.
Figure 10.1 and Figure 10.2 visualise the correlations between the distances based on $\hat{\mathbf{s}}[n]$ and $\hat{\mathbf{s}}'[n]$ with and correctness value i, as well as the relationship between the $\hat{\mathbf{s}}[n]$, $\hat{\mathbf{s}}'[n]$ values. These show that the effect of the hearing loss simulation on the SSSR distance measures varies greatly between the two SSSR representations. From the two $\mathcal{G}_{\mathrm{MSE}}$ plots, it can be observed that the absolute magnitude of the XLSR distances are significantly greater than those of HuBERT. In both cases, the application of the hearing loss simulation in (Eq. 2.56) had only a small impact on the distribution.

The $\mathcal{G}_{OL_{\text{MSE}}}$ plots follow a similar pattern but inverted in terms of the difference in magnitude of the distances, with the HuBERT distances being generally larger than the XLSR. However, the application of the hearing loss simulation in (Eq. 2.56) has a drastic effect on the distribution for HuBERT, shifting all of the points upward.

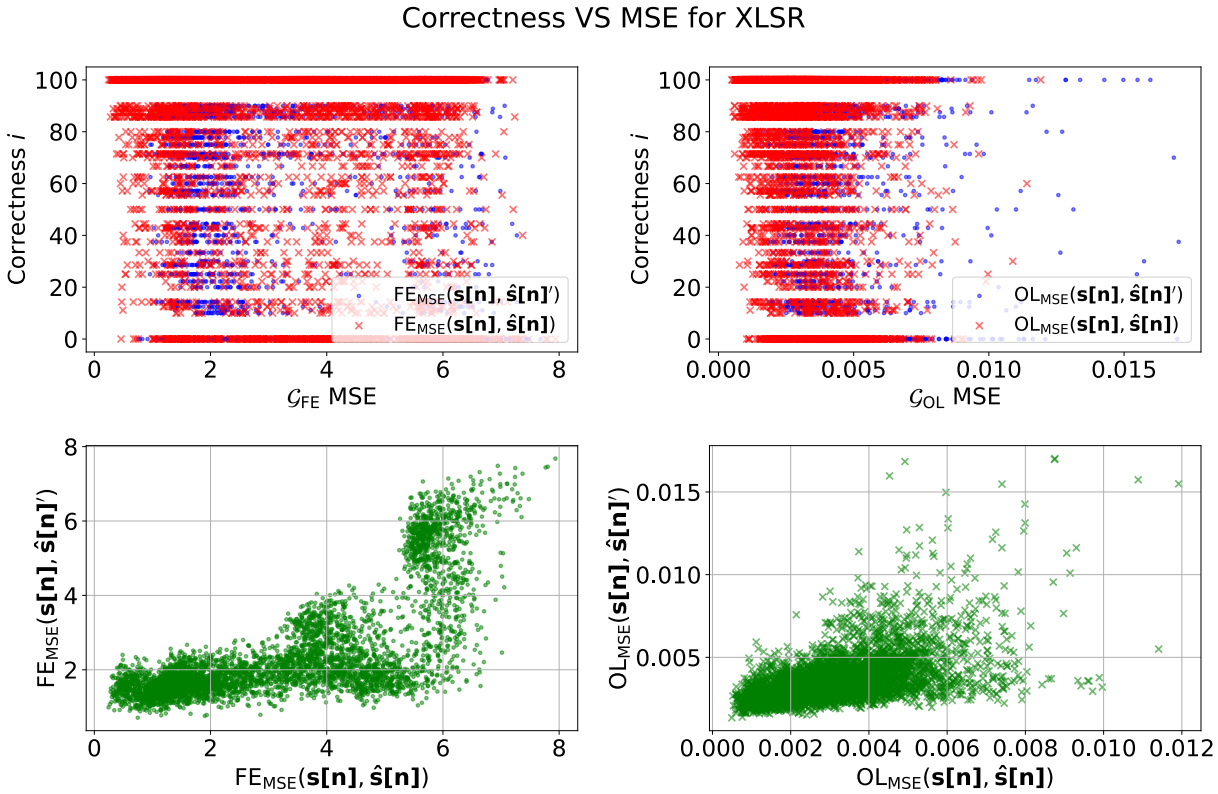

Figure 10.1: *Scatter plots showing the correlation between distances d_{FE} (Eq. 10.1) for XLSR features and d_{OL} (Eq. 10.2) Correctness i for the CPC1 training set (upper panels). Scatter plots showing the relationship between the $\hat{\mathbf{s}}'[n]$ and $\hat{\mathbf{s}}[n]$ distances (lower)*

10.4 SSSR-based Intelligibility Prediction

This section proposes the use of SSSRs as features in non-intrusive neural intelligibility prediction networks. Following the findings from Table 10.1, both the hearing aid output signal $\hat{\mathbf{s}}[n]$ and that signal processed by the hearing loss simulation $\hat{\mathbf{s}}'[n]$ are used as the input audio to the models, as no conclusive best representation is indicated by these results.

10.4.1 Model Structure and Experiment Setup

A model structure inspired by (Tamm et al., 2022) is chosen for the SI prediction network. Five feature extraction methods are used; outputs of $\mathcal{G}_{\text{FE}}$ and $\mathcal{G}_{\text{OL}}$ for both, HuBERT and XLSR representations, as well as a spectrogram representation denoted as SPEC. After the feature

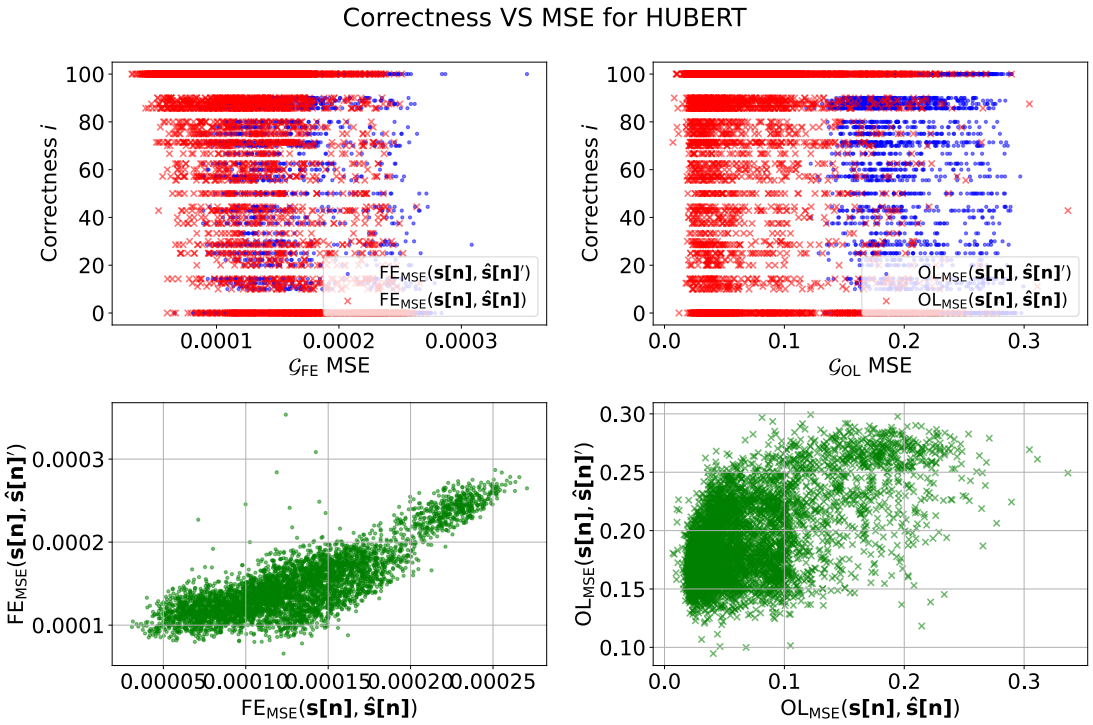

Figure 10.2: *Scatter plots showing the correlation between HuBERT d_{FE} and d_{OL} and Correctness i in the CPC1 training set (upper).Scatter plots showing the relationship between the $\hat{\mathbf{s}}'[n]$ and $\hat{\mathbf{s}}[n]$ distances (lower)*

extraction, the resultant representation is processed by 2 BLSTM layers with an input size equal to the feature dimension F of the input and a hidden layer size of $F/2$. The final layer is an attention pooling feed-forward layer, similar to that in NISQA (Mittag et al., 2021) with a single output neuron and a sigmoid activation to output the predicted correctness $\hat{i}$ (normalised between 0 and 1). Note that due to different dimensions F of different feature representations, the number of parameters in each network varies from $923,906$ for the models using spectrogram representations to as many as $14,701,570$ for the models using the XLSR output layer, i.e. $\mathcal{G}_{\text{OL}}$.

The two input audio representations $\hat{\mathbf{s}}[n]$ or $\hat{\mathbf{s}}'[n]$ are used, i.e. the output of the hearing aid systems and the enhanced audio processed by the hearing loss simulation, as in (Eq. 2.56). As these audio representations have two channels, each channel is processed by the model separately; during training, the loss for each channel is computed and then summed before being back-propagated to the model. During validation and testing, the maximum value between each channel is taken as an approximation of the *better ear effect* (Zurek & Studebaker, 1993).

The spectrogram representation is created by a STFT with a window length of 20 ms, a hop length of 10 ms and an FFT size of 1024. All audio is re-sampled to 16 kHz such that it can be used as inputs to the SSSR models.

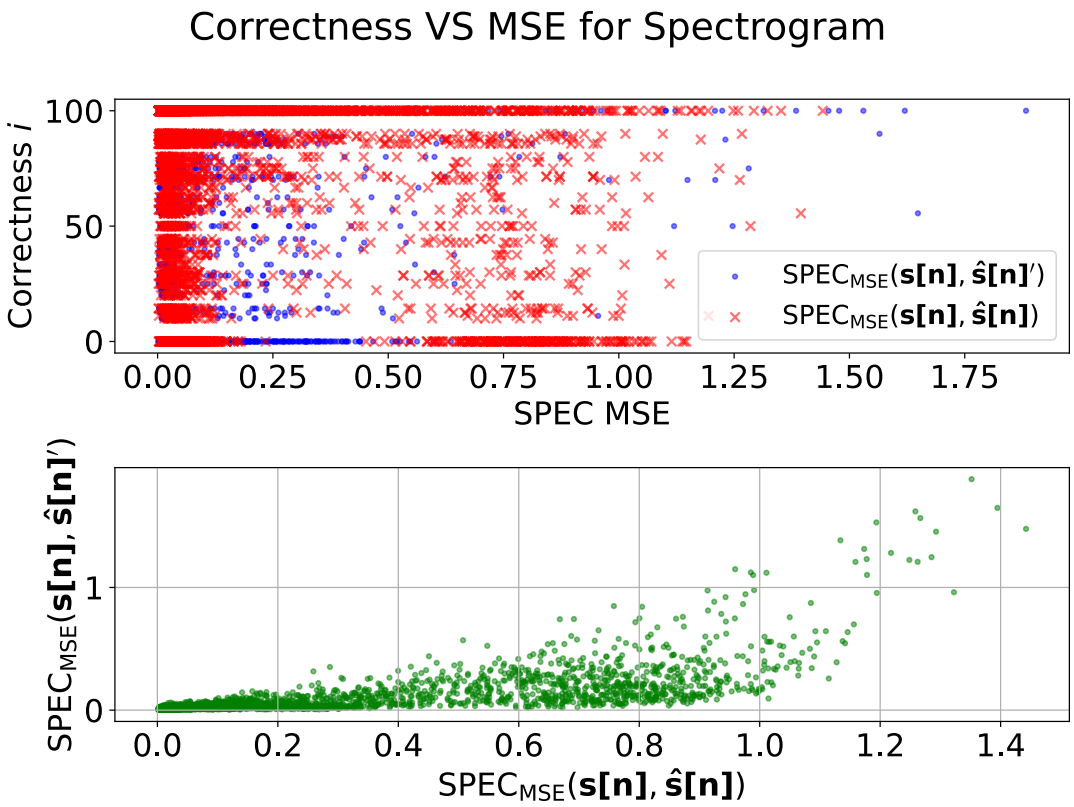

Figure 10.3: *Scatter plot showing the correlation between d_{SG} and Correctness i in the CPC1 training set*

10.5 Results

In addition to the intrusive (reference-signal-based) challenge baseline, the best-performing non-intrusive entries to the challenge are reported in this section as additional baselines, as the proposed system is also non intrusive. Challenge entry E23 (McKinney & Cauchi, 2022) makes use of contrastive predictive coding and vector quantisation features. E06 (Close, Hollands, et al., 2022) is similar to the proposed system, denoted by SPEC in the following, but uses a CNN based network structure. E33 (Edo Zezario et al., 2022) also utilises SSSRs as feature extraction, but spectrogram and learnable filterbank features are also used as model inputs. E29 (Tu et al., 2022) makes use of an information-theory-inspired approach, wherein the difference between internal representations in neural ASR systems is used to approximate human intelligibility, and was the overall best non-intrusive challenge entry.

10.5.1 Results on CPC1 Closed set

Table 10.2 shows the performance of the proposed systems for the CPC1 closed set. All proposed systems show comparable performance with the best-performing challenge entries, although, none of the proposed systems outperforms system E29. It should be noted, however, that the computation overhead to implement system E29 is significantly greater than any of the proposed systems here,

Model Name	RMSE	Var	Spearman	Pearson
CPC1 Baseline	*28.50*	–	*0.62*	–
E23 (McKinney & Cauchi, 2022)	*41.50*	–	*0.07*	–
E06 (Chapter 9)	*32.00*	–	*0.43*	–
E33 (Edo Zezario et al., 2022)	*24.10*	–	*0.75*	–
E29 (Tu et al., 2022)	***23.30***	–	***0.77***	–
SPEC $\hat{\mathbf{S}}_{\text{SPEC}}$	25.45	0.52	0.59	0.72
SPEC $\hat{\mathbf{S}}'_{\text{SPEC}}$	25.45	0.52	0.58	0.72
HuBERT $\hat{\mathbf{S}}_{\text{FE}}$	30.82	0.61	0.44	0.56
HuBERT $\hat{\mathbf{S}}'_{\text{FE}}$	26.64	0.53	0.56	0.70
HuBERT $\hat{\mathbf{S}}_{\text{OL}}$	**24.76**	**0.50**	0.59	**0.74**
HuBERT $\hat{\mathbf{S}}'_{\text{OL}}$	24.82	**0.50**	**0.61**	**0.74**
XLSR $\hat{\mathbf{S}}_{\text{FE}}$	25.01	**0.50**	0.60	**0.74**
XLSR $\hat{\mathbf{S}}'_{\text{FE}}$	25.33	0.51	0.60	0.72
XLSR $\hat{\mathbf{S}}_{\text{OL}}$	28.42	0.58	0.47	0.66
XLSR $\hat{\mathbf{S}}'_{\text{OL}}$	30.20	0.61	0.52	0.64

Table 10.2: *Non-Intrusive Prediction Performance on the CPC1 closed set. Best performances for baselines and proposed methods in boldface font.*

as several state-of-the-art ASR systems must be trained and fine-tuned for E29. Of the proposed systems trained on the outputs of the hearing loss simulation $\hat{\mathbf{s}}'[n]$, the best performing is the model which uses HuBERT output representations $\hat{\mathbf{S}}'_{\text{OL}}$ as features. This is consistent with the findings in Table 10.1 which shows that the distance measure using this representation had the highest correlation with i of those distances computed using $\hat{\mathbf{s}}'[n]$. Of those trained using the hearing aid outputs $\hat{\mathbf{s}}[n]$, HuBERT's output $\hat{\mathbf{S}}_{\text{OL}}$ is also the best performing achieving near identical performance to the $\hat{\mathbf{s}}'[n]$ model. In terms of the difference in performance between using earlier SSSR representations $\mathcal{G}_{\text{FE}}$ or output representations $\mathcal{G}_{\text{OL}}$ as features, this seems to depend on the SSSR used; for HuBERT the output layers perform best, while for XLSR the feature encoder layers show better performance.

10.5.2 Results of CPC1 Open set

Table 10.3 shows the performance of the proposed systems on the more challenging CPC1 open set (cf. Section 2.10.1). Performance of the proposed systems is significantly worse than that of the closed set for all systems, with a much larger variance in MSE in all cases, but all proposed systems still outperform the baseline. The poorer performance might be due to overfitting of the models to the training data, (in particular to the enhancement systems in the training set) as the test data contains unseen enhancement systems and listeners. All of the models here perform similarly poorly.

10.5.3 System and Listener-wise Analysis

For further analysis, Figure 10.4 and Figure 10.5 show ground truth and predicted correctness across the hearing aid systems and across the listeners in the CPC1 open testset for the HuBERT $\hat{\mathbf{S}}_{\text{OL}}$ and HuBERT $\hat{\mathbf{S}}'_{\text{OL}}$ models, respectively. Both models show similar performance across the

Model Name	RMSE	Var	Spearman	Pearson
CPC 1 Baseline	*36.50*	–	*0.53*	–
E23 (McKinney & Cauchi, 2022)	*43.70*	–	*0.05*	–
E33 (Edo Zezario et al., 2022)	*28.9*	–	*0.65*	–
E29 (Tu et al., 2022)	***24.60***	–	***0.73***	–
SPEC $\hat{\mathbf{S}}_{\text{SPEC}}$	32.84	1.29	0.35	0.50
SPEC $\hat{\mathbf{S}}'_{\text{SPEC}}$	29.16	1.15	0.57	0.60
HuBERT $\hat{\mathbf{S}}_{\text{FE}}$	33.69	1.30	0.27	0.45
HuBERT $\hat{\mathbf{S}}'_{\text{FE}}$	35.31	1.40	0.19	0.24
HuBERT $\hat{\mathbf{S}}_{\text{OL}}$	32.43	1.22	0.47	0.54
HuBERT $\hat{\mathbf{S}}'_{\text{OL}}$	29.66	1.14	0.60	0.61
XLSR $\hat{\mathbf{S}}_{\text{FE}}$	31.83	1.26	0.49	0.52
XLSR $\hat{\mathbf{S}}'_{\text{FE}}$	30.86	1.19	0.56	0.56
XLSR $\hat{\mathbf{S}}'_{\text{OL}}$	31.85	1.25	0.42	0.49
XLSR $\hat{\mathbf{S}}'_{\text{OL}}$	34.54	1.36	0.26	0.37

Table 10.3: *Non Intrusive Prediction Performance on the CPC1 open set.*

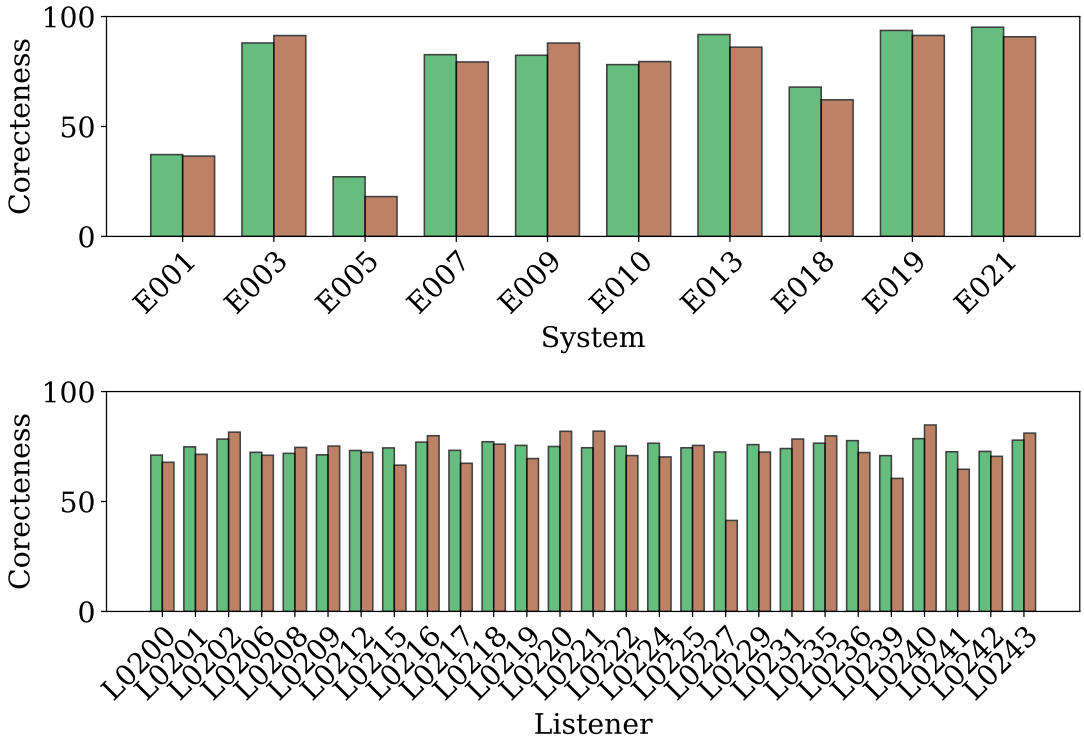

Figure 10.4: *System (top) and listener-wise (bottom) correctness prediction $\hat{i}$ (l./green) vs. true i (r./brown) using HuBERT output for $\hat{\mathbf{S}}_{\text{OL}}$ model on CPC1 closed set.*

different hearing aid systems, both successfully assigning low scores to the audio enhanced by the E005 hearing aid system. This indicates that the models are able (at some level) to detect the distortions introduced by this enhancement. Similarly, there is little difference in performance

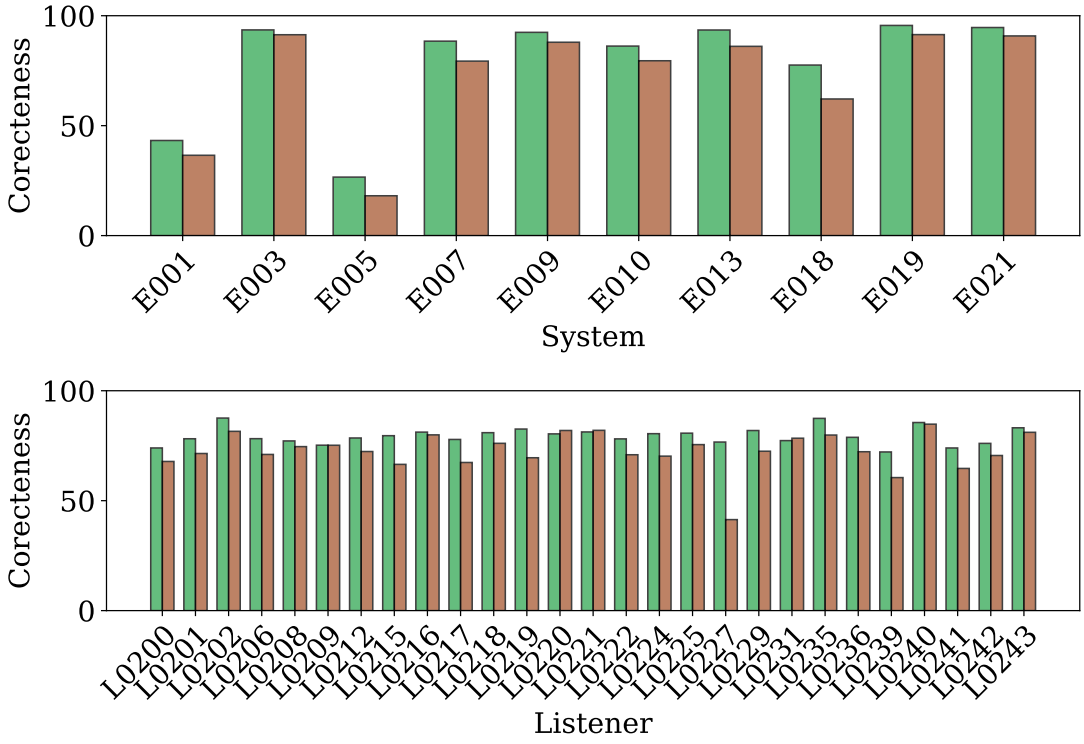

Figure 10.5: *System (top) and listener-wise (bottom) correctness prediction $\hat{i}$ (l./green) vs. true i (r./brown) using HuBERT outpur for $\hat{\mathbf{S}}'_{\mathrm{OL}}$ model on CPC1 closed set.*

across the subset of listeners for the two models; this suggests that the listener-specific hearing loss information which the $\hat{\mathbf{S}}'_{\mathrm{OL}}$ model has access to (encoded in the audio) does not aid in the intelligibility prediction performance. It should be noted that already the enhancement system (hearing aid) has (implicitly) access to the hearing loss information and is expected to process its input signal accordingly (cf. Figure 2.30). Interestingly, both models overestimate the intelligibility ratings of speaker L0227 who performs worse than average at the intelligibility task (cf. Figure 9.3). This suggests that L0227's lower performance is not due to their hearing loss but rather other unknown factor(s); audiogram information for this listener does not show that they have particularly severe hearing loss.

Figure 10.7 and Figure 10.6 show ground truth and predicted correctness across the hearing aid systems and across the listeners for the more challenging CPC1 closed testset for the HuBERT output for $\hat{\mathbf{S}}_{\mathrm{OL}}$ and HuBERT output for $\hat{\mathbf{S}}'_{\mathrm{OL}}$ models, respectively. Systems and listeners which are unseen during the training of the models are highlighted by bold-font. Here, the overfitting of the proposed system to the hearing aid systems during training can be observed by the poor performance on the unseen hearing aid system in the testset, E018. The overall lower performance of the proposed systems on the closed set is shown by the listener-wise plots, with both systems significantly overestimating the correctness versus the true value; however the encoding of the hearing loss information in $\hat{\mathbf{S}}'_{\mathrm{OL}}$ does appear to have some positive effect here.

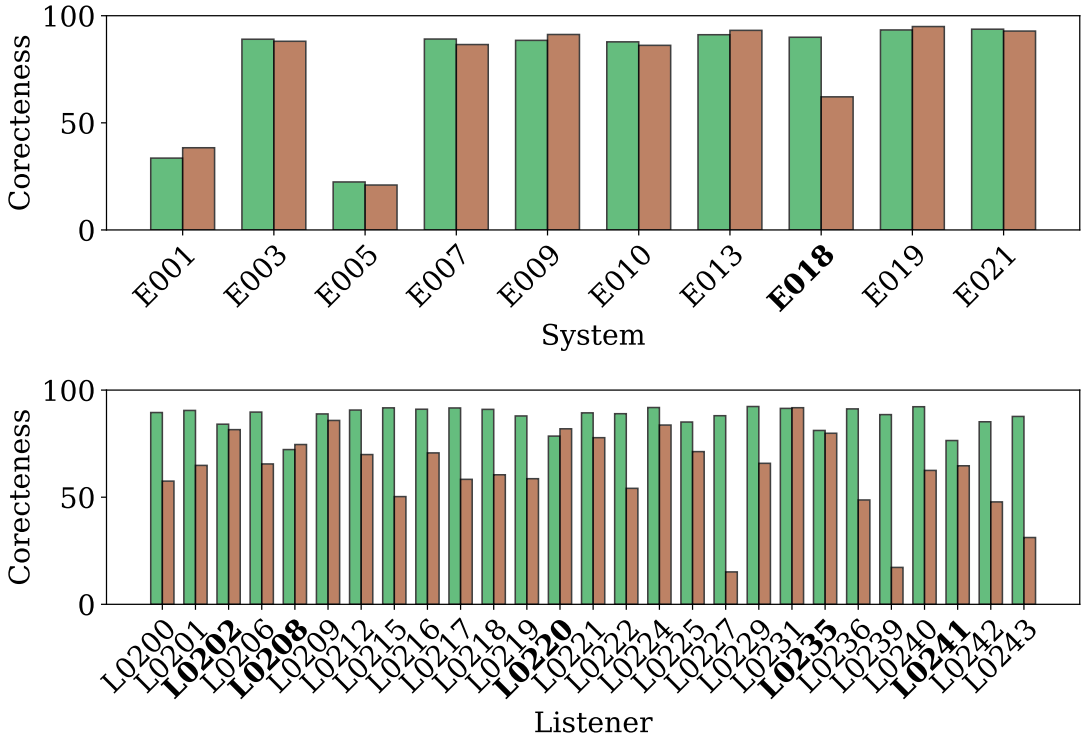

Figure 10.6: *System (top) and listener-wise (bottom) correctness prediction $\hat{i}$ (l./green) vs. true i (r./brown) using HuBERT output for $\hat{\mathbf{S}}_{\mathrm{OL}}$ model on CPC1 open set. Listeners and Systems unseen during training are bold.*

10.6 Summary

This chapter explores the use of SSSR models as feature extraction for non-intrusive SI prediction networks in comparison to traditional, spectrogram-based input. Both, the final SSSR representation and the intermediate output of the SSSR feature encoder are compared for the first time for an SI prediction task for hearing-impaired users. Results indicate that encoding the hearing loss of a particular listener via (an additional) hearing loss simulation does not typically improve performance. Additionally, models tend to overfit to specific hearing aid systems, as demonstrated by the results on the open set which might be alleviated by larger datasets released in the future.

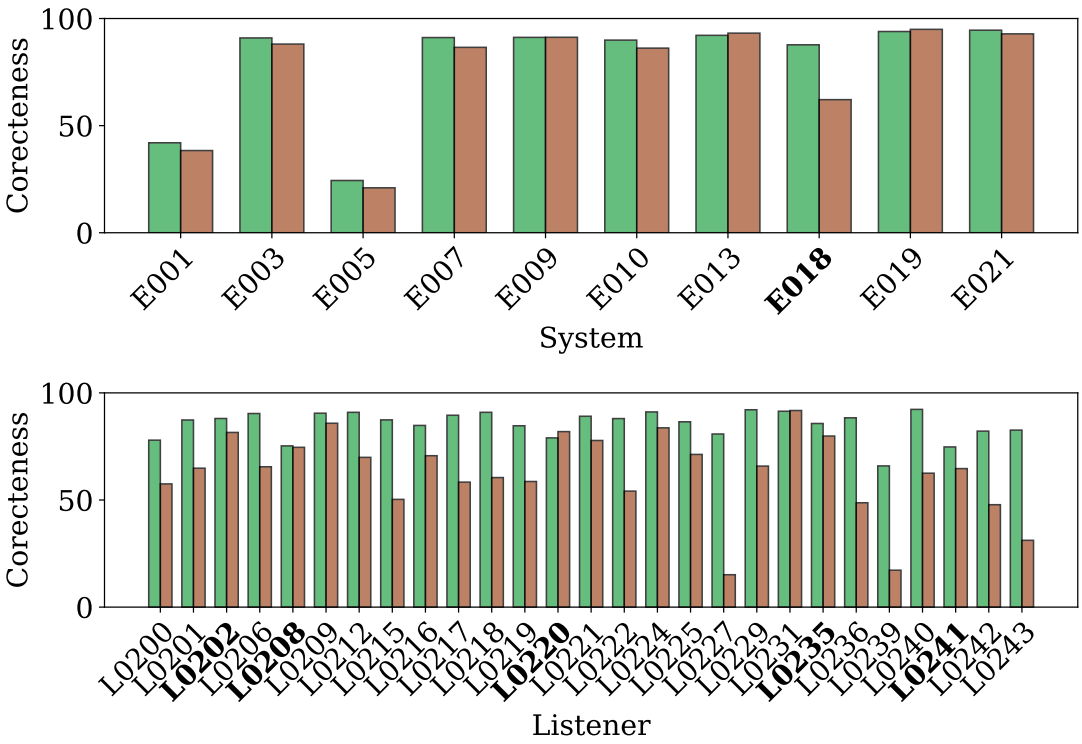

Figure 10.7: *System (top) and listener-wise (bottom) correctness prediction $\hat{i}$ (l./green) vs. true i (r./brown) using HuBERT output for $\hat{\mathbf{S}}'_{\mathrm{OL}}$ model on CPC1 open set. Listeners and Systems unseen during training are bold.*

Chapter 11

Speech Quality Prediction using SSSR and Intermediate ASR Features

There has been significant research effort developing neural-network-based predictors of SQ in recent years. While a primary objective has been to develop non-intrusive, i.e. reference-free, metrics to assess the performance of speech enhancement systems, recent work has also investigated the direct inference of neural SQ predictors within the loss function of downstream speech tasks. To aid in the training of SQ predictors, several large datasets of audio with corresponding human labels of quality have been created. Recent work in this area has shown that speech representations derived from large unsupervised or semi-supervised foundational speech models are useful input feature representations for neural SQ prediction.

In this chapter, feature representations generated by foundational models are analysed as input to a neural network for the SQ prediction task. Such features, which have primarily been developed as backbone models for ASR have proved to be useful feature representations for a number of speech related tasks (Close, Ravenscroft, et al., 2023b; Pasad et al., 2023). Experiments investigating different combinations of training data corpora are carried out, and the effects on test time performance analysed. Although non-intrusive SQ prediction is the main aim of this work, the identified best-performing models are analysed as intrusive and multi-headed (i.e predicting multiple labels at once) variants. Finally, a novel network structure incorporating recent developments in state-space models (Gu & Dao, 2023) is proposed and analysed. State-of-the-art performance is achieved on a common testset using the proposed model. Further, an implementation of the best performing model as a SQ metric is provided.

Table 11.1: *Experiment model structure and training data overview.*

Experiment	Features	Structure	Training Data
1. Feature Selection (Section 12.4)	$\mathbf{X}_{\text{FE}}$,$\mathbf{X}_{\text{OL}}$,$\bar{\mathbf{X}}_{\text{OL}}\mathbf{X}_{\text{E}}$,$\bar{\mathbf{X}}_{\text{E}}$,$\mathbf{X}_{\text{O}}$,$\mathbf{X}_{\text{MEL}}$	$\mathcal{D}_1$	NISQA
2. Training Data Selection (Section 12.5)	$\bar{\mathbf{X}}_{\text{E}}$	$\mathcal{D}_1$	NISQA,Tencent,IUB,PSTN
3. Task Variations (Section 11.5)	$\bar{\mathbf{X}}_{\text{E}}$	$\mathcal{D}_1$	NISQA, Tencent, PSTN
4. Model Variations (Section 11.6)	$\bar{\mathbf{X}}_{\text{E}}$	$\mathcal{D}_1$,$\mathcal{D}_2$,$\mathcal{D}_3$	NISQA, Tencent, PSTN

11.1 Speech Quality (SQ) Prediction Models

For non-intrusive speech quality prediction, the neural network $\mathcal{D}(\cdot)$ takes as input a feature representation

$$\mathbf{X}_{\mathrm{F}} = \mathcal{F}(x[n]) \tag{11.1}$$

of the speech or audio signal under test $x[n]$ and returns a predicted quality label $\hat{q}$. $\mathrm{F})(\cdot)$ is the feature extraction process. Typically, $\mathcal{D}(\cdot)$ is trained on data consisting of tuples $(x[n], q)$ where q is the *true* MOS quality label of the audio $x[n]$ obtained from signal assessment by human listeners. The loss function used to train $\mathcal{D}(\cdot)$ is often a simple MSE between the model output i.e the *predicted* score and the true quality label q:

$$L_{\mathcal{D}} = (\mathcal{D}(\mathbf{X}_{\mathrm{F}}) - q)^2. \tag{11.2}$$

Note that while MOS labels are typically expressed in the range 1 to 5, higher being better, for the ease of training of neural SQ predictors, q is typically normalised to a range between 0.2 and 1, which enables a sigmoid activation function on the final neural network layer to project to this label range. SQ prediction models can be broadly classifed into two types; *single-headed* models which predict only the MOS label and *multi-headed* models which predict MOS alongside some other label(s) of the input audio.
The structure of the first proposed SQ prediction models $\mathcal{D}_1(\cdot)$ is based on (Tamm et al., 2023), and is shown in Figure 11.1 together with the feature generation possibilities described in Section 2.5.2 and Section 2.5.3. Note that all feature extraction methods $\mathcal{F}(\cdot)$ used in this work output a 2-dimensional $T \times F$ representation. The base model (denoted as 'Prediction Model 1' in Figure 11.1) $\mathcal{D}_1(\cdot)$ consists of 4 transformer layers, followed by an attention pooling mechanism with a sigmoid activation function, which returns the predicted MOS score $\hat{q}$ normalised between 0.2 and 1. The input dimension (and thus the parameter count) of the transformer stage depends on the feature dimension F of the input feature, while the output dimension is fixed at 256. The attention pooling mechanism consists of two sequential linear layers, with softmax function applied at the output and is multiplied by the output of the Transformer block. The result of this multiplication is further fed into a final linear layer with a sigmoid activation to a single output neuron. This single output neuron represents the predicted MOS label $\hat{q}$ of the input audio.

Further variations of this model are also explored in later sections. In the proposed 'Prediction Model 2' $\mathcal{D}_2(\cdot)$ MAMBA (Gu & Dao, 2023) is utilised in the speech quality prediction task for the first time. A variant on the best-performing model structure introduced above is created by replacing the Transformer blocks with a bi-directional MAMBA block followed by a down-sampling linear layer. Following on from (X. Zhang et al., 2024) the BiMAMBA block consists of a single MAMBA structure and a 1D CNN layer. The input representation is flipped along the temporal axis T and both the flipped and unflipped representations are processed by the MAMBA structure in parallel. The concatenation of these unflipped and flipped MAMBA outputs are then processed by the 1D CNN layer. Finally 'Prediction Model 3' $\mathcal{D}_3(\cdot)$ consists of a single Linear layer prior the the attention head.

11.2 Mamba

MAMBA (Gu & Dao, 2023) is a recent innovation in neural network architecture. It is a structured state model (SSM) (Gu et al., 2022) but differs from the standard SSM design in two ways. Firstly it implements a selection mechanism which is dependant on the input sequence, allowing for the

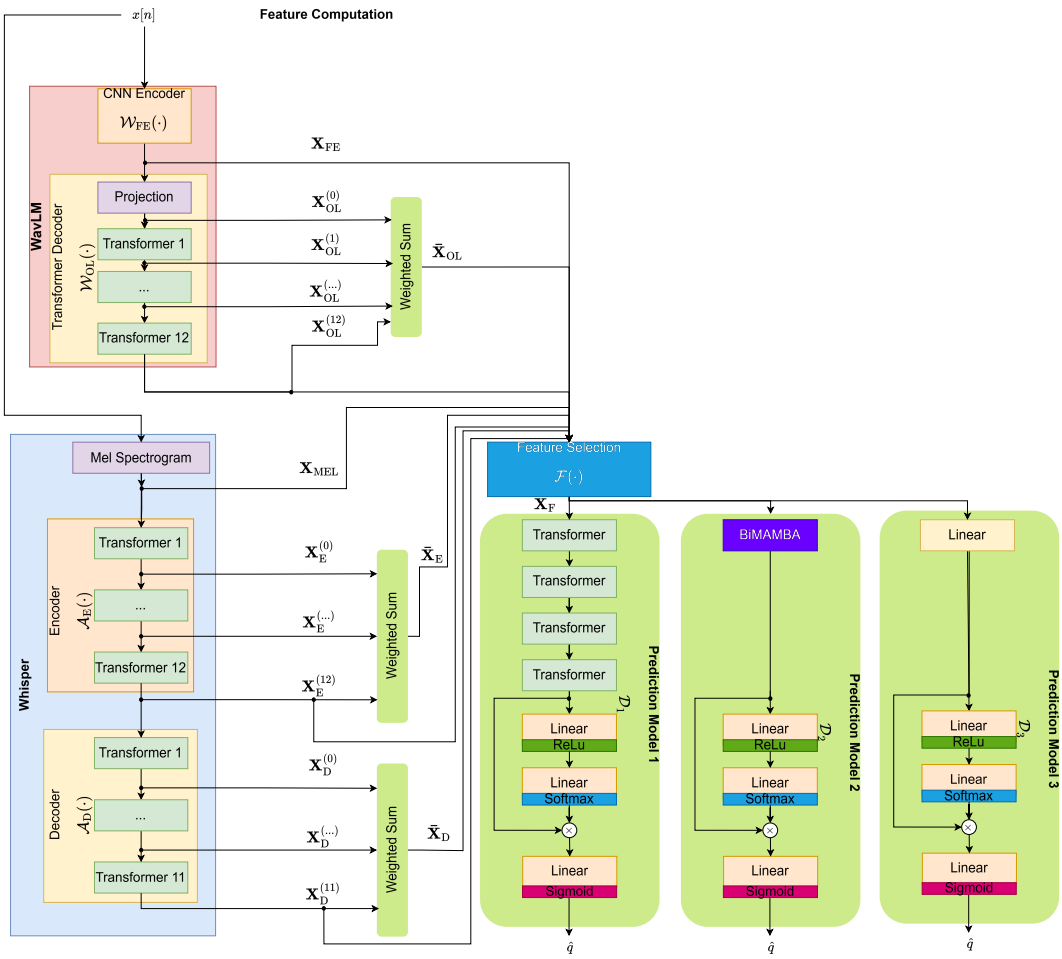

Figure 11.1: *General structure of proposed SQ prediction neural network(s) and feature extraction. Note that each 'Weighted Sum' block contains model parameters, i.e. layer weights* $\{\alpha^{(0)}, .., \alpha^{(12)}\}$ *which are updated during prediction model training.*

effective filtering of the information encoded in that input; this is somewhat analogous to the gating mechanism in a BLSTM unit. Secondly, it introduces a hardware-efficient algorithm which scales linearly with the length of the input sequence. Compared to the now ubiquitous Transformer structure, Mamba has shown improved performance in a number of tasks, and is significantly more efficient computationally. It has been applied to speech audio in the speech enhancement task (Chao et al., 2024) and ASR (X. Zhang et al., 2024) where it demonstrated state-of-the-art performance. For more detail on SSM, see Section 2.4.1.5.

11.3 Experiment 1 - Feature Selection

This section aims to uncover the best feature representation $\mathbf{X}_{\mathrm{F}}$ (cf. Section 2.5.2) for the SQ prediction task of those introduced above.

11.3.1 Experiment Setup

The SQ prediction model $\mathcal{D}_1$ is trained using a variety of different input feature representations $\mathbf{X}_{\mathrm{F}}$ introduced in Section 2.5.2 and as illustrated in Figure 11.1. All models are trained, validated and tested on the NISQA dataset, i.e. the respective LIVE and SIM subsets, then tested on each of the three available test sets (cf. Table 2.5). Following (Mittag et al., 2021), a training strategy where training stops only if the validation performance does not improve after 20 epochs is employed. The bias-aware loss function, scaling the contribution of the training samples in the loss computation based on the relative size of the training set/subset, as proposed in (Mittag et al., 2021) is also used here. The Adam (Kingma & Ba, 2014) optimiser is used with an initial learning rate of 0.00001, which is reduced by a factor of 0.1 if the validation loss does not improve after 15 epochs. All models are at first trained over a warmup epoch, where the learning rate increases up to the initial learning rate after each model update. A batch size of B of 128 is used. The best-performing epoch on the validation set in terms of validation loss is loaded at test time.

11.3.2 Results

Table 12.1 shows the results for each of the three NISQA test subsets for the trained $\mathcal{D}_1$ models, in comparison to the single-headed (i.e. predicting only the MOS label) baseline NISQA model (Mittag et al., 2021). The models are evaluated in terms of Spearman Correlation r and RMSE e. Note that the RMSE values shown are after a first-order mapping to be consistent with (Mittag et al., 2021).

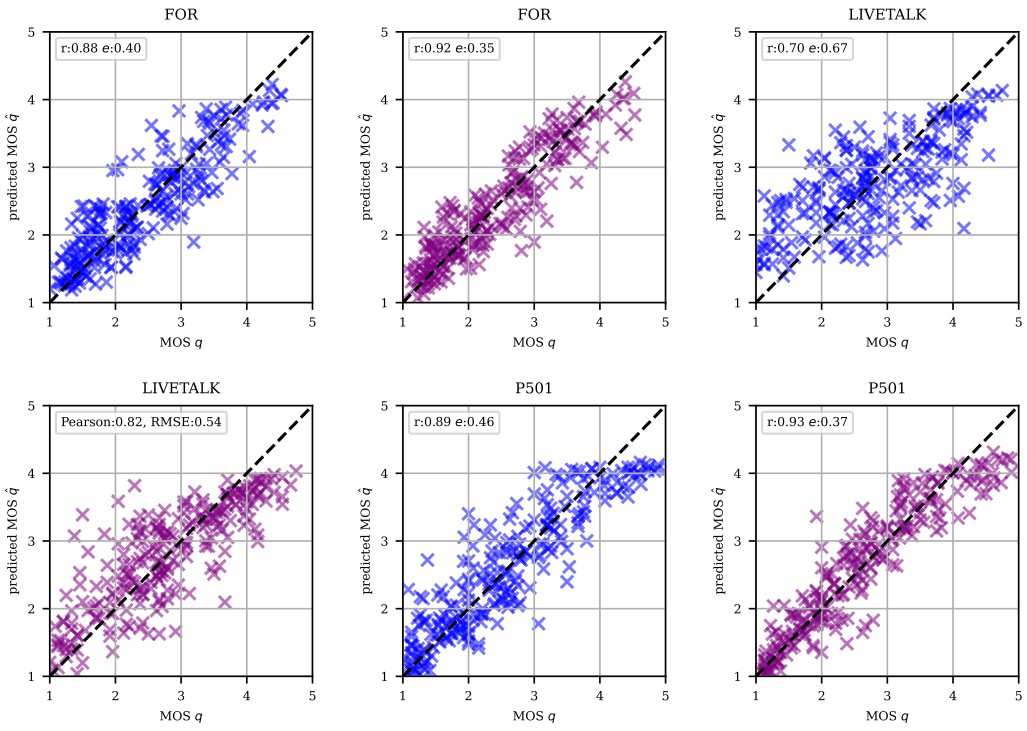

Figure 11.2: *Scatter plots for NISQA testset performance of single headed baseline NISQA (left) and best performing proposed $\mathcal{D}_1$ model using $\bar{\mathbf{X}}_{\mathrm{E}}$ (right).*

Models using the Whisper Encoder features show best performance, with the model utilising the

Table 11.2: *Predictor performance of $\mathcal{D}_1$ for best epoch (Ep.) in terms of Spearman Correlation r and RMSE e for different input features on the NISQA dataset.* ***Best*** *and* *second best* *shown in* ***Bold*** *and* *underline* *respectively.*

			FOR		LIVETALK		P501		AVERAGE	
Feature		**Ep.**	$r\uparrow$	$e\downarrow$	$r\uparrow$	$e\downarrow$	$r\uparrow$	$e\downarrow$	$r\uparrow$	$e\downarrow$
NISQA (Mittag et al., 2021)		*89*	*0.88*	*0.40*	*0.70*	*0.67*	*0.89*	*0.46*	*0.82*	*0.51*
$\mathbf{X}_{\mathrm{MEL}}$		6	0.18	0.85	0.20	0.92	0.37	0.95	0.25	0.89
WavLM	$\mathbf{X}_{\mathrm{FE}}$	87	0.72	0.61	0.51	0.81	0.73	0.70	0.65	0.71
WavLM	$\mathbf{X}_{\mathrm{OL}}^{12}$	74	0.83	0.48	0.70	0.68	0.87	0.50	0.80	0.58
WavLM	$\bar{\mathbf{X}}_{\mathrm{OL}}$	0	0.22	0.85	0.21	0.92	0.29	0.98	0.24	0.88
Whisper	$\mathbf{X}_{\mathrm{E}}^{12}$	23	<u>0.91</u>	<u>0.36</u>	<u>0.81</u>	<u>0.56</u>	<u>0.91</u>	<u>0.43</u>	<u>0.87</u>	<u>0.46</u>
Whisper	$\bar{\mathbf{X}}_{\mathrm{E}}$	30	**0.92**	**0.35**	**0.82**	**0.54**	**0.93**	**0.37**	**0.89**	**0.44**
Whisper	$\mathbf{X}_{\mathrm{D}}^{11}$	45	0.67	0.65	0.34	0.89	0.88	0.48	0.63	0.77
Whisper	$\bar{\mathbf{X}}_{\mathrm{D}}$	45	0.78	0.55	0.60	0.75	0.89	0.46	0.76	0.65

weighted feature sum $\bar{\mathbf{X}}_{\mathrm{E}}$ being the best performing on all three testsets. The performance on the LIVETALK testset is generally worse than that for the other two testsets; this is likely due to the fact that it consists of 'real' noisy recordings in a different language to the other NISQA subsets. The worst performing model was that which uses the $\mathbf{X}_{\mathrm{MEL}}$ feature.

Figure 11.2 show the distribution of MOS labels predicted by the baseline NISQA model and the best-performing $\mathcal{D}_1$ model (Whisper $\bar{\mathbf{X}}_{\mathrm{E}}$ in Table 12.1) which uses the weighted sum (cf. Figure 11.4) of Whisper Encoder features $\bar{\mathbf{X}}_{\mathrm{E}}$fkg as input. The improved performance of the proposed model over that of the baseline can be observed by the closer clustering of the predicted MOS values towards the dotted line.The greatest difference in performance between the baseline system and the proposed is on the most difficult testset LIVETALK.

11.4 Experiment 2 - Training Data Selection

Having established that the weighted sum of Whisper Encoder features $\bar{\mathbf{X}}_{\mathrm{E}}$ is the best performing of the proposed input features, this experiment aims to find which training datasets have the greatest effect on test performance, as well as enabling a fair comparison with other recently proposed SQ prediction systems.

11.4.1 Experiment Setup

The experiment setup is similar to that described in Section 12.4, except that only the $\bar{\mathbf{X}}_{\mathrm{E}}$ is used, and the other datasets introduced in Section 2.11. Unlike the NISQA dataset, the other datasets used (cf. Table 2.5) do not have defined validation sets; for these, 10% of the training sets are partitioned for validation, following (Shen et al., 2023). For the experiments which combine multiple datasets, the validation sets are combined similarly. All possible permutations of the evaluated datasets are used. The $\mathcal{D}_1$ model structure is used.

Table 11.3: *Training Data Ablation Study for best performing proposed $\mathcal{D}_1$ model.* ***Best*** *and* <u>second best</u> *shown in* ***Bold*** *and* <u>underline</u> *respectively.*

Training Data					FOR		LIVETALK		P501		AVERAGE	
NISQA	Tencent	IUB	PSTN	**Train Points**	r ↑	e↓	r ↑	e ↓	r ↑	e ↓	r ↑	e ↓
	✓			9250	0.82	0.50	0.83	0.56	0.83	0.56	0.83	0.54
✓				11020	0.92	0.35	0.82	0.54	0.93	0.37	0.89	0.44
✓	✓			20270	0.93	0.32	0.87	0.46	**0.93**	<u>0.37</u>	<u>0.91</u>	<u>0.38</u>
		✓		28800	0.27	0.84	0.42	0.85	0.41	0.92	0.37	0.87
	✓	✓		38050	0.85	0.46	0.76	0.62	0.79	0.62	0.80	0.57
✓		✓		39820	0.93	0.32	0.83	0.52	0.92	0.40	0.89	0.41
		✓		44809	0.92	0.34	0.77	0.60	0.88	0.48	0.86	0.47
✓	✓	✓		49070	0.93	0.32	0.86	0.48	0.91	0.42	0.90	0.41
	✓		✓	54059	0.91	0.36	0.85	**0.39**	0.90	0.45	0.89	0.40
✓			✓	55829	**0.94**	**0.29**	0.83	0.51	**0.94**	**0.35**	0.90	0.38
✓	✓		✓	65079	<u>0.94</u>	<u>0.30</u>	**0.88**	0.45	0.93	0.38	**0.92**	**0.38**
		✓	✓	73609	0.89	0.40	0.72	0.65	0.76	0.39	0.79	0.48
	✓	✓	✓	82859	0.92	0.34	0.81	0.55	0.83	0.56	0.85	0.48
✓		✓	✓	84629	0.94	0.30	0.87	0.46	0.93	0.39	0.91	0.38
✓	✓	✓	✓	93879	0.93	0.31	<u>0.88</u>	<u>0.45</u>	0.91	0.42	0.91	0.39

11.4.2 Results

Table 11.3 shows the results for the training data ablation experiment for the three NISQA test sets. The overall best-performing combination of training datasets is NISQA, Tencent and PSTN. By far the lowest-performing model is that trained solely on IUB; further, also a given combination of training datasets including IUB performs worse on average than that combination without IUB. As noted earlier in Section 2.11, this is likely due to the significantly different distribution of the MOS labels in this dataset relative to the others. The overall size of the training set has a lesser effect on performance and that the inclusion of data more similar to the test sets (i.e the NISQA training data) results in better performance; this can perhaps be attributed to the bias aware loss function used, which attempts to control for the imbalance in size between the component datasets. Including the Chinese language Tencent dataset in training generally improves performance on the German language LIVETALK testset; this can perhaps be attributed to these models being better able to generalise to languages other than English.

Table 11.4 shows a comparison of the best-performing system (Whisper $\bar{\mathbf{X}}_{\mathrm{E}}$ with $\mathcal{D}_1$,cf. Table 12.1) with three state-of-the-art neural SQ predictor systems (Mittag et al., 2021; Shen et al., 2023; Tamm et al., 2023). Results for the proposed system trained on the same combination of data are shown for a fair comparison. For all training data combinations, the proposed system outperforms the SOTA system; to the authors' knowledge, the results for the proposed system trained on the NISQA, Tencent and PSTN training data show the strongest correlation with human MOS labels at time of writing for the single-headed MOS label task on the NISQA testsets.

11.5 Experiment 3 - Task Variant Exploration

In this section, two variants on the MOS prediction task are explored, double-ended or *intrusive* prediction and *multi-label* prediction. Only the NISQA dataset has clean reference audio and multiple labels ('dimensions') to enable these variants. The expected result is that the additional data which these variants provide should improve the overall MOS prediction.

Table 11.4: *Comparison of $\mathcal{D}_1$ with SOTA systems*
Best *and* <u>second best</u> *shown in* ***Bold*** *and* <u>underline</u> *respectively.*

		FOR		LIVETALK		P501		AVERAGE	
Model	**Training Data**	r ↑	e ↓	r ↑	e ↓	r ↑	e ↓	r ↑	e ↓
NISQA Single Head (Mittag et al., 2021)	*NISQA*	*0.88*	*0.40*	*0.70*	*0.67*	*0.89*	*0.46*	*0.82*	*0.51*
Proposed $\bar{\mathbf{X}}_{\mathrm{E}}$	NISQA	<u>0.92</u>	<u>0.35</u>	0.82	0.54	<u>0.93</u>	<u>0.37</u>	0.89	0.44
MSQAT (Shen et al., 2023)	NISQA + Tencent + PSTN	0.90	0.39	0.85	0.51	0.92	0.42	0.89	0.44
Proposed $\bar{\mathbf{X}}_{\mathrm{E}}$	NISQA+ Tencent + PSTN	**0.94**	**0.30**	**0.88**	0.45	**0.93**	**0.38**	**0.92**	**0.38**
XLS-R SQA (Tamm et al., 2023)	Tencent + PSTN	0.90	0.38	0.83	0.52	0.89	0.46	0.82	0.51
Proposed $\bar{\mathbf{X}}_{\mathrm{E}}$	Tencent + PSTN	0.91	0.36	<u>0.85</u>	**0.39**	0.90	0.45	<u>0.89</u>	<u>0.40</u>

Table 11.5: *Results for Multi Headed $\mathcal{D}_1$ Models versus Single Head (MOS Only) Prediction*

	FOR		LIVETALK		P501		AVERAGE	
Model	r ↑	e ↓	r ↑	e ↓	**r** ↑	e ↓	**r** ↑	e ↓
NISQA Single Head	*0.88*	*0.40*	*0.70*	*0.67*	*0.89*	*0.46*	*0.82*	*0.51*
NISQA Multi Head	*0.87*	*0.43*	*0.65*	*0.72*	*0.89*	*0.46*	*0.80*	*0.54*
Proposed $\bar{\mathbf{X}}_{\mathrm{E}}$ Single Head	**0.92**	**0.35**	**0.82**	**0.54**	**0.93**	**0.37**	**0.89**	**0.42**
Proposed $\bar{\mathbf{X}}_{\mathrm{E}}$ Multi Head	0.91	0.36	0.69	0.58	0.92	0.41	0.84	0.45

11.5.1 Experiment Setup

For double-ended prediction, the Whisper encoder layers for both the test signal $x[n]$ and its corresponding clean reference signal $s[n]$ are computed and separately weighted and summed. The resultant weighted sums are then concatenated along the feature F dimension, before being passed to the network.

For multi-label prediction, 4 additional pooling attention heads (the three linear layer structure) are added to $\mathcal{D}_1$ with each being tasked with predicting *Noisiness*, *Coloration*, *Discontinuity* and *Loudness* labels, respectively.

The training setup is identical to that used in Section 12.4. The multi-headed variant of the NISQA baseline model is also trained, and its performance compared to the proposed model.

11.5.2 Results

Table 11.5 compares the performance of the baseline NISQA model and the proposed model for multi-head / multi-label prediction. In both cases, the proposed system outperforms the NISQA baselines. For both systems, tasking the model with additionally predicting the other speech dimensions from the input audio slightly degrades the performance of the main task, i.e. quality MOS prediction.

Table 11.6 compares the performance of the baseline NISQA model and the proposed model for intrusive MOS prediction. Note that results for the LIVETALK testsets are not shown as reference audio is not available for this data. For the P501 testset, access to the reference audio improves the performance of the baseline NISQA model slightly, while the performance of the proposed intrusive model remains the same as the non-intrusive version. However, for the FOR testset, the proposed intrusive model is able to achieve slightly better performance than its non-intrusive counterpart. On average, the proposed intrusive model somewhat outperforms the baseline intrusive NISQA model.

Table 11.6: *Results for Intrusive (I) versus Non-Intrusive (NI) Prediction*

	FOR		P501		AVERAGE	
Model	r ↑	e ↓	r ↑	e ↓	r ↑	e ↓
NISQA I	*0.90*	*0.39*	*0.88*	*0.48*	*0.89*	*0.44*
NISQA NI	*0.88*	*0.40*	*0.89*	*0.46*	*0.89*	*0.43*
$\bar{\mathbf{X}}_{\mathrm{E}}$ I	**0.94**	**0.31**	**0.93**	**0.37**	**0.94**	**0.34**
$\bar{\mathbf{X}}_{\mathrm{E}}$ NI	0.92	0.35	0.93	0.37	0.93	0.36

11.6 Experiment 4 - Model Variations

In this experiment, the $\mathcal{D}_2$ and $\mathcal{D}_3$ variations of the SQ prediction model structure (as shown in Figure 11.1) are explored.

11.6.1 Experiment Setup

Training data setup follows the best-performing training data combination of NISQA, Tencent and PSTN detailed in Table 11.3, while the other training configuration parameters remain the same as the previous experiments, however, the initial learning rate of the BiMAMBA based model ('Prediction Model 2' in Figure 11.1) is set to 0.0001 (rather than 0.00001).

11.6.2 Results

Table 11.7 compares results of the best-performing Transformer based model to the BiMAMBA-based model and the simple Linear model. The performance of the all three models is very similar, however, the BiMAMBA-based model *slightly* outperforms the Transformer-based model on average. The BiMAMBA model is also loaded from a significantly earlier training epoch (5 versus 68); this is likely due to the higher initial learning rate set for the BiMAMBA model. Overall, these results speak to the potential usefulness of MAMBA for this task.The simple Linear layer based model performs surprisingly well, with performance comparable to the significantly more complex Transformer and BiMAMBA based models.

In order to better understand why the proposed weighted sum of Whisper encoder features $\bar{\mathbf{X}}_{\mathrm{E}}$ is so effective, two additional models using the simple Linear layer base are trained. In the former, the region in $\bar{\mathbf{X}}_{\mathrm{E}}$ which is padding in the input (i.e for a 10 second long utterance the region after $T = 500$ in the 1500 long T axis) is set to 0. In the latter this is reversed, with the region which corresponds to the audio audio in X_{MEL} being set to 0. Figure 11.3 visualises the input features for these models. The models are trained on the best performing dataset combination outlined above. In order to prevent any batchwise padding operations from effecting the results, the batch size is set to 1 for this experiment for both models.

Table 11.8 shows the results for these two models. Interestingly, the model is able to perform well even with the region in $\bar{\mathbf{X}}_{\mathrm{E}}$ corresponding to the Mel spectrogram of the input audio signal is masked. This suggests that *the Whisper Encoder is utilizing the padding region to encode meaningful information*. However, this model struggles on the more difficult LIVETALK test set compared to the model which was trained on $\bar{\mathbf{X}}_{\mathrm{E}}$ with the padding region masked.

Table 11.7: *SQ Model Variation performance using simple Linear model base.*

				FOR		LIVETALK		P501		AVERAGE	
Feature	**Model Base**	**Parameter Count**	**Epoch**	r ↑	e ↓	r ↑	e ↓	r ↑	e ↓	r ↑	e ↓
Proposed $\bar{\mathbf{X}}_{\mathrm{E}}$	Transformer $\mathcal{D}_1$	2,440,719	68	**0.94**	0.30	**0.88**	**0.45**	0.93	0.38	0.92	0.38
Proposed $\bar{\mathbf{X}}_{\mathrm{E}}$	BiMAMBA $\mathcal{D}_2$	2,214,671	5	0.94	**0.29**	0.87	0.47	**0.94**	**0.35**	**0.92**	**0.37**
Proposed $\bar{\mathbf{X}}_{\mathrm{E}}$	Linear $\mathcal{D}_3$	**330,767**	85	0.93	0.31	0.87	0.46	0.93	0.37	0.91	0.38

Table 11.8: *Masked $\bar{\mathbf{X}}_{\mathrm{E}}$ performance for model $\mathcal{D}_3$*

		FOR		LIVETALK		P501		AVERAGE	
Feature	**Epoch**	r ↑	e ↓	r ↑	e ↓	r ↑	e ↓	r ↑	e ↓
Proposed $\bar{\mathbf{X}}_{\mathrm{E}}$ padding region masked	43	0.81	0.53	**0.82**	**0.53**	0.89	0.46	0.84	0.51
Proposed $\bar{\mathbf{X}}_{\mathrm{E}}$ signal region masked	57	**0.90**	**0.38**	0.78	0.59	**0.90**	**0.45**	**0.86**	**0.47**

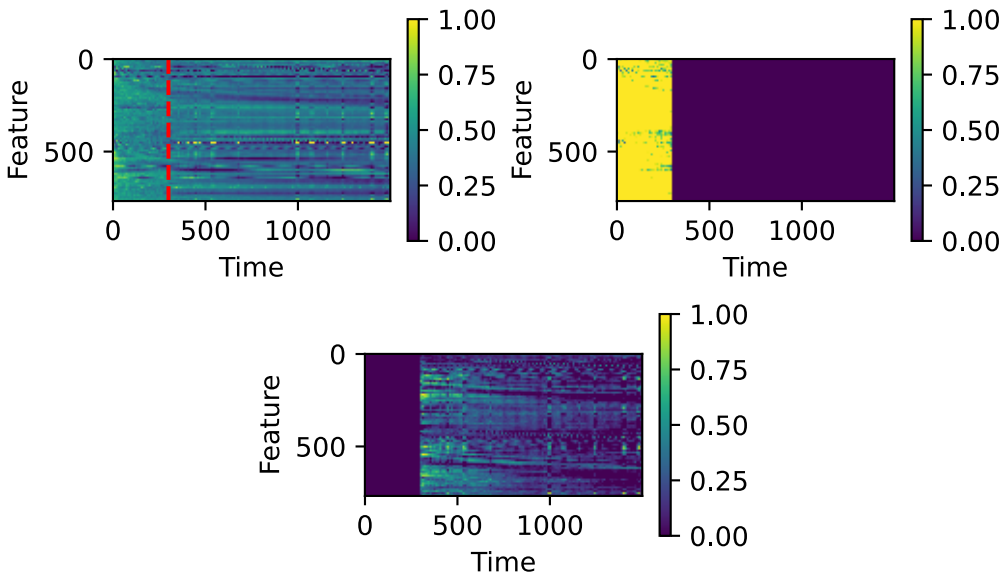

Figure 11.3: *Unmasked $\bar{\mathbf{X}}_{\mathrm{E}}$ features (top left), $\bar{\mathbf{X}}_{\mathrm{E}}$ with padding region masked (top right) and $\bar{\mathbf{X}}_{\mathrm{E}}$ with signal region masked (bottom)*

11.7 Analysis

11.7.1 Layer Weights

Figure 11.4 shows the learned layer weight model parameters for the weighted sum input features for models trained on NISQA only (being the general baseline) as well as those trained on NISQA, Tencent and PSTN (since this dataset combination showed best performance in Table 11.3). The weight values are generally larger for the models trained on the larger training set, however, the same general trend is consistent between the training setups. A particularly interesting feature of the WavLM $\bar{\mathbf{X}}_{\mathrm{OL}}$ weights is the extremely high weighted assigned to the first layer $\mathbf{S}_{\mathrm{OL}}^{(0)}$. This is the output projection of the output of the CNN encoder stage, which suggests that that representation most strongly encodes speech quality-related information, as supported by findings in (Close, Hain, et al., 2023a; Close, Ravenscroft, et al., 2023b).

The layer weights from the Whisper model are somewhat less immediately interpretable. For

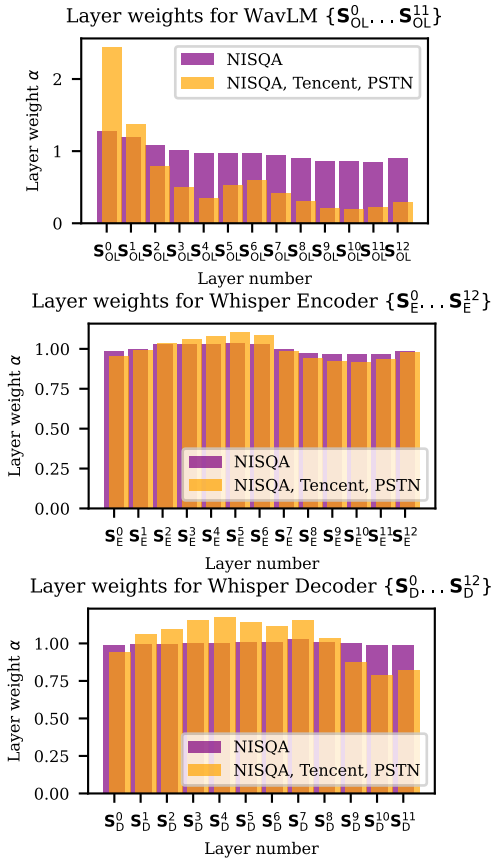

Figure 11.4: *Layer weights for models trained on NISQA only and NISQA, Tencent and PSTN for different input features: WavLM (top), Whisper Encoder (middles) and Decoder (bottom).*

the Whisper encoder, there is a slight bias towards layers 6 and 7. For the Whisper decoder, the final three layers are weighted significantly less than all preceding; this can perhaps be explained by these layers more strongly encoding linguistic 'word level' information useful for Whisper's primary application of ASR. This also explains the poor performance of $\mathbf{S}_{\text{D}}^{(11)}$ as an input feature in Table 12.1.

11.8 Summary

In this chapter, features derived from large pre-trained speech models are used as feature extraction stages for the speech quality label prediction task. State of the art performance is achieved by models which use a weighted sum of Whisper Encoder features as the input audio representation. A number of experiments are carried out, investigating variations to the training data, task and base model structure, all of which further speak to the strength of the proposed feature. Implementation code and checkpoints for the best performing model (dubbed Whisper based Speech Quality Assessment ('WhiSQA')) can be found here[1] along with a demonstrative inference script.

[1] `https://github.com/leto19/WhiSQA`

Chapter 12

Hallucinations in Perceptually Motivated Speech Enhancement

12.1 Introduction

In the CHiME7 challenge UDASE task (Leglaive et al., 2023; Leglaive et al., 2024) (cf. Section 2.9.3) it was shown that high metric scores from non-intrusive neural SQ predictors do not always match with actual human MOS evaluation. The evaluation of the SE system entries to the UDASE task had two stages; first the entries were evaluated in terms of the scores from the DNSMOS (Reddy et al., 2022) neural non-intrusive SQ metric. Then, in the second evaluation stage, listening tests were conducted and MOS scores for audio enhanced by the challenge entries were computed from these listening tests. The best-performing system in the first evaluation stage is described in Chapter 5, an SE system which utilises a non-intrusive MetricGAN (Fu, Yu, Hung, et al., 2021) framework to directly optimise towards the DNSMOS metric. However, this system was scoring lowest of the entries going forward to the listening-test evaluation stage; by optimising directly for high DNSMOS scores, the SE system may learn to introduce specific distortions which result in high DNSMOS scores but which negatively impact the actual perceptual quality of the enhanced audio when assessed by humans.

In general, it was found in the UDASE task results (Leglaive et al., 2024) that quality ratings from non-intrusive quality predictors such as DNSMOS (Reddy et al., 2022) and TorchAudio-SQUIM (Kumar et al., 2023) did not correlate strongly with the MOS ratings obtained in the second evaluation stage by listening tests and that traditional intrusive signal processing based metrics such as PESQ and STOI showed significantly stronger correlation.

This chapter therefore has two major objectives. Firstly, to better understand how SE systems like that proposed in Chapter 5 learn to optimise their outputs towards neural non-intrusive SQ metrics during training. Secondly, to identify why neural non-intrusive SQ metrics fail to properly assess the human assessed quality of the output of SE systems, even in the setting that the SE system does not directly optimise the metric in training.

12.2 Non-Intrusive Speech Quality Predictor

The non-intrusive SQ predictor $\mathcal{D}(\cdot)$ used in this chapter is based on that introduced in Chapter 11 and consists of a Transformer (Vaswani et al., 2017) block, followed by a feed-forward attention block with a sigmoid activation on a single output neuron which represents the predicted quality $\hat{q}'$ of the input audio, normalised between 0 and 1.

The proposed predictor differs from that in (Tamm et al., 2023) as follows: Rather than an input feature derived from the XLS-R representation, the input feature of $\mathcal{D}(\cdot)$ is the output of the Transformer Encoder stage of a pre-trained Whisper (Radford et al., 2022) ASR network (cf. Section 2.5.3). This representation has been shown to be a useful feature representation for similar non-intrusive prediction tasks (Mogridge et al., 2024; Santiago Cuervo, Ricard Marxer, 2024). In this work, the `whisper-small`[1] model, trained on 680k hours of labelled speech data is used. The encoder stage of this model returns a representation of fixed dimension $F_{\text{Enc}} \times T_{\text{Enc}} = 768 \times 1500$. Note that the Whisper encoder block is used solely as a feature extractor, and its parameters are not updated during the training of $\mathcal{D}(\cdot)$.

The metric prediction network $\mathcal{D}(\cdot)$ is trained as follows: The MOS label q in most datasets is expressed as a value between 1 and 5, higher being better. In the training and inference of $\mathcal{D}(\cdot)$, this value is normalized between 0.2 and 1, which is denoted as q'. For a pair of audio and normalized MOS label $\{x[n], q'\}$, the model is trained with a loss between the output of the model (i.e. the predicted quality of $x[n]$) and the true normalized MOS label q':

$$L_{\mathcal{D}} = (\mathcal{D}(x[n]) - q')^2 \tag{12.1}$$

The model is trained following a scheme similar to that proposed in (Mittag et al., 2021) where training halts if the validation performance does not improve after 20 epochs.

The performance of the proposed non-intrusive metric prediction network $\mathcal{D}(\cdot)$, trained and tested on the NISQA (Mittag et al., 2021) dataset is shown in Table 12.1, compared to the NISQA baseline. The NISQA test set and (baseline) model are widely used benchmarks for the SQ prediction task. The proposed predictor network outperforms this baseline both in terms of spearman correlation r and RMSE across all three NISQA testsets (P501, FOR and LIVETALK), and is comparable or better than state-of-the-art systems (Shen et al., 2023; Tamm et al., 2023) on these testsets. In addition, a variant of $\mathcal{D}(\cdot)$, denoted as $\mathcal{D}_{\mathcal{B}}(\cdot)$ in Table 12.1, is trained based on additional datasets, i.e NISQA (Mittag et al., 2021), Tencent (Yi et al., 2022) and PTSN (Mittag et al., 2020) speech quality datasets, which shows similar, in mean further increased performance. See Chapter 11 for more details on Whisper based SQ predictors.

Testset	P501		FOR		LIVETALK		MEAN	
Model	r↑	RMSE↓	r↑	RMSE↓	r↑	RMSE↓	r↑	RMSE↓
NISQA	0.89	0.46	0.88	0.40	0.70	0.67	0.82	0.51
$\mathcal{D}(\cdot)$	**0.94**	**0.35**	0.93	**0.32**	0.81	0.54	0.89	**0.40**
$\mathcal{D}_B(\cdot)$	0.93	0.37	**0.94**	0.32	**0.85**	**0.50**	**0.91**	**0.40**

Table 12.1: *Proposed SQ Predictor compared with baseline NISQA model.*

[1] `https://huggingface.co/openai/whisper-small`

Loss	α in ((12.2))	PESQ	STOI	Composite			SISDR	DNSMOS			$\mathcal{D}_B(\cdot)$
				CSIG	CBAK	COVL		SIG	BAK	OVR	
-	*clean*	*4.50*	*1.00*	*5.00*	*5.00*	*5.00*	*91.14*	*4.27*	*4.36*	*3.88*	*0.67*
-	*noisy*	*1.97*	*0.92*	*3.34*	*2.44*	*2.63*	*8.44*	***4.24***	*3.32*	*3.36*	*0.58*
L_{RI} only, ((7.5))	1	**2.99**	**0.95**	**4.09**	**3.57**	**3.55**	**19.82**	4.14	4.42	3.86	0.65
$\leftarrow L$, ((12.2)) $\rightarrow$	0.9	2.93	0.94	3.96	3.52	3.44	19.70	4.12	**4.46**	**3.95**	0.68
	0.8	2.63	0.93	3.59	3.33	3.09	19.62	4.05	4.41	3.91	0.70
	0.7	2.72	0.94	3.78	3.38	3.24	19.39	4.08	4.30	3.78	0.71
	0.6	2.63	0.93	3.45	3.25	3.00	19.25	4.00	4.33	3.79	0.79
	0.5	2.65	0.93	3.57	3.29	3.07	19.43	4.04	4.36	3.84	0.77
	0.4	2.66	0.93	3.67	3.31	3.14	18.92	4.06	4.28	3.75	0.76
	0.3	2.68	0.93	3.79	3.34	3.22	18.98	4.11	4.38	3.83	0.77
	0.2	2.58	0.93	3.47	3.25	3.00	18.51	3.92	4.24	3.70	0.75
	0.1	2.37	0.91	3.29	3.10	2.79	17.72	4.02	4.29	3.75	0.76
L_{SQ} only, ((12.3))	0	1.43	0.41	1.00	1.03	1.02	-29.68	2.55	2.54	2.42	**0.88**

Table 12.2: *Performance of Speech Enhancement for different α in ((12.2)) for the VoiceBank-DEMAND testset.*
Best performance denoted in **bold** font. Unprocessed data denoted in *italic* font.

12.3 Speech Enhancement System

12.3.1 Model Structure

The DPT-FSNet (Dang et al., 2022) single-channel speech enhancement architecture which is based on the Dual Path Transformer (DPT) architecture is used as the baseline speech enhancement system denoted as $\mathcal{G}(\cdot)$ in this work. This model has shown state-of-the-art performance in this task, despite a relatively small parameter count. It takes as input the real and imaginary STFT components $\mathbf{X}_r$ and $\mathbf{X}_i$ of the noisy time domain signal $x[n]$, and returns mask matrices $\mathbf{M}_r$ and $\mathbf{M}_i$ which are multiplied with the inputs to produce estimated of the clean complex signal spectra, i.e. $\hat{\mathbf{S}}_r$ and $\hat{\mathbf{S}}_i$. These are then used as inputs to an ISTFT operation to produce the enhanced time domain audio $\hat{s}[n]$. For a detailed description of the architecture see Section 2.14.

12.3.2 Loss Function

To train the proposed adaptation of the DPT-FSNet network $\mathcal{G}(\cdot)$, an extended loss function

$$L = \alpha L_{\text{RI}} + (1-\alpha) L_{\text{SQ}} \tag{12.2}$$

is proposed which adds a loss term

$$L_{\text{SQ}} = (1 - \mathcal{D}_B(\hat{s}[n]))^2 \tag{12.3}$$

based on inference of the non-intrusive pre-trained SQ predictor (cf. Section 12.2) of the enhanced audio $\hat{s}[n]$ to the loss used in the original DPT-FSNet paper (Dang et al., 2022) L_{RI} is the real and imaginary STFT MSE loss term given in (4.1). Note that the time domain loss term (2.38) as outlined in (Dang et al., 2022) is not utilised here. The hyperparameter α in ((12.2)) is a value between 0 and 1 which controls the relative weight of the intrusive and non-intrusive terms which will be analysed in the following.

12.4 Experiment 1 - Scaling the Quality Estimator's Influence

In this experiment, the SE system $\mathcal{G}(\cdot)$ is trained for different α in ((12.2)), i.e. for varying degrees of influence of the quality estimator $\mathcal{D}_B(\cdot)$ in the loss function. In doing this, it is possible to compare the performance of at one pole, a traditional signal-processing-based intrusive loss function, i.e. L_{RI} in ((7.5)) only, and at the other a purely non-intrusive SQ predictor loss, i.e. L_{SQ} in ((12.3)) only, as well as points between these poles, i.e. the combined loss in ((12.2)) .

12.4.1 Experiment Setup

Each speech enhancement system model, i.e. for varying α is trained for 200 epochs on the VoiceBank-DEMAND (Valentini-Botinhao et al., 2016) training set (cf. Section 2.9.1). The Adam (Kingma & Ba, 2014) optimizer is used; following (Dang et al., 2022), a dynamic strategy to adjust the learning rate is employed, where the learning rate steadily increases during the first few model updates and then scales down over the remaining training epochs.

12.4.2 Results

Table 12.2 shows the speech enhancement performance of the experiment described in Section 12.4.1 for the VoiceBank-DEMAND testset. The models are evaluated by frequently-used signal-processing-based intrusive measures PESQ, STOI, the three terms of the Composite measure (Lin et al., 2019) CSIG, CBAK and COVL and the SI-SDR (Roux et al., 2018). The models are also evaluated using the non-intrusive neural SQ measure DNSMOS (Reddy et al., 2022) as well as in terms of the score assigned by $\mathcal{D}_B(\cdot)$ detailed above in Section 12.2.

The best performing model in terms of the standard intrusive measures is the model with $\alpha = 1$ in (12.2), i.e. where no inference of $\mathcal{D}_B(\cdot)$ is used, and the loss function consists solely of the tempo-spectral distance in the loss term defined in (7.5). Generally, as the value of α decreases, so do the scores. At $\alpha = 0$ (i.e. solely using inference of $\mathcal{D}_B$ as defined in (12.3) as the loss function), the performance degrades significantly, being drastically worse than even the input noisy data in all intrusive measures. The difference in performance between an $\alpha = 0$ and $\alpha = 0.1$ is stark, suggesting that even a small weighting of the intrusive loss term (7.5) is enough to greatly improve performance.

Performance assessed by the non-intrusive measures in the right part of Table 12.2 follows a somewhat different pattern. All models degrade the DNSMOS SIG score in comparison to the noisy (as well as the clean) audio. This is consistent with the findings in masking-based SE in general and for the UDASE task in particular (Leglaive et al., 2024), where all enhancement systems show degraded DNSMOS SIG with the exception of those systems which explicitly optimise towards it in training. While the DNSMOS ratings generally decrease as α does, the model for $\alpha = 0.9$ outperforms the model with $\alpha = 1$ in terms of the BAK and OVR components. Furthermore, the model for $\alpha = 0.9$ performs only slightly worse than the model for $\alpha = 1.0$ in terms of the intrusive metrics, suggesting that a small weighting of (12.3) might be beneficial to the overall audio quality. However, this DNSMOS performance should be interpreted with some scrutiny; the results here show that some of the models outperform even the clean reference audio in terms of DNSMOS, which might be surprising in the first instance. However, later spectrogram analysis (cf. Figure 12.1) shows that *clean* signals sometimes contain noise (primarily breathing sounds) which are removed

by the SE system. Furthermore, all DNSMOS scores for $\alpha = 0$ are much too high given that this system completely destroys the input signal. As noted in Chapter 5, this might be due to an extreme failure to generalise in DNSMOS. As it is to be expected, the $\mathcal{D}_B(\cdot)$ score increases as α decreases and inference of $\mathcal{D}_B(\cdot)$ is weighted more heavily in the loss function.

12.4.3 Spectrogram Comparison

Figure 12.1 exemplarily shows spectrograms for the VoiceBank-DEMAND testset file `p232_005.wav`; clean reference $\mathbf{S}$ and noisy audio $\mathbf{X}$ (top two panels) as well as enhanced signals $\hat{\mathbf{S}}$ for different α in (12.2) are shown. With the exception of $\alpha = 0$, all models successfully remove the distortion tone present in the first 2 seconds of the input noisy signal at approx. 500 Hz, and generally do enhance the noisy input such that it resembles the clean reference.

As the value of α decreases however, a distortion in the first half second of the audio becomes more prominent. This distortion is interesting for a number of reasons. It does not resemble the noise in this region in the noisy input signal and occurs consistently in appearance spectrally and in audible sound across all audio enhanced by the models, indicating that it can be best characterised as a *hallucination* of the enhancement model(s). This hallucination is most prominent in the model where $\alpha = 0$; other than the hallucination the outputs of this model consist of seemingly meaningless content which does not resemble the noisy input signal at all. Given that the hallucination appears more strongly as the influence of the quality predictor-based loss term (12.3) increases, it is likely caused by the speech enhancement system learning to *trick* $\mathcal{D}_B(\cdot)$. The consistent form of the distortion can also be explained as follows; during the training of $\mathcal{D}_B$, it learned to assign a high-quality rating for input audio which contained a sound like this hallucination. *Then during the training of the SE models, the SE models learn to exploit this quirk of the training of $\mathcal{D}_B$ by introducing the hallucination in order to minimise the loss function.* The consistent temporal position of the hallucination can be explained by the short non-speech region at the start of the audio file which is often present across all audio in the VoiceBank-DEMAND and similar datasets. The presence of this hallucination is likely the cause of the decrease in performance in terms of intrusive signal processing metrics Table 12.2 while the non-intrusive neural SQ metrics change less uniformly; the intrusive metrics all involve a direct comparison with the reference audio which explicitly penalise the presence of the hallucination. The hallucination has a *speech-like* characteristic which is possibly the reason that the SQ predictor models reward its presence.

12.5 Experiment 2 - Listening Test

In order to better understand the performance of the trained SE models and the human perception of the hallucination distortion, a small listening test experiment was carried out.

12.5.1 Setup

Noisy audio files from the VoiceBank-DEMAND testset and audio enhanced by enhancement models with α values of 0, 0.1, 0.5 and 0.9 were randomly selected for a total of 15 files (3 files from each of the 4 α values plus the noisy signal). The ITU-T P.835 (*Subjective test methodology for evaluating speech communication systems that include noise suppression algorithm.* 2003) methodology was used, inspired by (Leglaive et al., 2023). 16 participants sequentially rated each

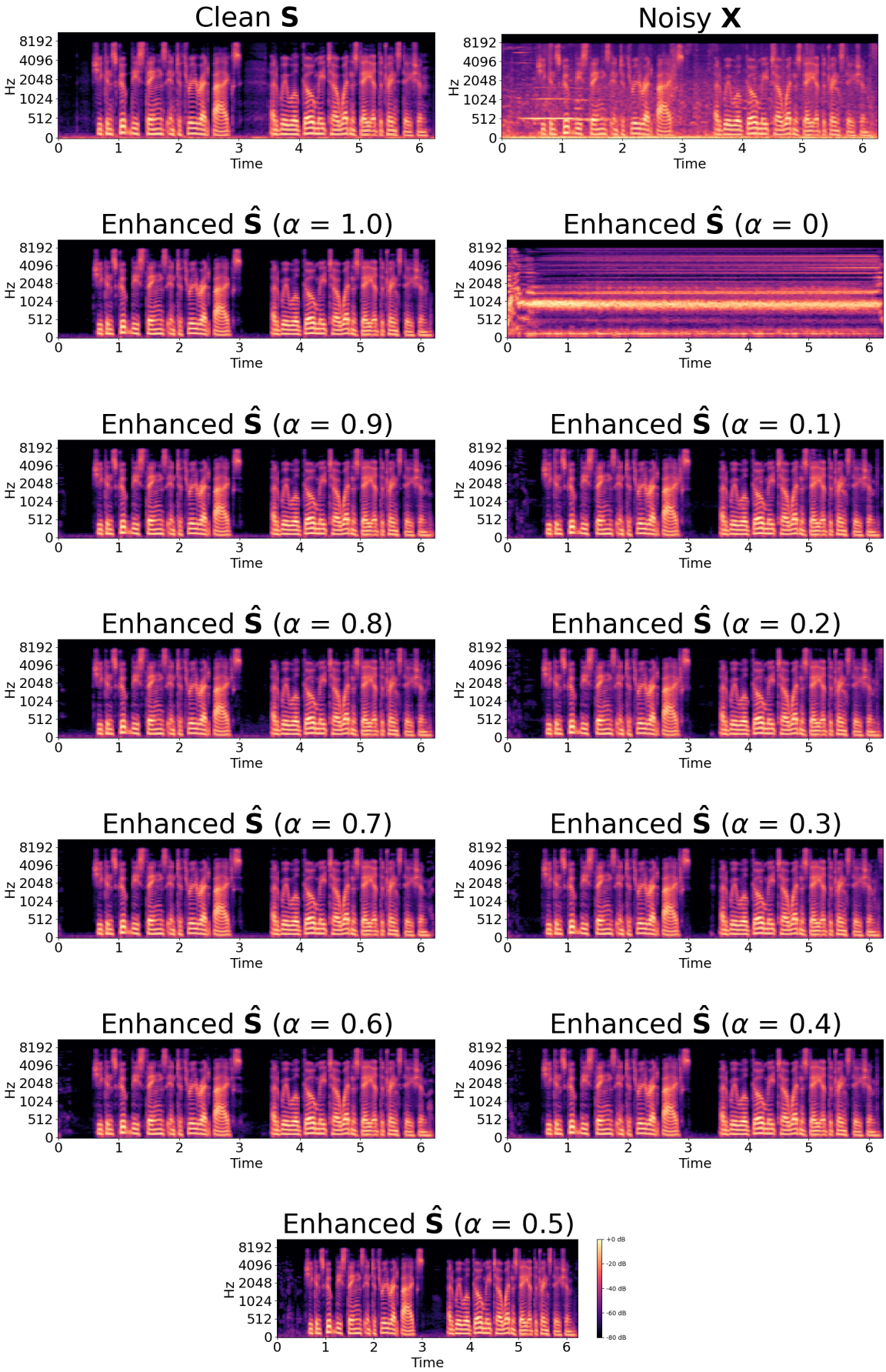

Figure 12.1: *(Magnitude) spectrogram comparison for differing values of α.*

	SIG		BAK		OVRL	
α	**MEAN**	**STD**	**MEAN**	**STD**	**MEAN**	**STD**
noisy	***4.54***	*0.65*	*2.92*	*0.82*	*3.67*	*0.88*
1.0	4.50	0.62	**4.67**	0.66	**4.42**	0.74
0.9	4.31	0.72	4.54	0.65	4.25	0.73
0.5	4.06	0.86	3.58	1.03	3.63	0.96
0.1	4.35	0.70	3.83	0.69	3.94	0.81

Table 12.3: *Listening Test Results*
Best performance denoted in **bold**. Unprocessed data denoted in *italic*.

file in terms of the naturalness of the speech signal, the intrusiveness of the background noise distortion and overall quality on 5 point Likert scales (i.e SIG, BAK and OVRL), for a total of 48 ratings per audio file. The listening test audio is available online[2].

12.5.2 Results

The results of the listening test are shown in Table 12.3. In terms of signal quality SIG, the noisy input audio scores the highest; this is in line with the MOS results reported in (Leglaive et al., 2023) and results in Table 12.2. For BAK and OVRL, the listening-test results follow those of the metrics in Table 12.2 with the model for $\alpha = 1.0$ being the best performing.

Interestingly, $\alpha = 0.1$ significantly outperforms $\alpha = 0.5$ in all aspects in the listening tests, outperforming even $\alpha = 0.9$ which showed the best performance in Table 12.2 in terms of SIG. The low BAK and OVRL scores of for $\alpha = 0.5$ and 0.1 suggest that the hallucination is perceptible, but that the listeners considered it as an aspect of the background rather than a distortion in the speech signal itself. This is important when considering the disconnect between intrusive metric scores and human perception MOS. An intrusive metric like PESQ is *directly comparative* such that deviation in the test signal from the oracle reference signal always results in a lower output score. On the other hand, human MOS is *indirectly comparative*; the score is informed wholly by the listener's preconceived notion of speech quality, which varies not only between individuals but also unconsciously over time during the listening test. Likewise, non-intrusive SQ predictors are also indirectly comparative, with the output score informed by the training data. This is exemplified clearly by comparing the Composite measure CSIG score of noisy audio in Table 12.2 with the analogous DNSMOS SIG score in the same table and the real MOS SIG average in Table 12.3. The noisy signals are generally dissimilar to the clean references, meaning that the intrusive CSIG score suffers but this does not *in reality* mean that the human perception (or a predictor of human perception) of the speech distortion suffers drastically.

12.6 Summary

In this chapter, SE models which are optimised using non-intrusive neural SQ predictors are shown to produce hallucinatory artefacts in output audio. These hallucinations do not represent meaningful content but are learned by the SE system in order to optimise the audio towards maximising the score

[2] `https://leto19.github.io/nisqa_se_demo.html`

awarded by the SQ predictor. Intrusive metrics like PESQ are sensitive to these hallucinations, and they are shown to generally be perceptible in human listening tests.

Part V

Conclusions

Chapter 13

Concluding Remarks

In this thesis, a number of advances and novel methods in the domain of SE and assessment of SE systems were proposed. There were three main topic areas; psychoacoustic training objectives of NNSE as implemented in the MetricGAN framework, the incorporation of SSSR features into the training objectives of NNSE systems and the prediction of human derived labels of SQ and SI directly from audio signals.

Several novel extensions and applications of the MetricGAN framework were proposed and evaluated in this thesis. In Chapter 3 a novel extension designed to improve the generalisation of the metric prediction component of the framework was proposed. Then, in Chapter 4 further variations on the framework were proposed, including the incorporation of new more advanced NN structures for the SE component. Then in Chapter 5 and Chapter 6, the MetricGAN framework is expanded into incorporating the prediction of a non-intrusive MOS predictor. It is here that a potential pitfall with the framework becomes apparent, especially in the case where a non-intrusive metric is optimised towards; the NNSE component learns to produce audio which the metric being optimised towards rewards, while in reality (as shown by listening test results) the signal is degraded.

Chapter 7 proposes the use of SSSR encoder output representations in a loss function for NNSE training. Further, it is shown that the proposed distance measure correlates strongly with traditional signal processing based measures of speech quality, as well as with human MOS labels. Chapter 8 further develops this idea, while also introducing a new framework for the creation of datasets for NNSE training and testing.

In Chapter 9 both metric prediction and direct SI prediction are applied to the CPC1 task. Then in Chapter 10 this concept is further developed by incorporating SSSR derived feature representations for the SI prediction task. This approach is applied to the SQ prediction task in Chapter 11 and expanded by incorporating features derived from Whisper. Finally, Chapter 12 details experiments involving taking inference of an SQ prediction during NNSE training. Similar to the later work involving MetricGAN, it is found that optimisation towards this SQ predictor does not result in generally good NNSE performance in terms of human listening test results.

Chapter 14

Future Research

The work presented in this thesis leaves open a number of avenues for potential future research.

One emerging direction of NNSE research is the application of generative techniques such as *diffusion* (Gonzalez et al., 2024; Richter et al., 2023; Yen et al., 2023) or synthesis decoders (Song et al., 2024). A commonality of these approaches is that they do not consider any perceptual characteristics of the generated speech in their training; a natural direction of research is to attempt to introduce a MetricGAN like SQ optimisation training objective for diffusion based models. This is likely a challenge, as the optimal method to introduce real world distortion as guiding supervision into the training of diffusion NNSE systems remains elusive, and an open topic of ongoing research (Richter, de Oliveira, et al., 2024).

Another key quality of generative approaches is that traditional intrusive signal processing based metrics struggle to assess their performance. This is usually due to the the destructive nature of the generative processes involved, some information (in particular timing and prosodic) present in the input signal is lost at the output. As such there is a great need for a standardised, reliable and *non-intrusive* SQ measure. However, as demonstrated by Chapter 5 and Chapter 12 both direct and indirect optimisation of such a metric is possible and consistently produces audio which *fools* the metrics while in reality degrading the signal, as repeatedly shown by listening test results. It could be argued that directly optimising towards the evaluation metric is unwise, and essentially constitutes as *adversarial attack* on the metric. However, this is also true of MetricGAN, and recent work (de Oliveira, Welker, et al., 2024) has shown that PESQ, the standard intrusive signal processing based metric to optimise towards in MetricGAN, is also susceptible to adversarial attack. This is not to say that optimisation towards SQ predictors as NNSE training objective is an approach which is entirely without merit or potential application; rather that it should be used with some caution and with proper systems of (human) evaluation in place. Further, it is the opinion of the author that if a metric is to be trusted as a means of evaluation it follows that it should also be trusted as a means of model optimisation. As such, the aim of future SQ prediction system research should be to improve robustness to the assessment of ‘out of domain‘ (i.e dissimilar to the training corpora) data.

This problem of metrics is compounded by the related issue of common datasets. The most widely used common dataset for the single channel SE task which has been used throughout this work is VoiceBank-DEMAND (Valentini-Botinhao et al., 2016). Its popularity is seemingly sustained in part by its relatively small size and ease of availability but also by simple inertia; since every other publication benchmarks its proposed system on the dataset, any newly proposed system must also be benchmarked for a fair comparison. For example, the SUPERB (Yang et al., 2021) universal

speech benchmark suite includes VoiceBank-DEMAND the SE benchmark component. VoiceBank-DEMAND's ubiquity is bolstered further by the simple fact that it is easy to obtain impressive test time performance due to the fact that the testset is significantly *easier* to enhance than the trainset. Another less commonly discussed problem with VoiceBank-DEMAND is the lack of a defined validation set; this renders many comparisons between systems unreliable. For example, in (Cao et al., 2022), the authors compare their proposed system which is trained over the entirety of the trainset with results from (Fu, Yu, Hsieh, et al., 2021) where 10% (roughly 1000 samples) of the trainset were excluded for validation. This example has been reconciled in Table 4.9 in Chapter 4, but this is generally a time-consuming and costly practise and untenable for the rapidly changing landscape of NNSE research. Yet another issue with VoiceBank-DEMAND relates to its reproducibility or lack thereof. While all of the source audio and MATLAB code used to generate the dataset is available, the portion of the 10 minute DEMAND noise files which was selected to be mixed with the 3 - 10 second VoiceBank clean speech audio was selected *randomly* with no defined seed. As such it is *impossible* (within a anthropological timespan) to directly resimulate VoiceBank-DEMAND and obtain an identical dataset. Finally, there is the issue of the content of the VoiceBank audio. It consists of English read speech by a number of people (male and female identifying) from across the British isles, primarily with English or Scottish regional accents. The text the speakers are reading is sourced primarily from contemporary (at time of recording) newspapers from the same regions, as well as *phonetically rich* text such as the 'Rainbow Passage' (which has been criticised in recent years (Dietsch et al., 2023)). The set of speakers, even with the diversity in their manner of speech and accent is representative of a vanishingly small proportion of native English speakers globally, let alone speakers of other languages. Likewise, the linguistic content of the speech which relates mainly to late 1990s and early 2000s British politicians, sports teams and current events holds little relevance to the the world of today.

In this work, there have been attempts to design a successor to, or improvement on VoiceBank-DEMAND, firstly in the more acoustically realistic 'rerecorded' dataset proposed in Chapter 4 and more concretely in the CommonVoice-DEMAND dataset framework proposed in Chapter 8. Future research expanding on these attempts to build a new common consensus dataset for the NNSE task is believed to be of critical importance.

The use of SSSR derived representations in loss functions as proposed in Chapter 7 and Chapter 8 present a number of possibilities for future research. At time of writing, a very recent work (Babaev et al., 2024) has found that by combining an SSSR encoder loss with a very small weighting of a traditional STFT loss term, superior enhancement performance can be achieved. Another potentially promising course might be to design a fine-tuning task for the SSSR representation which further enhances its usefulness in SE loss functions. Further, intermediate features derived from large, weakly supervised ASR systems such as Whisper could be used in SE loss functions in the same manner. Another angle is the application of the SSSR encoder loss to other tasks such as speech synthesis (Shi et al., 2024) or voice conversion (Sadov et al., 2023).

Bibliography

Abdulatif, S., Armanious, K., Sajeev, J. T., Guirguis, K., & Yang, B. (2021). Investigating Cross-Domain Losses for Speech Enhancement. *2021 29th European Signal Processing Conference (EUSIPCO)*, 411–415. https://doi.org/10.23919/EUSIPCO54536.2021.9616267 (cf. p. 65)

Andersen, A. H., de Haan, J. M., Tan, Z.-H., & Jensen, J. (2018). Refinement and validation of the binaural short time objective intelligibility measure for spatially diverse conditions. *Speech Communication*, *102*, 1–13. https://doi.org/https://doi.org/10.1016/j.specom.2018.06.001 (cf. p. 39)

Ardila, R., Branson, M., Davis, K., Henretty, M., Kohler, M., Meyer, J., Morais, R., Saunders, L., Tyers, F. M., & Weber, G. (2020). Common Voice: A Massively-Multilingual Speech Corpus. *Proc. Conf. on Language Resources and Evaluation* (cf. pp. 45, 100).

Azadi, S., Fisher, M., Kim, V. G., Wang, Z., Shechtman, E., & Darrell, T. (2018). Multi-Content GAN for Few-Shot Font Style Transfer. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cf. p. 36).

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer Normalization. https://arxiv.org/abs/1607.06450. (Cf. p. 30)

Babaev, N., Tamogashev, K., Saginbaev, A., Shchekotov, I., Bae, H., Sung, H., Lee, W., Cho, H.-Y., & Andreev, P. (2024). FINALLY: fast and universal speech enhancement with studio-like quality. https://arxiv.org/abs/2410.05920. (Cf. pp. 2, 146)

Babu, A., Wang, C., Tjandra, A., Lakhotia, K., Xu, Q., Goyal, N., Singh, K., von Platen, P., Saraf, Y., Pino, J., Baevski, A., Conneau, A., & Auli, M. (2022). XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale. *Proc. Interspeech 2022*, 2278–2282. https://doi.org/10.21437/Interspeech.2022-143 (cf. p. 33)

Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems* (pp. 12449–12460). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf. (Cf. pp. 31, 33)

Bagchi, D., Plantinga, P., Stiff, A., & Fosler-Lussier, E. (2018). Spectral feature mapping with mimic loss for robust speech recognition (cf. p. 29).

Bahdanau, D., Cho, K., & Bengio, Y. Neural machine translation by jointly learning to align and translate [3rd International Conference on Learning Representations, ICLR 2015 ; Conference date: 07-05-2015 Through 09-05-2015]. English (US). In: 3rd International Conference on Learning Representations, ICLR 2015 ; Conference date: 07-05-2015 Through 09-05-2015. 2015, January (cf. p. 22).

Barker, J., Marxer, R., Vincent, E., & Watanabe, S. (2015). The third CHiME speech separation and recognition challenge: dataset, task and baselines. *Proc. IEEE ASRU 2015*, 504–511 (cf. p. 44).

Barker, J., Akeroyd, M., Bailey, W., Cox, T. J., Culling, J. F., Firth, J., Graetzer, S., & Naylor, G. (2024). The 2nd Clarity Prediction Challenge: A machine learning challenge for hearing aid intelligibility prediction. *ICASSP* (cf. p. 6).

Barker, J., Akeroyd, M., Cox, T. J., Culling, J. F., Firth, J., Graetzer, S., Griffiths, H., Harris, L., Naylor, G., Podwinska, Z., Porter, E., & Munoz, R. V. (2022). The 1st Clarity Prediction Challenge: A machine learning challenge for hearing aid intelligibility prediction. *Proc. Interspeech 2022*, 3508–3512. https://doi.org/10.21437/Interspeech.2022-10821 (cf. pp. 45, 111, 116)

Barker, J., Watanabe, S., Vincent, E., & Trmal, J. (2018). The Fifth 'CHiME' Speech Separation and Recognition Challenge: Dataset, Task and Baselines, 1561–1565. https://doi.org/10.21437/Interspeech.2018-1768 (cf. p. 45)

Becerra, H., Ragano, A., & Hines, A. (2022). Exploring the influence of fine-tuning data on wav2vec 2.0 model for blind speech quality prediction. *Proc. Interspeech 2022*, 4088–4092. https://doi.org/10.21437/Interspeech.2022-10766 (cf. p. 115)

Beerends, J., Schmidmer, C., Berger, J., Obermann, M., Ullmann, R., Pomy, J., & Keyhl, M. (2013). Perceptual Objective Listening Quality Assessment (POLQA), The Third Generation ITU-T Standard for End-to-End Speech Quality Measurement Part I-Temporal Alignment. *AES: Journal of the Audio Engineering Society*, *61*, 366–384 (cf. p. 40).

Bella, G., Batsuren, K., & Giunchiglia, F. (2021). A Database and Visualization of the Similarity of Contemporary Lexicons. In K. Ekštein, F. Pártl, & M. Konopík (Eds.), *Text, Speech, and Dialogue* (pp. 95–104). Springer International Publishing. (Cf. p. 104).

Bengio, Y. (2009). Learning Deep Architectures for AI. *Foundations*, *2*, 1–55. https://doi.org/10.1561/2200000006 (cf. pp. 2, 14)

Braun, S., & Tashev, I. (2020). A consolidated view of loss functions for supervised deep learning-based speech enhancement. https://doi.org/10.48550/ARXIV.2009.12286. (Cf. p. 66)

Bulut, A., & Koishida, K. (2020). Low-Latency Single Channel Speech Enhancement Using U-Net Convolutional Neural Networks. *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6214–6218. https://doi.org/10.1109/ICASSP40776.2020.9054563 (cf. p. 2)

Cao, R., Abdulatif, S., & Yang, B. (2022). CMGAN: Conformer-based Metric GAN for Speech Enhancement. https://doi.org/10.48550/ARXIV.2203.15149. (Cf. pp. 15, 51, 65, 66, 77, 78, 80, 88, 146)

Chai, L., Du, J., & Lee, C.-H. (2018). Acoustics-guided evaluation (AGE): a new measure for estimating performance of speech enhancement algorithms for robust ASR. *ArXiv*, *abs/1811.11517* (cf. p. 29).

Chao, R., Cheng, W.-H., Quatra, M. L., Siniscalchi, S. M., Yang, C.-H. H., Fu, S.-W., & Tsao, Y. (2024). An Investigation of Incorporating Mamba for Speech Enhancement. (Cf. pp. 27, 127).

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Li, J., Kanda, N., Yoshioka, T., Xiao, X., Wu, J., Zhou, L., Ren, S., Qian, Y., Qian, Y., Zeng, M., Yu, X., & Wei, F. (2022). WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing. *IEEE J. Sel. Topics in Signal Processing*, *16*, 1–14. https://doi.org/10.1109/JSTSP.2022.3188113 (cf. pp. 33, 104)

Close, G., Hain, T., & Goetze, S. (2022). MetricGAN+/-: Increasing Robustness of Noise Reduction on Unseen Data. *EUSIPCO 2022* (cf. pp. 5, 6, 41, 70, 88, 110).

Close, G., Hain, T., & Goetze, S. (2023a). The Effect of Spoken Language on Speech Enhancement using Self-Supervised Speech Representation Loss Functions. *Proc. WASPAA* (cf. pp. 5, 6, 133).

Close, G., Hain, T., & Goetze, S. (2024). Hallucination in Perceptual Metric-Driven Speech Enhancement Networks. *2024 32nd European Signal Processing Conference (EUSIPCO)*, 21–25 (cf. pp. 5, 7).

Close, G., Hain, T., & Goetze, S. (2023b). Non Intrusive Intelligibility Predictor for Hearing Impaired Individuals using Self Supervised Speech Representations. *Proc. Workshop on Speech Foundation Models and their Performance Benchmarks (SPARKS), ASRU sattelite workshop* (cf. pp. 5, 6).

Close, G., Hain, T., & Goetze, S. (2023c). PAMGAN+/-: Improving Phase-aware Speech Enhancement Performance via Expanded Discriminator Training. *journal of the audio engineering society*, (10656) (cf. pp. 5, 6).

Close, G., Hain, T., & Goetze, S. (2025). WhiSQA: Non-Intrusive Speech Quality Prediction Using Foundation Model Features. *Submitted to IEEE Transactions on Audio, Speech, and Language Processing* (cf. pp. 5, 6, 41).

Close, G., Hollands, S., Goetze, S., & Hain, T. (2022). Non-intrusive Speech Intelligibility Metric Prediction for Hearing Impaired Individuals. *Proc. Interspeech 2022*, 3483–3487. https://doi.org/10.21437/Interspeech.2022-10182 (cf. pp. 5, 6, 119)

Close, G., Ravenscroft, W., Hain, T., & Goetze, S. (2023a). CMGAN+/+: The University of Sheffield CHiME-7 UDASE Challenge Speech Enhancement System (cf. pp. 5, 6).

Close, G., Ravenscroft, W., Hain, T., & Goetze, S. (2024). Multi-CMGAN+/+: Leveraging Multi-Objective Speech Quality Metric Prediction for Speech Enhancement. *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 351–355. https://doi.org/10.1109/ICASSP48485.2024.10448343 (cf. pp. 5, 6)

Close, G., Ravenscroft, W., Hain, T., & Goetze, S. (2023b). Perceive and predict: self-supervised speech representation based loss functions for speech enhancement. *Proc. ICASSP 2023* (cf. pp. 5, 6, 103, 125, 133).

Cooper, E., Huang, W.-C., Toda, T., & Yamagishi, J. (2022). Generalization Ability of MOS Prediction Networks. *Proc. ICASSP* (cf. pp. 81, 89, 115).

Cosentino, J., Pariente, M., Cornell, S., Deleforge, A., & Vincent, E. (2020). LibriMix: An Open-Source Dataset for Generalizable Speech Separation. (Cf. pp. 45, 82, 90).

Dang, F., Chen, H., & Zhang, P. (2022). DPT-FSNet: Dual-Path Transformer Based Full-Band and Sub-Band Fusion Network for Speech Enhancement. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6857–6861. https://doi.org/10.1109/ICASSP43922.2022.9746171 (cf. pp. 15, 54, 137, 138)

Dauphin, Y. N., Fan, A., Auli, M., & Grangier, D. (2017). Language modeling with gated convolutional networks. *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, 933–941 (cf. p. 29).

David, J., E. E., & McDonald, H. S. (1956). Note on Pitch-Synchronous Processing of Speech. *The Journal of the Acoustical Society of America*, *28*(6), 1261–1266. https://doi.org/10.1121/1.1908613 (cf. p. 2)

Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *28*(4), 357–366. https://doi.org/10.1109/TASSP.1980.1163420 (cf. p. 32)

Defossez, A., Synnaeve, G., & Adi, Y. (2020). Real Time Speech Enhancement in the Waveform Domain. (Cf. p. 2).

Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2020). *Mathematics for Machine Learning*. Cambridge University Press. (Cf. p. 18).

de Oliveira, D., Grinstein, E., Naylor, P. A., & Gerkmann, T. (2024). LASER: Language-Queried Speech Enhancer. *2024 18th International Workshop on Acoustic Signal Enhancement (IWAENC)*, 90–94. https://doi.org/10.1109/IWAENC61483.2024.10694503 (cf. p. 2)

de Oliveira, D., Welker, S., Richter, J., & Gerkmann, T. (2024). The PESQetarian: On the Relevance of Goodhart's Law for Speech Enhancement. https://arxiv.org/abs/2406.03460. (Cf. p. 145)

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proc. of ACL 2019*, 4171–4186. https://doi.org/10.18653/v1/N19-1423 (cf. pp. 32, 33)

Dieleman, S., Oord, A. v. d., & Simonyan, K. (2018). The challenge of realistic music generation: modelling raw audio at scale. *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 8000–8010 (cf. p. 32).

Dietsch, A. M., Mocarski, R., Hope, D. A., Woodruff, N., & McKelvey, M. (2023). Revisiting the Rainbow: Culturally Responsive Updates to a Standard Clinical Resource. *American Journal of Speech-Language Pathology*, *32*(1), 377–380. https://doi.org/10.1044/2022_AJSLP-22-00215 (cf. p. 146)

Dong, X., & Williamson, D. S. (2020). A pyramid recurrent network for predicting crowdsourced speech-quality ratings of real-world signals. *Interspeech*, 4631–4635 (cf. pp. 46, 47).

Edo Zezario, R., Chen, F., Fuh, C.-S., Wang, H.-M., & Tsao, Y. (2022). MBI-Net: A Non-Intrusive Multi-Branched Speech Intelligibility Prediction Model for Hearing Aids. *Proc. Interspeech 2022*, 3944–3948. https://doi.org/10.21437/Interspeech.2022-10838 (cf. pp. 115, 119–121)

Euler, L. (1770). On the sum of series involving the Bernoulli numbers [Written in 1768. Published in Opera Omnia, Series 1, Volume 15, pp. 91-130.]. *Novi Commentarii academiae scientiarum Petropolitanae*, *14*, 129–167 (cf. p. 28).

Falk, T. H., Parsa, V., Santos, J. F., Arehart, K., Hazrati, O., Huber, R., Kates, J. M., & Scollie, S. (2015). Objective quality and intelligibility prediction for users of assistive listening devices: Advantages and limitations of existing tools. *IEEE signal processing magazine*, *32*(2), 114–124 (cf. p. 108).

Feng, Y., & Chen, F. (2022). Nonintrusive objective measurement of speech intelligibility: A review of methodology. *Biomedical Signal Processing and Control*, *71*, 103204 (cf. p. 108).

Fleischman, E. W. (1998). WAVE and AVI Codec Registries. https://doi.org/10.17487/RFC2361. (Cf. p. 11)

Fu, S.-W., Liao, C.-F., Tsao, Y., & Lin, S.-D. (2019). MetricGAN: Generative Adversarial Networks based Black-box Metric Scores Optimization for Speech Enhancement. (Cf. pp. 2, 15, 41, 65, 78).

Fu, S.-W., Tsao, Y., Hwang, H.-T., & Wang, H.-M. (2018). Quality-Net: An End-to-End Non-intrusive Speech Quality Assessment Model based on BLSTM. (Cf. p. 41).

Fu, S.-W., Wang, T.-W., Tsao, Y., Lu, X., & Kawai, H. (2018). End-to-End Waveform Utterance Enhancement for Direct Evaluation Metrics Optimization by Fully Convolutional Neural Networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(9), 1570–1584. https://doi.org/10.1109/TASLP.2018.2821903 (cf. pp. 16, 39, 96, 98, 103)

Fu, S.-W., Yu, C., Hsieh, T.-A., Plantinga, P., Ravanelli, M., Lu, X., & Tsao, Y. (2021). MetricGAN+: An Improved Version of MetricGAN for Speech Enhancement. *Proc. Interspeech 2021*, 201–205. https://doi.org/10.21437/Interspeech.2021-599 (cf. pp. 6, 41, 49, 58, 61, 65, 67, 80, 88, 146)

Fu, S.-W., Yu, C., Hung, K.-H., Ravanelli, M., & Tsao, Y. (2021). MetricGAN-U: Unsupervised speech enhancement/ dereverberation based only on noisy/ reverberated speech (cf. pp. 2, 6, 41, 81, 135).

Fukushima, K. (1969). Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements. *IEEE Transactions on Systems Science and Cybernetics*, *5*(4), 322–333. https://doi.org/10.1109/TSSC.1969.300225 (cf. p. 28)

Garofolo, J., Lamel, L., Fisher, W., Fiscus, J., & Pallett, D. (1993). DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. *NASA STI/Recon Technical Report N*, *93*, 27403 (cf. p. 16).

Goetze, S., Warzybok, A., Kodrasi, I., Jungmann, J., Cauchi, B., Rennies, J., Habets, E., Mertins, A., Gerkmann, T., Doclo, S., & Kollmeier, B. (2014). A Study on Speech Quality and Speech Intelligibility Measures for Quality Assessment of Single-Channel Dereverberation Algorithms. *Proc. Int. Workshop on Acoustic Signal Enhancement (IWAENC 2014)*. https://doi.org/10.1109/IWAENC.2014.6954293 (cf. p. 29)

Gonzalez, P., Tan, Z.-H., Østergaard, J., Jensen, J., Alstrøm, T. S., & May, T. (2024). Investigating the Design Space of Diffusion Models for Speech Enhancement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *32*, 4486–4500. https://doi.org/10.1109/TASLP.2024.3473319 (cf. p. 145)

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Networks. https://doi.org/10.48550/ARXIV.1406.2661. (Cf. p. 35)

Graetzer, S., Akeroyd, M., Barker, J. P., Cox, T. J., Culling, J. F., Naylor, G., Porter, E., & Muñoz, R. V. (2020). Clarity: Machine Learning Challenges to Revolutionise Hearing Device Processing. https://doi.org/10.48550/ARXIV.2006.11140. (Cf. p. 6)

Graetzer, S., Akeroyd, M. A., Barker, J., Cox, T. J., Culling, J. F., Naylor, G., Porter, E., & Viveros-Muñoz, R. (2022). Dataset of British English speech recordings for psychoacoustics and speech processing research: The clarity speech corpus. *Data in Brief*, *41*, 107951. https://doi.org/https://doi.org/10.1016/j.dib.2022.107951 (cf. p. 111)

Graetzer, S., Barker, J., Cox, T. J., Akeroyd, M., Culling, J. F., Naylor, G., Porter, E., & Muñoz, R. V. (2021). Clarity-2021 Challenges: Machine Learning Challenges for Advancing Hearing Aid Processing. *Proc. Interspeech 2021*, 686–690. https://doi.org/10.21437/Interspeech.2021-1574 (cf. p. 108)

Gu, A., & Dao, T. (2023). Mamba: Linear-Time Sequence Modeling with Selective State Spaces. (Cf. pp. 26, 125, 126).

Gu, A., Dao, T., Ermon, S., Rudra, A., & Ré, C. (2020). HiPPO: Recurrent Memory with Optimal Polynomial Projections. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in Neural Information Processing Systems* (pp. 1474–1487). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2020/file/102f0bb6efb3a6128a3c750dd16729be-Paper.pdf. (Cf. p. 26)

Gu, A., Goel, K., & Ré, C. (2022). Efficiently Modeling Long Sequences with Structured State Spaces. (Cf. pp. 25, 126).

Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., & Pang, R. (2020). Conformer: Convolution-augmented Transformer for Speech Recognition. (Cf. pp. 24, 80).

Gumbel, E. (1954). *Statistical Theory of Extreme Values and Some Practical Applications: A Series of Lectures*. U.S. Government Printing Office. (Cf. p. 32).

Haeb-Umbach, R., Heymann, J., Drude, L., Watanabe, S., Delcroix, M., & Nakatani, T. (2021). Far-Field Automatic Speech Recognition. *Proceedings of the IEEE*, *109*(2), 124–148. https://doi.org/10.1109/JPROC.2020.3018668 (cf. p. 2)

Hashmi, A. (2021). Perceptual Evaluation of Speech Quality for Inexpensive Recording Equipment. *Acoustics*, *3*(1), 200–211. https://doi.org/10.3390/acoustics3010014 (cf. p. 47)

Haykin, S. (2009). *Neural Networks and Learning Machines*. Prentice Hall. https://books.google.co.uk/books?id=K7P36lKzI_QC. (Cf. p. 27)

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, 1026–1034. https://doi.org/10.1109/ICCV.2015.123 (cf. p. 28)

Hendriks, R., Gerkmann, T., & Jensen, J. (2013). DFT-Domain Based Single-Microphone Noise Reduction for Speech Enhancement: A Survey of the State of the Art. *Synthesis Lectures on Speech and Audio Processing*, *9*, 1–80. https://doi.org/10.2200/S00473ED1V01Y201301SAP011 (cf. pp. 2, 12)

Hendrycks, D., & Gimpel, K. (2016). Gaussian Error Linear Units (GELUs). (Cf. p. 29).

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Comput.*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735 (cf. p. 20)

Hsieh, T.-A., Yu, C., Fu, S.-W., Lu, X., & Tsao, Y. (2020). Improving Perceptual Quality by Phone-Fortified Perceptual Loss using Wasserstein Distance for Speech Enhancement. https://doi.org/10.48550/ARXIV.2010.15174. (Cf. p. 33)

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., & Mohamed, A. (2021). HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. https://doi.org/10.48550/ARXIV.2106.07447. (Cf. pp. 32, 33, 79)

Hua, B.-S., Tran, M.-K., & Yeung, S.-K. (2018). Pointwise Convolutional Neural Networks. https://arxiv.org/abs/1712.05245. (Cf. p. 24)

Huang, G., Liu, Z., Maaten, L. V. D., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2261–2269. https://doi.org/10.1109/CVPR.2017.243 (cf. p. 51)

Huber, R., & Kollmeier, B. (2006). PEMO-Q—A New Method for Objective Audio Quality Assessment Using a Model of Auditory Perception. *IEEE Transactions on Audio, Speech, and Language Processing*, *14*(6), 1902–1911. https://doi.org/10.1109/TASL.2006.883259 (cf. p. 40)

International Telecommunication Union. (2015). *Recommendation ITU-R BS.1534-3 Method for the Subjective Assessment of Intermediate Quality Level of Audio Systems* (ITU-R Recommendation). ITU. https://www.itu.int/rec/R-REC-BS.1534-3-201510-I. (Cf. pp. 38, 47)

Subjective test methodology for evaluating speech communication systems that include noise suppression algorithm. (Standard). (2003). International Telecommunication Union (ITU). (Cf. pp. 38, 139).

P.10 : Vocabulary for performance, quality of service and quality of experience (Standard). (2017). International Telecommunication Union. (Cf. p. 38).

Jang, J., & Koo, M.-W. (2023). The SGU Systems for the CHiME-7 UDASE Challenge. *7th International Workshop on Speech Processing in Everyday Environments (CHiME 2023)*, 39–44. https://doi.org/10.21437/CHiME.2023-8 (cf. p. 85)

Jensen, J., & Taal, C. H. (2016). An Algorithm for Predicting the Intelligibility of Speech Masked by Modulated Noise Maskers. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *24*(11), 2009–2022. https://doi.org/10.1109/TASLP.2016.2585878 (cf. p. 39)

K. A. Reddy, C., Gopal, V., & Cutler, R. (2020). DNSMOS: A Non-Intrusive Perceptual Objective Speech Quality metric to evaluate Noise Suppressors. *2020 International Conference on Acoustics, Speech, and Signal Processing*, 6493–6497. https://www.microsoft.com/en-us/research/publication/dnsmos-a-non-intrusive-perceptual-objective-speech-quality-metric-to-evaluate-noise-suppressors/ (cf. pp. 40, 41)

Kates, J. M., & Arehart, K. H. (2014). The Hearing-Aid Speech Perception Index (HASPI). *Speech Communication*, *65*, 75–93. https://doi.org/https://doi.org/10.1016/j.specom.2014.06.002 (cf. pp. 37, 40)

Kingma, D., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations* (cf. pp. 60, 70, 96, 102, 112, 128, 138).

Kleene, S. C. (1956). Representation of Events in Nerve Nets and Finite Automata. In C. E. Shannon & J. McCarthy (Eds.), *Automata Studies* (pp. 3–42). Princeton University Press. https://doi.org/doi:10.1515/9781400882618-002. (Cf. p. 16)

Kollmeier, B., Warzybok, A., Hochmuth, S., Zokoll, M., Uslar, V., Brand, T., & Wagener, K. (2015). The multilingual matrix test: Principles, applications, and comparison across languages: A review. *International Journal of Audiology*, *54*(sup2), 3–16. https://doi.org/10.3109/14992027.2015.1020971 (cf. p. 111)

Kounovsky, T., & Malek, J. (2017). Single channel speech enhancement using convolutional neural network. *2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)*, 1–5. https://doi.org/10.1109/ECMSM.2017.7945915 (cf. pp. 2, 15)

Kumar, A., Tan, K., Ni, Z., Manocha, P., Zhang, X., Henderson, E., & Xu, B. (2023). TorchAudio-Squim: Reference-less Speech Quality and Intelligibility measures in TorchAudio. (Cf. pp. 3, 41, 135).

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, *1*(4), 541–551. https://doi.org/10.1162/neco.1989.1.4.541 (cf. p. 19)

Lee, A., Gong, H., Duquenne, P.-A., Schwenk, H., Chen, P.-J., Wang, C., Popuri, S., Adi, Y., Pino, J., Gu, J., & Hsu, W.-N. (2022). Textless Speech-to-Speech Translation on Real Data. *ACL*. https://doi.org/10.18653/v1/2022.naacl-main.63 (cf. p. 33)

Leglaive, S., Borne, L., Tzinis, E., Sadeghi, M., Fraticelli, M., Wisdom, S., Pariente, M., Pressnitzer, D., & Hershey, J. R. (2023). The CHiME-7 UDASE task: Unsupervised domain adaptation for conversational speech enhancement. (Cf. pp. 40, 45, 82, 85, 90, 135, 139, 141).

Leglaive, S., Fraticelli, M., ElGhazaly, H., Borne, L., Sadeghi, M., Wisdom, S., Pariente, M., Hershey, J. R., Pressnitzer, D., & Barker, J. P. (2024). Objective and subjective evaluation of speech enhancement methods in the UDASE task of the 7th CHiME challenge. (Cf. pp. 45, 135, 138).

Li, A., Liu, W., Zheng, C., & Li, X. (2021). Embedding and Beamforming: All-neural Causal Beamformer for Multichannel Speech Enhancement. (Cf. p. 2).

Lim, J. S., & Oppenheim, A. V. (1978). All-pole modeling of degraded speech. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, *26*, 197–210 (cf. p. 2).

Lin, Z., Zhou, L., & Qiu, X. (2019). A composite objective measure on subjective evaluation of speech enhancement algorithms. *Applied Acoustics*, *145*, 144–148. https://doi.org/10.1016/j.apacoust.2018.10.002 (cf. pp. 37, 40, 61, 70, 95, 103, 138)

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, *28*(2), 129–137. https://doi.org/10.1109/TIT.1982.1056489 (cf. p. 32)

Lu, X., Tsao, Y., Matsuda, S., & Hori, C. (2013). Speech enhancement based on deep denoising Auto-Encoder. *Proc. Interspeech*, 436–440 (cf. p. 16).

Lu, Y., Li, Z., He, D., Sun, Z., Dong, B., Qin, T., Wang, L., & Liu, T.-Y. (2019). Understanding and Improving Transformer From a Multi-Particle Dynamic System Point of View. https://arxiv.org/abs/1906.02762. (Cf. p. 24)

Luo, Y., & Mesgarani, N. (2018). TasNet: Time-Domain Audio Separation Network for Real-Time, Single-Channel Speech Separation. *ICASSP 2018*, 696–700. https://doi.org/10.1109/ICASSP.2018.8462116 (cf. p. 98)

Luo, Y., & Mesgarani, N. (2019). Conv-TasNet: Surpassing Ideal Time–Frequency Magnitude Masking for Speech Separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *27*(8), 1256–1266. https://doi.org/10.1109/taslp.2019.2915167 (cf. p. 19)

Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing* (cf. pp. 28, 51).

Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., & Smolley, S. P. (2016). Least Squares Generative Adversarial Networks. https://doi.org/10.48550/ARXIV.1611.04076. (Cf. p. 15)

Martín-Doñas, J., Gomez, A., Gonzalez Lopez, J., & Peinado, A. (2018). A Deep Learning Loss Function Based on the Perceptual Evaluation of the Speech Quality. *IEEE Signal Processing Letters*, *PP*, 1–1. https://doi.org/10.1109/LSP.2018.2871419 (cf. p. 40)

McKinney, A. F., & Cauchi, B. (2022). Non-Intrusive Binaural Speech Intelligibility Prediction From Discrete Latent Representations. *IEEE Signal Processing Letters*, *29*, 987–991. https://doi.org/10.1109/lsp.2022.3161115 (cf. pp. 119–121)

Mittag, G., Cutler, R., Hosseinkashi, Y., Revow, M., Srinivasan, S., Chande, N., & Aichner, R. (2020). DNN No-Reference PSTN Speech Quality Prediction. *Proc. Interspeech 2020* (cf. pp. 46, 47, 136).

Mittag, G., Naderi, B., Chehadi, A., & Möller, S. (2021). NISQA: A Deep CNN-Self-Attention Model for Multidimensional Speech Quality Prediction with Crowdsourced Datasets. *Interspeech 2021*. https://doi.org/10.21437/interspeech.2021-299 (cf. pp. 3, 41, 46, 47, 95, 115, 118, 128–131, 136)

Mogridge, R., Close, G., Sutherland, R., Hain, T., Barker, J., Goetze, S., & Ragni, A. (2024). Non-Intrusive Speech Intelligibility Prediction for Hearing-Impaired Users using Intermediate ASR Features and Human Memory Models. *Proc. ICASSP 2024* (cf. pp. 5, 6, 35, 136).

Nejime, Y., & Moore, B. C. J. (1997). Simulation of the effect of threshold elevation and loudness recruitment combined with reduced frequency selectivity on the intelligibility of speech in noise. *The Journal of the Acoustical Society of America*, *102*(1), 603–615. https://doi.org/10.1121/1.419733 (cf. p. 109)

Oliver, B., Pierce, J., & Shannon, C. (1948). The Philosophy of PCM. *Proceedings of the IRE*, *36*(11), 1324–1331. https://doi.org/10.1109/JRPROC.1948.231941 (cf. p. 11)

O'Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. https://arxiv.org/abs/1511.08458. (Cf. p. 19)

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: An ASR corpus based on public domain audio books. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5206–5210. https://doi.org/10.1109/ICASSP.2015.7178964 (cf. pp. 33, 79, 82, 90)

Pasad, A., Shi, B., & Livescu, K. (2023). Comparative Layer-Wise Analysis of Self-Supervised Speech Models. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*. https://doi.org/10.1109/ICASSP49357.2023.10096149 (cf. p. 125)

Pascual, S., Bonafonte, A., & Serrà, J. (2017). SEGAN: Speech Enhancement Generative Adversarial Network. *Proc. of Interspeech*, 3642–3646 (cf. pp. 15, 61).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems 32* (pp. 8024–8035). Curran Associates, Inc. (Cf. p. 111).

Paul, D. B., & Baker, J. M. (1992). The Design for the Wall Street Journal-Based CSR Corpus. *Proceedings of the Workshop on Speech and Natural Language*, 357–362. https://doi.org/10.3115/1075527.1075614 (cf. p. 44)

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust Speech Recognition via Large-Scale Weak Supervision. (Cf. pp. 6, 33, 34, 136).

Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for Activation Functions. https://arxiv.org/abs/1710.05941. (Cf. pp. 24, 28)

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., & Sutskever, I. (2021). Zero-Shot Text-to-Image Generation. https://doi.org/10.48550/ARXIV.2102.12092. (Cf. p. 36)

Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., Chou, J.-C., Yeh, S.-L., Fu, S.-W., Liao, C.-F., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., … Bengio, Y. (2021). SpeechBrain: A General-Purpose Speech Toolkit. (Cf. pp. 60, 61, 97, 102, 111).

Ravenscroft, W., Goetze, S., & Hain, T. (2022). Att-TasNet: Attending to Encodings in Time-Domain Audio Speech Separation of Noisy, Reverberant Speech Mixtures. *Frontiers in Signal Processing*. https://doi.org/10.3389/frsip.2022.856968 (cf. pp. 2, 32, 99)

Ravenscroft, W., Close, G., Goetze, S., Hain, T., Soleymanpour, M., Chowdhury, A., & Fuhs, M. C. (2024). Transcription-Free Fine-Tuning of Speech Separation Models for Noisy and Reverberant Multi-Speaker Automatic Speech Recognition. *Interspeech 2024*, 4998–5002. https://doi.org/10.21437/Interspeech.2024-1264 (cf. pp. 5, 6)

Ravenscroft, W., Goetze, S., & Hain, T. (2023). On Data Sampling Strategies for Training Neural Network Speech Separation Models. *EUSIPCO 2023* (cf. p. 89).

Reddy, C. K. A., Gopal, V., & Cutler, R. (2022). DNSMOS P.835: A Non-Intrusive Perceptual Objective Speech Quality Metric to Evaluate Noise Suppressors. (Cf. pp. 3, 81, 135, 138).

Richey, C., Barrios, M., Armstrong, Z., Bartels, C., Franco, H., Graciarena, M., Lawson, A., Nandwana, M., Stauffer, A., Hout, J., Gamble, P., Hetherly, J., Stephenson, C., & Ni, K. (2018). Voices Obscured in Complex Environmental Settings (VOICES) corpus (cf. p. 47).

Richter, J., de Oliveira, D., & Gerkmann, T. (2024). Investigating Training Objectives for Generative Speech Enhancement. https://arxiv.org/abs/2409.10753. (Cf. p. 145)

Richter, J., Welker, S., Lemercier, J.-M., Lay, B., & Gerkmann, T. (2023). Speech Enhancement and Dereverberation With Diffusion-Based Generative Models. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *31*, 2351–2364. https://doi.org/10.1109/TASLP.2023.3285241 (cf. p. 145)

Richter, J., Welker, S., Lemercier, J.-M., Lay, B., Peer, T., & Gerkmann, T. (2024). Causal Diffusion Models for Generalized Speech Enhancement. *IEEE Open Journal of Signal Processing*, *5*, 780–789. https://doi.org/10.1109/OJSP.2024.3379070 (cf. p. 2)

Rix, A., Beerends, J., Hollier, M., & Hekstra, A. (2001). Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, *2*, 749–752 vol.2. https://doi.org/10.1109/ICASSP.2001.941023 (cf. pp. 36, 39, 70, 95, 102, 103)

Roux, J. L., Wisdom, S., Erdogan, H., & Hershey, J. R. (2018). SDR - half-baked or well done? (Cf. pp. 38, 70, 82, 89, 103, 138).

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536. https://api.semanticscholar.org/CorpusID:205001834 (cf. p. 2)

Rumelhart, D. E., & McClelland, J. L. (1987). Learning Internal Representations by Error Propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations* (pp. 318–362). (Cf. p. 20).

Sadov, K., Hutter, M., & Near, A. (2023). Low-latency Real-time Voice Conversion on CPU. https://arxiv.org/abs/2311.00873. (Cf. p. 146)

Santiago Cuervo, Ricard Marxer. (2024). Temporal-hierarchical features from noise-robust speech foundation models for non-intrusive intelligibility prediction. *Clarity Workshop 2022*. https://claritychallenge.org/clarity2023-workshop/papers/CPC2_E011_report.pdf (cf. pp. 35, 136)

Sector, I. T. U. T. S. (2011). *Objective measurement of active speech level* [ITU-T Recommendation P.56]. (Cf. p. 43).

Shen, K., Yan, D., & Dong, L. (2023). MSQAT: A multi-dimension non-intrusive speech quality assessment transformer utilizing self-supervised representations. *Applied Acoustics*, *212*, 109584. https://doi.org/https://doi.org/10.1016/j.apacoust.2023.109584 (cf. pp. 129–131, 136)

Shi, R., Bär, A., Sach, M., Tirry, W., & Fingscheidt, T. (2024). Non-Causal to Causal SSL-Supported Transfer Learning: Towards A High-Performance Low-Latency Speech Vocoder. *2024 18th International Workshop on Acoustic Signal Enhancement (IWAENC)*, 359–363. https://doi.org/10.1109/IWAENC61483.2024.10694644 (cf. p. 146)

Song, Y., Kim, D., Kang, H.-G., & Madhu, N. (2024). Spectrum-Aware Neural Vocoder Based on Self-Supervised Learning for Speech Enhancement. *2024 32nd European Signal Processing Conference (EUSIPCO)*, 16–20. https://doi.org/10.23919/EUSIPCO63174.2024.10715278 (cf. p. 145)

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*(56), 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html (cf. p. 24)

Stone, M. A., & Moore, B. C. J. (1999). Tolerable hearing aid delays. I. Estimation of limits imposed by the auditory path alone using simulated hearing losses. *Ear and hearing*, *20 3*, 182–92 (cf. p. 46).

Sun, L., Du, J., Dai, L.-R., & Lee, C.-H. (2017). Multiple-target deep learning for LSTM-RNN based speech enhancement. *2017 Hands-free Speech Communications and Microphone Arrays (HSCMA)*, 136–140. https://doi.org/10.1109/HSCMA.2017.7895577 (cf. p. 14)

Sun, T., Gong, S., Wang, Z., Smith, C. D., Wang, X., Xu, L., & Liu, J. (2021). Boosting the Intelligibility of Waveform Speech Enhancement Networks through Self-supervised Representations. *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 992–997. https://doi.org/10.1109/ICMLA52953.2021.00163 (cf. p. 2)

Sutherland, R., Close, G., Hain, T., Goetze, S., & Barker, J. (2024). Using Speech Foundational Models in Loss Functions for Hearing Aid Speech Enhancement. https://arxiv.org/abs/2407.13333. (Cf. pp. 5, 6)

Taal, C. H., Hendriks, R. C., Heusdens, R., & Jensen, J. (2011). An Algorithm for Intelligibility Prediction of Time–Frequency Weighted Noisy Speech. *IEEE Transactions on Audio,*

Speech, and Language Processing, *19*(7), 2125–2136. https://doi.org/10.1109/TASL.2011.2114881 (cf. pp. 37, 39, 70, 103)

Tal, O., Mandel, M., Kreuk, F., & Adi, Y. (2022). A Systematic Comparison of Phonetic Aware Techniques for Speech Enhancement. https://doi.org/10.48550/ARXIV.2206.11000. (Cf. p. 33)

Tamm, B., Balabin, H., Vandenberghe, R., & hamme, H. V. (2022). Pre-trained Speech Representations as Feature Extractors for Speech Quality Assessment in Online Conferencing Applications. *Interspeech 2022*. https://doi.org/10.21437/interspeech.2022-10147 (cf. pp. 41, 115, 117)

Tamm, B., Vandenberghe, R., & Van hamme, H. (2023). Analysis of XLS-R for Speech Quality Assessment, 1–5. https://doi.org/10.1109/WASPAA58266.2023.10248049 (cf. pp. 126, 130, 131, 136)

Thiemann, J., Ito, N., & Vincent, E. (2013). *DEMAND: a collection of multi-channel recordings of acoustic noise in diverse environments* (Version 1.0) [Supported by Inria under the Associate Team Program VERSAMUS]. Zenodo. https://doi.org/10.5281/zenodo.1227121. (Cf. p. 41)

Thireou, T., & Reczko, M. (2007). Bidirectional Long Short-Term Memory Networks for Predicting the Subcellular Localization of Eukaryotic Proteins. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, *4*, 441–6. https://doi.org/10.1109/tcbb.2007.1015 (cf. p. 21)

Tu, Z., Ma, N., & Barker, J. (2022). Unsupervised Uncertainty Measures of Automatic Speech Recognition for Non-intrusive Speech Intelligibility Prediction. *Proc. Interspeech 2022*, 3493–3497. https://doi.org/10.21437/Interspeech.2022-10408 (cf. pp. 119–121)

Tustin, A. (1947). A method of analysing the behaviour of linear systems in terms of time series. *Journal of the Institution of Electrical Engineers - Part IIA: Automatic Regulators and Servo Mechanisms*, *94*, 130–142. https://doi.org/10.1049/ji-2a.1947.0020 (cf. p. 26)

Tzinis, E., Adi, Y., Ithapu, V. K., Xu, B., Smaragdis, P., & Kumar, A. (2022). RemixIT: Continual Self-Training of Speech Enhancement Models via Bootstrapped Remixing. *IEEE Journal of Selected Topics in Signal Processing* (cf. pp. 83–85, 91).

Tzinis, E., Wang, Z., & Smaragdis, P. (2020). Sudo RM -RF: Efficient Networks for Universal Audio Source Separation. *MLSP 2020*. https://doi.org/10.1109/mlsp49062.2020.9231900 (cf. pp. 83, 84, 91)

Valentini-Botinhao, C., Wang, X., Takaki, S., & Yamagishi, J. (2016). Investigating RNN-based speech enhancement methods for noise-robust Text-to-Speech. *SSW* (cf. pp. 11, 13, 41, 58, 68, 95, 96, 138, 145).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. (Cf. pp. 3, 22, 136).

Wältermann, M. (2013). Dimension-based Quality Modeling of Transmitted Speech. https://api.semanticscholar.org/CorpusID:63687570 (cf. p. 47)

Wang, C., Riviere, M., Lee, A., Wu, A., Talnikar, C., Haziza, D., Williamson, M., Pino, J., & Dupoux, E. (2021). VoxPopuli: A Large-Scale Multilingual Speech Corpus for Representation Learning, Semi-Supervised Learning and Interpretation. *ACL Proceedings* (cf. p. 33).

Wang, Y., Narayanan, A., & Wang, D. (2014). On Training Targets for Supervised Speech Separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *22*(12), 1849–1858. https://doi.org/10.1109/TASLP.2014.2352935 (cf. pp. 14, 15)

Weninger, F., Erdogan, H., Watanabe, S., Vincent, E., Le Roux, J., Hershey, J. R., & Schuller, B. (2015). Speech Enhancement with LSTM Recurrent Neural Networks and its Application to

Noise-Robust ASR. In E. Vincent, A. Yeredor, Z. Koldovský, & P. Tichavský (Eds.), *Latent Variable Analysis and Signal Separation* (pp. 91–99). Springer International Publishing. (Cf. p. 51).

Wichern, G., Antognini, J., Flynn, M., Zhu, L. R., McQuinn, E., Crow, D., Manilow, E., & Roux, J. L. (2019). WHAM!: Extending Speech Separation to Noisy Environments. (Cf. pp. 82, 90).

Xiong, F., Appell, J., & Goetze, S. (2012). System Identification for Listening-Room Compensation by means of Acoustic Echo Cancellation and Acoustic Echo Suppression Filters. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*. https://doi.org/10.1109/ICASSP.2012.6287932 (cf. p. 2)

Xu, Y., Du, J., Dai, L.-R., & Lee, C.-H. (2014). An Experimental Study on Speech Enhancement Based on Deep Neural Networks. *IEEE Signal Processing Letters*, *21*(1), 65–68. https://doi.org/10.1109/LSP.2013.2291240 (cf. p. 14)

Xu, Y., Du, J., Dai, L.-R., & Lee, C.-H. (2015). A Regression Approach to Speech Enhancement Based on Deep Neural Networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *23*(1), 7–19. https://doi.org/10.1109/TASLP.2014.2364452 (cf. p. 16)

Xu, Z., Strake, M., & Fingscheidt, T. (2021). Deep Noise Suppression With Non-Intrusive PESQNet Supervision Enabling the Use of Real Training Data. https://doi.org/10.48550/ARXIV.2103.17088. (Cf. p. 41)

Yang, S.-w., Chi, P.-H., Chuang, Y.-S., Lai, C.-I. J., Lakhotia, K., Lin, Y. Y., Liu, A. T., Shi, J., Chang, X., Lin, G.-T., Huang, T.-H., Tseng, W.-C., Lee, K.-t., Liu, D.-R., Huang, Z., Dong, S., Li, S.-W., Watanabe, S., Mohamed, A., & Lee, H.-y. (2021). SUPERB: Speech Processing Universal PERformance Benchmark. *Interspeech 2021*, 1194–1198. https://doi.org/10.21437/Interspeech.2021-1775 (cf. p. 145)

Yen, H., Germain, F. G., Wichern, G., & Roux, J. L. (2023). Cold Diffusion for Speech Enhancement. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. https://doi.org/10.1109/ICASSP49357.2023.10096064 (cf. pp. 2, 145)

Yi, G., Xiao, W., Xiao, Y., Naderi, B., Möller, S., Wardah, W., Mittag, G., Cutler, R., Zhang, Z., Williamson, D. S., et al. (2022). ConferencingSpeech 2022 Challenge: Non-intrusive Objective Speech Quality Assessment (NISQA) Challenge for Online Conferencing Applications. *arXiv preprint arXiv:2203.16032* (cf. pp. 46, 47, 136).

Zhang, B., Gu, S., Zhang, B., Bao, J., Chen, D., Wen, F., Wang, Y., & Guo, B. (2022). StyleSwin: Transformer-Based GAN for High-Resolution Image Generation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11304–11314 (cf. p. 36).

Zhang, X., Zhang, Q., Liu, H., Xiao, T., Qian, X., Ahmed, B., Ambikairajah, E., Li, H., & Epps, J. (2024). Mamba in Speech: Towards an Alternative to Self-Attention. (Cf. pp. 27, 126, 127).

Zhang, Z., Han, R., Wang, Z., Xia, X., Xiao, Y., & Xie, L. (2023). The NWPU-ByteAudio System for CHiME-7 Task 2 UDASE Challenge. *7th International Workshop on Speech Processing in Everyday Environments (CHiME 2023)*, 19–22. https://doi.org/10.21437/CHiME.2023-4 (cf. p. 85)

Ziaee, A., & Çano, E. (2023). Batch Layer Normalization A new normalization layer for CNNs and RNNs. *Proceedings of the 6th International Conference on Advances in Artificial Intelligence*, 40–49. https://doi.org/10.1145/3571560.3571566 (cf. p. 30)

Zurek, P. M., & Studebaker, G. (1993). Binaural advantages and directional effects in speech intelligibility. *Acoustical factors affecting hearing aid performance*, *2*, 255–275 (cf. p. 118).