Analysis, Disentanglement, and Conversion of Singing Voice Attributes

Brendan O'Connor

PhD Thesis

School of Electronic Engineering and Computer Science
Queen Mary University of London

January, 2024

Statement of Originality

I, Brendan O'Connor, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Signature: ***Brendan O'Connor***
Date: *15/01/2023*

Abstract

Voice conversion is a prominent area of research, which can typically be described as the replacement of acoustic cues that relate to the perceived identity of the voice. Over almost a decade, deep learning has emerged as a transformative solution for this multifaceted task, offering various advancements to address different conditions and challenges in the field. One intriguing avenue for researchers in the field of Music Information Retrieval is singing voice conversion - a task that has only been subjected to neural network analysis and synthesis techniques over the last four years.

The conversion of various singing voice attributes introduces new considerations, including working with limited datasets, adhering to musical context restrictions and considering how expression in singing is manifested in such attributes. Voice conversion with respect to singing techniques, for example, has received little attention even though its impact on the music industry would be considerable and important. This thesis therefore delves into problems related to vocal perception, limited datasets, and attribute disentanglement in the pursuit of optimal performance for the conversion of attributes that are scarcely labelled, which are covered across three research chapters.

The first of these chapters describes the collection of perceptual pairwise dissimilarity ratings for singing techniques from participants. These were subsequently subjected to clustering algorithms and compared against existing ground truth labels. The results confirm the viability of using existing singing technique-labelled datasets for singing technique conversion (STC) using supervised machine learning strategies. A dataset of dissimilarity ratings and timbral maps was generated, illustrating how register and gender conditions affect perception.

In response to these findings, an adapted version of an existing voice conver-

sion system in conjunction with an existing labelled dataset was developed. This served as the first implementation of a model for zero-shot STC, although it exhibited varying levels of success. An alternative method of attribute conversion was therefore considered as a means towards performing satisfactorily realistic STC. By refining 'voice identity' conversion for singing, future research can be conducted where this attribute, along with more deterministic attributes (such as pitch, loudness, and phonetics) can be disentangled from an input signal, exposing information related to unlabelled attributes.

Final experiments in refining the task of voice identity conversion for the singing domain were conducted as a stepping stone towards unlabelled attribute conversion. By performing comparative analyses between different features, singing and speech domains, and alternative loss functions, the most suitable process for singing voice attribute conversion (SVAC) could be established.

In summary, this thesis documents a series of experiments that explore different aspects of the singing voice and conversion techniques in the pursuit of devising a convincing SVAC system.

Acknowledgements

As warned by the head of our PhD programme, this has indeed been the most challenging journey I could ever have imagined taking. In my application to pursue this degree, I recall expressing my excitement to be on the frontier of technological innovation. While it has been a thrill to work in such an environment, the journey came with a level of fatigue that has frequently stolen the wind from my sails - a sensation my optimistic character has rarely let me experience in the past. This fatigue comes in waves over three acts: realising that the work to be done has been grossly underestimated, doing the work, and experiencing self-doubt when the work refuses to be done right. There are, however, roughly two weeks of elation when the work works out and is validated by third parties before this cycle repeats itself. Wading in such uncharted waters can be lonely (especially during the COVID-19 pandemic) and intimidating. In such times I have learned when it is reasonable to rely on my own abilities and when it is smart to lean on other people for various forms of support.

Firstly, I must acknowledge all my eclectic and friendly colleagues in the Media and Arts Technology Centre for Doctoral Training, whose gleeful attitudes were vital in priming me with enthusiasm and resilience for this journey. My humble gratitude goes out to the organisers of this programme for funding my research[1]. Thanks to Nick Bryan-Kinns and Jonathan Winfield, who introduced us to the PhD world, facilitated a warm and welcoming environment within our forums, and have been there to provide encouragement, advice, and guidance throughout our time at Queen Mary.

The crowd at C4DM has been equally supportive. I have seemingly been working with superstars in this field, and I am relieved to know that most of them

[1]This work was supported by the EPSRC and AHRC grant EP/L01632X/1

would join me for an evening beer to unwind or an afternoon coffee to rejuvenate. To them, I offer my thanks for their advice and companionship. I will also take this opportunity to do the same for my non-academic close friends, who have supported me over many a pint.

Thanks to all the academic staff who have hired me for assistance or have taught me about fascinating things, and particularly to Dan Stowell and George Fazekas for their guidance and helpful comments on my work as I progressed through the stages.

I must sincerely thank my supervisor, Professor Simon Dixon. Although he was the first to admit that his initial hands-on familiarity with neural networks was limited, it never ceased to amaze me how he always knew the right questions to ask. He taught me to focus on the important questions without getting sidetracked by technical difficulties. When discussing problems I would have agonised over for weeks, Simon had this remarkable ability to remove the tension from such concerns. His relaxed attitude towards problem solving allowed me to regain many missed hours of sleep. This man is full of wisdom, perspective, patience, and kindness.

I am grateful for my family abroad. They have continuously assured me that I am the right man for the job, displayed great interest in my research, and filled me with love and joy when it was needed. I am grateful for my partner and her three boys, who have given me emotional support, comic relief, and a place to finally feel at home in London. I am grateful for my son, born a year ago this month, who has given me such excitement and everything to look forward to in the years to come. Jaxson, I hope that my efforts here will contribute towards a life for you that is beautiful, fun, stable, and full of love.

Contents

1 Introduction **1**

1.1 Motivation and Aims 1

1.1.1 The Frontier of Singing Voice Conversion 1

1.1.2 Academic and Industrial Interest 2

1.2 Aim 3

1.3 Thesis Structure 4

1.4 Contributions 5

1.5 Associated Publications 6

1.6 Author's Background 7

2 Principles of Machine Learning **9**

2.1 Tasks and Learning Strategies 9

2.1.1 Task Types 10

2.1.2 Supervised Learning 10

2.1.3 Unsupervised Learning 11

2.1.4 Semi-Supervised Learning 11

2.1.5 Self-Supervised Learning 11

2.2 Data Preprocessing 11

2.2.1 Imbalanced Datasets 12

2.2.2 Dataset Splits 13

2.2.3 Dimensionality Reduction 13

2.2.4 Normalisation 14

2.2.5 Augmentation 15

Audio Domain 15

Frequency Domain 16
Latent Space Domain 17
2.3 Objective Function 17
2.3.1 Distance Metrics 18
2.3.2 Classification Metrics 19
Classification Evaluation Metrics 20
2.4 Gradient Descent 22

3 Introduction to Neural Networks **24**
3.1 Modules and Mechanisms 24
3.1.1 The Perceptron 24
3.1.2 Activation Functions 25
3.1.3 Dense/Linear/Fully-Connected Layer 26
3.1.4 Convolutional Layer 27
3.1.5 Normalisation Layer 29
3.1.6 Skip Connections 30
3.1.7 Gated Blocks 31
Gated Linear Units 32
3.1.8 Recurrent Layer 32
3.1.9 Attention Layer 35
Attention Mechanism 35
Simplified Attention Mechanism 36
Self-Attention Layers 36
3.2 Architectures 37
3.2.1 WaveNet 37
3.2.2 Autoencoders 39
Standard Autoencoders 39
Variational Autoencoder 39
3.2.3 Generative Adversarial Networks 40
3.2.4 Teacher-Student System 41
3.2.5 Transformer Model 42
3.2.6 Diffusion Model 42

4 Background **44**
4.1 The Voice . . . 44
4.1.1 Physiology . . . 44
Use of Respiratory Function . . . 45
Vocal Folds . . . 46
Vocal Tract Filtering . . . 48
4.1.2 Voice for Singing . . . 49
Vocal Register . . . 49
Expressivity . . . 50
Taxonomy of Vocal Sounds . . . 51
Voice Identity . . . 52
4.1.3 Speech and Singing . . . 53
Domain-Specific Characteristics . . . 53
Choosing a Dataset . . . 54
4.2 Perception of Sound . . . 56
4.2.1 Listening Studies . . . 56
Multidimensional Scaling Techniques . . . 56
Interpretation of Timbral Space . . . 57
Experiment Design . . . 58
4.2.2 Analysis and Evaluation Methods . . . 59
Statistical Analysis . . . 59
Clustering Techniques . . . 60
Evaluation with Computational Metrics . . . 60
Subjective Evaluation . . . 61
4.3 Spectral Representations of Audio . . . 62
4.3.1 Spectrum . . . 63
Adaption for Human Perception . . . 63
4.3.2 Cepstrum . . . 64
Mel-Generalised Cepstrum . . . 65
4.3.3 Spectral Envelope . . . 66
4.3.4 Mel-Frequency Cepstral Coefficients . . . 66
4.3.5 Vocoder . . . 67
WORLD Vocoder . . . 68

4.4 Neural Networks for Audio and Voice-Related Tasks 72
4.4.1 Alternative Loss Components 72
Embedding Loss 72
Latent Loss as Regularisation 73
Cycle-Consistency Loss 74
Contrastive Loss 74
4.4.2 Audio Analysis 77
4.4.3 Disentanglement 80
Conditioning 80
Vector Quantisation 82
Transfer Learning 82
Auxiliary Classifiers 83
Gaussian Mixture Modelling 83
4.4.4 Voice Conversion and Synthesis Systems 84
Considerations for Voice Conversion Tasks 84
Autoencoder Systems 86
VAE Systems 87
Other Architectures 89
TTS and SVS Systems 90
4.4.5 Audio Synthesis 93
Digital Signal Processing 94
GANs 94
Diffusion Models 95

5 Perceptual Spaces for the Singing Voice **96**
5.1 Introduction 97
5.2 Method 99
5.2.1 Stimuli 99
Experimental Requirements 99
VocalSet 100
5.2.2 Sampling from VocalSet 103
Register-Matching 103
Stimuli Post-Processing 104

5.2.3 Listening Study Setup . . . 105
5.2.4 Participants . . . 105
Participant Questionnaire and Instructions . . . 106
Data Reliability Management . . . 108
Pilot Study Feedback . . . 108
5.2.5 Data Encoding . . . 108
5.2.6 Participant and Data Screening . . . 109
5.2.7 Analysis . . . 111
Clustering . . . 111
Statistical Tests . . . 113
Matrices to 2D-plots . . . 113
5.3 Results and Discussion . . . 114
5.3.1 Data Screening . . . 114
5.3.2 Cluster Scores . . . 115
5.3.3 Experimental Conditions (Controlled Variables) . . . 119
Best *k* Distributions . . . 119
Cluster Score Distributions . . . 120
Pairwise Class Distance Distributions . . . 122
5.3.4 Multidimensional Scaling . . . 123
Testing by Participant Conditions . . . 125
Correlations Among Participant Features . . . 125
5.4 Conclusion . . . 129
5.4.1 Results . . . 129
5.4.2 Reflection . . . 130
5.4.3 Future Work . . . 131

6 Zero-shot Singing Technique Conversion 132
6.1 Introduction . . . 133
6.2 AutoVC System . . . 134
6.2.1 Input Features . . . 134
6.2.2 Details of AutoVC . . . 135
VIE Conditioning . . . 135
Bottleneck Calibration . . . 136

Training Phase . . . 137
Conversion Phase . . . 138
Architectures and Hyper-parameters . . . 138
Waveform Synthesis . . . 140
6.3 AutoSTC System . . . 141
6.3.1 Datasets . . . 141
6.3.2 STE Encoder . . . 142
6.3.3 Training . . . 145
STE Conditioning . . . 145
Hyper-Parameters . . . 145
Sequential Datasets . . . 147
Evaluation metrics . . . 147
6.4 Listening Study . . . 149
6.4.1 Setup . . . 149
6.4.2 Task Description . . . 150
6.4.3 Results . . . 152
Naturalness and Similarity Scores . . . 152
Discussion . . . 153
6.5 Conclusion . . . 157
6.5.1 Future Work . . . 157

7 Singing Voice Identity Embedding and Conversion 159
7.1 Introduction . . . 159
7.1.1 Motivation . . . 159
7.1.2 Chapter Summary . . . 160
7.2 VIE Experiments . . . 161
7.2.1 Input Features . . . 162
DAMP Intonation Dataset . . . 162
WORLD Spectral Envelope Generation . . . 162
Additional WORLD Features . . . 163
Results . . . 164
7.2.2 WORLD versus Mel-Spectrogram Features . . . 166
Results . . . 166

7.2.3 Domain Task Comparisons . . . 167
Dataset Curation . . . 167
Imbalanced Dataset Comparison . . . 168
GE2E Losses . . . 169
Embedding Visualisations . . . 171
7.3 Singing Voice Conversion Task . . . 175
7.3.1 Loss Function Comparison . . . 176
7.3.2 Disparity between Same and Cross Domain Inference . . . 177
7.3.3 Same and Cross-Domain VIE Encoder Comparison . . . 178
7.3.4 Evaluation . . . 179
Listening Study Evaluation . . . 179
Objective Metrics Evaluation . . . 180
Stimuli . . . 180
7.3.5 Results . . . 181
Participant Ratings . . . 181
AutoSVIC Loss . . . 184
Embedding Space Visualisations . . . 185
Cosine Similarity Metrics . . . 185
Singer Identity Disentanglement . . . 187
Discussion . . . 188
7.4 Conclusion . . . 190
7.4.1 VIE Encoder Experiments . . . 190
7.4.2 SVIC Experiments . . . 192
Results for Question 1 . . . 192
Results for Question 2 . . . 193
7.4.3 Future Research . . . 194

8 Conclusion **196**
8.1 Summary of Contributions . . . 196
8.1.1 Perception of Singing Techniques . . . 196
8.1.2 Conversion Models . . . 197
8.1.3 Datasets of Differing Voice Representations . . . 198
8.1.4 Data Representation . . . 198

8.1.5 Conversion Evaluation 199
8.2 Future Work 200
8.2.1 Listening Study 200
8.2.2 Improving Models 200
STE Encoder 201
VIE Encoder 201
Voice Conversion Network 202
Audio Synthesis 202
Input Data 202
8.3 Final Remarks 203

A Listening Study Details from Chapter 5 **205**
A.1 GOLD-MSI Perceptual Ability Questions 205
A.2 Participant Feature Distributions 206

B Listening Study Details from Chapter 6 **208**
B.1 STC Listening Study Questions 208

List of Figures

2.1 A confusion matrix, labelling all types of predictions for a binary classification task 21

2.2 ROC graph demonstrating how the decision boundaries can be adjusted to produce an arc in the sensitivity-specificity space. A larger area under the curve between the ROC and the diagonal means higher performance. 22

3.1 Diagram of the perceptron, where b represents the bias, w_n represents the nth weight, and $f(x)$ represents the activation function. . 25

3.2 *Sigmoid*, *ReLU*, *Tanh* and *Step* activation functions 26

3.3 Convolutional process on a 2D feature map, using a kernel of size 3, stride of 2, and padding of 1. The green tiles represent the output feature map while the blue ones represent the input feature map. 28

3.4 A diagram of the convolutional process. 29

3.5 Flow chart depicting the use of skip connections as detailed by He et al. [2015]. x goes through multiple transformation processes considered as $f(x)$, the output of which is reunited with x by addition. 30

3.6 Depiction of a gated block, where a $sigmoid$ signal is gating a $tanh$ signal (or vica versa) 31

3.7 A diagram of the RNN process unfolded across time, where: U, W, and V are the weights for the input, hidden and output states respectively; and x, s, o and t represent the input, hidden, output and timestamp, respectively. 33

3.8 Diagram of the inner architecture of the (a) LSTM and (b) GRU units. 34
3.9 Flow diagram depicting a stack of dilated, causal convolution layers. The flow of information between layers is shown with arrows. 38
3.10 Flow diagram depicting the WaveNet's architecture. 38

4.1 Anterior view (left) and plan view (right) of the larynx. 45
4.2 Lateral view of the vocal tract. 46
4.3 A 2-dimensional representation of the placement of the phonation modes, where the vertical axis and horizontal axis correspond to glottal airflow and subglottal pressure, respectively. 48
4.4 Flowchart illustrating WORLD's analysis algorithms extract F0, spectral envelope and aperiodic information. 69
4.5 Flowcharts illustrating how vocal fold vibrations are determined in the STRAIGHT and WORLD vocoder systems. The $*$ symbol represents convolution. 71
4.6 A Similarity matrix, arranged so that the first axis represents utterance embeddings j, the second axis represents the vocalist centroids k, and the cells represent the similarity scores between the corresponding pair of axes elements. In this illustration, the coloured cells represent where the identity of the utterances matches that of the centroid. 76
4.7 An illustration of the effects of the GE2E softmax loss. Circles and triangles represent embeddings and centroids, respectively. Dotted arrows represent the repelling force between unmatched embedding-centroid pairs, while the solid-line arrow represents the attracting force between matched elements. 77

5.1 Example of a dissimilarity matrix generated from participants' pairwise ratings. Shortened versions of the labels *straight, belt, breathy, fry* and *vibrato* are featured along the axes. Values closer to 1 indicate very strong dissimilarity, while values closer to 0 indicate strong similarity. In this case, data relating to one of three *straight* singing audio clips is missing. 101

5.2 Hierarchical diagram of sampled stimuli. Colours represent different levels of conditional groups, while ellipses signify that the contents of its condition block imitate the contents of the block representing the same hierarchical condition group on the far left. 102
5.3 View of interface used by participants for rating dissimilarities between a single reference recording and multiple comparative recordings. 106
5.4 Correlation matrix between dissimilarity matrices of each participant, indicating significantly more uncorrelated matrices than correlated ones. 115
5.5 Correlation matrices of data between individual listening sessions of male singers 116
5.6 Correlation matrices of data between individual listening sessions of female singers 117
5.7 Distributions across all data for best accuracy (left column) and silhouette (right column) scores across all values of k, using agglomerative (top row) and k-means (bottom row) clustering. . . . 118
5.8 Scatter plots displaying correlations between metrics using k-means and agglomerative clustering algorithms. 119
5.9 Distributions for best accuracy (left column) and silhouette (right column) scores across all values of k, for each register condition. . 120
5.10 Distributions for best accuracy (left column) and silhouette (right column) scores across all values of k, for each gender condition. . 121
5.11 Plots displaying the perceptual space of singing techniques after dimensionality reduction, grouped by low, high, male and female singing conditions. Axes are not labelled as MDS does not produce coordinates based on predefined measurable concepts, due to its inherent objective of summarising data based on shared correlations, variances and relative distances. Further post-hoc objective and subjective evaluations however, could be used to determine the meaning of each dimension. 124

5.12 Box plots illustrating distributions of scores for each category across the y-axis, where there was measured correlation with the instrumentation categories of an ordinal nature shown on the x-axis. 127
5.13 Scatter plots illustrating a mild correlation between the measurements of the attributes labelled on the x and y-axes. 128

6.1 Flowchart illustrating the information flow of the AutoVC system. The secondary cycle illustrates how AutoVC's output is reinserted as its input for a secondary pass of the system to obtain new embeddings generated from the synthesised output. 135
6.2 Flowcharts illustrating three scenarios where: (a) the bottleneck is too small to encode linguistic content; (b) the bottleneck is too large, allowing it to also encode voice identity content; (c) the bottleneck is just the right size to allow E_{VC} to disentangle *only* linguistic content from the spectrogram. 136
6.3 Flowchart illustrating the flow of information in AutoVC, with interchangeable pathways (indicated by dashed modules and edges) to be used for either the training or conversion phase. 139
6.4 Flowchart providing an in-depth illustration of the architecture of the VC network portion of AutoVC. The numbers seen above individual layers display the dimension size of their output. Numbers below the resampling layers indicate the factor by which they are up/downsampled. 139
6.5 Flowchart illustrating the architecture of the STE encoder. The numbers seen above individual layers indicate the dimension size of their output, while the numbers below indicate the max-pooling kernel size. This diagram shows reshaping as if the batch size is 1, and the number of chunks is 6 (0.5s each) 144
6.6 Sum-of-squared distances for STEs (y-axis), plotted across a range of k values (x-axis) 146
6.7 Interfaces used for (a) the naturalness task, and (b) the similarity task 150

6.8 Bar graphs displaying listening test scores. Condition groups are colour-coded together from left to right as: models, subsets, genders, source techniques and target techniques. **Top:** Naturalness (MOS values and confidence intervals) for all conditions. **Bottom:** Similarity scores as determined by Equation 6.7. 153

7.1 GE2E loss contours for VIE encoders using different variations of WORLD feature. Legend indicates which parameter of the WORLD generation process was changed. The contour for 'frame duration' only covers training steps from 12500 to 27500 due to lost data, but still conveys the drop in loss caused by this adaptation. 165

7.2 GE2E loss contours for VIE encoders over 160k training steps, trained on the DI (blue) and VCTK (red) when trained on mel (solid lines) and World (dashed lines) features. 167

7.3 GE2E contours displaying VIE encoders separately trained on LS_1.2k (red) and DI_1.2k (blue) datasets. The pink contour is based on an encoder that was pretrained on LS_1.2k and continued training on the DI_1.2k. 170

7.4 t-SNE generated maps of VIEs representing (a) speech and (b) singing clips from the NUS dataset, generated from encoders trained on the **LS_1.2k (speech)** data. 172

7.5 t-SNE generated maps of VIEs representing (a) speech and (b) singing clips from the NUS dataset, generated from encoders trained on the **DI_1.2k (singing)** data. 173

7.6 t-SNE generated maps of VIEs representing (a) speech and (b) singing clips from the NUS dataset, generated from encoders trained on **LibriSpeech and VoxCeleb1** data. 174

7.7 Diagram of the VIE encoder and the SVIC network components in the AutoSVIC network, with a secondary cycle partition illustrating how encodings for the reconstructed data are obtained (as explained in Figure 6.1). Vector comparisons used for reconstruction loss (L_{rec}), bottleneck regressor loss (L_{BN}) and singer identity embedding regressor loss (L_{VIE}) are shown with dotted connectors. X represents a mel-spectrogram and BN represents the residual data (singer identity-independent information) encoded in the bottleneck. 176
7.8 Bar graphs displaying results for naturalness, similarity and cosine similarity. 182
7.9 Total loss contours for each of the four trained AutoSVIC models. 185
7.10 t-SNE generated maps of DI-originating VIEs using the (a) DI-pretrained and (b) LSVC-pretrained encoders. 186
7.11 Classification accuracy of classification layers being appended to the encoders of the DI, DI-BN and DI-VIE models, indicating the amount of singer identity information still entangled in the models' bottlenecks. 189

A.1 Bar graphs (subplots (b) and (c)) and distributions (subplots (a) and (d)) illustrating the spread of participant features. Subplot (b) presents non-musicians and musicians in their abbreviated form 'Non-Mus' and 'Mus' respectively. Subplot (c) omits the third option 'significant', as these participants would have been filtered out during the screening processes. 206
A.2 Subplots *(a)* to *(e)* present bar graphs and distributions for participant scores generated from their dissimilarity ratings. Subplot *(e)* presents the distribution of ratings across all participants. 207

List of Tables

5.1 Mann Whitney U results for significant differences between register conditions 122
5.2 Mann Whitney U results for significant differences between gender conditions 122
5.3 Statistically significant differences between the different participant features including accuracy and silhouette scores. The subscript 'S' or 'P' indicates whether the correlation was of type Spearman or Pearson. 126

6.1 Table presenting losses (and training step count in parentheses) when using the VocalSet dataset for evaluation. All sequences of dataset combinations are shown, with the first, second and third dataset in the sequence being reported in the left, middle and right sections of the table. The optimum training path is highlighted in bold. Training on a dataset that leads to an increase in loss is indicated with a circumflex, at which point that path is abandoned. For space, the dataset names have been shortened as follows: VCTK:Vc, VocalSet:Vs, MedleyDB:Md. 148
6.2 Table presenting losses (and training step count in parentheses) when evaluating on the MedleyDB dataset. 148
6.3 Mann Whitney U results for significant differences between sample groups relating to naturalness. 154
6.4 Mann Whitney U results for significant differences between sample groups relating to similarity. 155

7.1 WORLD parameter configuration comparison. Entries for Pitch and Aperiodicity columns indicate whether these were included in the input features, while text in bold highlights the change being tested from the baseline (top row) configuration. 164
7.2 Presents GE2E losses for multiple VIE encoders, pretrained on either the LS1.2k or DI_1.2k and evaluated on NUS speech or singing subsets. The 'Loss Difference' column displays the increase in GE2E loss incurred when the encoder goes from same to cross-domain inference. 170
7.3 Mann Whitney U results of significant differences between samples of different conditions relating naturalness, similarity and cosine similarity. 183
7.4 Classification accuracy results for models using different loss functions. 188

List of abbreviations

AE	Autoencoder
AI	Artificial Intelligence
ASR	Automatic Speech Recognition
BCE	Binary Cross-Entropy
BBTT	Backpropogation Through Time
BLSTM	Bidirectional LSTM
CCE	Categorical Cross-Entropy
C4DM	Centre for Digital Music
CNN	Convolutional Neural Network
CMMR	Computer Music Multidisciplinary Research
DAMP	Digital Archive of Musical Performances (dataset)
DCT	Discrete Cosine Transform
DDSP	Differentiable DSP
DSP	Digital Signal Processing
DI	DAMP Intonation (subset of DAMP dataset)
DFT	Discrete Fourier Transform
GAN	Generative Adversarial Network
GE2E	Generalised End-to-End
GMM	Gaussian Mixture Model
GPU	Graphics Processing Unit
GLU	Gated Linear Unit
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model
IFT	Inverse Fourier Transform
KLD	Kullback Leibler Divergence
KW	Kruskal Wallis
LPC	Linear Predictive Coding
LS	LibriSpeech (dataset)

LSVC	LibriSpeech + VoxCeleb (dataset combination)
LSTM	Long Short-Term Memory
MCC	Mel-Cepstral Coefficient
MDS	Multidimensional Scaling
MLP	Multilayer Perceptron
MFCC	Mel Frequency Cepstral Coefficients
MFSC	Mel Frequency Spectral Coefficients
MGC	Mel-Generalised Cepstrum
MIDI	Musical Instrument Digital Interface
MIR	Music Information Retrieval
ML	Machine Learning
MLP	Multilayer Perceptron
MSD	Multiscale Discriminator
MOS	Mean Opinion Score
NN	Neural Network
PCA	Principal Component Analysis
PCD	Pairwise Class Distance
PWG	Parallel WaveGAN
QMUL	Queen Mary University of London
RNN	Recurrent Neural Network
SGP	Sub-Glottal Pressure
SVIC	Singing Voice Identity Conversion
SVAC	Singing Voice Attribute Conversion
SVM	Support Vector Machine
SW	Shapiro Wilks
SVS	Singing Voice Synthesis
STC	Singing Technique Conversion
STE	Singing Technique Embedding
STFT	Short-Time Fourier Transform
TTS	Text-to-Speech
VAE	Variational Autoencoder
VCTK	Voice Cloning Toolkit (dataset)
VIC	Voice Identity Conversion
VIE	Voice Identity Embedding
VUV	Voiced - Unvoiced
VQ	Vector Quantisation
WAET	Web Audio Evaluation Toolkit
w.r.t.	“with respect to”

Chapter 1

Introduction

The topic of this thesis is singing voice attribute analysis, disentanglement, and conversion. This chapter describes the motivation, aim, structure, and contributions of this thesis. It also presents several publications related to its main content chapters.

1.1 Motivation and Aims

1.1.1 The Frontier of Singing Voice Conversion

Although the advent of neural networks (NNs) was already gaining momentum in its new phase of the 'AI hype cycle' at the time this PhD began, literature on such models being applied to music was, of course, significantly more sparse than it is now. A few years earlier, Google's *AlphaGo* [Silver et al., 2016] demonstrated how it could develop its own strategies in the game of *Go* to outperform the Go master, Lee Sedol. Later that year, Google also produced *WaveNet* [van den Oord et al., 2016b], a model that was able to generate waveform audio with astonishing clarity. These exciting advances in computer science made the prospects of their application to the singing voice all the more exciting. There was already an accumulated excitement in using deep learning for singing voice separation [Huang et al., 2014, Jansson et al., 2017] and analysis [Schlüter and Grill, 2015, Leglaive et al., 2015]. However, there was very little research on the conversion of singing voices using NNs.

In contrast, there was (and still is) a large amount of literature on the application of NNs to the task of *spoken* voice conversion. As will be discussed in Section 4.1.3, through no surprise, the demand for spoken-voice-related technological advancements has always overshadowed its counterpart in the singing domain. Similarly, singing datasets are few in number and tiny in size (not to mention how few *annotated* datasets there are), as highlighted by [Stoller, 2020]. The fact that there was a sparse amount of research and resources available specifically concerning singing voice conversion made it clear that this would be an exciting niche at the intersection of Music Information Retrieval (MIR) and probabilistic machine learning (ML). The large amount of literature on spoken voice conversion has proven to provide significant insight into the topic of singing voice conversion.

1.1.2 Academic and Industrial Interest

Today, interest groups such as the 'Music Technology Group' in Barcelona, or the international 'Singing Voice Interest Group' have been vocal about their interest in NNs and the singing voice. A clear increase of relevant paper submissions to conferences such as ISMIR, ICASSP, and INTERSPEECH demonstrates a continuously rising worldwide interest in the subject.

As domains like speech and computer vision produce more convincing and novel probabilistic ML-based transformations, the inevitable interest of the entertainment industry has propelled further research in more artistically relevant domains such as singing. In recent years, there has been an explosion of applications that specialise in voice conversion, confirming that the entertainment industry is now a core player in this frontier of voice synthesis and conversion research.

Singing voice *attribute* conversion (SVAC) describes the process of replacing acoustic features that relate to a specific attribute of the voice such as identity, timbre, or singing techniques. It is apparent that technology of this variety has yet to fully find its place in the music production suites of heavy-hitting recording studios. The music industry has already benefited from singing voice manipulation such as pitch correction, vocoding, and formant manipulation[1]. The ability to switch a singer's particular attribute with that of any number of alternative singers,

[1]Examples of this include *Melodyne*, *iZotope RX*, and *Vocaloid*.

without the target singer needing to be present in the studio, would be a revolutionary change for composers, performers, producers, and music labels, saving time and money (although how this will be managed ethically and legally remains to be seen).

Such technology will have indisputable artistic value. Popular music has, over recent years, increasingly digitalised the character of the voice by all means available to it, even extending its use as a musical function to cater towards sound design as well as a vehicle for narration. In addition to the identified gap in the literature and interest among the research community and the industry, the prospect of putting SVAC technology in the hands of the artist is the final key motivation behind the research that makes up this thesis.

1.2 Aim

This work aims to contribute to the topic of SVAC with NNs by analysing how humans perceive singing, how existing voice identity conversion (VIC) systems perform when applied to the task of singing technique conversion (STC), how much transferable knowledge the VIC systems can learn between the speech and singing domains, and how singing VIC can improve.

An end-user's ability to interpolate between source and target singer attributes in a latent space is not so fruitful without an acceptable degree of control. As the space between such attributes becomes linear in its relation to human perception, control becomes more intuitive. As computers cannot be primed with human perception, this is something that must be provided. Unfortunately, little research is available on the perception of singing. The research presented in Chapter 5 attempts to fill this gap with respect to (w.r.t.) the perception of the singing technique. In theory, these findings, when applied to a model, should also encourage it to model singing techniques in a manner that is more similar to human perception.

To achieve any kind of attribute conversion, the attributes of interest must be disentangled from the data. The term disentanglement, as conceptualised in this thesis, is the process of separating a particular subset of information from a representation of data. This subset would exclusively relate to a particular attribute of the data, such as the melody of a singer. Replacing this subset with that of

another example of data (the target data), if similarly distributed, would result in a representation of the data containing the converted attribute. In theory, some of these conversions could result in unrealistic data. This, however, does not imply poor disentanglement, but poor execution where correlated relationships of certain attributes have not been properly considered. Chapter 6 of this thesis aims to explore the techniques available for disentanglement and to provide a solution specific to STC that can be used in a flexible manner.

The research presented in Chapter 7 aims to provide some insight into how much transferable knowledge there is between speech and singing in the context of VIC, and also attempts to fine-tune the identity conversion process for singing. The concept of voice identity can be understood as the acoustic cues within a voice signal that, when heard by a listener, serve as necessary predictors contributing to the perceived identity of that voice. In this thesis, identity is sometimes presented as its own attribute, although the predictors it consists of can also be though of as sub-attributes, such as singing technique or timbre.

1.3 Thesis Structure

Chapter 1 (this chapter) presents the motivation behind the thesis, what it aims to do, the structure it follows and the contributions it provides.

Chapter 2 and **Chapter 3** are primers on ML and NNs, respectively. This provides context and documentation of the journey of knowledge undertaken by the author, offers resources that aid comprehension or provide novel points of view, and highlight the main components of ML and NNs that require essential comprehension before any significant, meaningful contributions in this field of research can be made. Readers who are already confidently familiar with these concepts are welcome to jump ahead to the following chapter if they so wish.

Chapter 4 presents a review of the literature, covering previous research relevant to the experiments conducted in the subsequent content chapters. This includes research related to: the voice, its application to singing, and representation in datasets; the perception of sound, listening experiment design, and subjective rating analysis; and NN-based solutions to voice-related problems such as analysis, disentanglement, conversion, and synthesis.

Chapter 5 presents research that involved curating a collection of recordings that consisted of multiple singers using multiple singing techniques. These stimuli were presented in listening studies in which participants were required to rate their dissimilarity. The results were examined using clustering techniques and statistical analysis, and were presented as timbral maps on 2D planes for interpretation.

Chapter 6 presents research on STC. A singing technique classifier is first proposed, the final embedding layer of which is used to provide singing technique embeddings as conditioning elements to an autoencoder. This system results in an STC network that produces a set of diverse singer-converted recordings. They are evaluated for naturalness and similarity by participants in a listening test.

Chapter 7 presents research relating to VIC, focussing on comparisons between: speech and singing domains; mel-spectrograms and WORLD spectral envelopes as input features; and several modifications of an VIC network's objective function.

Chapter 8 concludes the research discussed in the previous three chapters. It summarises the findings and considers its contributions to the world in its current state. It also discusses some shortcomings and hypothesises how the research can be built upon in future work.

1.4 Contributions

The main contributions this thesis makes to researchers and society include the following:

- Chapter 2 and 3: A tutorial-style introduction to the field of ML and NNs.
- Chapter 5: A description of the perceptual differences of the singing voice when produced by different genders or vocal registers.
- Chapter 6: A report on the results of the first published (to the best of my knowledge) zero-shot singing technique classification and conversion experiment using NNs.
- Chapter 7: An analysis on cross-domain applications of singing and speech data.

- Chapter 7: A report on the differences between commonly used spectral features used for voice identity embedding generation.
- Chapter 7: A proposed objective function for SVAC networks that is robust against poor disentanglement, and includes a metric that explicitly measures the similarity between converted and target voice identities.

1.5 Associated Publications

This thesis presents the research and work conducted by the author between September 2019 and December 2023 at Queen Mary University of London (QMUL) during their PhD, excluding interruptions from July to September 2021, and February to April 2023. The following research was submitted for international peer-reviewed publications as follows:

1. Work described in Chapter 5 is a modified version (analysis only) of a publication at the 2020 Joint Conference on AI Music Creativity [O'Connor et al., 2020]
2. Work described in Chapter 6 was published at the International Symposium on Computer Music Multidisciplinary Research [O'Connor et al., 2021]
3. Work described in Section 7.3 in Chapter 7 was published at the 2023 Sound and Music Computing Conference [O'Connor and Dixon, 2023] (It is the author's intention to build upon the work in Section 7.2 before submitting for publication).

All original research described in this thesis was conducted by the author, which includes all implementations in Python code, presented in Github repositories as referenced (unless explicitly disclaimed otherwise). The writing of this thesis was guided by the primary supervisor, Professor Simon Dixon, who also co-authored the three publications mentioned above. As the secondary supervisor, Dr George Fazekas also contributed towards the development of research and is a co-author of the first two publications mentioned above. Dr Dan Stowell and Dr Huy Phan provided helpful feedback on the development of this thesis as independent assessors.

1.6 Author's Background

The interest in singing originated from my fascination with the voice, not just as a musical instrument, but as a general sound maker. A Bachelor's Degree in Classical Music sparked my curiosity for contemporary sound design and composition, which led me to pursue a Master's Degree in Electronic Music Composition at the University of West London in 2015. There, I took on a large-scale compositional project that explored how the voice could be stripped of its narrative functionality, and have different combinations of its attributes harvested for experimental compositional use, forcing new conceptions regarding the voice's role in music[2]. The author later received a studentship to join the Media and Arts Technology Centre for Doctoral Training (MAT CDT) in 2018, which involved participating in a year of *slightly* relevant modules before a proposal for the topic of this thesis was first drafted.

As required by the MAT CDT, the author elected additional modules at QMUL during the course of their research. Among these were introductory modules to computer programming, ML, and research methods, which attempted to compensate for the author's unfamiliarity with STEM material. It is important to the author that a word of acknowledgement is also given to non-academic resources, as reading research papers and text books alone would have made the transition from musician to computer scientist particularly painful. These included: blog posts, which explain concepts using an inviting and conversational style of writing that is easier to digest, omitting details that are not essential to conveying an idea, providing illustrations and demonstrations with code[3]; and online video tutorials [Velardo, 2020, Kumar et al., 2019, Khan Academy, 2006, Luis Serrano, 2013, Ng and StanfordOnline, 2009, Starmer, 2011] which provided similar styles of presentation and infographics. Sometimes, it is just helpful to have the same idea presented from a different perspective. In the final stages of writing this thesis, ChatGPT [OpenAI, 2023] has been a helpful resource, providing quick

[2]A composition from this project featured in *The 6th Irish Sound Science and Technology Convocation*, and can currently be heard at `https://soundcloud.com/radiofreeissta`

[3]Examples of such sources include, but are not limited to `https://machinelearningmastery.com`, `https://stackoverflow.com`, `https://uk.mathworks.com`, `https://towardsdatascience.com`

reminders of how concepts work, and equally importantly, how to communicate them effectively.

Chapter 2

Principles of Machine Learning

In this section, an overview of the techniques and terminologies relevant to ML is provided. ML refers to a branch of artificial intelligence dealing with models that learn from incoming data by adjusting their own parameters to be able to identify and mimic patterns in the data. In doing so, the model is trained to a point where it can automatically perform simple to complex tasks with minimal human intervention. ML models come in multiple forms. All models require data from which they can learn, a set of learnable parameters, and a process that steers these parameters towards an optimal configuration w.r.t. the main goal.

Key sources covering ML (and NNs, covered in the proceeding chapter) that have contributed significantly to the author's understanding of the relevant concepts include the literature by Chollet [2018], LeCun et al. [1998], Neuneier and Zimmermann [1998], Howard and Gugger [2020], Bell [2020], Goodfellow [2016], Russell and Norvig [2021].

2.1 Tasks and Learning Strategies

To choose the most appropriate model, one must consider the type of problem that needs to be solved, along with some prior knowledge of the distribution of the data being modelled. The type of problem dictates what type of approach is required, which in this field of research, can be broken down into any of the following types of learning strategies: *supervised, unsupervised, semi-supervised*,

or *self-supervised*. If a model is trained successfully using one or more of the aforementioned learning techniques, it should infer meaningful information from unseen data. This inference can be used for tasks such as detection, identification, verification, classification, or generation. In this section, we will describe the types of tasks available, along with appropriate learning strategies.

2.1.1 Task Types

The first type of task involves synthesising novel data. In most cases, it is of interest to synthesise data that is similar to those of a particular subset of observations from a dataset or predict the next element in a sequence. This requires the use of *generative* models such as Boltzmann Machines, Hidden Markov Models (HMMs) and a variety of NN architectures which will be discussed in Section 3.2.

Discriminative or *classification* models, on the other hand, are designed to discriminate between the various classes or clusters of data seen in the dataset. Some models designed for classification tasks include support vector machines (SVMs), decision trees, logistic regression, and K-Nearest-Neighbour (kNN).

2.1.2 Supervised Learning

Supervised learning refers to the process of learning from a set of *labelled* data. Labels represent the ground truth, usually provided by human annotation, and act as targets from which a model can determine how correct or incorrect its predictions are. For example, a dataset may come packaged with a number of entries in the form of audio waveform formats representing mixed audio tracks. These entries may be accompanied with their corresponding labels that explicitly state which class (such as a music genre, artist, instrumentation list, sound event etc.) belongs to which audio track. By providing an ML model with such a dataset, it can be set with the task of learning what structures in the waveform data best predict the ground truth labels.

2.1.3 Unsupervised Learning

Unsupervised learning refers to the process of learning structures from data *without* using labels. In this context, the notion of correct or wrong answers is not so relevant. The model is only required to ascertain how similar entries in a dataset are to each other. The end result of unsupervised learning often comes in the form of dimensionality reduction and/or clustering.

2.1.4 Semi-Supervised Learning

Semi-supervised learning describes a hybrid learning style that combines supervised and unsupervised learning strategies. As will be seen in Section 4.4, semi-supervised strategies have sometimes been reported to outperform supervised strategies. It should therefore be noted that semi-supervised learning is not always a compromise based on the scarcity of labelled data.

2.1.5 Self-Supervised Learning

Self-supervised learning describes the process of attributing labels to otherwise labelless data using a single model, which is achieved by analysing features among data points for similarities and dissimilarities. *Contrastive learning* is a common method of self-supervised learning, where input data is presented to a model in batches that contain more than one example from multiple classes. From such a batch, multiple combinations of positively-matched and negatively-matched instances can be observed. Knowledge of whether a combination is positive or negative acts as a feedback signal, which encourages the model to force its embeddings for these instances to be similar or dissimilar. Positive instances can either be the result of segments taken from different examples of the same class or from multiple segments of the same data point.

2.2 Data Preprocessing

The first step in any ML pipeline is preprocessing the data. Depending on how the data was collected and where it came from, a number of transformations may

be necessary before it is suitable to pass to a model. This includes several stages of data analysis and transformation.

2.2.1 Imbalanced Datasets

When classes in a dataset are not uniformly distributed, the dataset is said to be imbalanced. The first and simplest way to deal with imbalanced datasets is to apply different weights to the loss criterion for each class.

Another solution to an imbalanced dataset is resampling. *Oversampling* is the process of replicating instances within the minority class to balance the ratio between classes. *Undersampling* is the process of removing instances from the majority class, which of course should not be done without considering the size and distribution of the dataset. In some cases, this basic solution has been found to lead to a decrease in training time without a significant decrease in performance [Batista et al., 2004].

Balancing a dataset can be also achieved by deleting data points that are overly similar or too close to one or more neighbours in latent space encodings [Radford et al., 2016]. Another method is an extension of basic oversampling, called the *Synthetic Minority Oversampling Technique* (SMOTE), which generates new instances based on the interpolation between existing instance vectors using a randomised weight [Gazzah and Amara, 2008, Fan et al., 2021], and has been found to contribute to significantly better results. Kovács [2019] presents an evaluation of further adaptations of SMOTE, of which there are many. One such improvement to this algorithm is fuzzy clustering (FC-SMOTE), which considers that a dataset's instances may not be evenly distributed, exhibiting heavy clustering in latent space. This algorithm only generates samples in regions where the minority class instances are clustered together, thereby ensuring that newly generated samples exist in an appropriate latent space region. FC-SMOTE has been shown to improve voice-related classification tasks using MFCCs as input [Fan et al., 2021].

2.2.2 Dataset Splits

Datasets are typically split into subsets for training, validation, and testing. The partitions in which datasets are split are often unofficial, ambiguous, or simply not stated. In these cases, it is important to verify whether an official training set is suggested by the dataset's authors or whether a researcher used a particular split. Won et al. [2021] discuss at length the considerations that should be taken when deciding how a dataset should be split.

In ML research, it is common to find split ratios of 70:30 to 80:20 with respect to training and validation partitions. The exact choice of split is often uninformed and simply inherited from previous research practice. The origin of a split usually considers a compromise between the demand of its users and the amount of data available. The trade-off between how well a model can be trained and how thoroughly it is evaluated is directly proportional to the training-validation ratio.

The *k-fold cross validation* method bypasses the need for predefined partitions. Instead, after determining a ratio split, multiple versions of the model are trained so that the entirety of the dataset is used as training and validation data across all models. If the ratio split was 80:20, then five models would be trained, and each would use a different fifth of the dataset as a validation set. An average validation loss across all models will then represent the model's ability to generalise to unseen data.

Even though the model is not being trained on the validation set, researchers learn from validation losses, which informs their choice in fine-tuning the model. Therefore, an indirect feedback loop exists between the validation set and the model. For this reason, a *test* set is created, which is a subset of a dataset that is kept entirely separate from the training-validation cycle and is only used to prove the final model's generalisation capabilities before deployment.

2.2.3 Dimensionality Reduction

In many cases, the use of raw data for ML purposes will be inefficient. The first consideration is the *curse of dimensionality*, which, as the name implies, warns practitioners that more dimensions do not necessarily imply better performance. In fact, in some cases it can be detrimental, confusing models by allowing them

to detect patterns in the data that are coincidental rather than meaningful. Dimensionality reduction is key to this issue and can be a manual, automatic, or hybrid process between computational analysis and manual feature selection.

Principal component analysis (PCA) is an unsupervised automatic process of dimensionality reduction. It transforms the features of the dataset into a smaller set of uncorrelated features called principle components, which represent the maximal variances of the data [Sammon, 1969]. Of course, reducing a dataset's features from 100 to 2 would omit a considerable amount of information from the data. PCA algorithms usually inform users of how much information is being lost by a dimensionality-limited solution. It is a non-iterative process and therefore reduces computational requirements. It can be used as a de-noising technique and reduces overfitting. However, it is limited to linear transformations and therefore is usually unsuitable for representing data with non-linear relationships [Anowar et al., 2021].

PCA can also be used as a precursor to the task of t-distributed Stochastic Neighbour Embedding (t-SNE), which is another form of dimensionality reduction, more useful for visualisation purposes, where data points have a tendency to group in clearly segregated clusters, allowing users to visually observe how well-defined different classes are to each other [van der Maaten and Hinton, 2008].

Multidimensional scaling (MDS) is an unsupervised process. It is frequently used as a means of data visualisation, and therefore usually requires reductions to 2 or 3 dimensions. MDS is focused on pairwise dissimilarities as opposed to the dataset distribution. It is computationally expensive due to the fact that it relies on generating a dissimilarity matrix between all data points in a dataset [Anowar et al., 2021].

2.2.4 Normalisation

A typical and indispensable trick of the trade is normalisation, which describes the process of rescaling the data so that it is transformed to a reasonable range and mean value. Including this step allows a model to converge more quickly. *Min-max* normalisation is the process of limiting the range of data to values between 0 and 1, and is formulated as

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{2.1}$$

where X' and X represent the normalised data and original data, while $X_{\max}$ and $X_{\min}$ represent the original data's maximum and minimal value, respectively.

However, this is sensitive to outliers and does not assume a Gaussian distribution (an important assumption often used in ML). *Standardisation* (or z-score normalisation) is a normalisation technique that transforms the data so that the distributions of each feature in the dataset possess 0 mean and unit variance. The transformation assumes prior Gaussian distributions among the features and is less sensitive to outliers. It is a widely accepted normalisation process that is known to improve rates of convergence [LeCun et al., 1998, Howard and Gugger, 2020, Neuneier and Zimmermann, 1998, Schlüter and Grill, 2015, Leglaive et al., 2015, Bengio et al., 2006]. It is parameterised as:

$$X' = \frac{X - \mu}{\sigma}. \tag{2.2}$$

where μ and σ represent the mean and standard deviation, respectively.

2.2.5 Augmentation

Augmentation is often necessary to increase the robustness of a model. It makes the model less likely to overfit, more generalisable to real-world conditions, and invariant to naturally observed transformations of the data.

'Unsupervised augmentation' is the application of augmentation without considering the label attached to the data, while 'supervised augmentation' considers the label. Lemley et al. [2017] have used both techniques to increase the accuracy of their systems by a significant amount. Cubuk et al. [2019] have shown that the use of unsupervised augmentation can even reduce errors by more than 30%. With audio data, augmentation can be applied in the time, frequency, or latent domains.

Audio Domain

Augmentations to the audio waveform itself are often preferable, as their application to a representation of reduced dimensionality is not always guaranteed to

reflect realistic or natural transformations of the signal.

The reversal of an audio signal is a perception-invariant augmentation w.r.t. timbre. While the signal itself will sound unnatural, most aspects of timbre perception such as spectral centroid will remain constant, as these are not time-varying. Polarity inversion (shifting the phase by 180°) is another example of augmentation that uniformly changes a fundamental feature of the information without affecting human perception [Nachmani and Wolf, 2019]. Bonada and Blaauw [2021] applied a modest amount of pitch shifting to their input data while temporally scaling it to the relevant spectrogram, facilitating pitch transformation while linearly scaling the timbre in frequency. Injecting noise of different types into waveforms has also been a common transformation [Qian et al., 2019, Bonada and Blaauw, 2021, Kusner and Hernández-Lobato, 2016]. Other possible augmentations that impose realistic augmentations to audio include filtering, delay, and reverb. There are a large number of music production suites, audio editing tools, software packages, and framework-specific libraries that are capable of such transformations.

When dealing with datasets that contain multiple stems, there are a few creative augmentations available. Researchers have often made use of pseudo-randomly combined stem tracks from different songs to create new examples of mixed audio signals or swapping left and right channels [Uhlich et al., 2017, Davies et al., 2014, Lee and Nam, 2019]. This often requires some analysis of key signature, tempo, and other musical attributes to determine whether the resulting mixed track would be a realistic representation. Another example of augmentation in stem-to-mixed approaches would be to modify the gain of each signal to change the 'mix' of the recording [Uhlich et al., 2017], or apply various types of DSP to enrich the diversity of perceived recording conditions for each track [Choi et al., 2021].

Frequency Domain

Cui et al. [2015], Jaitly and Hinton [2013] used *vocal tract length perturbation*, where the spectrograms are subject to frequency warping. Park et al. [2019] used rectangular masks over the spectrograms to train their model for speech recognition. Schlüter and Grill [2015] used modest pitch-shifting by scaling linear-

frequency spectrograms vertically, and applied time-stretching by doing the same on the horizontal axis. SpecAugment [Park et al., 2019] offers manipulations on the time and frequency axes of the spectrograms. SpeakerAugment [Wang et al., 2023b] does the same and uses vocoding techniques to alter pitch and formats in a disentangled manner. Basak et al. [2021] used the WORLD vocoder to impose pitched melodies on spoken sentences.

As spectrograms can be thought of as pictures, other transformations such as cropping, rotation, or blurring can also be used. *SimCLR* [Chen et al., 2020] is a contrastive learning method that utilises these types of augmentations to generate positive pairs whose embeddings are drawn towards a maximal agreement during training, thereby making the model's embeddings robust against augmentation and prioritising features relevant to class differentiation.

Latent Space Domain

Less frequently used, but still equally valid, is augmentation in the latent space of NN embeddings. The SMOTE technique mentioned in Section 2.2.1 can also be regarded as an augmentation technique of this variety. Nachmani and Wolf [2019] interpolated between the latent representations of two voices, which is analogous to creating a voice whose timbre lies between the timbre of two other voices.

2.3 Objective Function

An objective function is the global metric by which a model can determine how well it is performing. It combines all loss and regularisation components. Generally, it is more computationally feasible to minimise an objective function, and so the model's goal is to reach the global minimum of the cost function via gradient descent (discussed at the end of this subsection).

Loss components in an objective function represent a measurement of distance between a model's prediction and the ground truth values. Regularisation can be thought of as a force that stops the network from over-fitting to a problem parameterised by its loss function components. This can come in many forms, such as adding a secondary component to the objective function, restricting the

model's ability to adjust its weights either directly or through activation function clipping, adding dropout layers to a NN, the application of transfer learning, or simply adding noise to the data itself [Lee et al., 2019, Schlüter and Grill, 2015, Kusner and Hernández-Lobato, 2016].

2.3.1 Distance Metrics

The task of regression requires a model to predict a scalar or vector consisting of continuous data based on a set of input predictors. For such models, the objective function will usually rely on distance metrics to compute the distance between its predicted output and the ground truth data. The most common metrics used to determine similarity or dissimilarity between vectors are presented below.

The first of these metrics is the *mean absolute error* (MAE) metric. The absolute (non-negative) distances between each pair of corresponding components of two vectors are summed (this method of vector summation is known as the $L_1 norm$). All units of distance between the vectors therefore have an equal impact on the summed result, which is then averaged across the samples to compute the MAE loss (commonly referred to as the L_1 loss in ML):

$$\mathrm{MAE}(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|, \tag{2.3}$$

where n is the total number of samples, i is the sample index, and y and $\hat{y}$ are the true and predicted vectors, respectively.

An alternative to this is the *mean squared error* (MSE) metric. This takes the sum of the squared distances between each component of the compared vectors (this method of vector summation is known as the $L_2 norm$). Unlike MAE, squaring the errors means that larger component-wise distances will contribute more heavily to MSE than they would in MAE. All distances are then averaged across the samples to compute the MSE loss (commonly referred to as the L_2 loss in ML):

$$\mathrm{MSE}(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{2.4}$$

A further adaption to this metric is the *root mean square error* (RMSE), where the square root of the MSE value is obtained (making this loss computation more similar to the Euclidean distance):

$$\mathrm{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.5}$$

The *cosine similarity* gets the angular distance between two vectors, therefore providing a similarity measure that is not influenced by the magnitude of the vector and is capped between the values of one and minus one, conveying maximal or minimal similarity, respectively:

$$\mathrm{CS}(y, \hat{y}) = \cos(\theta) = \frac{\mathbf{y} \cdot \hat{\mathbf{y}}}{\|\mathbf{y}\|\|\hat{\mathbf{y}}\|} = \frac{\sum_{i=1}^{n} y_i \hat{y}_i}{\sqrt{\sum_{i=1}^{n} y_i^2}\sqrt{\sum_{i=1}^{n} \hat{y}_i^2}} \tag{2.6}$$

2.3.2 Classification Metrics

The task of classification, unlike regression, requires a model to predict a discreet output representing a category, based on set of input predictor values. For binary classification tasks, a specific type of objective function is used which relies on the *binary cross-entropy loss* (BCE). This measures the distance between the true and predicted probability distributions, parameterised as:

$$\mathrm{BCE}(y, \hat{y}) = -(y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})), \tag{2.7}$$

where y is the discreet label, $\hat{y}$ is the predicted value, and $\log$ is the natural logarithm.

For multiclass classification where there are more than two categories to choose from, the *categorical cross-entropy loss* (CCE) is used, which generalises the BCE function to handle multiple classes, parameterised as:

$$\mathrm{CCE}(y, \hat{y}) = -\sum_{i} y_i \cdot \log\left(\hat{y}_i\right), \tag{2.8}$$

where y_i is the true probability distribution of a sample belonging to class i, and $\hat{y}_i$ is the corresponding predicted probability distribution (and is the output of a

softmax function applied to the model's output for class i).

Finally, to produce the actual classification prediction, the class containing the maximum value in the class probability distribution is selected.

Classification Evaluation Metrics

To understand how well a model performs when classifying data, it is important to have a clear understanding of the different types of evaluation metrics available. These take into account the different types of predictions a model can make when considering ground truth labels.

A simple *accuracy* metric computes how many predictions are correct. However, this does not provide information about how well the model performs w.r.t. individual classes. It also gives us no information regarding how biased the model may be towards a particular class.

Fawcett [2006] offers a comprehensive explanation on more informative combinations of metrics. Considering a binary classification task (the logic of which can be extended to multi-classification), there can either be negative or positive prediction. If ground truth labels are known, then predictions can be broken down further into the following list and illustrated in Figure 2.1:

- true positives (TP)
- false positives (FP)
- true negatives (TN)
- false negatives (FN)

True-positive rate (also known as *sensitivity* or *recall*), reflects the proportion of positive classes that are correctly predicted by a model. The inverse is intuitively the case for the *false-positive rate* (also known as the *false alarm rate*).

In the context of binary classification, receiver operating characteristic (ROC) graphs are plots of the FP rate against the TP rate on the x and y axes, respectively. An example of this is presented in Figure 2.2[1]. The point (0,1) (the maximum

[1]By Masato8686819 - CC BY-SA 3.0, `https://commons.wikimedia.org/wiki/File:ROC_curve.svg`

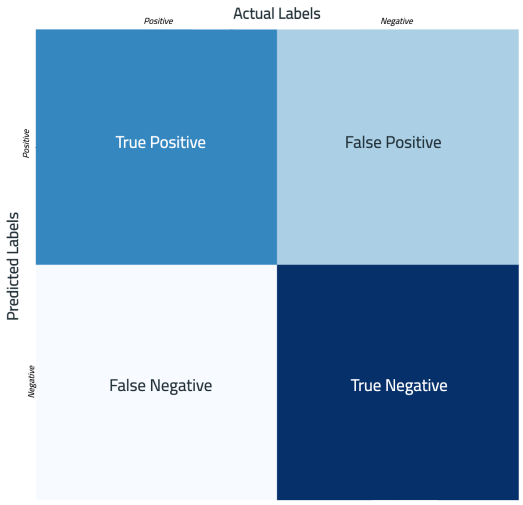


Figure 2.1: A confusion matrix, labelling all types of predictions for a binary classification task

top left corner of the graph) represents the performance of the perfect model. Models producing results closer to the left-hand side of this graph (near $x = 0$) can be thought of as conservative, only predicting positive cases when they are very certain. The diagonal of this graph represents 'random guessing' rates, and therefore any model that produces an ROC close to this line is considered to perform no better than chance. A binary classifier producing results on the lower right triangle of the graph, however, can simply have its outputs inverted, thereby mirroring its performance across the diagonal of the graph to produce symmetrically better results.

A hyper-parameter controlling the decision boundary threshold can be adjusted on a given model. Determining this threshold is usually dependent on the cost of false negatives or positives. Plotting the performance of a binary classification model where the decision boundaries are adjusted from one extreme to the other results in an arc. The area under the curve (AUC) is a quantitative method of determining how well the model performs, converting the 2-dimensional ROC into a single scalar value. ROCs have the desirable characteristic of being insensitive to class distribution, while accuracy, precision, and F-scores are sensitive. For multilabel classification tasks, the performances across multiple binary classifications for each class must be aggregated. In order to obtain an average performance

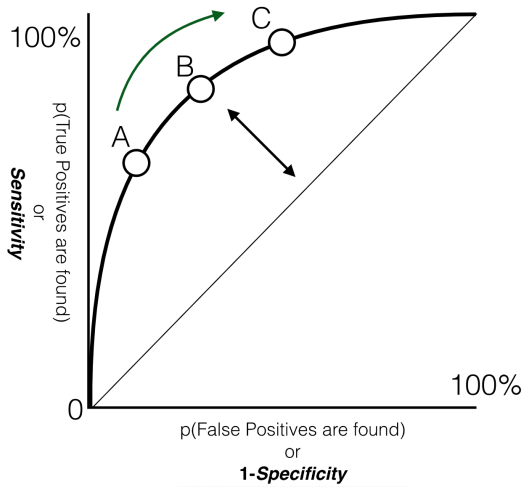


Figure 2.2: ROC graph demonstrating how the decision boundaries can be adjusted to produce an arc in the sensitivity-specificity space. A larger area under the curve between the ROC and the diagonal means higher performance.

in such instances, an average between these metrics must be calculated for each label [Choi et al., 2021].

2.4 Gradient Descent

An *optimiser* describes the mechanism in an ML model that adjusts its parameters in response to a loss value. In the context of NNs, the optimiser uses a parameter called the *learning rate* to determine the factor by which the network's weights are adjusted. A form of the *gradient descent* algorithm is the type of optimiser most commonly found in NNs.

Upon generating a loss value from the loss function, a model is able to determine the gradient of a designated cost function. From this gradient, the model can then determine in what direction and by what magnitude the weights should be adjusted in order to lower the loss value. This process is repeated every time a new loss value is generated from newly observed data until it reaches a minimum of the cost function. Other mechanisms, which are not covered here, assist this process in manners such as avoiding local minima in the objective function. It is also desirable that the cost function is convex, guaranteeing an optimal solution

with no local minima or saddle points.

The Adam optimiser has been used most consistently throughout recent advancements in NN-based tasks. Kingma and Ba [2014] report that Adam is designed to adjust its parameters automatically, is highly robust, requires little memory, requires virtually no tuning, and outperforms many of its predecessor optimisers.

Chapter 3

Introduction to Neural Networks

Having covered the main principles of ML, the focus can now be moved to NNs, which describe a specific family of ML models. Although the first NNs were proposed 80 years ago [McCulloch and Pitts, 1943], computational abilities, ML techniques, and data storage capacities have improved to facilitate the implementation of NNs to degree that has been revolutionary in the field of computer science today.

3.1 Modules and Mechanisms

A simple NN with a single layer relies on the same ML principles as a linear regression model, possessing the same capabilities. This section describes a list of common NN modules and mechanisms, that when stacked and combined, unlock the potential of ML architectures capable of complex tasks far beyond the capabilities of linear regression and other ML models.

3.1.1 The Perceptron

The smallest component in a NN is the *perceptron*. Each perceptron in a NN has multiple weighted inputs, as well as its own bias value, as seen in Figure 3.1. Each connection to a neuron is weighted. These weights are initialised by randomly sampling a probability distribution or by receiving their states' values from a previously trained model's parameters [Leglaive et al., 2015, Bengio et al.,

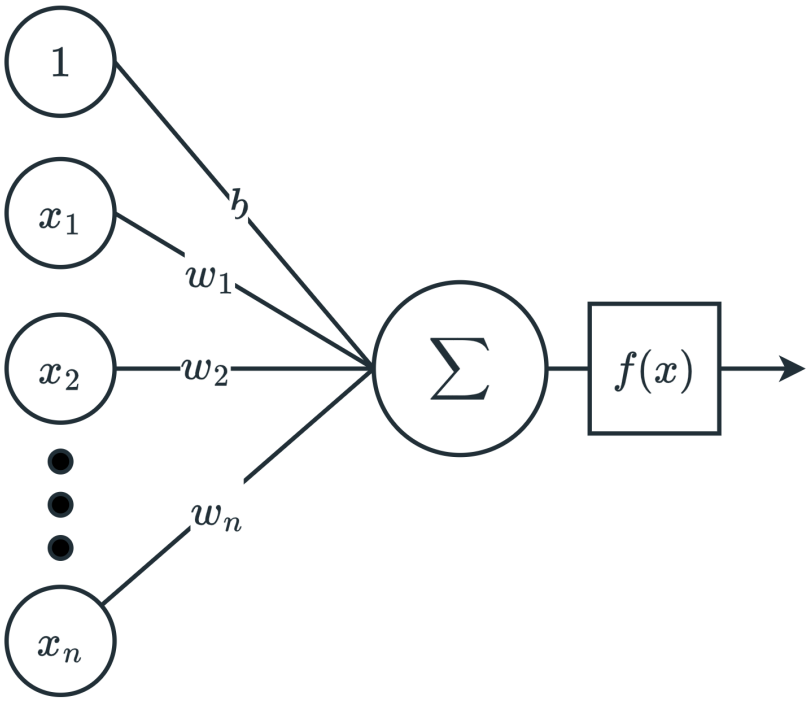


Figure 3.1: Diagram of the perceptron, where b represents the bias, w_n represents the nth weight, and $f(x)$ represents the activation function.

2006]. The combination of the bias and its inputs allows the perceptron to function like an information *gate* such as the gates *AND*, *OR* and *NOT*. Following more recent literature, perceptrons will herein be referred to as *neurons*.

3.1.2 Activation Functions

Before being passed to another neuron, the outgoing signal of a neuron passes through an *activation function*. This transforms the signal in a non-linear manner that is compressed to the limits of the activation functions' output. These functions are key to the non-linear capabilities of NNs.

Figure 3.2 illustrates the nature of the Sigmoid, Tanh, Step, and ReLU functions, the latter of which replaced the hidden layer Sigmoid functions for better system performance. The ideal type of activation layer depends on the type of output that the model is expected to produce.

The *softmax* function is a special activation function that ensures all the el-

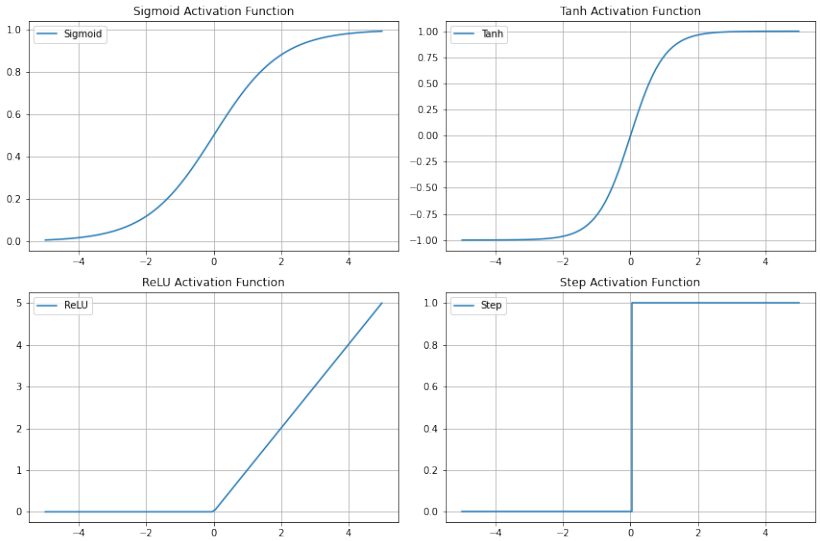


Figure 3.2: *Sigmoid*, *ReLU*, *Tanh* and *Step* activation functions

ements in a vector sum up to 1. It is therefore commonly used to convert logit vectors from previous layers to probabilities and is computed as follows:

$$\text{softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}, \tag{3.1}$$

where z_i represents the ith element of the input vector z, e is Euler's constant, and $\sum_{j=1}^{N} e^{z_j}$ represents the summation of all exponentiated elements in the input vector.

3.1.3 Dense/Linear/Fully-Connected Layer

In NNs, neurons are arranged in groups called *layers*. The manner in which neurons of a layer are connected to neighbouring layers defines what type of layer it is. It is the addition of non-linear functions and hidden layers between the inputs and outputs that allow NNs their non-linear mapping capabilities. The following subsections describe several common layers used in NNs.

When every output of an array of neurons is connected to every input of a proceeding array of neurons, this connectivity is referred to as a *dense*, *linear* or *fully-connected* layer. They are useful when there is no prior knowledge of how

any of the features generated from the previous layer should influence the next layer. Neurons within these of these layers are *not* connected to one another.

3.1.4 Convolutional Layer

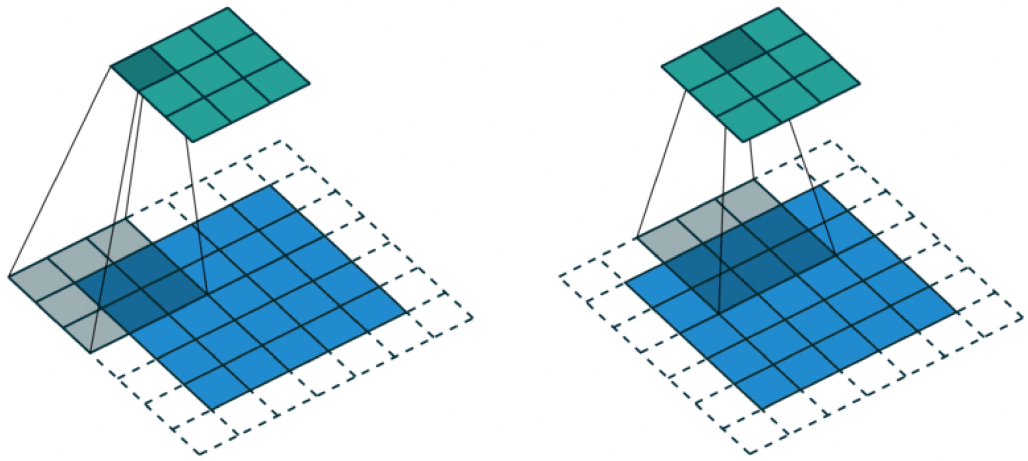
The linear layer does not, however, provide an economical or computationally feasible solution when working with data structures where the coordinates of neurons are important, like pixels in an image. A typical example of a 2-dimensional array is an image, and the positional arrangement of its pixels (and all derived features) will be referred to as a *feature map*.

A mechanism called the *convolutional* layer enforces sparse connectivity between its input (the first of which is the original image) and output feature maps. A NN model based on convolutional layers is intuitively called a convolutional neural network (CNN). It achieves convolutions by applying a kernel of a given shape (which consists of randomly initialised values) to its input feature map. The kernel is applied to each section of the input feature map, covering the space of its own shape. The sum of the element-wise products of the kernel and the windowed feature map determines the convolution layer's output value for the input feature of focus. It therefore acts as a convolutional filter and is applied as a 'sliding window' that moves across each axis of the feature map to generate a new filtered feature map. Convolutional layers have a padding option, which can be employed to ensure that the convolved feature map is of the same dimensions as the previous input feature map.

The *stride* parameter in convolutional operations determines the number of features skipped as the kernel window slides across dimensions. Figure 3.3[1] illustrates how each of the operations described so far works together to create a convolutional layer.

Pooling layers are commonly used in conjunction with convolutional layers. These use a kernel to scan the input feature maps in the same sliding-window manner as the convolutional layer's kernel, and output the average, maximum, or minimum value of the windowed feature map, depending on which type of pooling

[1]By Omegatron, reproduced by the Expat Licence, `https://commons.wikimedia.org/wiki/File:Convolution_arithmetic_-_Padding_strides.gif#filelinks`



(a) (b)

Figure 3.3: Convolutional process on a 2D feature map, using a kernel of size 3, stride of 2, and padding of 1. The green tiles represent the output feature map while the blue ones represent the input feature map.

layer is chosen. This results in an output feature map that is inversely proportional to the size of pooling layer's kernel size, effectively downsampling the image. However, some researchers challenge the value of max-pooling layers, producing comparable results with CNNs that use *only* convolutional layers [Springenberg et al., 2015].

The receptive field of a neuron in a convolutional layer describes how many of the surrounding input features it retains information for. As more convolutional and pooling layers are added, the receptive field increases, and the output features become more abstract in nature. Figure 3.4[2] is an illustration of how this works. In some applications, *causal* convolutions are necessary, where the receptive field can only include features of the past.

Fully-convolutional networks describe a network that does not utilise any other type of layer in its embedding process. These are inherently able to consider temporal features across input data, since the final layer's residual outputs have a global receptive field. They are favourable to convolutional networks with ap-

[2]By Aphex34 - CC BY-SA 4.0, `https://commons.wikimedia.org/w/index.php?curid=45679374`

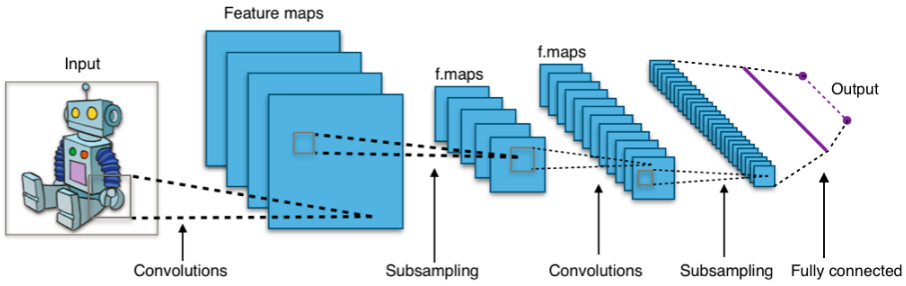


Figure 3.4: A diagram of the convolutional process.

pended dense or recurrent networks when it is necessary to be conservative with the number of tunable parameters, the convergence time, or when the model must accommodate inputs of arbitrary sizes [Choi et al., 2017, Chandna et al., 2019, Kameoka et al., 2020, Zhu et al., 2017].

There are also transposed convolutional layers. These are commonly used to transform a compressed version of data back into its original uncompressed format, like the decoder of a convolutional autoencoder. Some researchers such as AlBadawy and Lyu [2020], Kumar et al. [2019], Kong et al. [2020a] have used them in generators of a GAN to upscale to the dimensional space of the desired data format. To generate a bigger feature space than its input, it heavily pads its input features so that performing a standard convolution would involve the kernel sliding over more pixel values provided by the padding. Further details on how transposed convolutional layers work are presented by Dumoulin and Visin [2018].

CNNs also use *channels*, which refer to the number of feature maps that are analysed in each convolutional layer. Images are often encoded in RGB format, and so the first convolutional layer they encounter can facilitate an individual channel for red, green, and blue feature maps. Each channel undergoes its own convolutional kernel. As more kernels are used, more channels are generated, providing opportunities to learn more features.

3.1.5 Normalisation Layer

As NNs are trained, the distribution of inputs can vary widely between batches during training. This can be problematic as the shift in weights in response to gra-

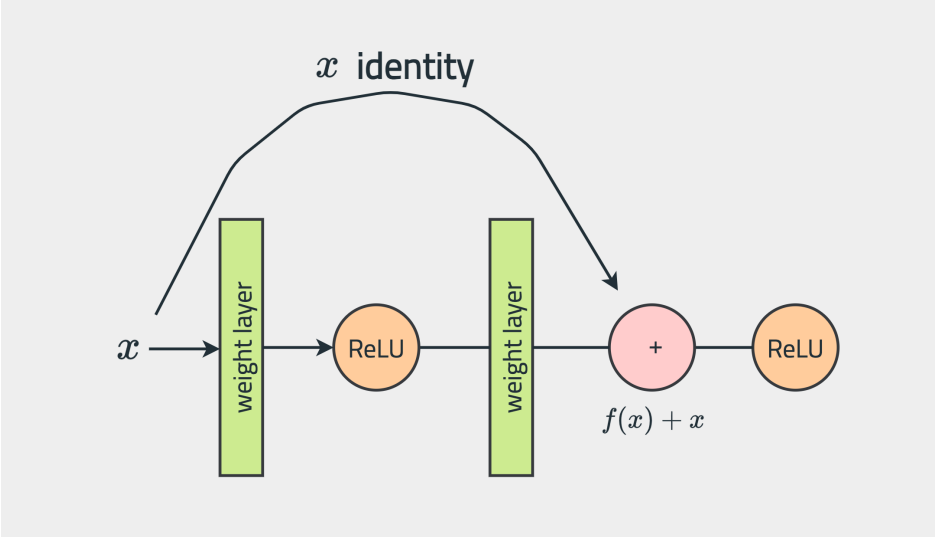


Figure 3.5: Flow chart depicting the use of skip connections as detailed by He et al. [2015]. x goes through multiple transformation processes considered as $f(x)$, the output of which is reunited with x by addition.

dient descent is made under the assumption that all other weights remain the same. This is of course not the case and forces the optimiser to pursue a moving target. *Batch normalisation* is a process where the output of a layer is standardised, as described in Section 2.2.4, across an incoming mini-batch (if the batch size is 1, then batch normalisation will not be applicable). This standardisation prevents weights from drastically changing. It improves the rate of convergence of the networks and, as a side effect, imposes some regularisation [Ioffe and Szegedy, 2015].

3.1.6 Skip Connections

Skip connections, or *residual* connections, were first introduced by He et al. [2015] for image recognition tasks. Figures 3.5 and 3.10 show skip connections, depicted by linking one layer's output to another that is multiple layers or transformations ahead of it. Figure 3.5 shows a chain of transformations summarised as $f(x)$ outputs an embedding that is added to its input via a skip connection that bypasses the function. Therefore, the parameters of $f(x)$ are optimised to produce a *residual* embedding.

As NNs get deeper when more layers are stacked together, the issue of vanish-

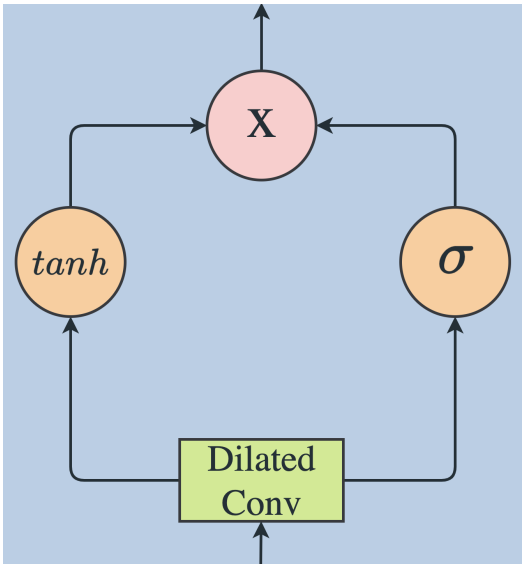


Figure 3.6: Depiction of a gated block, where a $sigmoid$ signal is gating a $tanh$ signal (or vica versa)

ing gradients becomes more prevalent, reducing the network's capacity to back-propagate efficiently towards layers further back in the architecture. Skip connections are used to mitigate this effect, allowing for easier optimisation. They also enable feature reusability by sending earlier information unfiltered further upstream to the destination point, without being affected by the layers in between. 'Short' skip connections usually occur between consecutive convolutional layers that don't change in tensor output shape, while 'long' ones usually occur between encoders and decoders, or any mirrored parts of a symmetrical NN architecture.

3.1.7 Gated Blocks

A gated block describes a mechanism in a network that decides how important information is, therefore providing some primitive notion of attention based on the input. This is achieved by processing an input through two parallel paths: one that contains a linear or nonlinear transformation, and the other that provides the 'gate' signal, such as a sigmoid function transformation. Figure 3.6 provides an illustration of this setup.

Gated Linear Units

Introduced by Dauphin et al. [2017], gated linear units (GLUs) are mechanisms that allow an incoming signal to gate in a manner that provides the flow of information with nonlinear capabilities while providing a linear path for the gradients during backpropogation. They are usually used after a causal convolutional layer to facilitate sequential predictions without requiring recurrence. To achieve this, a GLU requires two inputs from the previous layer, each of which has a separate set of weights and biases. The output of a GLU is the element-wise product of one input by the sigmoid-activated output of the other. This is parameterised in Equation 3.2 where: $\otimes$ represents element-wise multiplication; X is the output of a preceding layer; $\mathbf{W}$ and $\mathbf{V}$ are the individually determined weights while $\mathbf{b}$ and $\mathbf{c}$ are their biases; σ is the sigmoid activation function, and $h_l(\mathbf{X})$ is the GLU output.

$$h_l(\mathbf{X}) = (\mathbf{X} * \mathbf{W} + \mathbf{b}) \otimes \sigma(\mathbf{X} * \mathbf{V} + \mathbf{c}) \tag{3.2}$$

3.1.8 Recurrent Layer

Recurrent neural network (RNN) modules are a type of mechanism that allow NNs to deal with sequential data such as weather reports, audio data, or language. These mechanisms allow the network to take past information into consideration and process sequences of variable lengths. This property is achieved through the use of *parameter sharing* across timesteps. The prediction of an RNN's neuron at timestep t is fed back and reused at the neuron's inputs in conjunction with the next piece of data in the time series to make a prediction at the time step $t + 1$. See Figure 3.7[3], where the left side of the image presents a diagram of the RNN, and the right side presents the *unrolled* version with time moving from left to right. RNNs are considerably more computationally expensive than other layers described so far, as each element in the sequence must be computed recurrently before the next element can be predicted. The RNN undergoes backpropagation *through time*. This can lead to *vanishing gradients* as the weights' influence on

[3]By MingxianLin - CC BY-SA 4.0, `https://commons.wikimedia.org/wiki/File:RNN.png`

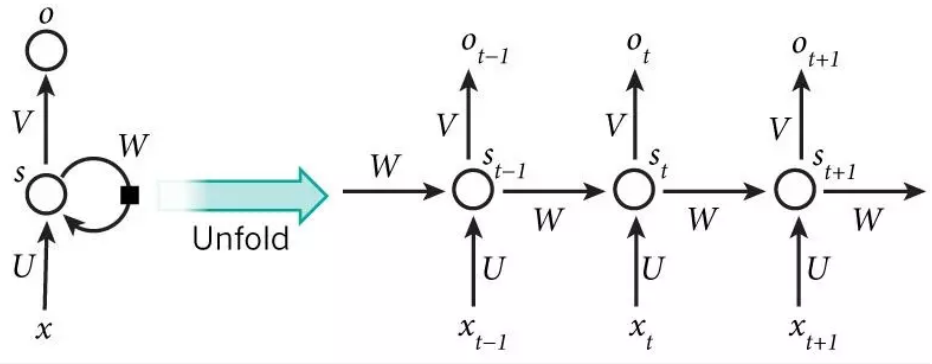
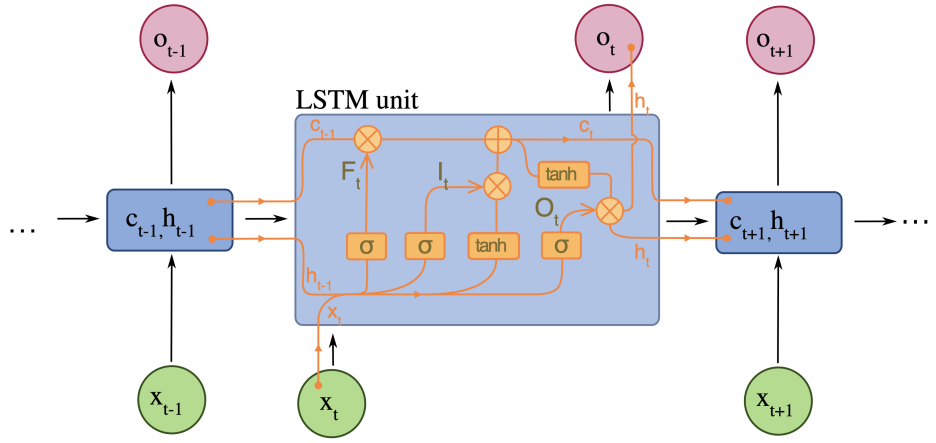


Figure 3.7: A diagram of the RNN process unfolded across time, where: U, W, and V are the weights for the input, hidden and output states respectively; and x, s, o and t represent the input, hidden, output and timestamp, respectively.

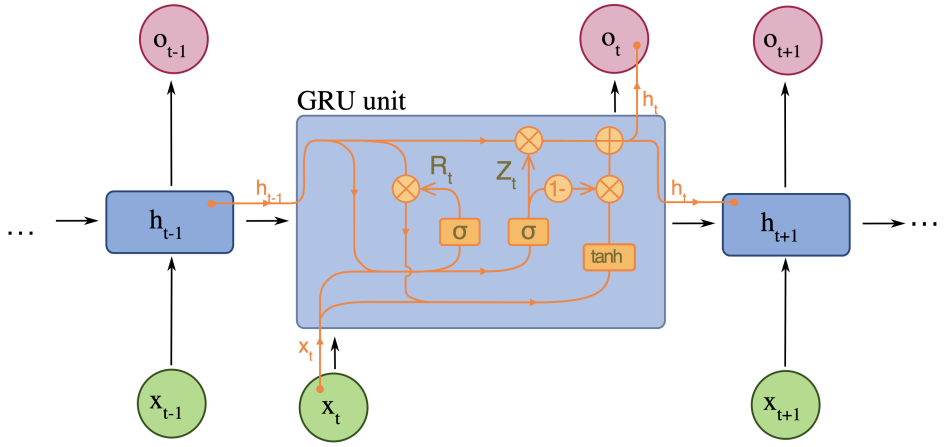
the outcome is diminished as time reverts, meaning that this model inherently assumes that information further in the past is less important. However, in many types of sequences, this is not the case.

The *long short-term memory* (LSTM) [Hochreiter and Schmidhuber, 1997] and *gated recurrent unit* (GRU) cell improves upon this issue, by making use of a hidden state cell that can carry information through many timesteps without degradation. As shown in Figure 3.8[4], these types of RNN use three gated connections (discussed in Section 3.1.7), allowing them to have individual control of how much input, previous, and current memory data are kept. Bidirectional implementations of these RNNs (BLSTMs and BGRUs) also exist, where both past and future information are combined to influence predictions, inherently requiring twice as much memory. GRUs have fewer parameters than LSTMs and, therefore, take less computational time to train.

[4]By Ixnay, - CC BY-SA 4.0, `https://commons.wikimedia.org/wiki/File:Long_Short-Term_Memory.svg`, `https://commons.wikimedia.org/wiki/File:Gated_Recurrent_Unit.svg`



(a)



(b)

Figure 3.8: Diagram of the inner architecture of the (a) LSTM and (b) GRU units.

3.1.9 Attention Layer

Attention Mechanism

The attention mechanism provides networks with the ability to pay attention to certain sets of features, irrespective of their temporal position, but dependent on their content relevance. Unlike RNN layers, attention mechanisms offer the more advanced capabilities of dealing with hidden states of variable sizes, capturing long-range dependencies in sequences, and parallelisation.

Attention mechanisms have been used as an independent structure to connect encoders and decoders. Bahdanau et al. [2016] introduced the concept of an attention layer as follows: The idea of attention is parameterised by weights α. The activation outputs of the preceding LSTM layer h are each multiplied by these weights to produce weighted observations. The sum of these weighted activations gives us the *context vector*, c. The equation for this is shown in Equation 3.3, where i refers to the current timestep, and j refers to the index of all other time steps in the input sequence.

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{3.3}$$

The context vector is fed to the current state s of an output RNN in conjunction with the previous state of that RNN (s_{i-1}) and output y_{i-1} to generate a prediction.

All outputs of elements in the sequence input are subjected to a softmax function to ensure all weighted activations sum to 1. The values for each weight can be learned through a single feed-forward layer, a:

$$\alpha_{ij} = a\left(s_{i-1}, h_j\right). \tag{3.4}$$

The attention mechanism has been used widely in the literature as a more powerful RNN with the ability to know where in a sequence the important information resides, without concern for vanishing gradients. Many of the more recent systems described in the following sections of this review of the literature will have attention mechanisms, although they may not be explicitly referenced.

Simplified Attention Mechanism

Raffel and Ellis [2016] suggest a simplification of this mechanism, where weights α_{ij} are learned by the function $a(h_j)$ instead of $a(s_{t-1}, h_j)$. Therefore, the content embedding does not require a proceeding RNN output layer, and all observations of h are summarised. However, this proposed method will not take into account the temporal order of sequences.

Self-Attention Layers

Vaswani et al. [2017] proposed an implementation of Bahdanau et al. [2016]'s attention mechanism that allowed them more flexibility in designing architectures. Their *self-attention* layer facilitated sequence transformation while using its *own* attention mechanisms to achieve this, while previous implementations of the attention mechanism required an encoder and decoder structure as its input and output, respectively.

The self-attention layer can handle changes in length to the input sequence without inducing a change in the number of parameters. It can perform sequence-to-sequence, label-to-sequence, or sequence-to-label tasks. Thanks to extensive use of matrix operations, self-attention can perform significantly efficient parallel processing, which makes very deep networks feasible.

Self-attention is computed using three learned matrices, referred to as *Query* (Q), *Key* (K), and *Value* (V). These can be conceptualised as linear layers, which take position-encoded embeddings of elements in the input sequence. These layers are combined to determine an attention weight for each input in a sequence. Once calculated, weights for all inputs are divided by the square root of the word embedding size, and put through a softmax function. The finalised weights are multiplied by the incoming value (representing an embedded element of the input sequence).

This concludes what is referred to as the "Scaled Dot-Product Attention". In the Transformer model, as proposed by Vaswani et al. [2017], multi-head attention is used, which applies the self-attention module multiple times in parallel to enhance the network's ability to model multiple relationships in the sequential input data. This seminal work has been highly influential across the landscape of deep

learning and has led to extremely powerful natural language processing (NLP) models such as BERT [Devlin et al., 2019] and GPT-3 [Brown et al., 2020].

3.2 Architectures

This section presents several specific NN architectures and systems, comprising various combinations of the mechanisms presented in the previous section. These particular architectures are presented as they have become prevalent in the field of machine learning for audio.

3.2.1 WaveNet

The WaveNet model has become a seminal piece of neural architecture and has been adopted by many researchers for its waveform-generative capabilities. It was introduced by van den Oord et al. [2016a] as a generative text-to-speech (TTS) audio model, influenced by previous architectures used in computer vision [van den Oord et al., 2016b]. It works as an autoregressive model, using causal convolutional layers to output a categorical distribution on each audio waveform sample x_t.

A network of this description can take advantage of the fact that conditional predictions of multiple time steps can be made in parallel, since all time steps for a given piece of audio are already known. Generating new audio with WaveNet, however, cannot make use of future samples. The original WaveNet implementation is, therefore, strictly autoregressive during inference, making it an especially slow process when considering the large number of previous samples required to provide a decent estimate of the next sample.

To mitigate the vanishing gradient issue commonly found in sequential modelling, *dilated* stacked convolutional layers are used, where filters are spread over areas much larger than their kernel size, as seen in Figure 3.9. By ignoring intermittent values between the neurons used for computation, these layers are able to work with significantly downsampled representations of the audio. Unlike pooling or striding convolutions, the output of these layers will be the same size as their input. A stack of these dilated layers allows networks to have large receptive

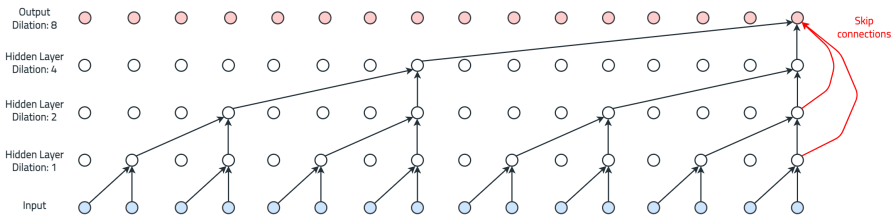


Figure 3.9: Flow diagram depicting a stack of dilated, causal convolution layers. The flow of information between layers is shown with arrows.

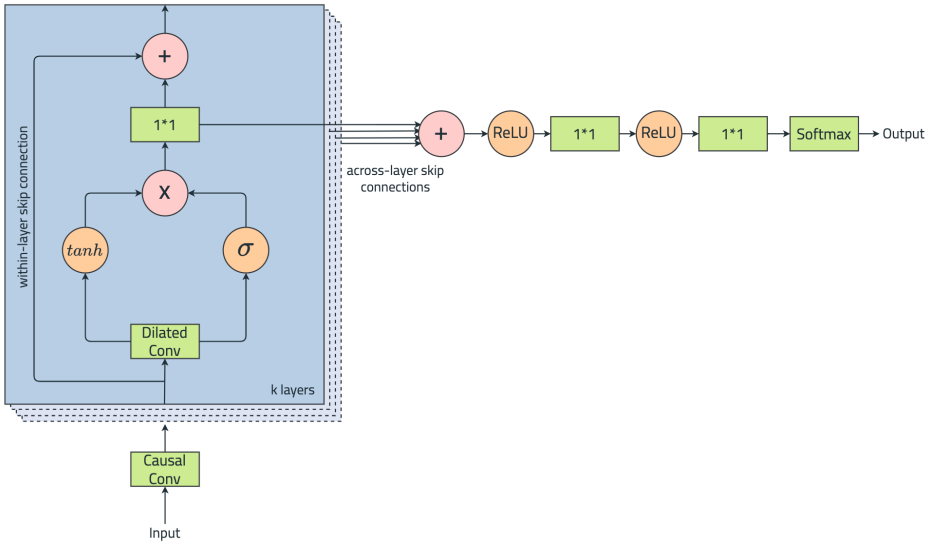


Figure 3.10: Flow diagram depicting the WaveNet's architecture.

fields without incurring excessive computational cost.

The output of each convolutional layer is fed to a residual block, as shown in Figure 3.10, where the residual blocks include a gate mechanism consisting of $tanh$ and $sigmoid$ activation paths in parallel, parameterised as

$$\mathbf{z} = \tanh\left(W_{f,k} * \mathbf{x}\right) \otimes \sigma\left(W_{g,k} * \mathbf{x}\right) \tag{3.5}$$

where: σ is a sigmoid function; k is the layer index; f and g represent the filter and gate; and W is the convolutional filter. The residual block's auxiliary outputs are summed and passed through the chain of ReLU functions, 1×1 convolutional filters and a softmax function to produce the next predicted sample.

With WaveNet, predictions can be conditioned on global or local features such

as speaker identity labels or spectrogram data, respectively.

3.2.2 Autoencoders

Standard Autoencoders

An *autoencoder* is an architecture consisting of an encoder and decoder. The encoder performs dimensionality reduction, reducing the input dimensionality to a specified vector size that best describes the multivariant nature of the data in a compact representation. Due to the restricted size of the encoder's output embeddings, this is commonly referred to as the *bottleneck*. The decoder then resynthesises the original data from this compact representation. The reconstructed data is compared to the original data, and a distance metric between the two is determined, from which the network can learn. The main loss for a standard autoencoder is shown in Equation 3.6, where g_ϕ represents the encoder function, f_θ represents the decoder function, n is the batch size, x is the input data, and ϕ/θ are the encoder/decoder weights.

$$L(\phi, \theta) = \frac{1}{n} \sum_{i=1}^{n} [x_i - f_\theta(g_\phi(x_i))]^2 \tag{3.6}$$

Variations of the encoder include the *denoising* autoencoder which adds noise to the input data but compares reconstructions to the noiseless version [Lewis et al., 2020]. As no labels are explicitly used in a standard autoencoder, these can be considered as unsupervised learning systems.

Variational Autoencoder

The *variational autoencoder* (VAE) was proposed to produce a seemingly novel output that mimics the distribution of the dataset on which it was trained [Kingma and Welling, 2014]. To model the intractable posterior distribution of a dataset $p(z|x)$, the approximate isotropic Gaussian distribution $q(z|x)$ is instead used. The Kullback-Leibler divergence (KLD) is calculated between the two distributions (see Equation 3.7) during training. This is reformulated in conjunction with the autoencoder reconstruction loss to give us the complete objective function of

the VAE shown in Equation 3.8 which comprises the reconstruction loss (left component) and the KL divergence (right component) [Kingma and Welling, 2014, Kumar, 2019, Goodfellow, 2016]. As the variational generative aspect of the VAE requires a stochastic sampling of learned Gaussian distributions, this would normally cause a break in the backpropogation process. Instead, VAEs must use the *reparameterisation trick*, where only the means and variances of the Gaussian distributions are learned. A stochastic process can then sample distributions possessing the learned statistics. Some excellent and concise descriptions of the Bayesian theory behind the VAE architecture can be found in [Leglaive et al., 2020, Luo et al., 2020b, Kumar, 2019, Kameoka et al., 2020].

$$D_{KL}(q_\phi(z|x)||p_\theta(z)) = \mathbb{E}_{q_\phi(z|x)}[\log(q(z|x)) - \log(p(z|x))] \tag{3.7}$$

$$L(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p_\theta(z)) \tag{3.8}$$

3.2.3 Generative Adversarial Networks

The Generative Adversarial Network (GAN) proposed by Goodfellow et al. [2014], is another network architecture designed to generate data that fits the distribution of a given dataset X. This is achieved by pitting two models against each other: the generative model (G) and the discriminator model (D). A GAN is trained so that G can generate data that is indistinguishable from that of the real dataset, while D discriminates whether it has witnessed synthesised or real data.

The generator comes in the form of a predefined NN architecture chosen by the user and is fed a prior of input noise $p_z(z)$. This is mapped through the generator to generate data that is similar to the instances of X. D takes either the output of G, or samples from X, and produces a single value representing the probability that the data came from X.

The objective of a GAN is for D and G to possess Nash equilibrium, implying that given the current state of one model, the other model has found the best response, and neither model can unilaterally improve its performance. Both models are trained in a turn-taking fashion by freezing one model's parameters at a time. Simultaneous training would not allow one network to have an advantage over the

other and avoid the Nash equilibrium objective. It is this close proximity tension between the models that encourages optimal convergence. Data is presented in batches consisting of examples taken from both X and G. By training D to predict the probabilities of its input's authenticity and training G to make this deceive D, the resulting objective function of the GAN is

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \quad (3.9)$$

After incrementally training both models, presuming G is generating high-quality samples and D is achieving 50% accuracy, the GAN is considered to be well trained. At this point, D can be removed, and we are left with G, a high-quality generator model.

GANs can suffer from *mode collapse*. This is when a GAN focuses primarily on certain modes in a multi-modal distribution and ignores all others due to insufficient incentive via the loss function. This may happen because D is focused on one particular mode that contributes greatly towards its objective function, and therefore G learns to model this mode well in response, for which it will be rewarded just as much or possibly more than learning the other modes. Mode collapse usually results in G producing outputs that no longer evolve. Attempts to mitigate mode collapse in GANs involve modifying the loss function, such as using the *Jenson-Shannon Divergence* or *Wasserstein* methods which measure the distances between the modelled and real data distributions [Tolstikhin et al., 2019, Kumar, 2019].

Along with mode collapse, GANs are particularly difficult to train, encountering issues such training instability, loss saturation, non-convex objective function optimisation, sensitivity to hyper-parameters, vanishing gradients, and the crucial need for high-quality, large datasets.

3.2.4 Teacher-Student System

The *teacher-student* paradigm is a good example of transfer learning (described in Section 4.4.3), where a *pretrained* model can be used to generate pseudo-labels. The pretrained model providing labels to an untrained model is analogous to a

teacher giving the student answers, hence the name. From the handed-down pseudo-labels combined with the corresponding input data, the student model is trained until it outperforms the teacher model.

3.2.5 Transformer Model

The Transformer is an encoder-decoder system, consisting primarily of deep stacks of multi-headed self-attention layers, skip connections and positional encoding mechanisms. In recent years, this architecture has experienced great success, achieving state-of-the-art (SOTA) results in many sequence-based tasks. Vaswani et al. [2017]'s work introduced the Transformer and shows that the self-attention-based module can be put in place of both convolutional and recurrent modules in a network. Its receptive field is global but possesses quadratic memory complexity. Section 3.1.9 describes how the self-attention layers work, and examples of its success will be presented in Section 4.4.

3.2.6 Diffusion Model

Among the most recent and successful branches of generative NN architectures is the *diffusion* model. It has the ability to iteratively convert noise into a structured form of data that is similar to samples from the dataset on which it is trained. The theory for these models came from non-equilibrium thermodynamics [Sohl-Dickstein et al., 2015].

The forward process of the model involves gradually destroying samples from the training data by subjecting them to invertible transformations in the form of additive Gaussian noise, until all traces of their original structure have disappeared. The model can learn the iterative process that brought the data towards this state of pure noise. After being trained, the model can then proceed through the backward process of the transformations it has learned to generate noiseless samples, the structure of which mimics the distribution of the training dataset. While the input for generation after training is usually randomly sampled noise, there are a number of conditional parameters that can be added to the training strategy that allow novel samples to be generated based on such a prompt. These models come with the benefit of not requiring any adversarial training, which, as previously

described, comes with many challenges. They have been known to outperform GANs and are able to perform a number of different tasks including novel data generation, manipulation, in-painting, and data description.

Chapter 4

Background

Having covered the basics of ML techniques and NN architectures, this chapter presents the remaining literature review, covering several areas relevant to the thesis. Sections 4.1 to 4.4 discuss the topics of the human voice, the perception of sound, spectral representations of audio, and NN systems designed for audio and voice analysis, disentanglement, conversion, and synthesis tasks.

4.1 The Voice

This section starts with a basic description of the vocal anatomy, proceeding with a description of how the vocal organs work together to produce vocal sounds. Subsection 4.1.2 discusses how the voice is used for singing and how it is aurally perceived. Subsection 4.1.3 concludes with a discussion of the differences between singing and speech content, and how these relate to existing datasets.

4.1.1 Physiology

The vocal organs refer to the collection organs that contribute towards the production of any type of vocal sound. This section is accompanied by several diagrams to aid readers in visualising how the vocal apparatus works. An excellent source of scientific explanation for how the voice works is that of Sundberg [1987], from which most of the information in this section originates unless cited otherwise.

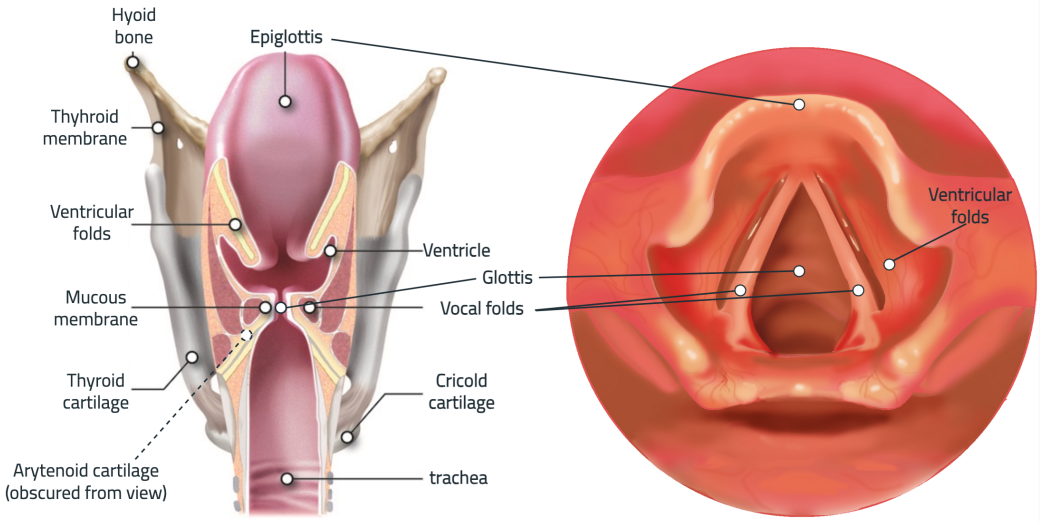


Figure 4.1: Anterior view (left) and plan view (right) of the larynx.

Figures 4.1[1] and 4.2[2] are provided to aid the reader in visualising the descriptions in the proceeding subsections.

Use of Respiratory Function

This section begins with a description of the respiratory system of the vocal organs. The vibrating component, air, is provided by the lungs. During inhalation and exhalation, an *underpressure* or *overpressure* (w.r.t. atmospheric pressure) is induced in the lungs and carried up to the glottis. It is commonly called subglottal pressure (SGP). The diaphragm, intercostal, and abdominal muscles are employed to modulate this pressure, directly influencing voice amplitude. Generally, there is a monotonic relationship between SGP and pitch and loudness in singing, particularly pronounced in the higher registers of vocalists. Consequently, various respiration techniques exert a discernible impact on these aspects of vocal sound production.

[1] **Left diagram credit**: "Cenveo - Drawing Larynx and vocal cords - English labels" at AnatomyTOOL.org by Cenveo, license: Creative Commons Attribution, (adapted). **Right diagram credit**: "Slagter - Drawing Larynx and vocal cords - no labels" at AnatomyTOOL.org by Ron Slagter, license: Creative Commons Attribution-NonCommercial-ShareAlike, (adapted).

[2] By Tavin - Own work, CC BY 3.0, `https://commons.wikimedia.org/w/index.php?curid=17388339`

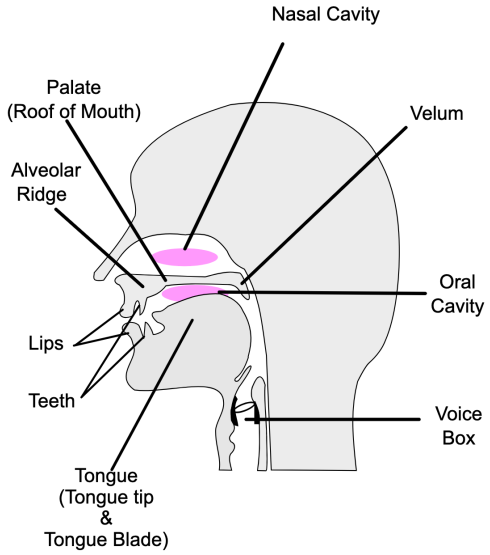


Figure 4.2: Lateral view of the vocal tract.

The *functional residual capacity* (FRC) of the lungs describes the volume of air they contain in the absence of passive inspiratory or expiratory forces. Upon descending below the FRC, the contraction of the abdominal muscles becomes vital to maintain SGP, ensuring the continuation of voiced sound production. In speech, sentences and phrases are mostly initiated with 50% of the lungs' *vital* (total) *capacity*, which is generally just over the FRC. The amount of inhalation increases proportionally to how loud a speaker wants to be. In singing, the inflation of the lungs is much higher and varies depending on the type of singing required. Long passages of projected singing often require close to 100% of the vital capacity of the lungs.

Vocal Folds

The *vocal folds* are thin protrusions at the position shown in Figure 4.1. They are located in the larynx (commonly called the voice box) and are approximately 9-20mm in length, depending on the size of the neck. The length of these folds dictates the pitch range of the vocalist. They are covered by a mucous membrane, and are separated by a gap called the *glottis*.

Adduction describes the state of the vocal folds when they are brought to a

close at their anterior by the *lateral cricoarytenoid* muscles. As air passes through the constricted passage, a Bernoulli force is induced, causing the vocal folds to vibrate which propagate the air up through the larynx. The rate of vibration of the vocal folds determines the fundamental frequency that is produced during voiced vocal sounds. Different frequencies of vibration occur when the length of the vocal folds is stretched, thereby increasing tension. This process creates what is called the *source* or *excitation* signal - a spectrum of sound which is yet to undergo the filtering process that takes place in the passage between the vocal folds, mouth and nose.

Abduction on the other hand, describes the folds' state when they are pulled apart from each other through the use of the *posterior cricoarytenoid* muscles, thereby removing the vibrating phenomenon. This state allows the vocal organs to produce more breathy utterances such as whispering, passive breathing, and other unvoiced vocal sounds.

Directly above the vocal folds are the ventricle folds, also covered by a mucous membrane. They are not usually used for voiced phonations and provide regulatory and protection functions in the respiratory system. However, they do contribute to the quality of pathological vocal sounds, which will be discussed in the following subsections. Above these is the epiglottis, which is a flat, flap-shaped cartilage structure, whose function is to stop food and liquids from entering further into the respiratory system.

Phonation modes describe the types of vocal sounds that are produced as a result of the behaviour of vocal folds. The timbral qualities that each mode has are the result of how the folds oscillate. There are four types of phonation modes called *neutral*, *pressed*, *breathy* and *flow* phonation. Figure 4.3 presents them in each quadrant of a 2-dimensional plane, parameterised by glottal airflow and SGP. However, while the 2D space might imply equal distributions of phonations to each quadrant, this is not the case in reality. A detailed summary of how each mode occupies this space is provided by Proutskova et al. [2013], who also points out that phonation modes are not linked to singing registers, introducing the question of how changes in pitch affect timbre while these vocal mechanisms remain constant.

When phonation occurs in the falsetto register, the vocal folds no longer make

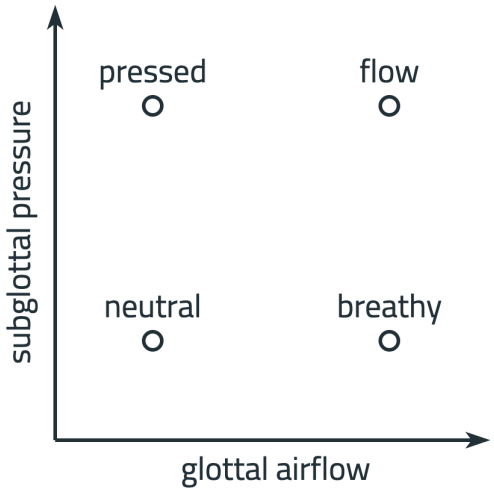


Figure 4.3: A 2-dimensional representation of the placement of the phonation modes, where the vertical axis and horizontal axis correspond to glottal airflow and subglottal pressure, respectively.

contact to close completely. Whispering is considered to be an extreme case of breathiness where the vocal folds are no longer vibrating enough to produce a periodic excitation source that would otherwise provide the voice with pitch. *Voiced* and *unvoiced* (VUV) sounds refer to the sound of the voice when it is with or without vibrating vocal folds, respectively. Whispering is rarely used in singing and has less scope for timbral diversity.

The *breathy* qualities in a voice occur when air passes through the partially opened glottis. This causes low *glottal resistance*, which refers to the ratio between SGP and transglottal airflow. Examples of this can be heard when one exhales as they lift something heavy, or when Marlyn Monroe uses her signature breathy voice. A phonation of this configuration requires an increase in SGP to retain the same amplitude.

Vocal Tract Filtering

After passing through the vocal folds, the source signal of the voice is propagated through the narrow cavity of the *vocal tract*, which describes the space between the vocal folds, lips, and nostrils. Like the epiglottis, the *velum* is another valve located at the back of the throat that connects the oral cavity to the nasal cavity.

This can open or close to regulate airflow in the nasal passage.

The area function of these cavities imposes a filtering function on the source signal and is controlled by the following: tongue shape, jaw, larynx, velum, and lips. These are called the *articulators*. Their adaptive nature allows the source signal to undergo a wide range of filtering functions that allow for the diversity of timbre and morphological nature of vocal sounds. A *phone* describes a distinct vocal sound, irrespective of how it relates to the construction of words in a language. A *phoneme* on the other hand, is the smallest unit of sound into which a given language can be broken down.

The vocal tract filters the excitation signal, attenuating some frequencies more than others. This process forms peaks in the spectral representations of vocal sounds called *formants*. The position of these formants accounts for the sound that is perceived once the excitation signal has been filtered and radiated through the lips, which is the result of particular articulator configurations.

4.1.2 Voice for Singing

Having covered vocal anatomy, attention will now be given to how the voice is used in singing. This section describes the voices physical restrictions and capabilities, opportunities for expressivity, vocal technique taxonomy, and the concept of voice identity.

Vocal Register

The idea of vocal *register* alone lacks a widely accepted definition. One definition that appears to fit most concepts and classifications of register, however, is that provided by Hollien [1974], who writes that register is a "range of consecutively phonated frequencies which can be produced with nearly identical vocal quality and that ordinarily there should be little or no overlap in fundamental frequency between adjacent registers". Here, we continue with Sundberg [1987]'s description of the most widely referenced registers.

Male registers consist mainly of two registers. The *modal* register describes the range of phonations apparent in a male's 'normal' speaking voice, utilising the lower frequency ranges. The *falsetto* register accommodates frequency ranges

above those of the modal range, allowing males to produce vocal sounds more similar to females in relation to their timbre and range.

Recognised female registers include the *chest*, *middle* and *head* registers. Some unusual registers shared between both genders is the *pulse* register, also called *vocal fry* (the latter of which will be more frequently used in this thesis). This defines the voice when it possesses 'creaky' qualities in the lowest frequency range of the voice, perceived as individual voice pulses. Conversely, the *whistle* register describes the extreme high end of the frequency range, characterised by its whistle-like timbral qualities.

Expressivity

In singing, expressivity is established by utilising whatever variations are available that are not explicitly established in predefined musical composition. The nature in which these compositions are retained come in different forms offering varying levels of detailed information (and a proportional amount of restriction), such as oral transmission alone, condensed music representations such as lead sheets, or classical music notation. Dimensions of expressivity are achieved through diversity in singing techniques, and differ in use between genres. Kayes [2015] explores this phenomenon and provides a description of how vocal mechanisms contribute to this effect, and also notes the tendencies of singers when considering what is comfortable for their voice in different musical contexts.

In most classical music, the communication of emotions is dictated by musical structure and content, often encapsulated in a score. Singers who closely honour the composer's direction and genre authenticity would be considerably constrained when conveying their own expression of an emotional state, as the melody carries musically semantic implications, the dynamics dictate intensity, and the tempo evokes an urgency connected to emotional arousal [Coutinho et al., 2014].

However, the *timbre* of the voice is only coarsely dictated by terms in the score, such as *dramatico* and *dolce*. These directions are often very general in nature, allowing for considerable variance in timbre, giving the singer an opportunity to become expressive and make use of the diversity of their voice as they see fit. The

timbral qualities of the voice are determined by the singing technique being used, which can be thought of as a preset of vocal mechanism configurations.

The exhalation of air is determined by a combination of glottal resistance and SGP. In singing, SGP has been found to be strongly correlated with the frequency and loudness of phonations [Bouhuys et al., 1966], which means that it can vary significantly across a musical phrase. As the difficulty of musical passages increases w.r.t. notes per second, proficiency in active use of the diaphragm muscles is therefore required to produce smooth singing.

Scherer et al. [2017] present a novel experiment where singers were asked to portray different emotions while singing an arbitrary sequence of notes and meaningless syllables, therefore eliminating musical and linguistic semantics and relying heavily on timbral diversity. Their results suggest that features such as loudness, dynamics, high perturbation variation, and formant amplitude are correlated with the emotions being conveyed.

Coutinho et al. [2014] explores the capabilities of the voice with regard to expressivity and how emotion can be communicated between production and perception. They express how essential it is to explore the dynamics of this dichotomy in order to devise systems that can digitally manipulate expressivity in the voice. As emphasised by Stylianou [2009], it is essential to understand how voice parameters contribute towards timbre and therefore expression, if we are to build systems that model this accurately.

Taxonomy of Vocal Sounds

There has been a considerable amount of disagreement and miscommunication between voice specialists regarding the taxonomy of vocal techniques and sound, as documented by Sundberg [1987], Proutskova [2019], Hollien [1974], Mörner et al. [1963], Gerratt and Kreiman [2001], which makes it difficult to present an exhaustive or precise list. García-López and Gavilán Bouzas [2010] compare values and perspectives of artistic and scientific professions specialising in the voice, and suggest that the differences between these two communities lead to a convoluted and inconsistent tapestry of technical terminology.

The biological production of singing, as noted by Sundberg [1987], is far eas-

ier to describe than how humans perceive it, which involves more subjectivity and analogous terminology. Heidemann [2016] provides an interesting approach on the subject of singing perception, where she presents a system to describe the perceptual vocal timbre of the voice. García-López and Gavilán Bouzas [2010], Sundberg [1977], Kayes [2015], Zhang [2016] provide detailed information on how vocal production techniques influence the perception of a singer's voice.

Gerratt and Kreiman [2001] investigated the phenomena of 'nonmodal phonations' that are considered pathological or less common among vocalisations, such as vocal fry and *supraperiodic* phonation, which describe a vocal signal that consists of repetitive patterns in the waveform that stretch beyond a single fundamental frequency period, or a pair of cycles alternating in intensity. Blomgren et al. [1998], Michel and Hollien [1968], Gerratt and Kreiman [2001] have demonstrated through perceptual tests that the acoustic features of vocal fry are highly salient compared to other modal phonations.

Voice Identity

Singers have a wide range of timbral diversity at their disposal that can be used to produce expressive singing. Singers' timbral diversity is subject to a variance that is limited by their physiological capabilities due to their age, gender, size and shape. However, variances in aural cues are not only bounded by the capabilities of the singer, but also preferences in performance style, such as those relating to dynamics, tempo, pitch and rhythmic deviation.

Perceived voice *identity* can be thought of as the acoustic effect of all parameters relating to expressivity and physiology working together, allowing listeners to discriminate between voices. Complex combinations of many of the aural cues mentioned above lead to a wide timbral palette that, when intentionally controlled by the singer, are manifested as *singing techniques*.

As a result, timbral features are heavily entangled with singing technique information, which accounts for the specific configurations of the vocal organs that allow a singer to express themselves with a wide timbral range. The entanglement between singing technique and voice identity information presents an interesting disentanglement problem to be explored in subsequent chapters.

In this thesis, the term singing *style* is reserved to describe the manner in which singing techniques are used, such as the temporal behaviour of the *vibrato* technique [Yamamoto et al., 2021]. It is important to make this clear, as much of the literature in the field of ML concerning voice conversion is inconsistent with voice attribute terminology, or is vague when defining it.

4.1.3 Speech and Singing

Spoken voice conversion and TTS tasks are in far more demand in the industry than singing-related tasks and have therefore monopolised the spotlight in voice analysis and synthesis research. Public domain speech datasets also vastly overshadow singing datasets in size and availability [Meseguer-Brocal et al., 2020], and so the research related to singing in this field has been limited in its capabilities in comparison to that of speech.

There is an understandable temptation to consider speech and singing datasets as one of the same domain. As will be clear in Section 4.4, most research related to singing analysis and synthesis has been inspired or copied from similar tasks used in the field of speech. However, it is important to consider how these two domains of vocal recordings differ before considering which methods should be transferred from one domain to the other. This section presents some of the main structural differences between speech and singing content, highlighting the inherent biases that come with them in terms of vocal use. Other work that considers the comparison of these domains includes Demirel [2022], Nercessian [2020], Saitou et al. [2004].

Domain-Specific Characteristics

In most Western music styles of singing, a strong majority of singing content is occupied by long sustained vowels. Consonants naturally occur at the start or end of notes, as they are not elongated in most cases.

Singing content features sustained spectral states while speech produces a rapid spectral morphology due to the fact that phones do not need to be sustained for any length of time as long as they are communicated effectively to a listener. This also highlights the fact that speech will always possess a higher

rate of syllables-per-minute than singing, especially since it is not restricted to a rigid metric time. The ratio of consonants to vowels is therefore higher in speech, thereby increasing the amount of spectral noise and its importance in speech.

Pitch contours are of great importance in singing. They are based on the targets of discrete pitches and durations, specific to the style and composition of the music being sung. Preservation of the structure of musical components is vital in singing, although pitch transposition and dynamic use of time such as (*rubato*) for the purpose of performance and/or expression are typical transformations in music. The arrangement of pitch and duration forms the concepts of harmonic and rhythmic structures of music. As long as these structures largely remain intact, pitch transposition and rubato are permissible in music for re-orchestration. Repetition of harmonic and rhythmic structures (such as verses and choruses) is also typical in sung content. Microvariations in these structures related to intonation, pitch, volume, and rhythm are often utilised by singers to deliver a unique interpretation of the musical score with expressive qualities. Researchers who explored the role of these variations in singing content include Savery et al. [2020], Saitou et al. [2004].

In speech, the requirements for preserving pitch contours and phone durations are much less rigid. This is because these attributes of vocal sounds are directly related to *intonation* and *prosody*, which are not dependent on a hierarchical structure of discrete units for frequency or timing. Instead, these components, along with vocal timbre, obey their own intrinsic functions, and can be transformed dynamically with interdependency to faithfully convey the grammatical, emotional, and semantic meaning of speech [Wagner and Watson, 2010, Scherer, 2003, Nolan, 2020].

Choosing a Dataset

Vocal datasets will vary in many ways, and so careful consideration must be given to the task before choosing the right dataset. For example, if the task is synthesis-based, datasets without audio files will be insufficient. Some other considerations include whether the dataset contains:

- single or multiple languages, vocalists, or microphone recordings

- singing or speech
- *a capella* or mixed music content
- processed or unprocessed vocal recordings (compression, reverb, pitch-shifting etc.)
- singer identity labels for each recording
- short or long excerpts, and overall size/duration of the dataset
- parallel recordings of multiple domains or styles
- near or far-field microphone techniques
- environmentally noisy or clean studio recording circumstances

There are a number of publications that provide content-specific lists of vocal datasets. A quick search online can produce numerous examples of curated lists, specific to the field or researchers' needs. These come in the form of online blog posts [3], literature reviews [Yamamoto et al., 2022, Rosenzweig et al., 2020], or lists associated with challenges related to speech [ISCA, 2023] and singing [Toda et al., 2023]. Recently, it has become increasingly common to utilise published collections that use scripts to collect raw audio from websites such as YouTube [Yamamoto et al., 2022, Kalbag and Lerch, 2022]. Singing datasets of particular relevance to this thesis include:

- MIR-1k [Hsu and Jang, 2010]
- MUSDB18 [Rafii et al., 2017]
- DALI [Meseguer-Brocal et al., 2020]
- MedleyDB [Bittner et al., 2014, 2016]
- NUS-48E [Duan et al., 2013]

[3] `https://github.com/RevoSpeechTech/speech-datasets-collection`, `https://openslr.org/resources.php`

- Phonation Mode Dataset [Proutskova et al., 2013]
- VocalSet [Wilkins et al., 2018]
- LibriSpeech [Panayotov et al., 2015]
- Voxceleb [Nagrani et al., 2017]
- VCTK [Veaux et al., 2017]

4.2 Perception of Sound

In this section, research and techniques related to sound perception are discussed. Section 4.2.1 covers the computation of dissimilarities, inferences from such data, and the experimental design of listening studies, while Section 4.2.2 discusses statistical analysis, clustering techniques, evaluation metrics and subjective evaluation.

4.2.1 Listening Studies

Multidimensional Scaling Techniques

It has been widely accepted that timbre is impossible to quantify through a single measurement, as it encompasses a large number of attributes [McAdams et al., 1995, Wedin and Goude, 1972, McAdams et al., 1992]. In perceptual listening studies, there have been disagreements between participants about what they have heard, as detailed in Section 4.1.2, making it difficult to create a unified perception-based analysis of the timbre. The use of multidimensional scaling (MDS) is first discussed in order to provide context when discussing timbral perception.

A typical method for developing timbral maps of an instrument is to conduct a listening test where participants are asked to rate the dissimilarities between every pair of sounds in a given set. These can be presented as dissimilarity matrices and converted into a representation of fewer dimensions via MDS. This approach is especially useful to represent the cognitive process of how people perceive and generalise the diversity of data within a given domain [Mugavin, 2008].

The first to use MDS to represent perceptual data were Kruskal [1964], Shepard [1962a,b], employing 'stress' metrics which quantify a solution's goodness-of-fit to the data, and 'nonmetric' techniques (due to the rank-ordered nature of the data) to reflect perceptual data monotonically in the MDS representation. This innovative work has paved the way for investigating timbre spaces and has been widely used in the relevant literature [Gerratt and Kreiman, 2001, McAdams et al., 1995, Wedin and Goude, 1972, Krimphoff et al., 1994, Serafini, 1993]. Within this field, MDS has undergone multiple adaptations and refinements. Carroll and Chang [1970] improved on the classical MDS with the INDSCAL algorithm, which avoids rotational invariance for simplified dimension interpretation and provides weights relating the contribution of participants' collected data to these dimensions [Mugavin, 2008]. INDSCAL has been used by Grey [1977] in his influential research on timbral maps for multiple instruments. Interpreting the distribution of data points across dimensions in an MDS representation requires a post-hoc analysis. McAdams et al. [1995] combined perceptual dissimilarities with acoustic parameters to generate timbre maps using the CLASCAL algorithm [Winsberg and De Soete, 1993], which verified dimensional interpretation by incorporating the values of these acoustic parameters into the MDS calculations.

Interpretation of Timbral Space

McAdams et al. [1995] concluded that the first two dimensions of their timbral maps were related to the *temporal* and *spectral envelope*, and that their data was highly correlated with that of Krimphoff et al. [1994]. The strong impact of these attributes on the timbral space is agreed upon by Grey [1977], Iverson and Krumhansl [1993]. Iverson and Krumhansl [1993]'s studies reported *spectral centroid* and 'sharpness of attack' as the meaning of the primary dimensions. Grey [1977] deduced that the three axes in his 3D representations could be interpreted as *spectral energy distribution*, *spectral flux* and *temporal patterns* in relation to the attack portion of the sound. While it is apparent that the first two dimensions often share similar meanings between experiments, the interpretation of the last dimension (in most cases the third) often varies between results, which may relate to the difference in datasets being used.

The pitch of a musical instrument can also have a considerable impact on how it is perceived. Grey [1977] found that the bassoon playing in a very high register was perceptually close to the brass instruments. The work of Wedin and Goude [1972] suggests that while elements such as attack transients assisted people in *identifying* an instrument, they did not have a significant effect on the perceptual structures of instrumental tones.

McAdams et al. [1995], Iverson and Krumhansl [1993] suggest that timbral associations with specific instrument mechanisms provide a bias in giving dissimilarity raters a cue that such a sound should be distinguished from others based on its class (Krimphoff et al. [1994] refer to these types of aural features as acoustic 'parasites'). Grey [1977] reports a similar phenomenon, in which particular articulations can disrupt clustering behaviour among instruments that would normally sit well within a family of instrumentation. In contrast, Iverson and Krumhansl [1993] investigated the influence of complete tones and their corresponding onsets / remaining segments on timbral spaces, concluding that the importance of instrumental timbre in entire tones cannot be attributed to either their onsets or remaining segments. They also concluded, like many others, that centroid frequencies and amplitude envelopes contributed heavily to the timbral space.

Experiment Design

In relation to timbral perception, there is little research focused on investigating what the ideal experimental design should be for listening studies. As a result, there is not much guidance on the following points.

The phrasing of the required task varies between experiments even though the data being sought out is the same. Iverson and Krumhansl [1993] for example, asked the question "How much would you have to change the first sound to make it sound like the second sound?" and presented a continuous scale ranging from "a little" to "a lot". Grey [1977] simply instructs participants to "rate the similarity of the two tones" from very dissimilar to very similar on a scale of 1 to 30, while McAdams et al. [1995] used smaller Likert scales between 1 (labelled very similar) and 7 (very dissimilar). The number of ratings required per listening session (in previously mentioned research, this ranged from 120 to 276

ratings) may also have an effect of fatigue on the listener that may significantly diminish the quality or consistency of their ratings. Listening fatigue is unavoidable after a certain period of time, so optimal durations, phrasing, and choice of interface should be carefully considered on a case-by-case basis. Grey [1977] reports that the order in which the comparisons are represented causes differences in the judgements between the participants. For this reason, it has become common practice to randomise the presented order of pairwise comparisons. Mehrabi [2018], Gerratt and Kreiman [2001] employed repetition within experiments to assess intra-participant reliability. However, even with this approach, there is still the issue of collecting saturated ratings because initial stimuli are mildly diverse in timbre (or the opposite). Most studies include a significant number of practice sessions, allowing participants to become familiar with the diversity of the stimuli before they can provide recorded dissimilarity data.

There have also been conclusions about how the profiles of participants influence their rating style. Wedin and Goude [1972], Carterette and Miller [1974] reported in their work that the use of participants with different levels of musical training did not cause differences between them. However, Serafini [1993] reported that musicians familiar with the sounds being evaluated (Gamalan instruments) attributed more importance to the attack of the sound than its resonant volume, while non-musicians' ratings reflected these equally. Gerratt and Kreiman [2001] used voice specialists in their perceptual tests on vocal pathology, which led to perceptual data that possessed strong clustering properties. However, the results of perceptual listening tests with the voice are quite different, where even voice specialists have difficulty agreeing on what was heard [Proutskova, 2019]. McAdams et al. [1995] used participants with different levels of musical training, which did not seem to influence their perception of class structures, although those with more of a musical background did offer more 'precise' ratings, which was hypothesised to be due to superior ear training.

4.2.2 Analysis and Evaluation Methods

Statistical Analysis

Greene and D'Oliveira [2005] provide an excellent breakdown of statistical analy-

sis methods that are suitable for any type of experiment in psychology. In addition to this, sources dedicated to the particular calculations used in the relevant statistical tests [Cohen, 2008, Black, 2023, Hope, 1968, Kruskal and Wallis, 1952, Mann and Whitney, 1947, Mumby, 2002] helped to explain the various experimental scenarios where terminology or applications can be ambiguous. Perugini et al. [2018] offer additional insight on power analysis, and have suggested the software `G*Power` to assist in the calculation.

Clustering Techniques

Clustering techniques are suitable methods of analysing data provided by participants of a listening study due to their unsupervised nature, as they do not impose classes on the data. McAdams et al. [1995] used nearest-neighbour clustering analysis to detect which of their participants performed significantly differently from others, highlighting instances where some individuals may have misinterpreted instructions. They used the Monte Carlo significance testing procedure [Hope, 1968] to determine the optimal number of classes that best represent the clustering nature of the data. Gerratt and Kreiman [2001] used the K-means algorithm to confirm that their dimensionality separated their 3 classes into statistically significant clusters. Iverson and Krumhansl [1993] averaged dissimilarity values across all the perceptual data of the participants to calculate the MDS spaces. Grey [1977] did similar calculations and applied the HICLUS hierarchical clustering algorithm [Johnson, 1967] to group the stimuli into clusters and assessed the compactness of these clusters.

Evaluation with Computational Metrics

In most cases of assessing the performance of an audio-generative model, it is typical to evaluate the performance of such a model using some kind of computational evaluation. The most basic form of evaluation will be the metric that was used to determine the model's performance during training. This can be used on the evaluation data to measure how well the model does its job and how well it generalises to unseen data. With generative networks in particular, as the measurement is not as simple as determining whether a label is right or wrong, other

means of evaluating the model's performance must also be considered.

The latent loss (described in detail in Section 4.4.1 is the distance between two vectors in latent space. This can be used to compare a model's predicted output to the original or target data. Third-party pretrained models such as *Resemblyzer*[4] or *Wespeaker*[5] have also been used to evaluate converted audio [Tan et al., 2021, Lei et al., 2022, Prihasto et al., 2023, Li et al., 2023], although this creates a upper bound evaluation limitation, dictated by how well such models are trained.

Subjective Evaluation

However, due to the variance in the quality of converted data, it is good practice to employ human-based evaluations. For this reason, we see listening tests being the most common type of subjective experiments used to measure the perceptual ratings based on the generated audio's naturalness or similarity to a target audio example. Human participants are expected to choose, rank, or rate the available stimuli.

The most common type of metric used to determine the quality of synthesised audio is the mean opinion score (MOS). To obtain an MOS value, multiple ratings from different participants are obtained for the same stimuli or different stimuli under the same conditions. The mean score of these values is then obtained and reported as the MOS, along with the standard deviations (although some researchers present standard error or confidence intervals).

The Multiple Stimuli with Hidden Reference and Anchor (MUSHRA) evaluation method has also been used. This involves a test where participants are given three types of stimuli: the reference, the anchor, and the test recordings. The inclusion of anchor and reference recordings allows participants to have a bounded perceptual scale to use, where the reference represents the upper bound and the anchor represents the lower bound ratings.

Guidance for the use of MOS and MUSHRA is provided by ITU-R [2015], ITU-T [2013], which also provides informed guidelines for experiment preparation, choice of assessors, scope, and effective implementation.

[4] `https://github.com/resemble-ai/Resemblyzer`
[5] `https://github.com/wenet-e2e/wespeaker`

Other types of evaluation tests include XAB tests, in which participants are instructed to choose between candidate stimuli based on how well they match a given reference stimulus, given a specified criterion [Tan et al., 2021]. There is also the preference test, where there is no target or reference stimulus present, and participants are required to simply choose the best option, again based on a given specified criterion. Another design collects pairwise ratings, in which participants are required to rate similarity (or dissimilarity) between two stimuli.

To summarise: the XAB test selects the best stimulus sample w.r.t. a reference; the preference test selects the best stimuli; and the pairwise similarity test quantifies the perceptual distance between two stimuli (one of which could be a test or reference stimulus). Popular challenges such as the Blizzard Challenge [ISCA, 2023] have offered the rationale for their choice in evaluation test methods and challenged the validity of ratings based on pairwise comparisons between two voices [Wester et al., 2016].

4.3 Spectral Representations of Audio

A fundamental consideration when designing a system for audio analysis or synthesis is how the input data will be represented. This section covers the most relevant methods by which sound has been represented in digital format for voice-related tasks.

The most basic audio representation is the result of the initial method by which sound is recorded on a digital system. After a sound has been recorded by a microphone and converted to a digital format, it is stored on a digital device in a format that reflects the fluctuations of the signal over time. This format is the most direct digital representation of the original waveform.

Determining whether the waveform representation is more appropriate for modelling than any representations derived from it requires a consideration of how much data and computational resources are available. Pons et al. [2017] showed that NNs using spectrograms performed better at audio tagging than those using waveforms, but disclaimed that larger datasets in the future will likely invert this observation. A diverse array of audio embedding types have been successfully used as input representations in the field of audio and ML research. This section

covers embedding types that are particularly relevant to MIR research. Much of the information presented in this section has been informed by Müller [2015], Velardo [2020], Gold et al. [2011], Dixon and Benetos [2020], unless stated otherwise.

4.3.1 Spectrum

A spectrum is used to define the position between a minimum and a maximum value. For most audio-relevant purposes, the frequency spectrum is bounded by the threshold of human hearing (20Hz-22kHz). In digital audio, this spectrum is represented as a set of discrete quantised points called bands, each of which is used to represent the amount of energy within the frequency region it is tuned to.

Transforming a time-domain audio waveform into a frequency-domain spectrogram can be achieved with the Discrete Fourier Transform (DFT). Given an audio signal of finite length (the analysis window), a DFT will produce an array of complex numbers that represent the amplitude and phase of the waveform across its bands. The resolution (number of bands) of the DFT is proportional to the size of the analysis window. As this spectrum contains no temporal information, it is typical to perform a Short-Time Fourier Transform (STFT), which describes the application of the DFT multiple times over a sequence of short analysis windows until the entire signal of interest has been analysed. The result is a series of short-duration spectra, concatenated over time to become a *spectrogram*. In audio, the periods by which spectra are generated are typically two to four times smaller than the DFT analysis window, allowing the spectrogram to capture high-resolution frequency information in more localised timesteps. Using overlapping windows also improves accuracy by reducing spectral leakage and smoothing out abrupt changes between adjacent windows. Computing the absolute values of the complex numbers of a spectrum produces the magnitude information, while computing their angles produces the phase information.

Adaption for Human Perception

In the majority of MIR applications, magnitude information has been regarded as the more informative data, while the phase is often considered less relevant

because of humans' perceptual invariance to it. In cases where further energy analysis is required, the magnitudes are squared to generate the *power* spectrum.

Human perception of both loudness and frequencies must also be taken into account. The human ear perceives loudness logarithmically, and relatively low-energy sound events can still be perceptually relevant if the environment is adequately quiet. To accommodate for this, the power spectrum can be converted into decibels to mimic the perception of humans using the equation

$$\mathrm{dB}(I) = 20 \cdot \log_{10}\left(\frac{I}{I_{\mathrm{TOH}}}\right), \tag{4.1}$$

where I refers to sound intensity and I_{TOH} is a reference point for the *threshold of hearing*, from which the relative decibel measurement is generated.

Humans also have a *logarithmic* perception of frequency. Their sensitivity to frequency is stronger at lower frequencies than at higher ones. Therefore, the log scale is often used for frequency to shift the importance of frequencies to the lower orders of the spectrum.

An alternative approach looks more closely at non-linear perception of frequency, and uses a custom scale to faithfully reproduce the sensitivity of the human ear. This was determined from the experiments of Stevens et al. [1937], where they measured listeners' responses to intervals across the frequency spectrum and generated a series of *critical bands*. These bands are implemented as a series of triangular filter windows applied to the spectrogram, resulting in 'mel' filterbanks. In other words, the mel filterbank is a calibrated set of spatially defined frequency bins to map the linear frequency scale to the mel scale, which is perceptually tuned to the sensitivity of the human ear.

After applying each of these transformations, the result is a log-magnitude mel-spectrogram, which is one of the most common forms of audio representation in the field of MIR.

4.3.2 Cepstrum

A 'cepstrum' is formally defined as the Inverse Fourier Transform (IFT) of the log magnitude of a DFT, but can more intuitively be thought of as the spectrum of a spectrum of a signal. Its application covers a wide range of domains, from

seismic to speech to music analysis. *Cepstrum* analysis is a tool used to measure the amount of periodicity in a spectrum over time. Due to the inverted nature of the maths involved, the terminology related to these types of transformations are partial reversals of the original terminology, such as the conversion of *spec*trum to *ceps*trum [Bogert et al., 1963].

In the same way that a DFT measures how well each sine wave resonates with the waveform being analysed, we can test the presence of periodicities against a spectrum as if it were a time-domain signal. The cepstrum is said to lie in the 'quefrency' domain, which is measured in time intervals. Strong peaks in this domain indicate which periodicities and their multiples are the most present in the frequency domain. The information relevant to the spectral envelope resides in the lower end of the quefrency axis, which also relates to the timbre of the voice and the shape of the vocal tract. The upper end of the quefrency axis contains information relevant to details such as the glottal pulse. The quefrency domain of the cepstrum can be divided to separate the contributions of the vocal tract filter and the excitation signal via *liftering* (the equivalent of filtering in the quefruency domain). The quefrency at which liftering occurs can be informed by considering what the lowest detectable fundamental frequency should be (bearing in mind that the lowest note for male singers is roughly 50Hz). By taking the logarithm of these two components of the cepstrum, they can be summed (instead of being multiplied, as in Equation 4.4) to produce the log of the speech signal:

$$\log |X(\omega)| = \log |E(\omega)| + \log |V(\omega)| \tag{4.2}$$

Mel-Generalised Cepstrum

A variation of the standard cepstrum that uses the mel scale is the mel-generalised cepstrum (MGC), which has been frequently used in speech processing [Bonada et al., 2016, Chandna, 2021, Kaneko et al., 2017]. This is advantageous over typical cepstrum analyses as it does not overestimate formant bandwidths and has a frequency resolution akin to the filter banks and phase response of the human auditory system. It is defined as the IFT of the generalised logarithmic spectrum calculated on a warped frequency scale [Tokuda et al., 1994]. The generalised

logarithmic function is defined in [Kobayashi and Imai, 1984] as the natural generalisation of the logarithmic function, parameterised as

$$s_\gamma(w) = \begin{cases} \frac{1}{\gamma}\left(w^\gamma - 1\right), & \gamma \neq 0 \\ \log w, & \gamma = 0 \end{cases} \tag{4.3}$$

where γ is a real number of $|\gamma| \leqslant 1$.

4.3.3 Spectral Envelope

The *spectral envelope* is the overall shape of the distribution of energy across a frequency spectrum. It provides a simplified representation of the amplitude-frequency plane, highlighting spectral content of the audio signal that captures timbral characteristics, peaks and troughs in the signal. As spectral envelopes are meant to convey a smooth version of the frequency response of spectral frames, they require relatively fewer coefficients to approach a useful function that approximates the true spectral distribution adequately.

Speech modelling conceptualises the phenomenon of speech as the result of two processes: the vocal tract frequency response and the excitation signal. The spectra of these components can be multiplied to produce the spectrum of the entire speech signal, formalised as

$$|X(\omega)| = |E(\omega)||V(\omega)|, \tag{4.4}$$

where $X(\omega)$, $V(\omega)$, and $E(\omega)$ represent the spectra of the speech signal, the vocal tract frequency response and excitation signal. Peaks in the spectral envelopes are heavily correlated with vocal formants. $E(\omega)$ can be approximated using a truncated cepstrum. A specific implementation of this will be described in Section 4.3.5.

4.3.4 Mel-Frequency Cepstral Coefficients

Mel-frequency cepstral coefficients (MFCCs) were originally developed for speech processing and automatic speech recognition (ASR) [Tiwari, 2009]. They are coefficients for a function that provides a rough estimate (depending on how few

coefficients are used) of a spectral envelope, making them suitable features for predicting utterances in speech and timbre in music. They have been very successful in providing a reduced representation of the voice and capturing the salient features of human perception. MFCCs are effective for analysis, but not for synthesis, as the operations required to generate MFCCs cannot be inverted to return to the audio's original uncompressed representation.

To generate MFCCs, as with the transformation from waveform to cepstrum, we first apply the DFT to a waveform and take the log of the resulting power spectrum. Mel filter-bank mapping is applied to the frequencies, producing the log mel-spectrum. Instead of an IFT, a discrete cosine transform (DCT) is applied to the spectrum[6], the result of which is the MFCC features. The number of Fourier components is typically truncated between indices 12 and 14, which has the effect of smoothing the spectrum.

4.3.5 Vocoder

The concept of the vocoder was first proposed by Dudley [1940]. It uses the principles of discrete frequency band and excitation source manipulation to represent the vocal tract and glottal pulse configurations in voice production. This process represents the source-filter model, which is still used today in modern vocoders. It allows users to manipulate spectral and fundamental frequency (henceforth referred to as F0) information independently.

This process has been converted to the digital domain in numerous implementations that have improved in quality over the decades. The STRAIGHT vocoder [Kawahara et al., 1999] was recognised as the first vocoder to compete with the more natural-sounding waveform concatenation systems (the concatenation of sonified phonemes was considered the best approach to vocal synthesis due to its SOTA results and simplicity). It improved upon issues relevant to its predecessors, which included the removal of periodicity interference, the production of smoother F0 contours and a reduction in the amount of perceived buzziness in synthesised audio. Further variations of this vocoder provided real-time analysis

[6]The DCT is very similar to the IFT, but is computationally a more reasonable choice for this application. It also has the added advantages of decorrelating information shared between mel banks, and allows for dimensionality reduction.

and synthesis capabilities, but at the cost of more simple algorithms that degrade audio quality [Banno et al., 2007].

NNs designed to synthesise waveform audio from vocal acoustic features (such as spectral envelopes or mel-spectrograms) are commonly referred to as *neural vocoders*. Examples of these can be found in Section 4.4.5.

WORLD Vocoder

The WORLD vocoder [Morise et al., 2016] was built to offer real-time applications while maintaining high-quality voice synthesis. At the time of publication, this vocoder was achieving SOTA results in objective and subjective evaluation, and has since been a popular choice of audio representation when performing voice-modelling tasks. Often in previous literature, mel-cepstral coefficients (MCCs) generated from WORLD's spectral envelope [Suzuki et al., 2022, Zhang et al., 2020, Li et al., 2022a, Tan et al., 2021, Huang et al., 2021b, Du et al., 2021, Zhang et al., 2020, Kameoka et al., 2020, Tobing et al., 2019, Chen et al., 2019, Arakawa et al., 2019, Fang et al., 2018, Kaneko and Kameoka, 2017] have been used. Sometimes the spectral envelope itself [Lu et al., 2020, Zhou et al., 2021] (with particular authors reducing its dimensionality by truncating it and applying frequency warping in the mel-cepstral domain before reproducing its spectral envelope form as log-Mel Frequency Spectral Coefficients (MFSCs) [Chandna, 2021, Nercessian, 2021, Blaauw and Bonada, 2018]) has been used instead. Very recently, there has been a trend of using WORLD simply for its F0-generative capabilities [Takahashi et al., 2023, Shen et al., 2023, Li et al., 2023, Zhang et al., 2022, Wu et al., 2022, Li et al., 2021a].

This vocoder uses several algorithms that had previously achieved SOTA results to produce the WORLD feature set, comprising F0, spectral, and aperiodic information. By default, the vocoder uses the DIO algorithm to produce the F0, the CheapTrick algorithm to produce the spectral envelope, and the PLATINUM algorithm to predict the aperiodic parameter. A visualisation of how these algorithms work together is provided in Figure 4.4.

The DIO algorithm [Morise et al., 2009] first applies a series of low-pass filters to its waveform audio input signal, until the filtered signal possesses only a

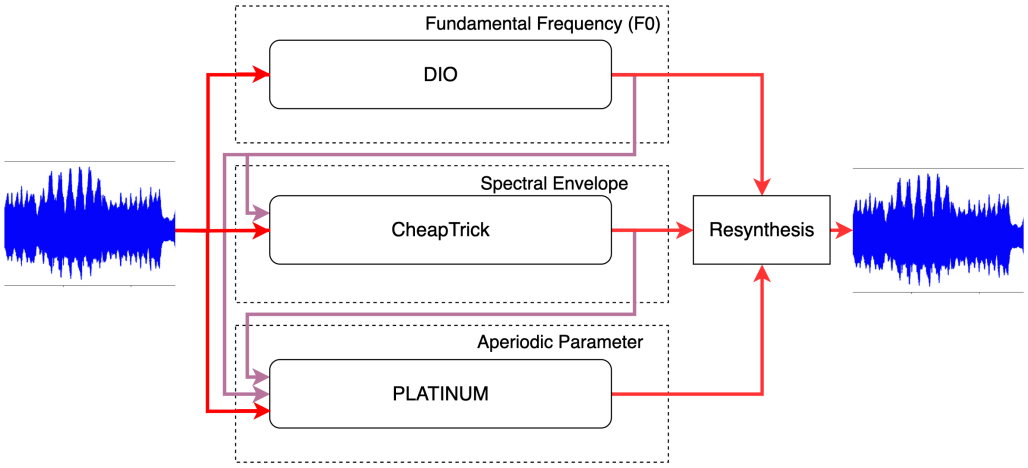


Figure 4.4: Flowchart illustrating WORLD's analysis algorithms extract F0, spectral envelope and aperiodic information.

fundamental component. As this fundamental component should resemble a sine wave, it should have the same periodicity between positive/negative zero-crossing intervals and peak/trough intervals. The standard deviations of these intervals in the filtered signal act as a reliability measure. After several F0 candidates are obtained, the one with the highest reliability is chosen as the true F0 label.

An alternative pitch estimation algorithm available to users of the WORLD vocoder algorithm[7], is 'Harvest'. It produces F0 candidates in a similar manner to DIO, and overlaps F0 candidates across neighbouring frames to calculate the likely fundamental frequencies in frames where there is too much noise. This effectively reduces the amount of false unvoiced frame predictions. Further steps involving F0 selection, contour interpolation, and VUV decision revisions, (all of which are covered in detail in Morise [2017]) contribute towards Harvest's refined pitch estimation.

The CheapTrick algorithm [Morise, 2014] uses both DIO's output and the input waveform audio signal to generate a spectral envelope. This algorithm uses a time-varying window frame in its spectral analysis function to remove the influence of the temporal position of the windowing function. Using a combination of this F0-adaptive windowing, spectral smoothing, and liftering on the signal's

[7]as provided in the Python-wrapped implementation at `https://github.com/JeremyCCHsu/Python-Wrapper-for-World-Vocoder`, `v0.3.2`

cepstrum, it generates the spectral envelope. This can then be convolved with the glottal pulse signal supplied by the DIO algorithm.

The PLATINUM algorithm [Morise, 2012, Fang, 2021] uses the output of the spectral envelope of CheapTrick and the output F0 of DIO to compute aperiodic information. Unlike previous vocoders such as STRAIGHT, which convolve an aperiodic response with white noise and a periodic response with a pulse train, the PLATINUM algorithm instead first determines the *excitation signal*, which is then convolved with the minimum phase response to produce the vocal fold vibration, as seen in Figure 4.5. However, the temporal position of each vocal fold vibration must be determined. To do this, the temporal centre t_a of each voiced section is determined. Values within the interval $t_a \pm T_0$ (where T_0 represents the F0 period) are squared, and the maximal value indicates the origin point of the vocal fold vibration. The other origins of the positions of the vocal folds can then be determined based on the F0 contour. The flow diagrams for these processes are shown in Figure 4.5. As WORLD only needs to do one convolution, it is considered more efficient than the STRAIGHT vocoder.

The excitation signal, $x_p(t)$, is obtained by taking the IFT of the extracted excitation spectrum $X_p(\omega)$

$$x_p(t) = \mathcal{F}^{-1}\left[X_p(\omega)\right], \tag{4.5}$$

which itself is determined by dividing the observed spectrogram $X(\omega)$ by its minimum phase response $S_m(\omega)$

$$X_p(\omega) = \frac{X(\omega)}{S_m(\omega)}. \tag{4.6}$$

The minimum phase response itself is calculated by computing the FT of the cepstrum $c_m(\tau)$, determined as follows:

$$c_m(\tau) = \begin{cases} 2c(\tau) & (\tau > 0) \\ c(\tau) & (\tau = 0) \\ 0 & (\tau < 0) \end{cases} \tag{4.7}$$

where $c(\tau)$ is the cepstrum of Cheaptrick's smoothed spectral output $P_l(\omega)$.

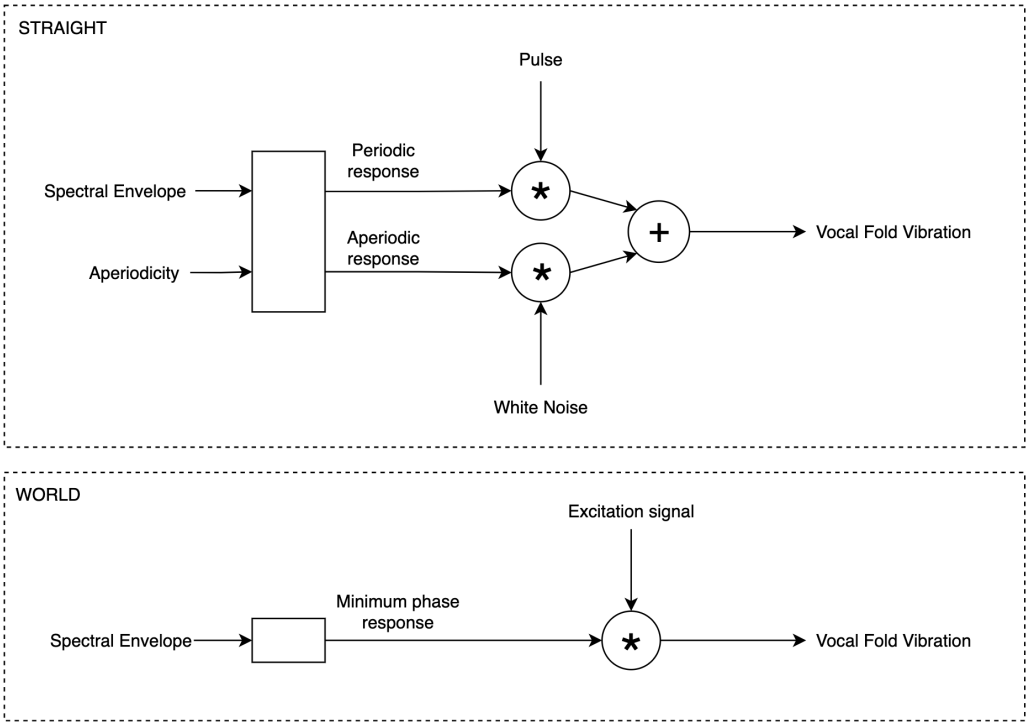


Figure 4.5: Flowcharts illustrating how vocal fold vibrations are determined in the STRAIGHT and WORLD vocoder systems. The $*$ symbol represents convolution.

$$c(\tau) = \mathcal{F}^{-1}\left[\log\left(P_l(\omega)\right)\right]. \tag{4.8}$$

.

In summary, the PLATINUM algorithm uses the chain of formulas listed above to determine aperiodic features that account for variances in the spectrum that are not accounted for by the harmonic spectral envelope.

4.4 Neural Networks for Audio and Voice-Related Tasks

This section discusses data representation transformations and adaptations of existing NN frameworks that have contributed toward analysis and synthesis tasks relevant to singing attribute conversion. It is split into subsections focussing on alternative loss components, audio analysis, disentanglement, conversion and audio synthesis.

4.4.1 Alternative Loss Components

Audio-generative networks often rely on a pixel-wise comparison for reconstruction loss, such as those described in Section 2.3.1. While this type of loss has been successfully and consistently implemented in this field of research, alternative domain-informed loss metrics have been proposed. These losses can be categorised as embedding, latent, cycle-consistency and contrastive losses, which will be discussed below.

Embedding Loss

Feature-matching describes the process of using the features generated by the activations of intermittent layers in an NN architecture [Mroueh et al., 2017, Salimans et al., 2016, Larsen et al., 2016, Kumar, 2019, Caillon and Esling, 2021] as target values from which loss values are computed. By doing this, embeddings of multiple similar networks, or embeddings generated by different input representations, are encouraged to be similar as training progresses.

A particular type of feature-matching loss is the *latent regressor loss*, which describes a measurement of the distance between the output embeddings of a network. These embeddings will often relate to more *perceptually relevant* features, representing higher-level characteristics [Johnson et al., 2016] that can be compared between original and reconstructed data, or between two different instances where the information being encapsulated in the embedding should be similar. It is also motivated by the intuition that generative networks are forced to rely on the relevant latent space embeddings in addition to other inputs.

Latent regressor loss has been frequently used to improve the results of various audio-related tasks [Donahue et al., 2017, Nercessian, 2020, Lee et al., 2019, Qian et al., 2019, Nachmani et al., 2018, Nercessian, 2020], and particularly with speaker identity embeddings [Du et al., 2021, Liu et al., 2018, Cai et al., 2020, Huang et al., 2021a, AlBadawy and Lyu, 2020]. Abstract but interpretable features can also be generated by DSP-based feature extractors, such as encoders for pitch, phones, and loudness. As these latter examples also represent high-level characteristics of an audio signal, there is no reason why considering losses between these features could not also assist models in evaluating their own performance.

Latent Loss as Regularisation

Regularisation, as described in Section 2.3, is a component of the objective function that prevents a model from overfitting to a certain subset of data. Consider an encoder that is pretrained on one domain (the source domain) of input data and an untrained encoder that will use another domain (the target domain) reflecting the same data. Examples of these domains could be solo and mixed tracks, or spectral and MIDI representations. A loss taken between the embeddings of a target encoder and a pretrained source encoder would encourage the target encoder to extract the same information as the pretrained source encoder. Chandna et al. [2020] used this technique to train their autoencoder to extract embeddings from mixed music tracks as if they were solo tracks. Luo and Su [2018] extracted MIDI representations from audio, clean vocal recordings from distorted vocal recordings, and clean solo vocal recordings from mixed music recordings.

Esling et al. [2018] used MDS across a collection of 5 previous listening studies to produce a timbral map for different instruments. A loss between a VAE's bottleneck embeddings and the timbral maps facilitated a regularisation term that would teach the VAE to extract information from its input data in a manner that matched human perception, which led to better reconstruction and generalisation. It is surprising to see that this technique has not been applied to SVAC. If perceptual distances in NN latent spaces representing vocal attributes are shown to be of a complex or non-linear nature, these could surely benefit from a similar regularisation process.

Cycle-Consistency Loss

The cycle-consistency loss was formulated in response to the challenging task of performing style transfer on data from domain X so that it matches the style of domain Y. Previously, this required *paired* data that demonstrated how similar classes differed only by their style (or attribute of interest). Zhu et al. [2017] overcame this challenge by considering two separate generators A and B that are the mathematical inverse of one another, implying that $B(A(x)) \approx x$. This equation represents back-translation, which describes a conversion process where the source and target data have been reversed. In [Sennrich et al., 2016], they used synthesised source-data in 50% of the training data, which was generated by converting already-converted data of the target class back to the source class. Cycle-consistency loss has been used in much subsequent research tackling tasks related to style or attribute conversion [Amodio and Krishnaswamy, 2019, Zhou et al., 2016, Kaneko and Kameoka, 2017, Zhu et al., 2017, Luo et al., 2020a, AlBadawy and Lyu, 2020]. In voice conversion, this has primarily been applied to the voice identity attribute. Investigating how well cycle-consistency will assist in conversions where attributes are so tightly entangled, such as voice identity and singing technique, would be an informative and progressive step in SVAC.

Contrastive Loss

Contrastive learning was briefly introduced in Section 2.1.5, which is facilitated by a contrastive loss computation.

By utilising the multi-track recordings in the MUSDB18 dataset, Lee et al. [2019] trained a triplet loss network with monophonic recordings as the anchor and mixed recordings as the positive/negative items. With contrastive learning, they were able to encourage the network to extract embeddings that were the same for same-singer content, regardless of whether the input was just vocals or a mixed track.

Wan et al. [2018] used constrastive learning to learn speaker embeddings from speaker audio recordings using the Generalised End-to-End (GE2E) loss, which is a particular contrastive loss that makes use of multiple classes within a batch, clustering techniques in the latent space, and end-to-end functionality. The model used to facilitate this was designed for extracting voice identity embeddings from mel-spectrograms. This consisted of a straightforward architecture of 3 stacked LSTM layers, the last output of which is fed to a linear layer. However, it is the computation of the loss function that is unique and is worth describing in detail here.

The GE2E loss takes M random clips of utterances i from N random vocalists j packaged as a minibatch that is fed to the network during training. The output embeddings of the model's final linear layer are first subjected to L_2 normalisation as seen in the following equation, where $\mathbf{x}$ represents input features, $\mathbf{w}$ represents model weights, and $f()$ represents the forward pass function of the network that produces the output embeddings.

$$\mathbf{e}_{ji} = \frac{f\left(\mathbf{x}_{ji}; \mathbf{w}\right)}{||f\left(\mathbf{x}_{ji}; \mathbf{w}\right)||_2} \tag{4.9}$$

The mean of all embeddings for each vocalist seen in each minibatch can then be computed to represent the vocalist k's embedding centroid, c_k, in latent space:

$$\mathbf{c}_k = \frac{1}{M} \sum_{m=1}^{M} \mathbf{e}_{km}. \tag{4.10}$$

Similarity scores are then generated between all utterance embeddings $\mathbf{e}_{ji}$ and their corresponding vocalist centroids using the equation

$$\mathbf{S}_{ji,k} = w \cdot \cos\left(\mathbf{e}_{ji}, \mathbf{c}_k\right) + b, \tag{4.11}$$

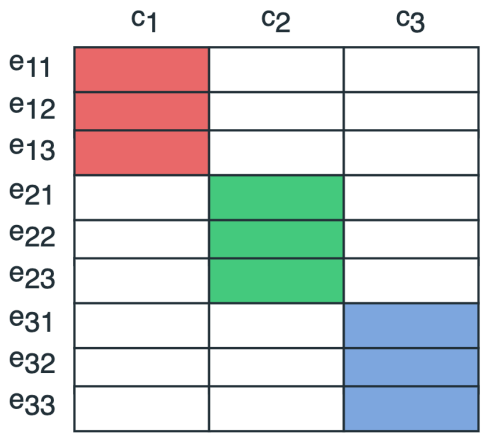


Figure 4.6: A Similarity matrix, arranged so that the first axis represents utterance embeddings j, the second axis represents the vocalist centroids k, and the cells represent the similarity scores between the corresponding pair of axes elements. In this illustration, the coloured cells represent where the identity of the utterances matches that of the centroid.

where w and b are learnable weight and bias parameters. w is constrained to be positive to ensure larger similarity scores for larger cosine similarities. This vector of cosine similarities is rearranged as a matrix, herein referred to as the similarity matrix, as seen in Figure 4.6, where unmatched (with respect to vocalists) embedding-centroid pairs are white and matched embedding-centroid pairs are in colour.

From this similarity matrix, the softmax computation of the GE2E loss can be derived, which has been shown to perform best for text-independent vocalist verification tasks, where there are no lexical restrictions on the utterances being used during training. The softmax loss for each embedding is defined as:

$$L\left(\mathbf{e}_{ji}\right) = \mathbf{S}_{ji,j} - \log \sum_{k=1}^{N} \exp\left(\mathbf{S}_{ji,k}\right) \tag{4.12}$$

The notation $\mathbf{S}_{ji,j}$ refers to the similarity between embedding $\mathbf{e}_{ji}$ and the centroid of the same vocalist. The log term normalises the similarity scores by summing the exponential of each element across all vocalists k. This encourages the model to output similarity scores $\mathbf{S}_{ji,k}$ close to 1 when $k = j$, or 0 if this is not the

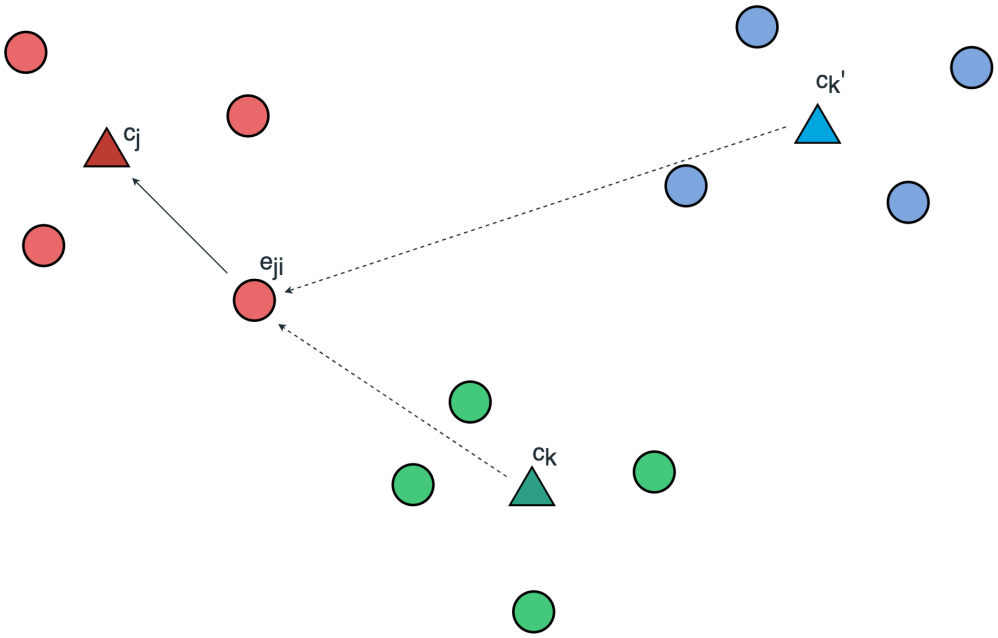


Figure 4.7: An illustration of the effects of the GE2E softmax loss. Circles and triangles represent embeddings and centroids, respectively. Dotted arrows represent the repelling force between unmatched embedding-centroid pairs, while the solid-line arrow represents the attracting force between matched elements.

case. This loss function has the effect of pushing embeddings of the same vocalist towards their vocalist's centroid while simultaneously being repelled by all other vocalist centroids, as seen in Figure 4.7

Finally, the total GE2E loss over the entire similarity matrix, and therefore batch of utterances, is computed by summing the losses for each embedding:

$$L_{GE2E}(\mathbf{x};\mathbf{w}) = \sum_{j,i} L\left(\mathbf{e}_{ji}\right) \tag{4.13}$$

4.4.2 Audio Analysis

The task of voice analysis can be broken down into the tasks of utterance transcription (relating to phonef, speech, and lyrical classification), music information retrieval (relating to timbre, harmonic, or note/voice segmentation), and vocal attribute classification or embedding. This section presents architectures and tech-

niques used to generate voice identity labels or embeddings, both of which have been used to facilitate voice conversion tasks. There are, of course, many ML algorithms that are more than suitable for audio classification tasks if the problems are appropriately matched, such as support vector machines (SVMs), random forests, logistic regression, and a number of clustering algorithms, although these are outside the scope of this literature review.

Label and embedding generation is typically achieved with supervised and unsupervised methods, respectively. McCallum et al. [2022] compared these two approaches when developing general-purpose audio embeddings for downstream MIR tasks. The embeddings generated by both methods yielded state-of-the-art results for music tagging. They found that unsupervised learning benefited from restricting the domain of the data between up and downstream tasks, and resulting embeddings generalised across a wider range of tasks than those of supervised learning. The question of how well such embeddings can generalise to downstream tasks of different domains would be of great interest to the field of voice conversion, where speech datasets far outnumber singing datasets.

Log-magnitude mel-spectrograms have become a common alternative to waveform representations due to their intuitive structure where they present audio as an image, and reduced dimensionality (depending on the required processing steps). As a result, many techniques from the field of computer vision have been successfully implemented when audio is presented as a spectrogram. CNN models are capable of considering the shape, location, and/or textures of sounds found in spectrograms. These can be indicative toward the identification of a sound event or acoustic scene. CNNs have contributed to excellent results for music classification tasks such as audio event detection [Hershey et al., 2017], audio tagging [Choi et al., 2017], pitch detection [Kim et al., 2018], instrument classification [Costa et al., 2017], and lyrics transcription [Demirel et al., 2020].

Derived from the popular VGG model Simonyan and Zisserman [2015] which is a CNN architecture, the VGG*ish* model[8] inspired by Hershey et al. [2017], is a pretrained CNN-based architecture, the pre-classification layer of which can produce fixed-size embeddings from audio files of an arbitrary length. These have

[8] `https://github.com/tensorflow/models/blob/master/research/audioset/vggish/README.md`

been shown to be suitable for various music and audio-based downstream tasks. Clustering algorithms have also been used on mel-spectrograms and VGGish-produced audio embeddings to effectively cluster audio events for downstream tasks [Jansen et al., 2017].

Choi et al. [2017] extensively examined the application of CNNs and RNNs w.r.t. music tagging. It was noted that RNNs do a better job of summarising the temporal information than CNNs, as they operate on a global scale of observation rather than a local one. However, a hybrid model of both called the Convolutional Recurrent Neural Network (CRNN) could sequentially attribute detailed patterns in local contexts, before analysis for temporal dependencies was executed, which provided the best results for this task. In the same work, they also investigated the performances of CNNs, the convolution kernels of which: delayed frequency axis dimensionality reduction; advanced it; or gradually employed it with temporal axis reduction using a 2D kernel of equally sized axes. They found that the latter performed best and, while it did not perform as well as the CRNN, it did have significantly fewer parameters that made it comparable.

The appropriateness of computer vision techniques must, however, be carefully considered in the context of spectrograms. A shape in a spectrogram can mean something significantly different depending on where it lies on the frequency axis. This means that the CNN's property of translational invariance across the frequency axis is not desirable. To avoid this, Blaauw and Bonada [2018] and Luo and Su [2018] have used 1-dimensional CNNs and distributed spectral frequency bins across the channel axis instead. While subsequent chapters explore voice analysis and synthesis, experimentation was necessary between 2D and 1D convolutions to determine its appropriateness for the classification of vocal attributes such as singing technique.

Recent research has enjoyed the luxury of using more modern architectures, such as the Transformer [Vaswani et al., 2017]. Won et al. [2021] presented the Music Tagging Transformer, which outperformed all previous architectures in audio tagging. The model was trained using *noisy* teacher-student training. By adding noise to the student's input data with each student-teacher iteration, it becomes more robust than the teacher, and the two models can swap roles iteratively to continually increase robustness. Knowledge distillation via parameter reduc-

tion between the models was shown to improve the network's performance of semi-supervised audio tagging. Lu et al. [2021]'s SpecTNT model makes use of two transformers that model the temporal and spectral-based features. Transformers have also excelled in beat-tracking [Cheng and Goto, 2023], pitch transcription [Toyama et al., 2023], and MIDI infilling [Malandro, 2023]. However, as Damböck et al. [2022] highlight, the trade-off between slightly better results and computational cost may not always be favourable.

4.4.3 Disentanglement

As initially described in Chapter 1, disentanglement describes the process of separating information that relates to a particular attribute of the data. Examples of this include removing melodic information from an instrument recording, colour from a photograph, or patient-specific information from a medical report. This section discusses methods that have been used to achieve disentanglement in tasks related to audio and voice.

Conditioning

By continually providing a network with labels via an auxiliary input during training, it learns to depend on them to adjust its parameters to model the attribute class the label represents. This is called *conditioning*.

One-hot encodings have been used to condition networks for the purpose of disentanglement [Kameoka et al., 2020, Liu et al., 2021a, Wu et al., 2020, Chou et al., 2018, Van den Oord et al., 2017, Lu et al., 2020, Kameoka et al., 2020]. If there is a finite number of classes to which an attribute can belong, such as those of pitch classes [Luo et al., 2020b], gender, or phonemes [Li et al., 2021b], then the one-hot encoding format is a perfectly suitable representation. However, it does not allow a model to consider a new class of data or variation of a predefined class. The result of this restriction will be presented in Section 4.4.4.

Recent research, has focused more on zero-shot conversions, where systems are able to take unseen examples as both the source and target signals. Such flexibility in conversions can be achieved by replacing one-hot encoding vectors with embedding vectors [Qian et al., 2020a, Lee et al., 2020]. When using descrip-

tive embeddings, conditioning can encourage the encoder of an autoencoder to prioritise encoding information unrelated to the conditioning vector, resulting in disentanglement in the bottleneck. For a set number of classes, the mean values of multiple feature embeddings across each class are often used to represent individual instances of those classes [Qian et al., 2019, Zhou et al., 2021, Li et al., 2021b]. It should be considered however, that generalising over a large amount of class instances would only be useful when the features of such classes are expected to have a narrow standard deviation. Experimentation may be necessary to determine how useful averaging is for SVAC. It seems intuitive for voice identity conversion, but modelling more elusive and dynamic attributes such as singing techniques may require a different strategy.

Tan et al. [2021] discuss how averaged representations force models to rely on a finite number of static embeddings, which helps them to perform better voice identity conversion, but generalises more poorly to unseen target embeddings. They report better conversion performance when using an additional network that takes F0 information, plus live-generated and speaker-averaged embeddings to produce a new *adjusted* embedding. Li et al. [2022b] proposed a U-net to model voice identity embeddings (VIEs) with instance normalisation modules [Ulyanov et al., 2017] after each downsampling block to produce hierarchical speaker embeddings at multiple granularities.

Other conditioning factors used to disentangle timbre include loudness [Liu et al., 2021a, Nercessian, 2020], phonetics (usually from pretrained linguistic networks) [Liu et al., 2021a, Shen et al., 2018, Bonada and Blaauw, 2021, Nercessian, 2020, Sun et al., 2016, Skerry-Ryan et al., 2018, Polyak et al., 2020, Li et al., 2022b, Du et al., 2021, Li et al., 2021b], and pitch-related features either as one-hot encodings or continuous data [Bonada and Blaauw, 2021, Qian et al., 2020a, Nercessian, 2020, Li et al., 2022b, Polyak et al., 2020, Tan et al., 2021, Fang et al., 2018, Lee et al., 2019]. With such deterministically generated features that account for well disentangled attributes, the concatenation of these as conditioning vectors could lead towards the disentanglement of attributes that are rarely labelled or too abstract to be deterministically predicted. This concept is an important motivation behind the work presented in this thesis.

Vector Quantisation

Another method that facilitates attribute disentanglement is 'vector quantisation' (VQ). Van den Oord et al. [2017] combine discrete latent spaces of a VAE model with VQ, which is the process of mapping an input vector to the most similar vector in a lookup table (also called a codebook or dictionary in relevant literature). The encoder's output is mapped to its nearest neighbour in the lookup table. As this process of VQ breaks the differentiable chain, the gradients are simply copied from the decoder to the encoder (*straight through estimation*). An L_2 loss is then used to bring the contents of the lookup table closer to the encoder's outputs. By conditioning the decoder on speaker identity, the network can disentangle linguistic content from its input. However, VQ is restricted to modelling attributes of a discrete nature where interpolation between them is not meaningful. There are some attributes in SVAC where this would be desirable, such as gender. Conversely, it would be pointless for attributes of a more dynamic nature such as vocal emotion.

Transfer Learning

Pan and Yang [2010] describes transfer learning as the transfer of knowledge between models and domains. This is implemented in many forms, such as taking a pretrained network and inferring on a different dataset domain to the one it was trained on [Bittner et al., 2022]; freezing a model's weights and using its output for downstream tasks [Qian et al., 2019]; or using the latent regressor loss regularisation techniques discussed in Section 4.4.1.

Nercessian [2020] pretrained a *singing* voice identity conversion network on a large amount of speech data, before fine-tuning it to learn specific details about the voice in a musical context. They note that this kind of generalisation allows a model to be used as a 'Universal Background Model', which means it can harness any unlabelled voice-recorded data to improve its performance. Wiewel et al. [2020] discuss how catastrophic forgetting (where a network can quickly forget about features that are relevant across the global distributions of the data) can be imposed when a network focuses on a new subset of data. In cases where there is more than one dataset to be used for pretraining, the order in which a model

learns from these datasets may affect its ability to model the final dataset, and is therefore worth empirically investigating.

Auxiliary Classifiers

Attaching a classifier as an auxiliary output to a specific pathway of a main network can facilitate disentanglement. It does this by ensuring that the information that passes through such a pathway contains class-correlated features. Attaching two classifiers to two parallel pathways would encourage the information to disentangle itself so that each pathway contains information specifically filtered to maximise each classifier's performance. Luo et al. [2020b] used two classifiers appended to parallel latent spaces in their VAE model to achieve attribute disentanglement. Kameoka et al. [2020] used an auxiliary classifier in the latent space to ensure their VAE encoded conditioning features as well as input features.

Confusion modules are classifiers that pass an *inverted* loss penalty to the main network's objective function. This encourages the network to avoid passing any information to the confusion module that might help improve its classifications. In other words, the confusion module ensures that class-correlated information is removed from its pathway. Confusion modules have been used to improve information disentanglement by Nachmani and Wolf [2019], Li et al. [2021b], Deng et al. [2020], Mor et al. [2019].

Considering the ability of such auxiliary classifiers to rely on class-correlated features, they could also be connected to the pathway of a NN to ensure that the correct information is being transmitted (or is *not* in the case of confusion modules) as expected between modules. Monitoring the performance of such classifiers would therefore be an effective method for evaluating degrees of disentanglement.

Gaussian Mixture Modelling

Vanilla VAE structures assume $p(\mathbf{z}) = \mathcal{N}(0, I)$. However, the complexity of X can often be oversimplified by operating on the assumption that it can be represented by a unimodal latent space. Multi-modal distributions in latent space are therefore difficult to map to a single latent space. Gaussian Mixture Variational

Autoencoders (GMVAEs) [Dilokthanakul et al., 2017] extend the prior to incorporate a mixture of Gaussians, each of which is linked to a particular aspect of the observed data. This has been used to disentangle the data attributes in a hierarchical manner, and can be facilitated using auxiliary classifiers to channel the relevant information towards each distribution. Examples of its implementation for the conversion of voice attributes can be found by Hsu et al. [2019], Luo et al. [2020b]. This could be a promising mechanism to incorporate in SVAC systems should multi-modal distributions exist that are too difficult to model.

4.4.4 Voice Conversion and Synthesis Systems

This subsection focuses specifically on how NN systems use analysis and disentanglement methods to achieve voice conversion and synthesis tasks.

Although widely used across the vast majority of recent literature, the term 'voice conversion' is ambiguous, as it does not describe which attribute of the voice is being converted. In order to make this clearer, terms that specify the attribute of interest will be used. These include STC, voice identity conversion (VIC), and *singing* VIC (SVIC).

VIC is the process of modifying a speaker's acoustic vocal characteristics so that their voice sounds like it belongs to another. The same definition applies to SVIC, replacing 'speaker' with 'singer'. However, consideration must be given to the differences between speech and singing, as described in Section 4.1.3.

Considerations for Voice Conversion Tasks

Different models have different conversion capabilities, so it is often best to consider this before deciding which model or architecture to use. The different types of conversion include the following:

- one-to-one: Converting from a single class to only one other class
- one-to-many: Converting from a single class to a finite number of classes (seen during training)
- one-to-any: Converting from a single class to any infinite number of classes (classes not seen during training)

- other permutations of the previous types, such as any-to-one, many-to-any or any-to-any etc.
- interpolated conversions between target voices
- zero/one/few-shot conversion: Signifying the number of examples needed before a network can convert the vocal attribute effectively.

When attempting VIC, it is also worth considering whether changing the pitch range will improve the perception of naturalness and similarity of the converted output audio, especially if the conversion is between genders. Qian et al. [2020a] found that in their previous work [Qian et al., 2019], cross-gender conversions led to inconsistencies in the F0 contours, which fluctuated unnaturally between pitch ranges exclusive to either gender. This was solved by standardising the F0 contours and remapping them to the statistical values of the target speaker's F0 contours during inference. For SVIC, Nercessian [2020] converted the features of the spectral envelope, completely removing the information of F0 from the conversion process. After conversion, the F0 contours were shifted to the octave that best matched the range of the target singer and recombined with the other voice attributes for resynthesis. However, such imposed octave shifts are naive to the capabilities of the target singer, and do not consider how such a pitch range would affect the timbre of the singer. This issue is more prominent in SVIC than VIC, as pitch ranges standard deviations will generally be much larger for singers than speakers. This entanglement between pitch and timbre in singing should be addressed before imposing pitch-shifting.

To achieve vocal attribute conversion, a typical system consists of an encoder that produces linguistic embeddings. Training an encoder to disentangle this from an audio-based input can be achieved by conditioning it on attribute-specific labels or embeddings from an auxiliary source, such as a pretrained/DSP-based encoder or manual annotation. A decoder then combines these embeddings to resynthesise the data in its voice-converted state. This system has been fitted to autoencoders, VAEs, GANs, and other hybrid architectures, all of which are discussed in the following subsections.

Autoencoder Systems

Qian et al. [2019] proposed the conditioned autoencoder, AutoVC. This uses VIEs as the conditioning factor for an autoencoder's decoder, which originates from the same speaker that produced the encoder's input features. The bottleneck's dimensionality is calibrated manually so that the encoder can only store limited information about its input data. As the decoder is conditioned on VIEs, the encoder can learn to prioritise encoding information unrelated to voice identity during training, producing a disentangled representation of *primarily* linguistic content. After training, the input features and conditioning VIEs can be generated from different speakers, facilitating VIC. Jia et al. [2018] use the same concept for TTS, where there is no need for bottleneck calibration since the input to the network contains text, requiring no disentanglement from any voice attributes. The voice identity information is provided by the embedding layer of an auxiliary encoder that was pretrained to classify voice identity.

AutoVC's bottleneck, however, contained leaked information relating to pitch contours. This was addressed in [Qian et al., 2020a], where the issue was solved by conditioning the decoder on extracted pitch contours as well. Qian et al. [2020b] continued to add more conditioning terms to allow the network flexibility in converting timbre, pitch, and rhythm.

While AutoVC provides an attractively simplistic solution to VIC, the manual calibration of the bottleneck is a poorly constructed solution that can only be determined after evaluating a previous calibration's conversion capabilities. This is especially time-consuming if relying on human evaluation and would benefit from an automatic evaluation system and mechanism that is robust against information that is leaked through the bottleneck.

Nercessian [2020] adapted AutoVC for the task of SVIC, by conditioning the network on pitch contours and transposing them into a suitable register for converted singing (omitting concerns about pitch shifting as addressed in Section 4.4.4), achievable through the implementation of the WORLD vocoder [Morise et al., 2016]. They did not however, use a singer-pretrained encoder, leaving the influence of this domain mismatch unexplored.

Li et al. [2021b] combine the work of Shen et al. [2018], Wang et al. [2017],

and suggest that the approaches of Hsu et al. [2017], Luo et al. [2019] allow redundant noise in datasets to worsen the generated audio quality. Their network takes an audio input and disentangles it into F0 contour, VIE, and linguistic embeddings. An additional pathway is fed to an encoder that has the task of extracting residual information not captured by any of the first three disentangling paths. This is called the *mel encoder*. Its input is a mel-spectrogram and its outputs are first regulated by a voice identity confusion module. The outputs are then concatenated with the VIE and fed to a *mel regression* module, which is trained to rebuild the mel-spectrogram. The effect of this is that the mel encoder's outputs should equate to musical and acoustic information, such as intonation and emotional features. This pathway was shown to improve the similarity and naturalness of converted voices. This technique of using a confusion module to reduce the amount of noise in pathways resulting from extensive disentanglement is a promising mechanism that could be an essential tool when designing disentanglement systems to expose unlabelled attribute information.

There is often a mirrored architecture between the encoder and decoder of an autoencoder architecture. However, an additional module of CNN layers (sometimes called a spectrogram refinement module) appended to the decoder has been shown to enhance the quality of spectral output representations [Lee et al., 2020, Qian et al., 2019, Shen et al., 2018, Luo et al., 2020b].

VAE Systems

Kameoka et al. [2020] use a VAE to perform VIC, and use auxiliary classifier to tackle *mode collapse* in the model. Mode collapse occurs when lower-dimensional latent spaces z tend to provide weaker or noisier signals to a strong decoder, causing the decoder to generate mean representations of the data. VAEs typically consist of a *continuous* latent space, the nature of which contributes to this problem, making it more complex. However, using a *discrete* latent space where only a finite set of configurations relate to a given number of classes, decreases the likelihood of mode collapse. As discussed in Section 4.4.3, Van den Oord et al. [2017] introduced VQ to the VAE architecture for the task of VIC, and conditioned the decoder with one-hot voice identity vectors to extract VIEs from the a

learned lookup table. After training, VIC can be achieved by providing a target voice's VIE to the decoder while the source voice's VIE is given to the encoder's input. However, shortcomings of the VQ-VAE include its limitation of being capable of only many-to-many VIC and its inability to interpolate between classes. It is, however, easier to train because of the inherently less amount of variance required for modelling. As previously mentioned, this trade-off may be favourable for voice attributes that are not typically dynamic, such as gender or potentially the use of the vocal fry singing technique.

Hsu et al. [2019] use a GMVAE architecture, using Tacotron2 [Shen et al., 2018] as the encoder base. They model two separate latent spaces using GMMs - one for labelled and one for unlabelled attributes, incorporating two levels of hierarchical latent variables, the first level being categorical, and the second level being a continuous multivariate Gaussian variable. The GMVAE allowed them to infer VIEs from noisy utterances of an unseen speaker and generate clean speech that approximates the voice of that speaker. The unseen attributes can be considered as the residual variances apparent after removing conditioning labelled attributes, which include noise level and speaker rate. Fine-grained control over these attributes in the bottleneck was also possible, thanks to the hierarchical breakdown of the input data's components. The application of this research to singing, where there may be more scope for expressivity, would be an exciting extension of this research. Luo et al. [2020b] was able to achieve nonparallel, many-to-many VIC and STC using a GMVAE, but the VAE's ability to manipulate residual variances representing unlabelled attributes has yet to be applied to singing. This domain may yield more dimensions for expression than in speech, which could lead to more refined methods of singing voice manipulation.

Wu and Lee [2020] found that their use of VQ and instance normalisation with an autoencoder allowed them to extract VIEs by subtracting post-VQ vectors from pre-VQ vectors. VIC was achieved by using two trained encoders, one for the source voice and one for the target voice. Their subsequent work [Wu et al., 2020] restructured the VQ implementation in a U-net architecture, which was reported to produce superior results to its predecessor. They also observed that the more quantisation they apply, the better the reconstruction, but with less disentanglement. This trade-off between audio quality and target-attribute similarity is

commonly witnessed in VIC networks.

Other Architectures

Most of the GAN implementations explored in relation to voice synthesis are devised for singing voice synthesis (SVS) or TTS, and so will be described in next subsection. This subsection covers systems that utilise the autoencoder architecture, and enhance its synthesis by appending a GAN to its output for high-fidelity capabilities.

AlBadawy and Lyu [2020] use a VAE as the generator in a GAN system, equivalent to a VAE-GAN, for VIC. They use one encoder to extract linguistic content and a separate decoder is trained for every target singer. They use the VAE's reconstruction loss when the source and target speaker are the same, and the GAN loss when evaluating how realistic a converted voice is. Cycle-consistency and latent regressor losses are also used in this system. This system however, seems inefficient, as the use of multiple decoders for each target speaker is excessive, time consuming to create, and technically restricts each end-to-end network to any-to-one conversions.

Lu et al. [2020] use a similar setup to Qian et al. [2019] with no mention of bottleneck calibration, and uses one-hot identity and F0 vectors for conditioning. As they tackle SVIC, they also condition this on F0. However, voices converted between genders consistently lack the correct target F0 and sometimes intonation. Like AlBadawy and Lyu [2020], the appendage of a discriminator, with a Wasserstein loss that makes it a VA*W*-GAN, allows the autoencoder to further improve its output, although there is no mechanism to guarantee that the F0 has been appropriately transposed to the target singer's range.

Zhou et al. [2021] use the same VAW-GAN architectural setup as in [Hsu et al., 2017], and conduct voice *emotion* conversion. They use descriptive emotional embeddings instead of one-shot vectors, allowing it to work with unseen voices. The results, however, sound more like VIC than emotion conversion, or at least seems to rely heavily on the conversion of pitch range and timbre to convert emotion. Researchers should be cautious when asking participants to compare the audio of a converted voice to a reference voice based on emotional similarity, as

this is an unintuitive task that could lead towards choices heavily influenced by confounding variables in the stimuli.

Caillon and Esling [2021] proposed the Realtime Audio Variational autoEncoder (RAVE). Due to its specific subnetwork architectures, it is specifically designed to take audio waveforms as input. The RAVE architecture involves a standard VAE-style training strategy as a first stage. After acceptable reconstructions are attained, an adversarial network is applied to the VAE's output. The VAE's decoder and discriminator are then trained to achieve a finely tuned generator network, mitigating the effect of issues highlighted by Kameoka et al. [2020] about mode collapse and oversmoothed VAE output.

DiffSVC [Liu et al., 2021b] is a diffusion model that takes a mel-spectrogram as its input, and uses auxiliary input channels for linguistic, F0 and loudness features to control the attributes of the synthesised voice. The model is trained using a dataset of one singer. After being sufficiently trained, DiffSVC has encoded the timbre of the singer on which it was trained and uses the auxiliary inputs to control the content. However, features not accounted that would also be encoded include attributes such as emotion, accent, singing technique, and other confounding variables that may affect realistic SVIC. One particular draw back about this type of model is its restricted capability of exclusively performing any-to-one conversions.

TTS and SVS Systems

This section covers ML systems that take input representations such as text/lyrics and pitch (usually leaving timbre as the remaining attribute to be encoded by the model itself) and produce spectral envelopes, spectrograms, or waveforms - the latter of which often require separately trained NN modules, which are covered in Section 4.4.5. While these models address a problem outside the scope of voice conversion, they do use analysis, disentanglement, conditioning, and audio synthesis techniques that are commonplace in voice conversion networks.

Bonada et al. [2016] used a unit selection model with two databases to produce expressive singing. The first of these consisted only of expressive vowels to assist their system in generating expressive songs. This database featured a singer

performing various melodies with different vowels for each consecutive note in an expressive manner (similar to the data produced in [Scherer et al., 2017]). A second database for timbre consists of phonemes being spoken in monotonic fashion, progressing at one syllable per beat. Using these datasets, expressive vowel performance is first generated and used as input control to the proceeding synthesis step which uses the monotonic phoneme dataset. The Viterbi algorithm is used to apply unit selection-based costs to a string of samples from the respective databases for concatenation.

Blaauw and Bonada [2018] proposed the Neural Parametric Singing Synthesiser (NPSS) - an SVS model that produces spectral envelopes of singing. The model is trained on pitch, lyrics, and phonetic timing data, which are also used for control during the synthesis phase. WORLD features originating from one singer's recordings were used as input features to the model. The SVS system consists of a phonetic timing model, a pitch model, and a timbre model, each of which is conditioned by the previous model. The latter is the main module of interest, which houses a WaveNet architecture. It takes WORLD features as input during training and uses F0 and linguistic features during inference. NPSS is large and complex in how information flows through it, but demonstrates the amount of engineering required to produce realistic versions of singer recordings from a parametric NN. NPSS was later improved upon by Zhang et al. [2021], where a new architecture involving a Wasserstein GAN was 219 times faster in computational speed, while maintaining a similar performance.

Chandna et al. [2019] implement a GAN for SVS, using an autoencoder for the generator. In addition to the GAN's Wasserstein loss, a reconstruction loss is also used during training. The autoencoder has skip connections between each layer of the encoder and decoder, similar to the U-net architecture [Stoller et al., 2018, Ronneberger et al., 2015]. It is conditioned by linguistic and VIE information in the form of one-hot vectors, and *continuous* F0 data. It is also conditioned on WORLD vocoder features. The network does not outperform NPSS, but is comparable. However, when the conditional features stem from a singer that is of a different gender to that of the VIE input, many phonemes are poorly articulated. One way to improve upon this would be to include an auxiliary phonetic classifier in the networks objective function.

Lee et al. [2020] proposed a GAN, where the generator consisted of pitch and formant mask encoders conditioned on VIEs, the outputs of which were multiplied together to get a mel-spectrogram. A spectrogram refinement module up-scales this to an uncompressed linear spectrogram, as would be derived by taking the STFT of the original audio. The GAN was trained using an objective function containing the following components: reconstruction loss of the mel-spectrogram, reconstruction loss of the linear spectrogram, and discriminator loss. This is one of the few examples that has directly dealt with the fact that both pronunciation and tuning precision are conditional on the singer's style and, by extension, their identity.

Bonada and Blaauw [2021] trained an autoencoder on multiple source singers and a single target singer to perform TTS using semi-supervised training. They trained two encoders: the first, E_A, trained on spectrograms, while the other, E_L, was trained on phonetic encodings. Neither input's durations are more than several hundred milliseconds. To ensure the encoders produce similar embeddings, a stochastic mechanism randomly swaps the pathways between either encoder's output and the bottleneck (L_1 loss between both outputs is used for training). The bottleneck encodings are concatenated with F0 contours and VIEs, and then sent to a decoder architecture to produce a spectrogram output, from which a reconstruction loss is calculated. After training the autoencoder, the E_A output is used to train a new instance of the decoder that is specific to a target voice. After this phase of training, E_L can then be used in conjunction with the newly trained target-singer-specific decoder to synthesise spectrograms from phonetic and F0 content. Like AlBadawy and Lyu [2020], this solution to semi-supervised TTS requires a separately trained encoder for every target speaker, facilitating only any-to-one conversions.

DSP techniques have in recent years entered the world of NNs, popularised by the introduction of the *differentiable* DSP (DDSP) autoencoder [Engel et al., 2019]. This model can be trained with a simple reconstruction loss, bypassing the challenges of adversarial training or autoregressive predictions, which frequently come with the territory of audio synthesis. From this, the network can learn the parameters for the DSP components that determine the output audio waveform. Since the DDSP parameters can be used in a modular fashion, the DDSP autoen-

coder can take component configurations from one generation, superimpose them on another, remove them entirely, or achieve timbral transfer by using a model trained on one domain to infer from another.

This has paved the way for voice synthesisers such as SawSing [Wu et al., 2022], which takes inspiration from DDSP's modular nature, using NNs to generate filtering coefficients for its DSP components: a harmonic subtractive synthesiser which filters a sawtooth waveform (whose acoustic properties enforce phase continuity between partials); and a noise subtractive synthesiser which filters uniform noise. The synthesised results are measured to be better than or comparable to its competitors, with audibly fewer artefacts. Nercessian [2021] replace the decoder of their previous architecture [Nercessian, 2020] with a Vocoder module that contains convolution and dense layers connecting to a harmonic oscillator and noise filter DDSP component. DDSP has been shown to perform approximately 20% worse than WaveRNN [Kalchbrenner et al., 2018]. As a trade-off, it offers a 40% reduction in its number of parameters, which may be suitable depending on the demands of the user.

DiffSinger [Liu et al., 2022] is an SVS diffusion model that converts noise into mel-spectrograms, while taking music score information as conditioning factors. It is able to overcome the over-smoothing outputs and fragile training equilibrium inherent in other generative systems such as GANs, but the nature of its architecture means that it can only utilise the voice of a single singer. *DiffVoice* [Liu et al., 2023] combines elements from a VAE, GAN, and diffusion model to produce a TTS system. The authors encode training data to a VAE's latent space (improved upon by adversarial training) and model the temporal length and latent variables with the diffusion architecture. The additional use of a VIE network as a conditioning factor allows DiffVoice to achieve one-shot voice conversions that produce consistently high-fidelity audio. However, its sampling speed is considerably slow during inference.

4.4.5 Audio Synthesis

Before the field of audio synthesis was saturated with NN applications, there were a number of alternative synthesis methods that converted acoustic features into

waveform audio. In the field of voice synthesis, this included concatenative synthesis, Linear Predictive Coding (LPC), and vocoders. The following subsection describes several seminal voice synthesis methods that have been continually used by researchers to synthesise waveforms from acoustic features - an indispensable step used for illustrative purposes by most researchers of voice conversion that do not use end-to-end models.

Digital Signal Processing

The Griffin-Lim algorithm [Griffin and Lim, 1983] is a deterministic method of estimating waveforms based on magnitude spectrograms where there is no phase information retained. The generated audio usually possesses many artefacts, since half of the data (the phase) of the original waveform is missing in a magnitude-only spectrogram. Kumar et al. [2019] had reported in their experiments, that when evaluated in listening tests, audio synthesised by the Griffin-Lim algorithm achieved a MOS of 1.72, while the corresponding original recordings achieved an MOS of 4.19. Considering the number of higher-quality audio synthesis options available, Griffin-Lim is best left for situations where one needs a quick waveform conversion for demonstrative purposes, or does not have access to an ML framework.

GANs

The most common type of architecture used for audio synthesis has been the GAN. *MelGAN* [Kumar et al., 2019] was proposed 3 years after WaveNet (described in Section 3.2.1), and while it did not perform as well for subjective perceptual evaluations, it was comparable with WaveNet, with one sixth the amount of parameters and 100s to 1000s of times the speed on CPU and GPU, respectively. MelGAN does not take noise as input to its generator, which is fully convolutional, but a mel-spectrogram instead. It uses a multiscale discriminator (MSD) architecture, which downsamples the audio for different rates of analysis. It is non-autoregressive, and could replace any autoregressive model that synthesises waveforms.

Parallel WaveGAN [Yamamoto et al., 2020] (PWG) significantly outperformed

WaveNet in listening tests, achieving an MOS of 4.16 over WaveNet's MOS of 3.33. PWG also trains 2.5 times faster than WaveNet, with nearly a third of the amount of parameters. PWG's generator uses a WaveNet, using *non*-causal convolutions, noise as input during training, is non-autoregressive at both training and inference time, and uses multi-resolution STFT losses as well as adversarial loss.

Shortly after, Kong et al. [2020a] proposed *HiFi-GAN*. Its generator is similar to MelGAN's, and uses *multi-receptive field fusion* to analyse its output for patterns at multiple scales. Three sub-discriminators operate on different chunks of the audio signal. HiFiGAN also uses the MSD architecture from Kumar et al. [2019] to consecutively analyse the audio. It outperformed all publicly available models at the time (but was not compared to PWG), scoring an MOS of 4.36 while MelGAN scored 3.79.

Diffusion Models

DiffWave [Kong et al., 2020b] is a diffusion model that can produce high-fidelity audio while being both unconditioned or conditioned, on either mel-spectrograms or one-hot VIEs. For each of these tasks, its results are significantly better or comparable to its competitors. Takahashi et al. [2023] found the *PriorGrad* diffusion model [Lee et al., 2022], to be suitable for speech but not for singing because of the wider range of expression across multiple dimensions. To better model the singing voice, they used three diffusion layers, each of which operated at different sampling rates. The higher sampling rates are conditioned on the output of the lower sampling models. This provides a hierarchical model that produced SOTA synthesis results, outperforming PWG and PriorGrad without adding computational cost.

Chapter 5

Perceptual Spaces for the Singing Voice

5.1 Introduction

The subsequent sections provide a comprehensive account of the experiment detailed in the author's publication [O'Connor et al., 2020]. However, since this publication, considerable revisions have been made to the data analysis section in order to make the research more understandable, justify the choice of statistical tests, and exchange less relevant results for more relevant and interesting ones. The experiment entailed the collection of data through listening studies, wherein participants were exposed to a repertoire of vocalisations and tasked with rating their dissimilarities. The dissimilarity ratings were then subjected to clustering and statistical analysis to ascertain the number of ways in which participants perceive vocalisations under different gender and pitch register conditions. Furthermore, the study sought to evaluate the alignment between participants' perceptions and the designated ground truth labels for different singing techniques. The data then underwent dimensionality reduction to yield singular points representing each technique, projected onto a two-dimensional plane to facilitate a visual understanding of participants' perceptual spaces. This chapter culminates in a discussion encompassing the notable findings concerning the perception of singing techniques and potential avenues for further improvement in research methodologies.

This research aims to fill a significant gap in the existing literature pertaining to timbral perception of singing voices. Limited research has delved into the timbral spaces of human voices, particularly in the context of singing techniques. To address this gap, the present study endeavours to visually represent and quantitatively assess the perceptual dissimilarities among various singing techniques. A critical examination of the validity of the prevailing terminology for categorising singing techniques is needed, as the extent to which distinct techniques possess perceptual salience warranting their classification as separate entities is not well documented. The existing ontology for singing techniques however, is justified in that it appropriately categorises vocal utterances in a singing context based on the mechanical processes of the relevant vocal organs [Sundberg, 1977, Blomgren et al., 1998]. It has also been noted that terminologies have been convoluted over time by diverging interests between artistic and scientific communities [García-

López and Gavilán Bouzas, 2010, Gerratt and Kreiman, 2001].

This investigation also explores the question of whether listeners coming from diverse musical and audio backgrounds perceive these techniques in divergent ways. Previous work [McAdams et al., 1995, Serafini, 1993] reported significant differences between participants of musical backgrounds, while Carterette and Miller [1974], Wedin and Goude [1972] reported none. Kreiman et al. [1993] and Labuschagne and Ciocca [2016] have highlighted the significant amount of variance present across listening studies, and Proutskova [2019] has shown that even experts in vocal physiology can be in considerable disagreement about vocal mechanics or technique classification when listening to vocal recordings.

Finally, a dataset of perceptual dissimilarity ratings would also prove useful to future research in generative networks for the singing voice. Esling et al. [2018] have shown that utilising timbral spaces as a regularisation component in VAE training has led to latent spaces that are more interpretive, offer more control, and enhance the quality of synthesised output. It was anticipated that the perceptual ratings collected in this experiment could be used in a similar way if the results were satisfactory.

5.2 Method

The aim of this experiment is to explore and present the structure of perceptual space for a given number of singing techniques. Volunteering participants were asked to listen to every pair of audio clips in a given set of stimuli and rate how dissimilar they sound from one another. The following five singing techniques as labelled in the VocalSet dataset [Wilkins et al., 2018] were used: *vocal fry*, *breathiness*, *neutral*, *belt*, and *vibrato*. These were cherry-picked from VocalSet's selection of 17 singing techniques, as they are the most recognisable and commonly used vocalisation types in Western popular music [Kayes, 2015].

Once a sufficient number of participants had submitted dissimilarity ratings, the data was then subjected to a combination of clustering and statistical analysis. Multidimensional scaling (MDS) was used to visualise how these techniques relate to one another in perceptual space on a two-dimensional plane. The proceeding subsections discuss the stimuli, listening study and analysis methods in greater detail.

5.2.1 Stimuli

Experimental Requirements

The term 'listening session' is used here to describe an online listening study session which presents audio clips for participants to rate. Each session is randomly assigned to a participant and contains recordings of a single singer. The total number of required ratings per listening session is determined by m, the number of stimuli available for pairwise distance evaluation. The calculation for the total number of independent pairwise distances C, is:

$$C = \frac{m \cdot (m-1)}{2}. \tag{5.1}$$

As we are including self-similarity comparisons for evaluation in this experiment, we add the value of m to this formula, giving us the final equation as shown in Eq. 5.2 [Cameron, 1994, McAdams et al., 1995]. As each singing technique should be featured evenly in these listening sessions, the number of examples per

technique n, would be the value of m divided by the number of classes, which is 5.

$$C = \frac{m \cdot (m+1)}{2} \tag{5.2}$$

While a large value for the total number of dissimilarity ratings C would provide more robust statistical measurements, a trade-off between C and how much listening fatigue would be experienced by participants was considered. It was empirically calculated with prior pilot studies, that it would take approximately one minute to produce five pairwise distance ratings. 25 minutes of listening was considered a reasonable amount of time to expect participants to remain alert and attentive to detail. Considering this amount of time, the average speed of participants, and the anticipated interest from potential participants, three examples per class was deemed an appropriate choice. This leads to a total of 15 (5×3) unique stimuli and 120 ratings per listening session. These decisions were influenced by similar past work on listening studies [McAdams et al., 1995, Grey, 1977].

However, due to an unfortunate error in the design of the listening study, dissimilarity ratings involving 1 of the 15 stimuli was not recorded. This reduced the total stimuli to a size of 14 examples and 105 pairwise distance ratings per listening session. An example of a dissimilarity matrix generated from a participant's ratings is shown in Figure 5.1.

Figure 5.2 shows the hierarchical structure of the stimuli for each listening session. Three male and three female singers were randomly selected from VocalSet. Each listening session contains two versions of a singer's vocalisations: high or low-register singing. Each of these sessions contains 3 examples of 5 singing techniques. There are twelve sessions in total that participants could be assigned to.

VocalSet

At the time of this study, there were several datasets available to the public that contain annotations for different singing techniques, such as the Phonation Modes Dataset [Proutskova et al., 2013], The Singing Voice Dataset [Black et al., 2014],

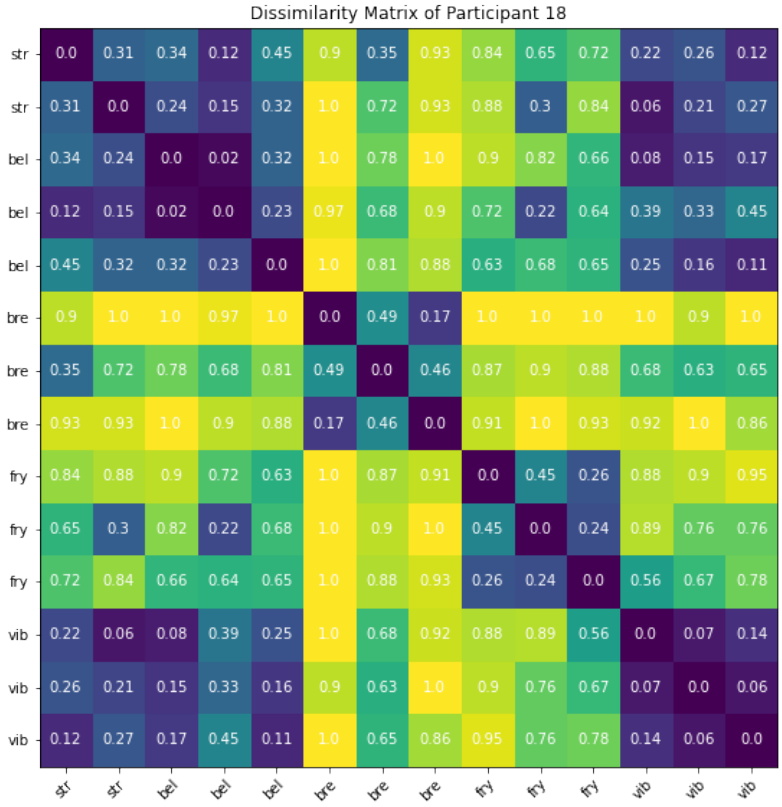


Figure 5.1: Example of a dissimilarity matrix generated from participants' pairwise ratings. Shortened versions of the labels *straight, belt, breathy, fry* and *vibrato* are featured along the axes. Values closer to 1 indicate very strong dissimilarity, while values closer to 0 indicate strong similarity. In this case, data relating to one of three *straight* singing audio clips is missing.

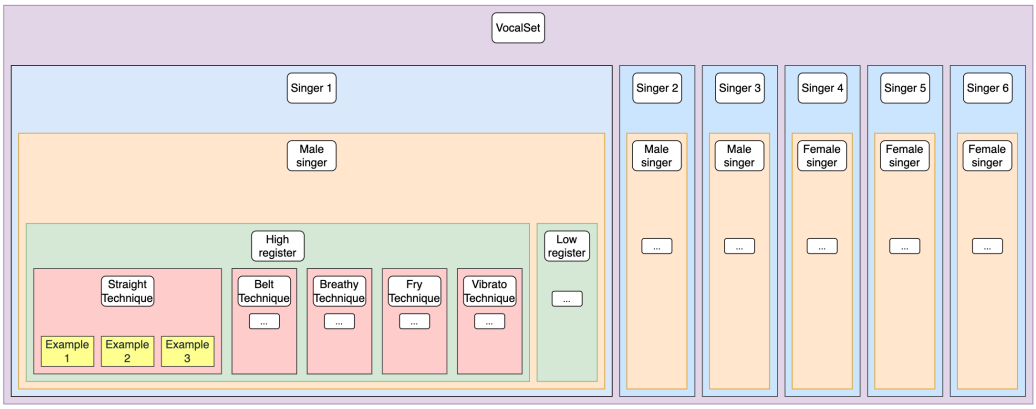


Figure 5.2: Hierarchical diagram of sampled stimuli. Colours represent different levels of conditional groups, while ellipses signify that the contents of its condition block imitate the contents of the block representing the same hierarchical condition group on the far left.

VocalSet [Wilkins et al., 2018] and the Vocobox dataset[1]. Of these candidates, VocalSet was chosen as the source of stimuli, as it contains 20 singers of diverse ranges and ages, 10.1 hours of recordings, claims to be unbiased toward any genre, contains strong annotations and focuses on a range of vocal techniques. Wilkins et al. [2018] have also expressed that their dataset was designed especially for vocal technique classification and style transfer tasks. The singers perform arpeggios, scales, sustained tones, and short musical excerpts using different vowels and vocal techniques.

From VocalSet, only audio clips of the five aforementioned singing techniques were considered. Audio clips were randomly selected from this subset, restricted by the required conditions for each listening session. This subset initially contained an unbalanced number of classes, so examples were randomly deleted from the larger class sets until the subset was class-balanced. Recordings labelled as *excerpts* were also omitted, as these were disproportionately long compared to all other examples, and only existed for the *vibrato* class.

[1] `www.github.com/vocobox/human-voice-dataset`

5.2.2 Sampling from VocalSet

As 15 examples satisfying singer, gender, register and technique conditions had to be selected from the VocalSet subset, a random sampling process with imposed restrictions was required. One second of audio per example was considered suitable for this study, which allows time-varying features in the voice to become evident to participants, reflecting considerations in previous works such as that of McAdams et al. [1995]. The next section describes how audio clips were randomly selected for auditioning before being added to the listening sessions' stimuli pool.

Register-Matching

A register-matching algorithm was applied in order to determine whether an audio clip would satisfy the singing register condition (low or high) of a given listening session. To do this, reference pitches that are close to the lower and upper-bound limitations of a singer's range (to maximise the diversity between low and high registers without unduly restraining the amount of potential candidate content) must be calculated. They are therefore determined by analysing the pitches of each singer's entire repertoire prior to the random sampling process.

To do this, a series of continuous pitch values were generated for all audio files in the VocalSet subset using Sonic Annotator[2] [Cannam et al., 2010] in conjunction with the pYIN algorithm [Mauch and Dixon, 2014] plugin, which possesses specified parameters to mimic the output of the annotation software, Tony[3]. A median filter window of 77 frames (approximately 0.45s in length) was empirically chosen to adequately smooth out the resulting pitch contours.

Next, global means and standard deviations were computed across all recordings for each singer. Any sections of pitch contours that contained zero values were excluded from the statistical calculations. These zero values indicate unvoiced, excessively noisy, or silent segments that do not provide meaningful pitch information. Subsequently, the pitch references for the low and high registers

[2] `https://code.soundsoftware.ac.uk/projects/sonic-annotator`

[3] `https://code.soundsoftware.ac.uk/projects/tony/wiki/PYIN_Parameters`

were determined by subtracting or adding one standard deviation to the global mean for each singer.

During the random sampling process, the register-matching algorithm computes a mean pitch value from the sampled candidate audio clip in a similar fashion. It then compares this local mean value to reference pitches for that singer's low/high register. However, if more than 25% of the pitch contour contains consecutive zero values, the audio clip is deemed unsuitable and the random sampling process is restarted.

It was hypothesised that the task of rating dissimilarity w.r.t. technique would be too difficult if vocalisations possessed a wide range of pitches within a listening session. Conversely, restricting pitches to be precisely the same had proven to be too restrictive for the sampling process to select a sufficiently random set of stimuli satisfying all conditions simultaneously. Therefore, a tolerance of two semitones in pitch variance was applied, as it allows for an adequately diverse representation of vocalisations without applying significantly diverse changes in timbre. If a sampled audio clip yielded a mean pitch value within two semitones of the relevant register's reference pitch, it passed this stage of the sampling process.

The pitch contours generated for many instances of *vocal fry* techniques, however, were frequently very poor. These vocalisations often featured a significant amount of aperiodicity, where the fundamental frequency is difficult to determine, if it indeed exists at all. The technique can also produce sub-harmonic frequencies a number of octaves below all other singer vocalisations, often making it impossible to obtain a meaningful local mean pitch value that is close to a central pitch. If sampled audio clips cannot satisfy the listening session's register condition requirements after 20 iterations of randomly sampling from a given singer, then the pitch-matching requirement is bypassed.

Stimuli Post-Processing

There was a large amount of variance in perceived volumes between singers and techniques. Extracted audio clips were therefore normalised to make the comparative task easier for participants, which is a common step in preparing stimuli for listening studies [McAdams et al., 1992, 1995]. They were also converted to

128kps mp3 format using *Pydub*[4], which was assessed to be a suitable quality for the task.

5.2.3 Listening Study Setup

As this research was conducted in late Spring of 2020, COVID restrictions made it impossible for listening studies to be conducted in person. In response to this, an online version was created that could instead be done remotely. The Web Audio Evaluation Toolkit (WAET) [Jillings et al., 2015] was used to design the listening study interface. This was chosen over other listening study interface designs due to its flexibility in interface development and currently active support from C4DM alumni creators. The first web pages of the listening study covered a written introduction, ethical documentation, and instructions to participants on how to interact with the interface. The WAET API collected relevant personal information from the participants relating to their listening environment, musical training, perceptual abilities, and individual profile data such as age and gender (although neither of these were used in the analysis).

The interface consisted of 20 web pages. Each page presented one reference audio clip and up to 8 comparative audio clips, each of which was accompanied by its own slider as seen in Figure 5.3. Participants were instructed to move the sliders to a position that represented how dissimilar each comparative audio clip was to the reference audio. From the lowest to the highest value, the sliders were annotated with the following evenly spaced text prompts: 'Extremely different', 'different', 'similar' and 'extremely similar'. These prompts were placed to assist participants by indicating the sliders' polarity, encouraging consistency, and implying some linearity across the sliding space.

5.2.4 Participants

A call for participants was sent out via mailing lists within QMUL and among the international MIR-related community of ISMIR, to which 61 volunteers responded. The minimum requirement for the study was to have an interest in mu-

[4] `https://github.com/jiaaro/pydub`

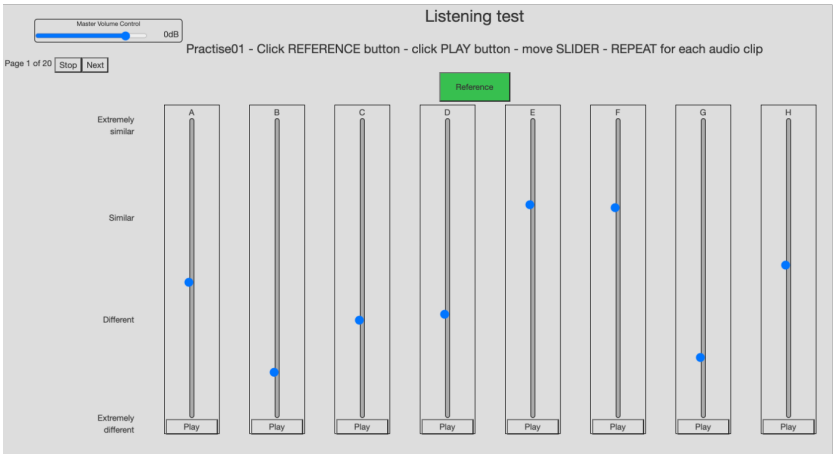


Figure 5.3: View of interface used by participants for rating dissimilarities between a single reference recording and multiple comparative recordings.

sic, which allowed for a wide range of participants from different levels of musical background to contribute. Previous literature (see Section 4.1.2) on general timbral mapping has reported diverse findings regarding the differences between participants of various musical backgrounds. How this affects the perception of vocal techniques remains to be determined.

Participant Questionnaire and Instructions

The following list presents a set of questions given to participants regarding their age, gender, listening environment and musical abilities [McAdams et al., 1995, 1992]. Square brackets after each question containing a list indicate that the question is multiple choice. Questions 1-5 were asked before the practice round. Question 6 was asked after the practice round, and Question 7 was asked at the end of the study:

1. Please indicate what listening equipment you intend to use for this experiment (Headphones are preferable). If you wish to change your setup, please do so before continuing and refresh this page [Inbuilt speakers, external speaker, ear/headphones]

2. How would you assess your current listening environment on a scale of 1 (very noisy) to 5 (very quiet)?

3. Please provide your age in the space below.

4. Please provide your gender identity in the space below.

5. What instrument are you best at playing?

6. Having just done the practice round, do you think you have any hearing impairments that would affect your ability to perform similar tasks? [Yes, No]

7. Do you have any other comments regarding your evaluations or any other aspect of the study?

It also contained a set of questions from the 'Perceptual Ability' subsection of the GOLD-MSI questionnaire [Müllensiefen et al., 2014], which were developed to provide information regarding the participants' musically relevant perceptive skills. These are presented in Appendix A. As a precautionary start to the study, participants were first asked to set their volume to the minimum value, and then click 'play' for a test audio clip that featured a speaker. Participants were asked to fade up the volume until the speaker roughly sounded as if they were one meter away, which ensured an audible and safe listening volume for the rest of the study. They were then given instructions regarding the main task, which involved listening to pairs of vocalisations and rating their dissimilarities. The description of how they should conceptualise and rate dissimilarity is quoted below:

> We are interested in measuring how differently listeners perceive the sounds of a singer's voice when utilising various singing techniques. In this experiment, you will be comparing multiple unedited and unprocessed recordings of one individual singer. Your task is to rate how similar or different the singer's sustained vocalisations sound to you, due to different singing techniques. The challenge, therefore, is to rate vocal similarities IRRESPECTIVE of the singer's changes in pitch (notes) and utterance (vowels) between recordings.

At the end of the experiment, participants were invited to give open feedback regarding their experience of the experiment and the techniques they used for rating dissimilarity.

Data Reliability Management

Before participants began the recorded section of the study, they were first introduced to the format of the experiment via 3 'practice rounds' where their responses to the given task were not recorded. This allowed them to become familiar with the interface and format, with the added benefit of exposing them to a wide range of vocalisations. To counteract any bias the order of the tasks might cause, they were presented in a randomised order, and vocalisations from the practice rounds came from singers who did not feature in the recorded section of the same listening study. To minimise the participants' fatigue, they were prompted to take short breaks twice during the study (another frequently used technique in listening studies).

Pilot Study Feedback

Participant feedback during pilot study sessions led to several changes in the publicised version which are as follows: the practice rounds were increased from one to two pages; the task description was refined to inform participants that perceptual differences in audio production, audio quality, and the singer's musical abilities could be ignored when rating dissimilarity. It was also made clear that none of the voices were synthesised and that all vocalisations in each listening session were performed by a single singer.

5.2.5 Data Encoding

The questionnaire that participants answered was designed in such a way that the answers to multiple-choice questions could be ranked. This section describes several methods used to combine, interpret and convert participants' answers so that they may be used in automatic processing and statistical analysis.

Potential answers to Question 1 regarding participants' listening equipment were ranked in the order they were shown from 1-3. The answers to Question 2 about their listening environment were ranked 1-5. The sum of these two answers provides an overall *listening potential* score ranging from 2-8.

Questions requiring written answers were converted by the author to integers

representing either ordinal or categorical data to facilitate subsequent analysis. Participants were asked at the end of the study in Question 7, to optionally provide comments regarding their ratings or the study itself. It was found that during the pilot study, this type of question would coax participants to disclaim any uncertainties they had about doing the task correctly. This answer was converted by the author to ordinal categories ranging from 0 to 4, reflecting how well the participant understood their task. As part of the screening process, participants who scored 0 were removed from the dataset.

The GOLD MSI's perceptual ability questions allowed participants to answer on a 7-point Likert scale ranging from 'strongly disagree' to 'strongly agree'. These were summed up in accordance with the GOLD-MSI subscale scoring template to determine each participant's perceptual ability score.

An additional method for assessing musical aptitude for this listening study was derived from participants' answers to Question 5 where they were asked what instrument they played, if any. From this, ordinal category values were assigned as 0, 1, or 2, depending on whether the participant was a non-musician, musician or vocalist.

Finally, a post-hoc evaluation of participant feedback to Question 6 provided some insight regarding how well they understood the task, or what they found challenging due to particular interface design features. The researcher interpreted these answers and assigned an ordinal score from 0-3. This reflected the participants self-proclaimed understanding of the task and is termed the *task comprehension* score. An additional category '4' was reserved for participants who chose not to answer.

5.2.6 Participant and Data Screening

Kreiman et al. [1993], Labuschagne and Ciocca [2016] have highlighted a number of issues that relate to poor perceptual ratings, such as listeners' diverse backgrounds, biases, poor task descriptions, task-listener interactions, and random errors. To counteract these potential pitfalls, participants' submitted data was screened in several ways to ensure that the dataset used for analysis was reasonably reliable. The result of these screening processes was evaluated by manually

examining the dissimilarity matrices generated from the participants' data (see Figure 5.1), which provides some visual indication of how much structure and noise is likely in a participant's submitted ratings. All data screening took place after the listening studies had finished, except in cases where participants reported hearing impairments that might affect their ability to perform similar tasks, or yielded a *listening potential* score less than 3, their listening study was terminated with an explanation, as they were considered ineligible.

If participants understood the task properly, their generated dissimilarity matrices would have a vector of zeros along the diagonal axis. This would reflect their assertion that there is no difference in singing technique between two audio clips of the same recording. If participants did not understand the task or provided unreliable data, the diagonal axis would possess a considerable amount of noise. Due to the design of the listening study's interface, which facilitates multiple comparative audio clips to rate against one reference clip per page, it is reasonable for participants to make at least one mistake in identifying identical vocal techniques (same audio clips). It is also reasonable to consider that participants may not be perfectly accurate in dragging the sliders all the way to 'exactly' zero. We therefore omit participants who have more than 2 ratings for self-audio comparisons that are more than 0.1, and refer to this metric as *poor identity recognition*.

During a listening session, participants were unknowingly subjected to two *repeated* pages randomly hidden among the unique pages. Participants who were careless or lacked attention were expected to be unlikely to accurately repeat their ratings between repeated pages. Mehrabi [2018] used Spearman's rank-type correlation between repeated tasks to detect unreliable data, while Gerratt and Kreiman [2001] calculated percentage errors for repeated ratings deviating from more than a single point on a Likert scale. Neither of these approaches seemed suitable in this context due to the continuous nature of the rating interface. Instead, an RMSE value was calculated between ratings for repeated tasks. Participants' data were removed from the dataset if their corresponding RMSE value was above a given threshold value. During the pilot studies, we asked some participants to perform careless and rushed ratings, while others were asked to provide cautious and considered ratings. Examining this data allowed us to determine the RMSE threshold that separates good from poor raters to be 0.4. Participants' generated

RMSE values above this threshold will be intuitively referred to as the *inconsistency* score.

To visualise how much inter-participant agreement existed in the reduced dataset, a correlation matrix was generated between each dissimilarity matrix of all participants. This step was proposed by McAdams et al. [1995] in order to find outliers, or participants who had a suspiciously different style of rating to others. However, in order to compare dissimilarity matrices, we must ensure that the order and number of classes per participant matrix are the same. As an error with data collection led to omitting ratings relating to one random audio clip per session, dissimilarity matrices did not contain information in a uniform manner. To make up for this discrepancy, a dummy row and column were added to each matrix in the place where the randomly omitted ratings should be. Each cell in these added vectors contained an average of the participant's ratings across their listening session. This ensured that ratings would be uniformly presented across all participants' dissimilarity matrices without inducing a bias.

5.2.7 Analysis

Clustering

All data that passed the screening tests were subjected to a combination of clustering analysis techniques, as in the work of McAdams et al. [1995], Gerratt and Kreiman [2001], Grey [1977] to deduce whether the existing labels of the stimuli were appropriate, as well as how different groups of participants perceived the stimuli in relation to their ground truth labels. Clustering algorithms were fed the dissimilarity matrices of each participant, where each row can be thought of as a data point representation, and each column as a feature vector. Clusters were first predicted using two methods for solutions $\{k \in \mathbb{Z} \mid 2 \leq k \leq 14\}$: the *agglomerative clustering* using *Ward linkage*; and the *K-means* algorithm.

To estimate the accuracy of participant's perceptual space w.r.t. the ground truth, *sklearn's* `adjusted_rand_score` metric was used, which provides a measure of similarity between the groups of predicted and ground truth labels. This is calculated by considering all pairs of samples and counting all pairs that are assigned to the same or different clusters between the ground truth and pre-

dicted cluster assignments. This provides a list of comparisons that are true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). The following formula produces the Rand Index (RI) [Rand, 1971]

$$\frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}. \tag{5.3}$$

Finally, the *Adjusted* Rand Index (ARI) is computed using the following formula in order to consider chance level similarities between clustering solutions, calculated as

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathrm{ExpectedRI}}{\max(\mathrm{RI}) - \mathrm{ExpectedRI}} \tag{5.4}$$

where Expected RI is the expected RI if clustering was randomised.

Cluster *cohesion* (how similar data points to one another within a cluster) is measured by getting the mean intra-cluster distance, while cluster *separation* (how far data points are to other clusters) is measured by getting the mean nearest-cluster distance between clusters. The following equation calculates each of these measurements w.r.t. individual samples, and combines them to produce a silhouette score. This is computed as

$$(b - a)/max(a, b), \tag{5.5}$$

where a represents the distance between a sample and other samples of the same cluster (cohesion), and b represents the distance between a sample and the nearest cluster it is not a part of (separation). The silhouette score for the entire solution is the average of these scores across each sample. Scores of 0 and 1 imply clustering solutions that heavily overlap or separate, respectively, and a negative score implies incorrect clustering where a data point is not assigned to its nearest cluster. The distribution of these metrics for different numbers of cluster solutions k, were compared across all participants, as well as between groups of participants that were subjected to a different register (low/high) or gender (male/female) condition.

Statistical Tests

Participant features are the answers participants provided to the questionnaire that were converted to interval or ordinal data, which reflect the following: hearing impairments, listening potential, musician category, MSI score, task comprehension, poor identity recognition, and inconsistency. Participants' silhouette and accuracy scores were also included in the feature sets. The combination of these features leads to a collection of ordinal and continuous data, and so Spearman's and Pearson's rank correlation was used to test for the strength of monotonic relationships between them.

Kruskal-Wallis and Mann-Whitney U tests were used to determine whether the imposed conditions of a listening session led towards better or worse cluster silhouette or accuracy scores for values of k that provided the best clustering scores on the majority of the dataset. The choice of statistical test methods was primarily informed by Greene and D'Oliveira [2005], Mumby [2002].

Matrices to 2D-plots

Perceived distances between specific classes (and not specific audio clips) will be referred to as *pairwise class distances* (PCDs). As there are 5 technique classes to rate between, the total number of PCDs using Eq. 5.2 is 15. The collected 105 pairwise distance ratings were first reduced to a 5-dimensional dissimilarity matrix to present the 15 PCDs. These matrices were then summarised across all participants to yield a single dissimilarity matrix per condition. The methods used to do this summarisation were dependent on whether the PCDs were found to be normally distributed or not.

These condition-specific PCD dissimilarity matrices were then projected onto a 2-dimensional plane using MDS, which has been frequently used in the literature to visualise perceptual data [Grey, 1977, Kruskal, 1964, Shepard, 1962a,b, Gerratt and Kreiman, 2001]), leading to an intuitive visual representation of the perceptual space. The non-metric MDS algorithm[5] was used to produce the best-fitting dimensionally reduced representation of the data points.

[5] `https://scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html`

5.3 Results and Discussion

In this section a combination of parametric and non-parametric statistical tests will be used. All data will be tested for normal distributions using the Shapiro-Wilk test with a p-value of 0.05 [Shapiro and Wilk, 1965], and visual verification of distributions before choosing an appropriate statistical test. The distributions of *participant features* are shown as histograms and bar charts in Appendix A.2, illustrating the diversity among participants.

5.3.1 Data Screening

After closing the call for participants, data of 61 listening sessions was extracted from WAETs XML files in which they were stored. The screening process from Section 5.2.6 was applied to ensure participants and their ratings met the minimal criteria. Six participants failed to provide data that passed this process. These included three whose feedback scored '0' on task comprehension, 2 who failed inconsistency test, and 2 who failed the poor identity recognition task (one of whom already failed the inconsistency test).

A resulting dataset of 55 participants was encoded in HDF5 format and used for analysis. The results of a Spearman correlation matrix between participants' 15-dimensional dissimilarity matrices (generated as described in Section 5.2.6) can be seen in Figure 5.4. This shows that there is a considerable *lack* of correlation and a large amount of variance between participants' ratings across that dataset.

As each listening session was used roughly 5 times, it is also sensible to view correlation matrices between participants' data when grouped by session. Figures 5.5 and 5.6 show these correlations. The unbalanced matrix dimensionality between subplots is due to the removal of certain participants' data due to screening processes. These plots also surprisingly demonstrate a considerable lack of correlation between participants, meaning that even when participants listened to the same stimuli, there was a considerable amount of variance between their dissimilarity matrices.

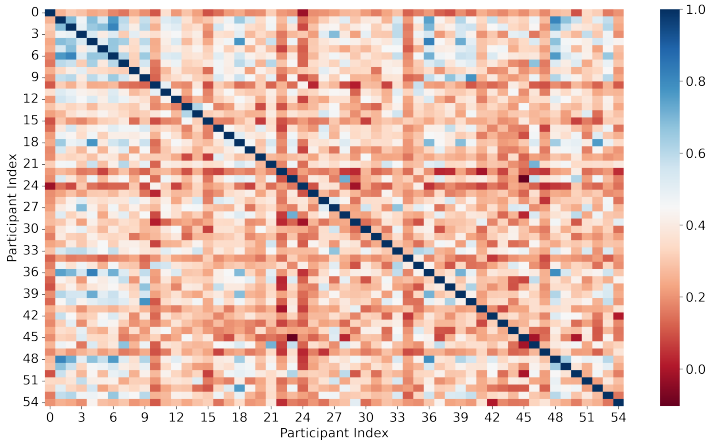


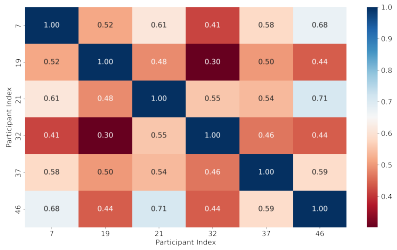
Figure 5.4: Correlation matrix between dissimilarity matrices of each participant, indicating significantly more uncorrelated matrices than correlated ones.

5.3.2 Cluster Scores

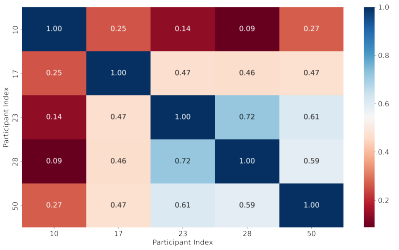
Clustering algorithms for values of $\{k \in \mathbb{Z} \mid 2 \leq k \leq 14\}$ were first applied across all participants' data, from which accuracy and silhouette scores could be calculated. Values of k with the highest cluster scores were stored for each participant, and the k value that most frequently yielded the highest scores was considered the optimal value w.r.t. the score measured. The distribution of these values is presented in Figure 5.7.

We can infer from this that there are more participants perceiving the stimuli in a manner that leads towards clustering that best fits the ground truth labels when $k = 5$ than for any other value of k. Conversely, the optimum k value for silhouette scores was 2, suggesting that the compactness and separateness of clusters among participants' data are poor and need the minimal number of clusters to obtain high silhouette scores. This suggests that participants are not perceiving the different classes as particularly salient from one another.

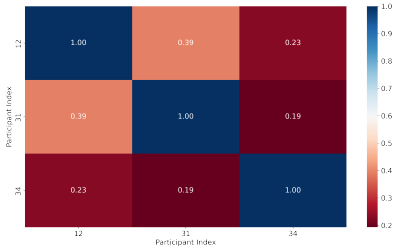
As the distributions between both clustering algorithms were similar for accuracy and silhouette scores, a correlation test was conducted between them. After confirming normal distributions for all score types, the Pearson correlation test was used. Results shown in Figure 5.8 illustrate correlations for silhouette and



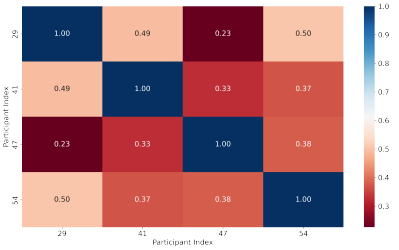
(a) Data Correlation for Session 'm1low'



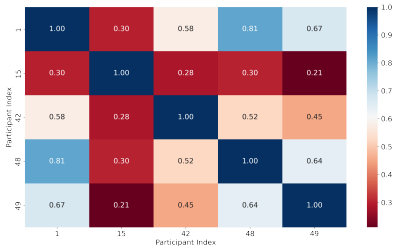
(b) Data Correlation for Session 'm1high'



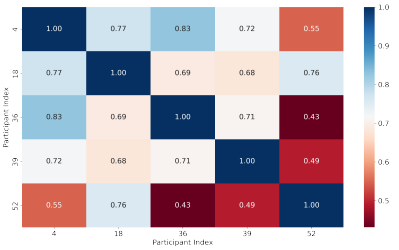
(c) Data Correlation for Session 'm2low'



(d) Data Correlation for Session 'm2high'

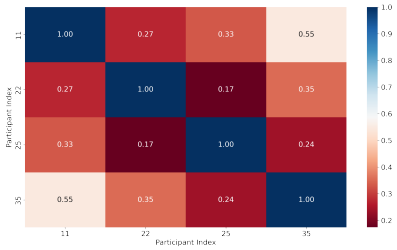


(e) Data Correlation for Session 'm4low'

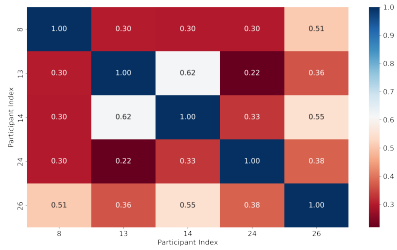


(f) Data Correlation for Session 'm4high'

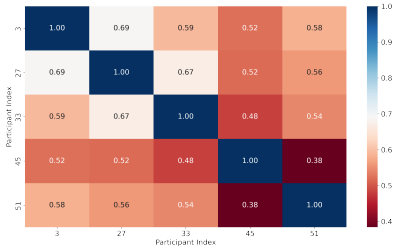
Figure 5.5: Correlation matrices of data between individual listening sessions of male singers



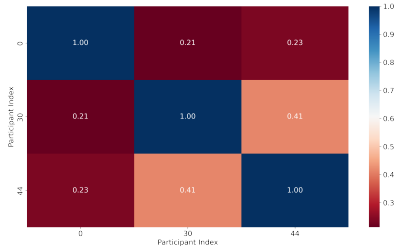
(a) Data Correlation for Session ‘f2low’



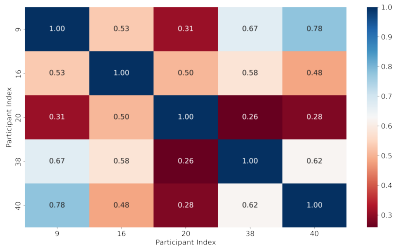
(b) Data Correlation for Session ‘f2high’



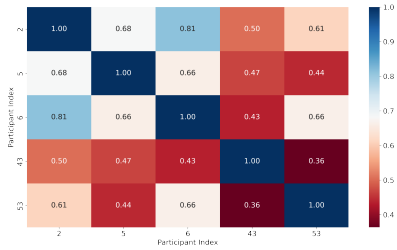
(c) Data Correlation for Session ‘f3low’



(d) Data Correlation for Session ‘f3high’



(e) Data Correlation for Session ‘f5low’



(f) Data Correlation for Session ‘f5high’

Figure 5.6: Correlation matrices of data between individual listening sessions of female singers

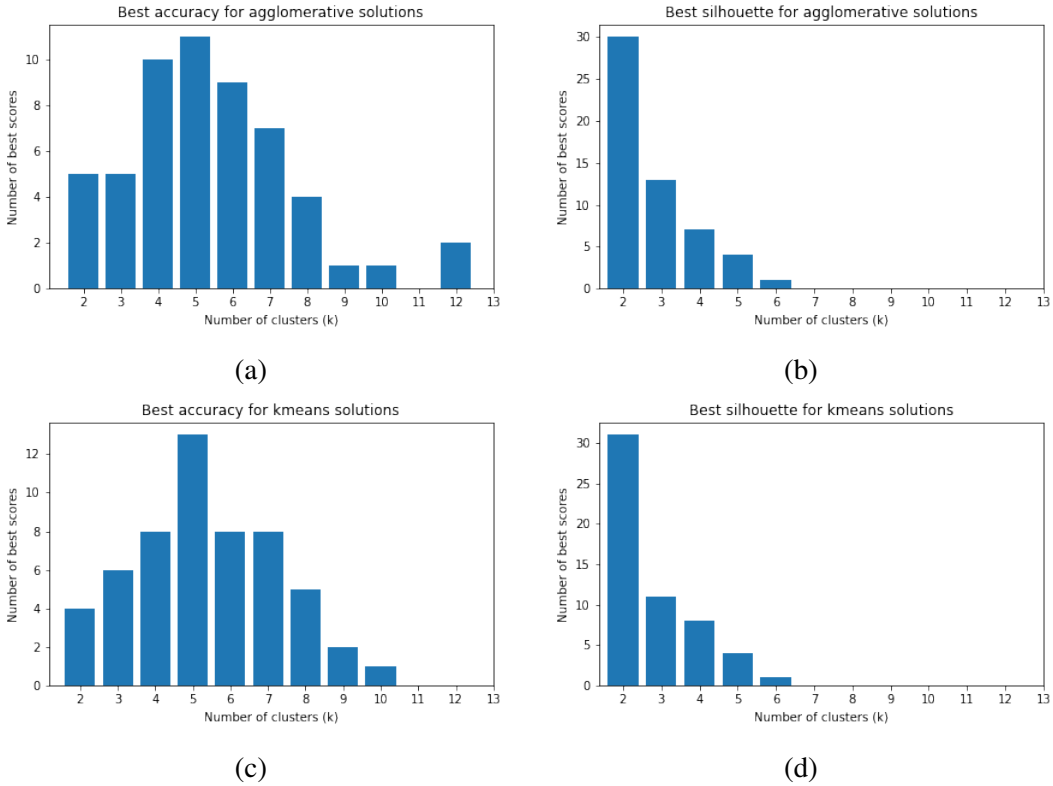
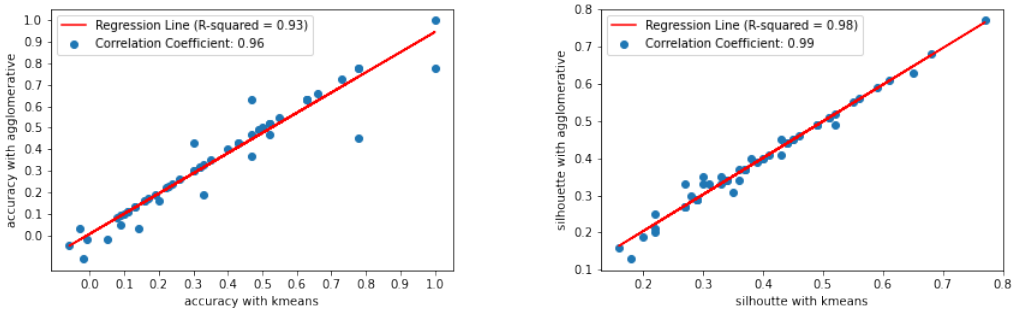


Figure 5.7: Distributions across all data for best accuracy (left column) and silhouette (right column) scores across all values of k, using agglomerative (top row) and k-means (bottom row) clustering.



(a) (b)

Figure 5.8: Scatter plots displaying correlations between metrics using k-means and agglomerative clustering algorithms.

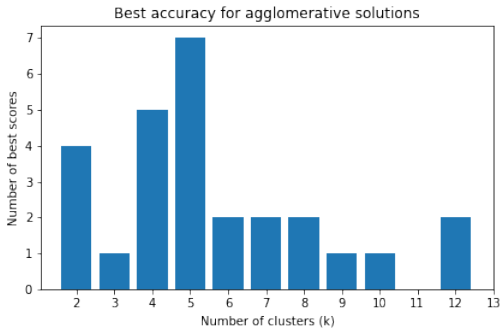
accuracy scores of 0.99 and 0.96 ($p < 0.001$) respectively. For this reason, the k-means algorithm is no longer reported in this analysis, as it was considered a redundant measurement.

5.3.3 Experimental Conditions (Controlled Variables)

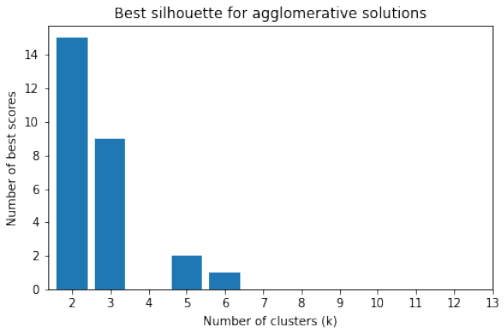
In this section, we examine how collected data may have been affected by the conditions imposed by the listening sessions, namely the register and gender conditions. The dataset was split to group listening sessions of specific conditions together for these analyses.

Best *k* Distributions

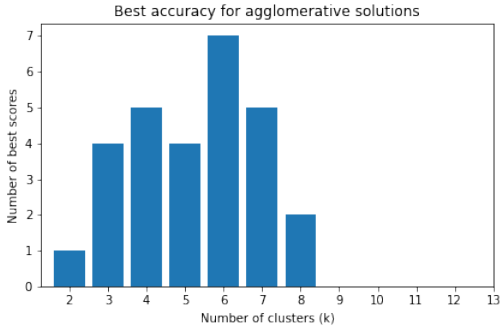
After defining the conditional groups, clustering scores were generated for each one in the same way as described in Section 5.3.2 above. The results for samples grouped by register conditions are shown in Figure 5.9, while the gender conditions are shown in Figure 5.10. None of the silhouette best-k distributions were found to be normally distributed, and only some distributions of accuracy best-k distributions were normally distributed. A Kruskal Wallis test concluded that there were no statistically significant differences between register or gender conditions for silhouette scores. A Kruskal Wallis and one-way ANOVA test (both were investigated due to the ambiguous nature of the distributions) concluded the



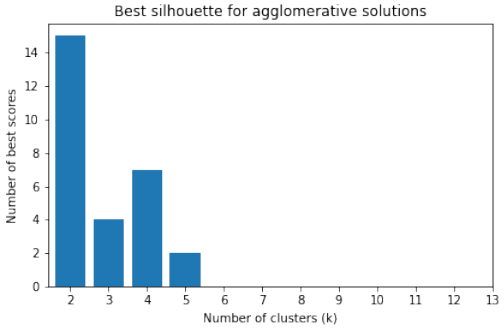
(a) 'high' condition best accuracy



(b) 'high' condition best silhouette



(c) 'low' condition best accuracy



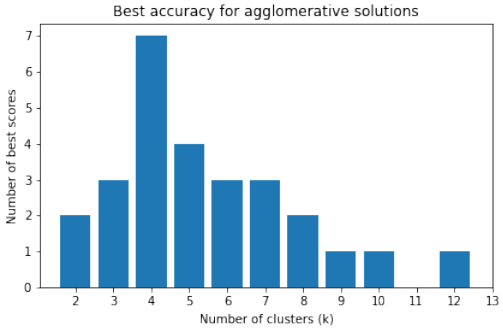
(d) 'low' condition best silhouette

Figure 5.9: Distributions for best accuracy (left column) and silhouette (right column) scores across all values of *k*, for each register condition.

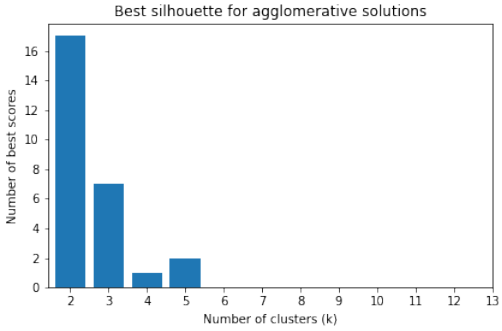
same for accuracy distributions. This suggests that the conditions had no effect on how well participants' perception of the stimuli related to the ground truth labels, or how well the stimuli clustered in their perceptual space. However, given the nature of what is being measured, a considerably larger sample size would be required to give statistical analyses enough power to reject the null hypothesis with confidence.

Cluster Score Distributions

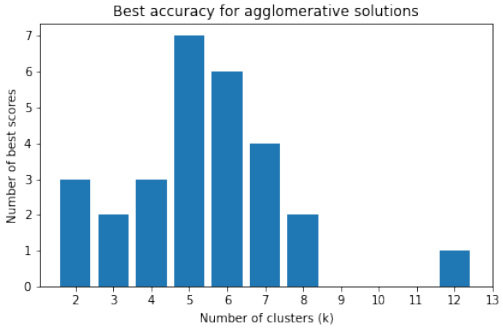
Using the number of ground truth labels in the stimuli as a frame of reference, accuracy and silhouette scores were regenerated using agglomeration clustering with $k = 5$, and split into conditional groups for comparative analysis. Accuracy scores were normally distributed and subjected to a one-way ANOVA test, while



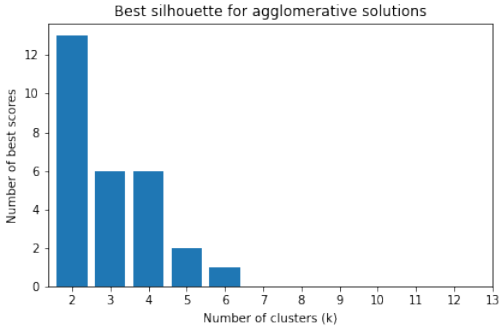
(a) ‘female’ condition best accuracy



(b) ‘female’ condition best silhouette



(c) ‘male’ condition best accuracy



(d) ‘male’ condition best silhouette

Figure 5.10: Distributions for best accuracy (left column) and silhouette (right column) scores across all values of k, for each gender condition.

Table 5.1: Mann Whitney U results for significant differences between register conditions

PCD	*U*	*p*	Higher Median	Lower Median	Effect
belt-belt	532.5	<0.01	low=0.32	high=0.24	0.41
straight-belt	526.5	<0.02	low=0.59	high=0.5	0.39
fry-vibrato	501.0	<0.05	low=0.85	high=0.76	0.33
belt-vibrato	497.0	<0.05	low=0.56	high=0.47	0.31

Table 5.2: Mann Whitney U results for significant differences between gender conditions

PCD	*U*	*p*	Higher Median	Lower Median	Effect
straight-breathy	575.0	<0.001	male=0.7	female=0.47	0.52
fry-vibrato	205.0	<0.005	female=0.91	male=0.72	-0.46
vibrato-vibrato	226.5	<0.02	female=0.22	male=0.11	-0.4
breathy-breathy	234.0	<0.02	female=0.2	male=0.09	-0.38
belt-fry	257.5	<0.05	female=0.82	male=0.67	-0.32

half the silhouette scores were non-normally distributed and therefore subjected to a Mann-Whitney U test. No significant differences existed between conditions for either score, indicating that neither register nor gender conditions influenced how well participants' perceptions agreed with ground truth labels, or the salience of clusters in their perceptual space.

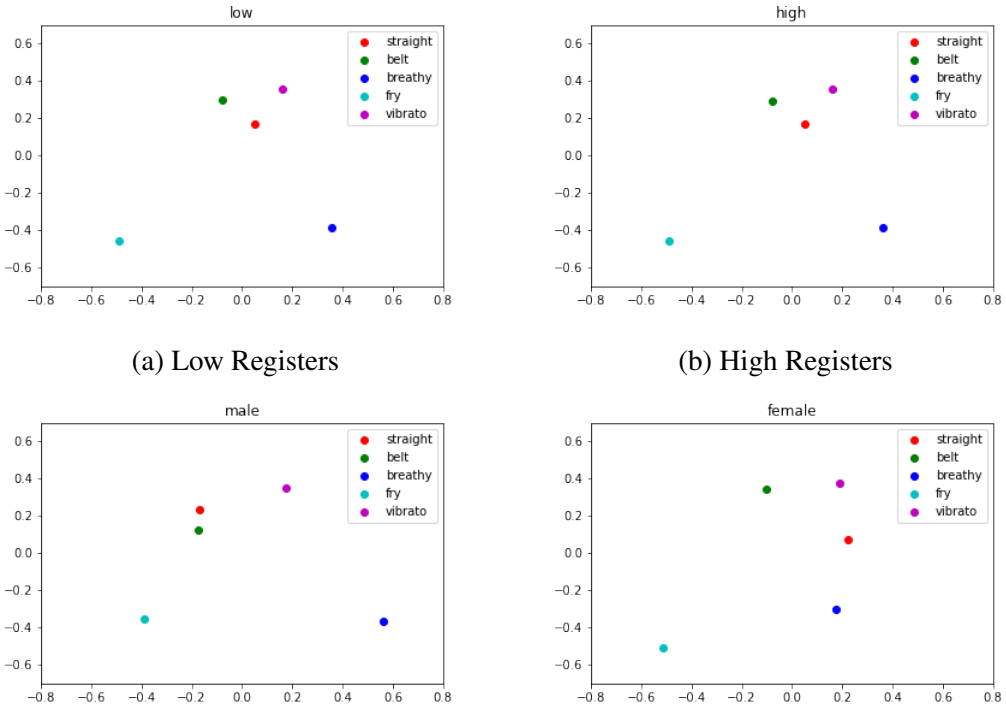
Pairwise Class Distance Distributions

When split into conditional groups, PCDs were found to be non-normally distributed. Accordingly, these were subjected to Mann-Whitney U tests, the results of which (along with the reported effect size [Cohen, 1988, Perugini et al., 2018, Fritz et al., 2012]) are displayed in Tables 5.1 and 5.2 respectively. While it may not be necessary to discuss each result for PCDs, it is interesting to note that within-class distances such as *vibrato-vibrato* or *breathy-breathy* PCDs were significantly larger for females than males. It is also interesting to note that the majority of detected differences between gender conditions are larger for female voices, while all detected differences between register conditions were larger for low registers.

5.3.4 Multidimensional Scaling

As described in Section 5.2.7, the 14-dimensional matrices were compressed to a dimensionality of 5, combining stimulus-specific dissimilarities into PCDs. To determine the best method of summarisation, the distribution of the relevant class-pair ratings first had to be considered. This type of data analysis check has often been side-stepped in the pursuit of presenting a conclusive generalised behaviour of the data, [McAdams et al., 1995, Iverson and Krumhansl, 1993, Wan et al., 2018], admittedly in cases where the end seems to justify the (statistical) means. In this case, the Shapiro-Wilk test and visual assessment of the histogram distributions led to the conclusion that the majority of these ratings were not normally distributed. Many of the distributions featured a ceiling or floor effect, where a strong majority of participants chose rating '0' or '1', representing the maximum or minimal difference in the aural perception of singing techniques. Using methods such as square root or logarithmic transformations to normalise the distributions did not seem appropriate as the interpretation of these w.r.t perceptual dissimilarities would not be intuitive or useful, nor would such transformations help against such ceiling or floor effects where values would remain the same. To mitigate the effects of non-normally distributed data and better represent the central mass of these distributions, the median was used instead of the mean [Sainani, 2012]. MDS was then performed on these median PCD values to produce perceptual maps of singing techniques, shown in Figure 5.11.

Some general observations can be made but with caution due to non-normal distributions. The perceptual spaces between low and high registers are generally very similar. Although multiple significant differences were detected between these conditions in the previous section, they were generally medium in size. Gender conditions, on the other hand, display more visible differences between their perceptual maps. With respect to *straight*, *belt* and *vibrato* techniques, male singers' *vocal fry* technique seems to be closer than females, while female singers' *breathy* technique seems to be closer than males. The distance between *straight* and *belt* is noticeably smaller in male singer perceptual spaces than in females.



(a) Low Registers

(b) High Registers

(c) Male Singers

(d) Female Singers

Figure 5.11: Plots displaying the perceptual space of singing techniques after dimensionality reduction, grouped by low, high, male and female singing conditions. Axes are not labelled as MDS does not produce coordinates based on predefined measurable concepts, due to its inherent objective of summarising data based on shared correlations, variances and relative distances. Further post-hoc objective and subjective evaluations however, could be used to determine the meaning of each dimension.

Testing by Participant Conditions

Statistical analyses were also conducted when data was grouped by participant conditions, which refer to categorical participant features as described in Section 5.2.7.

A Kruskal Wallis test was initially used to compare accuracy scores when samples were grouped based on whether participants were non-musicians, musicians, or vocalist-specialists. Results showed a significant difference between one of these groups, which resulted in a posthoc Mann-Whitney U test being conducted with the Bonferroni correction applied to the significance threshold. Significant differences were found in accuracy samples between non-musicians and vocalists ($U = 10, p = 0.005$) as well as musicians and vocalist-specialists ($U = 76.5, p = 0.006$), where the vocalist accuracy medians were highest. It should be noted however, that all statistical reports involving participant's instrumentation must be considered with caution, as the 'non-musician' and 'vocalist' categories consisted of small sample sizes of 10, while the 'musician' category contained 35 samples.

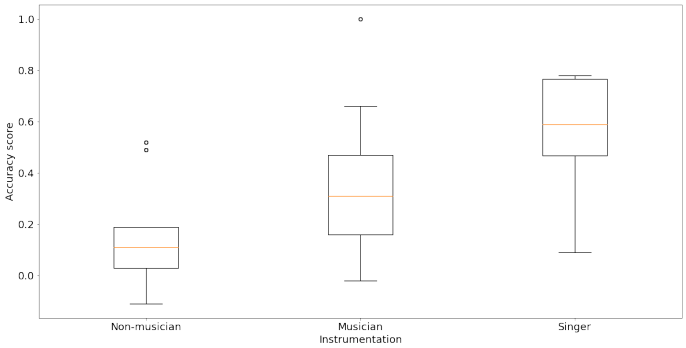
Correlations Among Participant Features

Participant features as described in Section 5.2.7 were compared against one another using the Spearman's rank or Pearson correlation, based on the type of data and distribution. Statistically significant correlations are shown in Table 5.3. They are also illustrated via scatter-plots in Figure 5.13 for continuous data, and boxplots for ordinal data in Figure 5.12. The highest of these correlations is between silhouette and accuracy, which was not surprising as it implies the perceptive spaces that best agreed with ground truth labels also featured strong cluster structures. The second highest correlation was between instrumentation and MSI scores. It is hypothesised that singers had a tendency to attain higher MSI scores as the questions were slightly biased towards singing ability or perception of singers. Instrumentation and MSI were also mildy to strongly correlated with silhouette and accuracy scores respectively, intuitively suggesting that specific knowledge one has on the stimuli grants one an informed perceptual bias that adheres to the existing taxonomy, with confidence reflected in well-defined clus-

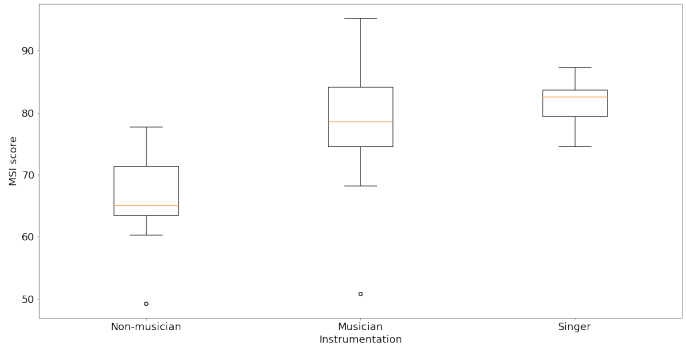
Table 5.3: Statistically significant differences between the different participant features including accuracy and silhouette scores. The subscript 'S' or 'P' indicates whether the correlation was of type Spearman or Pearson.

feature A	**feature B**	**r-value**	**p-value**
Silhouette	Accuracy	0.62_P	$<$0.001
Instrumentation	MSI	0.48_S	$<$0.001
Instrumentation	Accuracy	0.47_S	$<$0.001
MSI	Accuracy	0.47_P	$<$0.001
MSI	Silhouette	0.36_P	$<$0.01
Instrumentation	Silhouette	0.27_S	$<$0.05
Inconsistency	Silhouette	-0.27_S	$<$0.05

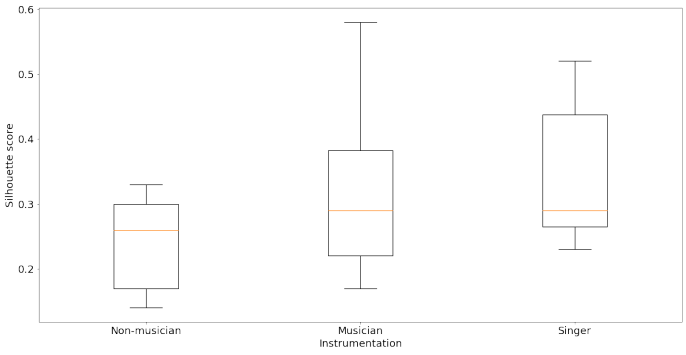
ters. Participants' inability to repeat ratings for repeated pairwise comparisons (inconsistency) was correlated negatively with silhouette scores. This intuitively suggests that participants who were less consistent in their rating criteria produce poorly defined clusters in perceptual space.



(a)



(b)



(c)

Figure 5.12: Box plots illustrating distributions of scores for each category across the y-axis, where there was measured correlation with the instrumentation categories of an ordinal nature shown on the x-axis.

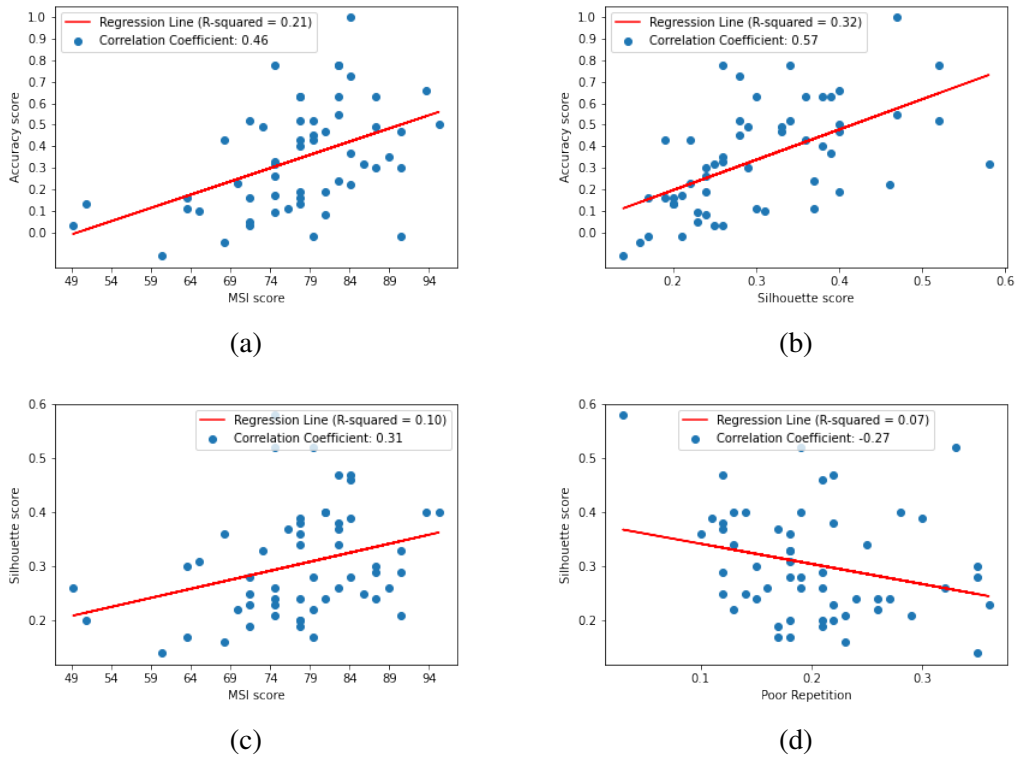


Figure 5.13: Scatter plots illustrating a mild correlation between the measurements of the attributes labelled on the x and y-axes.

5.4 Conclusion

5.4.1 Results

The results of this experiment cover numerous aspects regarding the participants, their PCD ratings, and the perceptual space between these PCDs. These were compared between the conditions of register and gender.

While $k = 5$ (the number of ground truth classes) resulted in the highest count above any other value for best accuracy scores, it was still not the case for a majority of participants' data. This combined with low silhouette scores for $k = 5$ across the dataset implies participants had difficulty perceiving salient features among the stimuli and frequently provided data that did not perceptually match the ground truth classes. No statistically significant differences were found between the best k distributions for clustering scores when grouped by conditions. This was also the case when comparing distributions of accuracy and silhouette scores under the same conditions, suggesting that these conditions had no effect on how participants' perceptions of the stimuli clustered.

When testing for statistical differences between PCDs however, a number of differences were detected between conditions. Interestingly, all detected differences in distances were significantly lower in low register conditions, and the majority of detected differences were larger for female conditions. The low register condition resulted in larger within-class variances for *belt* vocalisations, while the female singer condition had the same effect on *vibrato* and *breathy* vocalisations.

MDS was used to convert dissimilarity matrices into coordinates for each singing technique on a two-dimensional plane. This gives readers a visual of how the perceptual space of the singing technique is mapped out for different conditions. They suggest perceptual maps are very similar between register conditions, and significantly different between gender conditions.

Testing for correlations between participant features resulted in a number of findings presented in Table 5.3 in the previous section. Accuracy and silhouette scores were the most correlated of participant features. In general, participants with a more relevant musical background to the task produce better clustering scores.

5.4.2 Reflection

A global correlation matrix across all participant data showed very little correlation between them. Further posthoc tests confirmed that there was a large amount of noise even between data generated from the same listening sessions. In conjunction with the findings summarised in the previous section, it seems clear that a considerable amount of noise exists among participant ratings. This could be due to the task description needing further refinement; the fact that the task itself may have been too difficult; or the choice of interface. Revisiting any of these aspects may improve the listening study, and lead to more normally distributed data, where differences between conditions might be more pronounced.

Many participants reported in their feedback the factors that influenced or dictated their dissimilarity evaluations. An exhaustive list of these factors is as follows: performer's lack of control, soft/harshness, clean/dirty, distortion, dynamics, temporal pitch variation, subglottal pressure, larynx placement, resonance, nasalness, open/closed mouth, total amount of notes per sample, melody, emotion, register mechanisms and *assumed* class types. Many of these attributes imply that there is a considerable degree of uncertainty regarding the dissimilarity evaluation task and it is reasonable to believe this has caused a significant level of noise in the results.

The amount of noise in collected data could also be related to the fact that the VocalSet dataset itself has its own shortcomings. Many recordings contain multiple techniques, despite being labelled with only one. The quality of performances seems to vary considerably between singers. Due to the nature of the *fry* technique and variance in performance style, its pitch is often a number of octaves below the singer's intended pitch (and the dataset's implied pitch label). Wilkins et al. [2018] specify that singers were placed close to the microphone, but it may be the case that there is variance in this 'closeness', deduced from an audible proximity effect (an increase in lower frequencies due to the singers' closeness to a microphone) between singers - however this has not been quantitavely measured.

5.4.3 Future Work

A repository for this work is available online[6], where an anonymised version of the collected data is available, as well as a walkthrough of the clustering, MDS and statistical analysis techniques used. It is the author's hope that this will be useful to other researchers interested in doing the same type of study, and steer them towards designing an experiment that gives more robust results. Another potentially interesting question to investigate (that wasn't covered in this chapter) in future research is whether the gender of a listener affected how they perceived same or different-gendered singers.

The PCDs can also be used to generate coordinates that can be used as part of a regularisation term in generative NNs. If there is an improved difference in voice synthesis when using this additional data as a contributing loss component, then it is clearly a meaningful representation of perceptual space.

As five was still the most frequently successful value of k across the dataset w.r.t accuracy scores, it is appropriate to assume that the existing taxonomy is well suited to most informed listeners and can continue to be used in future studies.

[6] `https://github.com/Trebolium/VoicePerception`

Chapter 6

Zero-shot Singing Technique Conversion

6.1 Introduction

Chapter 5 explored the perceptual spaces of singing techniques, where it was suggested that listeners from a more musical background (along with clustering metrics) were more likely to perceive timbral structures that favoured the ground truth labels. Although some shortcomings of VocalSet have been listed, it was found to distribute vocalisations into the most perceptually appropriate sets, making it a potentially suitable dataset for facilitating singing technique conversion (STC). Perceptual spaces between pitch register conditions were not significantly different, while those between gender conditions induced more noticeable but not large differences. These findings suggest that it is reasonable to attempt to use one system to model singing techniques across all conditions to perform STC.

The task of STC could have an influence on music production similar to other voice manipulation tasks described in Section 1.1.2, as it opens up the possibility of artistically manipulating a singer's *performance*, rather than simply quantising their pitch or swapping out their voice for another. With the explosion of probabilistic ML techniques in recent years, there has been a great deal of research focused on voice transformation for speech, while less attention has been given to singing. The topic of transforming the expression of the singing voice is almost untouched, leaving an unexplored gap in voice synthesis.

This chapter covers research that was presented at the CMMR 2021 symposium [O'Connor et al., 2021]. Section 6.2 describes the components and architecture used in Qian et al. [2019]'s implementation of their spoken voice conversion system. Section 6.3 describes how this is repurposed for the task of STC, highlighting the number of ways in which this research differs from that of Qian et al. [2019]. Section 6.4 documents a listening test where participants were asked to rate the similarity (w.r.t. the target singing techniques) and naturalness of converted audio produced by a trained STC network. These ratings are then analysed and discussed in order to evaluate the performance of the model. Examples of models' synthesised audio output are available to audition online [1].

[1] `https://trebolium.github.io/singing_technique_conversion/`

6.2 AutoVC System

At the time of initiating this experiment in Winter of 2020, a number of variations on autoencoder and GAN architectures were available that largely relate to, or are specifically designed for, VIC (see Section 4.4.4). However in this experiment, the singing technique is the attribute of interest being tested for conversion, and so the question remains as to whether existing architectures are appropriate for the singing voice domain, and whether solutions to the conversion task can be applied specifically to the singing technique. Among these, the AutoVC system [Qian et al., 2019] was considered as the most suitable candidate for singing technique conversion, due to its state-of-the-art results for the task of VC, elegant solution towards linguistic disentanglement, zero-shot capabilities, and influence on proceeding literature and research. The remainder of this section will cover all aspects of the original implementation. Section 6.3 will describe proposed alterations to facilitate STC.

6.2.1 Input Features

AutoVC is designed to take mel-spectrograms as input features. In the original implementation, the audio was first resampled to a standard sampling rate of 16kHz. It was passed through a Butterworth highpass filter to remove low-frequency components below 30Hz using a filter order of 5. A small amount of white noise was added to the audio signal to improve AutoVC's generalisation to new audio, improving its robustness. The audio signal was then converted to a log mel-spectrogram representation, using the processes described in Section 4.3.1. The parameters for this transformation included an FFT length and hopsize of 1024 and 256 respectively, a mel-filter bank of dimension 80, and a frequency range of 90Hz to 7.6kHz. The resulting computed log mel-spectrograms (herein simply referred to as spectrograms) were normalised between a range of 0 and 1, and stored. Spectrograms were randomly sampled from a dataset, and a random chunk of T frames was extracted from the sampled spectrogram.

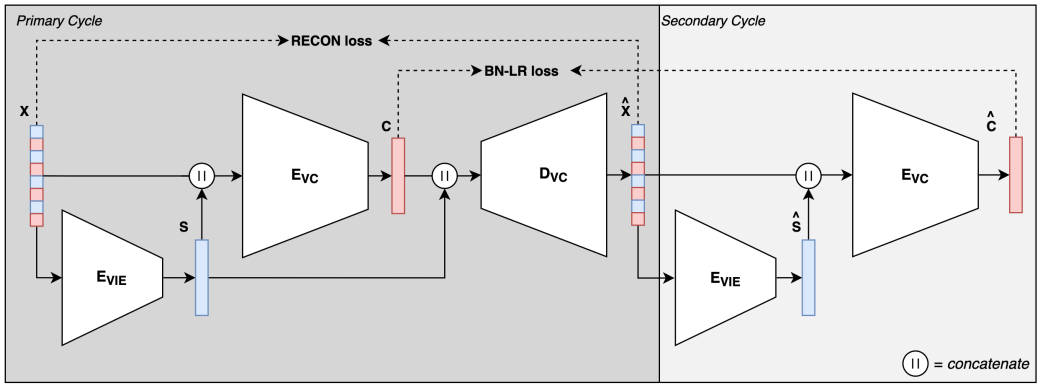


Figure 6.1: Flowchart illustrating the information flow of the AutoVC system. The secondary cycle illustrates how AutoVC's output is re-inserted as its input for a secondary pass of the system to obtain new embeddings generated from the synthesised output.

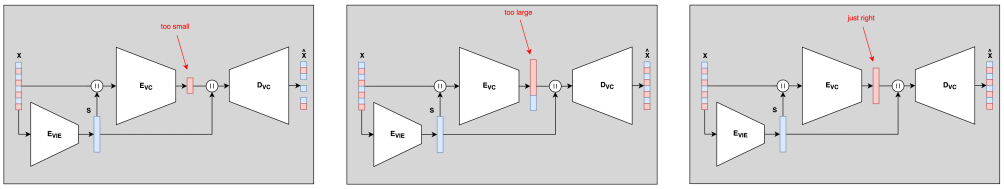
6.2.2 Details of AutoVC

As described in Section 4.4.4, AutoVC is an autoencoder that relies on the combined approach of VIE conditioning and bottleneck calibration. Figure 6.1 illustrates the AutoVC system as a flowchart from left to right. After training, it has learned a disentangled representation of the input data, which allows it to perform VIC.

AutoVC can be broken down into two sub-networks. The first sub-network will be referred to as the VC network. It is an autoencoder and is the primary engine behind the conversion task, the encoder of which will be referred to as E_{VC}. The second is the VIE network, E_{VIE}, which was pretrained to produce embeddings exhibiting feature variances that describe the unique, discriminative qualities of each voice.

VIE Conditioning

As shown in Figure 6.1, the output embeddings of E_{VIE} are concatenated with its spectrogram input features, $\mathbf{X}$, before being fed to VC network's encoder, E_{VC}. As $E_{VIE}(\mathbf{X})$ is a one-dimensional feature vector while its prospective concatenative partners are 2-dimensional representations, it is broadcast across their temporal axis, allowing the two representations to be of the same temporal dimension



(a) Too small (b) Too large (c) Just right

Figure 6.2: Flowcharts illustrating three scenarios where: (a) the bottleneck is too small to encode linguistic content; (b) the bottleneck is too large, allowing it to also encode voice identity content; (c) the bottleneck is just the right size to allow E_{VC} to disentangle *only* linguistic content from the spectrogram.

for concatenation.

While this step makes the input dimensions significantly larger, it is advantageous in that it presents an embedding which represents voice identity information entangled with linguistic content, and an embedding that represents the already disentangled voice identity information. Inherently, the network will have a much easier time extracting information unrelated to voice identity from these combined embeddings. The embeddings from E_{VC} are again concatenated with VIEs before being fed to the decoder D_{VC}.

Bottleneck Calibration

If the network is continuously provided with accurate, consistent VIEs, it only needs to encode non-VIE content (such as linguistic content) at the bottleneck in order to reconstruct the original spectrogram representation using D_{VC}. If the bottleneck is made to be too small, reconstruction will suffer due to insufficient data encoding. However, if it is too large, this allows room for other non-linguistic information from the spectrogram (such as voice identity) to be encoded, compromising the amount of disentanglement. Ideally, the bottleneck should be calibrated so that it has *just* enough capacity to encode the amount of information that needs to be disentangled. Fig. 6.2 illustrates each of these scenarios.

Training Phase

Like a standard autoencoder, the VC network is trained to reconstruct its input using a reconstruction loss L_{rec}, which is obtained by getting the L_2 loss between the input and reconstructed spectrograms, computed as

$$L_{rec} = \frac{1}{m \times n} \sum_{j=1}^{m} \sum_{i=1}^{n} (\mathbf{X}_{ij} - \hat{\mathbf{X}}_{ij})^2, \tag{6.1}$$

where m and n are the dimensionality of the time and frequency axis of $\mathbf{X}$, and j and i are the indices of each dimensions.

An additional loss component L_{BN} calculates the L_1 loss between the bottleneck encodings for the original data $E_{VC}(\mathbf{X})$ and the reconstructed data $E_{VC}(\hat{\mathbf{X}})$:

$$L_{BN} = \frac{1}{n} \sum_{i=1}^{n} |E_{VC}(\mathbf{X})_i - E_{VC}(\hat{\mathbf{X}})_i| \tag{6.2}$$

While the original paper [Qian et al., 2019] says very little about why this loss component is included (upon personal contact, the author only stated that it was empirically found to improve convergence), it is presumed the rationale behind it is based on its ability to encourage the linguistic content of the original and reconstructed data to be the same, and also discourage the inclusion of voice identity information in a bottleneck that suffers from condition (b) in Figure 6.2. This has been described in the literature as a latent regressor loss.

The first section of D_{VC} is close to a mirrored version of E_{VC}, and reproduces the estimated spectrogram $\tilde{\mathbf{X}}$ with this architecture alone. However, Shen et al. [2018] proposed the *postnet* mechanism, which is an additional CNN that is trained to produce the residual spectral data $\mathbf{X}_{residual}$ from its input spectrogram. This residual spectral information is added to the estimated spectrogram, resulting in the final output of D_{VC} being the high-fidelity reconstructed spectrogram, $\hat{\mathbf{X}}$:

$$\hat{\mathbf{X}} = \tilde{\mathbf{X}} + \mathbf{X}_{residual}. \tag{6.3}$$

A reconstruction loss $\mathrm{L}_{rec-est}$ between the original and estimated spectrogram,

$$L_{rec-est} = \frac{1}{m \times n} \sum_{j=1}^{m} \sum_{i=1}^{n} (\mathbf{X}_{ij} - \tilde{\mathbf{X}}_{ij})^2, \tag{6.4}$$

is added to the loss function of the VC network, which was empirically shown to improve convergence. The final loss function therefore consists of a weighted combination of the two reconstruction losses and a latent regressor loss, parameterised by μ and λ (in the original work, these were simply set to a value of 1):

$$L_{total} = L_{rec} + \mu L_{rec-est} + \lambda L_{BN}. \tag{6.5}$$

Conversion Phase

After the VC network has been satisfactorily trained, VIC can be achieved by providing E_{VIE} with the spectrogram of a target speaker, $\mathbf{X}_t$, while E_{VC} is provided with the spectrogram of a source speaker, $\mathbf{X}_s$. As D_{VC} has been trained to combine disentangled VIEs with linguistic content, it will be able to process the source linguistic content $E_{VC}(\mathbf{X}_s)$ with the target VIE $E_{VIE}(\mathbf{X}_s)$ to produce the voice-converted spectrogram representation $\mathbf{X}_{s \to t}$. This process is illustrated in Figure 6.3.

Architectures and Hyper-parameters

Figure 6.4 is provided to aid readers in visualising the architecture of the VC subnetwork in AutoVC as described in this section.

E_{VC} consists of three 5x1 convolutional layers, the first layer of which is designed to accept tensors of the same size as $\mathbf{X}$. 1-dimensional convolutions are preferable for the task at hand, as they avoid the translational invariance inherent in 2-dimensional CNNs [Blaauw and Bonada, 2018]. Each of these layers has 512 channels and proceeds with batch normalisation and ReLU activation, and will be herein abbreviated as *conv-norm* layers. The output of the last conv-norm layer is passed to a stack of 2 BLSTM layers, outputting forward and backward cells of dimension 32, which in turn yields a 64 dimensional embedding when combined. To facilitate a condensed bottleneck representation, downsampling by a factor of

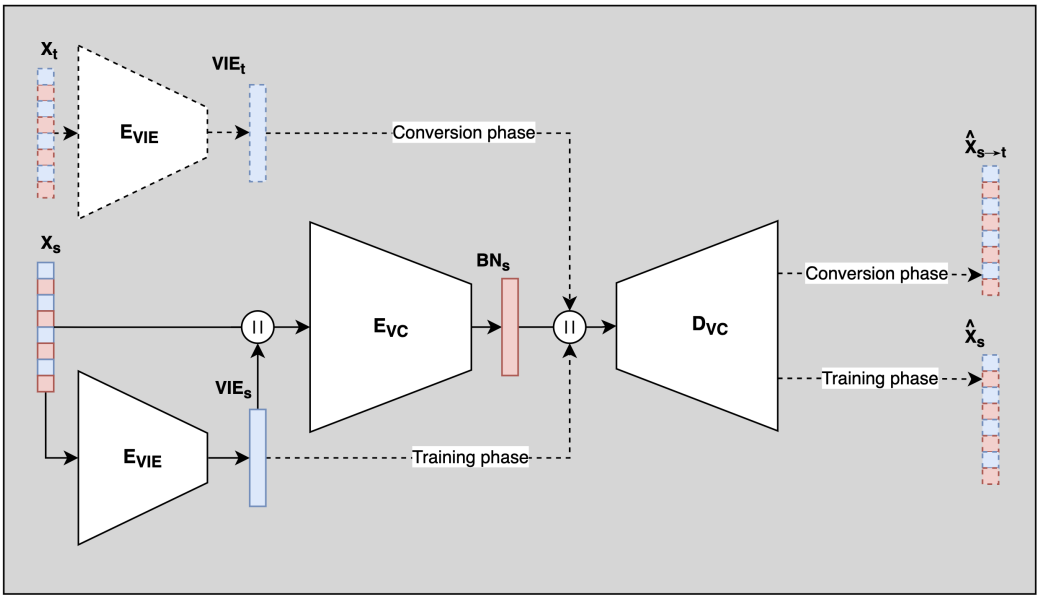


Figure 6.3: Flowchart illustrating the flow of information in AutoVC, with interchangeable pathways (indicated by dashed modules and edges) to be used for either the training or conversion phase.

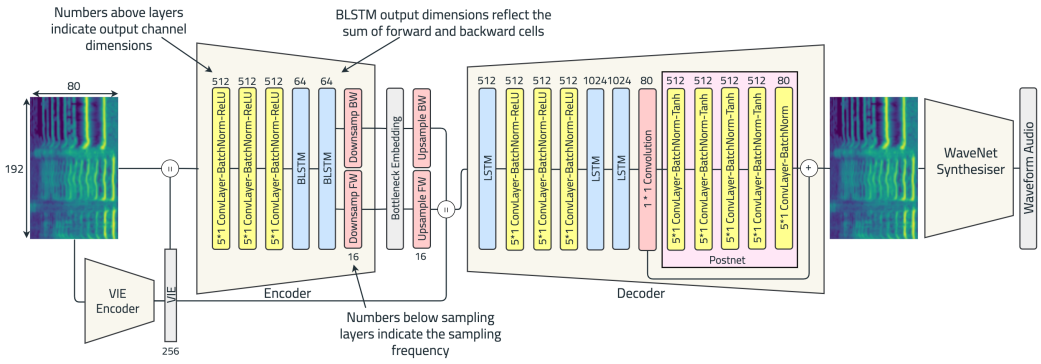


Figure 6.4: Flowchart providing an in-depth illustration of the architecture of the VC network portion of AutoVC. The numbers seen above individual layers display the dimension size of their output. Numbers below the resampling layers indicate the factor by which they are up/downsampled.

16 across the temporal axis is applied to the BLSTM outputs.

The bottleneck embeddings are reconcatenated with VIEs, after which the combined tensor is upsampled by copying, to restore the original size of the temporal axis. They are then fed to D_{VC}, which contains the following chain of layers: a single uni-directional LSTM, a sequence of 3 conv-norm layers using 5x1 kernels, a stack of 2 LSTM layers, and a 1×1 convolutional layer to project the LSTM's encodings to the target dimensionality, n. This produces the initial estimated spectrogram, $\tilde{\mathbf{X}}$, (from which the $L_{rec-est}$ loss is obtained). This representation is then fed to the postnet CNN, which consists of five more conv-norm layers. As the postnet is trained to output the residual information of the spectrogram, this is added element-wise to its input as a skip connection, the summation of which produces the high-resolution $\hat{\mathbf{X}}$ that is used to obtain the L_{rec} loss.

Waveform Synthesis

The final module in the AutoVC system converts the refined spectrogram to an audio waveform. This module consists of a WaveNet pretrained on the VCTK dataset [Veaux et al., 2017], the parameter states of which are available at an online repository provided by the first author of the AutoVC paper [Qian et al., 2019][2]. It was trained using teacher-forcing, where spectrograms were resampled to match the audio signal and used as the ground-truth conditioning features [Shen et al., 2018].

[2] `https://github.com/auspicious3000/autovc`

6.3 AutoSTC System

As this chapter explores the possibility of using the AutoVC framework for STC, the network that conditions the VC network will be pretrained to produce singing technique embeddings (STEs) rather than VIEs. A detailed description of the VIE encoder will be presented in Chapter 7, where the AutoVC system will be used for the task of singing voice conversion (SVC). The implementation of AutoVC used in the experiment described in this chapter will be referred to as *AutoSTC*. This section describes the datasets used, architecture, and training process. Objective metrics are reported for the STE encoder as a classification model as well as for AutoSTC reconstruction losses.

6.3.1 Datasets

As in Chapter 5, VocalSet was again used to train the STE encoder. It underwent the same process as described in Section 5.2.1 to obtain a class-balanced subset. However, in this experiment the *lip-trill* technique was also included, as there was no concern about how an additional class would require exponentially more time in a listening test. The subset comprised of 1182 vocal recordings, estimated to be roughly 3 hours of recorded material covering the *belt, straight, vibrato, lip trill, vocal fry* and *breathy* singing techniques. As this reduced version of VocalSet was so small, it was only split by voices into training and test subsets at a ratio of 4:1. When training the STC network, a total of 1800 training steps would be required in order to effectively cover a single epoch of data. Spectrograms were normalised to have zero mean and unit variance, which sped up convergence by approximately 50%.

Given the fact that VocalSet is already a relatively small dataset, it was important to ensure that the STC network (equivalent to the VC network portion of AutoVC as described in Section 6.2.2) was trained on additional data to facilitate more accurate modelling of the voice. Nercessian [2020]'s work on singing voice conversion highlighted the fact that AutoVC could be trained as a universal background model [Hasan and Hansen, 2011]. The same can be done with AutoSTC, and so a 20-speaker subset of the VCTK dataset [Veaux et al., 2017] the raw

singer recordings from MedleyDB [Bittner et al., 2014], and the VocalSet subset were also used for training. These datasets were chosen because they represent the voice in diverse ways. VCTK presents the voice as speech, while MedleyDB presents the voice as a studio stem with varying levels of processing applied.

The audio from each dataset was subjected to a desilencing algorithm, which determines the energy envelope of an audio signal and outputs a probability as to whether vocal activity is occurring or not (the script for which was taken with permission from the work of Sarkar et al. [2022]). A probability threshold was empirically chosen for each dataset. Sections below this threshold were discarded and the neighbouring sections were concatenated together, yielding stitched versions of audio clips that were dense with recorded vocal material. After desilencing, MedleyDB yielded approximately 3.5 hours of singing material, equating to 2189 training steps per epoch. After desilencing, the VCTK subset yielded roughly 13 hours worth of content (8,030 steps per epoch). Both datasets were split into training and test subsets by a ratio of 4:1.

6.3.2 STE Encoder

The STE encoder E_{STE} was built specifically to produce features relevant to singing techniques. This was achieved by building a singing technique classifier and using the output of its pre-classification layer as the conditioning embeddings.

Initially, classification was attempted using the network proposed for this task by [Wilkins et al., 2018] who introduced the VocalSet dataset. This architecture consisted of 1D convolution, batch-normalisation, and max-pooling layers, performing convolutions of audio waveforms. A version of VocalSet as described in the same publication was prepared from which the Wilkins model could be trained and evaluated. A random grid search was performed for hyper-parameters optimisation, which concluded that no deviation from Wilkins et al. [2018]'s suggested configurations was necessary. The model was trained ten times using different random seeds for parameter initialisation and dataset splits, from which the highest results for metrics of 0.67 precision and 0.69 recall were obtained. The similarity between these scores and those reported by Wilkins et al. [2018] confirmed the model was constructed as described and performing as expected. The model was

then trained with the VocalSet subset described in Section 6.3.1 to provide a baseline accuracy score, which yielded an average accuracy score of 60% on the test subset.

A similar architecture was explored, differing only by the convolutional kernels which were 1-dimensional while frequency bins were separated across the network's channel/filter axis. This however, achieved an underwhelming average accuracy of 30%, which in hindsight is likely due to the fact that 1D convolution networks do not facilitate pattern detection across frequency bins or translational invariance on a 2D plane.

A customised architecture was explored, the development of which was heavily influenced by the VAE system used by Luo et al. [2020b]. The intuition of combining CNN, LSTM and Dense layers was informed by Choi et al. [2017]. The final proposed architecture is shown in Figure 6.5.

It takes the same spectrogram, $\mathbf{X}$, as the STC network of AutoSTC. Each input spectrogram was split into multiple chunks of duration 0.5 seconds, which was considered to be a suitably small length of time from which a reasonable prediction could be made for singing technique classification from audio [Luo et al., 2020b]. These chunks are distributed along the batch axis to facilitate parallel processing, allowing the network to accept inputs of any length of time, rounded off to the nearest half-second. They are fed to a feature extraction network consisting of four 2D-convolutional layers of kernel size 3×3 with padding. Each of these layers is followed by batch normalisation, ReLU activations, and max-pooling. The max pooling kernel sizes were determined to gradually funnel the dimensionality towards a single output after the 4th convolution layer for each of its 512 channels. Tensors are first reshaped to remove the individual chunks from the batch axis. These are followed by two dense layers, the output of which are of dimensions 512 and 256. The output is then reshaped to allow a stack of two bi-directional LSTMs to process each chunk in its original sequential context, of which there are 6 parts each representing 0.5s of information. The BLSTM outputs are reshaped so that chunks are distributed along the batch axis once more and fed to two final dense layers of output size 64 and 256. Finally, a 6-way classification layer determines which of the 6 singing techniques is being used in the input spectrogram.

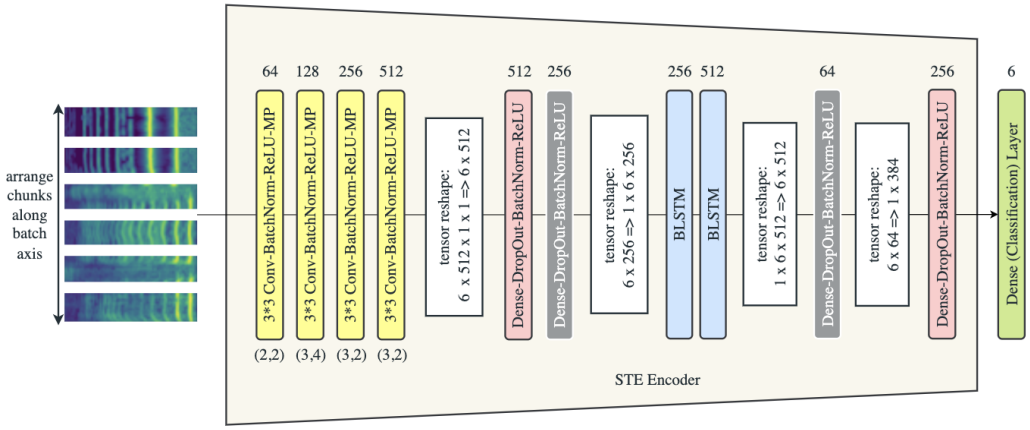


Figure 6.5: Flowchart illustrating the architecture of the STE encoder. The numbers seen above individual layers indicate the dimension size of their output, while the numbers below indicate the max-pooling kernel size. This diagram shows reshaping as if the batch size is 1, and the number of chunks is 6 (0.5s each)

A self-attention mechanism was originally included in the architectures as used in [Raffel and Ellis, 2016, Luo et al., 2020a]. However, this led to no significant improvement in accuracy or loss convergence, likely because the BLSTM layers already extracted sufficient context information.

A hyper-parameter optimisation grid search concluded that a learning rate of 0.004 and no dropout led towards the best convergence. This architecture was trained on the VocalSet subset 20 times with randomly initialised weights and dataset splits. The highest classification accuracy score on the validation subset was 86%, while its average performance was 75% after 13 epochs of training. It is worth noting that upon a post-hoc revision of this chapter's research, it was verified that removing the two dense layers (marked in grey in Figure 6.5) and the subsequent reshaping did not hinder the network's performance in any significant manner.

6.3.3 Training

STE Conditioning

In Qian et al. [2019]'s provided AutoVC implementation[3], averages of speaker embeddings were precomputed for multiple recordings of the same speaker. For STC however, these embeddings were individually generated for each example used during training, instead of preparing predetermined, averaged embeddings for each vocal technique. Unlike with speaker identity, the extent of use of most singing techniques is continuous in nature, and there are many examples where singing techniques have only been fractionally present. By providing examples with their own technique embeddings during training (as opposed to a predefined embedding determined only by the singing technique label), all variances between the established techniques are provided to the network.

The target embeddings however were always determined by averaging embeddings across the established vocal techniques. To verify the discriminative nature of these embeddings w.r.t. singing techniques, 200 vocalisations from VocalSet were randomly selected and fed to the STE encoder. The resulting embeddings were subjected to KMeans analysis, where clustering compactness was determined by obtaining the sum of squared distances (SSD) between embeddings $\mathbf{e}$ and their corresponding speaker j's centroid, C_j. Figure 6.6 displays a plot of SSDs for all values of $\{k \in \mathbb{Z} \mid 2 \leq k \leq 10\}$, where a distinct elbow can be observed for $k = 6$. Embedding clusters were therefore significantly more compact when the number of clusters was equal to the number of ground truth classes, indicating that the embeddings indeed had discriminative qualities w.r.t. types of singing techniques, and are appropriate for averaging.

Hyper-Parameters

As in the original implementation, $\mathbf{X}$ consists of 192 spectral frames of log mel-spectrums of 80 dimensions (herein simply refereed to as the spectrogram), equating to roughly 3 seconds of audio. Hyper-parameters optimisation was applied using a randomised search, which has been shown to cover search spaces more

[3] `https://github.com/auspicious3000/autovc`

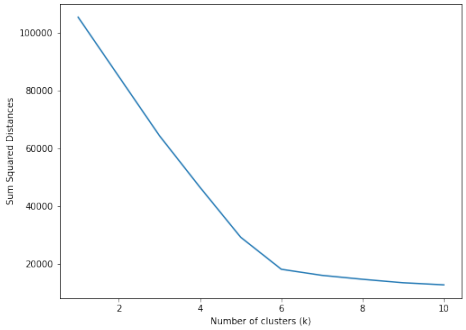


Figure 6.6: Sum-of-squared distances for STEs (y-axis), plotted across a range of k values (x-axis)

efficiently than a grid search [Bergstra and Bengio, 2012]. The search space was based on a uniform distribution, centred on the values proposed by Qian et al. [2019]. The random search variables included weight regularisation, dropout, and the learning rate, and underwent 50 iterations. After doing so, it was concluded that the original batch size and learning rate were indeed most suitable and led towards the model's best convergence w.r.t. its total loss metric. However, using an L_1 reconstruction loss instead of L_2 led towards faster convergence and better quality outputs. The L_1 loss was therefore used for the remainder of the experiment, defined as

$$L_{rec} = \frac{1}{m \times n} \sum_{j=1}^{m} \sum_{i=1}^{n} |X_{ij} - \hat{X}_{ij}| \tag{6.6}$$

Qian et al. [2019] have claimed that their inclusion of the L_{BN} loss in addition to the standard reconstruction loss helped preserve the original content and balance reconstruction quality with speaker disentanglement. However, this was found not to be the case in the preliminary runs of this experiment. The total losses for models trained over 100k training steps with and without the bottleneck encoding loss resulted in final loss values of 0.0237 and 0.0185 respectively. Output spectrograms of the model with both losses were also blurry and the audio lacked microtonal variation or vibrato, leaving a 'bubbliness' artefact in its absence. Vowels were also not consistently reproduced. These shortcomings how-

ever are less noticeable for speech than singing. This result is worth highlighting, as the latent regressor loss has been used in architectures of a similar nature [Nercessian, 2020, Jia et al., 2018, Qian et al., 2019].

Further preliminary trials were conducted to fine-tune AutoSTC's bottleneck size. The quality of reconstructed audio outputs were noted to relate only to the net size of the feature space, rather than having any exclusive correlation with the reduction of either temporal or frequency axes. The ideal downsampling factor was empirically found to be 16, resulting in a reduced temporal dimensionality of $(192 \div 16) = 12$. When the output dimensionality of the BLSTM is 32 in each direction (the combined size being 64), the resulting net size of the bottleneck becomes $12 \times 64 = 768$. Lower dimensionality representations resulted in deterioration in the reconstructed audio with artefacts of a very similar nature to those of the network trained with L_{BN}.

Sequential Datasets

All audio was converted to spectrograms to be used as input features to AutoSTC, using the process described in Section 6.2.1. These datasets were used sequentially one after another in the STC network's training schedule. This daisy-chaining of dataset training was repeated for all permutations of the three datasets while monitoring reconstruction losses. By doing this, the permutation that allowed the network to progressively learn from each dataset while minimising catastrophic forgetting (a common phenomenon in continual learning Wiewel et al. [2020]) can be determined. *Early Stopping* was employed so that AutoSTC stopped training with a dataset once the loss function w.r.t. that dataset's test subset no longer significantly improved after 50k training steps, at which point the next dataset was used for training.

Evaluation metrics

While AutoSTC was being trained on one of the three datasets, it was periodically evaluated on the VocalSet and MedleyDB validation subsets. Loss values and training cycle counts were recorded after each dataset was used. The purpose of evaluating on MedleyDB allows us to assess how well the STE encoder trained

Loss-Iteration for Vs					
Vc	**0.0653 (300k)**	Vs	0.0274 (100k)	Md	^
		Md	**0.0386 (150k)**	**Vs**	**0.0268 (50k)**
Vs	0.0347 (150k)	Vc	^	Md	-
		Md	^	Vs	-
Md	0.0500 (200k)	Vs	0.0290 (50k)	Vc	^
		Vc	^	Vc	-

Table 6.1: Table presenting losses (and training step count in parentheses) when using the VocalSet dataset for evaluation. All sequences of dataset combinations are shown, with the first, second and third dataset in the sequence being reported in the left, middle and right sections of the table. The optimum training path is highlighted in bold. Training on a dataset that leads to an increase in loss is indicated with a circumflex, at which point that path is abandoned. For space, the dataset names have been shortened as follows: VCTK:Vc, VocalSet:Vs, MedleyDB:Md.

Loss-Iteration for Md					
Vc	**0.0479 (500k)**	**Vs**	**0.0474 (150k)**	**Md**	**0.0265 (100k)**
		Md	0.0295 (150k)	Vs	^
Vs	0.0562 (150k)	Vc	0.0474 (100k)	Md	0.0301 (50k)
		Md	0.0370 (100k)	Vs	^
Md	0.0367 (150k)	Vs	^	Vc	-
		Vc	^	Vc	-

Table 6.2: Table presenting losses (and training step count in parentheses) when evaluating on the MedleyDB dataset.

on VocalSet generalises as a conditioning factor to other singing datasets. Tables 6.1 and 6.2 present these values. In addition to demonstrating which datasets led to better generalisation, it shows that the order in which datasets are fed to the network has a considerable impact on the final loss. The paths `Vc->Vs->Md` (spanning 750k training steps) and `Vc->Md->Vs` (500k steps) led to the lowest loss values for MedleyDB and VocalSet reconstruction respectively, and were used to train models that generated the examples used in our listening test presented in the next section.

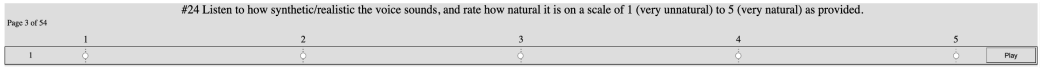
6.4 Listening Study

To assess the proposed network's performance of STC, a listening study was conducted in which the converted audio was evaluated by 19 participants, recruited in the same manner as in Section 5.2.4. The evaluation focused on determining the naturalness of singing technique-converted audio tracks, and the similarity between them and the target singing techniques. The stimuli in listening sessions were prepared with an equal number of conversions for each of the following conditions: model version, gender, source singing technique, and target singing technique.

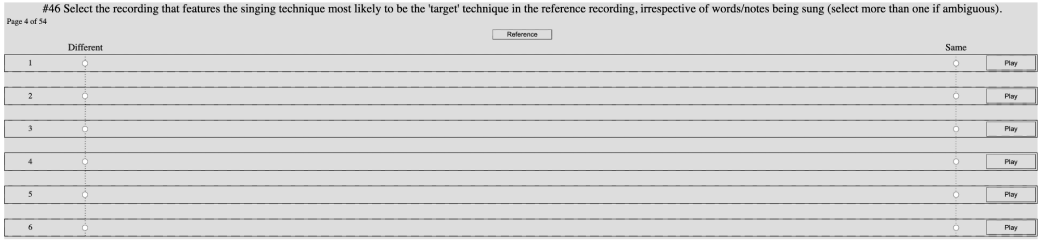
Three models were utilised in the study: Vs1, trained solely on VocalSet data; Vs2, trained on all three datasets using the optimal path w.r.t. VocalSet evaluation as shown in Table 6.1; and M1, trained on all three datasets using the optimal path w.r.t. MedleyDB evaluation as shown in Table 6.2. Vs1 and Vs2 produce converted audio originating from the VocalSet subset, while M1 converts only samples from the MedleyDB subset. The pretrained WaveNet mentioned in Section 6.2.2 was used to convert spectrograms to waveform audio formats.

6.4.1 Setup

Aspects of this listening study relating to the online web location, recruitment process, interface design toolkit (WAET) and delivery of documentation and instructions, were set up in the same manner as in Section 5.2.3. The Gold-MSI questions were omitted because it was not necessary to gather knowledge about musical aptitude for the task, and the more compressed question asking participants about their relationship with audio/music would suffice without consuming too much of the participant's time. This question, along with other questions relating to hearing impairments, device type, listening equipment, gender and age are presented in Appendix B.1. Two practice rounds were also presented, were participants were given 7 tasks from each type of question to perform.



(a) Naturalness Task



(b) Similarity Task

Figure 6.7: Interfaces used for (a) the naturalness task, and (b) the similarity task

6.4.2 Task Description

Each participant assessed 8 random examples from each model, while a balanced representation of gender and dataset subsets (for seen and unseen conversion) was maintained among the stimuli. Figure 6.7 is presented to assist readers in understanding the tasks described in this section. For evaluating naturalness, participants were asked to rate the synthetic voices on a scale from 1 (very unnatural) to 5 (very natural), similar to a MOS interface. While the MOS scale is technically by nature an ordinal scale, the descriptions accompanying each degree offer some implication that these points are evenly distributed. The vast majority of previous literature has reported the mean value of MOS results, and the word ‘mean’ is already present in its name. For these reasons, the means will be reported, although some may argue that reporting the median would be more appropriate.

In a separate task to evaluate similarity, participants were provided with a reference recording of a converted singing technique, along with 6 unlabelled candidate recordings from the same singer. The candidate recordings featured potential target techniques assigned to the reference recording. Participants were required to select the recording that they believed best represented the singing technique closest to that of the reference recording. Participants could select more than one if they felt uncertain.

As MedleyDB does not have annotations for singing techniques, candidate MedleyDB recordings showcasing each of the 6 singing techniques did not exist.

Therefore for cases where the reference recordings were converted MedleyDB examples, a singer of the same gender was randomly selected from VocalSet to represent the 6 candidate singing techniques.

Each participant completed these tasks for each of the 24 stimuli. Additionally, 6 resynthesised recordings of unconverted audio were evaluated for naturalness. These unconverted clips were obtained by converting audio waveforms into spectrograms and using WaveNet to convert them back to waveforms (without passing through the AutoSTC model). It was expected that due to WaveNet's training data being of a different domain, it would produce some artefacts when inferring from spectrograms of singer recordings, which should be accounted for. Rating unconverted clips quantifies the amount of unnaturalness caused by the conversions between spectrograms and waveforms. It allows the MOS for unconverted audio to be considered as a normalising constant factor that can be applied to naturalness, and therefore attribute the appropriate amount of naturalness resulting from the learned parameters of AutoSTC.

6.4.3 Results

Naturalness and Similarity Scores

The MOS for unconverted data was 3.75 with a standard deviation of 0.34, and is important to consider when analysing the results of the study. This highlights the fact that a considerable amount of perceived naturalness has already been lost during the wavenet resynthesis process, and that the MOS values for technique conversion should be considered with this in mind. Having said that, it should be restated that this experiment did not seek out SOTA results, but was instead investigating the difference between converted audio under different conditions.

To calculate the similarity score S for each condition, we used the formula in Equation 6.7, where P_n is a binary vector reflecting a participant's true/false predictions (identifying whether each candidate technique was the same as what was presented in the reference audio) for the nth task, C_n is a 1-hot vector reflecting the correct technique for the task, and N is the total number of tasks in the given condition. The similarity score is an average count of correct predictions weighted by the reciprocal of the number of predictions made for the corresponding task.

$$S = \frac{1}{N} \sum_{n=1}^{N} \frac{P_n . C_n}{||P_n||_1}. \tag{6.7}$$

For example, consider a participant was tasked with finding the correct target technique for a reference recording containing a *belt→vibrato* converted audio clip. If the participant chose *vibrato, belt* and *breathy*, the final similarity score for the reference recording would be $\frac{1}{3}$. If the participant incorrectly chose *vocal fry*, then the similarity score would be 0.

Figure 6.8 displays the results obtained from the listening study. These are colour-coded by the condition groups: model (blue), subset (green), gender (cyan), source technique (magenta) and target technique (yellow), where 'subset' differentiates whether the source singer was seen during AutoSTC training or not. The graphs' whiskers indicate the confidence intervals. The top graph displays MOS values for naturalness. It displays the MOS for unconverted audio in red above the 'org' label. This inclusion inherently normalises the graph scale which is intuitive as 'org' MOS represents the upper bound ceiling score. The lower graph displays

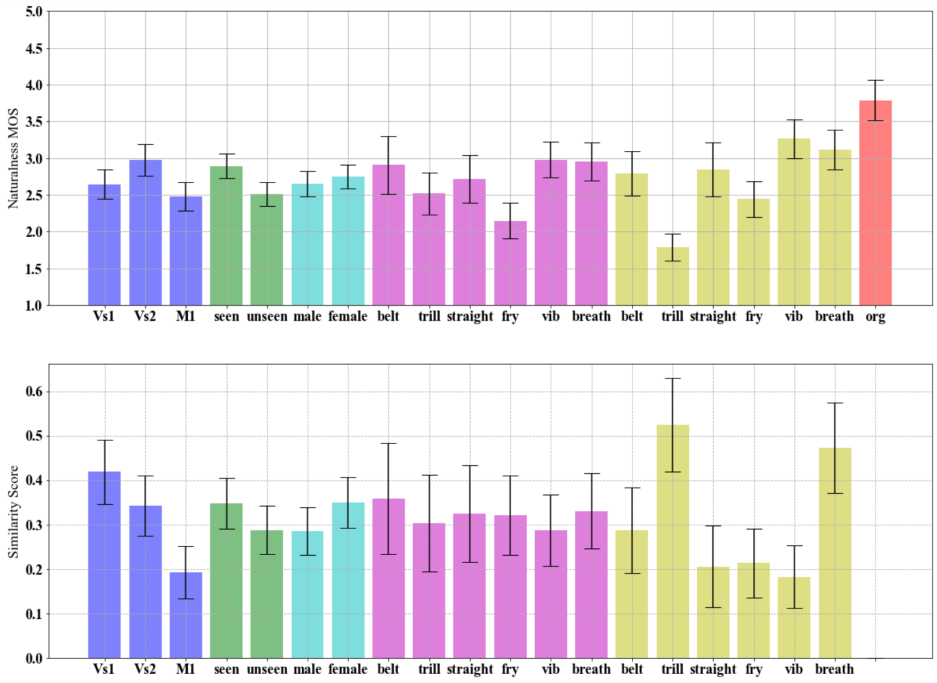


Figure 6.8: Bar graphs displaying listening test scores. Condition groups are colour-coded together from left to right as: models, subsets, genders, source techniques and target techniques. **Top:** Naturalness (MOS values and confidence intervals) for all conditions. **Bottom:** Similarity scores as determined by Equation 6.7.

similarity scores. The combination of these two graphs provides insight into how each of the models performs, and what conditions influence the naturalness and similarity of the converted singing. The majority of similarity scores across all conditions were higher than the probability of random guessing, which is 0.167. From a Spearman's rank analysis, it was observed that naturalness and similarity scores exhibited no significant correlation.

Discussion

Samples of naturalness scores of each condition were tested against other samples of the same condition for statistically significant differences using a Mann-Whitney U test, after the samples failed the normalised distribution test described in Section 5.3. The same was done with samples of similarity scores. Tables 6.3

and 6.4 present statistically significant differences detected among samples for naturalness and similarity ratings, respectively.

Condition Group	**Group 1**	**Group 2**	**U**	***p***
Model	Vs2	Vs1	13088	< 0.04
Model	Vs2	M1	13884	< 0.002
subset	seen	unseen	30362	< 0.001
source technique	fry	belt	658	< 0.018
source technique	fry	trill	1209	< 0.03
source technique	fry	straight	1750	< 0.02
source technique	fry	vibrato	2932	< 0.001
source technique	fry	breathy	2864	< 0.001
target technique	belt	trill	4123	< 0.001
target technique	belt	fry	3421	< 0.04
target technique	belt	vib	2323	< 0.03
target technique	trill	breathy	1114	< 0.001
target technique	trill	straight	915	< 0.001
target technique	trill	fry	2164	< 0.001
target technique	trill	vib	1201	< 0.001
target technique	fry	vib	2090	< 0.001
target technique	fry	breathy	1830	< 0.001

Table 6.3: Mann Whitney U results for significant differences between sample groups relating to naturalness.

Condition Group	Group 1	Group 2	U	*p*
Model	M1	Vs1	8378	< 0.001
Model	M1	Vs2	9143	< 0.001
target t	trill	belt	3595	< 0.01
target t	trill	straight	2247	< 0.001
target t	trill	fry	4353	< 0.001
target t	trill	vib	4503	< 0.001
target t	breathy	belt	3541	< 0.01
target t	breathy	straight	2261	< 0.001
target t	breathy	fry	3786	< 0.001
target t	breathy	vib	3899	< 0.001

Table 6.4: Mann Whitney U results for significant differences between sample groups relating to similarity.

The Vs2 sample was significantly higher than Vs1 and M1. This suggests that providing the network with multiple datasets improves its ability to synthesise more natural-sounding data, but suggests that AutoSTC does not generalise well to unseen datasets of different distributions. The source technique *fry* led to significantly lower naturalness ratings. This is likely because there may be large acoustic variances among singers' execution of the *fry* technique. The target technique *trill* received the lowest score and was statistically lower than all other target condition samples, It is possible that the acoustic features for this technique require high-fidelity synthesis due to its salient features residing in high frequency content. The target technique condition *vibrato* scored the highest, and was statistically higher than all other samples, except that of *breathy*. This may be because the network is making very subtle changes when synthesising the *vibrato* technique that do not relate to frequency modulation, resulting in fewer audio artefacts (this will be expanded upon in the last paragraph of this section). It could, however, also be the case that participants perceive the singing voice as more natural when *vibrato* is present.

The inclusion of all datasets in training Vs2 seemed to diminish its ability to accurately convert techniques, although the similarity samples for this condition were not statistically significant from Vs1. The M1 model was statistically worse than both other models, indicating that the features learned to generate technique embeddings from the STE encoder were not generalisable to data outside

the dataset it was trained on. No statistically significant difference was found between the condition groups for gender, subset or source technique groups.

When the target techniques were *trill* and *breathy*, this lead to the highest scores, and samples from these groups were statistically higher than all other groups apart from each other. This suggests that the conversions to a *trill* technique may be easily identifiable, even if its not very natural. The model also seems do comparatively well at converting to *breathy* techniques, possibly because the acoustics of air turbulence are easier for the model to synthesise, or present a clear aural cue for this type of phonation.

Notably, when the target technique was *vibrato*, models scored the lowest for similarity (although this was only statistically significant when compared to the *trill* and *breathy* techniques). This suggesting that synthesising this technique was particularly challenging. This difficulty can be attributed to the fact that many *non-vibrato* examples in VocalSet also feature a substantial amount of frequency modulation (the fundamental feature of *vibrato*), implying the multi-class nature of the data, despite the fact that the ground truth labels only feature a single class per recording. As a result, disentangling this confounding feature from other techniques was likely a challenge for AutoSTC, and it would have instead focused on other features associated with *vibrato* (such as the phonation mode), which would be less apparent to listeners than the presence of frequency modulation.

6.5 Conclusion

In this experiment, a network for vocal technique classification has been proposed, achieving an average score of 75% accuracy on VocalSet data. The design of this network made it possible to re-purpose the influential AutoVC network for the task of STC, making it the first network to perform zero-shot conversion on singing techniques at the time of publication. Preliminary experiments demonstrated that including the L_{BN} loss (Eq. 6.2) significantly slowed down convergence w.r.t. reconstruction loss and led to significantly worse audio output. It has been demonstrated that the order in which the model is trained on datasets makes a difference to how well it is primed to perform on the final dataset, and thereby improves its ability to reconstruct input spectrograms.

However, it has been observed from the results of the listening study that improvements to reconstructed data does not necessarily imply that the performance of STC will also improve. As the inclusion of multiple datasets seemed to diminish the model's performance in STC, the conclusion can be drawn that features generated by the STE encoder were not robust enough to generalise to recordings from other datasets.

6.5.1 Future Work

As the STE encoder was trained via supervised learning using the VocalSet data, it is reasonable to deduce that the contents of VocalSet are too restrictive in their representation of the singing voice to be generalisable. Additionally, it is considered that the presence of frequency modulation in other techniques in VocalSet may have influenced the STE encoder to give less importance to this feature w.r.t. encoding *vibrato*. During the developmental stage, there were some limited cases where AutoSTC successfully converted to a *vibrato* technique with frequency modulation, suggesting that AutoSTC is capable of converting singing technique features of a temporal dependency. To train a robust STC model, however, would require datasets with multi-class labels.

The use of augmentation techniques when training the STE may potentially improve the generalisation of the network for unseen data. Applying the GE2E

loss [Wan et al., 2018] to the STE to fine-tune its output embeddings is an alternative unexplored method. However, due to VocalSet's shortcomings, it seems clear that using this dataset is not an optimal solution to achieve STC.

In addition, it may be worth investigating how AutoSTC performs when conditioned on more attributes such as speaker identity, pitch contour, and vowel sound. Considering STC as a similar task to VC and the availability of considerably larger datasets for speech, it may also worthwhile to explore the effects of pretraining the model for VC using a VIE encoder, before switching to the STE encoder and training it for STC.

In future work, alternative options to the speech-trained wavenet vocoder will be considered, as it has introduced artefacts to the audio and proven to induce lower MOS ratings on converted resynthesised audio. Recent alternatives have been proposed in Section 4.4.5.

Chapter 7

Singing Voice Identity Embedding and Conversion

7.1 Introduction

7.1.1 Motivation

In the previous chapter, an STE encoder was the conditioning factor to an STC network. The converted results were generated using the VocalSet evaluation set, and were only partially convincing. It was deduced that the dataset the encoder was trained on, VocalSet, represented singing techniques in a way that did not generalise to unseen data. This was concluded when STC was performed on audio from external datasets using an STE encoder pretrained on VocalSet was considerably less successful than within-dataset inferences. It therefore seems appropriate to consider alternative methods of STC that do not rely on specifically labelled data and can generalise to the majority of singer recordings.

One method of approaching vocal attribute disentanglement for which labelled data is scarce, is to exhaustively disentangle all information for which labelled data is already available. This would leave residual information encoded from vocal input signals relating to attributes such as singing techniques and other information that is unaccounted for by labelled data [Hsu et al., 2019]. These residual encodings, if captured as probability statistics in a VAE system, could then be manipulated to perform more realistic conversions than those achieved in Chapter

6. Attributes that are strongly labelled or can be deterministically inferred include pitch, loudness and phonetics. However, the most frequently sought-after vocal attribute for conversion is voice identity. While this is often considered a global attribute for each vocalist, it is more elusive in that it requires descriptive embeddings that heavily rely on timbral qualities which are dynamic in nature.

7.1.2 Chapter Summary

As a step towards unlabelled singing voice attribute conversion (SVAC) by exhaustive disentanglement, this chapter focuses on the process of VIC (first described in Section 4.4.4), ensuring that the established conversion systems are also appropriate for SVIC. It repurposes AutoVC for the task of SVIC, which as done in the previous chapter, can be separated into two parts: the SVIC network (the autoencoder) and the pretrained VIE encoder. This model will herein be referred to as AutoSVIC, reflecting the purpose of its current implementation, and will be discussed in detail over two separate sections.

The first section compares how a VIE encoder performs when trained on different features. After establishing the ideal feature set, it proceeds by investigating the difference in performance between same and cross-domain inference. Often in literature, the concept of transfer learning is used to justify cases where a model trained on one domain can be used on another related domain, such as using a VIE encoder pretrained on speech to infer from singing data [Nercessian, 2020, Demirel, 2022, Polyak et al., 2020, Toda et al., 2023]. There is not a lot of research that attempts to measure how similar the speech and singing domains are, and the potential differences in performance between same and cross-domain applications are seldom measured.

The second section describes the process of SVIC, where an AutoSVIC model is used to investigate the effects of using pretrained VIE encoders for same and cross-domain applications. It also challenges the justification of including certain loss components in the objective function and offers alternatives. Objective metrics and subjective human evaluations were collected w.r.t the audio outputs of models trained on these different loss functions to deduce how they affected the model and audio.

Audio examples relating to this chapter can be heard in the accompanying online repository[1].

7.2 VIE Experiments

The VIE encoder is a NN designed to encode spectrograms finite embedding space that represents the voice. As described in Section 6.2.2, it is trained to ensure that its output embeddings represent variances of the input signals that are most relevant towards the identity of the vocalist. The VIE encoder is a fundamental component in the architectures of many VIC models, used to provide descriptive representations (unlike one-hot encodings) of voices that can be used for attribute disentanglement.

This section covers a set of experiments focused on the VIE encoder, which will later be applied as a conditioning factor to the SVIC network (the autoencoder) in Section 7.3, the combination of which amounts to the AutoSVIC system. Feature sets are explored, optimised and tested in the speech and singing domain to determine which ones lead to the encoder's best performance. The best features are then used to investigate how well the encoder performs cross-domain inference in comparison to same-domain inference. In this chapter, 'vocalist' will be used as an umbrella term that can refer to either speakers or singers, depending on the context.

The VIE encoder's generalised end-to-end (GE2E) loss function [Wan et al., 2018] facilitates self-supervised training by contrastive learning, encouraging the model's output embeddings to cluster together in latent space when they originate from the same vocalist. Details on how this is achieved are provided in Section 4.4.1.

[1] `https://trebolium.github.io/vc_for_singing`

7.2.1 Input Features

DAMP Intonation Dataset

The performances of VIE encoders used under the various feature conditions described in this section were trained and evaluated on the DAMP Intonation (DI) Dataset [Wager et al., 2019]. This dataset is a collection of 4702 recordings (consisting of 3556 singers and 474 songs) of singers with strong intonation, extracted from the parent repository, the Digital Archive of Musical Performances (DAMP)[2], which is available to the public by request. These recordings were originally collected by the karaoke application, Smule[3], primarily consisting of amateur singers recorded with amateur equipment, such as smartphones. This means there is inherently a considerable amount of background noise such as faintly heard backing tracks, ambience and other miscellaneous sound events. It also consists of many non-singing segments due to singers waiting for their backing tracks' instrumental sections to conclude. The contents of the Intonation dataset were selected from the DAMP dataset by its authors [Wager et al., 2019] based on semi-supervised methods using MIR-related features. For this experiment, silences of the DI dataset were removed using the same desilencing process described in Section 6.3.1.

WORLD Spectral Envelope Generation

Since its publication in 2016, the WORLD vocoder's features [Morise et al., 2016] have been frequently used for voice-modelling tasks across the literature. As previous voice conversion literature primarily makes use of either mel-spectrograms or features originating from the WORLD spectral envelopes, WORLD features were selected as a contending feature input to the VIE encoder. A Python wrapper for WORLD[4] was used to generate these features. However, there is a considerable amount of inconsistency regarding how these features are generated, and little explanation for the specifications used.

[2] `https://ccrma.stanford.edu/damp/`

[3] `https://www.smule.com`

[4] `https://github.com/JeremyCCHsu/Python-Wrapper-for-World-Vocoder`, v 0.3.2

In an attempt to rectify this, multiple parameter adjustments were made, deviating from their default settings, to determine whether any of these changes had a significant effect on the task of generating well-defined VIEs. The adjusted parameters included: expanding the F0 range from 71-800Hz to 50-1100Hz, based on the extended frequency range of singers' capabilities [Hess, 2012]; switching between the two pitch estimation algorithms, 'Dio' and 'Harvest'; and frame duration between 5ms and 10ms, based on what has been used in previous literature.

The harmonic output features of WORLD were then reduced to 80 dimensions (and where applicable, the aperiodic features were reduced to 4 dimensions), via the dimensionality reduction process outlined in Section 4.3.5[5], yielding a compressed 80-dimensional version of a harmonic spectral envelope, computed for human perception in the form of log MFSCs. Mel-cepstral coefficients (MCCs) have also been predominantly been used in previous literature for voice-modelling tasks. WORLD's original output spectral envelope, MFSCs, and MCCs were tested as input features to the VIE encoder in chunks of 128 successive frames, representing a total duration of approximately 2 seconds of audio.

Additional WORLD Features

The effect of including WORLD's aperiodic information in the feature input was also investigated. If effective, a post hoc comparative test with a clean dataset without noisy recordings, like VocalSet, would be necessary to determine whether the aperiodic information contains voice-specific information, or whether the GE2E loss improves due to cues in the background noise informing the encoder which recording (and therefore voice) is which.

Considering the differences found in the perceptual maps in Section 5.3.4 between low and high registers, as well as Sundberg [1977]'s report on pitch registers affecting the voice organs, the inclusion of pitch information was also investigated to determine whether it would lead to more efficient convergence. Pitch information was first obtained from the WORLD vocoder, which comes in a tuple format of F0 frequency floats and voiced/unvoiced booleans. The F0 data was converted

[5]The implementation of WORLD feature post-processing by the authors of `https://github.com/MTG/WGANSing` was used to transform the spectral envelope.

F0 Range	Pitch Algorithm	Pitch	Aperiodicity	Frame duration	Post-processing
50-1100Hz	Dio	False	False	5ms	MFSCs
71-800Hz	Dio	False	False	5ms	MFSCs
50-1100Hz	**Harvest**	False	False	5ms	MFSCs
50-1100Hz	Dio	**True**	False	5ms	MFSCs
50-1100Hz	Dio	False	**True**	5ms	MFSCs
50-1100Hz	Dio	False	False	**10ms**	MFSCs
50-1100Hz	Dio	False	False	5ms	**MCC**
50-1100Hz	Dio	False	False	5ms	**None**

Table 7.1: WORLD parameter configuration comparison. Entries for Pitch and Aperiodicity columns indicate whether these were included in the input features, while text in bold highlights the change being tested from the baseline (top row) configuration.

to a MIDI one-hot encoding format. This ranged from 31 (note G1) to 84 (C6), reflecting the singers' extended range of frequencies. An additional dimension was added for unvoiced segments, giving a total dimensionality of $55 \times 128 = 7040$ (pitch $\times$ timesteps).

Table 7.1 presents a comprehensive display of what configurations were changed when engineering WORLD features to be used as input features to the VIE encoder.

Results

Figure 7.1 shows the resulting loss contours for VIE encoders when trained on each variation of the aforementioned features, where the legend's labels indicate which parameter was changed. It is clear that WORLD's unprocessed spectral envelope led to the worst results. MCCs performed significantly better. However, MFSC-based features performed best, and all variations of these features (except for frame duration adaptation) had no effect on either the speed of convergence or minimal loss values. It can be concluded from this result that: the pitch range 71-800Hz (WORLD's default setting) is enough to capture the frequency range of singing voices in this dataset that is relevant to voice identity; WORLD's different pitch estimation algorithms have a negligible effect on spectral representation; the

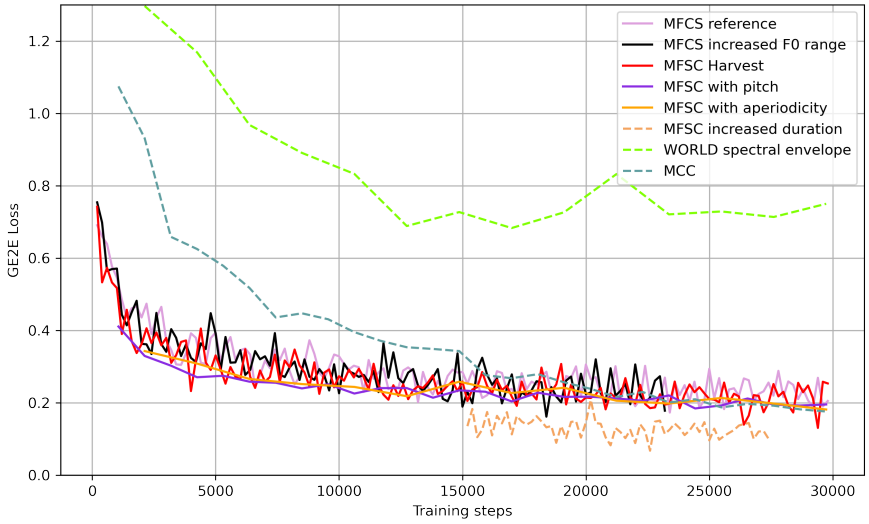


Figure 7.1: GE2E loss contours for VIE encoders using different variations of WORLD feature. Legend indicates which parameter of the WORLD generation process was changed. The contour for 'frame duration' only covers training steps from 12500 to 27500 due to lost data, but still conveys the drop in loss caused by this adaptation.

behaviour of aperiodic features is not unique to each singer; and pitch information was not useful for discriminative VIE generation. The latter result was particularly surprising, as it was expected that providing some register-based information would imply some expectancy of how the vocal organs' configurations might affect the timbral output. Accepting the null hypothesis, however, suggests that this effect is negligible or indistinguishable among vocalists.

While doubling the frame duration significantly reduced the GE2E loss, this adjustment effectively doubled the length of the input window and was retrospectively disregarded as a meaningful configuration manipulation. This is because as the temporal context of the input increases, the corresponding averaged embeddings will be inherently closer to the centroid of the cluster for a given vocalist, which leads towards a reduction in GE2E loss values which is not related to an improvement in VIE generation.

7.2.2 WORLD versus Mel-Spectrogram Features

Having concluded that WORLD's MFSC-based features of any variation led to the VIE encoder's best GE2E loss contours, this variation of WORLD features using WORLD's default settings as seen in the first entry of Table 7.1 was used (herein simply referred to as WORLD features). These features could then be compared against mel-spectrogram features for the task of VIE generation. Both feature types were generated from data represented at a sampling rate of 16kHz, using an FFT frame length of size 1024 with a hop length of 256 samples (frame duration of 16ms). Each feature set was reduced to a dimensionality of 80 and was again fed in chunks of 128 timesteps to the VIE encoder. Detailed descriptions for how each of these feature types is generated can be found in Section 4.3. The DI and VCTK datasets, representing singing and speech data respectively, were used to see how the use of each of these features affected VIE generation in the context of singing and spoken domains.

Results

Figure 7.2 shows the GE2E loss contours of the VIE encoder when trained on each dataset with each set of features, amounting to four training sessions in total.

The WORLD loss contours (dashed lines) begin with a considerably lower loss value than the mel-spec loss contours, suggesting that the reduction of information in the spectral envelope is an effective approximation towards the concept of VIEs. However, after approximately 30k training steps, the mel-spectrogram contours cross over the WORLD contours, suggesting that while spectral envelopes are suitable approximations, the spectral detail of mel-spectrograms provides additional information that correlates with the identity of the vocalist. This behaviour between mel-spectrograms and WORLD features is the same for both the speech and singing datasets and so it can be concluded that mel-spectrograms are the superior feature representations for such tasks.

Pitch features (generated in the same manner as described in Section 7.2.1) were concatenated with mel features in a separate training session in an attempt to speed up convergence. However, no noticeable difference was observed between this model and one trained with only mel-spectrogram features.

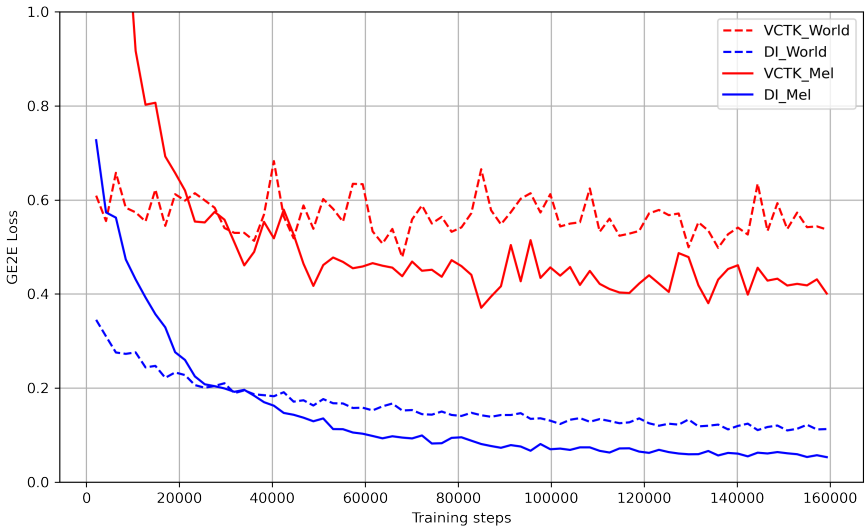


Figure 7.2: GE2E loss contours for VIE encoders over 160k training steps, trained on the DI (blue) and VCTK (red) when trained on mel (solid lines) and World (dashed lines) features.

7.2.3 Domain Task Comparisons

This section describes the process of generating datasets, and training and evaluating VIE encoders in the context of same and cross-domain inference.

Dataset Curation

As the most suitable set of features for VIE encoding have been established, an investigation into how VIE encoders perform cross-domain tasks between speech and singing could now be conducted using these features. Customised subsets of the LibriSpeech [Panayotov et al., 2015] and DI datasets were used to separately train two VIE encoders on either speech or singing.

A subset of the LibriSpeech dataset, labelled as 'other', contains noisy audio recordings of a similar nature to the DI dataset. Preliminary auditions of the LibriSpeech subset confirmed that it consisted of amateur recording techniques with imbalanced frequency responses and background noise that are comparable with DI's audio quality (for this reason, LibriSpeech was chosen over VCTK to rep-

resent speech in this experiment). It was subjected to the same aforementioned loudness detection algorithm as the DI dataset. It was then curated so that the durations of content per vocalist were the same as those of the DI dataset. A subset of the DI dataset was also generated, so that it contained the same number of vocalists as the LibriSpeech subset. Both subsets contained 1233 vocalists and an average of 166 seconds of content per vocalist. These subsets are herein referred to as DI_1.2k and LS_1.2k. Furthermore, these were randomly split by vocalist into train and validation subsets by a ratio of 8:2.

As the intention of this experiment was to compare how well the two trained VIE encoders generalised to cross-domain and same-domain data, it would be biased to compare inferences from a dataset that either encoder was trained on. For example, it would not be particularly useful to compare the loss values of both encoders on the DI dataset, while one of them was already pretrained on data from this same distribution.

A third dataset was therefore chosen for evaluation, allowing for a more robust claim of generalisation between all domain comparisons. The NUS dataset [Duan et al., 2013] was an ideal choice for dual-domain comparisons, as it conveniently contains 12 vocalists, both singing and speaking the same text in different partitions, making it well-balanced in content. This was also subjected to the same desilencing process as previously described. Each partition was independently used to evaluate the VIE encoder for same-domain and cross-domain inference. The average GE2E losses for each model were obtained to determine how effectively they produced vocalist-specific VIEs.

Imbalanced Dataset Comparison

It is well known that speech datasets are of many orders larger than singing datasets. It is therefore also relevant to investigate whether enough information can be extracted from very large speech datasets in order to generalise to singing data. The VIE encoder used was supplied by [Qian et al., 2019][6] and pretrained for 1 million training steps on the entirety of LibriSpeech and VoxCeleb1 [Nagrani et al., 2017] (herein abbreviated together as LSVC), the latter of which contains

[6]The trained model's parameters were provided at `https://github.com/auspicious3000/autovc`

noisy, untreated spoken audio segments extracted from YouTube clips. It has therefore been trained on roughly 22k hours of diverse spoken content (roughly 500 times as much data as seen in the DI_1.2k pretrained encoder), giving it an advantage of potentially learning more information that may lead towards better generalisation for singing voice inference. This encoder is also included in our NUS evaluations.

GE2E Losses

Figure 7.3 displays the GE2E loss contours of models trained on DI_1.2k and LS_1.2k. During training, both models stopped converging at a GE2E loss value of approximately 0.094, within the 90kth iteration period (although models remained training until 100k training steps were completed). Both contours display similar bumps when reaching loss values of 0.6 and 0.4, suggesting some structural similarities in data encodings between the singing and speech domains. It can also be seen that the encoder trained on singing initially converges slightly faster.

Pretraining a network on one domain has been known to prime it with priors that are partially generalisable to another related domain, giving it a head start in comparison to a network that is initialised with random weights. This is illustrated via the solid pink contour in Figure 7.3, which represents the GE2E loss of an encoder that was trained on the DI_1.2k dataset, but was pretrained on LS_1.2k. This model only requires approximately one-tenth the number of training steps as an encoder without pretraining to converge. This encoder has benefited from seeing both datasets, allowing it to achieve the lowest recorded loss of 0.076.

GE2E losses generated from NUS-based inferences are presented in Table 7.2, with a breakdown of the differences in loss when comparing cross-domain to same-domain applications. An increase in the GE2E losses is evident when comparing cross-domain to same-domain inferences. This gap is significantly smaller for the encoder pretrained on singing. This suggests that using a speech-pretrained encoder on singing data for VIE generation would be less successful than the inverted scenario of using a singing-pretrained network on speech data. Interestingly, the loss gap between same and cross-domain inferences for the

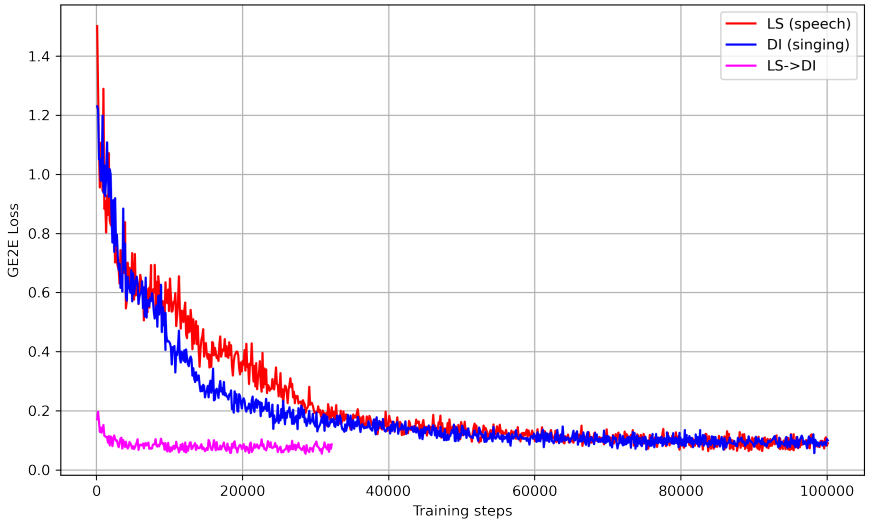


Figure 7.3: GE2E contours displaying VIE encoders separately trained on LS_1.2k (red) and DI_1.2k (blue) datasets. The pink contour is based on an encoder that was pretrained on LS_1.2k and continued training on the DI_1.2k.

LSVC-pretrained encoder is slightly larger than that obtained from the LS_1.2k-pretrained encoder. This suggests that the provision of more speech data further fined-tuned the LSVC-pretrained encoder for speech, and consequentially led towards a larger loss when inferring from singing data.

Training Data	**Evaluation Data**	**GE2E Loss**	**Loss Difference**
DI_1.2k (singing)	NUS (singing)	0.52	0.09
DI_1.2k (singing)	NUS (speech)	0.61	
LS_1.2k (speech)	NUS (speech)	0.33	0.42
LS_1.2k (speech)	NUS (singing)	0.75	
LSVC (speech)	*NUS (speech)*	*0.42*	*0.45*
LSVC (speech)	*NUS (singing)*	*0.87*	

Table 7.2: Presents GE2E losses for multiple VIE encoders, pretrained on either the LS1.2k or DI_1.2k and evaluated on NUS speech or singing subsets. The 'Loss Difference' column displays the increase in GE2E loss incurred when the encoder goes from same to cross-domain inference.

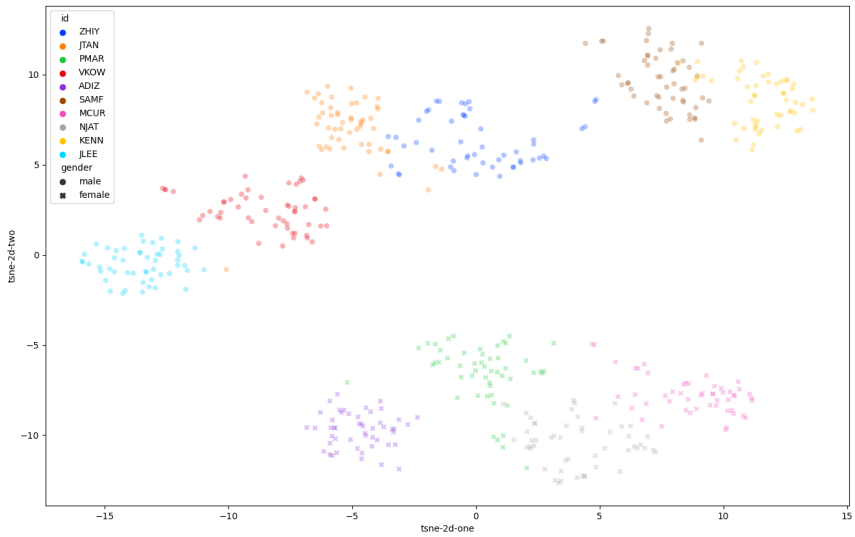
Embedding Visualisations

Differences in GE2E loss metrics alone do not provide much intuition about how exclusively these embeddings describe their respective vocalist, which in latent space would be reflected by compact clusters of same-vocalist embeddings. It is also unclear how significant these differences are for downstream tasks. To visualise this, the t-SNE algorithm was applied to VIE encoder outputs generated from the NUS dataset. This allows each embedding to be represented as a point on a 2-dimensional plane. The distance between points indicates how different the embeddings are from one another. This allows for a visualisation of how compact singer embedding clusters are. The results of this process are shown for encoders pretrained on the LS_1.2k (Figure 7.4), DI_1.2k (Figure 7.5) and LSVC datasets (Figure 7.6) when inferring from the NUS speech data, as well as singing data. Male and female embeddings are represented by circles and 'x's respectively.

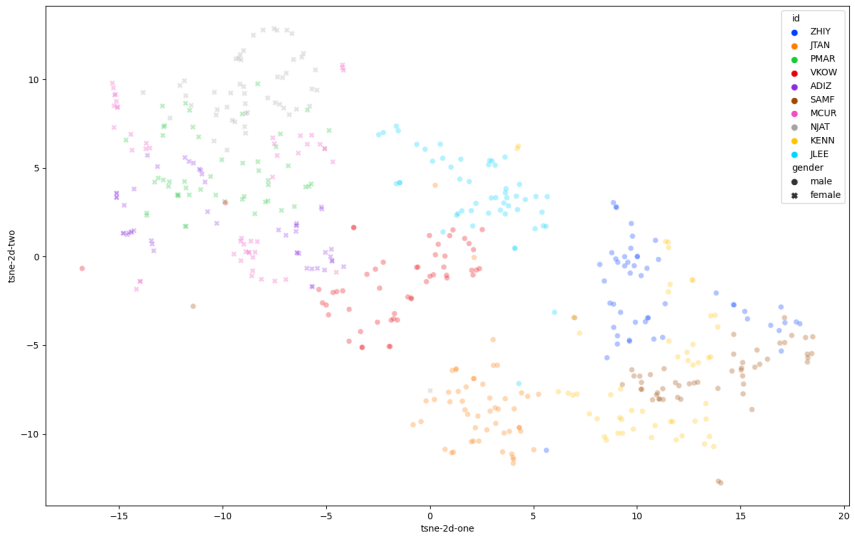
All encoders for each domain produce clusters of vocalist-specific VIEs. VIE clusters exhibit more separation in same-domain cases than in cross-domain cases. While Figure 7.6 suggests that the LSVC-pretrained encoder produces the strongest clustering cohesion and separation seen among these maps, which was for same-domain speech inferences, the difference between same and cross-domain inferences are generally similar between the LS_1.2k and LSVC pretrained encoders. These observations are in line with the results shown in Table 7.2, where the loss difference between the LSVC and LS_1.2k was very small.

A segregation between the genders can be seen for speech domain t-SNE maps, while singing domain maps do not exhibit separation to the same magnitude. It is hypothesised that this is due to the fact that in the case of speakers, the F0 range has smaller variances and larger differences in mean between the genders. The same gap is less pronounced, with wider variances in singing data. Genders were not explicitly given as part of the NUS dataset, and so they were inferred upon aural examination by the author to provide some insight[7]. Any other patterns of clustering behaviour among the t-SNE plots are too ambiguous to comment on.

[7]While subjective gender approximation is currently a delicate topic, it should be clear that this was only done as a last resort due to lack of data, and provides some general insight into how the models encode the aural qualities that most frequently succeed in differentiating the two genders.

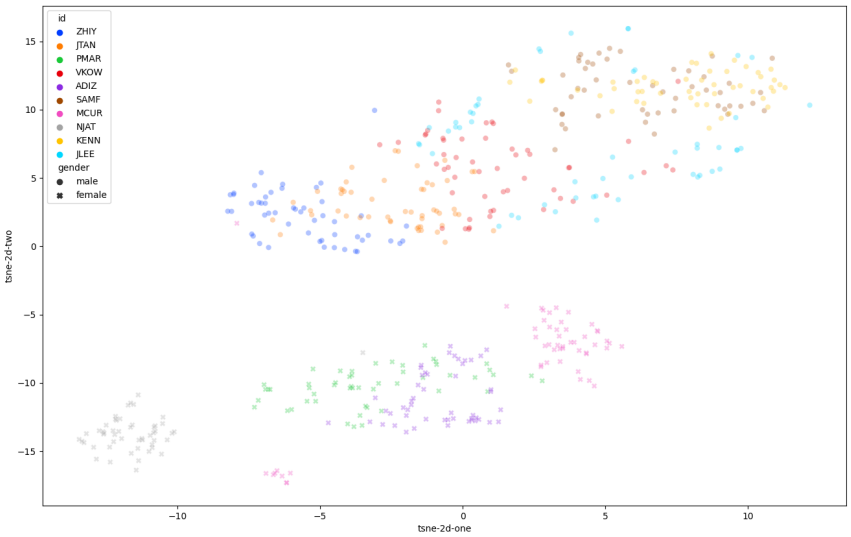


(a) Speaker embeddings generated from the LS_1.2k-trained encoder

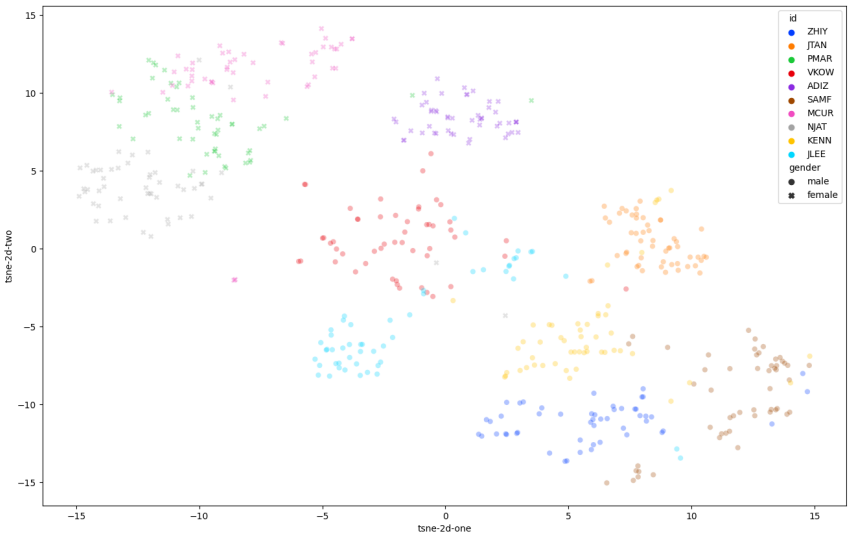


(b) Singer embeddings generated from the LS_1.2k-trained encoder

Figure 7.4: t-SNE generated maps of VIEs representing (a) speech and (b) singing clips from the NUS dataset, generated from encoders trained on the **LS_1.2k (speech)** data.

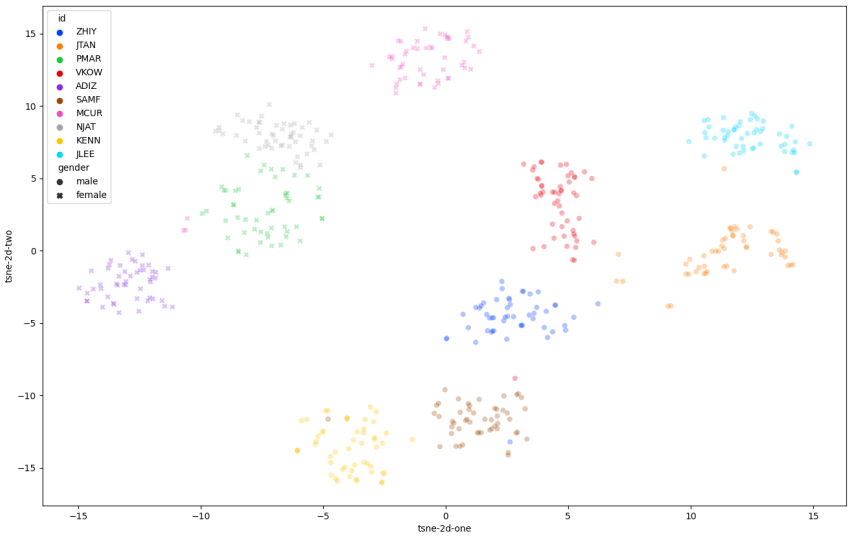


(a) Speaker embeddings generated from the DI_1.2k-trained encoder

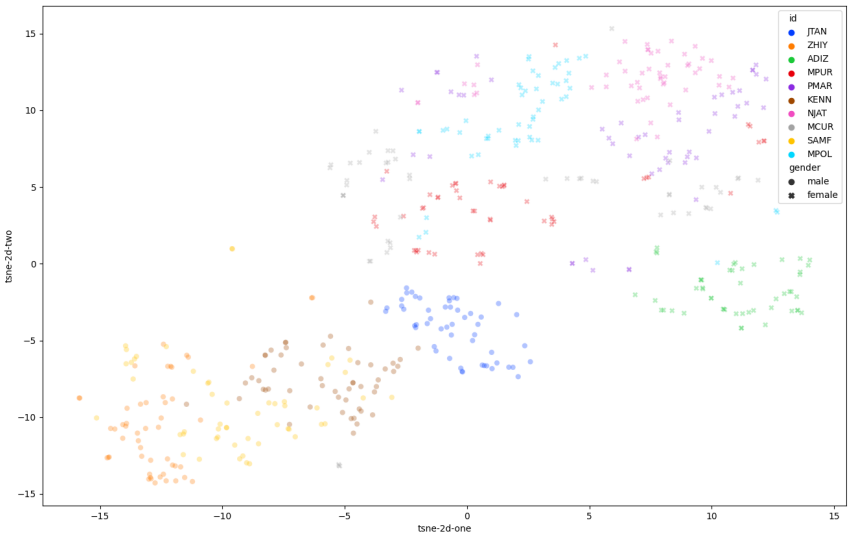


(b) Singer embeddings generated from the DI_1.2k-trained encoder

Figure 7.5: t-SNE generated maps of VIEs representing (a) speech and (b) singing clips from the NUS dataset, generated from encoders trained on the **DI_1.2k (singing)** data.



(a) Speaker embeddings generated from the LSVC-trained VIE



(b) Singer embeddings generated from the LSVC-trained VIE

Figure 7.6: t-SNE generated maps of VIEs representing (a) speech and (b) singing clips from the NUS dataset, generated from encoders trained on **LibriSpeech and VoxCeleb1** data.

Differences and similarities between balanced and imbalanced dataset comparisons have been measured and illustrated. As a matter of relevant interest to the field, the implications of the differences between speech and singing-trained encoders on SVIC will be examined in the proceeding section, along with other conditions devised from separate motivations.

7.3 Singing Voice Conversion Task

This section describes experiments using the AutoSVIC network to facilitate the task of SVIC. The architecture (excluding the conditioning encoder) is the same as that of the AutoSTC model used in Chapter 6. As AutoSVIC makes use of VIE encoders, it inherently is trained to disentangle voice identity information from the input signal. If this is done correctly, the bottleneck should have the capacity to encode only non-identifying information such as pitch infrmation, singing techniques, phonetics and other variances that are unaccounted for. In the speech domain, there is some dispute over whether accent is included in timbral perception. However, this is usually outside the scope of SVIC research, likely due to very little accent diversity within datasets.

In this section, trained models are compared against one another to determine how they are affected by several conditions which include using VIE encoders trained on different domain data, and different loss functions. SVIC and resynthesis are achieved in the same manner as described in Section 6.2.2: during the inference phase, the pretrained encoder receives target singer data and the same WaveNet vocoder described in Section 6.2.2 facilitates spectrogram-to-waveform conversion. These voice-converted audio clips are subsequently used in a listening study in order to evaluate the models' performances. Objective measurements such as reconstruction loss, cosine similarity, and singer classification are also used to provide more insight into the models' performance and how this was achieved.

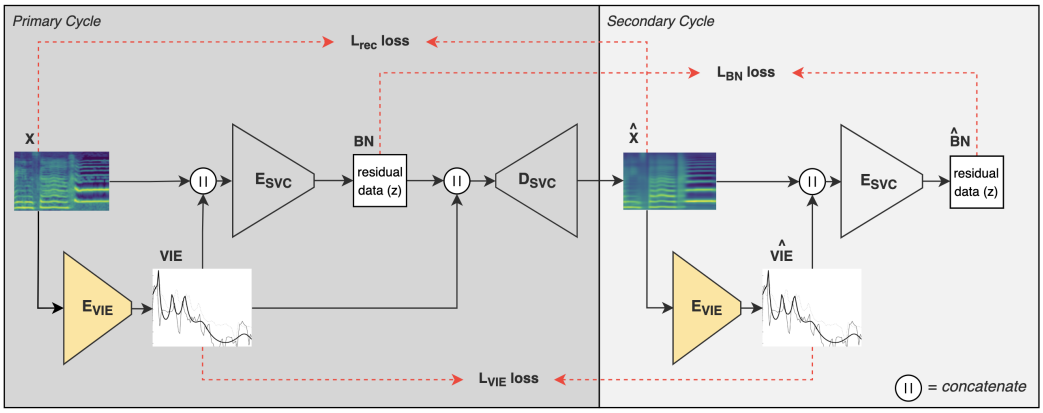


Figure 7.7: Diagram of the VIE encoder and the SVIC network components in the AutoSVIC network, with a secondary cycle partition illustrating how encodings for the reconstructed data are obtained (as explained in Figure 6.1). Vector comparisons used for reconstruction loss (L_{rec}), bottleneck regressor loss (L_{BN}) and singer identity embedding regressor loss (L_{VIE}) are shown with dotted connectors. X represents a mel-spectrogram and BN represents the residual data (singer identity-independent information) encoded in the bottleneck.

7.3.1 Loss Function Comparison

Findings in Chapter 6 included the fact that AutoVC's L_{BN} loss (described in Section 6.2.2) hindered performance when doing singing technique conversion tasks [O'Connor et al., 2021]. This experiment was designed to deduce whether the exclusion of this loss would also be beneficial in the context of SVIC. As an alternative solution, the notion of a latent regressor w.r.t. the VIE embeddings was also proposed, as this was hypothesised to improve performance without the need for bottleneck capacity calibration. An illustration of how each of these losses relates to the AutoSVIC system is provided in Figure 7.7.

Three AutoSVIC models were trained for 500k iterations using a batch size of 2, an L1 reconstruction loss, a learning rate of 10^{-4}, the ADAM optimiser, and mel-spectrogram features (extracted from the DI dataset in the same manner as described in Section 7.2.2) as input features to both the VIE encoder and SVIC network.

The effects of loss components on SVIC are being investigated in this experiment, and no claims are being made regarding SOTA results. Consequently, the DI dataset by which AutoSVIC was trained was reduced to 25% of its original size

(a different reduction to the subset generated for VIE training, but maintaining the same train and test subset ratios) for the sake of economical experimentation. Computational time was further reduced by precomputing singer VIEs. This was done by computing the mean of all VIEs collected for each singer. The precomputed VIEs are stored in a lookup table for retrieval during the training phase.

The models are listed as follows, accompanied by a description of how they differ from one another:

- **DI**: Use of the DI-pretrained VIE encoder, using the reconstruction loss only
- **DI-VIE**: Use of the DI-pretrained VIE encoder using the reconstruction and L_{VIE} loss
- **DI-BN**: Use of the DI-pretrained VIE encoder using the reconstruction and L_{BN} loss

7.3.2 Disparity between Same and Cross Domain Inference

To confirm that there is a significant difference in performance when a VIC model pretrained on one voice domain must infer from another, a preliminary experiment was first conducted to demonstrate the disparate outcomes between same and cross-domain inferences. This was done by training an AutoVC model on 20 speakers from the VCTK dataset as described in the original work [Qian et al., 2019] for 500k training steps, using the LSVC-trained encoder. A second model was trained on a reduced DI dataset. Multiple examples of speech and singing audio clips were then fed to the AutoVC models for resynthesis in their voice-converted forms. Examples of this can be heard in the online repository provided at the beginning of this chapter.

While speech-converted audio generated from a speech-trained model mimics the phonetic content, prosody and timbre of the source recording, none of these aspects remain intact when the model attempts to convert singing data. The converted audio contained very different phonetics to the original audio. There are often fast successions of variable phonetics being produced in place of sustained vowels. While some phonetic approximations are reasonably similar, the majority

are unpredictable and rapid, suggesting that the speech-pretrained network has difficulty dealing with sustained vowels, causing shifts in the probabilistic outcome that would be of a frequency more typically seen in speech data. Converted audio clips possess some changes in pitch that are influenced by the input. However, the variance of the pitch in converted audio is significantly smaller than that of the source inputs, matching the frequency range of speaker prosody rather than a singer's F0 range. Most importantly, the converted voice identity (or even gender) is seldom similar to the target singer's voice.

The reverse case, where a singing-trained model takes speech as its input for conversion, also yields peculiar artefacts. The converted audio clips frequently exhibit a pseudo-quantization of the speaker's pitch, resulting in melodic instead of prosodic pitch contours. There are also octave leaps that cause disjointed pitch contours, counteracting the general shape of the prosodic line. As with the speech-trained conversions on singing data, the resulting audio frequently does a poor job of converting the singer identity and can even get the gender wrong.

Based on these artefacts and differences, it is clear when comparing speech conversions against singing conversions that using a voice conversion model for cross-domain inference is far from suitable, thereby justifying an investigation into how much the history of the pretrained encoder contributes towards AutoSVIC's reconstruction and conversion capabilities.

7.3.3 Same and Cross-Domain VIE Encoder Comparison

In the previous section, differences between GE2E losses were established for same and cross-domain applications of the VIE encoder. Visualisations via t-SNE plots affirmed these differences, but the question of how significant these differences are and how they affect downstream tasks remained uncertain. It was concluded that singing data has more generalisable information than speech data. It would be reasonable to assume that SVIC models conditioned on these same VIE encoders would perform similarly in comparison to each other. Having now demonstrated disparity in cross-domain applications of VIC models, an investigation on how this is affected when only the VIE encoder of an AutoSVIC model is pretrained on speech data can be conducted, while the rest of AutoSVIC is trained

on singing data. It was anticipated that this would have a smaller but noticeable effect of deterioration in AutoSVIC's performance.

However, another unanswered question that would be of considerable interest to other researchers is whether a VIE encoder pretrained on large-scale speech datasets would yield similar results in a downstream SVIC task, as this addresses more real-world circumstances where VIE models have been pretrained on speech datasets many times larger than singing datasets. For this reason, the following fourth AutoSVIC model was trained in the same way as the original three from Section 7.3.1, and compared against the *DI-BN* model:

- **LSVC-BN**: Use of the LSVC-pretrained VIE encoder, using the reconstruction and L_{BN} loss

7.3.4 Evaluation

Listening Study Evaluation

For subjective evaluations, a listening study was conducted, consisting of 23 participants aged between 24 and 53 years old. Aspects of this study relating to the online web server, recruitment process, interface design toolkit (WAET) and delivery of documentation and instructions, were set up in the same manner as in Section 5.2.3. Each participant was given two tasks for each of the stimuli, instructed via the following prompts:

- Naturalness Question: "Click the PLAY button and rate how natural/realistic this voice is, on a scale of 1 (very unnatural) to 5 (very natural)"
- Similarity Question: "Do the vocalisations from the REFERENCE and PLAY audio sound like they could have been made by the same singer? Rate using the scale from 1 (definitely not) to 5 (definitely)".

The interface for each type of question consisted of a reference recording and a 5-point Likert scale. The similarity question also included an audio clip of the target singer that participants used for comparison with the reference recording. While the wording of the similarity question might seem awkwardly phrased, it

was carefully crafted so as not to give participants the impression that the voices were *meant* to sound the same. It also ensures that participants consider only the *identity* of the singer in their rating, rather than conditioning their answer on the similarity of other vocal attributes alone such as timbre, singing techniques, resonance, or musical ability.

Participants were given a practice round consisting of 4 naturalness and 4 similarity questions, to get familiar with the interfaces and tasks before beginning the recorded part of the study. The results of this study will be presented and discussed in Section 7.3.5.

Objective Metrics Evaluation

As the audio quality in the models' outputs degrades, participants' ability to assess similarity between a target singer and the converted audio's singer may suffer. Perception of naturalness and similarity could therefore be correlated. Analysis of the models' performances therefore included cosine similarities between VIEs of converted audio clips and the target audio from which they were generated. This allows for the investigation of the models' conversion performance in a manner that is disentangled from naturalness. Having observed the results between cosine and subjective similarity, analysing the visualisations of the embedding space and reconstruction loss contours was necessary, providing further information regarding AutoSVIC's performance. Vocalist classification metrics are also presented in this section to determine how different loss functions affected VIE information disentanglement.

Stimuli

In previous work on singing voice conversion [Nercessian, 2020], a similar architecture was trained with pitch-conditioning embeddings. During the conversion, pitch information from the source data were transposed by an octave to match the octave closest to the average range of the target singer. This is a forceful application of pitch shifting that does not take the source singer's pitch range into consideration or how singer timbre can change with pitch - but in the context of some SVIC methods or contexts the necessity to do so can be appreciated. However, for

complete authenticity in SVIC, octave shifts were not applied. Instead, a pitch-matching algorithm was applied so that a random target singer was selected from a pool of singers with roughly the same vocal range as the given source singer clip. This also ensures that evaluations of the network's performance can compare the speaker identities of two audio clips without being affected by changes in pitch register.

The listening study featured audio clips from the four models. Each model generated four converted audio clips using the test subset of the DI dataset, each of which represented one of the four source-target gender conditions $M-M, M-F, F-M, F-F$, where M and F stand for male and female, respectively. Similarity and naturalness tasks were required for each of these stimuli. Four *unconverted* audio clips were also included (see the last paragraph of Section 6.4.2 for a description of how and why these clips were used), bringing the total number of stimuli per listening session to $(4 \times 4) + 4 = 20$, with each stimulus requiring two ratings from participants.

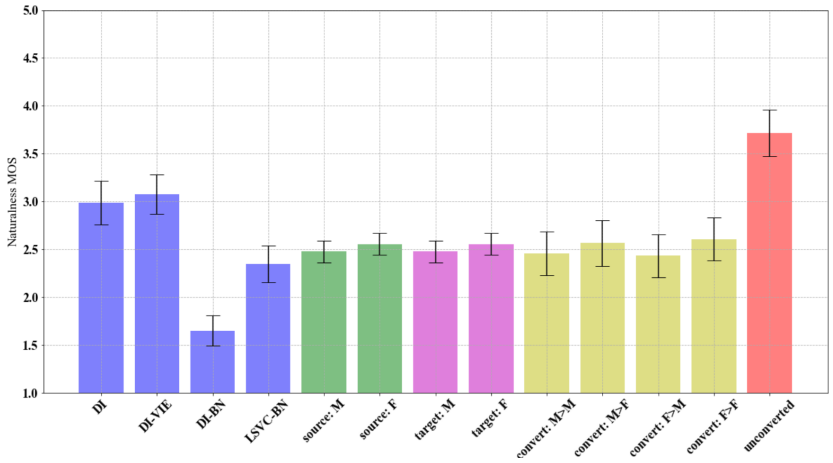
7.3.5 Results

In this section we sequentially describe the results of the listening study, embedding space visualisations, cosine similarities and disentanglement metrics. Section 7.4 combines these findings and discusses their implications.

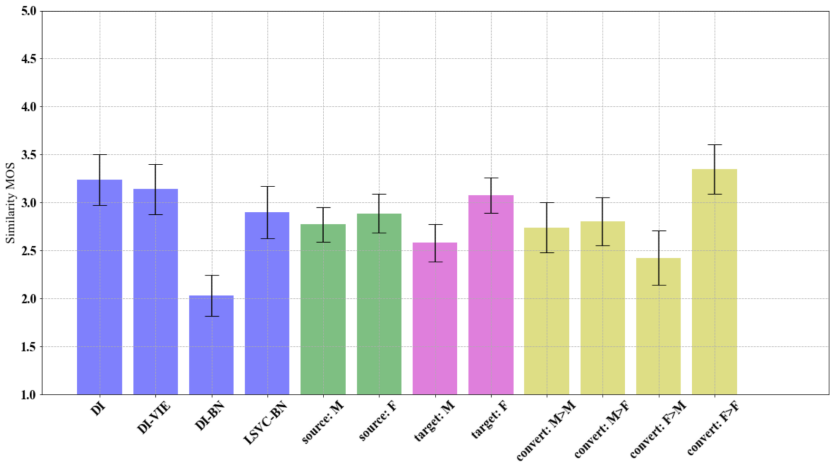
Participant Ratings

Subplots (a) and (b) of Figure 7.8 display the results of the listening study for perceived naturalness and similarity as MOS results under different conditions. Condition groups are colour-coded together from left to right as 'models', 'source genders', 'target genders', and 'gender pairs'. Note that the naturalness rating for unconverted audio clips was 3.72, which can be considered as the approximate upper bound ceiling of perceptual evaluation due to the resynthesis process.

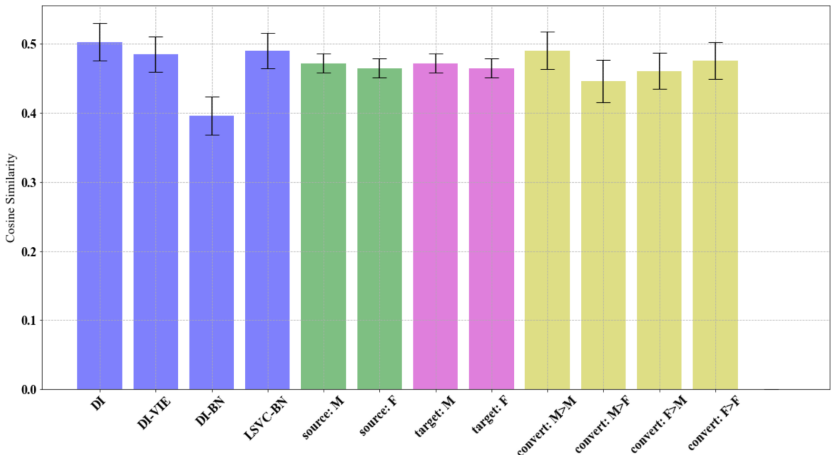
While most rating distributions for various experimental conditions were visually interpreted as normally distributed, a significant number did not pass the Shapiro-Wilk test. Consequently, we employed the Mann-Whitney U test to assess the statistical significance of rating samples across different conditions. The



(a) Naturalness results



(b) Similarity results



(c) Cosine Similarity results

Figure 7.8: Bar graphs displaying results for naturalness, similarity and cosine similarity.

Rating Type	Condition Group	Group 1	Group 2	*U*	*p*
Naturalness	model	DI	DI-BN	7013	<0.001
Naturalness	model	DI	LSVC-BN	5616	<0.001
Naturalness	model	DI-VIE	DI-BN	7256	<0.001
Naturalness	model	DI-VIE	LSVC-BN	5845	<0.001
Naturalness	model	LSVC-BN	DI-BN	2420	<0.001
Similarity	model	DI	DI-BN	6394	<0.001
Similarity	model	DI-VIE	DI-BN	6252	<0.001
Similarity	model	LSVC-BN	DI-BN	2671	<0.001
Similarity	target gender	F	M	13388	<0.001
Similarity	gender-pair	F $\rightarrow$ F	M $\rightarrow$ M	3110	<0.005
Similarity	gender-pair	M $\rightarrow$ F	F $\rightarrow$ M	4948	<0.05
Similarity	gender-pair	F $\rightarrow$ F	M $\rightarrow$ F	3193	<0.005
Similarity	gender-pair	F $\rightarrow$ F	F $\rightarrow$ M	2648	<0.001
Cosine	model	DI	DI-BN	6122	<0.001
Cosine	model	DI-VIE	DI-BN	5957	<0.001
Cosine	model	LSVC-BN	DI-BN	7144	<0.001

Table 7.3: Mann Whitney U results of significant differences between samples of different conditions relating naturalness, similarity and cosine similarity.

results of this analysis can be found in Table 7.3.

Audio generated by DI and DI-VIE scored significantly better for naturalness than the LSVC-BN and DI-BN. Those of DI-BN where significantly lower than all other models' audio, which was also the case for similarity scores, indicating that the inclusion of the L_{BN} significantly hindered AutoSVIC's overall performance. The reason for this is likely because an additional loss component has been included in the loss function with equal weighting, which has the potential to slow down the network's convergence rate when it needs to optimise for multiple objectives.

It is worth noting that despite the similarity in nature between the DI-VIE and DI-BN models' latent losses, there is no corresponding statistically significant drop in performance when the DI-VIE model is compared to the DI model. This observation underscores the importance of ensuring that the decoder gives priority to the utilisation of the conditioning VIEs by preserving this information in the output via the L_{VIE} loss, leading to significantly enhanced SVIC compared to the

utilisation of residual information from the bottleneck via the L_{BN} loss. The L_{VIE} loss also regularises AutoSVIC weight updates in favour of the SVIC decoder (while the L_{BN} inherently favours the SVIC encoder), forcing the network to focus on its resynthesis capabilities, which may account for the higher naturalness scores attributed to it.

The performance of the LSVC-BN model was particularly surprising. While this model was trained using a VIE encoder pretrained on a large speech corpus, it was still perceived to have produced more natural and convincing conversions than the DI-BN model (which also has the advantage of being trained on data from the same dataset as the rest of the AutoSVIC model). This seems counter intuitive as the GE2E losses shown in Table 7.2 suggest that cross-domain inference was less successful when using a speech-pretrained encoder to infer from singing data.

No other statistically significant differences were found between any other samples of naturalness scores. When the target gender was female, generated audio was rated significantly higher than its male counterpart. The $M \rightarrow F$ and $F \rightarrow F$ both scored higher than their male counterparts, although only the latter's score was of statistical significance. This suggests either that AutoSVIC has more success in converting audio to female voices, or that these conversions and their target voices sound more similar to participants.

AutoSVIC Loss

Figure 7.9 offers an objective metric for loss contours that reinforces some of the naturalness ratings reported in the listening study. These losses represent the sum of all the loss components in each model's objective function. It is, therefore, important to note that the variations in loss between the DI model and other models may be attributed to the fact that no additional losses were added to the reconstruction loss in the DI model. Notably, the LSVC-BN and DI-BN models exhibited similar performance, suggesting they should yield similar naturalness scores. Meanwhile, the DI-VIE model's loss contour falls between the models using either L_{rec} alone or L_{rec} with L_{BN}. These observations will be considered in the context of a broader examination of multiple evaluation methods, which will be discussed in the subsequent sections.

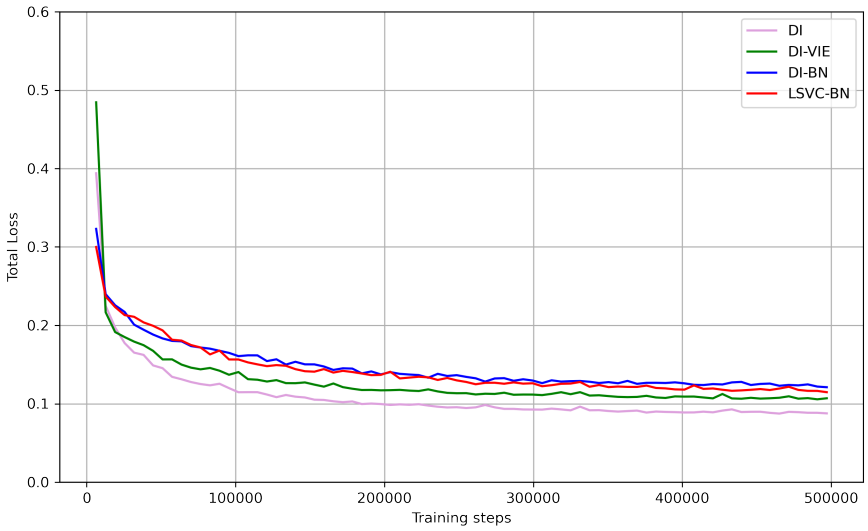


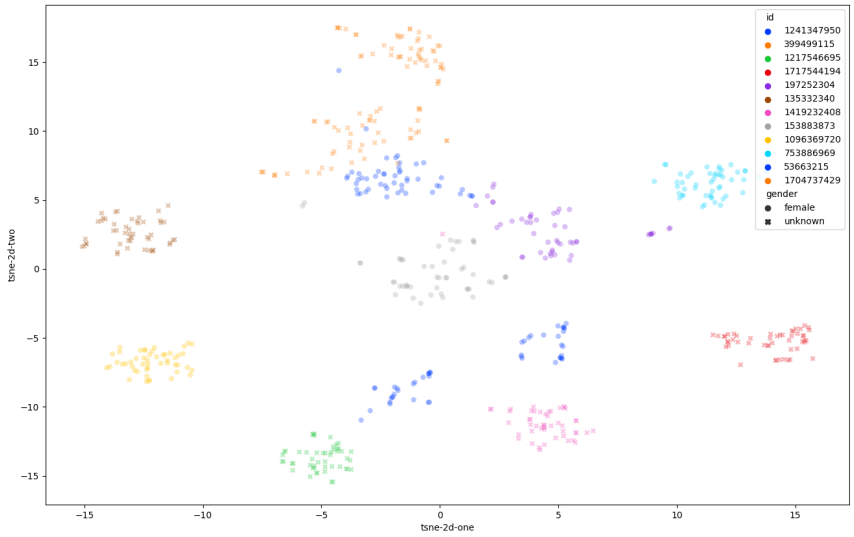
Figure 7.9: Total loss contours for each of the four trained AutoSVIC models.

Embedding Space Visualisations

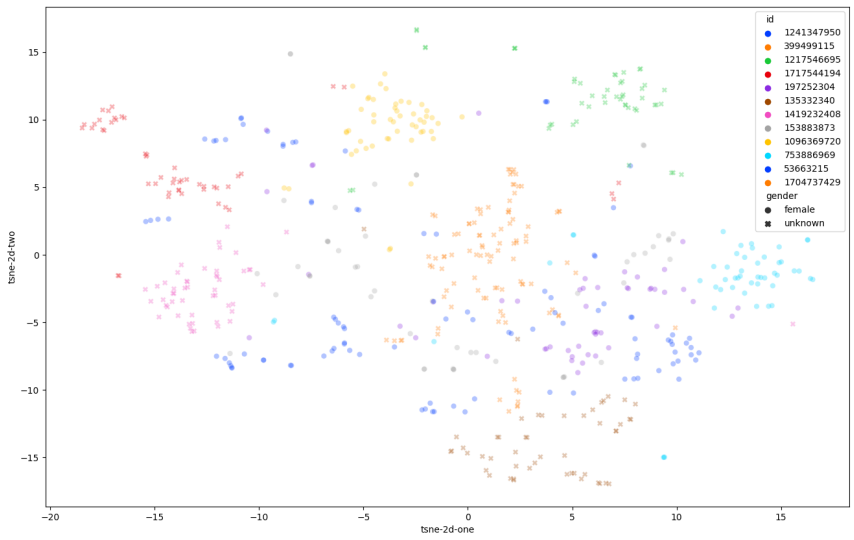
Figure 7.10 features t-SNE plots which illustrate the clustering structures of different vocalist VIEs, where subplot (a) and (b) represent embeddings inferred from the DI-pretrained encoder and LSVC-pretrained encoder respectively. It is clear from these plots that the DI encoder produces vocal clusters that are significantly more separated and compact that those of the LSVC encoder. As the listening test results indicate that the LSVC encoder produces better results, the question must be asked: Why do the LSVC encoder's embeddings, which exhibit poor clustering tendencies, produce VIEs that contribute towards superior voice-converted audio outputs? Hypotheses attempting to explain this will be offered in Section 7.3.5

Cosine Similarity Metrics

During the analysis of the listening study's results in Section 6.4.3, no significant correlation was detected between similarity and naturalness. In this case, a strong correlation between the two types of ratings was visually evident, and a Pearson test revealed a correlation coefficient of $r = 0.82, p < 0.002$. In light of the ro-



(a) DI test data embeddings generated from the DI-pretrained encoder



(b) DI test data embeddings generated from the LSVC-pretrained encoder

Figure 7.10: t-SNE generated maps of DI-originating VIEs using the (a) DI-pretrained and (b) LSVC-pretrained encoders.

bustness of this positive correlation, caution should be exercised in assuming that the perception of successful SVIC is independent of the perception of naturalness by participants. It is hypothesised that when participants were prompted to evaluate the confidence that the reference recording and the voice-converted recordings originated from the same singer, lower similarity ratings may have resulted from one or both of the following circumstances: artefacts introduced by the SVIC process in the audio file would lead to a loss of timbre clarity, or the timbre of the converted voice itself was dissimilar to the target voice.

Converted and target voice similarity is therefore assessed using cosine similarities between the respective VIEs, which are displayed in subplot (c) of Figure. 7.8. As shown in the lower entries of Table 7.3, the only statistically significant differences are between the DI-BN model and all other models. Statistically significant differences in other condition groups for subjective similarity were not present in cosine similarity scores, which is also the case for naturalness scores.

Singer Identity Disentanglement

The amount of disentanglement between singer-identifiable and residual information was analysed by appending a classification layer to the output of the SVIC encoders of the three trained AutoSVIC models using the DI-pretrained VIE encoder. 20 singers from the DAMP test subset were used for this classification task. The resulting classification accuracies enable a comparison of the amount of singer-identifiable information remaining in AutoVC's bottleneck vectors between different models.

The summary of accuracy results pertaining to bottleneck classification layers across the three models is presented in Table 7.4 and shown for intuitive visualisation in Figure 7.11 for the sake of convenience. Notably, the DI model employing solely L_{rec}, demonstrates a classification accuracy of 45%. This finding suggests that while a bottleneck of dimension 256 (equivalent to 16 timestep by 16 frequency bin dimensions) may suffice for disentanglement in the context of the VCTK speech dataset, as previously reported in the work of Qian et al. [2019], it does not hold true for the DAMP singing dataset. Consequently, it becomes evident that a significant proportion of voice identity information remains entangled

Loss components used	Classification accuracy
L_{rec}	45%
L_{rec} and L_{VIE}	35%
L_{rec} and L_{BN}	23%

Table 7.4: Classification accuracy results for models using different loss functions.

in the bottleneck.

In contrast, the DI-BN model implementing both L_{rec} and L_{BN}, produces the lowest classification accuracy of 23%. This observation suggests that the network's SVIC encoder minimally encodes singer identity information, thereby facilitating maximum disentanglement of singer identity from the input data. This outcome is driven by the SVIC encoder's incentive to prioritise encoding the residual content, which in turn results from the presence of conditioning VIEs.

Furthermore, the DI-VIE model incorporating L_{rec} and L_{VIE} loss functions achieves an accuracy of 35%, representing a 10% decrease in classification compared to the DI model trained solely with L_{rec}. Its scoring between the DI and DI-BN models is reflected in the accuracy contours of Figure 7.9, which indicates a moderate amount of disentanglement. This decline from the DI model's accuracy score signifies enhanced disentanglement and is attributed to the network's greater reliance on the decoder, leveraging the conditioning VIEs. Nevertheless, this improvement does not guarantee the avoidance of voice identity information being encoded by the SVIC encoder, a distinction elucidated by the persistently higher accuracy in comparison to the DI-BN model. This affirms that the L_{VIE} loss component offers the additional advantage of resilience against sub-optimal disentanglement within the bottleneck. Hence, the utilisation of L_{VIE} obviates the need for manual bottleneck capacity calibration, which can be a long process, especially when human evaluations are required to determine this.

Discussion

By observing both subjective and objective similarity metrics, a few conclusions can be drawn. Firstly, while cosine similarities exhibited no significant difference between gender-pair conditions, participants perceived conversions to be more successful when the conversion was between two female singers, while objective

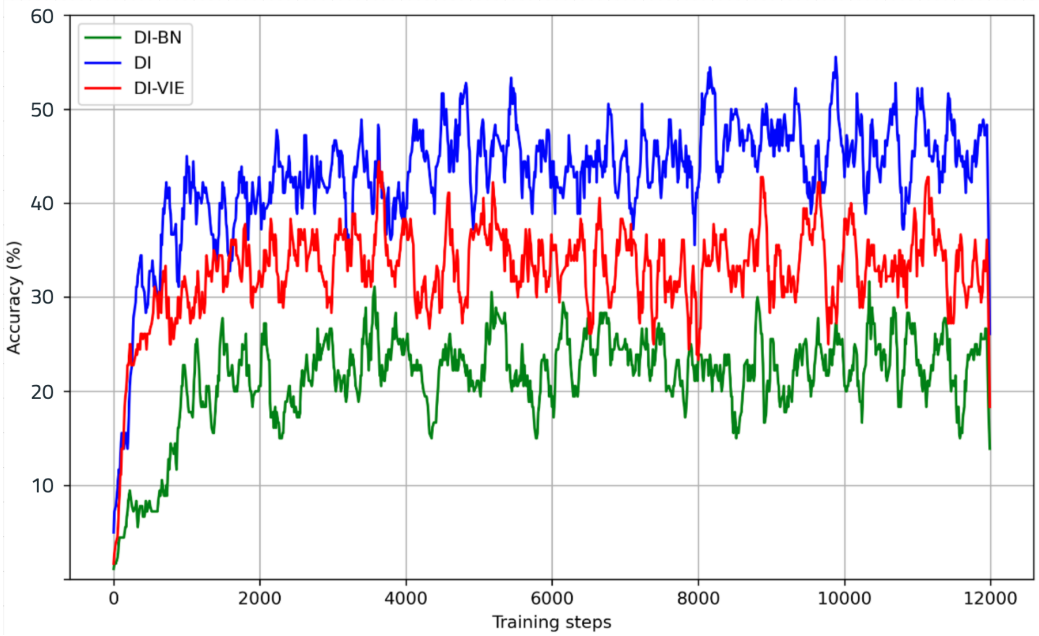


Figure 7.11: Classification accuracy of classification layers being appended to the encoders of the DI, DI-BN and DI-VIE models, indicating the amount of singer identity information still entangled in the models' bottlenecks.

measurements showed similarity to be generally equal for all gender-pair conversions. This serves as a clear illustration of the critical importance of employing both subjective and objective evaluation methods when assessing model performance. One possible explanation for this observation might be that the diversity of timbre among females may be lower than it is for males, meaning that any conversion towards a female voice will be more likely to be heard as similar to a target female voice than if the target were male.

The LSVC-pretrained encoder led towards better SVIC than the DI-pretrained encoder even though the latter produced strong clustering VIEs, while both corresponding AutoSVIC models' loss contours seen in Figure 7.9 were roughly the same. From these seemingly contradictory results, it can only be concluded that despite strong clustering tendencies, the DI encoder does not represent the aural cues of voice identity as well as the LSVC encoder. Their corresponding AutoSVICs could therefore still achieve similar loss, only that in the case of the DI-BN model, the conversions were not only related to voice identity cues.

This may be because features relating to voice identity (and largely to timbre) exhibit significantly more variance in the singing domain than the speech domain, due to more dimensions of expressivity such as timbre and singing technique. These expressive manifestations can often redirect or mask the default representation of voice identity. Coarsely speaking, vocalists of the same gender with similar pitch ranges and singing techniques would be less distinguishable in a singing context than a spoken one. Additionally, the fact that cues such as accents, pace and intonation are less relevant in the singing domain makes the task of voice verification and discrimination even more challenging. Consequently, the DI-pretrained encoder may have attempted to encode additional features that are constant cues across singer recordings, but not directly related to singer identities, such as:

- background noise
- microphone frequency response
- room acoustics
- singer's proximity to the microphone

These variances are far less apparent in the datasets seen by the LSVC-trained encoder. By producing embeddings that relate to these non-vocal confounding features, AutoSVIC could then perform decent conversions w.r.t. these embeddings, while listeners would not perceive strong conversions of singer identity.

7.4 Conclusion

7.4.1 VIE Encoder Experiments

The first half of this chapter covered the VIE encoder. WORLD feature engineering was explored, from which the conclusion was drawn that none of the implemented changes in WORLDs algorithmic configurations improved the encoder's ability to generate singer-specific VIEs for the singing voice using the GE2E loss. These changes included:

- Expanding WORLD's default configuration (71-800Hz) for F0 range
- Including WORLD aperiodic features
- Including WORLD pitch features
- Using either of WORLD's alternative pitch estimation algorithms

Simultaneously, it was concluded that WORLD's unprocessed spectral envelope features, and MCCs derived from these features performed more poorly than MFSCs derived from these features. As a result, the MFSC representation of WORLD features was used in subsequent experiments.

After establishing this version of WORLD's feature representation as most suitable for VIE generation, they were compared against mel-spectrogram features. It was found that although the feature engineering that goes into generating WORLD's features provided the encoder with a head start in providing explicit VIE-relevant information, after 30k training steps, mel-spectrograms outperform these features consistently across both singing and speech domains.

The effects of cross-domain applications of the VIE encoders were then tested, where they were given mel-spectrogram features from either singing or speech domain datasets, and evaluated on a third dataset consisting of both domains in its separate partitions. Results showed that intuitively, there was an increase in the GE2E loss for cross-domain applications for both singing-to-speech and speech-to-singing conditions. However, the increase in loss (and therefore deterioration in performance) was larger when inferring from singing data using a speech-pretrained encoder. The gap in performance between same and cross-domain applications was slightly wider for an encoder pretrained on large speech datasets (LSVC), suggesting that in realistic cases where models are trained on very large speech corpora, there is still a larger drop in performance when inferring from singing data than in the domain-inverted (singer-to-speech) circumstances. Two-dimensional plots of embeddings in a latent space provide a visual for how the differences in GE2E loss measurements affect clustering tendencies, which were in line with the observed differences in GE2E losses between same and cross-domain inferences.

In summary, the findings from this research determined that changing the default configurations previously listed for WORLD feature generation did not improve the VIE encoder's task of VIE generation. MFSCs derived from WORLD features facilitated VIE generation better than MCCs or unprocessed WORLD spectral envelope features. However for this same task, mel-spectrograms performed better than these features, and were therefore used in all remaining experiments. Cross-domain inferences yielded measurably higher loss values than same-domain inferences. The differences in loss values between same and cross-domain inferences were measured to be larger for speech-to-singer than for singer-to-speech cross-domain conditions, indicating that for VIE generation, singing data has more information that is transferable to speech data than the inverse scenario.

The interpretation of these findings, however, is adjusted in the proceeding section which takes new data into account.

7.4.2 SVIC Experiments

The second half of this chapter addresses the two following research sub-questions in relation to the task of SVIC:

1. How does a speech-pretrained VIE encoder affect AutoSVIC when trained on singing data for singing data inference?

2. How do alternative regularisation terms in AutoSVIC's objective function affect its performance?

Results for Question 1

Results from Section 7.2.3 indicate that cross-domain inference for VIE generation performs worse than same-domain inferences, although the effect size of this on downstream tasks was unclear. To determine this, the encoder pretrained on a large corpus of speech data and encoder pretrained on the DI dataset were applied to AutoSVIC networks. A listening test was conducted where participants were asked to rate the converted recordings' similarity to target singers, as well as their naturalness (audio quality). Results suggested that the AutoSVIC model

using the *speech*-pretrained encoder produced converted audio that was better in quality and similarity than those using the singer-pretrained encoder. Cosine metrics (generated between the converted VIEs of the recordings and the target singer recordings) produced similar results.

From this, it was deduced that the DI dataset's diverse recording conditions may have contributed towards optimising same-vocalist clusters' VIEs in a latent space influenced by these non-vocal acoustic features. The weighting of importance to these confounding features may be increased by the fact that there are a number of singer-domain considerations that can make differentiating between singers particularly difficult. The speech-pretrained encoder, on the other hand, was trained on a large speech dataset, containing a large amount of professional, well-constrained recording conditions as well as some amateur recording conditions.

Results for Question 2

Results described in Section 6.3.3 suggested that the L_{BN} loss hindered AutoSTC's convergence, exposing the uncertainty of whether it was similarly detrimental towards VIC. The L_{VIE} loss was proposed as an alternative regularisation term to the L_{BN} loss, which is attributed more to the network's decoding capabilities rather than its encoder. AutoSVIC networks were therefore trained with the following objective functions (with model names in parentheses):

1. L_{rec} (DI model)
2. L_{rec} + L_{BN} (DI-BN model)
3. L_{rec} + L_{VIE} (DI-VIE model)

Listening tests concluded that the DI-BN model produced the worst quality audio with the least convincing VICs, while the other two performed similarly to one another. However upon disentanglement metric generation, it was found to best disentangle voice identity information from the bottleneck. The DI-VIE model was measured to have *some* disentanglement while the DI model had the least amount of disentanglement. This indicates that while the DI-VIE model could

perform on par with the DI model, it was more capable of disentangling information, while its conversion capabilities remained robust against the remaining voice identity information in the bottleneck. This makes it an attractive regularisation term for voice conversion systems and tasks where bottleneck disentanglement is not a primary concern, as calibration requires subjective evaluations, slowing down the optimisation process considerably.

In summary, the L_{VIE} is favourable over the L_{BN} loss as it achieves the same qualities as a model without regularisation terms, but remains robust against disentanglement, inherently requiring less bottleneck calibration, and explicitly acts as an assurance that the voice identities will be successfully encoded in the decoder of AutoSVIC. Of course, the advantages of this regularisation term will be dependent on the goals of the researcher. While including an L_{VIE} loss has not been shown to cause better audio outputs than its exclusion, it is proposed that this precautionary loss may be more advantageous when training an SVIC network for longer on larger, more complex datasets, which can be confirmed in future work.

Analysing results from the listening test showed that naturalness and similarity ratings were strongly correlated. For this reason, cosine similarities between converted and target voices were also measured, which yielded similar results. This suggests that the AutoSVIC network's reconstruction capabilities incrementally improved with its conversion capabilities.

7.4.3 Future Research

Verifying whether the conditions of the DI dataset recordings affected the VIEs exclusivity to singer identity information is fairly straightforward. Based on the list of potential confounding variables provided in Section 7.3.5, this could be addressed with augmentation via audio processing such as injecting noise, altering frequency responses, gain etc. More advanced options include some light pitch shifting (perhaps no more than a semitone, with corrected formant positioning) to avoid any potential conditioning by musical key, or a preprocessing step that uses sound-source separation to remove the background noise instead of masking it.

Listening test naturalness MOS results were in most cases lower than 3.0. Compared to other literature, this would be considered quite low, but can be at-

tributed to several aspects of the research. The primary loss of quality and similarity likely comes from the resynthesis process. In the interest of time, an older spectrogram-to-waveform converter was used, which is now likely superseded by newer models that produce cleaner audio, such as Parallel WaveGAN [Yamamoto et al., 2020] or Hifi-WaveGAN [Wang et al., 2022]. The number of training steps was capped at 500k, and only a quarter of the DI dataset was used for training. Both of these aspects could be extended to improve the model's performance. Pitch conditioning has also been shown to improve disentanglement and conversion processes [Qian et al., 2020a], and would allow for more efficient training.

Chapter 8

Conclusion

8.1 Summary of Contributions

In this thesis, considerations related to the attributes of the singing voice were explored. These include human perception of singing techniques, disentanglement and conversion of specific attribute information from a recording, the utility of voice datasets that present the voice in different contexts, feature selection, and converted audio evaluation.

8.1.1 Perception of Singing Techniques

Dissimilarity ratings between vocalisations demonstrating different singing techniques were collected in Chapter 5. Clustering analysis concluded that five (the number of hidden singing technique labels) clusters best represented the participants' perception of the data, indicating that VocalSet's ground-truth singing technique classes are suitable. Statistical analyses revealed differences between PCDs under different conditions. Timbral maps were generated from participant ratings using MDS, which also illustrated these differences. These results are useful to musicologists who wish to better understand how humans perceive singing techniques and software engineers who wish to utilise such perceptual data as a regularising factor in their ML models.

Correlation analyses between participants' features and cluster scores revealed that, in general, participants with more musical knowledge produced better ratings

with stronger clustering tendencies and agreement with the ground truth data. This is an important finding that should be considered when researchers recruit for future listening studies.

8.1.2 Conversion Models

Chapter 6 presented a singing technique classifier. It scored an average accuracy of 75% on a six-way classification task of VocalSet data. After training, the classification layer was removed, allowing the classification network to become an STE encoder. The output STEs were used to condition an AutoVC-like model to achieve disentanglement and conversion of the singing technique. This was referred to as AutoSVC. Unlike AutoVC, AutoSTC produced the most natural converted spectrograms when using an L_1 reconstruction loss and no latent regressor loss.

Another attribute that was subject to disentanglement was voice identity. In Chapter 7, VIE and SVIC were the subjects of investigation. Using the proposed architecture and loss function of Wan et al. [2018] for voice verification, VIEs were produced. After training on a singing dataset, the output of this VIE encoder was used to condition another AutoVC-like architecture to achieve SVIC, referred to as AutoSVIC.

When using a latent loss w.r.t. VIEs and an L_1 reconstruction loss, the AutoSVIC model was evaluated to produce the most natural audio and convincing conversion to target singers, as opposed to other versions where it used reconstruction loss and latent loss w.r.t. bottleneck encodings. It also proved to be robust against poor disentanglement, as significant speaker information was still retained in its bottleneck encoding.

As both the VIE and STE attribute representations were in the form of descriptive embeddings and not one-hot encodings, they facilitate any-to-any or zero-shot conversions, which is far more convenient for most research and industrial tasks than other conversion types, such as many-to-many.

8.1.3 Datasets of Differing Voice Representations

A training strategy was outlined in Chapter 6 that allowed AutoSTC to learn sequentially from multiple datasets that presented the voice in different contexts. The implementation included the following datasets: one featuring classically trained dry-recorded singers (VocalSet), another featuring dry-recorded speech (VCTK, [Veaux et al., 2017]), and the last featuring post-processed, *a capella* stems of singer recordings originally intended to be used as part of a mixed track (MedleyDB [Bittner et al., 2014]). By monitoring loss values with respect to the evaluation subset of a chosen dataset, an optimal path could be calculated that minimised catestrophic forgetting and the final loss value.

It was concluded from the results of AutoSTC's listening study that VocalSet is too small a dataset in size and in scope for a network to learn embeddings that describe the singing techniques and would generalise to other datasets.

To determine the amount of transferable knowledge between the speech and singing domains, VIE encoders were pretrained on either a speech or singing dataset and subsequently evaluated on the remaining dataset domain. A larger gap in the GE2E loss values was observed when using a speech-pretrained encoder on singing than a singing-pretrained encoder on speech (this was still the case even when using an encoder pretrained on a large-scale speech dataset). This suggests that more knowledge can be transferred from singing to speech data than speech to singing data.

8.1.4 Data Representation

Numerous variations of WORLD-generated spectral envelopes were compared to mel-spectrograms as input features to a VIE encoder for the task of VIE generation. For all comparisons in either speech or singing domains, mel-spectrograms supported superior performances, indicating that the VIE encoder found more discriminative features in these than WORLD-based features. This is an important finding that future researchers may benefit from, as previous literature has shown that many researchers still use WORLD features over mel-spectrograms for voice conversion tasks.

8.1.5 Conversion Evaluation

Several novel methods have been proposed to evaluate the audio outputs of the conversion models detailed in this thesis. Standard methods such as computing MOS data from perceptual ratings were also used.

In a listening study designed to evaluate AutoSTC's converted audio outputs, a forced-choice task was given to participants, where they could choose more than one target class that best matched the reference audio file that was converted. A novel formula generated a positive score if any of the choices were correct, which was inversely proportional to the number of choices made. This determined a similarity score between the singing techniques of the converted and the target audio.

The results showed that nearly all conversion conditions scored above the chance level for similarity and provided some insight into which conditions led to the most similar and natural conversions. The AutoSTC model that used the optimal permutations of datasets for VocalSet produced worse similarity and better reconstruction scores than the model trained only on VocalSet. This shows that singing technique similarity will not necessarily improve with naturalness.

When evaluating the converted output of AutoSVIC, standard methods for evaluating similarity and naturalness were used. However, upon observing a correlation between both measurements, a third objective metric was introduced to produce a similarity metric between converted and target singer recordings that was certain to be disentangled from naturalness: the cosine similarity between the VIEs of both recordings. Participants rated the similarity of converted to target audio significantly higher than the similarity as computed by the cosine score, when the target singer was female. This shows how important it is to use both subjective and objective metrics.

The results of the listening study concluded that the AutoSVIC with the encoder pretrained on a small singing dataset was outperformed by the version that used a large speech dataset. It was hypothesised that because the singing dataset consisted of low quality amateur recordings with varying environmental acoustics, the VIE encoder trained on this data may have relied on cues unrelated to the voice, leading to potentially better GE2E loss results, but poorer VIE conver-

sion. Its reliance on such confounding cues would be increased by the fact that differentiating between singers is more difficult than between speakers.

8.2 Future Work

8.2.1 Listening Study

While conclusive results were drawn from the listening study in Chapter 5, there was a considerable amount of noise present in the data. For more robust timbral maps, several suggestions can be made that would improve future experiments of a similar nature.

The amount of noise in the data may have been due to the unintutive task listeners were given, where they had to quantify a perceptual distance between two vocalisations, while disregarding differences in their vowel sounds, pitch, and features that linked to their perceived identity. For similar listening studies in the future, such noise in data can be avoided by ensuring all notes and phonetics between compared stimuli are identical and not within a tolerance interval.

There was also a surprising lack of correlations between participants' data for the same listening sessions, as seen in the relevant correlation matrices. Silhouette scores suggested that the clustering behaviour of participant data was not particularly strong (although this may in fact just be the way humans naturally perceive stimuli). However, with more participants, these uncertainties could be mitigated and stronger statistical analysis could be achieved with the expected move towards normally distributed PCDs.

8.2.2 Improving Models

The findings presented in this thesis have related to voice analysis, disentanglement, conversion, and evaluation. There has been no claim to SOTA results regarding SVAC, as the research has been focused on model self-comparisons under different conditions.

When the research of Chapter 6 was conducted in 2020, the AutoVC architecture used was achieving SOTA results for zero-shot voice conversion. Although

newer models claiming SOTA results existed by the time the research of Chapter 7 was conducted, AutoVC remained the model of choice as the author felt their intimate familiarity with its structure and dependancies offered them efficiency in being able to investigate the effects of cross-domain applications, alternative input features, and alternative objective functions on the resulting converted audio. Advice in the form of a recent paper rejection suggested that the adoption of newer models would produce better quality results that would lead toward more accurate ratings among listening study participants. This advice seems to agree with the results of the AutoSVIC listening study of Chapter 7, where the naturalness and similarity ratings were strongly correlated.

STE Encoder

Therefore, improving reconstruction and minimising audio artefacts is vital when looking to improve the models. One viable method of doing so would be to switch from RNN-based architectures to Transformer-based ones. As described in Section 4.4, these architectures have significantly outperformed their CNN-based rivals in audio tagging, and so it seems reasonable to consider that the STE encoder may also benefit from such an architectural upgrade.

VIE Encoder

Transformer architectures have also been proposed for speaker identification models [Wang et al., 2023a], making them a viable architecture to incorporate into the VIE encoder. To further develop the VIE encoder, it also seems important to acknowledge how different vocal registers or phonations originating from the same voice could generate slightly different VIEs. Section 4.4.3 mentions two promising methods for further developing the non-static nature of VIEs, such as that of Tan et al. [2021] where they continually adjust embeddings in a lookup table as the model is exposed to more utterances, or Li et al. [2022b], who strive to capture the fluid nature of VIEs by using a U-net to produce hierarchical speaker embeddings at multiple granularities.

Voice Conversion Network

AutoVC has been repurposed in this thesis to achieve SVIC and STC. It can, of course, be used to convert any attribute of a signal, given the right conditioning factors. Adaptions to the AutoVC architecture could include the introduction of attention mechanisms and Transformers, which have led to improved results [Lin et al., 2021]. There are other new systems of different architectures, such as GlowVC [Proszewska et al., 2022] and StarGANv2 [Li et al., 2021a], which both use VIE conditioning and have reportedly outperformed AutoVC. As in RAVE [Caillon and Esling, 2021] and other recently proposed systems, a discriminator can also be appended to the output of the chosen voice conversion model, and trained with its decoder to enhance its reconstructive capabilities. Incorporating GMMs to simplify the distributions of data to be converted could also improve performance. AutoVC's dependency on pretrained VIE encoders could also be removed by applying VQ techniques, which have also been reported to improve AutoVC's performance [Tang et al., 2022].

Audio Synthesis

The WaveNet architecture was used to convert spectrograms into waveform audio. There are several areas of potential modification to this. Section 4.4.5 describes several generations of improved audio synthesis networks, the latest of which are HiFi-GAN Kong et al. [2020a] and HiFi-WaveGAN Wang et al. [2022]. In addition to this, it would, of course, be favourable to train such models specifically on singing data, rather than the speech-trained WaveNet used in the previous two chapters.

Input Data

One final amendment towards improving the models' performances is to feed them more data. As Nercessian [2020] identified, the AutoVC framework can be pretrained as a universal background model, which means it can learn meaningful representations from most types of voice recording data to reconstruct, without the requirement of labels. Therefore, there is a good opportunity to train a voice conversion model on the vast amounts of speech-domain data available. In

recent years, as described in Section 4.1.3, scripts have been provided that allow researchers to retrieve data from media repositories such as YouTube for a wide variety of music, specific to the particulars of a dataset. With source separation software approaching astonishing results in recent years, it may be a fruitful endeavour to use these capabilities to produce a capella recordings from mixed tracks.

The experiments here have frequently only made use of subsets of datasets and a capped number of training steps. Increasing both of these could, of course, only improve results. The number of auxiliary inputs for conditioning could also be increased (as was originally planned when attempting to access the information of the singing technique as residual disentangled information, mentioned in Section 7.1.1) to maximise the disentanglement, facilitating more individual control over the attributes of the voice.

8.3 Final Remarks

While new voice conversion systems are continually being proposed, in many cases architectures are re-implemented with minor modifications to better suit their task, and only a small subset of these is applicable to the task of zero-shot voice conversion where the output is the same representation as the input. However, even in this narrowly restricted definition of the task, there are many components that contribute to the conversion process. Each of these can be improved, replaced, or removed in future revisions of the system, allowing the literature to become quite vast, exploring many potential areas of improvement.

SVS models are becoming so convincing that it is almost impossible to know whether a recording is fake or not, especially when many of its acoustic qualities are masked by the mixed recording it sits within. This was beginning to become the case in 2017 when NPSS was first introduced [Blaauw and Bonada, 2018], and SVS capabilities are even more impressive with newer networks such as DiffSinger [Liu et al., 2022]. It is an exciting time to see how computational advances have enabled NNs to effectively synthesise entirely fictional yet realistic data.

However, I have yet to see a network that encapsulates a voice so convincingly that I, the human discriminator, have been outdone by such a generator in that I

cannot tell the difference between synthesised and real singers. Such a technology would have to be able to model a voice so perfectly in its expressive entirety, possessing the skill to deliver a dynamic, emotional performance that is moving or exhilarating in nature.

To this end, I would be excited to extend my research towards the continuous manipulation of latent spaces that is available through architectures such as VAEs. However, as someone who has a (possibly) unhealthy obsession with exhaustively exploring all possibilities in my local space before moving towards newer and exciting ones, I believe it is still essential for researchers to master attribute disentanglement before venturing off into the fields of dynamic attribute control.

Although this thesis does not address ethical or legal issues, it is still important to consider such concerns that come with SVAC, such as the potential infringement of copyright and intellectual property rights. Superimposing voice attributes of different singers onto source recordings could lead to threats of legal action and breaches of copyright, especially when the EU's General Data Protection Regulation (GDPR) law protects the usage of personal data such as voice recordings. Until official legal policies have been put in place, it is advisable to gain permission and licensing from the original owners of the target voices one wishes to imitate, before attempting to use SVAC models to produce converted voices in the public domain. This can be misconstrued as an attempt to mislead listeners about the voice's owner. The research presented in this thesis has been conducted for academic purposes, and the datasets used to train models to produce converted audio are all publically available. It should go without saying, that none of the models or techniques described here should be used to contribute towards the generation of synthesised voices for unethical purposes.

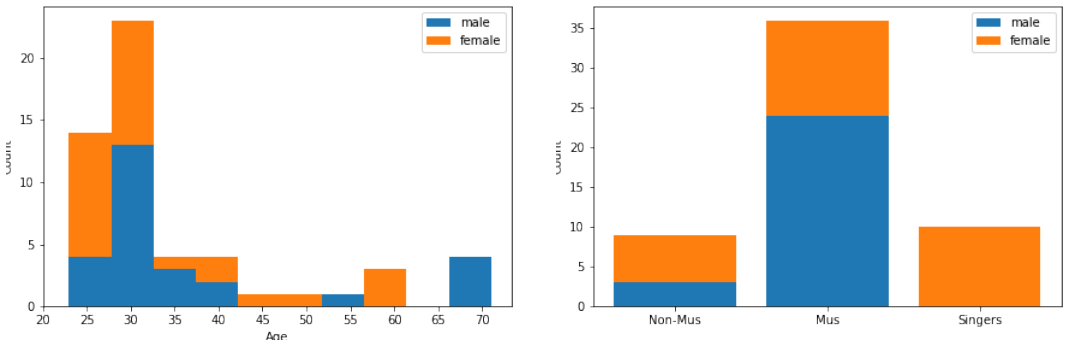
Appendix A

Listening Study Details from Chapter 5

A.1 GOLD-MSI Perceptual Ability Questions

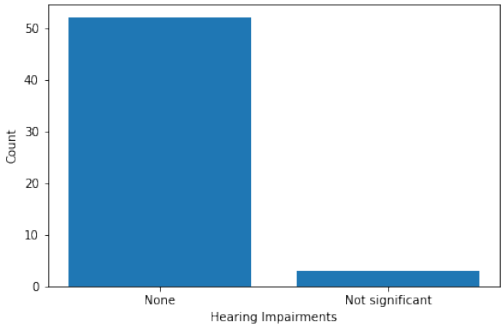
The following questions were extracted from the GOLD-MSI 'Perceptual Abilities' subset [Müllensiefen et al., 2014], and use a 7-point agreement scale:

- I am able to judge whether someone is a good singer or not.
- I usually know when I'm hearing a song for the first time.
- I find it difficult to spot mistakes in a performance of a song even if I know the tune.
- I can compare and discuss differences between two performances or versions of the same piece of music.
- I have trouble recognising a familiar song when played in a different way or by a different performer.
- I can tell when people sing or play out of time with the beat.
- I can tell when people sing or play out of tune.
- When I sing, I have no idea whether I'm in tune or not.

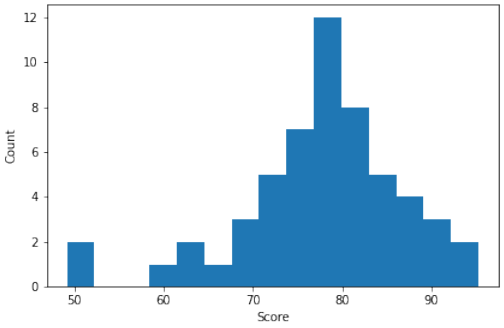


(a) Age

(b) Instrumental ability



(c) Hearing Impairments



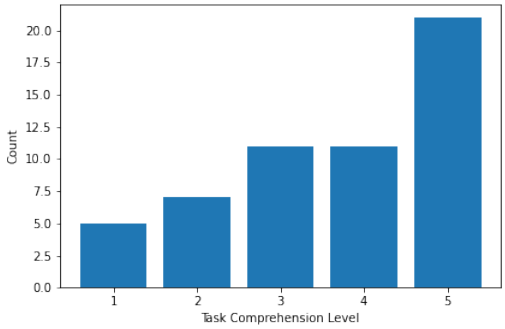
(d) MSI

Figure A.1: Bar graphs (subplots (b) and (c)) and distributions (subplots (a) and (d)) illustrating the spread of participant features. Subplot (b) presents non-musicians and musicians in their abbreviated form 'Non-Mus' and 'Mus' respectively. Subplot (c) omits the third option 'significant', as these participants would have been filtered out during the screening processes.

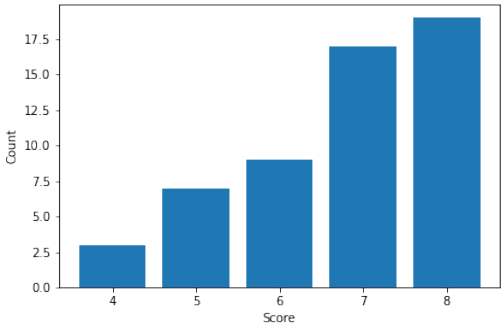
- When I hear music I can usually identify its genre.

A.2 Participant Feature Distributions

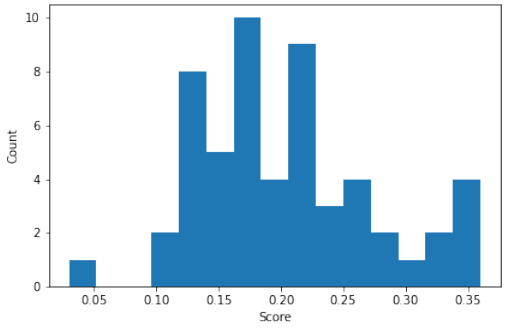
Figure A.1 displays a set of bar graphs and distributions generated from participants' answers to the pre-experiment questions. Figure A.2 displays similar graphs w.r.t. participant scores generated from their submitted dissimilarity ratings, as described in Section 5.2.7.



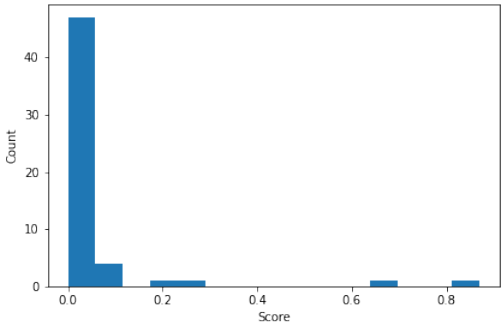
(a) Task Comprehension



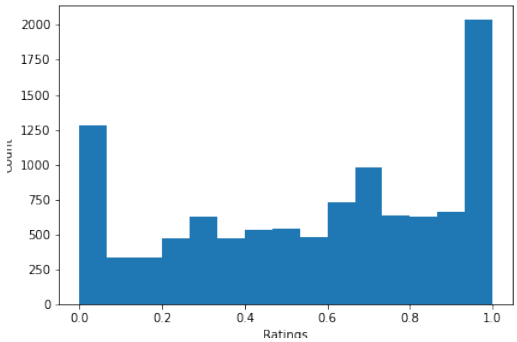
(b) Listening Potential



(c) Inconsistency



(d) Poor Identity Recognition



(e) Pairwise Distance Ratings

Figure A.2: Subplots *(a)* to *(e)* present bar graphs and distributions for participant scores generated from their dissimilarity ratings. Subplot *(e)* presents the distribution of ratings across all participants.

Appendix B

Listening Study Details from Chapter 6

B.1 STC Listening Study Questions

The following list presents the questions given to participants for the listening study of Chapter 6. Potential answers of a multiple choice format (or answers required by participants) are shown in the square brackets beside each question. Questions 1-6 were asked before the practice round. Question 7 was asked at the end of the study:

1. Please confirm that you do not have any hearing impairments that you think could affect your ability to listen to and evaluate audio clips. [Confirm - required by participant]

2. Please confirm you are using a computer and not a phone/tablet for this study, and that you are either using Safari or Chrome as your browser. If you need to change your setup, please do so before continuing and refresh this page. [Confirm - required by participant]

3. Please indicate what listening equipment you intend to use for this experiment (Headphones are preferable). If you wish to change your setup, please do so before continuing and refresh this page [Inbuilt speakers, external speaker, ear/headphones]

4. Please provide your age in the space below. [Integer - required by participant]

5. Please describe your gender as it applies to you in the space provided, or leave it blank if you prefer not to disclose. [String - optionally provided by participant]

6. Please select one option that best describes your relationship with music/audio:

    - I don't have a particular interest in it
    - I enjoy listening to it
    - I am an amateur musician/audio engineer
    - I am studying it
    - It is the basis of my profession

7. Thank you for your answers. Do you have any other comments regarding your evaluations, or any other aspect of the study you think the researcher should be aware of?

Bibliography

Ehab A. AlBadawy and Siwei Lyu. Voice conversion using speech-to-speech neuro-style transfer. In *Proceedings of INTERSPEECH*, pages 4726–4730, Shanghai, China, October 2020. ISCA. doi: 10.21437/Interspeech.2020-3056.

Matthew Amodio and Smita Krishnaswamy. TraVeLGAN: Neural machine translation by jointly learning to align and translate. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8975–8984, Long Beach, CA, USA, June 2019. IEEE. doi: 10.1109/CVPR.2019.00919.

Farzana Anowar, Samira Sadaoui, and Bassant Selim. Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE). *Computer Science Review*, 40(1): 100378, May 2021. ISSN 15740137. doi: 10.1016/j.cosrev.2021.100378.

Riku Arakawa, Shinnosuke Takamichi, and Hiroshi Saruwatari. Implementation of DNN-based real-time voice conversion and its improvements by audio data augmentation and mask-shaped device. In *10th ISCA Workshop on Speech Synthesis (SSW 10)*, pages 93–98. ISCA, September 2019. doi: 10.21437/SSW.2019-17.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473 [cs, stat]*, May 2016.

Hideki Banno, Hiroaki Hata, Masanori Morise, Toru Takahashi, Toshio Irino, and Hideki Kawahara. Implementation of realtime STRAIGHT speech manipula-

tion system: Report on its first implementation. *Acoustical Science and Technology*, 28(3):140–146, 2007. doi: 10.1250/ast.28.140.

Sakya Basak, Shrutina Agarwal, Sriram Ganapathy, and Naoya Takahashi. End-to-end lyrics recognition with voice to singing style transfer. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 266–270, Toronto, ON, Canada, June 2021. IEEE. doi: 10.1109/ICASSP39728.2021.9415096.

Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1):20–29, June 2004. ISSN 1931-0145, 1931-0153. doi: 10.1145/1007730.1007735.

Jason Bell. *MACHINE LEARNING: Hands-on for Developers and Technical Professionals*. John Wiley & Sons, 2020. ISBN 978-1-119-64225-1.

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 19. Curran Associates, Inc., 2006.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(2):281–305, 2012.

Franca Bittner, Marcel Gonzalez, Maike L. Richter, Hanna Lukashevich, and Jakob Abeßer. Multi-pitch estimation meets microphone mismatch: Applicability of domain adaptation. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru (India), 2022.

Rachel Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Bello. Medleydb: A multitrack dataset for annotation-intensive MIR research. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, page 6, Taipei (Taiwan), 2014.

Rachel M Bittner, Julia Wilkins, Hanna Yip, and Juan P Bello. Medleydb 2.0 : New data and a system for sustainable data collection. In *Proceedings of*

the International Society for Music Information Retrieval Conference (ISMIR), page 3, New York City (USA), 2016.

Merlijn Blaauw and Jordi Bonada. A neural parametric singing synthesizer modeling timbre and expression from natural songs. *Applied Sciences*, 7(12):1313, 2018. doi: 10.3390/app7121313.

Dawn A. A. Black, Li Ma, and Mi Tian. Automatic identification of emotional cues in chinese opera singing. In *Proceedings of International Conference on Music Perception and Cognition and the 5th Conference for the Asian-Pacific Society for Cognitive Sciences of Music*, Seoul, South Korea, 2014.

Ken Black. *Business Statistics: For Contemporary Decision Making, 8th Edition*. John Wiley & Sons, 2023.

Michael Blomgren, Yang Chen, Manwa L. Ng, and Harvey R. Gilbert. Acoustic, aerodynamic, physiologic, and perceptual properties of modal and vocal fry registers. *The Journal of the Acoustical Society of America*, 103(5):2649–2658, 1998. ISSN 0001-4966. doi: 10.1121/1.422785.

B. P. Bogert, M. J. R. Healy, and J. W. Tukey. The quefrency alanysis of time series for echoes: Cepstrum, pseudo autocovariance, cross-cepstrum and saphe cracking. In *Proceedings of the Symposium on Time Series Analysis*, 1963.

Jordi Bonada and Merlijn Blaauw. Semi-supervised learning for singing synthesis timbre. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7083–7087, Toronto, ON, Canada, June 2021. IEEE. doi: 10.1109/ICASSP39728.2021.9413400.

Jordi Bonada, Martí Umbert, and Merlijn Blaauw. Expressive singing synthesis based on unit selection for the singing synthesis challenge 2016. In *Interspeech*, pages 1230–1234, San francisco, United States, 2016. doi: 10.21437/interspeech.2016-872.

A Bouhuys, D F Proctor, and J Mead. Kinetic aspects of singing. *Journal of Applied Physiology*, 21(2):483–496, 1966. ISSN 8750-7587, 1522-1601. doi: 10.1152/jappl.1966.21.2.483.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

Zexin Cai, Chuxiong Zhang, and Ming Li. From speaker verification to multispeaker speech synthesis, deep transfer with feedback constraint. In *Proceedings of INTERSPEECH*, Shanghai, China, August 2020. ISCA.

Antoine Caillon and Philippe Esling. RAVE: A variational autoencoder for fast and high-quality neural audio synthesis. *arXiv preprint arXiv:2111.05011*, December 2021.

Peter J. Cameron. *Combinatorics: Topics, Techniques, Algorithms*. Cambridge University Press, October 1994. ISBN 978-1-107-39337-0.

Chris Cannam, Mark Sandler, Michael O. Jewell, Christophe Rhodes, and Mark d'Inverno. Linked data and you: Bringing music research software into the semantic web. *Journal of New Music Research*, 39(4):313–325, December 2010. ISSN 0929-8215. doi: 10.1080/09298215.2010.522715.

J. Douglas Carroll and J Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319, 1970. ISSN 1860-0980. doi: 10.1007/BF02310791.

E. C. Carterette and J. R. Miller. Perceptual space for musical structures. *The Journal of the Acoustical Society of America*, 56(S1):S44–S44, 1974. ISSN 0001-4966. doi: 10.1121/1.1914187.

Pritish Chandna. *Neural Networks for Singing Voice Extraction in Monaural Polyphonic Music Signals*. PhD thesis, Universitat Pompeu Fabra, September 2021.

Pritish Chandna, Merlijn Blaauw, Jordi Bonada, and Emilia Gómez. WGANSing: A multi-voice singing voice synthesizer based on the Wasserstein-GAN. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pages 1–5. EURASIP, September 2019. doi: 10.23919/EUSIPCO.2019.8903099.

Pritish Chandna, Merlijn Blaauw, Jordi Bonada, and Emilia Gómez. Content based singing voice extraction from a musical mixture. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 781–785, Barcelona, Spain, May 2020. IEEE. doi: 10.1109/ICASSP40776.2020.9053024.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, June 2020.

Xin Chen, Wei Chu, Jinxi Guo, and Ning Xu. Singing voice conversion with non-parallel data. In *Proceedings of the Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 292–296. IEEE, March 2019. doi: 10.1109/MIPR.2019.00059.

Tian Cheng and Masataka Goto. Transformer-based beat tracking with low-resolution encoder and high-resolution decoder. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Milan (Italy), 2023.

K. Choi, G. Fazekas, M. Sandler, and K. Cho. Convolutional recurrent neural networks for music classification. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2392–2396, New Orleans, LA, USA, March 2017. IEEE. doi: 10.1109/ICASSP.2017.7952585.

Keunwoo Choi, Janne Spijkervet, Minz Won, and Ashis Pati. Music classification: Beyond supervised learning, towards real-world applications. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Online, November 2021.

François Chollet. *Deep Learning with Python*. Manning Publications Co, Shelter Island, New York, 2018. ISBN 978-1-61729-443-3.

Ju-chieh Chou, Cheng-chieh Yeh, Hung-yi Lee, and Lin-shan Lee. Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations. In *Proceedings of INTERSPEECH*, pages 501–505, Hyderabad, India, 2018. ISCA. doi: 10.21437/Interspeech.2018-1830.

Barry H. Cohen. *Explaining Psychological Statistics*. John Wiley & Sons, 2008. ISBN 978-0-470-00718-1.

Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Routledge, Hillsdale, N.J, 2 edition edition, July 1988. ISBN 978-0-8058-0283-2.

Yandre M. G. Costa, Luiz S. Oliveira, and Carlos N. Silla. An evaluation of convolution neural networks for music classification using spectrograms. *Applied Soft Computing*, 52(1):28–38, March 2017. ISSN 1568-4946. doi: 10.1016/j.asoc.2016.12.024.

Eduardo Coutinho, Klaus R. Scherer, and Nicola Dibben. Singing and emotion. In *The Oxford Handbook of Singing*, pages 296–314. Oxford University Press, Oxford, UK, 2014.

Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019.

Xiaodong Cui, Vaibhava Goel, and Brian Kingsbury. Data augmentation for deep neural network acoustic modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(9):1469–1477, September 2015. ISSN 2329-9290, 2329-9304. doi: 10.1109/TASLP.2015.2438544.

Maximilian Damböck, Richard Vogl, and Peter Knees. On the impact and interplay of input representations and network architectures for automatic music tagging. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru (India), 2022.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, page 9. PMLR, 2017.

Matthew E. P. Davies, Philippe Hamel, Kazuyoshi Yoshii, and Masataka Goto. AutoMashUpper: Automatic creation of multi-song music mashups. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12): 1726–1737, December 2014. ISSN 2329-9304. doi: 10.1109/TASLP.2014.2347135.

Emir Demirel. *Deep Neural Networks for Automatic Lyrics Transcription*. PhD thesis, Queen Mary University of London, London, UK, 2022.

Emir Demirel, Sven Ahlbäck, and Simon Dixon. Automatic lyrics transcription using dilated convolutional neural networks with self-attention. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, July 2020. doi: 10.1109/IJCNN48605.2020.9207052.

Chengqi Deng, Chengzhu Yu, Heng Lu, Chao Weng, and Dong Yu. Pitchnet: Unsupervised singing voice conversion with pitch adversarial network. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7749–7753, Barcelona, Spain, May 2020. IEEE. doi: 10.1109/ICASSP40776.2020.9054199.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv.1810.04805*, May 2019.

Nat Dilokthanakul, Pedro A. M. Mediano, Marta Garnelo, Matthew C. H. Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with Gaussian Mixture Variational Autoencoders. *arXiv preprint arXiv:1611.02648*, January 2017.

Simon Dixon and Emmanouil Benetos. Music Informatics (Lectures), 2020.

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *International Conference on Learning Representations*, 2017.

Hongqiang Du, Xiaohai Tian, Lei Xie, and Haizhou Li. Optimizing voice conversion network with cycle consistency loss of speaker identity. In *Spoken Language Technology Workshop (SLT)*, pages 507–513. IEEE, January 2021. doi: 10.1109/SLT48900.2021.9383567.

Zhiyan Duan, Haotian Fang, Bo Li, Khe Chai Sim, and Ye Wang. The NUS sung and spoken lyrics corpus: A quantitative comparison of singing and speech. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1–9. IEEE, October 2013. doi: 10.1109/APSIPA.2013.6694316.

Homer Dudley. The carrier nature of speech. *Bell System Technical Journal*, 19 (4):495–515, 1940. ISSN 1538-7305. doi: 10.1002/j.1538-7305.1940.tb00843.x.

Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, January 2018.

Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts. DDSP: Differentiable digital signal processing. In *Proceedings of the International Conference on Learning Representations (ICLR)*. PMLR, 2019.

Philippe Esling, Axel Chemla–Romeu-Santos, and Adrien Bitton. Generative timbre spaces: Regularizing variational autoencoders with perceptual metrics. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris (France), 2018.

Ziqi Fan, Yuanbo Wu, Changwei Zhou, Xiaojun Zhang, and Zhi Tao. Class-imbalanced voice pathology detection and classification using fuzzy cluster oversampling method. *Applied Sciences*, 11(8):3450, January 2021.

Fuming Fang, Junichi Yamagishi, Isao Echizen, and Jaime Lorenzo-Trueba. High-quality nonparallel voice conversion based on cycle-consistent adversarial network. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5279–5283, Calgary, AB, Canada, April 2018. IEEE. doi: 10.1109/ICASSP.2018.8462342.

Guolin Fang. Assessing vocoders for English and Icelandic. Master's thesis, Reykjavík University, Reykjavík, Iceland, 2021.

Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27 (8):861–874, June 2006. ISSN 01678655. doi: 10.1016/j.patrec.2005.10.010.

Catherine O. Fritz, Peter E. Morris, and Jennifer J. Richler. Effect size estimates: Current use, calculations, and interpretation. *Journal of experimental psychology: General*, 141(1):2–18, 2012.

Isabel García-López and Javier Gavilán Bouzas. The singing voice. *Acta Otorrinolaringologica (English Edition)*, 61(6):441–451, 2010. ISSN 2173-5735. doi: 10.1016/S2173-5735(10)70082-X.

Sami Gazzah and Najoua Essoukri Ben Amara. New oversampling approaches based on polynomial fitting for imbalanced data sets. In *International Workshop on Document Analysis Systems*, pages 677–684. IAPR, September 2008. doi: 10.1109/DAS.2008.74.

Bruce R. Gerratt and Jody Kreiman. Toward a taxonomy of nonmodal phonation. *Journal of Phonetics*, 29(4):365–381, 2001. ISSN 0095-4470. doi: 10.1006/jpho.2001.0149.

Bernard Gold, Nelson Morgan, and Daniel P. W. Ellis. *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. Wiley-Blackwell, Oxford, 2nd ed edition, 2011. ISBN 978-0-470-19536-9.

Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv.1701.00160*, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, pages 2672–2680. Curran Associates, Inc., 2014.

Judith Greene and Manuela D'Oliveira. *Learning To Use Statistical Tests In Psychology*. McGraw-Hill Education (UK), December 2005. ISBN 978-0-335-21680-2.

John M. Grey. Multidimensional perceptual scaling of musical timbres. *The Journal of the Acoustical Society of America*, 61(5):1270–1277, 1977. ISSN 0001-4966. doi: 10.1121/1.381428.

D. Griffin and J. Lim. Signal estimation from modified short-time Fourier transform. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 8, pages 804–807, Boston, MA, USA, April 1983. IEEE. doi: 10.1109/ICASSP.1983.1172092.

T. Hasan and J. H.L. Hansen. A study on universal background model training in speaker verification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 19(7):1890–1899, September 2011. ISSN 1558-7916. doi: 10.1109/TASL.2010.2102753.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, 2015. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.90.

Kate Heidemann. A system for describing vocal timbre in popular song. *Music Theory Online*, 22(1), March 2016. ISSN 1067-3040. doi: 10.30535/mto.22.1.2.

Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson. CNN architectures for large-scale audio classification. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135, New Orleans, LA, USA, March 2017. IEEE. doi: 10.1109/ICASSP.2017.7952132.

W. Hess. *Pitch Determination of Speech Signals: Algorithms and Devices*. Springer Science & Business Media, December 2012. ISBN 978-3-642-81926-1.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735.

Harry Hollien. On vocal registers. *Journal of Phonetics*, 2(2):125–143, March 1974. ISSN 00954470. doi: 10.1016/S0095-4470(19)31188-X.

Adery C. A. Hope. A simplified monte carlo significance test procedure. *Journal of the Royal Statistical Society: Series B (Methodological)*, 30(3):582–598, 1968. ISSN 2517-6161. doi: 10.1111/j.2517-6161.1968.tb00759.x.

Jeremy Howard and Sylvain Gugger. *Deep Learning for Coders with Fastai and PyTorch*. O'Reilly Media, Inc., June 2020. ISBN 978-1-4920-4549-6.

Chao-Ling Hsu and Jyh-Shing Roger Jang. On the improvement of singing voice separation for monaural recordings using the mir-1k dataset. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 18(2):310–319, February 2010. ISSN 1558-7924. doi: 10.1109/TASL.2009.2026503.

Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang. Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks. In *Proceedings of INTERSPEECH*, pages 3364–3368, Stockholm, Sweden, August 2017. ISCA. doi: 10.21437/Interspeech.2017-63.

Wei-Ning Hsu, Yu Zhang, Ron J Weiss, Heiga Zen, Yonghui Wu, Yuxuan Wang, Yuan Cao, Ye Jia, Zhifeng Chen, Jonathan Shen, Patrick Nguyen, and Ruoming Pang. Hierarchical generative modeling for controllable speech synthesis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, page 27. PMLR, 2019.

Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. Singing-voice separation from monaural recordings using deep recurrent neu-

ral networks. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Taipei (Taiwan), 2014.

Rongjie Huang, Feiyang Chen, Yi Ren, Jinglin Liu, Chenye Cui, and Zhou Zhao. Multi-singer: Fast multi-singer singing voice vocoder with a large-scale corpus. In *ACM International Conference on Multimedia*. ACM Press, December 2021a.

Wen-Chin Huang, Tomoki Hayashi, Yi-Chiao Wu, Hirokazu Kameoka, and Tomoki Toda. Pretraining techniques for sequence-to-sequence voice conversion. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29 (1):745–755, 2021b. ISSN 2329-9304. doi: 10.1109/TASLP.2021.3049336.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, March 2015.

ISCA. Blizzard Challenge. https://www.synsig.org/index.php/Blizzard_Challenge_2023, 2023.

ITU-R. *Method for the Subjective Assessment of Intermediate Quality Level of Audio Systems*. ITU-R Recommendation BS.1116-3, 2015.

ITU-T. *Methods for Objective and Subjective Assessment of Speech Quality - Mean Opinion Score Interpretation and Reporting*. ITU-T Recommendation P.800.2, 2013.

Paul Iverson and Carol L. Krumhansl. Isolating the dynamic attributes of musical timbre. *The Journal of the Acoustical Society of America*, 94(5):2595–2603, 1993. ISSN 0001-4966. doi: 10.1121/1.407371.

Navdeep Jaitly and Geoffrey E Hinton. Vocal Tract Length Perturbation (VTLP) improves speech recognition. In *ICML Workshop on Deep Learning for Audio, Speech and Language*, 2013.

Aren Jansen, Jort F. Gemmeke, Daniel P. W. Ellis, Xiaofeng Liu, Wade Lawrence, and Dylan Freedman. Large-scale audio event discovery in one million

YouTube videos. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 786–790, New Orleans, LA, USA, March 2017. IEEE. ISBN 978-1-5090-4117-6. doi: 10.1109/ICASSP.2017.7952263.

A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde. Singing voice separation with deep U-Net convolutional networks. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017.

Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, and Yonghui Wu. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31. Curran Associates, Inc., 2018.

Nicholas Jillings, David Moffat, Brecht De Man, and Joshua D. Reiss. Web audio evaluation tool: A browse-based listening environment. In *Proceedings of the Sound and Music Computing Conference (SMC)*, Maynooth, Ireland, 2015.

Justin Johnson, Alexandre Alahi, Li Fei-Fei, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision – ECCV 2016: 14th European Conference*, Lecture Notes in Computer Science, pages 694–711, Amsterdam, The Netherlands, 2016. Springer International Publishing. ISBN 978-3-319-46475-6. doi: 10.1007/978-3-319-46475-6_43.

Stephen C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967. ISSN 1860-0980. doi: 10.1007/BF02289588.

Vedant Kalbag and Alexander Lerch. Scream detection in heavy metal music. *arXiv preprint arXiv:2205.05580*, May 2022.

Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron Oord, Sander Dieleman, and Koray Kavukcuoglu. Efficient neural audio synthesis. In *Proceedings*

of the International Conference on Machine Learning (ICML), pages 2410–2419. PMLR, July 2018.

Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, and Nobukatsu Hojo. ACVAE-VC: Non-parallel many-to-many voice conversion with auxiliary classifier variational autoencoder. In *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, October 2020.

Takuhiro Kaneko and Hirokazu Kameoka. Parallel-data-free voice conversion using cycle-consistent adversarial networks. *arXiv preprint arXiv:1711.11293*, November 2017.

Takuhiro Kaneko, Hirokazu Kameoka, Kaoru Hiramatsu, and Kunio Kashino. Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks. In *Proceedings of INTERSPEECH*, pages 1283–1287, Stockholm, Sweden, August 2017. ISCA. doi: 10.21437/Interspeech.2017-970.

Hideki Kawahara, Ikuyo Masuda-Katsuse, and Alain de Cheveigné. Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds. *Speech Communication*, 27(3):187–207, April 1999. ISSN 0167-6393. doi: 10.1016/S0167-6393(98)00085-5.

Gillyanne Kayes. *How Does Genre Shape Thee Vocal Performance of Female Singers?* PhD thesis, Institute of Education University of London, London, 2015.

Khan Academy. Khan Academy Tutorials. https://www.youtube.com/@khanacademy, 2006.

J. W. Kim, J. Salamon, P. Li, and J. P. Bello. Crepe: A convolutional representation for pitch estimation. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 161–165, Calgary, AB, Canada, April 2018. IEEE. doi: 10.1109/ICASSP.2018.8461329.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv.1412.6980*, 2014.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv.1312.6114*, 2014.

T. Kobayashi and S. Imai. Spectral analysis using generalised cepstrum. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(6):1235–1238, December 1984. ISSN 0096-3518. doi: 10.1109/TASSP.1984.1164454.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 17022–17033. Curran Associates, Inc., 2020a.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *Proceedings of the International Conference on Learning Representations (ICLR)*. PMLR, 2020b.

György Kovács. An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets. *Applied Soft Computing*, 83(1):105662, October 2019. ISSN 1568-4946. doi: 10.1016/j.asoc.2019.105662.

Jody Kreiman, Bruce Gerratt, Gail Kempster, Andrew Erman, and Gerald Berke. Perceptual evaluation of voice quality. *Journal of Speech Language and Hearing Research*, 36(1):21–40, February 1993. doi: 10.1044/jshr.3601.21.

J. Krimphoff, S. McAdams, and S. Winsberg. Caractérisation du timbre des sons complexes. II. Analyses acoustiques et quantification psychophysique (French) [Characterisation of the timbre of complex sounds. II. Acoustic analysis and psychophysical quantification]. *Le Journal de Physique IV*, 04(5):625–628, 1994. ISSN 1155-4339. doi: 10.1051/jp4:19945134.

J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964. ISSN 0033-3123, 1860-0980. doi: 10.1007/BF02289565.

William H. Kruskal and W. Allen Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621, 1952. ISSN 0162-1459. doi: 10.2307/2280779.

Ahlad Kumar. Deep Learning Tutorials:19-34 Variational Autoencoders and GANs. https://www.youtube.com/watch?v=w8F7_rQZxXk, 2019.

Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brebisson, Yoshua Bengio, and Aaron Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32. Curran Associates, Inc., October 2019.

Matt J. Kusner and José Miguel Hernández-Lobato. GANs for sequences of discrete elements with the gumbel-softmax distribution. *arXiv preprint arXiv:1611.04051*, (arXiv:1611.04051), 2016.

Ilse Bernadette Labuschagne and Valter Ciocca. The perception of breathiness: Acoustic correlates and the influence of methodological factors. *Acoustical Science and Technology*, 37(5):191–201, 2016. ISSN 1346-3969, 1347-5177. doi: 10.1250/ast.37.191.

Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1558–1566. PMLR, June 2016.

Yann LeCun, Leon Bottou, Genevieve B. Orr, and Klaus Robert Müller. Efficient Backprop. In Genevieve B. Orr and Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade*, Lecture Notes in Computer Science, pages 9–50. Springer, Berlin, Heidelberg, 1998. ISBN 978-3-540-49430-0. doi: 10.1007/3-540-49430-8_2.

Juheon Lee, Hyeong-Seok Choi, Chang-Bin Jeon, Junghyun Koo, and Kyogu Lee. Adversarially trained end-to-end korean singing voice synthesis system. *arXiv preprint arXiv:1908.01919*, August 2019.

Juheon Lee, Hyeong-Seok Choi, Junghyun Koo, and Kyogu Lee. Disentangling timbre and singing style with multi-singer singing synthesis system. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7224–7228, Barcelona, Spain, May 2020. IEEE. doi: 10.1109/ICASSP40776.2020.9054636.

Kyungyun Lee and Juhan Nam. Learning a joint embedding space of monophonic and mixed music signals for singing voice. *arXiv:1906.11139*, June 2019.

Sang-gil Lee, Heeseung Kim, Chaehun Shin, Xu Tan, Chang Liu, Qi Meng, Tao Qin, Wei Chen, Sungroh Yoon, and Tie-Yan Liu. Priorgrad: Improving conditional denoising diffusion models with data-dependent adaptive prior. In *Proceedings of the International Conference on Learning Representations (ICLR)*. PMLR, 2022.

Simon Leglaive, Romain Hennequin, and Roland Badeau. Singing voice detection with deep recurrent neural networks. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 121–125, South Brisbane, QLD, Australia, April 2015. IEEE. doi: 10.1109/ICASSP.2015.7177944.

Simon Leglaive, Xavier Alameda-Pineda, Laurent Girin, and Radu Horaud. A recurrent variational autoencoder for speech enhancement. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 371–375, Barcelona, Spain, May 2020. IEEE. doi: 10.1109/ICASSP40776.2020.9053164.

Yi Lei, Shan Yang, Jian Cong, Lei Xie, and Dan Su. Glow-WaveGAN 2: High-quality zero-shot text-to-speech synthesis and any-to-any voice conversion. *arXiv preprint arXiv.2207.01832*, July 2022. doi: 10.48550/arXiv.2207.01832.

Joseph Lemley, Shabab Bazrafkan, and Peter Corcoran. Smart augmentation learning an optimal data augmentation strategy. *IEEE Access*, 5:5858–5869, 2017. ISSN 2169-3536. doi: 10.1109/ACCESS.2017.2696121.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART:

Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703.

Dayong Li, Xian Li, and Xiaofei Li. DVQVC: An unsupervised zero-shot voice conversion framework. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, Rhodes Island, Greece, June 2023. IEEE. doi: 10.1109/ICASSP49357.2023.10095393.

Rui Li, Dong Pu, Minnie Huang, and Bill Huang. UNET-TTS: Improving unseen speaker and style transfer in one-shot voice cloning. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8327–8331, Singapore, Singapore, May 2022a. IEEE. doi: 10.1109/ICASSP43922.2022.9746049.

Xu Li, Shansong Liu, and Ying Shan. A hierarchical speaker representation framework for one-shot singing voice conversion. In *Proceedings of INTERSPEECH*, pages 4307–4311, Incheon, Korea, September 2022b. ISCA. doi: 10.21437/Interspeech.2022-11305.

Yinghao Aaron Li, Ali Zare, and Nima Mesgarani. StarGANv2-VC: A diverse, unsupervised, non-parallel framework for natural-sounding voice conversion. *arXiv preprint arXiv:2107.10394*, July 2021a.

Zhonghao Li, Benlai Tang, Xiang Yin, Yuan Wan, Ling Xu, Chen Shen, and Zejun Ma. PPG-based singing voice conversion with adversarial representation learning. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7073–7077, Toronto, ON, Canada, June 2021b. IEEE. doi: 10.1109/ICASSP39728.2021.9414137.

Yist Y. Lin, Chung-Ming Chien, Jheng-Hao Lin, Hung-yi Lee, and Lin-shan Lee. Fragmentvc: Any-to-any voice conversion by end-to-end extracting and fusing fine-grained voice fragments with attention. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages

5939–5943, Toronto, ON, Canada, June 2021. IEEE. ISBN 978-1-72817-605-5. doi: 10.1109/ICASSP39728.2021.9413699.

Jinglin Liu, Chengxi Li, Yi Ren, Feiyang Chen, and Zhou Zhao. DiffSinger: Singing voice synthesis via shallow diffusion mechanism. In *Proceedings of the Conference on Artificial Intelligence*, volume 36 of *10*, pages 11020–11028. AAAI Press, March 2022.

Songxiang Liu, Jinghua Zhong, Lifa Sun, Xixin Wu, Xunying Liu, and Helen Meng. Voice conversion across arbitrary speakers based on a single target-speaker utterance. In *Proceedings of INTERSPEECH*, pages 496–500, Hyderabad, India, September 2018. ISCA. doi: 10.21437/Interspeech.2018-1504.

Songxiang Liu, Yuewen Cao, Na Hu, Dan Su, and Helen Meng. Fastsvc: Fast cross-domain singing voice conversion with feature-wise linear modulation. In *Proceedings of the International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, July 2021a. doi: 10.1109/ICME51207.2021.9428161.

Songxiang Liu, Yuewen Cao, Dan Su, and Helen Meng. DiffSVC: A diffusion probabilistic model for singing voice conversion. In *Proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 741–748. IEEE, May 2021b.

Zhijun Liu, Yiwei Guo, and Kai Yu. Diffvoice: Text-to-speech with latent diffusion. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, Rhodes Island, Greece, 2023. IEEE. doi: 10.1109/ICASSP49357.2023.10095100.

Junchen Lu, Kun Zhou, Berrak Sisman, and Haizhou Li. VAW-GAN for singing voice conversion with non-parallel training data. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 514–519. IEEE, December 2020.

Wei-Tsung Lu, Ju-Chiang Wang, Minz Won, Keunwoo Choi, and Xuchen Song. SpecTNT: A time-frequency transformer for music audio. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Online, October 2021.

Luis Serrano. Serrano Academy Tutorial. https://www.youtube.com/@SerranoAcademy/videos, 2013.

Yin-Jyun Luo and Li Su. Learning domain-adaptive latent representations of music signals using variational autoencoders. In *Proceedings of the International Conference on Learning Representations (ICLR)*, page 8. PMLR, 2018.

Yin-Jyun Luo, Kat Agres, and Dorien Herremans. Learning disentangled representations of timbre and pitch for musical instrument sounds using gaussian mixture variational autoencoders. In *Proceedings of the International Conference on Learning Representations (ICLR)*. PMLR, 2019.

Yin-Jyun Luo, Kin Wai Cheuk, Tomoyasu Nakano, Masataka Goto, and Dorien Herremans. Unsupervised disentanglement of pitch and timbre for isolated musical instrument sounds. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Montréal (Canada), October 2020a.

Yin-Jyun Luo, Chin-Cheng Hsu, Kat Agres, and Dorien Herremans. Singing voice conversion with disentangled representations of singer and vocal technique using variational autoencoders. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3277–3281, Barcelona, Spain, May 2020b. IEEE. doi: 10.1109/ICASSP40776.2020.9054582.

Martin E. Malandro. Composer's assistant: An interactive transformer for multi-track midi infilling. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Milan (Italy), July 2023. doi: 10.48550/arXiv.2301.12525.

H. B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1):50–60, 1947. ISSN 0003-4851.

Matthias Mauch and Simon Dixon. PYIN: A fundamental frequency estimator using probabilistic threshold distributions. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 659–663, Florence, Italy, May 2014. IEEE. doi: 10.1109/ICASSP.2014.6853678.

Stephen McAdams, Jean-christophe Cunible, Robert P. Carlyon, C. J. Darwin, and Ian John Russell. Perception of timbral analogies. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 336(1278):383–389, 1992. doi: 10.1098/rstb.1992.0072.

Stephen McAdams, Suzanne Winsberg, Sophie Donnadieu, Geert De Soete, and Jochen Krimphoff. Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. *Psychological research*, 58(3):177–192, 1995.

Matthew C* McCallum, Filip Korzeniowski, Sergio Oramas, Fabien Gouyon, and Andreas Ehmann. Supervised and unsupervised learning of audio representations for music understanding. In *Proceedings of the International Conference on Learning Representations (ICLR)*. PMLR, December 2022.

Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, December 1943. ISSN 1522-9602. doi: 10.1007/BF02478259.

Adib Mehrabi. *Vocal Imitation for Query by Vocalisation*. PhD thesis, Queen Mary University of London, London, UK, 2018.

Gabriel Meseguer-Brocal, Alice Cohen-Hadria, and Geoffroy Peeters. Creating DALI, a large dataset of synchronized audio, lyrics, and notes. *Transactions of the International Society for Music Information Retrieval*, 3(1):55–67, June 2020. doi: 10.5334/tismir.30.

John F. Michel and Harry Hollien. Perceptual differentiation of vocal fry and harshness. *Journal of Speech Language and Hearing Research*, 11(2):439–443, June 1968. doi: 10.1044/jshr.1102.439.

Noam Mor, Lior Wolf, Adam Polyak, and Yaniv Taigman. A universal music translation network. In *Proceedings of the International Conference on Learning Representations (ICLR)*. PMLR, 2019.

Masanori Morise. PLATINUM: A method to extract excitation signals for voice synthesis system. *Acoustical Science and Technology*, 33(2):123–125, 2012. ISSN 1346-3969, 1347-5177. doi: 10.1250/ast.33.123.

Masanori Morise. CheapTrick, a spectral envelope estimator for high-quality speech synthesis — Elsevier Enhanced Reader. *Speech Communication*, 67: 1–7, September 2014. doi: 10.1016/j.specom.2014.09.003.

Masanori Morise. Harvest: A high-performance fundamental frequency estimator from speech signals. In *Proceedings of INTERSPEECH*, pages 2321–2325, Stockholm, Sweden, August 2017. ISCA. doi: 10.21437/Interspeech.2017-68.

Masanori Morise, Hideki Kawahara, and Haruhiro Katayose. Fast and reliable f0 estimation method based on the period extraction of vocal fold vibration of singing voice and speech. In *Audio Engineering Society Conference: 35th International Conference: Audio for Games*. Audio Engineering Society, February 2009.

Masanori Morise, Fumiya Yokomori, and Kenji Ozawa. WORLD: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Transactions on Information and Systems*, E99.D(7):1877–1884, 2016. ISSN 0916-8532, 1745-1361. doi: 10.1587/transinf.2015EDP7457.

M. Mörner, F. Fransson, and G. Fant. Voice register terminology and standard pitch. *STL-QPSR*, 4(4):17–23, 1963.

Youssef Mroueh, Tom Sercu, and Vaibhava Goel. McGan: Mean and covariance feature matching GAN. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2527–2535. PMLR, July 2017.

Marie E. Mugavin. Multidimensional scaling: A brief overview. *Nursing Research*, 57(1):64–68, 2008. ISSN 0029-6562. doi: 10.1097/01.NNR.0000280659.88760.7c.

Daniel Müllensiefen, Bruno Gingras, Jason Musil, and Lauren Stewart. The musicality of non-musicians: An index for assessing musical sophistication in the

general population. *PloS one*, 9(2):e89642, 2014. ISSN 1932-6203. doi: 10.1371/journal.pone.0089642.

Meinard Müller. *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. Springer, July 2015. ISBN 978-3-319-21945-5.

Peter J Mumby. Statistical power of non-parametric tests: A quick guide for designing sampling strategies. *Marine Pollution Bulletin*, 44(1):85–87, January 2002. ISSN 0025326X. doi: 10.1016/S0025-326X(01)00097-2.

Eliya Nachmani and Lior Wolf. Unsupervised Singing Voice Conversion. *arXiv preprint arXiv:1904.06590*, September 2019.

Eliya Nachmani, Adam Polyak, Yaniv Taigman, and Lior Wolf. Fitting new speakers based on a short untranscribed sample. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 3683–3691. PMLR, July 2018.

Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: A large-scale speaker identification dataset. In *Proceedings of INTERSPEECH*, pages 2616–2620, Stockholm, Sweden, August 2017. ISCA. doi: 10.21437/Interspeech.2017-950.

Shahan Nercessian. Zero-shot singing voice conversion. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 70–76, Montréal (Canada), 2020.

Shahan Nercessian. End-to-end zero-shot voice conversion using a DDSP vocoder. In *Proceedings of the Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 1–5. IEEE, October 2021. doi: 10.1109/WASPAA52581.2021.9632754.

Ralph Neuneier and Hans Georg Zimmermann. How to train neural networks. In Genevieve B. Orr and Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade*, Lecture Notes in Computer Science, pages 373–423. Springer, Berlin, Heidelberg, 1998. ISBN 978-3-540-49430-0. doi: 10.1007/3-540-49430-8_18.

Andrew Ng and StanfordOnline. Stanford Online Tutorials. https://www.youtube.com/@stanfordonline, 2009.

Francis Nolan. Intonation. In *The Handbook of English Linguistics*, chapter 20, pages 385–405. John Wiley & Sons, Ltd, 2020. ISBN 978-1-119-54061-8. doi: 10.1002/9781119540618.ch20.

Brendan O'Connor and Simon Dixon. A comparative analysis of latent regressor losses for singing voice conversion. In *Proceedings of the Sound and Music Computing Conference (SMC)*, pages 289–295, Stockholm, Sweden, June 2023. ISBN 978-91-527-7372-7. doi: 10.5281/zenodo.8136568.

Brendan O'Connor, Simon Dixon, and George Fazekas. An exploratory study on perceptual spaces of the singing voice. In *Proceedings of the Joint Conference on AI Music Creativity*, volume 1, Stockholm, Sweden, October 2020. ISBN 978-91-519-5560-5.

Brendan O'Connor, Simon Dixon, and George Fazekas. Zero-shot singing technique conversion. In *Proceedings of the International Symposium on Creativity and Music Multidisciplinary Research (CMMR)*, pages 235–244, Tokyo, 2021.

OpenAI. ChatGPT [Large language model]., 2023.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. ISSN 1558-2191. doi: 10.1109/TKDE.2009.191.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, South Brisbane, QLD, Australia, April 2015. IEEE. doi: 10.1109/ICASSP.2015.7178964.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Proceedings of INTER-*

SPEECH, pages 2613–2617, Graz, Austria, September 2019. ISCA. doi: 10.21437/Interspeech.2019-2680.

Marco Perugini, Marcello Gallucci, and Giulio Costantini. A practical primer to power analysis for simple experimental designs. *International Review of Social Psychology*, 31(1):1–23, July 2018. ISSN 2397-8570. doi: 10.5334/irsp.181.

Adam Polyak, Lior Wolf, Yossi Adi, Yaniv Taigman, Ehab A. AlBadawy, and Siwei Lyu. Unsupervised cross-domain singing voice conversion. In *Proceedings of INTERSPEECH*, pages 801–805, Shanghai, China, October 2020. ISCA. doi: 10.21437/Interspeech.2020-1862.

Jordi Pons, Oriol Nieto, Matthew Prockup, Erik M Schmidt, Andreas F Ehmann, and Xavier Serra. End-to-end learning for music audio tagging at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, page 5. Curran Associates, Inc., 2017.

Bima Prihasto, Yi-Xing Lin, Phuong Thi Le, Chien-Lin Huang, and Jia-Ching Wang. CNEG-VC: Contrastive learning using hard negative example in non-parallel voice conversion. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, Rhodes Island, Greece, June 2023. IEEE. doi: 10.1109/ICASSP49357.2023.10094995.

Magdalena Proszewska, Grzegorz Beringer, Daniel Sáez-Trigueros, Thomas Merritt, Abdelhamid Ezzerg, and Roberto Barra-Chicote. GlowVC: Mel-spectrogram space disentangling model for language-independent text-free voice conversion. *arXiv preprint arXiv:2207.01454*, July 2022.

Polina Proutskova. *Investigating the Singing Voice: Quantitative and Qualitative Approaches to Studying Cross-Cultural Vocal Production*. PhD thesis, Goldsmiths University of London, London, 2019.

Polina Proutskova, Christophe Rhodes, Tim Crawford, and Geraint Wiggins. Breathy, resonant, pressed – automatic detection of phonation mode from audio recordings of singing. *Journal of New Music Research*, 42(2):171–186, 2013. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298215.2013.821496.

Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson. AutoVC: Zero-shot voice style transfer with only autoencoder loss. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 97, pages 5210–5219. PMLR, 2019.

Kaizhi Qian, Zeyu Jin, Mark Hasegawa-Johnson, and Gautham J. Mysore. F0-consistent many-to-many non-parallel voice conversion via conditional autoencoder. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6284–6288, Barcelona, Spain, May 2020a. IEEE. doi: 10.1109/ICASSP40776.2020.9054734.

Kaizhi Qian, Yang Zhang, Shiyu Chang, David Cox, and Mark Hasegawa-Johnson. Unsupervised speech decomposition via triple information bottleneck. In *Proceedings of the International Conference on Machine Learning (ICML)*, page 11. PMLR, 2020b.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*. PMLR, January 2016.

Colin Raffel and Daniel P. W. Ellis. Feed-forward networks with attention can solve some long-term memory problems. *arXiv preprint arXiv:1512.08756*, September 2016.

Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Mimilakis Stylianos Ioannis, and Rachel Bittner. The MUSDB18 corpus for music separation. December 2017. doi: 10.5281/zenodo.1117372.

William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, December 1971. ISSN 0162-1459. doi: 10.1080/01621459.1971.10482356.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, volume 9351,

pages 234–241. Springer International Publishing, Cham, 2015. ISBN 978-3-319-24573-7 978-3-319-24574-4. doi: 10.1007/978-3-319-24574-4_28.

Sebastian Rosenzweig, Helena Cuesta, Christof Weiß, Frank Scherbaum, Emilia Gómez, and Meinard Müller. Dagstuhl choirset: A multitrack dataset for MIR research on choral singing. *Transactions of the International Society for Music Information Retrieval*, 3(1):98–110, July 2020. doi: 10.5334/tismir.48.

Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Series in Artificial Intelligence. Elsevier, 4 edition, 2021. ISBN 978-1-292-40117-1.

Kristin L. Sainani. Dealing with non-normal data. *PM&R*, 4(12):1001–1005, December 2012. ISSN 1934-1482. doi: 10.1016/j.pmrj.2012.10.013.

Takeshi Saitou, Naoya Tsuji, Masashi Unoki, and Masato Akagi. Analysis of acoustic features affecting "singing-ness" and its application to singing-voice synthesis from speaking-voice. In *Proceedings of INTERSPEECH*, pages 1925–1928, Jeju Island, Korea, October 2004. ISCA. doi: 10.21437/Interspeech.2004-476.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, pages 2234–2242. Curran Associates, Inc., 2016.

J.W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5):401–409, May 1969. ISSN 1557-9956. doi: 10.1109/T-C.1969.222678.

S. Sarkar, E. Benetos, and M. Sandler. EnsembleSet: A new high-quality synthesised dataset for chamber ensemble separation. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru (India), December 2022.

Richard Savery, Lisa Zahray, and Gil Weinberg. Emotional musical prosody: Validated vocal dataset for human robot interaction. In *Proceedings of the Joint*

Conference on AI Music Creativity, Sweden, Stockholm, 2020. doi: 10.30746/978-91-519-5560-5.

K Scherer. Vocal communication of emotion: A review of research paradigms. *Speech Communication*, 40(1-2):227–256, April 2003. ISSN 01676393. doi: 10.1016/S0167-6393(02)00084-5.

Klaus R. Scherer, Johan Sundberg, Bernardino Fantini, Stéphanie Trznadel, and Florian Eyben. The expression of emotion in the singing voice: Acoustic patterns in vocal performance. *The Journal of the Acoustical Society of America*, 142(4):1805–1815, 2017. ISSN 0001-4966. doi: 10.1121/1.5002886.

Jan Schlüter and Thomas Grill. Exploring data augmentation for improved singing voice detection with neural networks. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, page 6, Malaga, Spain, 2015.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 86–96, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1009.

Sandra Serafini. *Timbre Perception of Cultural Insiders: A Case Study With Javanese Gamel an Instruments*. PhD thesis, University of British Columbia, Vancouver, British Columbia, Canada, 1993.

S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965. ISSN 0006-3444. doi: 10.2307/2333709.

Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, Rif A. Saurous, Yannis Agiomvrgiannakis, and Yonghui Wu. Natural TTS synthesis by conditioning Wavenet on Mel spectrogram predictions. In *Proceedings of the International Conference on Acoustics, Speech and Signal Pro-*

cessing (ICASSP), pages 4779–4783, Calgary, AB, Canada, April 2018. IEEE. doi: 10.1109/ICASSP.2018.8461368.

Kai Shen, Zeqian Ju, Xu Tan, Yanqing Liu, Yichong Leng, Lei He, Tao Qin, Sheng Zhao, and Jiang Bian. NaturalSpeech 2: Latent diffusion models are natural and zero-shot speech and singing synthesizers. *arXiv preprint arXiv:2304.09116*, April 2023.

Roger N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. I. *Psychometrika*, 27(2):125–140, 1962a. ISSN 1860-0980. doi: 10.1007/BF02289630.

Roger N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. II. *Psychometrika*, 27(3):219–246, 1962b. ISSN 1860-0980. doi: 10.1007/BF02289621.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016. ISSN 1476-4687. doi: 10.1038/nature16961.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*. PMLR, 2015.

RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron J Weiss, Rob Clark, and Rif A Saurous. Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron. In *Proceedings of Machine Learning Research*, volume 80, pages 4693–4702, 2018.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2256–2265. PMLR, June 2015.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR Pattern Recognition and Image Processing Workshop*, April 2015.

Josh Starmer. StatQuest with Josh Starmer Tutorials. https://www.youtube.com/@statquest, 2011.

S S Stevens, J Volkmann, and E B Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190, 1937.

Daniel Stoller. *Deep Learning for Music Information Retrieval in Limited Data Scenarios*. PhD thesis, Queen Mary University of London, London, UK, 2020.

Daniel Stoller, Sebastian Ewert, and Simon Dixon. Wave-U-Net: A multi-scale neural network for end-to-end audio source separation. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

Y. Stylianou. Voice Transformation: A survey. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3585–3588, Taipei, Taiwan, 2009. IEEE. doi: 10.1109/ICASSP.2009.4960401.

Lifa Sun, Kun Li, Hao Wang, Shiyin Kang, and Helen Meng. Phonetic posteriorgrams for many-to-one voice conversion without parallel data training. In *Proceedings of the International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, July 2016. doi: 10.1109/ICME.2016.7552917.

Johan Sundberg. The acoustics of the singing voice. *Scientific American*, 236(3): 82–91, 1977. ISSN 0036-8733. doi: 10.1038/scientificamerican0377-82.

Johan Sundberg. *The Science of the Singing Voice*. Cornell University Press, Dekalb, Ill, 1987. ISBN 978-0-87580-542-9.

Kohei Suzuki, Shoki Sakamoto, Tadahiro Taniguchi, and Hirokazu Kameoka. Speak like a dog: Human to non-human creature voice conversion. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual*

Summit and Conference (APSIPA ASC), pages 1388–1393. IEEE, November 2022. doi: 10.23919/APSIPAASC55919.2022.9980306.

Naoya Takahashi, Mayank Kumar, Singh, and Yuki Mitsufuji. Hierarchical diffusion models for singing voice neural vocoder. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, Rhodes Island, Greece, June 2023. IEEE. ISBN 978-1-72816-327-7. doi: 10.1109/ICASSP49357.2023.10095749.

Zhiyuan Tan, Jianguo Wei, Junhai Xu, Yuqing He, and Wenhuan Lu. Zero-shot voice conversion with adjusted speaker embeddings and simple acoustic features. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5964–5968, Toronto, ON, Canada, June 2021. IEEE. doi: 10.1109/ICASSP39728.2021.9414975.

Huaizhen Tang, Xulong Zhang, Jianzong Wang, Ning Cheng, and Jing Xiao. AVQVC: One-shot voice conversion by vector quantization with applying contrastive learning. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4613–4617, Singapore, Singapore, May 2022. IEEE. ISBN 978-1-66540-540-9. doi: 10.1109/ICASSP43922.2022.9746369.

Vibha Tiwari. MFCC and its applications in speaker recognition. *International Journal on Emerging Technologies*, 1(1):19–22, 2009.

Patrick Lumban Tobing, Yi-Chiao Wu, Tomoki Hayashi, Kazuhiro Kobayashi, and Tomoki Toda. Non-parallel voice conversion with cyclic variational autoencoder. In *Proceedings of INTERSPEECH*, pages 674–678, Graz, Austria, September 2019. ISCA. doi: 10.21437/Interspeech.2019-2307.

Tomoki Toda, Wen-Chin Huang, Lester Violeta, Songxiang Liu, and Jiatong Shi. Singing voice conversion challenge 2023. http://www.vc-challenge.org/, 2023.

Keiichi Tokuda, Takao Kobayashi, Takashi Masuko, and Satoshi Imai. Mel-generalized cepstral analysis - a unified approach to speech spectral estimation. In *Proceedings of the International Conference on Spoken Lan-*

guage Processing (ICSLP), pages 1043–1046. ISCA, September 1994. doi: 10.21437/ICSLP.1994-275.

Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein Autoencoders. In *Proceedings of the International Conference on Learning Representations (ICLR)*. PMLR, December 2019.

Keisuke Toyama, Taketo Akama, Yukara Ikemiya, Yuhta Takida, Wei-Hsiang Liao, and Yuki Mitsufuji. Automatic piano transcription with hierarchical frequency-time transformer. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Milan (Italy), July 2023. doi: 10.48550/arXiv.2307.04305.

Stefan Uhlich, Marcello Porcu, Franck Giron, Michael Enenkl, Thomas Kemp, Naoya Takahashi, and Yuki Mitsufuji. Improving music source separation based on deep neural networks through data augmentation and network blending. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 261–265, New Orleans, LA, USA, March 2017. IEEE. doi: 10.1109/ICASSP.2017.7952158.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, November 2017. doi: 10.48550/arXiv.1607.08022.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016a.

Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *arXiv:1601.06759 [Cs]*. PMLR, August 2016b.

Aaron Van den Oord, Oriol Vinyals, and koray Kavukcuoglu. Neural discrete representation learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pages 6306–6315. Curran Associates, Inc., 2017.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11):2579–2605, 2008. ISSN ISSN 1533-7928.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30. Curran Associates, Inc., 2017.

Christophe Veaux, Junichi Yamagishi, and Kirsten MacDonald. CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit (SUPERSEDED), April 2017.

Valerio Velardo. Mel-frequency cepstral coefficients explained easily, October 2020.

Sanna Wager, George Tzanetakis, Stefan Sullivan, Cheng-i Wang, John Shimmin, Minje Kim, and Perry Cook. Intonation: A dataset of quality vocal performances refined by spectral clustering on pitch congruence. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 476–480, Barcelona, Spain, May 2019. IEEE. ISBN 978-1-4799-8131-1. doi: 10.1109/ICASSP.2019.8683554.

Michael Wagner and Duane G. Watson. Experimental and theoretical advances in prosody: A review. *Language and Cognitive Processes*, 25(7-9):905–945, September 2010. ISSN 0169-0965. doi: 10.1080/01690961003589492.

Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883, Calgary, AB, Canada, April 2018. IEEE. doi: 10.1109/ICASSP.2018.8462665.

Chunhui Wang, Chang Zeng, and Xing He. HiFi-WaveGAN: Generative adversarial network with auxiliary spectrogram-phase loss for high-fidelity singing voice generation. *arXiv preprint arXiv:2210.12740*, October 2022.

Dongmei Wang, Xiong Xiao, Naoyuki Kanda, Takuya Yoshioka, and Jian Wu. Target speaker voice activity detection with transformers and its integration with end-to-end neural diarization. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, Rhodes Island, Greece, 2023a. IEEE. doi: 10.1109/ICASSP49357.2023.10095185.

Kai Wang, Yuhang Yang, Hao Huang, Ying Hu, and Sheng Li. Speakeraugment: Data augmentation for generalizable source separation via speaker parameter manipulation. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, Rhodes Island, Greece, June 2023b. IEEE. doi: 10.1109/ICASSP49357.2023.10094767.

Yuxuan Wang, R.J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: Towards end-to-end speech synthesis. In *Proceedings of INTERSPEECH*, pages 4006–4010, Stockholm, Sweden, August 2017. ISCA. doi: 10.21437/Interspeech.2017-1452.

Lage Wedin and Gunnar Goude. Dimension analysis of the perception of instrumental timbre. *Scandinavian Journal of Psychology*, 13(1):228–240, 1972. ISSN 1467-9450. doi: 10.1111/j.1467-9450.1972.tb00071.x.

Mirjam Wester, Zhizheng Wu, and Junichi Yamagishi. Analysis of the voice conversion challenge 2016 evaluation results. In *Interspeech 2016*, pages 1637–1641, September 2016. doi: 10.21437/Interspeech.2016-1331.

Felix Wiewel, Andreas Brendle, and Bin Yang. Continual learning through one-class classification using VAE. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3307–3311, Barcelona, Spain, May 2020. IEEE. doi: 10.1109/ICASSP40776.2020.9054743.

Julia Wilkins, Prem Seetharaman, Alison Wahl, and Bryan Pardo. Vocalset: A singing voice dataset. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 468–474, Paris (France), 2018.

Suzanne Winsberg and Geert De Soete. A latent class approach to fitting the weighted Euclidean model, clascal. *Psychometrika*, 58(2):315–330, 1993. ISSN 1860-0980. doi: 10.1007/BF02294578.

Minz Won, Keunwoo Choi, and Xavier Serra. Semi-supervised music tagging transformer. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, page 8, Online, 2021.

Da-Yi Wu and Hung-yi Lee. One-shot voice conversion by vector quantization. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7734–7738, Barcelona, Spain, May 2020. IEEE. doi: 10.1109/ICASSP40776.2020.9053854.

Da-Yi Wu, Yen-Hao Chen, Hung-yi Lee, Ehab A. AlBadawy, and Siwei Lyu. VQVC+: One-shot voice conversion by vector quantization and U-net architecture. In *Proceedings of INTERSPEECH*, pages 4691–4695, Shanghai, China, October 2020. ISCA. doi: 10.21437/Interspeech.2020-1443.

Da-Yi Wu, Wen-Yi Hsiao, Fu-Rong Yang, Oscar Friedman, Warren Jackson, Scott Bruzenak, Yi-Wen Liu, and Yi-Hsuan Yang. DDSP-based singing vocoders: A new subtractive-based synthesizer and a comprehensive evaluation. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru (India), August 2022.

Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. Parallel Wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203, Barcelona, Spain, May 2020. IEEE. ISBN 978-1-5090-6631-5. doi: 10.1109/ICASSP40776.2020.9053795.

Yuya Yamamoto, Juhan Nam, Hiroko Terasawa, and Yuzuru Hiraga. Investigating time-frequency representations for audio feature extraction in singing technique classification. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 890–896. IEEE, December 2021.

Yuya* Yamamoto, Juhan Nam, and Hiroko Terasawa. Analysis and detection of singing techniques in repertoires of J-POP solo singers. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru (India), December 2022.

Chen Zhang, Jiaxing Yu, LuChin Chang, Xu Tan, Jiawei Chen, Tao Qin, and Kejun* Zhang. PDAugment: Data augmentation by pitch and duration adjustments for automatic lyrics transcription. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru (India), December 2022.

Yixuan Zhang, Yuzhou Liu, and DeLiang Wang. Complex ratio masking for singing voice separation. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 41–45, Toronto, ON, Canada, June 2021. IEEE. doi: 10.1109/ICASSP39728.2021.9414398.

Zhaoyan Zhang. Mechanics of human voice production and control. *The Journal of the Acoustical Society of America*, 140(4):2614–2635, 2016. ISSN 0001-4966. doi: 10.1121/1.4964509.

Zining Zhang, Bingsheng He, Zhenjie Zhang, Ehab A. AlBadawy, and Siwei Lyu. GAZEV: GAN-based zero-shot voice conversion over non-parallel speech corpus. In *Proceedings of INTERSPEECH*, pages 791–795, Shanghai, China, October 2020. ISCA. doi: 10.21437/Interspeech.2020-1710.

Kun Zhou, Berrak Sisman, Rui Liu, and Haizhou Li. Seen and unseen emotional style transfer for voice conversion with a new emotional speech dataset. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 920–924, Toronto, ON, Canada, June 2021. IEEE. doi: 10.1109/ICASSP39728.2021.9413391.

Tinghui Zhou, Philipp Krahenbuhl, Mathieu Aubry, Qixing Huang, and Alexei A. Efros. Learning dense correspondence via 3d-guided cycle consistency. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 117–126, Las Vegas, NV, USA, June 2016. IEEE. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.20.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2242–2251, Venice, Italy, October 2017. IEEE. ISBN 978-1-5386-1032-9. doi: 10.1109/ICCV.2017.244.