Applications of Discriminative, Generative and Predictive Deep Learning Processes to Solo Saxophone Practice

Mark Hanslip

PhD

University of York

School of Arts and Creative Technologies

June 2023

Abstract

Modelling of audio data through deep learning provides a means of creating novel sounds, processes, ideas and tools for musical creativity, yet its actual usefulness is relatively underexplored. Only a handful of researcher-practitioners are using AI models in their musical works, and artistic research into applications of deep learning modelling to instrumental practice and improvisation currently occupies an even smaller niche.

The research presented in this thesis and accompanying portfolio is an examination of potential creative applications of statistical modelling of audio data, through deep learning processes, to instrumental music practice; these processes are classification of a live input, generation of raw audio samples and sequential prediction of pitch. The goal of this work is, through the development of processes and creation of musical works, to generate knowledge concerning the practicality of modelling the systematic aspects of an instrumental improvised practice, the creative usefulness of such models to the practitioner, and the musical and technical 'behaviours' of specific classes of deep learning architecture with respect to the data on which the models are trained.

These concerns are addressed through a practice-based research methodology consisting of multiple steps: recording original audio datasets; pre-processing audio data as appropriate to model architecture and task; training statistical models; artistic experimentation and development of software, resulting in novel processes for musical creativity; and creation of artistic outputs, resulting in a portfolio of recordings and notated scores.

This project finds that deep learning can play useful roles in both technical and creative processes: classification can not only form the basis of interactive systems for improvisation but also be suggestive of new compositional structures; outputs of generative models of raw audio not only return valuable information about the training data but also generate useful source material for technical instrumental practice, improvisation and composition; notated outputs from symbolic-domain predictive models can also be richly suggestive of compositional ideas and structures for electroacoustic improvisation. This rich diversity of applications found posits AI as creative assistant, teacher and as deeply personalised tool for the instrumental practitioner.

When considering the utility of this work to others, there will be specific variances not covered by this project: appropriate choices of data representations, data-preprocessing techniques, model architectures and their training parameters will vary according to task, instrument, genre and taste, as will of course the character of others' creative outputs. However, the abundance of affordances and future directions this work uncovers gives confidence of its utility for other instrumental practitioners and researchers.

Given the pace of ongoing development of deep learning methods for modelling of audio and their still-limited adoption by creative practitioners, I hope that this thesis will motivate further explorations of the unique creative potential of these technologies by instrumental practitioners, improvisers and practice-based researchers in the wider field of AI for musical creativity.

This research was supported by a UK Research and Innovation (UKRI) grant via the Arts and Humanities Research Council (AHRC). The programme of funding support was delivered by the White Rose College of Arts and Humanities (WRoCAH).

It was carried out under the supervision of

Dr. Federico Reuben

Acknowledgements

Dr Federico Reuben for supervision, rewarding collaborations and friendship.

Caryn Douglas, Clare Meadley and all at White Rose College of Arts and Humanities (WRoCAH) for exceptional support across all stages of the process including funding, training and advice and guidance.

Dr Tom Collins for including me in his research lab during and following the pandemic, and for the very enjoyable AI Song Contest collaborations.

All staff at the School of Arts and Creative Technologies at University of York for various opportunities to present my research and musical outputs in presentations and public concerts.

Steve Mead, Toby Jones and all at Manchester Jazz Festival for the MJF Digital Originals commission in 2021.

Dr Adam Fairhall for encouraging me to do this in the first instance and for offering gentle encouragement, support and a listening ear along the way.

Roger, Christine, Hanny and John for patience, practical support and love.

Author's Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for a degree or other qualification at this University or elsewhere. All sources are acknowledged as references.

Contents

List of Figures

Glossary of Abbreviations and Terms

CPU	Central Processing Unit
GPU	Graphical Processing Unit
CQT	Constant-Q Transform
MFCC	Mel Frequency Cepstral Coefficient
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
WaveGAN	A variant of GAN for modelling raw audio signals
RNN	Recurrent Neural Network
SampleRNN	A variant of RNN for modelling raw audio signals.
In-head	Compositional statement preceding improvisation
Out-head	Compositional statement following improvisation
Extended technique	Non-standard music instrumental technique
Multiphonic	Simultaneous production of multiple tones through the use of extended techniques
Register	A pre-specified region of a musical instrument's pitch range
Altissimo	'False' upper register of saxophone accessed through non-standard fingerings

Chapter 1

Introduction

1.1 Overview

The work presented in this thesis concerns computational modelling of systematic musical processes for creative purposes. It presents multiple approaches to the creative application of statistical models of original audio data in the areas of instrumental practice, improvisation and composition for improvisation. Examples of such applications are presented in a portfolio of musical works, accompanied by a written commentary drawing out the research elements embedded in the portfolio and giving insights about the technical and creative processes.

Recent advancements in the computer science sub-field of deep learning have resulted in and continue to generate a plethora of new approaches to and architectures for creating statistical models of data. Those featured in this work, which were among the most popular model architectures at the outset of this project, include convolutional neural networks (CNNs), typically used for discriminative tasks such as image classification[1], generative adversarial networks (GANs), in which a discriminative model and a mirror-image generator model are trained in parallel most often to perform a range of image-domain tasks including synthesis, style transfer and super-resolution[2], and recurrent neural networks (RNNs), commonly used for predictive modelling of sequential data[3].

Through combination with music information retrieval technologies, adaptations of these innovations to the audio domain offer music practitioners the ability to model large bodies of recorded audio, opening up new creative possibilities. The field of AI-driven practice-based music research is still small and emerging but is growing, with contemporary classical composers and practitioners of electronic and pop music in particular beginning to use AI-based tools in their work. Research into applications of deep neural networks to improvisation and instrumental practice is, however, harder to find, with the majority of current practitioners tending to train their models on already-completed musical works. Example practitioners who exhibit this predominant tendency are composers Rob Laidlow and Emily Howard: the former based new compositional ideas in his work 'Silicon' on samples generated from SampleRNN models of BBC Philharmonic Radio Broadcasts, while the latter trained models of recordings of their own string quartets to generate new

[1] Rawat, Waseem and Zenghui Wang, 'Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review', *Neural Computation* 29, Issue:9, MIT Press, Sep, 2017, 2352-2449.

[2] Creswell, Antonia, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta and Anil A. Bharath, 'Generative Adversarial Networks: An Overview', *IEEE Signal Processing Magazine* 35, Issue:1, IEEE, Jan 2018, 53-65.

[3] Lipton, Zachary C., John Berkowitz and Charles Elkan, 'A Critical Review of Recurrent Neural Networks for Sequence Learning', unpublished paper, May 2015, arXiv:1506.00019.

compositional ideas[4] [5]; another is Dadabots, whose ongoing live streams of AI-generated death metal and free jazz are effectively concatenated outputs of SampleRNN models of recordings by death metal band Archspire[6] and John Coltrane and Rashid Ali's seminal free jazz album 'Interstellar Space'[7] respectively. It is with this somewhat surprising gap in mind that the work presented here aims to address the following research questions:

- What are the practical implications of using recordings of systematic instrumental practice on the saxophone as training data for deep learning models for applications in creative music practice?
- To what creative ends can these models be applied in the context of improvisation, instrumental practice and composition for improvisation? To what extent do they contribute to these applications and in what specific ways?
- What are the qualities and behaviours of specific model architectures in relation to the data on which they are trained?

1.2 Motivation

The initial impulse for this project was a curiosity about relationships between systematic processes and creative outputs within my practice. Before my acquaintance with deep learning, this simply meant the relationships between the systems of musical information I was practising privately and the content of my improvisations in common playing situations such as rehearsals, performances and recordings. As an obsessive practiser of musical systems I was bemused that what I worked on privately and what I played when improvising either unaccompanied or in group settings seemed only tenuously correlated. Additionally inspired by Douglas Hofstadter's notion of a system *acquiring a self*[8], I wanted to investigate this apparent disconnect further.

This initial idea is propogated throughout the work presented here. It is especially relevant to the core methodology of recording audio datasets of instrumental practice sessions, training deep learning models of them and observing what kind of *selfhood* they acquire. This idea of throwing ideas or concepts I find initially attractive or interesting into the 'black box' of deep learning (an over-used term to describe the - overstated, I now realise - inaccessibility of deep learning models to human inspection) and fine-tuning the result for creative ends strongly resonates with how I approached my improvisational practice at the outset of this research: practicing material I'm attracted to, throwing it into the black box and finding ways of working with what comes out.

1.3 Aims and Objectives

The aims of this work are:

- To explore and generate knowledge of the creative applications of various classes of deep learning architecture to instrumental practice, solo- and computer-augmented improvisation and composition for improvisation;

[4] Emily Howard, 'shield for String Quartet', musical score, 2022, https://www.editionpeters.com/product/shield/ep73579.

[5] Robert Laidlow, 'Robert Laidlow (2022): Silicon', YouTube video, posted by 'RNCM PRiSM', Mar 18 2023, https://youtu.be/3xmpywK0ACA.

[6] Carr, C.J. and Zack Zukowski, 'RELENTLESS DOPPELGANGER', YouTube video, posted by Dadabots, Sep 4, 2019, https://www.youtube.com/live/MwtVkPKx3RA?feature=share.

[7] Carr, C.J. and Zack Zukowski, 'OUTERHELIOS - Free Jazz - neural generated - Coltrane', YouTube video, posted by Dadabots, Jan 28, 2020, https://youtu.be/C0dOin79Hm0.

[8] Hofstadter, Douglas, *Goedel, Escher, Bach: An Eternal Golden Braid (Twentieth Anniversary Edition)*, Basic Books, 1999, 8

- To explore and generate knowledge of the ways in which machine learning models can augment and influence approaches to improvisation and composing for improvisation;
- To develop a portfolio of machine learning-augmented musical works for solo saxophone that reflect these explorations;
- To develop and signpost processes and musical works that other researchers and musicians will find useful for engagement with AI for musical exploration and creation.

The objectives through which these aims are to be achieved are as follows:

- To create a substantial original repository of audio data comprised of recordings of instrumental practice sessions;
- To develop understanding of and proficiency in the computational aspects of this work such that a meaningful, in-depth engagement with deep learning processes can be made;
- To develop machine learning pipelines for pre-processing and training statistical models of the datasets;
- To develop additional processes for working with the trained models' outputs, such as pitch-based onset detection, automated notation and real-time interactive loops;
- To make code and data for these processes available for others to use;
- To investigate creative applications of the trained models through a period of artistic experimentation;
- To create audio recordings of musical works that reflect these experiments, with accompanying notated scores where appropriate;
- To document and reflect on these processes and the insights generated through a written commentary.

The creative outputs are intended to be a significant extension of my existing practice. It is my hope that other researcher-practitioners, musicians and developers working in or seeking to work in the field of machine learning for musical creativity will find these outputs, along with the the datasets and programs generated through the research process, useful. These outputs represent my first recorded works of solo improvisation and my first recorded works made in collaboration with computers.

1.4 Methodology

The research methodology through which the work in this thesis and portfolio was carried out is described in this section. The steps described from 1.4.2 ('Data Pre-Processing') onwards recur in the structure of each of the core research chapters.

1.4.1 Dataset Creation

At this early stage of the project, it was necessary to create a repository of audio datasets on which various AI models could be trained in order to carry out this work. Doing so entailed a period of recording known material derived from various aspects of my instrumental practice, from purely systematic technical exercises to broadly defined approaches to improvisation. The process and outputs of this phase of the project are described in detail in chapter 3 ('Datasets').

1.4.2 Data Pre-Processing

Data pre-processing varies according to which model architecture the data transformation was intended for, but can be summarised as experimentation with data representations of audio and with techniques for data augmentation. Examples of audio representations used in this thesis are:

- Time-frequency spectrograms, in which changes over time in an audio signal's frequency components are represented in a 2-dimensional image. These are the primary data representation used in the work presented in Chapter 4;
- Raw audio waveforms, a high-resolution format comprising discreet amplitude measurements over time. This data representation is used throughout the work presented in Chapter 5;
- Symbolic representation of pitch, a reductive format in which only notes considered to be melodically relevant are represented numerically then converted to strings (computational representation of text). This data representation is used and discussed further in Chapter 6.

Data augmentation techniques are common machine learning practice for artifically inflating the size of a dataset while still adding 'unseen' data to the training set; this is appropriate when wishing to perform deep learning modelling of relatively small datasets[9]. Common techniques in other domains include rotation of image data[10] and the use of synonyms in text data[11]. Augmentations used in this thesis were typically done at an early stage on the raw data and include the following:

- Pitch shifting (raising and/or lowering the pitch of the data by a semitone/half-step);
- Time stretching (slightly speeding up and/or slowing down the data without changing the pitch);
- Inverting the waveform (flipping the polarity of the waveform's samples, creating a new data representation that is perceptually indistinguishable from the original; this technique is only suitable when modelling raw audio data).

These transformed copies of the original data are then recombined with it, enlarging the dataset considerably. Further details on data augmentations are discussed in chapters 4 and 5.

1.4.3 Training and Testing the Models

Model training was an experimental process in which training statistics such as *loss* calculations - measures of the distance between model performance and the ground-truth data - were monitored to ensure that the data was being succesfully modelled to some degree. Assuming the loss calculations were indicative of this, judgement of a training run's success was determined through its real-world usefulness. What this looked like in practice varied according to task and model architecture: for the classification tasks described in Chapter 3, 'model performance' constitutes whether the trained model can reliably classify my live input correctly; for raw audio generation and melodic prediction tasks in Chapters 5 and 6, subjective evaluation of the outputs' quality and/or degree of

[9] Taylor, Luke and Geoff Nitchke, 'Improving Deep Learning with Generic Data Augmentation', *IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, 2018.

[10] Shorten, Connor and Taghi M. Khoshgoftaar, 'A survey on Image Data Augmentation for Deep Learning', *Journal of Big Data* 6, 60, SpringerOpen, 2019.

[11] Liu, Pei, Xuemin Wang, Chao Xiang and Weiye Meng, 'A Survey of Text Data Augmentation', *2020 International Conference on Computer Communication and Network Security (CCNS)*, (IEEE, Aug 2020).

novelty and interest seemed the best approach. This approach is what I deemed most appropriate for the creative context: while I felt loss statistics were well worth tracking in all training experiments, 'real-world performance' hinged more on questions such as 'is it good *enough* to begin to make creative work with?' and 'do these outputs interest me?' rather than empirical measures of model performance.

1.4.4 Artistic Experimentation and Software Development

This phase of the research involved concurrent processes of idea-forming, developing code scripts to help realise these ideas, and periods of play through which the ideas would be realised. For ideas that were persevered with and not abandoned, typical research outputs from this phase were short recordings that served as indicators of creative potential and prototypes of later portfolio pieces. There was considerable trial-and-error at this stage of the research process: if the creative application of a trained model through development of these computer programs and artistic experiments did not seem to be as fruitful as hoped, then alternative applications would be explored.

As will be explored in Chapters 4, 5 and 6, development of additional scripts beyond those required to simply run inference on a trained model proved central to the artistic process at this stage. Since inferring on a trained statistical model by itself was not sufficient for useful creative work, it proved necessary to create additional structures around the models before an idea could be realised musically. This created an iterative development process in which an idea would be developed and tested cyclically, sometimes looping back to the earlier 'idea' phase if abandoned.

1.4.5 Creation of Artistic Outputs

In this phase of the research, processes and experiments developed in the previous phase were consolidated into more substantial creative outputs. Here, the process comes full circle back to a music-making setup, augmented by the technologies developed in previous stages. Processes of creative consolidation at this stage include what to me are long-familiar ways of working such as the creation of notated scores and using these scores as a basis for instrumental improvisation, as seen in the work in Chapters 5 and 6; more newly-developed processes are recording in real-time co-creation with interactive computer programs (such as in 'SoloSoloDuo' in Chapter 4 and 'b.io' in Chapter 5) and live digital audio effects processing (as heard in the outputs of Chapter 6).

1.5 Contributions

The main contributions of this thesis are:

- A set of original, publicly available audio datasets containing recordings of my instrumental practice;
- A set of publicly-available code bases for recreating the deep learning models and engaging with the processes developed throughtout this thesis;
- A range of strategies for applying AI to instrumental practice, improvisation and composition for improvisation, described in Chapters 4, 5 and 6 of this written commentary;
- A portfolio of original recordings and scores created using outputs of deep learning models trained on the aforementioned datasets and employing the aforementioned strategies;

- Broader reflections on the applications of AI for musical creativity arising from this work, presented through the written commentary.

In the following sections I provide detail on the contents of each of the key contributions of datasets, portfolio and written commentary.

1.5.1 Datasets

This section is an overview of the audio datasets I created in order to carry out this work; a more in-depth view is provided in Chapter 3.

- **'Exercises'** consists of two recorded datasets of me practicing exercises on tenor saxophone.
  - **'Tone Rows'**, contains 12-tone exercises derived from the 'Tone Rows' section of Slonimsky's Thesaurus of Scales and Melodic Patterns[12].
  - **'Scales and Arpeggios'** is recordings of me practicing various technical scale and arpeggio exercises.
- **'Registers'** consists of three sub-datasets of recorded exercises, improvised variations on these exercises and outright improvisation confined to specific registers of the tenor saxophone. These are named as **'Lower Register', 'Middle Register'** and **'Upper Register'**.
- The **'Improvisation'** datasets are recordings of me improvising long-form solos.
  - **'Melodic Improvisation'** contains improvisation with a conventional instrument technique and a melodic-rhythmic focus.
  - **'Timbral Improvisation'** dataset is based on extended techniques with a particular focus on the use of multiphonics.
- The **'G Major'** dataset contains improvisation and loosely defined exercises in the key of concert G major. Recorded in response to a collaborative project towards an entry to the 2021 AI Song Contest, it is the only dataset in this project with a fixed tempo and tonal center.

A selection of these datasets is publicly available to download and re-use:
https://huggingface.co/datasets/markhanslip/markhanslip_phd_saxophone_data.

1.5.2 Code

- The folder **'Ch4_Classification_Code'** contains the code used in Chapter 4 to create the portfolio piece 'SoloSoloDuo'.
  - **'CNN.py'** contains code for training and running inference on a convolutional neural network architecture for 64x64 pixel-sized images;
  - **'DataProcessing.py'** contains code for pre-processing audio data in spectrograms suitable for discriminative modelling with the convolutional neural networks in 'CNN.py';
  - **'PitchExtraction.py'** and **'Timbral Extraction.py'** contain code for logging machine listening data for improved interactions;
  - **'train.py'** contains the necessary code for training a convolutional neural network on the spectrogram data outputted by calling the 'DataProcessing.py' class;

[12] Slonismky, Nicolas, 'Twelve-Tone Patterns', *Thesaurus of Scales and Melodic Patterns*, Scribner, 1947, 173-175

  - **'interact.py'** allows the user to interact on-the-fly with the trained model.

The above processes formed the basis of version 2 of portfolio piece 'SoloSoloDuo', described below and in Chapter 4.
The code is publicly available: https://github.com/markhanslip/PhD_Ch4_CNN.

- The folder **'Ch5_SampleRNN_Code** contains a Colab notebook containing code blocks for the following processes:
  - Downloading datasets from the provided HuggingFace repository;
  - Pre-processing and augmenting the data to make it suitable for modelling with the SampleRNN architecture;
  - Configuring and training a SampleRNN model;
  - Selecting trained models from checkpoints in the training process and generating new audio files from them.

SampleRNN models trained via the above processes formed the basis of the portfolio pieces 'b.io', 'Workshop II', 'Lrrning' and 'The Lows' described below and in Chapter 5. The code used to augment the data, train a SampleRNN model and generate new outputs is publicly available, as is the code used in the interactive system for portfolio piece 'b.io': https://github.com/markhanslip/PhD_Ch5_SampleRNN.

- The folder **'Ch6_Char-RNN_Code** contains a Colab notebook with code for the following processes:
  - Downloading data from the provided HuggingFace repository;
  - Cloning the GitHub repo containing code for the following steps;
  - Pre-processing and (optionally) augmenting the data into text format to make it suitable for modelling with the modified Char-RNN architecture;
  - Defining the modified Char-RNN architecture;
  - Training a Char-RNN model;
  - Predicting streams of pitches from the trained model, wrapping them in Lilypond code and rendering them to notation in PDF format.

The notation generated via the above processes formed the basis of the portfolio pieces 'i prompt u', 'Taps' and 'Strange Loops' described below and in Chapter 6. The code for preprocessing the data to text, modelling it with the RNN and generating notation from it is publicly available: https://github.com/markhanslip/PhD_Ch6_Char_RNN.

1.5.3 Portfolio of Creative Outputs

- **'SoloSoloDuo'** is a structured improvisation for improviser and laptop in which two contrasting improvised solos are segmented into a sample library. These samples become the computer output component of an interactive duet in which sample choice is mediated by audio classification and frequency analysis. I recorded two versions which illustrate progression in the software development process and in my understanding of how to navigate the interactions. An audio-visual rendering of the first version of this piece was premiered online at AIMC 2022 in September 2022[13].

[13]Mark Hanslip, 'SoloSoloDuo', YouTube video, posted by 'AI Music Creativity 2022', Sep 20 2022, https://youtu.be/Nin8GIIZW-4.

- **'b.io'** is a short interactive duet with banks of pre-generated samples generated from SampleRNN[14] models of the 'Register' datasets. Choice of sample playback is informed by on-the-fly frequency analysis. An audio-visual version of this output was premiered under the name 'b.io' at ISMIR 2022 in Bangalore, India in December 2022[15].

- **'Duo with WaveGAN'** is a short interactive duet in which a WaveGAN[16] generator trained on the 'Melodic Improvisation' dataset is placed in an interactive loop and prompted for on-the-fly sample generation. Interaction and sample playback is mediated through use of frequency analysis of both live input and generated sample.

- **'Workshop I' and 'Workshop II'** are electroacoustic compositions that represent the use of samples generated from WaveGAN (piece I) and SampleRNN (piece II) models of the 'Exercises' datasets as material for technical practice. Recordings of the looped samples and of myself practising along with them are layered on top of sine tones, the frequencies of which are determined by analysis of the samples. Underpinning each piece is an environmental recording of a weaving workshop sampled from the publicly-available BBC Sound Effects Archive[17].

- **'Lrrning'** is a composition intended for use as a springboard for solo improvisation. The melodies, played in a set sequence and with repetitions in the manner of many Steve Lacy compositions, are notated from curated samples generated from a SampleRNN model of the 'Tone Rows' dataset.

- **'The Lows'** is a composition for solo improvisation focused on the lower register of the tenor saxophone. It's melodies are notated from curated samples generated from a SampleRNN model of the 'Lower Register' dataset.

- **'Fake Gander' and 'Gander'** are effectively the same composition, made of curated samples generated from a WaveGAN model trained on both 'Exercises' datasets. 'Fake Gander' includes an 'improvisation' generated from the WaveGAN model, whereas 'Gander' is an acoustic performance in which the composition serves as a framing for solo improvisation.

- **'Major Piece'** is a composition made up of curated samples generated from a WaveGAN model of the 'G Major' dataset. The performance features a looping environment written in SuperCollider. The improvised section takes place over a loop of the main melody; looping is also used in the out-head in which the three main phrases are dubbed on top of each other.

- **'Gandering 1'** is an audiovisual piece consisting of layered samples generated from WaveGAN models of externally-sourced audio datasets of improvisation. The only output in the portfolio to be based on other musicians' data, it warrants inclusion both for the additional insights generated to model architecture behaviour when training on recordings of improvisation and for the additional example of WaveGAN's affordance as a tool for sample-based music creation. The piece was part of

[14]Mehri, Soroush, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville and Yoshua Bengio. 'SampleRNN: An Unconditional End-to-End Neural Audio Generation Model', (conference paper, ICLR 2017: Fifth International Conference on Learning Representations, Palais des Congrès Neptune, Toulon, France, April 24-26, 2017).

[15]Mark Hanslip, 'b.io', webpage, Dec 6 2022, https://ismir2022program.ismir.net/music_346.html

[16]Donahue, Chris, Julian McAuley and Miller Puckette. 'Adversarial Audio Synthesis' (conference paper), ICLR 2019: Seventh International Conference on Learning Representations, Ernest N. Morial Convention Center, New Orleans, May 6-9, 2019.

[17]https://sound-effects.bbcrewind.co.uk/

a Manchester Jazz Festival 2021 Digital Originals commission and premiered online in early 2022 on the festival's digital media channels[18].

- **'i prompt u'** is a composition created from curated outputs of a predictive RNN trained on a composite text dataset created through frequency and onset analysis of all datasets (except 'Timbral Improvisation'). The model is trained using an implementation of a character-level RNN[19] that I adapted for numeric strings and further adapted for automated notation of its outputs. Melodic phrases are ordered so as to progress from clearly implied tonal centers to atonality. Granular delay and pitchshift effects are applied, with the type and degree of effect varied to reflect the tonal progression of the phrases.

- **'Taps'**, another composition created from Char-RNN outputs, is an atonal piece in which staccato melodies are interspersed with short, improvised sounds; these sounds are explicitly derived from the final, repeated notes of each composed phrase. A multi-tap delay effect is used throughout, with transitions from composed melody to improvisation additionally delineated by changes in the effect parameters.

- **'Strange Loops'** is another composition for improvisation created from curated Char-RNN outputs. The outputs were chosen for their shared implied tonal center of E harmonic major. The piece employs a pedal-based environment for looping and granulation written in SuperCollider.

1.5.4 Written Commentary

The written contributions by chapter are as follows:

- Chapter 3 (Datasets) is a discussion of the dataset contents and their relationship to my practice.

- Chapter 4 (Audio Classification in Practice) discusses the process of training an image-domain CNN adapted to audio to computationally model my perception of my instrumental approaches and embedding of the trained model in a real-time interactive loop. The resulting program's creative application in the form of two iterations of 'SoloSoloDuo', a structured improvisation for saxophone and computer is presented.

- Chapter 5 (Generative Modelling of Raw Audio in Practice) describes the use of WaveGAN and SampleRNN architectures for generative modelling of my datasets in raw audio form. This prompts discussion of the generated samples' characteristics in relation to their respective architectures and the ground truth data, leading to assumptions of model architecture 'behaviours'. Their creative application to real-time interactive duos (in the form of 'Duo I' and 'Duo II') is presented and behaviours of each model architectures in this context and influence of the samples on my playing are discussed. My experience of using generated samples as material for technical practice is discussed and presented creatively in the form of 'Workshop', an electroacoustic composition. Applications of SampleRNN and WaveGAN outputs to compositions for solo improvisation in the form of 'Lrnning', 'The Lows', 'Gander' and 'Major Piece' are described and discussed.

- Chapter 6 (Symbolic-Domain Melodic Prediction in Practice) presents an exploration of the adaptation of an early deep learning model architecture for text prediction as a

[18] Mark Hanslip, 'Gandering 1', YouTube video, posted by 'Manchester Jazz Festival', Jan 18 2022, https://youtu.be/5dIxUWNGndc.

[19] Andrej Karpathy, 'The Unreasonable Effectiveness of Recurrent Neural Networks', blog post, *Andrej Karpathy Blog*, May 21, 2015, http://karpathy.github.io/2015/05/21/rnn-effectiveness/

means to generated notated pitches, which become the basis for three semi-structured electroacoustic improvisations. A novel, fully automated process is presented in which rhythmically salient pitches are extracted from a composite dataset made up of several of those presented in Chapter 3. A character-level recurrent neural network is trained, output predictions are generated by prompting the trained model, and these outputs are piped to notation software. Creative applications of this process in which the raw notated outputs are hand-curated and used as the basis for three works for solo improviser, 'i prompt u', 'Strange Loops' and 'Taps', are presented and discussed.

- Chapter 7 (Conclusions) summarises the work presented and its key themes and outline some steps for further development.

1.6 Conclusion

In this chapter I outlined the rationale for this research, presented the core research questions this work seeks to answer and described the motivations that led to seeking to carry out this research in the first instance. I enumerated the aims of this research and the objectives to be fulfilled in order to meet the proposed aims, and described the methodology through which the research was carried out. I then outlined the core contributions to existing knowledge this thesis and portfolio aims to make. The following chapter provides a more comprehensive view of the wider context in which this project takes place.

Chapter 2

Background

2.1 Introduction

The following sections describe the musical context and academic research that this project builds upon. First, I discuss the history of solo improvisation on the tenor saxophone to provide context for the material presented in the datasets. I discuss specific practitioners of solo improvisation on tenor saxophone and of solo electro-acoustic saxophone practice, and their relevance to the creative outputs in the portfolio. This leads to a discussion of innovations in the computer science field of deep learning and of the specific model architectures used to train the statistical models as part of this research. I then conclude this chapter with a view of practitioner-researchers currently working in the emerging field of AI for musical creativity in order to better locate the portfolio outputs in their research context.

2.2 Musical Context

In this section I will first describe the historicalcontext and history of solo improvisation on the tenor saxophone to provide context for the datasets of instrumental practice presented in Chapter 3. I discuss several key practitioners of solo improvisation on tenor saxophone and the influence they bring to bear on both the datasets and some of the creative portfolio outputs. I then offer a look at saxophonists with a solo electro-acoustic practice to provide further context for electro-acoustic pieces in the portfolio outputs, before concluding this section with a theory of ideas generation in solo improvisation; this theory provides additional context for material contained in the more improvisation-focused datasets and several of the portfolio outputs.

2.2.1 Solo Improvisation and the Tenor Saxophone

Of the multiple recordings of solo saxophone improvisation throughout the histories of jazz, free jazz and improvised musics, the majority of those considered seminal were recorded on members of the saxophone family besides the tenor. These include influential recordings such as Evan Parker's 'Monoceros'[1] (on soprano saxophone), Anthony Braxton's 'For Alto'[2] (on alto saxophone), John Zorn's 'The Classic Guide to Strategy'[3] (on mostly dismantled clarinet and alto saxophone parts and objects), Hamiet Bluiett, Jr.'s 'Birthright: A Solo Blues Concert'[4] (on baritone saxophone) and a number of Steve Lacy's solo soprano saxophone albums. Joe McPhee's 'Tenor', generally considered to be a historically important recording of solo saxophone improvisation, is a rare example of such

[1] Evan Parker, *Monoceros*, Incus Records, 1978, LP (since reissued).
[2] Anthony Braxton, *For Alto*, Delmark Records, 1971, LP.
[3] John Zorn, *The Classic Guide to Strategy: VOLUMES ONE AND TWO*, Tzadik Records, 1996, CD.
[4] Hamiet Bluiett, Jr., *Birthright: A Solo Blues Concert*, India Navigation, 1977, LP.

an album recorded on tenor saxophone.

This paucity of the instrument's representation in solo improvisation is surprising considering its long-term ubiquity in jazz: the main instrument of choice for Lester Young, Chu Berry, Paul Gonsalves, John Coltrane, Sonny Rollins, John Gilmore, Warne Marsh, Joe Henderson and so many others, it continues to be a prominent instrument in jazz and improvised music today. It is also interesting to note that what is generally acknowledged to be the first recording of unaccompanied improvisation on the saxophone was Coleman Hawkins' 'Picasso'[5], on tenor saxophone.

The history of improvised music is still being written though, and many recordings of unaccompanied improvisation on tenor saxophone have been made in more recent years. These include David Liebman's 'Colors'[6], an explicitly emotional, expressionist outing, and Bill McHenry's 'Solo'[7], a more contained affair focused on strict motivic development of compositional themes. Some of those that bear specifically on my practice and the work in this project include the first half of John Butcher's 'Bell Trove Spools'[8] (he switches to soprano for the second half), Evan Parker's 'Chicago Solo'[9] and Ellery Eskelin's 'Solo Live at Snug's'[10], as well as their practices in general.

John Butcher's 'Bell Trove Spools' exemplifies the use of multiphonics in solo saxophone improvisation. Multiphonics are generated by non-standard combinations of keys and concurrent manipulations of the breath and embouchure. This mode of sound production is one of the characteristic features of the 'Timbral Improvisation' dataset discussed in Chapter 3. A deeply embedded part of my practice, having been worked on and used in my performances and recordings since the mid-2000s, it also appears in many of the creative outputs in this work, particularly 'SoloSoloDuo' in Chapter 4, 'Gander' and 'The Lows' in Chapter 5 and 'Taps' and 'Strange Loops' in Chapter 6.

Evan Parker's 'Chicago Solo', which unusually for him consists solely of solo improvisations on tenor saxophone, contains features that I have adapted to my own practice. This includes the use of multiphonics as described above but also the use of repeated circular phrases incorporating overtones - tones of the harmonic series produced by effectively overblowing a fundamental pitch. This technique can be heard in the 'Timbral Improvisation' dataset. Evan's personal practice of sections of the Slonimsky Thesaurus of Scales and Melodic Patterns[11] has also influenced this work: the contents of the 'Tone Rows' dataset are adapted from the 'Tone Rows' section of the Slonimsky book[12].

Ellery Eskelin's 'Solo Live at Snugs' showcases a melody-centred approach to solo improvisation that informs my approach to creating the 'Melodic Improvisation' dataset presented in Chapter 3. Eskelin's near-total eschewal of extended techniques such as multiphonics and over-blowing on this record (which he was previously well-known as an exponent of) provides a strong example of how one can restrict oneself to specific aspects of ones vocabulary without sacrificing musical interest. His successful merging of jazz-based language with more abstract intervallic constructions have influenced my own efforts to do the same.

[5]Coleman Hawkins, 'Picasso', *The Verve Story - Disc One: 1944-53*, Verve Records, 1994, CD.

[6]David Liebman, *Colors*, Hatology, 2003, CD.

[7]Bill McHenry, *Solo*, Underpool, 2018, CD.

[8]John Butcher, *Bell Trove Spools*, Northern Spy, 2012, CD and Digital, https://johnbutcher.bandcamp.com/album/bell-trove-spools-2

[9]Evan Parker, *Chicago Solo*, Okkadisk, 1997, CD.

[10]Ellery Eskelin, *Solo Live at Snug's*, hatOLOGY, 2015, CD.

[11]Evan Parker, interviewed by Frances-Marie Uitti, *Contemporary Music Review* 25, Issues 5-6, Oct-Dec 2006, 411-416.

[12]Nicolas Slonimksy, *Thesaurus of Scales and Melodic Patterns.*

2.2.2 Composition for Solo Saxophone Improvisation

These last three albums would typically be considered examples of 'free' improvisation, being absent of any explicit compositional statements. By contrast, a defining feature of Steve Lacy's solo albums such as 'Clinkers'[13] and 'November'[14] is the use of compositions consisting of repeated motifs as springboards for solo improvisation. Bill McHenry's aforementioned album 'Solo' also features the use of repetitious compositional statements, but unlike Lacy, who departs significantly from the compositional statement as soon as it ends, McHenry's improvisations often consist of subtle continuations of and variations on the pre-composed material. Lacy and McHenry's approaches to composition *for* improvisation inform my solo works 'Lrrning', 'The Lows' and 'Gander' in Chapter 5. Lacy's use of intervallic construction of melody in both composition and solo improvisation is also a strong influence on these works and on the melodic aspect of my improvising, an influence that can be heard in the above-mentioned works and also in 'Taps' and 'i prompt u' in Chapter 6.

2.2.3 Electro-acoustic Saxophone Improvisation

While these last examples pay attention to acoustic solo playing, part of the musical context of this project is electro-acoustic saxophone improvisation. When discussing examples of electro-acoustic saxophone improvisation in practice, I find it useful to make a distinction between saxophonists who take full responsibility for the 'electro' aspect of their electro-acoustic music, and those who are collaborating with computer musicians. This distinction matters because it seems clear that a saxophonist would make different choices of electronic augmentation of *their own* sound than would a computer musician asserting their own musical choices. Additionally, the dynamic created by playing solo is quite distinct from the dynamic of a duo, in which two musicians with their own aesthetics engage in a dialogue. Since the outputs of this research occupy the former dynamic, I will start by paying attention to practitioners specifically engaged in it, before discussing some interesting examples of the latter formation.

Examples of saxophonists combining their practice with electronics in a self-directed manner are rare: for the quality of both instrumental improvisation and use of electronics the work of Jorrit Dijkstra stands out, his solo electro-acoustic albums '30 Micro-Stems'[15] and 'Never Odd or Even'[16] being exemplars of solo saxophone improvisation augmented with electronics. His use of looping as a primary tool on several tracks of '30 Micro-Stems' informs 'Major Piece' in Chapter 5 and 'Strange Loops' in Chapter 6; his use of pitch shifting and delay on 'Toodeledoo' on 'Never Odd or Even' informed the effects used in 'i prompt u' and his use of multi-tap delay effects in 'Mind The Gaps' on '30 Micro-Stems' informed my use of the same in 'Taps', both in Chapter 6 also. Dijsktra's music sounds as though it could have been made in a modern digital environment such as Max/MSP or SuperCollider but all of the electronic augmentation in his music is made via an analog setup. Another example of an accomplished saxophonist expanding their practice outwards by adding electronic augmentation is Sam Gendel, whose recorded output is almost exclusively electro-acoustic. His extensive use of drone-like loops on tracks 'East LA Haze Dream' and 'ZeroZero' on his album 'Pass If Music' informs my use of the same on 'Strange Loops' in Chapter 6.[17]

[13] Steve Lacy, *Clinkers*, HatHut, 1978, CD

[14] Steve Lacy, *November*, Intakt Records, 2010, CD and Digital https://steve-lacy.bandcamp.com/album/november

[15] Jorrit Dijkstra, *30 Micro-Stems*, TryTone Records, 2002, CD and Digital, https://jorritdijkstra.bandcamp.com/album/30-micro-stems.

[16] Jorrit Dijkstra, *Never Odd or Even*, Driff Records, 2015, CD and Digital, https://jorritdijkstra.bandcamp.com/album/never-odd-or-even.

[17] Sam Gendel, *Pass If Music*, Leaving Records, 2018, Cassette and Digital,

A less well-known example is Floros Floridis's 'F.L.O.R.O. IV - Future Learning of Radical Options'[18]. A more aggressively experimental outing than either of Dijsktra's solo albums, an especially interesting aspect of this work is Floridis' use of samples of pre-existing albums on the tracks 'Post-traditional mood' and 'Stones,breaths,drops...name it', playbacks of which are triggered by playing long notes. This adds a poly-stylistic aspect to the music which is further enhanced by the clear influence of traditional Greek music on Floridis' playing, bringing further stylistic reference and cultural flavour to this often resolutely abstract domain. While Floridis' music here is much more polystylistic and inclusive of non-saxophone-generated sounds than the works in this project's portfolio, his roughly equal prioritisation of instrumental improvisation and electronic augmentation resonates with the interactive duets 'b.io' and 'Duo with WaveGAN' in Chapter 5 and 'Strange Loops', 'Taps' and 'i prompt u' in Chapter 6.

Moving onto saxophonists working with computer musicians, Guillaume Orti's collaboration with Olivier Sens, 'Reverse' [19] illustrates well the distinction between an instrumentalist choosing their own electronic augmentations vs the choices a computer musician might make. Here, while Sens does process Orti's sound directly, on tracks such as 'Miss Ann - fragmentation', 'Freedom jazz dance - équivalence' (covers of well-known jazz compositions by Eric Dolphy[20] and Eddie Harris[21], respectively) and 'Drift - dream drum', he also functions almost as a drummmer in his use of drum-like (possibly drum-derived) timbres and rhythmic sequences; his software is designed such that he is able to be significantly interactive from this aesthetic position.

Cecilia Lopez and Ingrid Laubrock's 'Maromas'[22] is a more recent electro-acoustic collaboration between an improvising saxophonist and a technologist that highlights, in a different way to 'Reverse', the additional material a second participant brings to the 'electro' element, as contrasted with a solo electro-acoustic effort. While there is plenty of real-time effects processing that could conceivably have been controlled by Laubrock had this been a solo album, there is significant additional intervention by Lopez with material that sounds and feels very independent of Laubrock's input, regardless of how it might have been generated. Lopez's inputs and processing tend towards the distorted and have a more modular synth-type feel than Sens' more delicate interventions, resulting in a harsher soundworld and more unrestrained improvising.

2.2.4 Ideas Generation in Solo Improvisation

Moving away from musical practitioners and towards academic research, one domain-specific framework for ideas generation in solo improvisation stands out in the literature and provides an invaluable framework for consideration of many the above-described works and some of those created during this project. Jeff Pressing's model of ideas generation in solo improvisation described in the article 'Improvisation: Methods and Models'[23] contextualises structural techniques he had observed in solo improvisation by arguing that generation of ideas in this context can be essentially boiled down to one of two approaches:

https://leavingrecords.bandcamp.com/album/pass-if-music.

[18] Floros Floridis, *F.L.O.R.O. IV - Future Learning of Radical Options*, To Pikap Records, 2019, Vinyl and Digital, https://topikaprecords.bandcamp.com/album/f-l-o-r-o-iv-future-learning-of-radical-options.

[19] Guillaume Orti & Olivier Sens, *Reverse*, Quoi de neuf docteur, 2009, CD and Digital.

[20] Eric Dolphy, 'Miss Ann', *Far Cry*, New Jazz Records, 1962.

[21] Eddie Harris, 'Freedom Jazz Dance', *The In Sound*, Atlantic Records, 1965.

[22] Cecilia Lopez & Ingrid Laubrock, *Maromas*, Relative Pitch Records, 2023, CD and Digital, https://relativepitchrecords.bandcamp.com/album/maromas.

[23] Jeff Pressing, 'Improvisation: Methods and Models' in *Generative Processes in Music: The Psychology of Performance, Improvisation, and Composition* ed. John Sloboda (Oxford University Press, 2001), 129-156.

associate-type generation, in which the content of the previous idea markedly informs the next; and *interrupt-type*, in which the content of the previous idea is discarded in favor of new, contrasting material. While this concept may seem reductive, it can be deployed subtly: interruptive generation can add variety within a larger section of similar (or more 'associative') material; associations in the mind of the improviser can also be made between aesthetically divergent musical ideas. Additionally, Pressing proposes that these modes can occur at various structural levels: mid-phrase, between phrases and between sections.

This concept provided valuable context for engagement with solo improvisation at the outset of my research and its influence can be heard throughout the datasets and portfolio of outputs. These ideas prompted reflections on how other improvisers approach ideas generation in their own work. For example, in 'Solo Live at Snug's', Eskelin's use of what Jeff Pressing would term 'associate-type' generation of ideas, which takes the form of repeating phrase gestures with modifications to content, and beginning new phrases with ideas from the end of the previous phrase, also provides context for my own generation of ideas in the melodic domain. In addition to the 'Melodic Improvisation' dataset in Chapter 3, these concepts appear in 'SoloSoloDuo' in Chapter 4, 'Lrrning', 'The Lows', 'Gander' in Chapter 5 and 'i prompt u' in Chapter 6. In John Butcher's solo work such as that heard on 'Bell Trove Spools', his tendency to sustain and deploy subtle variations of a specific cluster of similar multiphonics on the tenor saxophone can be contextualised as 'associate-type' generation whereas his abandonment of the idea in favour of a highly contrasting timbre can be considered more 'interruptive', an approach I adopted when recording the 'Timbral Improvisation' dataset. In Orti/Sens' 'Trio - Event Process'[24], Orti's saxophone inputs are continually *interrupted*, thwarted even, by the computer's interventions; this interruptive human-computer dynamic is also heard in my interactive pieces 'SoloSoloDuo', 'b.io' and 'Duo with WaveGAN'.

2.3 Statistical Modelling of Audio with Deep Learning

Having established the musical background of this research, I will now examine the computer science innovations in the fields of deep learning for classification, generation and prediction upon which the AI aspects of this project draw.

Deep learning can be defined as a subset of machine learning characterised by densely layered neural network architectures whose large *weight spaces* are capable of modelling complex relationships within correspondingly large datasets. These advancements are made possible by a confluence of statistical innovations such as backpropogation[25] (a reversal of the 'chain rule' in mathematics in which a reverse calculation is performed back along the path traced by the 'forward pass', the outputs of which are then saved to the trained weight space; this calculation is essentially the where the 'learning' in deep learning happens) and new loss functions[26] (measures of the distance between the model's knowledge of the dataset and the 'ground truth' data); the adaptation of existing computer graphics hardware (GPUs) to the dense matrix operations deep learning tasks necessitate has also been a significant factor in recent advancements in the field. These innovations have led to a surge in new model architectures for diverse applications, one that is ongoing: during the time spent on this project, significant new architectures for a range of audio tasks were released, some of which are described in the section 'Ongoing Innovations in Machine Learning for Audio' below.

[24]Guillaume Orti & Olivier Sens, 'Trio - Event Process', *Reversed*, Quoi de neuf docteur, 2009.

[25]Rojas, Raúl, 'The Backpropagation Algorithm' in *Neural Networks*, Springer, 1996, 149–182.

[26]Wang, Qi, Yue Ma, Kun Zhao and Yingjie Tian, 'A Comprehensive Survey of Loss Functions in Machine Learning' in *Annals of Deep Learning* 9, Springer, 2022, 187–212.

2.3.1 Discriminative Modelling of Audio Spectrogram Data with CNNs

Convolution Neural Network architectures are an innovation in the field of machine learning that, through the use of *convolution* operations in place of standard dense layers, have expanded the possibilities of image modelling in particular [27]. A benchmark example often used to illustrate their power is the discriminative modelling of large image datasets such as ImageNet, which consists of over 1 million images belonging to 1000 object classes; deep convolutional model architectures such as ResNets[28] have made accurate discriminative modelling of such datasets possible.

At the outset of this research it was common practice to co-opt these model architectures intended for image classification for the task of audio classification by using pre-categorised spectrogram representations of audio signals - most commonly time-frequency representations such as mel and STFT spectrograms - in place of a regular image dataset. Digging into this trend, in 'Comparison of Time-Frequency Representations for Environmental Sound Classification using Convolutional Neural Networks'[29], Muhammad Huzaifah presented an empirical study of the performance of various time-frequency representations for audio classification. The paper shows mel spectrograms to outperform STFT spectrograms and wavelet transforms while motivating further investigation of the constant-Q transform (CQT) for musically-focused classification tasks. As a result of this work my own choice of audio representation for classification in the work presented in Chapter 4 was made more straightforward, beginning with mel spectrograms and changing to CQTs upon noticing a performance improvement.

Audio classification with convolutional architectures and spectrograms has been shown to work quite well, such as in Bian, Wang et al's 'Audio-Based Music Classification with DenseNet And Data Augmentation' and in the technique's appearance in popular technologies such as Cornell University's BirdNET app [30]. However, doubts are being cast on the fundamental suitability of convolutional architectures to classification of audio spectrograms. While it is not questioned that CNNs are highly effective at complex discriminative modelling tasks and that spectrograms are the least-lossy audio representation after the raw signal and an obvious choice for such tasks in the audio domain, there is a suspected mismatch between architecture, which is optimised for *object* classification within natural images, and time-frequency audio representations. It is additionally interesting to note that audio classification functionality in the Flucoma toolkit[31] is restricted to multi-layer perceptron or k-nearest neighbours models: simpler, less intensive methods that the authors deemed sufficient and appropriate for fast audio classification. Clearly, optimal methods for audio classification with machine learning are both context-dependent and an open question.

[27] Rawat, Waseem and Zenghui Wang, 'Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review', *Neural Computation* 29, 2352-2449.

[28] He, Kaiming, Xiangyu Zhang, Shaoqing Ren and Jian Sun, 'Deep Residual Learning for Image Recognition', unpublished paper, arXiv:1512.03385.

[29] Huzaifah, Muhammad, 'Comparison of Time-Frequency Representations for Environmental Sound Classification using Convolutional Neural Networks', (unpublished paper, arXiv:1706.07156).

[30] Kahl, Stefan, Connor M. Wood, Maximilian Eibl and Holger Klinck,'BirdNET: A deep learning solution for avian diversity monitoring' in *Ecological Informatics* 61, Mar 2021, ScienceDirect, 101236.

[31] Tremblay, Pierre Alexandre, Owen Green, Gerard Roma, Alexander Harker, 'From collections to corpora: Exploring sounds through fluid decomposition', paper presented at ICMC 2019: 45th International Computer Music Conference, Elmer Holmes Bobst Library, MORE, New York University, USA, Jun 16-23, 2019.

2.3.2 Generative Modelling of Audio Data with GANs

Generative Adversarial Networks (GANs) are a startling innovation first proposed by Goodfellow et al in 2014[32] and optimised for image generation tasks through the use of convolutional layers by Radford et al in their DCGAN (Deep Convolutional Generative Adversarial Network) model architecture[33]. DCGAN consists of a discriminative convolutional architecture whose role is to learn the ground truth data and distinguish it from outputs generated by a second 'generator' model whose architecture mirrors that of the discriminator model. While the generator initially outputs gaussian noise vectors, its weights are continually updated with the discriminator's outputs in a process of *adversarial* training, the goal of which is for the generator to create samples the discriminator cannot differentiate from the ground truth data. Once something close to this goal has been reached, the trained generator weights can be used to generate plausible recombinations of the dataset.[34]

In 'Adversarial Audio Synthesis'[35], Donahue et al present WaveGAN, an adaptation of this DCGAN architecture for modelling raw audio signals in which fixed-length raw audio signals are modelled through the use of 1-dimensional convolutional layers. This model architecture is used throughout much of the work in Chapter 5 as a framework for creating statistical models of several of the datasets presented in Chapter 3. Novel audio samples are generated from these models and are applied creatively in a variety of ways. In the paper the authors also discuss their preceding work on SpecGAN, an adaptation of DCGAN for modelling audio data from time-frequency spectrograms, noting that WaveGAN's outputs are considered to be of higher quality and highilghting the difficulty of restoring the generated spectrograms back to raw audio. Marafioti et al address this problem to an extent in their TiFGAN architecture[36], with which 1 second-long invertible log-scaled STFT spectrograms are generatively modelled, by using state-of-the-art algorithms for phase recovery and STFT reconstruction and arguably outperforming WaveGAN as a result. I unfortunately encountered difficulty getting their implementation to run and chose to focus my efforts on modelling my datasets with WaveGAN for straightforward practical reasons.

2.3.3 Generative and Predictive Modelling of Raw and Symbolic Audio Data with RNNs

Recurrent Neural Networks (RNNs) are intended for modelling of sequential data typically used for prediction of text and numerical data in which tokenized representations of fixed-length randomised chunks of the data set are learned through the use of either LSTM (long-short-term memory) or GRU (gated recurrent unit) cells[37]. In the fast-growing field of language modelling they have long been superseded by the advent of transformer models

[32]Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville and Yoshua Bengio, 'Generative Adversarial Nets' (conference paper, NIPS 2014: 28th Conference on Neural Information Processing Systems, Palais des Congrès de Montréal, Canada, Dec 8-13, 2014).

[33]Radford, Alex, Luke Metz and Soumith Chintala, 'Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks', (poster presentation, ICLR 2016: 4th International Conference of Learning Representations, Caribe Hilton, San Juan, Puerto Rico, May 2-4, 2016).

[34]Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville and Yoshua Bengio, 'Generative adversarial networks' in *Communications of the ACM* 63, No. 11, ACM Digital Library, Nov 2020, 139-144.

[35]Donahue, McAuley and Puckette. 'Adversarial Audio Synthesis'.

[36]Marafioti, Andrés, Nicki Holighaus, Nathanaël Perraudin and Piotr Majdak,'Adversarial Generation of Time-Frequency Features with Application in Audio Synthesis', paper presented at ICML 2019: 36th International Conference on Machine Learning, Long Beach Convention Centre, Long Beach, California, USA, Jun 9-15, 2019.

[37]Cahuantzi, Roberto, Xinye Chen and Stefan Güttel, 'A comparison of LSTM and GRU networks for learning symbolic sequences', unpublished paper, Sep 2019, arXiv:2107.02248.

[38] but for audio practitioners they still hold significant creative potential. Below I describe an early innovation in RNNs and its adaptation to musical data, followed by a unique adaptation of RNNs to modelling of raw audio signals.

In 'The Unreasonable Effectiveness of Recurrent Neural Networks'[39], Andrej Karpathy presents an architecture for learning text data at the character level and, once trained, predicting continuation of text strings from a given prompt. This model architecture is adapted for conditional generation of melodic material in the form of string-recasted MIDI-note representations of pitch in Chapter 6. A precedent for adapting Char-RNN for music generation has been set by Sturm et al in their work on 'Folk-RNN'[40], which has proven itself capable of generating outputs closely akin to Irish folk music in style Google Magenta's Performance RNN[41] uses an LSTM-based RNN to model MIDI data of piano performances that includes expressive timing and dynamics, and notably is capable of generating polyphonic outputs. While Performance RNN focuses on tonal classical piano music, in 'Towards a Deep Improviser: A Prototype Deep Learning Post-tonal Free Music'[42], Dean et al model symbolic representations of post-tonal and post-metric piano music with a view to creating a virtual free improvising pianist. Other examples of adapting RNNs for symbolic-domain music generation include Eck and Schmidhuber's modelling of improvisation on a 12-bar blues structure in an attempt to move beyond note-event-level prediction and towards modelling larger-scale sequences with recognisable structural patterns.[43]

It should be noted here that the above examples of symbolic audio data modelling with RNNs are engineered for the automated prediction of listenable musical sequences, the work I present in Chapter 6 takes a more stripped-back approach, generating only sequences of pitches. While this might appear to be a backwards step compared with the above approaches, it emerges that these minimally-specified sequences are sufficient to spark creative action. I also take the approach of engineering an end-to-end pipeline for data pre-processing, modelling, prediction and notation which represents a further divergence from the above approaches.

Mehri et al present in 'SampleRNN: An Unconditional End-To-End Neural Audio Generation Model'[44] a multi-layer recurrent neural network architecture for unconditional generation of raw audio signals. This model architecture is also used for modelling the raw audio signals of my datasets, generated outputs of which form the basis of various creative outputs. I favour the 'prism-samplernn' implementation developed at RNCM's PRiSM lab by Sam Salem and Christopher Melen[45]. A prominent example of SampleRNN's creative application is described in 'Generating Albums with SampleRNN to Imitate Metal, Rock

[38] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin, 'Attention is All You Need', unpublished paper, arXiv:1706.03762.

[39] Andrej Karpathy, 'The Unreasonable Effective of Recurrent Neural Networks'.

[40] Sturm, Bob, Joao Felipe Santos and Iryna Korshunova, 'Folk Music Style Modelling by Recurrent Neural Networks with Long Short Term Memory Units' (conference paper), ISMIR 2015: 16th International Society for Music Information Retrieval Conference, Hotel NH Malaga, Malaga, Spain, Oct 26-30, 2015.

[41] Simon, Ian and Sageev Oore, 'Performance RNN: Generating Music with Expressive Timing and Dynamics', Magenta Blog, 2017, https://magenta.tensorflow.org/performance-rnn.

[42] Dean, Roger T. and Jamie Forth, 'Towards a Deep Improviser: a prototype deep learning post-tonal free music generator' in *Neural Computing and Applications* 32, 2020, 969–979.

[43] Eck, Douglas and Jurgen Schmidhuber, 'A First Look at Music Composition using LSTM Recurrent Neural Networks', technical report, Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale, Mar 2002.

[44] Mehri, Soroush, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville and Yoshua Bengio. 'SampleRNN: An Unconditional End-to-End Neural Audio Generation Model'.

[45] Christopher Melen and Sam Salem, 'PRiSM SampleRNN', code repository, https://github.com/rncm-prism/prism-samplernn

and Punk Bands' by CJ Carr and Zach Zuckowski aka Dadabots[46]; while my creative approaches diverge significantly from theirs, Dadabots' work provided initial inspiration for my first experiments with this model architecture and their informal but valuable and generous advice for dataset creation and parameter tuning with SampleRNN during this project (via their Discord forum and during informal online meetups) is gratefully noted.

2.3.4 Behaviours of Deep Generative Models

With respect to the behavioural characteristics of both Recurrent Neural Networks and Generative Adversarial Networks described above, Google DeepMind researcher Sander Dieleman's blog post 'Generating music in the waveform domain'[47] offers invaluable descriptors and observations: *mode-covering* behaviour, in which a generative model tries to account for the entire dataset in its trained distribution, and *mode-seeking* behaviour, in which the trained distribution tends to converge around certain features of the dataset at the expense of others. These behaviours are representative of those of RNNs (such as Char-RNN and SampleRNN, described above) and GANs (such as WaveGAN, described above) respectively and are helpful when trying to reason about relationships between trained models' generated samples and the ground truth data on which they were trained. The practical effects of these behaviours on models of my data are discussed in Chapters 5, 6 and 7, as are their implications for dataset creation and creative applications.

2.3.5 Generative Modelling of Image Data for Audio Visualisation

An additional deep learning task which I have used in this work is image generation applied to visualisation of audio, which, while tangential to the core research questions, deserves mention by virtue of the research innovations that made interesting visualisation of some of the portfolio pieces, for the purposes of online replacedpresentationinnovation, possible. In their implementation of the work presented in the paper 'Training Generative Adversarial Networks with Limited Data'[48], Karras et al presented an adaptation of StyleGAN2, a pre-existing GAN architecture for image generation. A key feature of their adaptation is the ability to achieve high-quality image generation using models trained through 'transfer learning', a process in which the later layers of a trained model are 'unfrozen' and re-trained on new data. While this model architecture has since been superseded by StyleGAN3 and to a greater extent by the advent of diffusion models for image generation, I have found it a valuable and enabling tool for creative engagement with images: its ability to create smooth interpolations between images makes it a tool with unique capabilities and one that is relatively straightforward to use thanks to this transfer learning functionality, which makes it possible to train on a relatively small dataset by retraining only the final layers of a provided, larger pre-trained model. Models that I trained with this architecture were used in conjunction with Mikael Alafriz's 'Lucid Sonic Dreams' program[49] to visualise creative outputs 'SoloSoloDuo' in Chapter 3 and 'Duo I' and 'Gandering 1' in Chapter 5 for the purposes of online presentation.

[46] CJ Carr and Zach Zukowski, 'Generating Albums with SampleRNN to Imitate Metal, Rock, and Punk Bands' (conference paper), MUME 2018: 6th International Workshop on Musical Metacreation, University of Salamanca, Spain, Jun 25-26, 2018

[47] Sander Dieleman, 'Generating Music in the Waveform Domain', blog post *Latest Posts - Sander Dieleman*, March 24, 2020, https://sander.ai/2020/03/24/audio-generation.html.

[48] Karras, Tero, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen and Timo Aila, 'Training Generative Adversarial Networks with Limited Data', unpublished paper, arXiv:2006.06676

[49] Alafriz, Mikael, 'Introducing "Lucid Sonic Dreams": Sync GAN Art to Music with a Few Lines of Python Code!', blog post at *Towards Data Science*, https://towardsdatascience.com/introducing-lucid-sonic-dreams-sync-gan-art-to-music-with-a-few-lines-of-python-code-b04f88722de1.

2.3.6 Ongoing Innovations in Machine Learning for Audio

As should be clear by now, innovation in deep learning architectures across multiple domains and tasks continues apace. In the audio domain it is interesting to note some divergence in recent trends. Generative models of audio continue, for the most part, to become larger, deeper, more data-hungry and energy-intensive, the most extreme example being OpenAI's Jukebox[50]. Antoine Caillon's RAVE[51] architecture is intended more for usage by practitioners than Jukebox, but still requires larger datasets and much longer training runs than any of the architectures explored in this thesis. Both models share the use of Variational Auto-Encoders (VAEs) as a first step for learning latent representations of the input data before any learning for generative modelling begins. An unusual addition to currently available audio generation architectures is 'Catch-A-Waveform'[52], a GAN designed to model and generate variations on as little as a few seconds of audio data. At the lighter end of the computational scale, the Flucoma[53] toolkit, as noted earlier, encourages creative practitioners to find applications of long-established, lightweight machine learning algorithms such as k-means clustering, linear regression and non-negative matrix factorisation.

Innovations in deep learning for tasks not covered in this thesis continue apace. One of the more startling of these is source separation, in which a stereo mix is 'stemmed' into its individual constituent tracks, currently exemplified for application to popular music by Meta AI's Demucs[54] architecture. Another is timbre transfer, in which the timbre of an instrument is modelled such that the sound of an unseen input can be transformed to that the modelled signal, even in real-time; the autoencoder part of aforementioned RAVE architecture and Google Magenta's DDSP[55] are the leading architectures for this task. Most recently, there has been a significant focus among AI companies on developing very large, language-prompted generative models. Examples of these in the audio domain are Stability AI's Stable Audio[56] and Udio's AI Music Generator[57]. While impressive in the quality of their outputs, these companies have also generated significant controversy owing to a perceived lack of transparency around ownership and consent for usage of the music on which they have trained their models. They are also clearly not targeted at music practitioners looking to incorporate AI into their workflows, so I won't dwell on them any further here.

2.4 AI for Musical Creativity

The portfolio outputs of this project are broadly situated in the emerging field of AI for Musical Creativity. Here I identify practitioner-researchers using AI in their musical workflows in order to better locate the portfolio outputs.

[50] Dhariwal, Prafulla, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford & Ilya Sutskever, 'Jukebox: A Generative Model for Music', unpublished paper, arXiv:2005.00341.

[51] Caillon, Anton and Philippe Esling, 'RAVE: A variational autoencoder for fast and high-quality neural audio synthesis', unpublished paper, arXiv:2111.05011.

[52] Greshler, Gal, Tamar Rott Shaham and Tomer Michaeli, 'Catch-A-Waveform: Learning to Generate Audio from a Single Short Example', *DeepAI*, blog post, Jun 11, 2021, https://deepai.org/publication/catch-a-waveform-learning-to-generate-audio-from-a-single-short-example.

[53] Tremblay, Pierre Alexandre, Owen Green, Gerard Roma, Alexander Harker, 'From collections to corpora: Exploring sounds through fluid decomposition', paper presented at ICMC 2019: 45th International Computer Music Conference, Elmer Holmes Bobst Library, MORE, New York University, USA, Jun 16-23, 2019.

[54] Rouard, Simon, Francisco Massa & Alexandre Défossez, 'Hybrid Transformers for Music Source Separation', unpublished paper, arXiv:2211.08553.

[55] Engel, Jesse, Lamtharn (Hanoi) Hantrakul, Chenjie Gu, Adam Roberts, 'DDSP: Differentiable Digital Signal Processing', paper presented at ICLR 2020: Eighth International Conference on Learning Representations, Online, Apr 26 - May 1, 2020.

[56] stability.ai, *Stable Audio*, version 2.0, web service, https://stability.ai/stable-audio.

[57] Udio, *AI Music Generator*, beta version, web service, https://www.udio.com/.

2.4.1 AI, Instrumental Practice and Improvisation

As was the case with finding practitioners of electro-acoustic improvisation on the saxophone, examples of instrumentalists directly applying machine learning methods to their own practice are rare. Jack Walker's 'Power Trio'[58], in which a KNN (or 'k nearest neighbours') classifier is used to mediate interactions between his Derek Bailey-influenced guitar playing and audio samples derived from recordings of bass and drums, is notable for the composer-improviser's instrumental abilities and for being a strong example of using machine learning classification as the backbone of an interactive system for improvisation. In chapter 3 I take a different approach to a similar idea, opting to interact with audio samples of greater length so as to accomodate entire phrases, using a convolutional neural network trained on spectograms in order to classify a larger window of musical content, and opting to write the software in Python rather than Walker's use of MaxMSP/Flucoma.

In 'Dialogues with Folk-RNN'[59], Luca Turchet presents a composition setting his own playing on augmented mandolin alongside outputs of a trained Folk-RNN model. Over the course of the piece, improvisatory sections played on his self-designed 'Smart Mandolin' alternate with sonifications of the model's MIDI outputs. It is unclear whether the 'dialogues' element referenced in the piece's title stretches as far as prompting the trained model on the fly with outputs from the improvised sections; the element of dialogue might be limited to his own improvised responses to the generated material. Turchet's use of real-time digital effects and of a character-level RNN mirrors my own in the portfolio outputs discussed in chapter 6. At points in the research process I did investigate using Char-RNN in a similar fashion, dialoguing with sonifications of its outputs, but disliked the robotic-sounding nature of its outputs and found duetting with them unenjoyable, opting to connect the trained model's outputs to notation software and pursuing creative application of those outputs instead.

A different application of AI to an improvisational instrumental practice is Guilherme Coelho's 'DDSP études for Tenor Saxophone & Violin'[60] in which a pre-trained model of solo violin recordings provided in Google Magenta's DDSP library is used to map the timbre of the violin timbre to the tenor saxophone. While applications of timbre transfer do not feature in this thesis, the work warrants inclusion here for the composer's direct application of AI to instrumental practice and for the obvious presence of improvisation on tenor saxophone in the work.

Practitioners of improvisation using machine learning technologies are, unsurprisingly, easier to find when looking past players of conventional musical instruments and towards computer musicians. Here, notable practitioners include Ted Moore, whose custom-designed software has included his own implementation of a multi-layer perceptron in SuperCollider which can be regarded as a precursor to the FluCoMa toolkit which he also worked on. An example of his use of machine learning in live improvisation can be heard in the piece 'shadow', co-improvised with saxophonist Kyle Hutchins in the group 'Binary Canary'[61]. Here Moore maps timbral analysis (using the mel frequency cepstral coefficient, a measure of the *rate* of frequency change in a window of sound typicaly used to describe timbre) of the outputs of a no-input mixing board to four distinct sound categories using a multi-layer perceptron classifier. These categories are then mapped to changes in the

[58] Jack Walker, 'Power Trio', YouTube video, posted by 'AI Music Creativity 2022', Sep 20 2022, https://youtu.be/U7S8quow0_U.

[59] Luca Turchet, 'Dialogues with Folk-RNN: Smart Mandolin performance at NIME 2018', YouTube video, posted by Luca Turchet, Jul 3 2018, https://youtu.be/VmJdLqejb-E

[60] Guilherme Coelho, 'AIMC 2021 | DDSP études for Tenor Saxophone & Violin', Vimeo video, posted by Guilherme Coelho, Jul 2021, https://vimeo.com/677268564/709378c6cf.

[61] Ted Moore & Kyle Hutchins, 'shadow', YouTube video, posted by 'Ted Moore', 10 June 2021, https://youtu.be/CgALDzMYcbc?si=QfCjxILzPrl9lJnq

lighting in the room. Another practitioner of computer improvisation beginning to incorporate deep learning models into their work is laptop musician Federico Reuben; his recent work in duo with saxophonist Franziska Schroeder features live processing of the outputs of RAVE models trained on her improvising into an improvised duo context.[62]

2.4.2 AI and Composition

The use of generative and predictive machine learning in the context of classical composition has increased recently. This is thanks in large part to the development of an up-to-date, straightforward-to-use and well-documented implementation of the SampleRNN architecture by the Christopher Melen at Royal Northern College of Music's PRiSM lab[63] which is helping to put AI sample generation into the hands of more composers.

An example of the use of SampleRNN as a compositional assistant is in the processes behind Emily Howard's 'Shield'[64] for string quartet. Howard's process here was to create a custom dataset of string quartet performances of her previous works, train a SampleRNN model of this data and transcribe selections of the generated samples. I follow a comparable process in portfolio outputs 'Lrrning' and 'The Lows' described in chapter 5. An interesting observation gleaned from observing a research presentation about this work is Howard's openness to including as the basis of her composed work outputs from earlier training checkpoints when the model is under-trained, and more generally to SampleRNN's quirks, such as its tendency to intersperse periods of grainy near-silence with sudden onsets of noise. In my own work with SampleRNN presented here, while novelty is absolutely a key criterion for sample selection and SampleRNN's self-evident weirdness is enjoyed, I try to prioritise timbral plausibility.

In movement III ('Soul') of Robert Laidlow's orchestral work 'Silicon'[65], playbacks of samples generated from a SampleRNN model of a very large dataset consisting of old BBC Philharmonic radio broadcasts are incorporated to the score, played by the same orchestra. Similarly to Howard, Laidlow's process of sample curation appears to be more open-ended than my own: some of the selected samples feature content modelled from spoken dialogue present in the dataset; this also speaks to a more cavalier approach to dataset building than my own! Also of interest in 'Silicon' is Laidlow's use of symbolic-domain AI in movements I and II ('Mind' and 'Body' respectively) in which he uses outputs from aforementioned Folk-RNN models and also MuseNet (a very large model released by OpenAI, based on their previous language-domain GPT-2 model and trained on a MIDI dataset) as compositional material.

2.4.3 AI and Popular Music

The popular music industry has long embraced new audio technologies, and as such it would be reasonable to expect that this affluent sector would embrace AI-driven innovations. However, popular music's relationship to AI-driven innovations in digital audio looks set to be more complicated, in part *because* of the sector's affluence compared with the improvised and contemporary musics already discussed: artists, copyright holders (many

[62] Franziska Schroeder & Federico Reuben, 'AI Music Improvisation by Franziska Schroeder and Federico Reuben using RAVE model/Stable Diffusion', YouTube video, posted by 'freuPinta', 22 April 2024, https://youtu.be/tI6BMrEf4jU?si=nbw8anNr1XBG6ieG

[63] Melen, Christoper, prism-samplernn, code repository, https://github.com/rncm-prism/prism-samplernn

[64] Emily Howard, 'shield for String Quartet', musical score, 2022, https://www.editionpeters.com/product/shield/ep73579.

[65] Robert Laidlow, 'Robert Laidlow (2022): Silicon', YouTube video, posted by 'RNCM PRiSM', Mar 18 2023, https://youtu.be/3xmpywK0ACA.

of which are record labels) and even sound engineers are on course to see their industries disrupted if innovations in deep learning for audio - particularly language-prompted generative models and AI-automated mixing and mastering - continue at pace. While interesting examples of high-profile pop artists embracing AI can be found, such as Grimes' Elf project[66]added (a public experiment with vocal timbre transfer in which she invites anyone to map what sounds like a DDSP model of her voice onto their own), much of the serious engagement with AI as a creative tool in popular music genres is happening at a lower level of public exposure.

A good place to find examples of this kind of work is in the annual AI Song Contest, in which teams of practitioner-researchers present innovative applications of machine learning-based technologies and, often, technologies they have implemented themselves, to the realm of songwriting. Examples include:

- 'Circus' by a group named 'The elephants and the', who used Tom Collins' Maia Markov algorithm to generate melodies, chords, basslines and beats and combine the resulting song with saxophone riffs and 'improvisation' generated from a WaveGAN model;[67]
- 'echoes from the distance' by ki, who used Google Magenta tools to compose a melody before taking the unusual step of using large language model ChatGPT to suggest chords to accompany the melody;[68]
- 'Noise to Water' by the satirically-named Aiphex Twins, who trained WaveGAN models on kick drum sample packs as well as recordings by Steve Reich, Aphex Twin and Boards of Canada. Leaving aside any potential copyright violation, an interesting aspect of their entry was their conclusion that this approach of using custom WaveGAN models afforded them greater artistic independence and freedom than out-of-the-box tools such as those engineered by Google Magenta.[69]

2.4.4 AI and NIMEs

The field of New Interfaces for Musical Expression shows diverse applications of new machine learning technologies for creative musical workflows. While the work in this thesis focuses on applications of *deep learning* data models trained on high-performance hardware, the NIME field's tendency to target low-latency, real-time applications is also reflected in its approaches to AI. This prioritisation was evident in Pelinski et al's 2022 workshop 'Embedded AI for NIME: Challenges and Opportunities'[70], in which the difficulties inherent in running machine learning algorithms on memory- and compute-constrained hardware were discussed alongside the potential rewards of doing so. Accordingly, recent projects in the area of AI for NIME such as Hantrakul and Kondak's GestureRNN[71], an LSTM-type RNN that targets the proprietary Roli Lightblock hardware, Martin and Torresen's

[66]Grimes, *elf.tech*, web service, https://elf.tech/connect.

[67]Tom Collins, Hanslip, Mark, Maloney, Liam, Quek, Lynette, Rime, Jemily, Zongyu (Alex) Yin, 'Circus', entry to AI Song Contest 2021, https://www.aisongcontest.com/participants/theelephantsandthe-2021.

[68]Ivana Shishoska & Kiril Trbojevikj, 'echoes from the distance', entry to AI Song Contest 2023, https://www.aisongcontest.com/participants-2023/ki.

[69]Phillipp Stolberg & Edgar Eggert, 'Noise to Water', entry to AI Song Contest 2022, https://www.aisongcontest.com/participants-2022/aiphex-twins.

[70]Pelinski, Teresa, Victor Shepardson, Steve Symons, Franco Santiago Caspe, Adan L Benito Temprano, Jack Armitage, Chris Kiefer, Rebecca Fiebrink, Thor Magnusson, Andrew McPherson, 'Embedded AI for NIME:Challenges and Opportunities' in Proceedings of NIME 2022: New Interfaces for Musical Expression, Waipapa Taumata Rau, Aotearoa, University of Auckland, New Zealand and online, 28 Jun - 1 Jul 2022.

[71]Hantrakul, Lamtharn and Zachary Kondak, 'GestureRNN: A neural gesture system for the Roli Lightpad Bloc' in Proceedings of NIME 2018: New Interfaces for Musical Expression, Campus of Virginia Tech, Blacksburg, Virginia, USA, June 3-6 2018.

mixed-density RNN[72] for predicting control parameters and Caramiaux et al's adaptation of a Monte Carlo technique for real-time classification of motion sensor data[73], all target low-dimensionality spaces such as movements and parameter controls that occur *during* creative acts with technologies.

This is a point of both divergence and similarity with the work in this thesis. The majority of this research project focuses on non-real-time approaches to the use of deep learning in the creative process, and broadly takes a 'money is no object' approach to computational needs. However, a point of similarity is in the targeting of parts of the creative process *besides* generation of the final product: outputs of the data models created in the following chapters are enablers of instrumental practice, improvisation and composition rather than generators of it.

2.5 Conclusion

This chapter has provided additional context for this project through a tour of the academic research and musical practices upon which this project builds. I began by describing the historical and current state of the primary musical contexts of solo tenor saxophone improvisation and electro-acoustic saxophone improvisation, in order to provide background for the datasets and portfolio outputs. I then described the technical innovations in the field of deep learning that directly bear on this research. I concluded with a view of practitioner-researchers in the emerging field of AI for musical creativity in order to provide additional context for the portfolio outputs.

The next four chapters (and the datasets and portfolio materials they comment on) constitute the core research carried out as part of this submission. These chapters will be followed by a concluding chapter in which the insights generated through this work will be described.

[72]Charles P. Martin and Toressen, Jim, 'An Interactive Musical Prediction System with Mixture Density Recurrent Neural Networks' in Proceedings of NIME 2019: New Interfaces for Musical Expression, Universidade Federal do Rio Grande do Sul Porto Alegre, Brazil, 3-6 June, 2019.

[73]Baptiste Caramiaux, Montecchio, Nicola, Atau Tanaka, Atau and Bevilacqua, Frédéric, 'Adaptive Gesture Recognition with Variation Estimation for Interactive Systems' in *ACM Transactions on Interactive Intelligent Systems*, Volume 4, Issue 4, 1–34.

Chapter 3

Datasets

3.1 Introduction

This chapter provides detailed views of the audio datasets I created to carry out this research. For each dataset, I first discuss musical and conceptual reasonings behind choices made regarding the recorded content. I then discuss specific aspects of each dataset's musical contents; notated examples further elucidate these choices and better inform the reader of its characteristics. The work in this chapter addresses the first of my research objectives outlined in Chapter 1 of creating a substantial repository of audio data comprised of instrumental practice sessions; it also fulfils the first stage of my research methodology (also outlined in Chapter 1) of 'Dataset Creation', and begins to address the first research question, 'To what extent can recordings of systematic instrumental practice be modelled computationally..?'.

3.2 Rationale

The common aim across all of these datasets was to establish a repository of original raw audio content that would provide a solid foundation for a personal artistic engagement with deep learning. While I acknowledge that personal creative work can also be made using externally-sourced datasets or pre-trained models, a motivator for this project was to explore and learn more about the space in which my own music resides and, in doing so, expand my creativity outwards and yield insights to the usefulness of deep learning to other practitioners. It therefore seemed clear that using my own data would be an essential part of the process. More broadly, I reasoned that using personal data makes for inherently more personal model outputs and speaks to a human-centred approach to AI than simply modelling externally-sourced data; doing so also sidesteps any potential ethical issues arising from using datasets containing copyrighted materials.[1]

3.2.1 Process of Recording Datasets

When recording each dataset I followed some informal, self-imposed guidelines intended to ensure a baseline of quality and cohesion of the raw data. These can be summarised as ensuring that each dataset is cohesive in musical content, pertain to a specific aspect of my practice and that the recorded sound should be, as far as reasonably practical, clean, consistent across all datasets (thus allowing for potential recombinations) and free of extraneous sounds.

The first and second points are closely related: since the aim was to create computational models of specific aspects of my practice, addressing the second aim to a degree

[1] Franceschelli, Giorgio and Mirco Musolesi, 'Copyright in generative deep learning' in *Data and Policy* 4, Cambridge University Press, 25 May 2022.

also addresses the first. The last point was addressed by using the same microphone, audio interface and software for all datasets (DPA 4099 clip microphone, Audient iD4 and Audacity respectively). My choice of microphone was informed by being advised by two sound engineers I had previously worked with, Miles Ashton (resident engineer at Ronnie Scott's Jazz Club in London) and Alex Fiennes (experienced and respected freelance sound engineer), of its strong off-axis rejection, which I reasoned would be helpful in mitigating the effect of recording in different rooms should that necessarily be the case. Additionally, I was working under the assumption that a clip microphone would result in greater consistency of signal than a stand-mounted microphone, since consistency with the latter would be more vulnerable to changes of standing position. I had also heard highly favourable reports of its sound quality from musical colleagues such as cellist Ecka Mordecai and reeds player Michael Perrett.

As we will see in the chapters on generative modelling, the initial confidence I took from this setup turned out to be slightly mistaken. The DPA 4099 is indeed an excellent microphone and was a solid practical choice for its room rejection, portability and ease of setup, but in reality it is not feasible to capture the full complexity of the tenor saxophone's acoustic output with a single microphone and certainly not with one that is attached to the bell section. The saxophone projects sound in myriad directions depending on what register of the instrument one is playing in. Capturing the sound around the bell section overly privileges the lower notes at the expense of more 'vented' notes, the sound of which escapes from open holes across the length of the instrument's body. I will revisit this point in Chapter 5, where this initial choice has implications for creating statistical models of the raw signals.

3.3 Musical Contents of Datasets

3.3.1 Exercises Datasets

As discussed in the Chapter 1, the initial impulse for this project was a curiosity about the relationships between systems of musical information in my private practice and creative outcomes when I improvised; this idea provided a starting point for recording the Exercises Datasets described below.

Tone Rows

The first dataset I created consists of exercises extrapolated from Nicolas Slonimsky's Thesaurus of Scales and Melodic Patterns[2]. I was first made aware of this book through hearing about its importance to historically significant improvising saxophonists such as Charlie Parker, John Coltrane and later Evan Parker[3]. Within my own practice I became most absorbed in the 12-tone row section of the book. What attracted me was its organisational principle of basing each row on a single interval and how this lent itself to a structured practice routine. The motivation behind incorporating 12-tone rows into my practice routine was a desire to bring a greater degree of abstraction to the kinds of melodies I was improvising. This kind of practicing correlates with what Derek Bailey describes as 'exercises worked out to deal specifically with the manipulative demands made by new material'[4].

Initially I recorded permutations of tone row exercises explicitly based on those in Slonimsky's volume. To maintain some consistency and quality, I first created a set of play-along tracks of these rows using functionality of SuperCollider's pattern classes. First,

[2]Slonimsky, Nicolas, *Thesaurus of Scales and Melodic Patterns.*
[3]Evan Parker, interviewed by Frances-Marie Uitti, *Contemporary Music Review.*
[4]Bailey, Derek, *Improvisation*, Da Capo Press, 1992, 110.

I programmed midi note array representations of each row, before wrapping these arrays in a pattern-generating function and recording a basic synthesised output to play along with at recording time. While only published beginning on low B-flat in the thesaurus, all tone rows adapted from the thesaurus appear in the dataset in all their 12 possible permutations and across different registers of the saxophone as well as in alternate variations throughout the Tone Rows dataset: every permutation of each row was looped 4 times, starting on low B-flat and transposing the entire row up a semitone each time as far as my range on the tenor saxophone permitted. Rows were recorded as far as altissimo F-natural one octave above the saxophone's non-altissimo register. Seeking variation and challenge, I then returned to the pitch sequences written in SuperCollider and proceeded to generate variations on them using randomisation. I also composed a tone row based on intervals of minor 9ths organised by the same logic as the major 7ths row (see Figure 3.11) of transposing three ascending interval pairs by ascending major thirds followed by three descending interval pairs by descending major thirds, effectively a tritone transposition of the first three pairs. By the time I recorded the randomised equivalents of the rows based on intervals of 7ths and 9ths I decided to raise the level of difficulty by transposing randomly selected notes by one or more octaves.

Below are notated examples of each of these rows starting at the lowest point of the saxophone's range, with its corresponding randomised tone row.

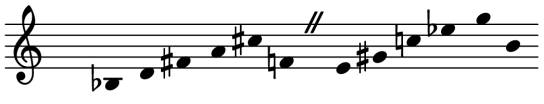

Figure 3.1: The original '3rds' tone row from Nicolas Slonimsky's Thesaurus of Scales and Melodic Patterns.

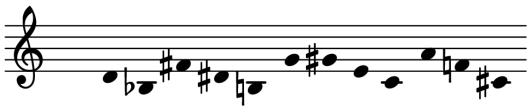

Figure 3.2: An example of a variation on Figure 3.1 that appears in the Tone Rows Dataset.

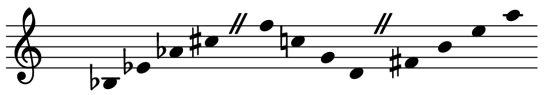

Figure 3.3: The original '4ths' tone row from Nicolas Slonimsky's Thesaurus of Scales and Melodic Patterns.

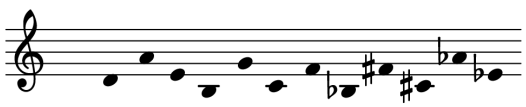

Figure 3.4: A variation on the '4ths' tone row generated using randomisation of the pitches seen in Figure 3.3.

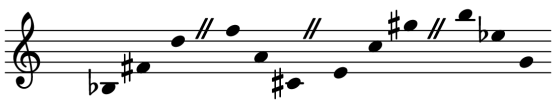

Figure 3.5: The original ‘6ths’ tone row from Nicolas Slonimsky’s Thesaurus of Scales and Melodic Patterns.

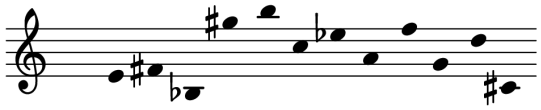

Figure 3.6: A variation on the ‘6ths’ tone row generated using randomisation of the pitches seen in Figure 3.5.

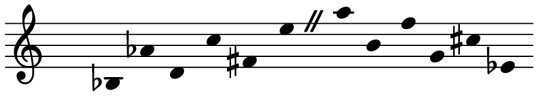

Figure 3.7: The original ‘minor 7ths’ tone row from Nicolas Slonimsky’s Thesaurus of Scales and Melodic Patterns.

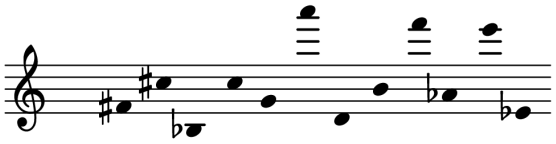

Figure 3.8: A variation on the ‘minor 7ths’ tone row generated using randomisation of the pitches seen in Figure 3.7, with additional randomised octave displacement of the pitches for greater interest and technical challenge.

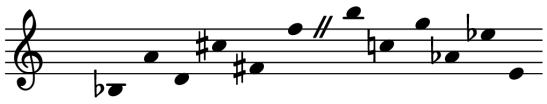

Figure 3.9: The original ‘major 7ths’ tone row from Nicolas Slonimsky’s Thesaurus of Scales and Melodic Patterns.

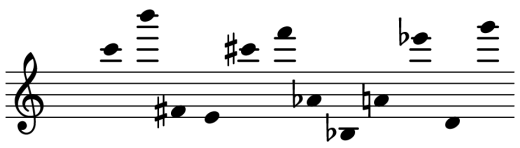

Figure 3.10: A variation on the ‘major 7ths’ tone row generated using randomisation of the pitches seen in Figure 3.9. As with the previous minor 7ths row, I saw fit to increase the difficulty and interest level by displacing randomly chosen pitches up or down an octave.

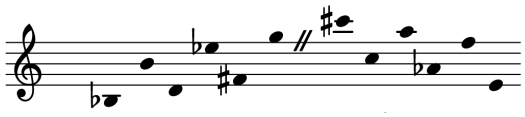

Figure 3.11: A tone row based on the interval of a minor 9th featuring the same structural logic as seen in Figure 3.9.

Scales and Arpeggios

Around the same time as creating the 'Tone Rows' dataset, I created another focusing on a more mundane aspect of my practice routine: scales. For a long time, a cornerstone of my routine had been to practise the overtone series starting on each note in the lower register followed by exercises based on the fundamental pitch's lydian dominant scale. The choice of scale was informed by its correspondence to the sequence of pitches in the overtone series:

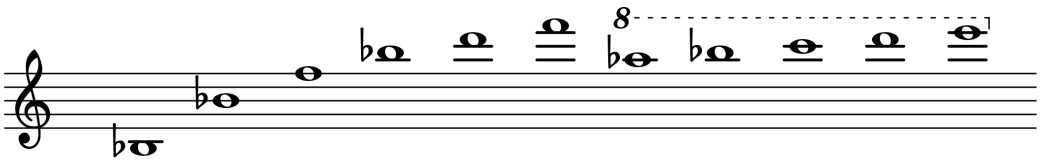

Figure 3.12: notes of the overtone series beginning on low b-flat

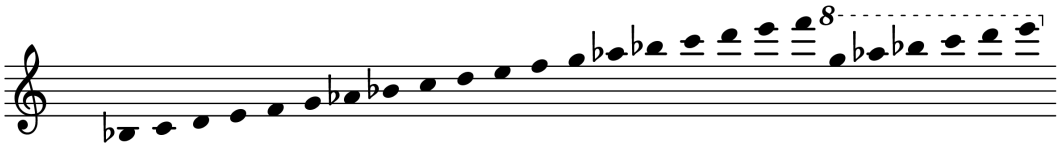

Figure 3.13: Lydian dominant scale starting on low b-flat, corresponding to the notes of the overtone series beginning on low b-flat.

This aspect of practicing correlates with what Derek Bailey describes as 'the musical equivalent of running on the spot, the sort of thing which might be useful to any player of any music'[5]. Since this practice sequence was well embedded in my regular routine committing it to recording seemed a straightforward and natural choice.

I initially recorded the Lydian dominant scale beginning on all notes between low Bb and E-natural, both staccato and legato. I then recorded the melodic minor scale beginning on low B, rising a semitone each time as far as low E. In doing so I covered most of the saxophone's range and in all 12 keys.

[5]Bailey, Derek, *Improvisation*, 110.

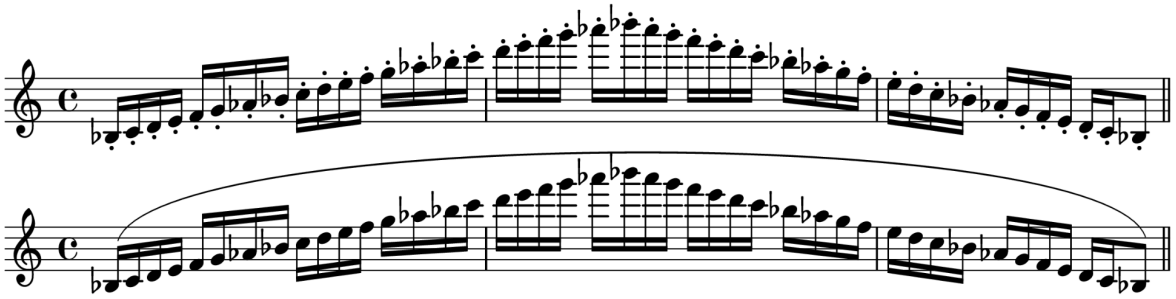

Figure 3.14: Lydian dominant scale, played staccato and legato.

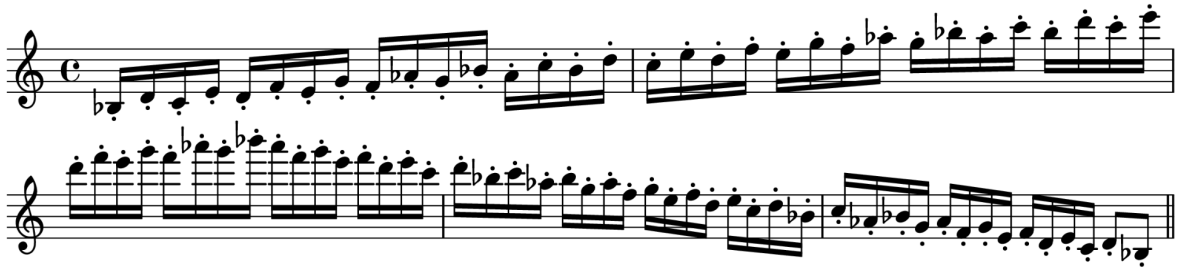

Figure 3.15: Lydian dominant scale in 3rds, played staccato.

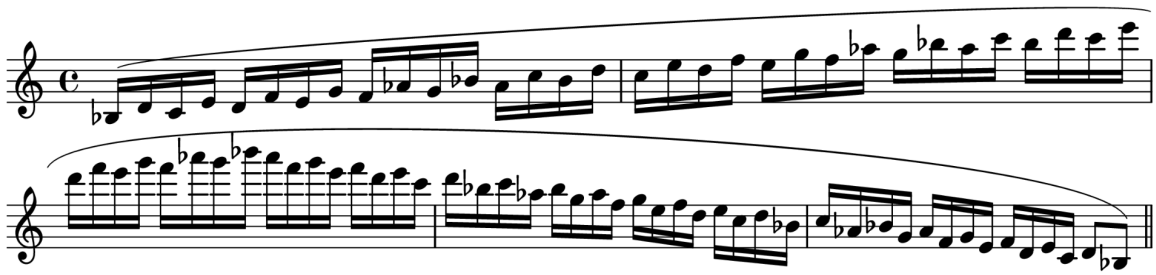

Figure 3.16: Lydian dominant scale in 3rds, played legato.

I then recorded some arpeggiated exercises based on these scales:

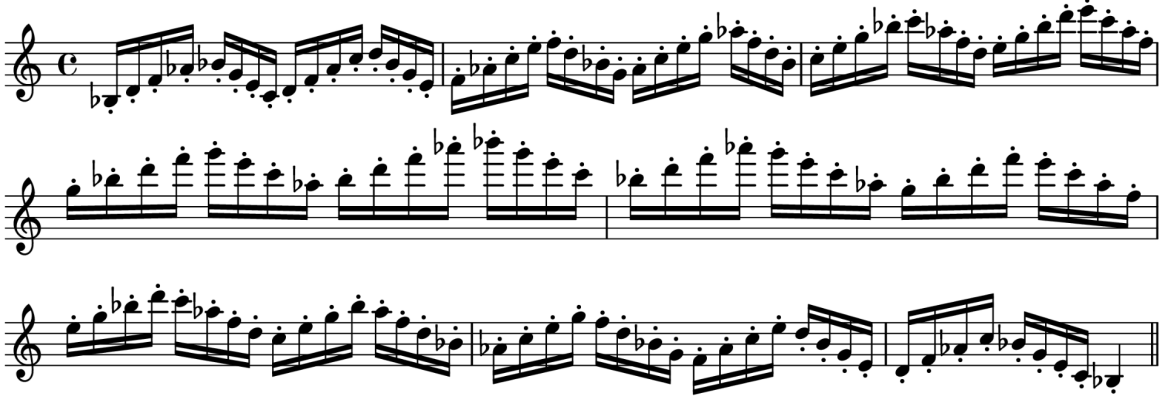

Figure 3.17: Arpeggiated B-flat lydian dominant exercise, played staccato.

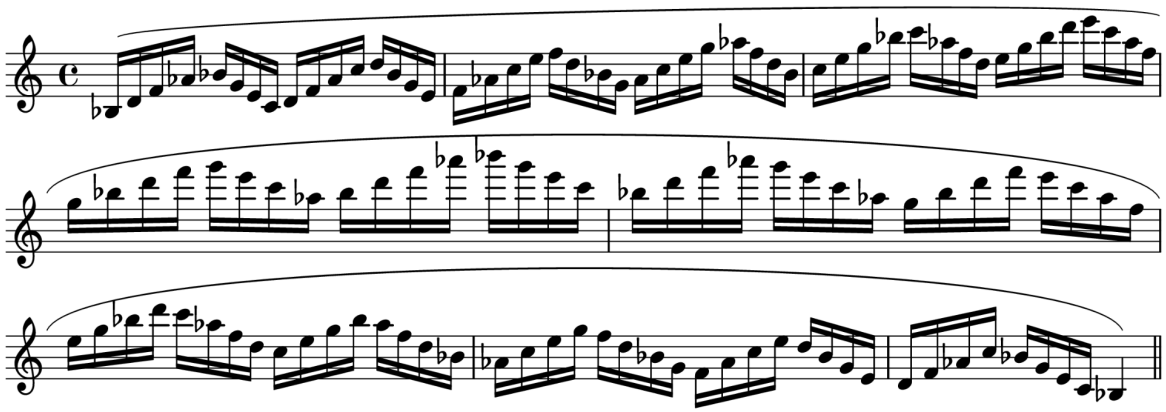

Figure 3.18: Arpeggiated B-flat lydian dominant exercise, played legato.

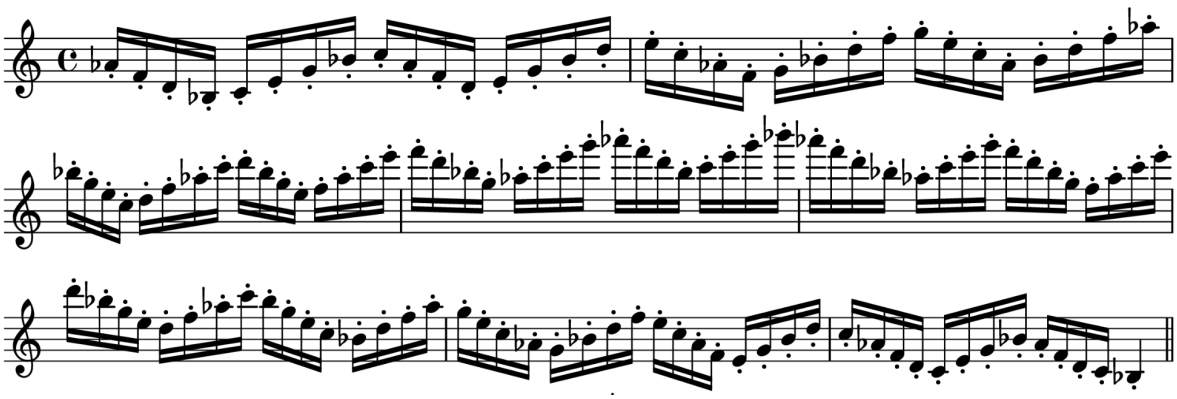

Figure 3.19: Inverse arpeggiated B-flat lydian dominant exercise, played staccato.

In the case of both the 'Tone Rows' and 'Scales and Arpeggios' datasets, my motivations were twofold. Firstly, I wished to build on my practice routine by factoring in additional challenges. Mainly though, I was motivated by the idea outlined in Chapter 1 of building computational models of the process of practising and of the impact a practice routine has on creative outputs. The possibility of modelling this process was an attractor to working with the generative and predictive machine learning architectures explored in Chapters 5 and 6. I also simply wanted to create as much original training data as reasonably possible so as to have multiple options for modelling later in the project.

3.3.2 Register-Specific Exercises and Improvisation

Lower Register Exercises and Improvisation

This dataset was the first of three in which I chose to focus exclusively on a specific register of the saxophone.

The following two examples show the range of musical freedoms exercised within the dataset, from stricter routines such as that in the first figure, to the more freely improvised material in the following figure, where the only constraints are to use melodic material (as opposed to timbrally-focused material) and to stick to the bottom (below middle-D) key register of the tenor saxophone:

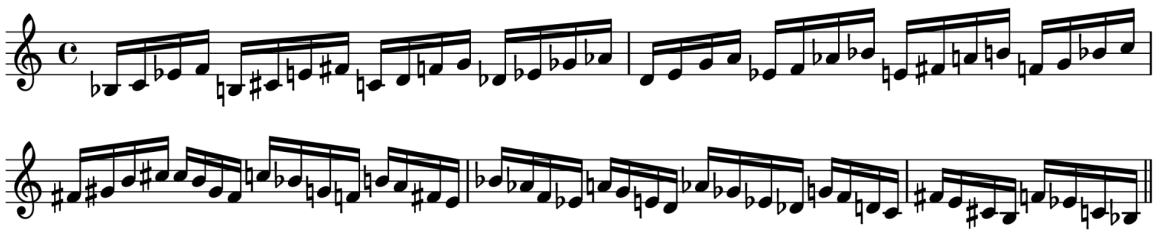

Figure 3.20: In this example from the lower register dataset a cell of '1-2-4-5' interval structure is transposed up a semitone on each iteration.

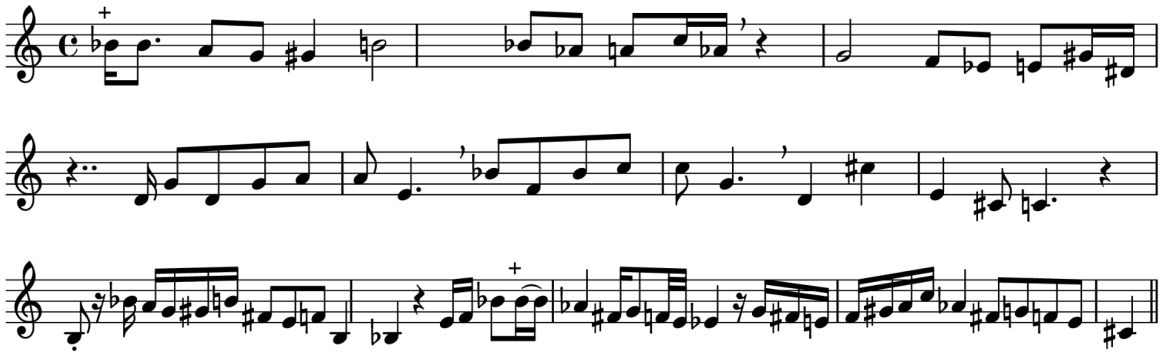

Figure 3.21: More freely improvised material from the 'Lower Register' dataset.

Middle Register Exercises and Improvisation

In the sub-dataset focusing on the middle register there was a mix of exercise-like material and more freely improvised phrases, with the two often combined. In figure 3.22 below, a semi-improvised phrase from the dataset is clearly based on a 1-2-4-5 melodic shape (previously shown in an explicitly exercise-like format in Figure 3.20):

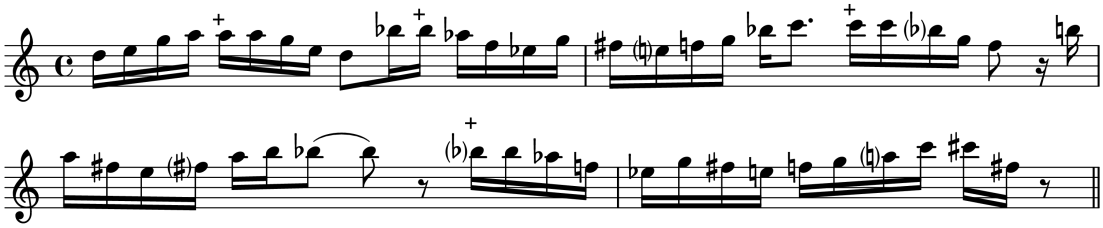

Figure 3.22: An example from the middle register dataset in which an improvised phrase contains transpositions of a cell of '1-2-4-5' interval structure.

A section of the dataset also contains quarter tone exercises as a precursor exercise to the use of microtones in my improvisations, a fairly niche endeavour owing to the musical territory microtones inhabit. Notably championed in improvised music by violist Mat Maneri who was introduced to the concept by his reeds-playing father Joe, microtonal pitch deviations on the saxophone can be achieved through contortion of the embouchure and through non-standard fingerings, my preference being towards the latter. At the time of recording this dataset I was trying to internalise a set of non-standard physical key combinations that seemed to me the most immediately practical options for passing easily between adjacent quarter tones. I did not manage to find a key combination for G quarter

sharp, hence in the example below I skip between G sharp to G natural before resuming the exercise.

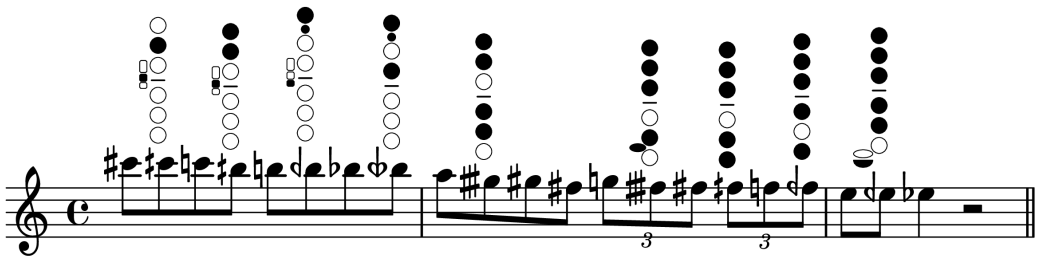

Figure 3.23: A descending quarter tone scale exercise from the middle register dataset.

Upper Register Exercises and Improvisation

The upper register-focused portion of this dataset is significantly less improvisatory than those focused on the lower or middle, for straightforward reasons. The most glaring of these is that my technical flexibility in the false upper region of the saxophone is significantly less than in the lower and middle registers. Secondly, my vocal range does not extend upwards to anywhere near this portion of the instrument's range, which results in a less resonant tone than I am able to achieve in the lower regions of the instrument. Lastly, I simply don't *hear* much in the way of improvisatory material in this region, possibly owing to it sitting so far outside of my vocal range (I like to be able to sing the things I play). Much of my engagement with this region of the instrument is more motivated by a desire for technical completeness and to keep up with the expected professional technical standard on the instrument than by any wish to spend much time up there. For these reasons, purely technical exercises make up the majority of this dataset.

In the examples below, exercises based explicitly on intervals of a major seventh and major sixth respectively are practiced into the false upper register:

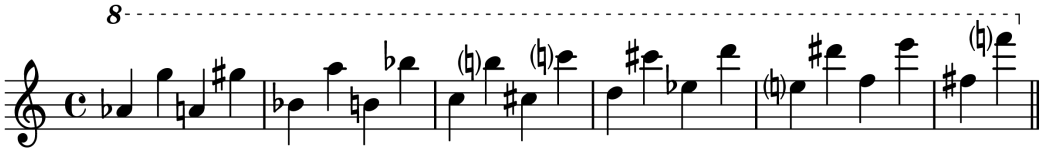

Figure 3.24: An ascending major 7ths exercise from the upper register dataset.

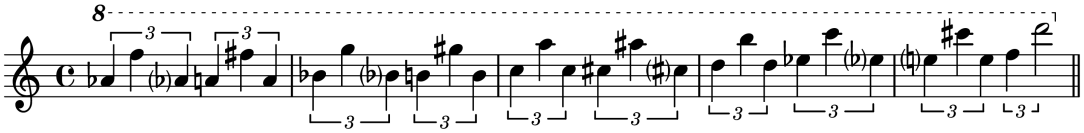

Figure 3.25: An ascending major 6ths exercise from the upper register dataset.

3.3.3 Improvisation Datasets

These datasets contain recordings of the practise of solo improvisation, focused on two distinct aspects of my practice which can most straighforwardly be characterised as melodic and timbral improvisation.

There were several motivations for making these recordings. Balancing the content of all my datasets was one: since those I had created up to this point consisted either wholly or partially of exercises, it made sense to also have some consisting more of improvised material. Another motivation concerned specific deep learning tasks that I was at an early stage of acquiring knowledge of: for real-time discriminative tasks, for example, it seemed important to have data that was representative of material I would realistically play in a performance context; I was additionally excited by the possibility of generative modelling of improvisation, unaware at this point of the practical challenges of modelling more diverse audio datasets. An additional musical motivator, first signposted in Chapter 1, was a desire to engage more deeply with solo improvisation, an endeavour I had found challenging. This area of practise correlates with what Bailey calls 'the bridge between technical practice and improvising'[6].

In the following sections I will describe the characteristic contents of each dataset as well as their idiomatic and cultural associations.

Melodic Improvisations

This dataset is comprised of mainly melodic musical material played with a largely conventional instrumental technique. The content of this material is derived from a variety of sources. The post-bop vernacular, where chromatic enclosures decorate otherwise recognisably tonal passages, pentatonic motifs frequently subjected to transpositions pertaining to their implied key centres, and more abstract intervallic melodies inhabiting a language closer to that of Schoenberg's serialism and of course derived from my aforementioned practice of tone rows. While examples of phrases based solely on each type of material can be found, phrases are also often made up of combinations of these approaches.

In the extract below, a phrase clearly derived from combining ideas from the '6ths' and '3rds' tone rows illustrated in figures 3.1 and 3.4 is then answered with a second phrase still referencing the '3rds' tone row but clearly suggestive of a C major tonality with an augmented fifth, providing some resolution to end the answering phrase:

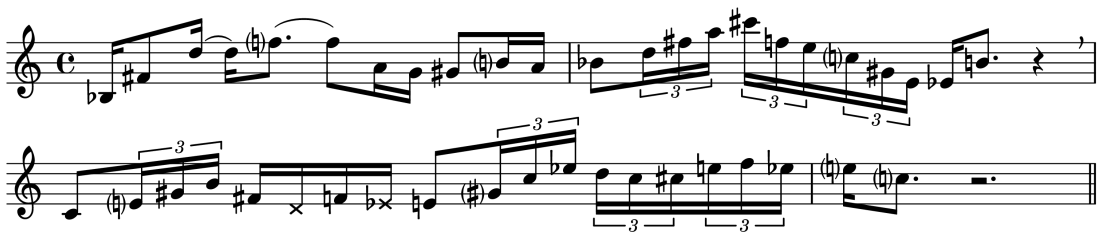

Figure 3.26: A pair of tone-row derived question-and-answer phrases from the 'Melodic Improvisation' dataset.

In this example, an E-flat minor pentatonic shape gives way to post-bop lines suggesting B major, F minor and G melodic minor tonalities:

[6]Bailey, Derek, *Improvisation*, 110.

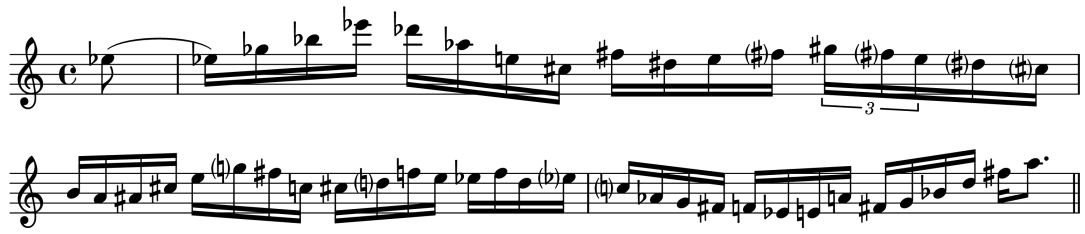

Figure 3.27: A more post-bop-style phrase from the 'Melodic Improvisation' dataset that passes through a number of implied key centers.

Where instrumental technique in this category could be said to deviate from the purely conventional is in the use of the altissimo or 'false' upper register and in the occasional use of microtones. False upper register playing requires the use of non-standard fingerings to exceed the saxophone's intended upper register limit, but its use among tenor saxophonists operating across musical genres is now so ubiquitous that to describe it as a non-conventional technique seems outdated (use of microtones in improvisation on the saxophone is described above in relation to figure 3.23 from the 'Middle Register' dataset). The extract below, which is derived from the ascending major 7ths exercise shown in figure 3.24 earlier, illustrates my general approach to altissimo playing as a means to simply extend the range of my usual phrases rather than as a distinct timbral effect or means of projecting emotional intensity:

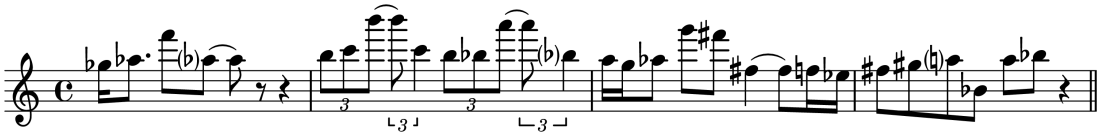

Figure 3.28: An example from the 'Melodic Improvisation' dataset with respect to figure 3.24 from the 'Upper Register' dataset.

Timbral Improvisations

This dataset is principally composed of a more timbre-based mode of improvisation in which the 'extended' vocabulary of the saxophone, most commonly multiphonics generated by non-standard combinations of keys and concurrent manipulations of the breath and embouchure, is emphasised. I present this aspect of my practice separately from melodic improvisation not simply because of its 'extended technique' classification but also because it represents an alternative way of thinking about the instrument: considering the saxophone as a *closed tube* first from which holes are then vented. The musical territory these techniques open up is quite distinct from jazz, sharing more in common with contemporary music, sound art and free improvisation, and is an additional reason to consider them separately.

A defining feature of this aspect of my practice is my tendency to either deploy these sounds in a drone-like fashion or to apply rhythmic variations to multiphonics. In the first figure below, a lower-register multiphonic is sustained, whereas in the second figure small rhythmic variations are applied to the same multiphonics by quickly venting and re-closing keys.

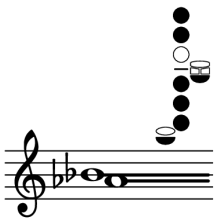

Figure 3.29: A sustained multiphonic.

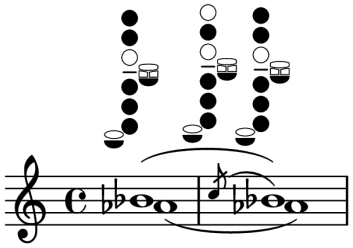

Figure 3.30: A multiphonic with rhythmic activity through key-venting.

A feature of multiphonics on the saxophone is their tendency to reveal themselves in 'clusters': once you find one, there is a good chance you will find others nearby. Another is their tendency to bear close similarity either through pitch content or fingering to regular notes on the saxophone, allowing for associations to be made. An example of both of these features is notated below, in which a low E-flat is used as a springboard to multiphonics both containing E-flat and with closely related fingerings:

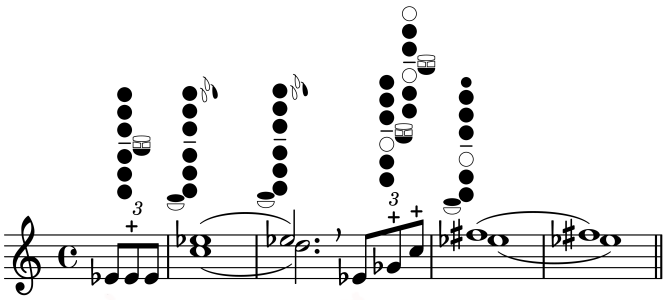

Figure 3.31: A phrase containing three related multiphonics stemming from the conventional E-flat fingering, as played in the Timbral Improvisation dataset.

Other multiphonics, however, bear little relation to conventional fingerings, have a less rational spread of pitches and are probably better found through more randomised experimentation:

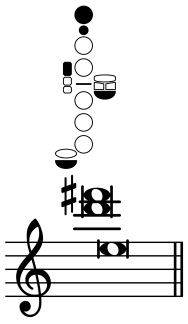

Figure 3.32: One of the more obscure multiphonics from the Timbral Improvisation dataset.

While I have personally found it preferable to discover multiphonics on the saxophone through a mix of controlled and more randomised experimentation, others may find pedagogical texts to be useful resources; Daniel Kientzy's 'Les sons multiples aux saxophones' stands out among available texts for its remarkable thoroughness and coverage of five members of the saxophone family.[7]

3.3.4 Major-Key Fixed-Tempo Improvisation and Exercises

This dataset was created in response to the remit of contributing AI-generated saxophone phrases to a pop song for submission to the 2021 AI Song Contest. A mix of loosely defined exercises and more improvisational material was recorded according to the parameters of the proposed song, which were a tonal centre of concert G major and a tempo of 120 beats per minute.

Although I have significant past experience of playing and recording strictly tonal music over the years in more commercial idioms than those I choose for my creative practice, I have more recently grown accustomed to the tonal and rhythmic freedoms afforded by free jazz and improvised music. As such, playing unaccompanied in the same key at the same tempo for long stretches of time quickly became tedious. To add stimulation, I recorded otherwise-similar material in the keys of concert G flat and A flat major before adjusting their pitch to G major using Audacity's built-in pitch shift algorithm.

The resulting raw dataset was just over 50 minutes of harmonically and rhythmically consistent saxophone phrases and exercises.

In this first example a phrase derived from the dominant bebop scale on E7 gives way to B minor pentatonic shapes. A rhythmic device popular with many contemporary jazz saxophonists is used in which an eighth-note phrase is grouped by a compound rhythm, in this case triplets:

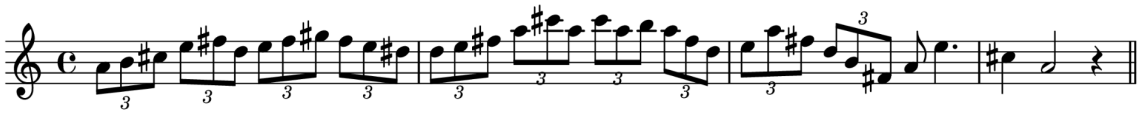

Figure 3.33: An improvised phrase in concert G major with triplet groupings.

In this second example, a post-bop-styled phrase with chromatic enclosures around

[7] Kientzy, Daniel, *Les sons multiples aux saxophones : pour saxophones sopranino, soprano, alto, ténor et baryton'*, Salabert, 1982.

an Ama7 (at instrument pitch) tonality gives way to a descending 5-3-2-1 melodic shape through the A major (at instrument pitch) scale in a style derived from Ornette Coleman:

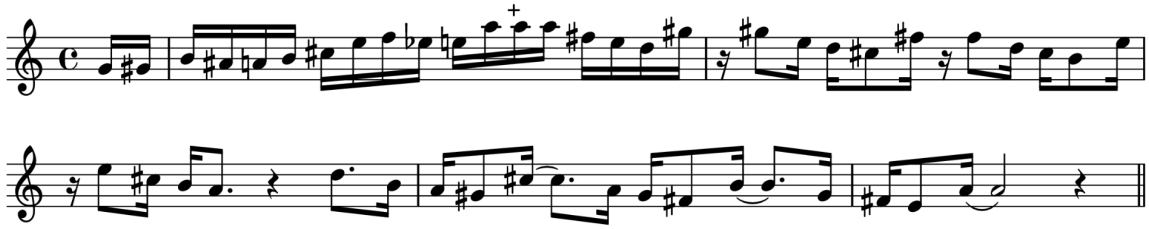

Figure 3.34: An improvised phrase in concert G major.

3.4 Conclusion

In this chapter, I have provided insight to the musical contents of the audio datasets I created for this project. The practical work described in this chapter fulfils the first of my research objectives: it has resulted in several hours of audio data to work with and, crucially, covers a range of aspects of instrumental practice, ranging from the purely technical (as in the 'Scales and Arpeggios' and 'Tone Rows' datasets) through to the practice of solo improvisation (as in the 'Melodic Improvisation' and 'Timbral Improvisation' datasets) and points on the spectrum in between the purely technical and the improvisational (as in the register-focused datasets and the 'G Major' dataset).

In fulfilling this first objective, the groundwork has been laid for answering the first research question, 'What are the practical implications of using recordings of systematic instrumental practice on the saxophone as training data for deep learning models for applications in creative music practice?' - having established a repository of recordings of instrumental practice, it is now possible to move forward onto answering this question more fully through experiments with modelling the data, and to address the remaining two research questions outlined in 1.1. These datasets will be referenced throughout the remainder of this thesis when discussing their use as training data for creating a variety of statistical models through deep learning. Selected models trained on these datasets will in turn form the basis for the creative works in the accompanying portfolio.

A selection of the datasets described in this chapter are available for others to download and use at this HuggingFace repository:

https://huggingface.co/datasets/markhanslip/markhanslip_phd_saxophone_data

Chapter 4

Audio Classification in Practice: 'SoloSoloDuo'

4.1 Introduction

This chapter presents an enquiry into the application of deep neural audio classification to improvised instrumental practice. The goal here was to create a computational tool able to distinguish the fundamental idioms within my practice with a view to this tool being used to mediate interactions. I begin by discussing the musical rationale behind the choices of categories in 'SoloSoloDuo' and for including the process of dataset creation within the final piece. I then describe the various processes involved such as data preprocessing, choosing an appropriate audio data representation and tuning its parameters, model architecture and training parameters in light of their intended purpose. I then present two versions of 'SoloSoloDuo', a structured improvisation for saxophone and computer which represents these processes.

The research described in this chapter represents the first attempt to provide answers to the first two research questions in 1.1, regarding the practicalities of training useful machine learning models of instrumental practice and the potential creative applications thereof. It represents the beginning of my efforts to address the second research objective of developing my understanding of and proficiency in the computational aspects of this research - since classification tasks are often regarded as the 'Hello World' of machine learning[1], it seemed a sensible place to start. Following on from this, the research presented in this chapter and Chapters 5 and 6 fulfils the subsequent objectives of developing machine learning pipelines, training models, developing additional scripts for working with the models, investigation of creative applications of the models and creation of audio recordings and accompanying scores. It also begins to address the objective of developing code for the processes outlined in this thesis and making those available to others. The processes described in this chapter can be found in the accompanying folder or at https://github.com/markhanslip/PhD_Ch4_CNN.

4.2 Rationale

In this section I will discuss distinctions between the modes of improvisation that characterise the 'Melodic Improvisation' and 'Timbral Improvisation' datasets and why these distinctions translate to a valid use-case for neural audio classification.

The distinction between these two broad areas of my practice is, from my perspective as the player, strongly embodied and based on my subjective experience. replacedTthe

[1]Unknown author, 'Say hello to the "Hello, World" of machine learning', webpage, https://developers.google.com/codelabs/tensorflow-1-helloworld.

distinction between conventional and so-called 'extended' instrumental techniques - standard and idiosyncratic – correlates with the contrasting physical sensations experienced while improvising in each mode. The former, being more rhythmically active and based largely on pitch relationships, emphasises technical fluency and dexterity in the hands, with the fundamental concerns of breath and embouchure feeling more automatic and subconscious. The latter relies more on internalisation of unconventional, comparatively static finger positions and emphasises the role of the breath and embouchure, with non-conventional manipulations of each required to produce certain sounds.

These experiential contrasts in turn speak to how I perceive these areas of my practice and how I perceive my practice as a whole, much less to an analytical perspective. It then follows that the task of creating an algorithm capable of delineating these two areas is an appropriate one for deep learning processes. 'Human-level' tasks such as this that would seem unreasonably complex to computationally model via a hand-coded program are especially suitable for deep learning-based approaches, the thinking being that if a human can differentiate between two approaches, then a deep learning model can also be trained to do so. It seemed therefore clear to me that differentiating these two musical areas was a valid use case for neural audio classification.

4.3 Technical Processes

In this section I will describe steps taken to create a discriminative model of the melodic and timbral improvisation datasets. These include preparing the raw audio files to make them suitable for use as classification data, augmenting the data to improve model performance, selecting an appropriate input representation and parameters thereof, and some practicalities of model training.

4.3.1 Data Pre-Processing

It stands to reason that the dataset should not contain any significant periods of silence, as they are tangential to the core task of differentiating musical content and would negatively impact model performance. Recalling the initially proposed use case of real-time inference over a live audio input, there are straightforward means by which silences, which would not be passed to the classifier, can be detected and filtered, such as using loudness analysis in combination with an amplitude threshold. Therefore the first preprocessing step was to remove any periods of silence according to given amplitude and time thresholds.

To maximise dataset size and hopefully improve model performance, the input audio was augmented by pitch shifting it by one semitone in either direction and concatenating the resulting audio, effectively increasing dataset size by a factor of three.

4.3.2 Data Representation

When choosing an audio representation and parameters therein, a key factor was perceptual validity: ensuring that the musical content contained within the representation is preserved. To first choose which input representations to experiment with, I consulted existing research on audio classification.

Muhammed Huzaifah surveyed a range of input representations for environmental audio classification tasks[2], favouring the commonly-used mel spectrograms but also positing that constant-Q transform (CQT) spectrograms may be better suited to musically focused discrimination tasks. After a period of my own experimentation with a range of input

[2]Huzaifah, Muhammed, 'Comparison of Time-Frequency Representations for Environmental Sound Classification using Convolutional Neural Networks.'

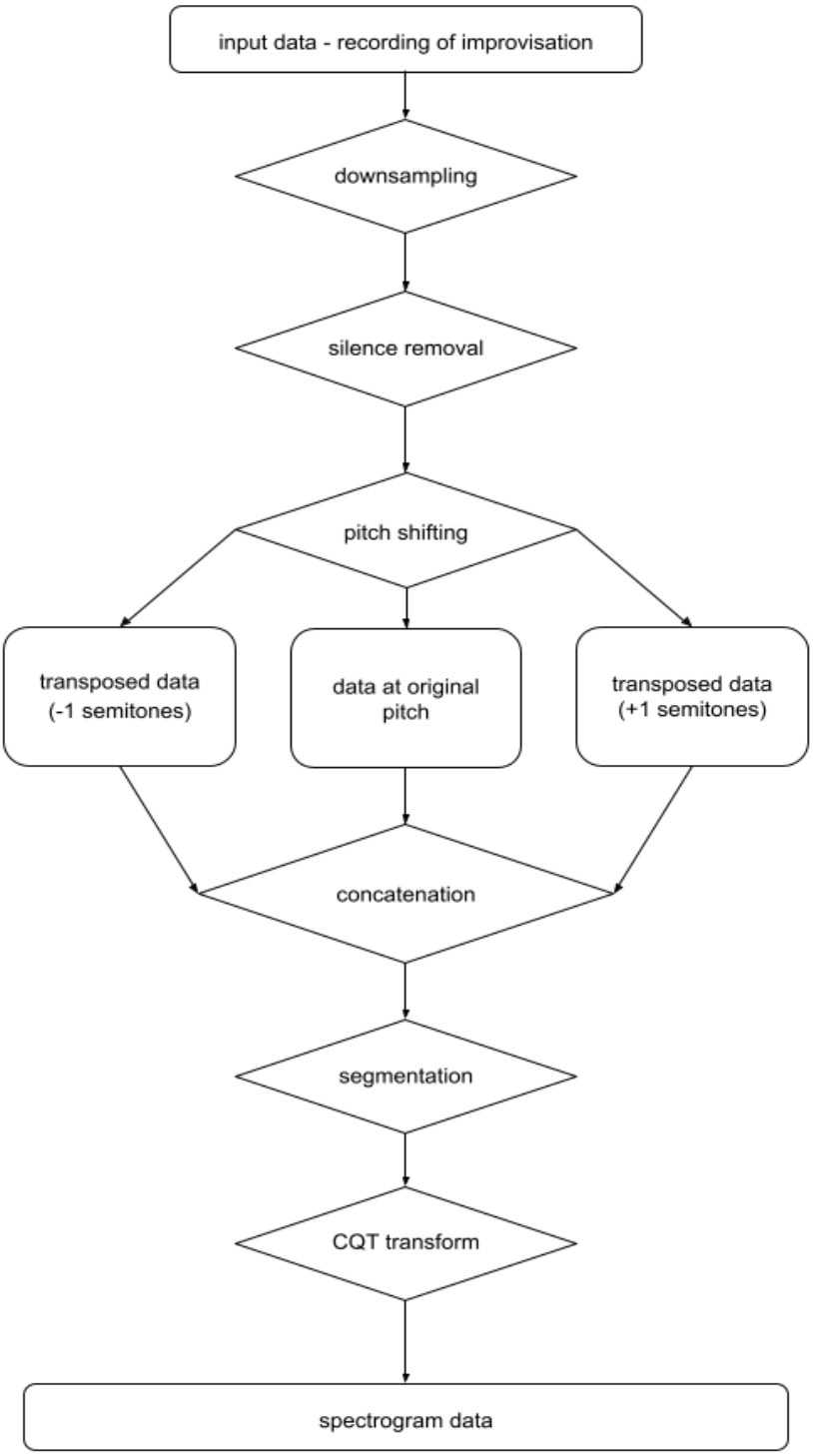

Figure 4.1: Flow diagram showing all stages of pre-processing the training data for this audio classification task.

representations, while mel spectrograms performed significantly better than STFT spectrograms there was a further improvement in classification accuracy when using the CQT for this task.

This makes sense considering that the CQT transform's use of variably-sized wavelets takes into account the greater perceptual significance of lower frequencies (i.e. the gap between two lower frequencies is perceptually and musically more important than the equivalent gap between two higher frequencies). This musically- and perceptually- motivated variable time/frequency resolution would appear to be the CQT transform's advantage over other input representations for my use-case and I would imagine other musically- and perceptually- motivated classification tasks.

When experimenting with input representations and tuning their parameters, a key to ensuring preservation of musical content was the use of the Griffin-Lim method[3] [4] to check for invertibility, ensuring audio spectrograms could be rendered back to their original audio content without excessive distortion of musical content.

Source audio from each class is separated into 32768-sample-long segments. Parameters for spectrogram generation are set to hop length 512, with the minimum frequency set at 50Hz to accommodate the lowest register of the tenor saxophone and ensure preservation of musical information, and resulting in 64x64 pixel-sized CQT spectrograms. Since at the time I was developing this process with the idea of training a model for live performance, I was keen to minimise spectrogram processing times as much as possible; to this end, I wanted a package that would improve on Librosa's[5] speeds, since the CQT is computationally intensive and time-consuming. This is not to criticize Librosa - their library contains ingenious use of the Numba project[6] for just-in-time compilation - but I found that spectrogram calculation and render time could be significantly lowered by using the nnAudio library[7] which offers a speedup versus Librosa of around a factor of 4 in this instance as it is able to leverage GPU computation.

CQT parameters used were tested to ensure that the spectrogram transforms could be inverted and rendered back to audio (using Librosa's Griffin-Lim CQT method) while preserving the core musical information, the intention again being to ensure all data passed to the model for training is perceptually representative.

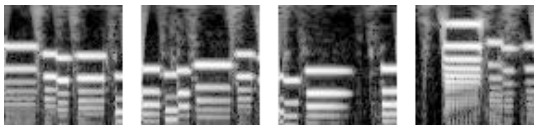

Figure 4.2: CQT spectrograms of melodic saxophone improvisation.

[3]Griffin, D.W and J. S. Lim, 'Signal estimation from modified short-time Fourier transform,' *ASSP* 32, no.2, EEE Trans., Apr, 1984, 236-243

[4]Perraudin, N., P. Balazs and P.L. Søndergaard, 'A fast Griffin-Lim algorithm,' *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Oct, 2013, 1-4

[5]McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, 'Librosa: Audio and Music Signal Analysis in Python.', Proceedings of the 14th Python in Science Conference, pp. 18-25. 2015.

[6]Kwan Lam, Siu, Antoine Pitrou and Stanley Seibert, 'Numba: a LLVM-based Python JIT compiler' in LLVM '15: Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, ACM Digital Library, Nov 2015, Article 7, 1-6.

[7]Cheuk, Kin Wai, Hans Anderson, Kat Agres and Dorien Herremans, 'nnAudio: An on-the-Fly GPU Audio to Spectrogram Conversion Toolbox Using 1D Convolutional Neural Networks', *IEEE* Vol. 8, Aug 24, 2020, 161981-162003.

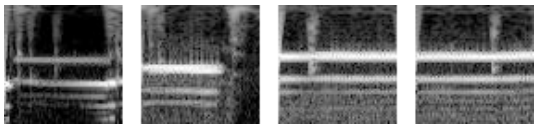

Figure 4.3: CQT spectrograms of timbral saxophone improvisation.

In the second iteration of this work a further data preprocessing step was taken in the form of applying salience modelling to the transformed CQT data, a technique more commonly used in polyphonic music transcription tasks[8]. This was in response to an issue encountered whereby an accurate classifier could be trained and interacted with in one session, but then the same model would not perform as well when I came back to use it in a different session. This I attributed to CQT's inherent sensitivity to variables such as environmental conditions, recording levels, microphone position, my reed, etc. As can be seen in the spectrograms in the figures 4.2 and 4.3 above, a good deal of noise is captured in addition to more pertinent musical information. Using Librosa's 'harmonic salience' function filters out this extraneous information, as can be seen in the salience spectrograms below.

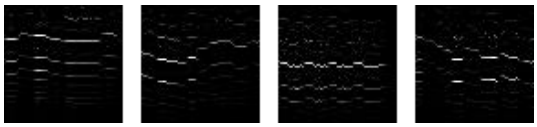

Figure 4.4: salience-modelled CQT spectrograms of melodic saxophone improvisation.

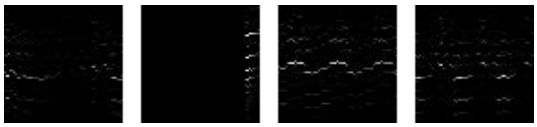

Figure 4.5: salience-modelled CQT spectrograms of timbral saxophone improvisation.

Curiously, the difference between the two categories is, looking at the salience spectrograms above, much less clear to the human eye than in the CQT spectrograms in figures 4.2 and 4.3. Training statistics were also not especially encouraging when I trained a model on this variant of my data. In practice though, it proved possible to train a classifier on salience spectrogram data that was sufficiently robust in practice - besides basic classification accuracy being 'good enough', it was also, crucially, less sensitive to acoustic variations than those trained on 'normal' CQT spectrograms. It proved possible to train a usable classifier with this approach on data recorded three years prior using a different microphone, saxophone and reed-mouthpiece configuration, effectively providing a workable solution to the issue of acoustic sensitivity of spectrograms for on-the-fly audio classification.

4.3.3 Model Architecture

After early experimentation with deep model architectures such as DenseNets[9] and ResNets[10], I opted for a 2-layer convolutional neural network architecture optimised for 64x64 images. This choice was informed by not wishing to resize the spectrograms and by

[8]Bittner, Rachel, Brian McFee, Justin Salamon, Peter Li and Juan P. Bello, 'Deep Salience Representations for F0 Estimation in Polyphonic Music', Proceedings of ISMIR 2017: 18th International Society for Music Information Retrieval Conference, Suzhou, China, Oct 23-27, 2017.

[9]Huang, Gao, Zhuang Liu, Laurens van der Maaten and Kilian Q. Weinberger, 'Densely Connected Convolutional Networks', unpublished paper, arXiv:1608.06993.

[10]He, Kaiming, Xiangyu Zhang, Shaoqing Ren and Jian Sun, 'Deep Residual Learning for Image Recognition', unpublished paper, arXiv:1512.03385.

an approach recommended by Andrej Karpathy[11] of ensuring that the dataset and trainable model weights are of similar size; taken at face value this suggests that roughly 60 minutes of audio data (after resampling, augmentation, and conversion to spectrograms) is an appropriate amount to the model's 4.36MB of trainable parameters, which in practice I think should be considered a bare minimum.

The commonly used Adam optimizer is favoured, the learning rate of which is set to 1e-4; further compensation for the small dataset size is added to the optimizer in the form of an L2 penalty value of 1e-4. While CUDA acceleration can be used to keep training times low, the model architecture is sufficiently shallow to be trainable in a modern CPU.

4.3.4 Inferring the Class of a Live Input Segment

When inferring over a live input, an audio stream is opened from which a 32768-long window is preserved. This audio segment is first analysed for loudness. If the loudness exceeds a given threshold then the program effectively determines that the input corresponds to musical action on the part of the improviser proceeds to the next stage; otherwise, no further consideration is given to the current input segment, the program loop reverts to the beginning and a new audio segment is recorded.

If the next stage is reached, the segment is converted to CQT spectrogram using the same parameters as those the model was trained on, and the model infers which solo the segment is most similar to. To mitigate false classifications and improve robustness, only the second consecutive classification of the same category is deemed to be definitive and allows progression to sample playback, otherwise the loop reverts as if no input was received or as if no class was confidently inferred. This step is necessary since there is no guarantee that the recorded window leading to the first classification contains enough relevant musical information to confidently predict the mode of playing; using the second classification improves confidence in the classifier's output.

Figure 4.6 describes the inference process in the context of the live interactive loop per the first iteration of SoloSoloDuo ('SoloSoloDuo_v1.wav' and 'SoloSoloDuo_v1.mp4').

Additional Filtering and Differentiation

In the first version of 'SoloSoloDuo' interactivity is almost entirely driven by the classification output alone, as described in figure 4.6. While there is a pre-classification filter in the form of a loudness threshold to ensure that a process would not be triggered by an environmental sound, sample selection is only governed by the classifier's output and the system's response was only randomly selected from the samples within the inferred class's corresponding sample folder.

The lack of any additional differentiation of samples meant the system was, in this first iteration, unpleasantly jarring to play with, to a degree that made putting this work into practice not especially enjoyable. While I could see some musical interest in the interruptive modality generated by this environment, I wanted to make the system feel more natural to play with and the generated outputs more musically coherent. To this end I added analyses of both the sample sets and live input.

At the stage of sample set creation and model training, two lookup tables are created into which key-value pairs of filename:analysis are entered. For the melodic category, frequency analysis is performed on each sample; for a single sample, the resulting frequency array is converted to MIDI note format for linearity and rounded to one decimal place

[11] Karpathy, Andrej, 'The Unreasonable Effectiveness of Recurrent Neural Networks'.

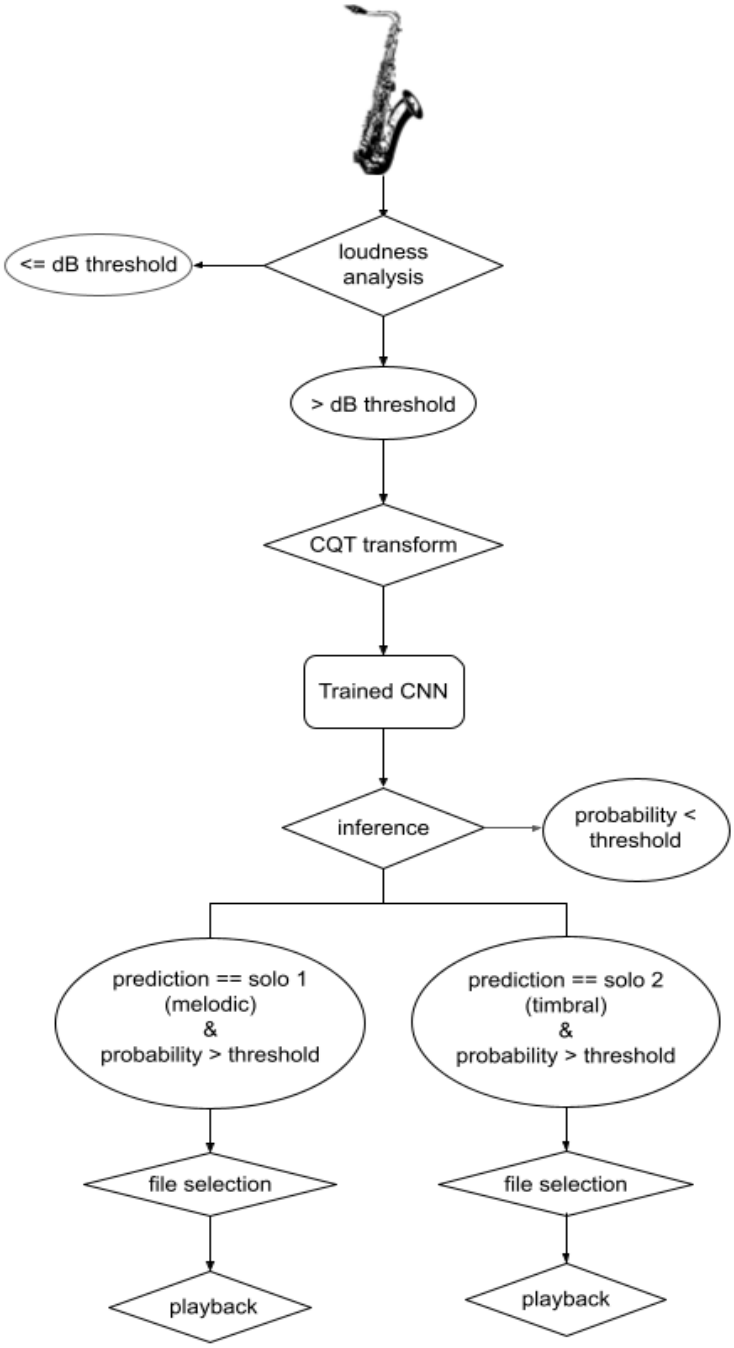

Figure 4.6: Flow diagram showing the inference process in the first iteration of SoloSolo-Duo.

to accomodate the quarter tones I preferred. Discrete differences along the length of this array are calculated, resulting in an array of values for the change between one pitch value and the next. This array is then parsed pair-wise according to the rule set below.

For a given pitch:

- If the next pitch is greater or less than the previous by 0.5 or less (equivalent to a quarter tone), then 0 (ie no onset detected);
- If the next pitch is greater or less than the previous by more than 0.5, then 1 (onset detected);
- If the current pitch is 0 (no pitch found, ie silence) and the next pitch is 0, then 0 (no onset detected);
- If the current pitch 0 and next pitch is 44 (bottom b-flat on tenor saxophone) or greater, then 1 (onset detected)

The underlying assumption here is that a substantial change of pitch (i.e. close to a semitone) in either direction represents an onset. This could equally be described as a note of melodic-rhythmic significance or a 'note event'.

The binary array this effective pitch-based onset detector outputs is then padded with a single zero at each end to make it the same length as the original pitch array. It is then multiplied element-wise by the pitch array. This processing stage outputs an array of zeros (no musically-significant pitch detected) and MIDI note values which are effectively the melodically salient note events. [12]

Here it is used to extract the first significant pitch, last significant pitch and number of significant pitches in a given phrase sample. This process is repeated over the entire melodic sample set and the resulting data is saved in a lookup table.

During interaction, the same process is applied on-the-fly to the recorded live input segment. If the classifier deems the input to be 'melodic', the program then performs this same process of pitch-based onset detection to the live input segment. It then tries to find a close match between the last pitch of my live input and the first pitch of a sample in the 'melodic' category's sample set and selects it for playback.

While this process was effective in creating a more natural-feeling interaction when playing melodically, finding an equivalent method for mediating sample selection from the 'timbral' category proved more challenging. The seemingly obvious choice of Mel Frequency Cepstral Coefficients (MFCCs), which return information about the rate of change in a signal's frequency and are widely considered a useful measure of timbre, turned out to be a mismatch with my preferred window size of 32768 samples or approximately 1.4 seconds, since a core assumption of using MFCCs for timbre analysis is that the content within a given window consists of a single sound.

In the end I opted for simple amplitude analysis over more sophisticated feature extraction. While amplitude alone does not typically return much useful information about a signal's perceptual content, here it is appropriate given the somewhat fixed nature of multiphonics on the saxophone; given the necessary manipulation of air stream and embouchure required to generate many of my preferred multiphonic timbres in the first instance, there is then very little room for further manipulation of the sound and so the majority of these

[12] This method of extracting musically salient information using only frequency analysis has also proved useful for dataset creation in the symbolic domain and will be revisited in Chapter 6.

sounds tend to occupy their own somewhat narrow and inflexible amplitude range.

To extract the most salient amplitudes of each sample I opted for isolating its loudest window given a window size of 160 milliseconds (the Praat-Parselmouth analyzer[13] [14] I used measures analysis step size in time, not samples). Similarly to the steps outlined above, this value is stored in a lookup table; during interaction the same analysis is performed on the live input and used to find the closest match. While not quite as effective as the equivalent process for finding more musically appropriate playback samples for the melodic category, this process mitigated the jarring effects of only relying on the classifier's inferred class output for sample selection in the timbral category and made the system more enjoyable to play with.

4.4 Musical Applications

4.4.1 'SoloSoloDuo'

'SoloSoloDuo' is an interactive structured improvisation that consolidates the processes described in this chapter. Each section is improvised according to its place in the schema which can be seen in its corresponding score. The first improvised 'Solo' section is in a style consistent with the 'Melodic Improvisation' dataset while the second is in the manner of the 'Timbral Improvisation' dataset. The third 'Duo' section is an interactive duet with samples taken from each 'Solo' section, with the interactions mediated by the classifier and, in version 2, additional frequency analysis and pitch-based onset detection of both the live input and samples.

The goal of this piece was, initially, to create a virtual improvising partner with whom I could at least practice or ideally make music with. Later in the process, once the interactive loop needed for the duo section was fully functioning, my focus shifted towards realising the piece and the goal became more about creating a musical representation of processes behind deep learning classification. As such, the influence of the discriminative model on this musical output goes beyond its role as the central component of the interactive software created to realise it. The process of pre-assigning data classes, creating the training data, training the model and using it to classify unseen inputs (commonly referred to in machine learning-speak as 'inference') defines the structure of the piece itself, in which the two solo sections can be seen to represent dataset creation and the duo section representing the inference process.

Version 1

Version 1 of 'SoloSoloDuo' was premiered at the AIMC 2022 conference, organised by researchers based at the Riken Institute in Japan but hosted online owing to the uncertainties around Covid regulations at the time of planning. In order to present this work in a format suitable for online presentation I created an audiovisual version.

The visual aspect was achieved with a combination of technologies. Fistly I created an amended version of Derrick Schultz's 'Rocks' dataset, re-rendering the images in grayscale, inverting them and enhancing the black background where necessary; I did this so that images could be faded to black to correspond with the ends of musical phrases. I then trained a StyleGAN2[15] model on this dataset using transfer learning from a pretrained model of

[13] Boersma, Paul and David Weenink, *Praat*, version 6.3.09, computer software, https://www.fon.hum.uva.nl/praat/

[14] Jadoul, Jannick, Bill Thompson and Jan de Boer, 'Introducing Parselmouth: A Python Interface to Praat', *Journal of Phonetics* 71 (2018): 1-15.

[15] Karras, Tero and Janne Jellsten, 'Training Generative Adversarial Networks with Limited Data'

the ffhq-1024 dataset. I then used this model in combination with Mikael Alafriz's 'Lucid Sonic Dreams' program[16] and the recorded audio to create the raw audiovisual output before editing the videos so that the rocks only appear during musical events.

When recording version 1, I felt that the interactions, despite the mediating influence of the classifier, were jarring and frequently too tangential to the input. In order to remedy this I added the processes described in the section 'Additional Filtering and Differentiation' to the interactive loop script and re-recorded 'SoloSoloDuo', the result of which is described in the section on version 2.

Version 2

Version 2 of 'SoloSoloDuo' follows the same schema as version 1 and in fact begins with a very similar phrase. This use of a stable starting point for improvisation hints at the use of composition as a jumping-off point for improvisation that will be explored in Chapter 5's outputs 'Lrrning', 'Gander' and 'The Lows'.

Of the additional filtering incorporated to the interactive loop in this version, I found the pitch-based onset detection used to mediate melodic interactions to be successful. Finding an output response sample with a starting pitch as close as could be found to the end pitch of the last input segment made the interactions feel significantly more natural and made for a more enjoyable playing experience. The amplitude-based filtering used to mediate playing in the timbral category was less succesful however, at first resulting in only a narrow range of the sample outputs ever being used. To mitigate this I expanded the threshold for finding an amplitude-based match to a degree where it was questionable that the filtering was really playing that much of a role in the interactions. Overall though, both melodic and timbral interactions felt significantly more natural than in version 1.

The time period between recording the two versions of 'SoloSoloDuo' allowed for some emotional distance from the feeling of my improvisational efforts being thwarted by the insensitive computer in version 1. This perspective led me to reflect that what had been perceived as overly jarring could be reframed as being in a permanent mode of 'interrupt-type' ideas generation. Looking at the interactions in this way helped me to compose myself during the interactive phase of version 2 and to embrace this interruptive modality as part of the piece's aesthetic. Combined with the more natural-feeling interactions effected by the additional filtering, this resulted in both a more enjoyable playing experience and better musical outcome in my view.

Further Reflection

When considering how the process of creatively engaging with deep learning classification and machine listening has influenced both the compositional structure of 'SoloSoloDuo' and its aesthetic, I find resonance with Mark Fell's assertion that 'for [him], aesthetic positions are just a side effect of using technologies - of exploring materials, processes, tools and their interactions'[17]. The aesthetic of 'SoloSoloDuo' is effectively the result of an exploration of how these technologies intersect with fundamental aspects of my instrumental practice. How the use of a classifier with a large analysis window, combined with phrase-level chunking for playback, creates an angular, interruptive aesthetic and, at points, the effect of having been multi-tracked; how the representation of the data classes in the solo sections give the piece its structure and define, to varying degrees, the content of each section; how my emotional responses to the interruptive modality of the 'Duo' section create

[16]Alafriz, Mikael, 'Introducing 'Lucid Sonic Dreams': Sync GAN Art to Music with a Few Lines of Python Code!'.

[17]Fell, Mark, *Structure and Synthesis: The Anatomy of Practice* (Urbanomic, 2021), 20

space and contrast: these are interactions between the familiar materials of an existing practice and unfamiliar processes and tools.

Andrea Parkins' description of her own journey from playing acoustic piano to working with custom-made software resonates with this last point about emotional responses to the technology shaping the music's character: she describes noticing 'the effort a (my) body must make in order to play my instruments, and how .. an immediate electronic gesture .. can thwart this effort .. creating new (unforeseen) ways to make sound and to listen.'[18] Given my own past as a predominantly acoustic instrumentalist, barring occasional performances with electronic musicians such as Federico Reuben, and that 'SoloSoloDuo' represents one of my first self-directed forays into the use of digital technology and custom-made software, this description of the the jarring nature of early encounters with such technologies and of the new artistic territory they can eventually open up feels especially apt.

Fell's positioning of aesthetic outcomes as being a side effect of exploring technologies, processes and materials and Parkins' experiences of opening up unfamiliar modes of music-making through (not always easy) encounters with new technologies are, for me, intertwined. While I am in agreement with Fell's position on aesthetic outcomes, his work tends to centre consistently around certain processes (granular synthesis) and tools (MaxMSP and associated external plugins) in which he is well-versed, whereas this thesis is in part an exploration of the interaction of familiar materials with less-familiar processes and tools. Parkins' more open, experimental approach, taking in a more diverse range of technologies and resulting in a diversity of uncomfortable-yet-creative encounters, has more in common with my approach on the processes and tooling side. These ideas are pertinent to the majority of work in the portfolio in the sense that their aesthetic positions are the results of exploring the interaction of deeply familiar materials and unfamiliar technologies, through a series of not-always-comfortable creative encounters.

4.5 Conclusion

By now it should be apparent that this initial interaction with the world of machine learning and making digital tools for music was a fairly uncomfortable music-making experience, and the chapters that follow can be seen to represent a journey of seeking out more rewarding encounters with deep learning technologies. That said, there is reward in reflecting on the artistic and technical insights generated through making of this piece.

Artistically speaking, this encounter with deep learning classification for musical interaction, while uncomfortable for reasons already explored, represents what to me is an important first step into an experimental music-making modality in which the aesthetic outcome is essentially the result of sometimes difficult interactions between my existing musical materials and unfamiliar technologies and processes. While it might seem odd for an improviser to speak of 'experimental music-making' as a new process it occurs to me that the acoustic group improvised music with to which I was already habituated is in some ways less experimental than the approach taken in this chapter. Generally speaking, in group improvised music the aesthetic positions of the players are well known in advance, and while the course that the music will take is unknown, an amount of what the performance will sound like can often be inferred in advance, especially when participants have shared histories. Interacting with a new technology forced me to adjust my expectation of what musical interactions feel like, and the interruptive modality the majority of these interactions inhabited is I think, on reflection, one of the more interesting aspects of the piece. A less fraught way in which the technology influenced the piece is in the classification

[18] Parkins, Andrea, 'Nothing To Be Scared Of' in *Grounds For Possible Music: On Gender, Voice, Language and Identity* ed. Julia Eckhardt (Errant Bodies, 2018), 132.

process's suggestion of compositional structure; in this sense, the piece is less experimental than an improvisation in which the overarching structure of a piece is, generally speaking, not known in advance.

On the technical side, a clear application of neural audio classification to interactive improvised practice is the reduction of need to hand-craft a machine listening process for cases where a solution would be prohibitively difficult or where the user does not feel equipped or motivated to engage in such work. Such tools can form the basis of new interfaces for human-computer interactions and the development of compositional structures, an example of which has been presented in this chapter.

For interactive application, it is of course necessary to train the classifier on the kinds of material it will be expected to differentiate at the point of usage. While this implies to a degree that one can improvise naturally when recording classification datasets, there is a need to provide clarity that in practice implies constraining oneself in order to stay within the bounds of what is being classified. Some improvisers might object to this curtailing of their natural playing instincts and to placing aspects of their practice within rigid categories, but for me it has proved a worthwhile and instructive exercise.

The process of building and interacting with the tools used in 'SoloSoloDuo', where the use of the trained classifier implies the ability to switch freely between categories and have the computer respond in kind, encouraged me to do exactly that, suggesting fresh recombinations of familiar improvisational material. This raises the idea of deep learning models of audio as teachers.

As can be seen in the programs developed in this chapter, a trained classifier can form the basis for and backbone of an interactive tool, but in my case and in likelihood others, it is insufficient by itself in practicality. Both before and following the classification process it proved necessary to add machine listening functionality such as amplitude thresholding to filter out silences and frequency and amplitude analysis for further differentiation of each category. Programmatic techniques such as the counter used to mitigate the unwanted effect of false categorisations were also found to be necessary. However what the trained model does do that these filtering steps do not is be suggestive of musical structures that in its absence would not have occurred to me. This again raises the idea of using an audio model as a creative assistant: as a generator of ideas and conceptsreplaced.; I will return to this theme in the next chapters.

Chapter 5

Generative Modelling of Raw Audio in Practice: Interactive Duos, 'Workshop' pieces, Compositions for Solo Improvisation

5.1 Introduction

This chapter presents an inquiry into creative musical applications of two deep learning architectures for unconditional raw audio generation, SampleRNN and WaveGAN. I train SampleRNN and WaveGAN models of datasets presented in Chapter 3, exploring training parameters and arriving at assumptions about training configurations in the process. Model architecture behaviours are discussed in light of their generated samples and how these relate to the ground truth data; lessons from modelling some external datasets are also outlined. Outputs from each model architecture are first applied in interactive contexts that draw upon the work in Chapter 4. They are then used as bases for electroacoustic compositions 'Workshop I' and 'Workshop II' which illustrate the process of using curated samples as source material for practice exercises. Samples from WaveGAN and SampleRNN models of external datasets are layered and looped in 'Gandering 1', an audiovisual composition. Finally, generated samples are transcribed and arranged into compositions that act as springboards for solo improvisation. The work in this chapter generates knowledge of good practices for creating datasets for generative modelling, of these model architectures' behaviours and of their applications to a range of experimental music practices; these insights have the potential to help make dataset creation, model training and useful creative work with their outputs more reasonably practical endeavours. It also generates knowledge of the potential applications of this category of deep learning architecture to instrumental practice, improvisation and composition.

5.2 Rationale

Generative deep learning can be characterised as the process of seeking a generalised probability distribution over a given dataset[1] and prompting the resulting statistical model to generate new material; in the best case, this new material is not simply a regurgitation of the input dataset but possesses some novelty of content and/or character. Similarly, Yin et al characterise this distinction as the difference between a generative model's outputs 'imitating' and 'stealing' from the dataset, which are cast in a positive and negative light respectively. In their formulation, the desired 'imitative' result implies *sounding like* but

[1] Sander Dieleman, 'Generating Music in the Waveform Domain', https://sander.ai/2020/03/24/audio-generation.html.

not *copying from* the dataset[2].

To a practitioner of improvisation, the process of generative deep learning is attractive in that it offers the potential to generate outputs which, while based on recordings of instrumental practice, are a product of those inputs that I would not necessarily think to play myself. As noted in Chapter 1, an early motivator for this project was the idea of building computational models of the process of practising and the impact practising has on improvised outputs. The possibility of modelling this process to generate new musical ideas was a strong attractor to generative machine learning models: I consider it a great strength and appeal of the generative modelling process to be the opportunity to explore a dataset and by implication the musical space the data inhabits. I will consider this idea further as inputs and outputs are discussed.

Statistical models of raw audio also offer the musician the enticing possibility of greater 'expressive' potential in computer generated samples, since the raw signal contains all of the complexity and nuance of instrumental timbre and the instrument's 'action space'[3]. This is especially important when attempting synthesis of the sound of an instrument as complex as the tenor saxophone, whose lack of uniformity of sound production and complexity of transients are not well captured by conventional synthesis methods.

5.2.1 Early Experiments

In my earliest experiments with WaveGAN, its capacity to generate fake saxophone samples that, while noise-laden, are both musically plausible and, crucially, novel, was evident; through the noise I could hear the content of my practice being recombined in ways that sounded fresh, exciting and unfamiliar to me. The simple fact that it was generating surprising saxophone lines that were of course based on a probability model of my own recorded inputs was plenty justification for pursuing working with this model architecture further.

Another aspect of WaveGAN that struck me early on was its speed of sample generation. While training a model takes several hours even when using relatively efficient parameters and downsampled data, the process of passing a noise vector through the trained generator weights resulting in a usable audio vector is trivially fast - measurable in milliseconds rather than seconds, assuming that the trained generator has already been loaded into RAM. This is obviously attractive from a simple efficiency and convenience perspective, but it also meant that samples could be generated on-the-fly inside an interactive loop. This idea of having a trained model that could generate content very quickly on demand further underlined this model architecture's potential as a creative tool.

By contrast, early experiments with training SampleRNN models on my data revealed almost an opposite set of attributes. This model architecture's ability to render clean, realistic-sounding waveforms, particularly in the lower register of the tenor saxophone, was striking; the presence of breath and key-click sounds in the rendered waveforms contribute a lot to this realism. Initial outputs also had a naive, undemonstrative character that I found pleasant to listen to and a welcome contrast with WaveGAN's noisy, more aggressive-sounding outputs. However, these initial SampleRNN outputs were poor in terms of modelling musical content: content-rich melodies were reduced to vague, scribbly gestures and the more subtle 2- and 3-note multiphonics were rendered as single notes, a

[2]Yin, Zongyu, Federico Reuben, Susan Stepney and Tom Collins, 'Measuring When a Music Generation Algorithm Copies Too Much: The Originality Report, Cardinality Score, and Symbolic Fingerprinting by Geometric Hashing' in *SN Computer Science* 3:340, SpringerLink, Jun 2022.

[3]Sander Dieleman, 'Generating Music in the Waveform Domain', https://sander.ai/2020/03/24/audio-generation.html.

consequence of SampleRNN's small context window.

On balance though, there was more than sufficient interest in the generated outputs from models trained using each of these architectures to justify further investigation into their possibilities.

5.2.2 Practical and Environmental Issues with Modelling Raw Audio

Generative models of raw audio are energy-hungry[4]. Since the raw audio waveform, consisting of tens of thousands of amplitudes per second, results in very large datasets by default, successfully building computational models of these datasets requires model architectures with a correspondingly high number of trainable weights. For the practitioner wishing to explore these tools, this translates to needing access to high-performance GPUs and being able to run them for several hours at a time. An additional complicating factor was a need, in the early stages of engaging with these large model architectures, for fast experimentation, a need incompatible with the use of HPC cluster infrastructure. Training generative machine learning models of raw audio therefore raised practical issues of access to the required hardware and electricity and some additional discomfort around the replacedenvironmentalethical issues concerning their energy consumption.

To address the need for fast experimentation, I took the step of building a Linux machine with a 24GB Nvidia RTX Titan GPU. Once I felt I had a handle on the process of training these models, I then decided to use Google's Colab service[5]. Using this service in conjunction with a Google Drive account also provided a convenient storage solution and allowed me to train multiple models simultaneously, speeding up the process of experimenting with parameters and finding a good model. My move to Colab was also partly motivated by recent increases in energy costs. Using GPUs provided by Google's cloud services at least partly addresses the environmental issue by using pre-existing hardware infrastructure located in Google's data centres; these are subject to environmental regulations and, being industry servers, are designed with efficiency in mind. According to the online machine learning emissions calculator tool presented by Lacoste et al[6], they are also apparently carbon-offset. By way of addressing the practical issue of time and further mitigating the emissions issue, I resolved to find training parameters that, as far as was reasonable without compromising audio output quality, minimised training times.

5.2.3 Related Model Architectures

Developments in the field of deep learning research for audio, as with most fields in which it is being applied, are at the time of writing fast-moving. When I began this work, WaveGAN and SampleRNN were, along with WaveNet[7] (a functioning implementation of which I struggled to find during this period of my research), the clearest available options for training generative models of raw audio.

Since that time other model architectures have emerged. The most visible and talked-

[4]Douwes, Constance, Philippe Esling and Jean-Pierre Briot, 'Energy Consumption of Deep Generative Audio Models', proceedings of ICASSP 2022: EEE International Conference on Acoustics, Speech and Signal Processing, Sand Expo and Convention Centre, Singapore, 22027 May 2022.

[5]Unknown author, webpage, https://colab.google

[6]Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, Thomas Dandres, 'Quantifying the Carbon Emissions of Machine Learning', unpublished paper, arXiv:1910.09700.

[7]van den Oord, Aaron, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior and Koray Kavukcuoglu, 'WaveNet: A Generative Model for Raw Audio', *Deepmind*, blog post, September 8, 2016 https://www.deepmind.com/blog/wavenet-a-generative-model-for-raw-audio.

about of these is RAVE from IRCAM's ACIDs team[8]. Despite a strong urge to engage with RAVE, finding sufficient research time to experiment with and create music using this model proved tricky. This is in part because RAVE is significantly more data-hungry than WaveGAN or SampleRNN, requiring datasets of several hours in length and careful curation of data content. Training times can also vary between 3 or 4 to 7 or 8 days depending on the size of the dataset, whereas it is possible to train WaveGAN and SampleRNN models in a few hours.

These practical considerations have led me to consider that while, based on its apparent capabilities and publicly available results with the author-created 'Darbouka' dataset, RAVE is a highly compelling model architecture that deserves investigation of its possibilities by musicians and practitioner-researchers, the practical obstacles to creating suitable datasets and training RAVE models on them make it almost deserving of a dedicated research project.

Another very different option for modelling raw audio signals is Catch-A-Waveform[9], an adaptation of the SinGAN[10] image-generating model architecture which generates variations on a single image input. While the convenience of this model architecture's low data length requirements - it can be trained on a single audio file lasting a few seconds - was initially attractive, I found in early experiments that despite its long training times it only generated minimal variations on the input material, with those variations insufficiently interesting to my ears to be creatively useful. Early experiments with longer inputs resulted in GPU out-of-memory errors. However in more recent experiments, aided by the advent and availability of Nvidia's 40GB A100 GPU, I have been able to train models of longer inputs with a longer context window that generate more interesting variations. These recent outputs also have better audio fidelity than WaveGAN, despite their architectural similarities. A fuller exploration of Catch-A-Waveform's potential as a creative tool is outside of the scope of this thesis and is a strong candidate for future work.

5.3 Technical Processes

5.3.1 Dataset Pre-Processing

A principal challenge of training large generative models is acquiring or creating coherent datasets of sufficient size. To this end, I applied processes of data augmentation to each dataset. In earlier experiments I pitch-shifted each dataset by a semitone in either direction, the rationale being that doing so increases data size while retaining musical plausibility, and without introducing digital artefacts as would be the result of more drastic pitch shifting by a wider interval. Doing so increases the quantity of input data by a factor of 3, albeit artificially; in the case of a very short initial recording, this could be pushed further by applying an additional semitone in either direction to increase dataset size by a factor of 5. In later experiments I found that when training models with SampleRNN, time stretch augmentation yielded much better outcomes than pitch shifting, a point I will return to in the 'Behaviour' section later in this chapter.

Additional augmentation can be applied by inverting the audio waveform's polarity. The rationale here is that although polarity-inverted audio is perceptually indistinguishable from its inverse, the resulting sequences are novel in terms of their data content; this

[8] Anton Caillon and Philippe Esling, 'RAVE: A variational autoencoder for fast and high-quality neural audio synthesis'.

[9] Gal Greshler, Tamar Rott Shaham and Tomer Michaeli, 'Catch-A-Waveform: Learning to Generate Audio from a Single Short Example'.

[10] Rott Shaham, Tamar, Tali Dekel, Tomer Michaeli, 'SinGAN: Learning a Generative Model from a Single Natural Image', unpublished paper, arXiv:1905.01164

approach effectively increases the dataset size by a further factor of two. As an example, applying this process to 30 minutes' worth of recordings, after two pitch-shift or time-stretch augmentations, results in 3 hours of input data.

The rationales for these augmentations are twofold. Firstly, enlarging my datasets in this fashion accords with my own observations of achieving better training metrics and results when erring on the side of more data and greater cohesion of input data; it also follows a general good-practice principle established by Andrej Karpathy of ensuring that the dataset and the trainable model weights are of a roughly similar order of magnitude[11] (for example, the combined trainable weights of WaveGAN generator and discriminator networks with a dimensionality multipler of 32 total roughly 150MB). Secondly, ensuring that each dataset has a distinct character – as opposed to simply combining less closely related datasets – retains coherence, allows for better judgement of outcomes, and at least somewhat simplifies the learning task. A more diverse dataset runs a greater risk of poor generalisation and poor outcomes.

Once the data has been augmented, silences exceeding a specified length and loudness threshold are truncated to a specified shorter length to remove unnecessary periods of silence. These thresholds are chosen carefully to retain coherence of musical phrases.

The data files are segmented according to each model architecture's implementation. In the case of WaveGAN, its reference implementation offers flexibility in this regards thanks to its streaming dataloader, the only constraint being to ensure that the length of each chunk exceeds the 'data_slice_len' parameter [12], which defines the length of the learned window and generated samples. In theory this means one could even just pass the entire dataset as a single file, but I eventually found it preferable to segment the input audio data according to musical phrases where possible using the same process presented in 'SoloSolo-Duo' in Chapter 4, where the input solos were chunked into musically coherent phrases by means of frequency analysis. Doing so helped to mitigate a tendency for the input segments to begin very abruptly, starting mid-phrase or even mid-note; when this was the case, generated samples sounded less musically plausible and chaotic. SampleRNN data segmentation is more straightforward, requiring the input data to be in 8 second chunks without necessity to chunk the data according to onsets or phrases.

5.3.2 Training WaveGAN

As mentioned in the subsection 'Practical and Ethical Issues with Modelling Raw Audio', I resolved to find parameters that kept training times reasonable without negatively affecting sample quality. To this end, with WaveGAN I favoured setting the model to learn and therefore generate chunks of audio with length 32768 data points, with the input data downsampled to 22050 Hertz, resulting in generated audio segments of roughly 1.47 seconds in length. I ensured that the dimensionality multiplier parameter was reduced from its default value of 64 to 32, resulting in a smaller number of trainable parameters that was more appropriate for my typical dataset sizes; this also had the effect of speeding up training times. A very significant training speedup specific to the author's implementation was obtained by converting the audio dataset to a bit depth of 16 and including the 'data_fast_wav' flag in the training parameters. These parameters combined meant that a model could be trained for around 100000 iterations, often sufficient to obtain reasonable results, in the space of a day. Earlier experiments before this discovery had taken

[11] Karpathy, Andrej, 'The Unreasonable Effectiveness of Recurrent Neural Networks', *Andrej Karpathy Blog*, May 21 2015, http://karpathy.github.io/2015/05/21/rnn-effectiveness/

[12] Donahue, Chris, 'wavegan', online code repository, posted by 'chrisdonahue', Apr 2018, https://github.com/chrisdonahue/wavegan/blob/master/train_wavegan.py/

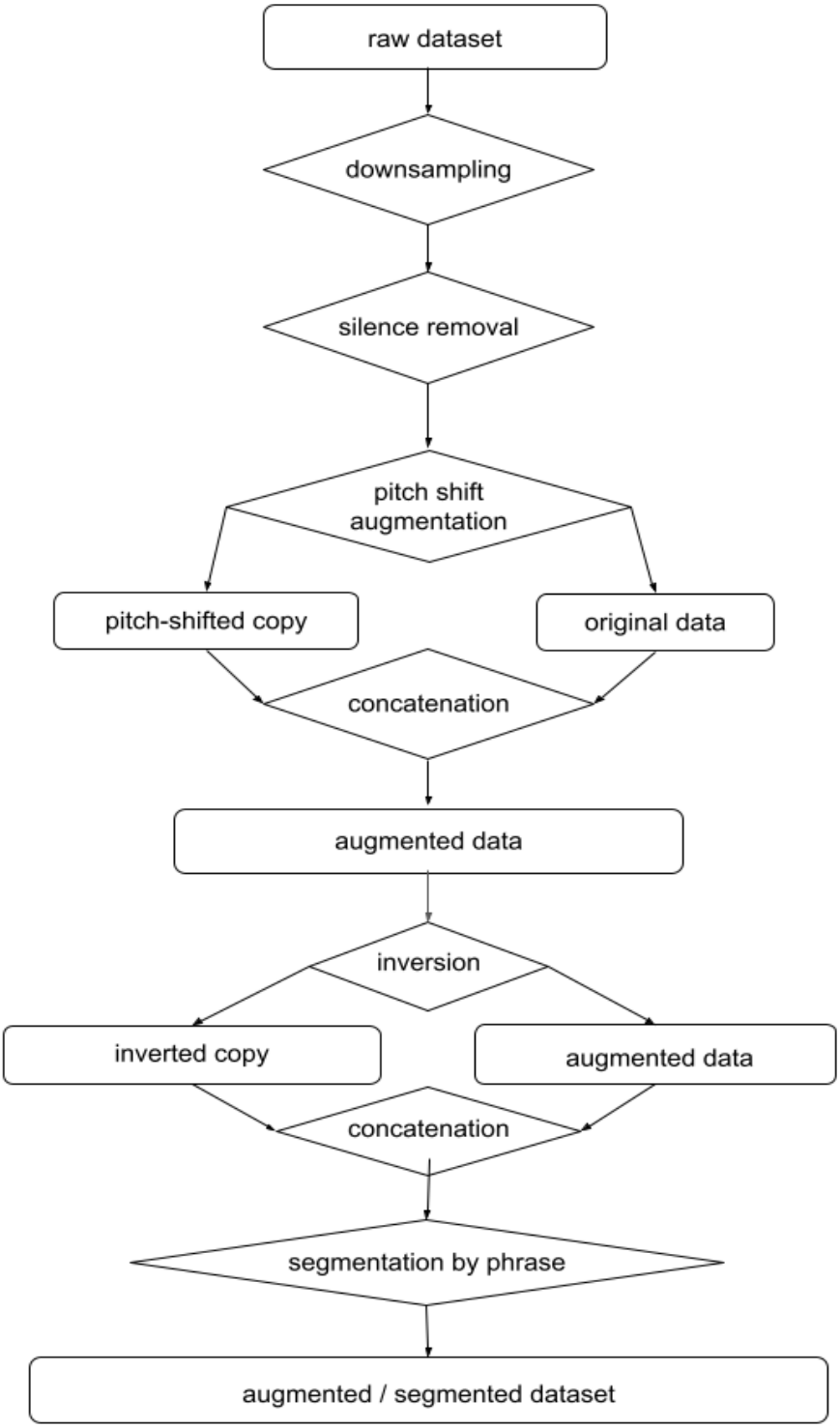

Figure 5.1: Flow diagram of dataset preprocessing pipeline for WaveGAN.

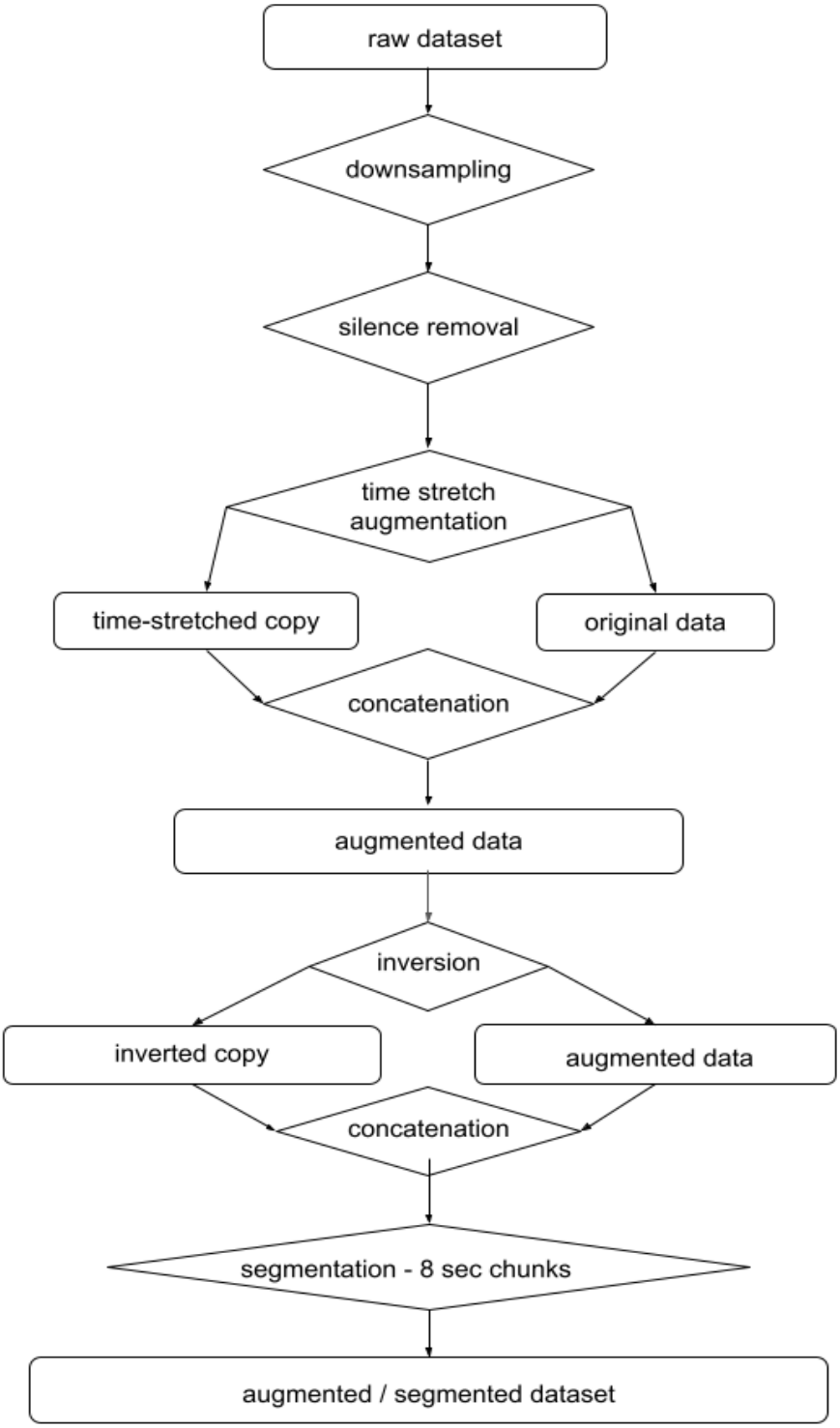

Figure 5.2: Flow diagram of dataset preprocessing pipeline for SampleRNN.

significantly longer.

Despite repeated experiments with numerous other changes to the model architecture and training parameters, the most tangible positive effect on quality of generated outputs was achieved through a combination of maximising dataset size within the boundaries of similar orders of magnitude, and ensuring that the same instrumental setup, recording environment and equipment was used to create the dataset. This need for consistency of recorded sound could not be overstated – bundling together separately-made recordings with even slightly diverse acoustic characteristics into one dataset negatively affected the consistency of generated samples. However, through data augmentation, a recording of a single session of a reasonable length such as 30 minutes can be used as a training set, with the need for uniformity guaranteed by the homogenous recording environment. Ensuring an appropriately large dataset through the augmentation techniques outlined in the 'Dataset Pre-Processing' section proved important for avoiding over-fitting, in which event the trained model simply generates material largely indistinguishable from the input data.

One adjustment made to train a WaveGAN on the more abstract musical content found in the timbral improvisation dataset was to use a 'leaky ReLU' activation function on the generator network as well as the discriminator. Doing so helped to avoid mode collapse when training on the timbral dataset. This approach was suggested by founding PyTorch engineer Soumith Chintala at his 2016 NIPS Workshop on Adversarial Training[13], proposing that 'soft' gradients that allow negative values can be a favourable choice with generative models, as opposed to the hard ReLU activations commonly found in classification models. This implementation change was only necessary because of the difficulty of training on this specific dataset and was not necessary for the other melody-based datasets. I suggest that this is because the relative lack of timbral diversity in the other datasets equates to a simpler learning problem. That said, results from the timbral dataset did not inspire me to create music with them, for reasons I will elaborate on in the 'Behaviours' section later in this chapter.

5.3.3 Training SampleRNN

In the majority of my experiments with SampleRNN I followed the above-stated principle of finding a balance between output fidelity and training times by downsampling my datasets to 22050Hz. This had the desired effect of keeping training times reasonable (typically less than a day although sometimes slightly longer with larger datasets). At a later stage of this project however I noticed the generated samples had a 'muted' quality and speculated that downsampling might be the cause. When I trained a model of the augmented Tone Rows dataset at 44100Hz this suspicion was confirmed - modelling the timbre of the tenor saxophone with SampleRNN does benefit from a higher sample rate. This increase in sample quality was inevitably and unfortunately associated with an increase in training times.

There are adjustments to SampleRNN model architectures that I observed to have a tangible effect on outcomes. These are in the type of RNN cell used (GRU or LSTM) and the inclusion or not of 'skip connections'. Adding skip connections to the architecture reliably resulting in an extension of the number of training epochs before auto-stop kicked in (triggered by no significant improvement in the loss for 3 epochs) but did not add any tangible improvement to the generated samples, while the use of GRU cells tended to result in a less attractive saxophone timbre reminiscent of excessive 'proximity effect', a common result of positioning a condenser microphone too close to the sound source[14]. The most

[13]Chintala, Soumith, 'NIPS 2016 Workshop on Adversarial Training, YouTube video, Uploaded by 'David Lopez-Paz', Feb 17 2017, https://www.youtube.com/watch?v=X1mUN6dD8uE

[14]Unknown author, webpage, posted May 2022, https://www.dpamicrophones.com/mic-

successful SampleRNN models of my datasets were all trained using LSTM-type RNN cells and without the use of skip connections. As a result, this is my go-to configuration for a first training run and I only try other three possible combinations of these parameters if results are unsatisfactory or if there is a sense that they could be improved upon with a different configuration.

The strongeest predictor of SampleRNN outcomes on my data has been the dataset itself. The principle considerations here are to ensure that the size of the dataset is at bare minimum equivalent to the size of trainable model weights and that the content of the dataset and augmentation technique used is appropriate to the model architecture's behaviour. For SampleRNN models this means use of time stretch augmentation (as opposed to pitch shifting), ensuring clarity and consistency of recorded signal, minimising diversity of timbral and musical content. And, if it feels justifiable, maintaining the recorded sample rate of 44100Hz.

5.3.4 Generation of Audio

Generating audio from the trained model differs by architecture in process and, importantly when considering potential applications, speed. The process of generating audio with WaveGAN essentially involves passing a gaussian noise vector forwards through the weight space of the trained generator model; the discriminator part of the model is not directly involved in the generation process, its purpose being to learn the dataset in order to help train the generator. As a function of its convolutional architecture, which learns and therefore generates a fixed time window (of 32768 samples in my experiments), all of the individual audio samples generated with WaveGAN are effectively calculated simultaneously, resulting in remarkably high speed of audio sample generation. Assuming the generator weights have already been loaded into memory, which does take some time, multiple audio files can be generated in parallel in a fraction of a second in the GPU; generation in the CPU is slower but still significantly faster than real time. There is a speed-flexibility trade-off here though: while extremely fast, WaveGAN can only generate fixed-length arrays the length of which was already determined at training time. Audio generation with SampleRNN occurs in the time domain, a consequence of its time-based architecture. Not surprisingly, this results in considerably longer generation times with SampleRNN than with WaveGAN. However, unlike WaveGAN, which can only output audio files of a fixed window length determined at training time, SampleRNN is more flexible, being capable of generating audio files of arbitrary length.

5.4 Behaviours

In this section I will describe what I observed to be behaviours of each model architecture in terms of the relationships between generated samples and dataset and of the perceived 'character' of their generated samples. While the focus is predominantly on models trained on my own datasets, I also present some additional observations gleaned from training models on externally-sourced datasets.

A general starting point for thinking about how each model architecture behaves is to know that WaveGAN's strength is in it ability to model complex musical *content* while its weakness, beyond the obvious noisiness of the generated samples, is in its tendency to converge on some aspects of the dataset to the detriment of others. SampleRNN, on the other hand, prioritises realism of timbre while, on my data at least, requiring repetitions of content in the dataset - either through recording the same content multiple times or data

university/source-dependent-proximity-effect-in-microphones

augmentations, or both - to help it to model musical content more recognisably.

WaveGAN's prioritisation of content is consequential of its adversarial architecture and its according 'mode-seeking' behaviour: its goal is to generate plausible fixed-length *windows* of *musical content*. The generator optimizes towards fooling the discriminator into mistaking its generated samples for the ground truth data; it can more easily achieve this by converging on specific aspects of the dataset. SampleRNN's prioritisation of timbral realism over content is a property of its time-domain structure and its 'mode-covering' behaviour: its goal is effectively to generate plausible *signals* that are in keeping with the *overall* distribution of the dataset.

These behaviours could be clearly seen when creating models of the same datasets with each architecture. SampleRNN models of the 'Tone Rows' dataset, which contains significant repetition of content, generated musically interesting content that struck a pleasing balance between capturing some of the character of the intervallic content while retaining some of 'its own' character distinct from the dataset. Especially strong SampleRNN results were obtained when I augmented this dataset by adding two time-stretched copies of the entire dataset and a polarity-inverted copy of each, effectively increasing the dataset size by a factor of six; the effect of doing so on the fidelity of the generated outputs can be heard in 'Fake Lrrning'. WaveGAN models of the same dataset, on the other hand, frequently just regurgitated the dataset's content.

WaveGAN exhibited strongest results - 'strongest' as judged by their perceived novelty - on the 'G Major' and 'Melodic Improvisation' datasets. This was, I suspect, due to their balance between consistency and variety of content. Neither dataset was so varied as to fall foul of the possible pitfalls of WaveGAN's 'mode-seeking' behaviour, in which case it simply generates samples relating to a small number of identified regions of the dataset. Each of these models' generated samples frequently-enough provided a sense of novelty from the dataset to be considered creatively useful; an example of this novelty would be the folksy quality of some curated phrases generated from the 'G Major' modelas can be heard in the raw WaveGAN outputs on which 'Major Piece' is based, included in the portfolio.

Generating interesting results with either architecture on the 'Timbral Improvisation' dataset proved problematic, my inference being that the dataset is so timbrally diverse that its effective statistical properties are a bit *too* complex and irrational for successful modelling to be a reasonable prospect. WaveGAN behaviour in this instance was to converge on a small number of the sounds in the dataset and simply recreate them, failing to generate any meaningful novelty. SampleRNN responded better to the more rhythmically active aspects of the dataset, generating sounds derived from instrumental key noise and the sound of the breath that I found interesting and attractive. However, the many sustained multiphonics in the dataset resulted in a lot of long, often single-note sounds that I found crude and uninteresting. Despite a lot of experimentation, in neither case did I manage to make what I felt was the beginning of successful creative work using samples generated from models of the 'Timbral Improvisation' dataset trained with either architecture.

5.4.1 Observed Behaviours with External Datasets

During my research project I was involved in side projects which involved generative raw audio modelling of externally authored datasets. Engaging with these projects generated useful additional observations about how some of the above-mentioned model architecture behaviours manifest in practice.

The first project, 'Gandering'[15], was a Manchester Jazz Festival 2021 Digital Originals commission in which I sub-commissioned three musical colleagues, Adam Fairhall, Johnny Hunter and Otto Willberg to record themselves improvising solo on piano, drums and double bass respectively, with a view to creating generative models of their datasets and creating audiovisual works out of them, using the audiovisual processes described in Chapter 4 as used in version 1 of 'SoloSoloDuo'. Training WaveGAN models of these datasets proved a valuable demonstration of WaveGAN's 'mode-seeking' behaviour as applied to datasets of improvisation. While I encouraged the contributors to limit the degree of timbral diversity in their solo sets, the three resulting datasets still contained greater diversity of sounds than in the majority of datasets I had created for this project.

The double bass dataset was the most unruly, featuring pizzicato playing in all registers as well as bowed playing across all registers. WaveGAN models trained on this dataset generated samples that reliably fell into a small number of categories - pizzicato mid-register, bowed low register bowed and bowed extreme high register. The model trained on the drums dataset exhibited similar behaviour, generated samples most strongly featuring either mid-range toms, cymbals or rimshots. Here, WaveGAN's observed 'behaviour' with datasets of timbral diversity was to effectively hone in on a small number of identifable regions and generate samples that sounded like them. The piano dataset fared better, presumably owing to its relative uniformity of timbre. My feeling with all of these datasets was that not only the diversity of timbre but also the wide range of pitch registers created an unfeasible learning challenge.

While I had initially planned to create 'fake' improvised solos from each dataset, the narrowness of scope of these generated samples prompted a rethink. In the end I created a piece based on stacking a small number of the samples on top of each other, the result being 'Gandering 1', presented in the 'Applications' section.

The second project was a collaboration with departmental colleagues Dr Tom Collins, PhD candidate and singer-songwriter Jemily Rime and rap artist G-Zone. This collaboration resulted in 'Nobody New' an entry to the 2022 AI Song Contest[16]. One of my contributions was to create SampleRNN models of vocal datasets contributed by Jemily and G-Zone (I also created small language models of song lyrics and created the video for the song).

This proved a valuable lesson in the importance of capturing a clear and direct recorded signal when training SampleRNN models. Doing so is a more straightforward endeavour when recording vocals since the relationship between sound source and microphone is very considerably more direct than when recording tenor saxophone. In addition, neither of these datasets featured significant timbral diversity and G-Zone's recordings in particular occupied a very narrow pitch range. The consistently high quality of the generated samples underlined that the datasets' shared attributes of consistency and clarity of recorded signal, minimal timbral diversity and narrow pitch range enabled easier statistical modelling.

5.4.2 Machine Learning Libraries

It seems worth noting that the most functional implementations of WaveGAN and SampleRNN are written in Tensorflow[17] and yield results of higher fidelity than their re-

[15]Hanslip, Mark, 'Gandering 1', YouTube video, https://youtu.be/5dIxUWNGndc.

[16]Collins, Tom, Alex Gonzalez, Jemily Rime, Jack McNeill and Mark Hanslip, 'Nobody New', YouTube video, uploaded by 'G-Zone', Jun 16 2022, https://youtu.be/PwBpW6LYue8.

[17]Abadi, Martin, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh

spective PyTorch[18] versions. When I undertook, principally as a learning exercise, to re-implement WaveGAN in PyTorch (which is more familiar to me than Tensorflow) using Mostafa el-Raeby's version[19] as a guide, I took pains to check that all model parameters were identical to the reference Tensorflow version. Similarly, samples generated from models trained with the PRiSM implementation of SampleRNN are of noticeably higher fidelity than the PyTorch version[20]. This would seem to suggest that in the implementation of equivalent functions within each engine, Tensorflow's underlying implementation might be somehow better suited to raw audio signals than PyTorch's.

5.4.3 WaveGAN, Noise and Loss Functions

As already mentioned, there is a noisiness in WaveGAN's outputs that could be off-putting to practitioners looking to engage in generative modelling of their own data. A plausible culprit for this noisiness could be in WaveGAN's loss calculations. As noted in Chapter 2, the loss function in machine learning serves as a measure of the distance between the model and ground truth data during the training process and is a critical componennt of any machine learning architecture. Since the outputs of WaveGAN's generator network start out from a place of random noise, it therefore stands to reason that its outputs remain noisy because the loss functions are mistaking perceptually irrelevant sounds for 'real' content, allowing them to become part of the generated outputs and even exaggerating them in the process.

Potential solutions to this problem are available now that did not exist at the time of WaveGAN's authoring. One such would be the use of an audio-specific loss function such as the error-to-signal ratio, a perceptually-motivated loss function specifically intended for modelling 1D audio signals[21]. This and other audio-specific losses are available to use in PyTorch via the 'auraloss' package.[22]

5.5 Musical Applications

5.5.1 Real-Time Interaction

In my first creative application of these models' outputs I built on the interactive work established in Chapter 4 by replacing the segmented phrases that acted as computer output material during interaction with outputs from WaveGAN and SampleRNN.

The initial goal of this work was to build on the software created for 'SoloSoloDuo' by having the system outputs be those of generative models of my data rather than simply re-

Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, 'TensorFlow: Large-scale machine learning on heterogeneous systems', 2015, computer software, tensorflow.org.

[18] Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai and Soumith Chintala. 'PyTorch: An Imperative Style, High-Performance Deep Learning Library'. *Advances in Neural Information Processing Systems* 32. Curran Associates, Inc., 2019. 8024-8035.

[19] el-Raeby, Mostafa, 'pytorch-wavegan', code repository, Jun 11, 2019, https://github.com/mostafaelaraby/wavegan-pytorch

[20] Kosakowski, Piotre, Katarzina Kanzka, Joachim Rishaug, 'samplernn-pytorch', code repository, Nov 19, 2017, https://github.com/deepsound-project/samplernn-pytorch

[21] Wright, Alex, Vesa Välimäki. 'Perceptual Loss Function for Neural Modelling of Audio Systems' (conference paper), ICASSP 2020: 45th International Conference on Acoustics, Speech, and Signal Processing, Online/Barcelona, May 4-8 2020.

[22] Christian J. Steinmetz, Reiss, Joshua, auraloss: A collection of audio-focused loss functions in PyTorch, code repository, https://github.com/csteinmetz1/auraloss.

combined segments of my playing. I had originally sought to train generative models of the 'Timbral Improvisation' dataset and continue to use the classifier model used in 'SoloSolo-Duo' to mediate interactions between myself and generative models of both melodic and timbral playing. However, as noted in the 'Behaviours' section above, attempts to model the 'Timbral Improvisation' dataset did not yield results that I deemed good enough for inclusion in an artistic output. As a result, the following two pieces only contain generated outputs from models of datasets with a melodic focus, and pitch analysis is used for mediating the interactions in these pieces.

'Duo with WaveGAN'

In this piece, a trained WaveGAN generator is placed inside an interactive loop similar to the one seen in version 2 of 'SoloSoloDuo' but without the trained classifier. As before, a live input segment is checked against an amplitude threshold to determine whether or not to proceed through the program or revert back to a new live input; also as before, the selected segment is then subject to frequency analysis and pitch-based onset detection to determine its last *salient pitch*. At this point, the WaveGAN generator is prompted with a noise vector and generates several samples. These samples are then subject to the same process of frequency analysis and pitch-based onset detection as the live input in order to determine the sample whose first pitch is the closest match with the last pitch of the live input. This sample becomes the first of the output phrase and is concatenated with other samples to form a longer phrase. The process can be seen in the diagram in figure 5.3.

On reflection, a further development of this work would be to apply the same process of 'last:first' sample matching to the individual WaveGAN samples that are concatenated to form the output phrase. This would further mitigate the interruptive and chaotic nature of the responses. There is also scope for generating more samples than would be needed for the response and discarding those that are not a good match. While the additional frequency analysis would add latency to the system, the extra samples generated would not, since WaveGAN sample generation happens in parallel on the GPU.

When developing this piece I experienced a process of accepting the discomfort of the interactions similar to that described in Chapter 4 in the process of developing 'SoloSolo-Duo'. Initially, my feeling on interacting with the WaveGAN samples was one of feeling thwarted; I felt the computer was obstructing my ability to play what I wanted. On persevering though I began to see the value in the interruptive nature of the interactions, as it was forcing me out of the comfort zone of playing my usual ideas into a state of increased alertness that I felt was beneficial to me in terms of flexibility of ideas generation and of technique. This state of close listening and responsiveness can be heard in the way I respond to the generated phrases, picking up on familiarities in phrase content and making specific pitches the bases of my responses. An uncomfortable playing experience, but a valuable one.

'b.io'

In 'b.io', source material for interactions is taken from two pools of pre-generated SampleRNN samples generated from models trained on the 'Lower Register' dataset and a composite of the 'Middle' and 'Upper Register' datasets. In this piece, interactions are mediated more simply than in the previous duet: if the *mean* pitch of a live input segment is below a threshold, then a sample generated from the model trained on the 'Lower Register' dataset is chosen as output; otherwise, the output sample is one generated from the 'Middle and Upper Register' model. This process is illustrated in figure 5.4.

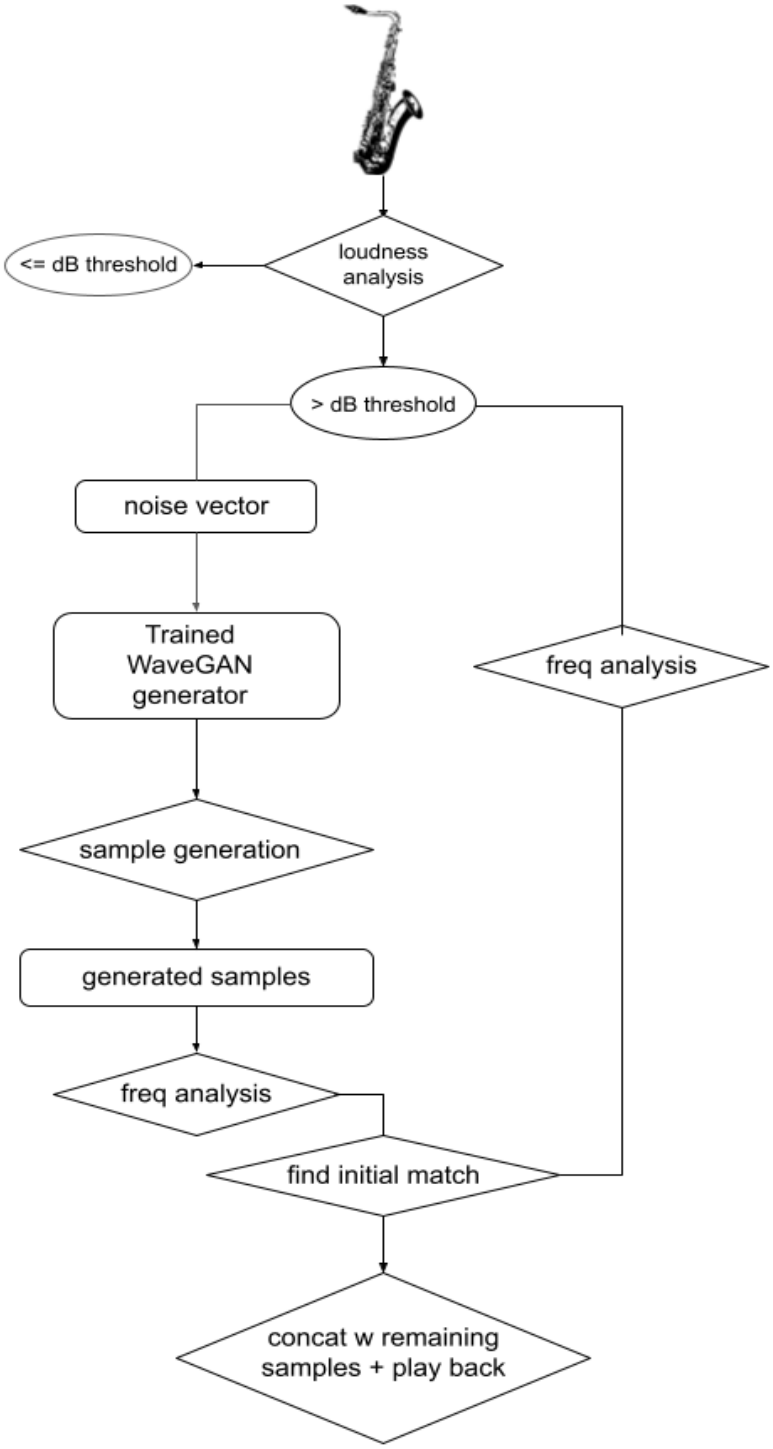

Figure 5.3: Flow diagram of the interactive process in 'Duo with WaveGAN'.

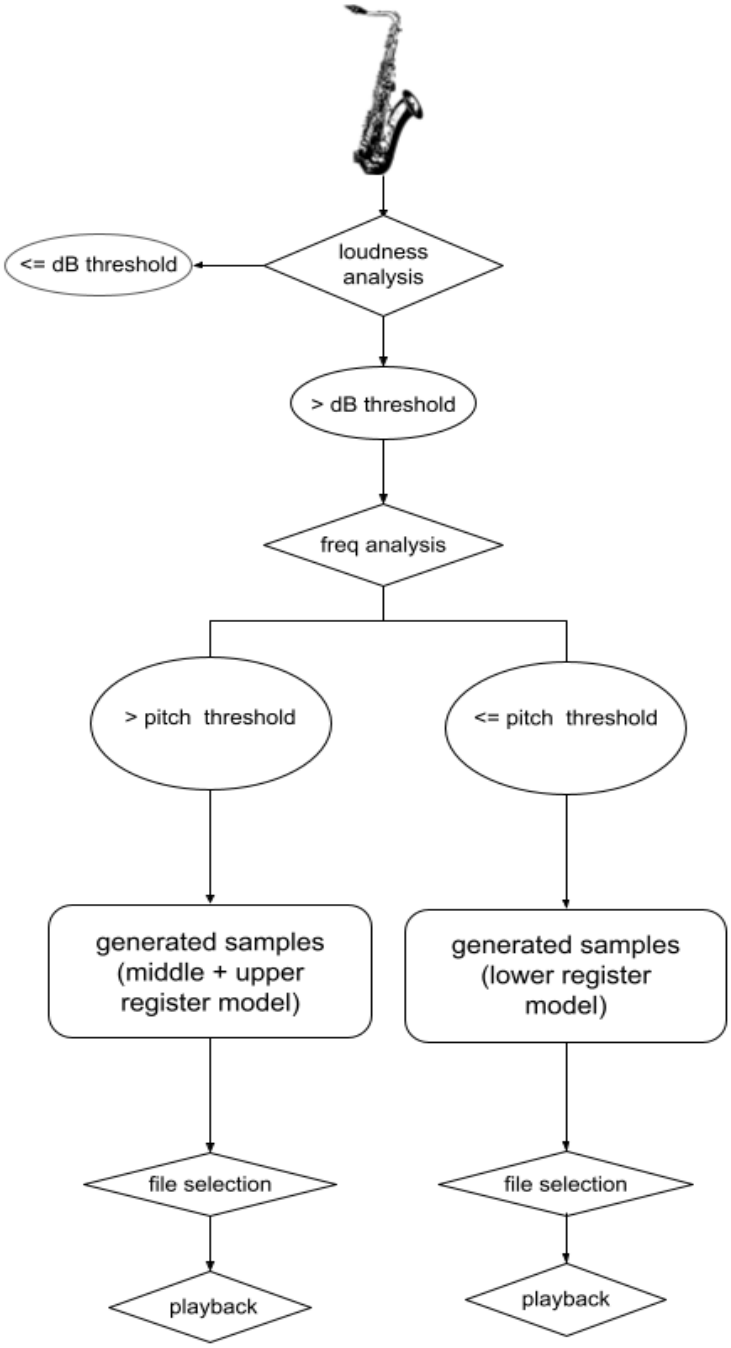

Figure 5.4: Flow diagram of the interactive process in 'b.io'.

This piece was premiered online and in-person at ISMIR 2022[23]. To make it suitable for online presentation I visualised the piece using the same process as in version 1 of 'SoloSoloDuo', passing the audio and a StyleGAN2 model trained using transfer learning on Derrick Schultz's 'Rocks' dataset to the 'Lucid Sonic Dreams' visualiser and adding further edits in Kdenlive.

In the initial development of this piece it was noticeable how much less fraught the interactions felt. I reflect that this is owing not just to the obvious spaciousness of the generated SampleRNN samples compared with those of WaveGAN but also to their timbral realism - it simply felt much easier to interact with a more realistic-sounding simulation of the saxophone. The surreal quality of the samples' musical content added to the enjoyment of playing with them; at several points I can be heard mimicking their peculiar qualities.

This raises a key point about both these model architectures' bevaviours that will be explored in subsequent outputs: their ability to suggest ideas that would not have occurred to me, despite the fact they were trained on my own data. Generative models of raw audio model these fresh ideas *in the image of one's instrumental practice.* This is an important additional affordance that adds to the sense of ownership over and surreality of these models' outputs: hearing unthought-of musical ideas being modelled in the sonic image of one's own instrumental practice is a unique experience and one richly suggestive of new musical ideas. The following outputs explore alternative applications of this unique quality of generative models of raw audio.

Further Reflection

Interacting in these ways with outputs from WaveGAN and SampleRNN models of my playing resulted in two quite distinct improvisatory experiences. In a sense, 'Duo with WaveGAN' was the more exciting and dynamic-feeling of the two. This feeling arises from the greater sense of interactivity created by the pitch-matching mechanism in the program, which simulates something akin to the kind of 'point-to-point' pitch-matching that can happen in group improvisation with human instrumentalists, wherein a player begins their current phrase on the last note or sound played by another group member. A sense of dynamism is also created by the system's outputs being generated on-the-fly. While the distinction between this and a system that uses pre-generated outputs, such as the system created for 'b.io', might not be apparent to the listener, to me as the system's author it represented a significant point of difference that subtly informed the playing experience.

This, however is not to say that it was the preferred playing experience - more that the system created for 'Duo with WaveGAN' is closer to what I would consider the ideal for an AI improvising partner in terms of the system's construction. 'b.io' was significantly more enjoyable to make, owing to several key qualities of the SampleRNN models' outputs: their timbral realism (breath sounds, key clicks and all), their novelty and divergence from the content of the training data, their tendency towards long notes and what I perceive to be their humorous, whimsical qualities. The simpler mechanism for mediating interactions, distinguishing inputs and selecting outputs only by instrumental register and not by exact pitch, also made for a less intense playing experience in 'b.io' than in 'Duo with WaveGAN' that I found more enjoyable. Writing this with some additional hindsight, I perceive less of a gulf in the quality of these two created works than I did at the time, but I still perceive 'b.io' to be the more accessible and enjoyable of the two works to listen back to.

Improvising with the generated outputs in an interactive loop proved to be a valuable practice experience. Most importantly, during sessions playing with each system I

[23]Hanslip, Mark, 'b.io', webpage, Dec 6, 2022, https://ismir2022program.ismir.net/music346.html.

observed myself mimicking the distinct expressive characters of each model's generated outputs: SampleRNN more whimsical-seeming phrases and WaveGAN's more aggressive-seeming outputs. Given time, these are gradually being absorbed into the overall fabric of my playing. I find observation especially exciting: that a statistical model of ones own playing has the potential to widen ones own expressive range. On a more technical level, I observed a definite 'sharpening' of my ear after a session of playing with either system but particularly the WaveGAN system, which was more challenging to navigate improvisationally. One especially valuable aspect of the challenge these systems presented was that, while I would initially fall back on playing 'stock phrases' owing to the initial discomfort I experienced, recognition of doing so forced me to dig deep and aim for a more 'in-the-moment' response - that is to say, the systems were forcing me to improvise more. These experiences feed into notions of 'AI as creative assistant' and 'AI as teacher', respectively; notions that recur throughout this project and will be revisited in Chapter 7.

5.5.2 Source Material for Practicing

'Workshop I' and 'Workshop II'

These pieces serve as a demonstration of the affordance of generated WaveGAN and SampleRNN outputs as raw material for technical practice. The combination of novelty of content and rendering of this content in my own sound afforded by these model architectures, particularly in the case of SampleRNN as already discussed, provided a rich repository of material for ear training and technical practice. This kind of practicing - learning phrases from a recording by ear - is fairly typical of how jazz improvisers absorb new material into the fabric of their playing, improve their ear and expand their technique. However, most jazz musicians tend to do this using other players' content. By contrast, I feel a stronger sense of authorship and ownership over these generated samples, having created the datasets and acquired the requisite skills for training models and generating samples, than I ever did when using other players' outputs as the basis for learning new material.

I found practicing these outputs on my instrument *highly* enjoyable and valuable: a rich resource for ear training, for technical practice and for assimilation of new material for improvising and composing over which I felt an unusual combination of strong authorship but also unfamiliarity. The aim of these pieces was to create a musical representation of this process of practicing the generated outputs on my instrument, motivated by a desire to simply reflect in a piece of music the value I found in engaging with this process.

The repetition of phrases in these pieces reflects the practice I was engaged in: I would loop a sample (or concatenated samples) I found musically interesting and figure out its content until I was able to play it along with the sample at full speed. In the interests of turning this process into a somewhat listenable output, I only include a recording of me playing along with each sample having already gone through this learning process.

I harmonized each sample-playalong pairing using the frequency analysis and pitch-based onset detection technique introduced in Chapter 4, extracting the melodically salient pitches and effectively stacking them into a chord. The resulting chord was rendered using sine tones in SuperCollider. Underpinning each piece is a field recording of a weaving workshop taken from the publicly available BBC Sound Effects archive[24]. As with 'Duo with WaveGAN', additional effects were added to match the noisier-sounding WaveGAN samples: pink noise was added to the sine tones, a busier environmental recording was used and I added distortion to my recorded sound. The SampleRNN version of 'Workshop' is cleaner- and more spacious-sounding in the manner of SampleRNN samples, using clean

[24]BBC Sound Effects, online data repository, https://www.sound-effects.bbcrewind.co.uk

sine tones, a less intensive-sounding field recording and no additional effects on my live sound.

Further Reflection

I have mixed feelings about this musical output. Insofar as it represents the process of practicing in this way, one could say my initial goal was met, but as a musical output in and of itself I see at as a starting point for future work. It lays some useful groundwork for using generatively-modelled outputs and environmental recordings in the context of non-realtime electro-acoustic composition, a field I hadn't engaged with previously. Were I to pursue this line of creative inquiry further I would use only SampleRNN outputs as their less intense character compared with WaveGAN outputs suits the spaciousness of the sound world. I would also pay closer attention to the sound design aspect of the piece, more carefully considering placement and balance of the different components in the mix. I would also make structural variations, as I now find the piece to be somewhat formulaic in its construction.

I will take a sentence here just to restate how unique the enjoyment of practicing these outputs was to me, and encourage any readers with an instrumental practice to try this process out, however onerous the initial data-gathering and model training might seem. I struggle to think of another existing method besides deep learning modelling through which these outputs, novel and yet entirely based on the most familiar and routine aspects of my practice, could have been made. Practicing in this way also acts as a bridge towards using generated outputs as basis for compositions for improvisation, explored in the following section.

5.5.3 Compositions for Solo Improviser

A fruitful application of generative models of raw audio has been in the composition of melodies that act as jumping-off points for solo improvisation. A clear point of reference for these works is the solo music of Steve Lacy. This association with Lacy's work first came about when using the generated samples for technical practice as in the 'Workshop' pieces: the repetition of samples brought to mind Lacy's compositions, a consistent feature of which is the repetition of phrases of intervallic construction; these compositions then serve as springboards for improvisation, a way of taking the improviser, in Lacy's own words, 'safely to the edge'.

This form of composition-for-improvisation has precedents within my own practice. For example in my compositions 'Monobrow' which featured on 'Revival Room' (with organist Adam Fairhall and drummer Johnny Hunter)[25], 'Spiders', which featured on 'The Adding Machine' by Twelves (with bassist Riaan Vosloo, guitarist Rob Updegraff and drummer Tim Giles)[26] and on 'Outhouse' (with reeds player Robin Fincker, bassist Johnny Brierley and drummer Dave Smith)[27], unison melodies serve as a springboard for collective improvisation. When using composed melody as a launchpad for improvisation in groups I and collaborators have usually felt it appropriate to use the melodic content and implied structure of the composed sections as a basis for the improvisation to some degree, eschewing, unless made as a deliberate choice, the tendency often seen in 1960s free jazz to play a melody before veering off into unrelated territory. In my approach to improvising on these compositions I have favoured a structured approach, taking account of their content in my

[25]Mark Hanslip, 'Monobrow', *Revival Room*, EfPi Records, CD and Digital, 2021, https://revivalroom.bandcamp.com/album/revival-room.

[26]Mark Hanslip, 'Spiders', *The Adding Machine*, Babel Label, CD and Digital, 2011, https://babel-label.bandcamp.com/album/the-adding-machine.

[27]Mark Hanslip, 'Spiders', *Outhouse*, Babel Label, CD and Digital, 2008, https://babel-label.bandcamp.com/album/outhouse.

improvisational decisions.

As has been noted in previous sections, a clear affordance of these models is in their ability to generate musical ideas that I wouldn't have thought of, despite their having been trained on materials I am more than familiar with. This aspect of novelty is at the core of the following outputs.

'Gander'

'Gander' is a short composition for solo improvisation derived from curated WaveGAN samples. The initial model was one of my earlier WaveGAN experiments, trained on a composite 'Exercises' dataset consisting of both the Tone Rows and Scales & Exercises datasets discussed in Chapter 3. This piece was the first output resulting from the above-described association of repeated WaveGAN samples with Lacy's compositional style. The first iteration of 'Gander' was an entirely synthetic composition and improvisation, with the composition section made up of the curated, repeated samples and the improvisation made up of randomly concatenated generated samples from the same model. This version can be heard in the track 'Fake Gander.wav'. I also visualised a phrase from the 'improvised' section of Fake Gander.

This initial fake version resulted in an 'improvisation' far more skewed towards the upper register than I would ever tend to venture in practice. I did however find the randomness of the phrases and, for the most part, their *lack* of connection to the composition surprisingly effective and attractive. By contrast, in the 'real' version I stuck with a more structure-based approach to improvising on 'Gander', identifying each phrase's implied key or internal structure and basing my improvised phrases on this.

'Lrnning'

The phrases that make up 'Lrnning' are curated, notated outputs from the most 'successful' SampleRNN model I managed to train, on the Tone Rows dataset discussed in Chapter 2. Two key reasons for this success were keeping the dataset at its original samplerate of 44100Hz and using time stretch data augmentation in lieu of the pitch shift I was previously using. This meant that the model took several days to train, but also resulted in high quality outputs in which the musical contents of the dataset were more successfully modelled than had previously been the case.

When approaching how to improvise on 'Lrnning', it seemed clear to me that a kind of musical structure was suggested by the arrangement of phrases. For example, the first 5 notes of phrase 1 suggest a melodic shape of B-Bb-F-E - a scale found in the Slonimsky Thesaurus of Scales and Melodic Patterns[28] - before giving way to notes more suggestive of a Bb7 / Fmi7 tonality. Phrases 2 to 4 are more obviously tonal: phrase 2 clearly implies a key center of G# minor, while phrase 3 strongly suggests Bb major and phrase 4 suggests C lydian. The lower register-centred 5th phrase is more atonal, while the remaining phrases suggest a key center of F minor.

These implied tonalities result in a loose structure that provides a useful scaffolding for solo improvisation that I followed throughout my improvisation. In total I improvised twice round this implied structure before returning to the composed melodies at the end.

[28]Slonimksy, Nicolas, *Thesaurus of Scale and Melodic Patterns.*

'The Lows'

The phrases that make up 'The Lows' are curated and notated outputs generated from SampleRNN models trained on the Lower Register dataset described in Chapter 3. As discussed earlier in this chapter, SampleRNN models trained on the Lower Register dataset yielded a more consistent quality of outputs owing to the more direct recorded signal in the lower register of the tenor saxophone and to the amount of reiteration of similar musical content in the dataset.

As with 'Lrnning', when approaching improvising on 'The Lows' I began by examining the content of each phrase and identifying a structure implied by the phrases. The phrases that make up 'The Lows' are, on the whole, less strongly suggestive of key centers than in 'Lrnning' - my conception of their content was influenced more by melodic shapes and direction. For example, looking at the score of 'The Lows', phrase 1 is marked by each sub-phrase beginning on a lower register note the pitch of which ascends by a semitone with each sub-phrase; phrase 2 is structured similarly, with each phrase beginning on lower register notes B, D and C. Phrase 3 however strongly implies a whole-tone scale while phrases 4 and 5 begin with an F7-type shape. The final phrase, played only once, is intended as a kind of 'send-off': a transition from the 'in-head' into improvisation and also an ending to the piece's 'out-head'.

As with 'Lrnning', I improvised over two cycles of this implied structure. In keeping with the piece, the majority of the improvisation happens in the tenor saxophone's lower register although I couldn't resist venturing upwards at times.

'Major Piece'

'Major Piece' is another shorter piece derived from curated WaveGAN samples, this time generated from a model trained on the 'G Major' dataset discussed in Chapter 3.

When figuring out how to approach improvising on this piece, I soon became disinterested; while I found the melody in and of itself attractive, its purely diatonic content felt insufficient to spark interesting improvisation. In the end I saw this as an opportunity to begin augmenting my solo improvisation with looping. I had been developing looping environments in SuperCollider in my spare time and 'Major Piece' seemed a clear candidate for looping: since its phrases share the same tonal center, they would likely sound nice layered on top of each other.

The final version features very little improvisation. There are four cycles through the composition's structure in total. Firstly, I play the melodies unaccompanied. This first rendition is then looped, over which I play semi-improvised phrases that are close in character and content to the original melody. This second layer is added to the loop, over which I play a set of more freely improvised phrases. Finally, in the out-head section, the loop is terminated and the phrases are looped individually and stacked on top of each other. Through the piece, a point of convergence is phrase 4, played only once on each iteration and played in unison, providing a nice structural signpost as well as a whimsical endpoint for the phrases.

Further Reflection

While the focus thus far in these reflective passages has been what it felt like to *play* with the trained models and/or their outputs, here it is more appropriate to reflect on the experience of *composing* with the generated outputs, before reflecting on the improvisational aspect of the resulting performances. When reflecting on how the compositional experience compares with my previous norm of composing either on the saxophone, at

the piano, or both, differences between that process and the one being described here are immediately apparent. Whereas in the past I would typically take a musical idea on the saxophone and work out variations of, accompaniments to and harmonisations of that material at the piano, here generation of ideas is taken care of by the trained model, and the piano is replaced with a DAW. Where I previously took note of improvised phrases that I found pleasing while playing, I now simply choose my preferred generated samples; while I would previously flesh out the composition at the piano, I now simply arrange them into a sequence that I find pleasing in the DAW before transcribing the result. This is not to say that I intend to abandon my old process, but there is appeal in the relative ease of production conferred upon the compositional process by the technology. This further example of AI functioning as a creative assistant demonstrates the potential of generative deep learning to streamline musicians' workflows.

A consistent feature of the improvisation in these pieces is how much of the ideas generation is of the 'associate-type', to return to to Pressing's formulation, and in turn, how much of the initial idea generation is based on the compositional statement. The degree of deviation from the composed material varies by piece. The improvisation on 'The Lows', for example, ends up in extended technique territory not explicitly suggested by the composition, but arrives there via a gradual deviation from the composed material, and even at this furthest remove is still within the overarching idea of sticking to the lower register of the saxophone. The improvisation on 'Lrrning' eventually deviates from the written material: the sequence of ascending phrases beginning at 8:32, for example, have no particular basis in the composition itself, but they follow on naturally from the previous idea, which in turn was arrived at through associative processes. At first listen, 'Gander' might seem to have the most interruptive character of these four pieces, but this quality stems from the composition's phrases contrasting each other - the improvisation is still largely very self-associative, with any apparent interruption stemming from reference back to a different phrase of the composition. 'Major Piece' differs here in that the limited amount of improvisation the performance contains has strict structural boundaries around it: the use of looping of the composed phrases confines any deviation from it to what fits around the loops, and as such these notions of associate- and interrupt-type ideas generation are less applicable.[29]

5.5.4 Sampling-based Music; Audio-visual Practices

'Gandering 1'

An output of the 'Gandering' project discussed in the 'Behaviours' section, 'Gandering 1' is the first of a set of four audiovisual pieces made for a commission for the 2021 Manchester Jazz Festival. My aim with the 'Gandering' project as a whole was to create work that synergised the audio and video components into a whole experience. This was a new way of thinking about making work to me - while there is precedent for the use of visual enhancement of the music in 'SoloSoloDuo' and 'b.io', and those pieces share the same visual aesthetic and tight coupling between audio and image, the audio-visual versions of 'SoloSoloDuo' and 'b.io' are visualisations of pre-existing musical works. By contrast, with 'Gandering', the audio and video components were realised in tandem. I later discovered that this is a common approach in the audiovisual art and research community: Louise Harris synthesises a survey of that community as highlighting the importance of 'a synergic relationship in which the combination of the two media creates a third, audiovisual space, .. greater than the sum of its parts.'[30] I also wanted there to be a close relationship

[29] Jeff Pressing, 'Improvisation: Methods and Models' in *Generative Processes in Music: The Psychology of Performance, Improvisation and Composition* ed. John Sloboda (Oxford University Press, 2001), 129-156.

[30] Harris, Louise, *Composing Audiovisually: Perspectives on Audiovisual Practices and Relationships*, Routledge, 2022, 43.

between the audio and video - a tight coupling of the interactions, and some meaningful correlation between the images themselves and the images the music implied.

The audio component consists of stacked WaveGAN samples generated from models of solo improvisation on double bass and piano, with the initial drum sample that underpins the piece having been generated from a SampleRNN model of solo drum improvisation.

The visual component was created through a similar process to that used in version 1 of 'SoloSoloDuo' in Chapter 4. I sourced a dataset of curated floorplans scraped from Instagram posts by Mayur Mistry, and modified the images through inversion and colour saturation. I then trained a StyleGAN2 model via transfer learning on the transformed dataset before visualising the audio using Mikael Alafriz's 'Lucid Sonic Dreams' program.

'Gandering' has in common with 'Lrrning', 'Major Piece', 'Gander' and 'The Lows' significant use of repetition-based structural development in 'Gandering' However, this piece represents a significant departure from my usual processes and as such the resulting music differs considerably from what I would usually make. A significant factor in this divergence from my usual working practices was the decision to work with external datasets and the openness to potential outcomes this entailed: at the time, I was still learning about the behaviours of these architectures, both in and of themselves and with respect to different training corpa, and as such figuring out how I was going to use the generated outputs happened very much on a 'let's try this and see-what-happens' basis. Another factor was the decision to make an explicitly audio-visual work, as opposed to making some music and then visualising it. This decision was very much influenced by the remarkable capabilities of the StyleGAN2 - LucidSonicDreams software pipeline.

While I was not well versed in canonical audio-visual works at the time of making 'Gandering', I find some aesthetic resonance with seminal pieces from the discipline's earlier years such as Mary-Ellen Bute's 'Dada' (1936)[31], Norman McLaren's 'Blinkity Blink (1955)[32] and Len Lye's 'Free Radicals' (1958)[33], particularly in their combination of a light-on-dark visual palette and tight coupling of the audible and visual elements. In 'Gandering', the latter element was partially a by-product of LucidSonicDreams' behaviour but also something I was seeking out - all my experiments with alternative means of visualising sound have sought out a high of audio-reactivity. The use of a light-on-dark visual palette was motivated by taste but also practicality: fades to black were a practical way of delineating sonic activity and silence, and omitting the RGB dimension made any necessary array operations more straightforward.

5.6 Conclusion

As can be seen in the range of outputs from the chapter, generative modelling of raw audio data has a diversity of applications for the creative instrumental practitioner. From my perspective these have included real-time interactivity, generation of novel material for practicing, use of samples in electroacoustic compositions and as source material for composing for solo improvisation; I have confidence that other practitioners will find additional applications. As explored in the sections reflecting on the creative outputs, I found particular value in a number of areas: enjoyable ear-training, expanding ones expressive

[31]Bute, Mary-Ellen, 'Dada', YouTube video, uploaded by 'Musica Visual', Jun 13 2022, https://youtu.be/ihhJxrY3Vig?si=GQR4H5oOsBJ0kIfO.

[32]McLaren, Norman, 'Blinkity Blink', YouTube video, uploaded by 'NFB', May 17 2015, https://youtu.be/q3YeWgUgPHM?si=5gcnmCgy5WBRVPgU.

[33]Lye, Len, 'Free Radicals', YouTube video, uploaded by 'Musica Visual', Jan 19 2023, https://youtu.be/t5ych1ikDfI?si=CUlUEZXmYmOlFpg2.

range, changing ones compositional workflow and venturing into new domains of creativity.

The work in this chapter also generates valuable knowledge of how best to approach modelling data using these architectures and of their behaviours. For the purposes of generating novel content, WaveGAN has been shown to work best with datasets that strike a balance between consistency of musical character and some variety within that content: best results were generated from the 'Melodic Improvisation' and 'G Major' datasets, which achieve this balance. It fared poorly with overly diverse datasets such as 'Timbral Improvisation' and those used in the 'Gandering' project, resulting in modelling only of specific regions of the dataset at the expense of others. With the 'Tone Rows' dataset, which contains significant repetition of musical content, it tended to yield outcomes that, while of higher fidelity than others owing to the easier learning task, sounded like reproductions of the dataset and thus failed to offer much in the way of novelty. Using pitch shifting as a data augmentation tool is a good approach when working with WaveGAN, since it effectively achieves both consistency and diversity in the dataset. Since the generated samples are invariably quite noisy, I see no reason to set the sample rate higher than 22050Hz, especially given the environmental concerns outlined in this chapter. Parameter changes found to have the most significant downward impact on training times have been to disable phase shuffle, decrease the dimensionality multiplier to 32 and enable the 'data_fast_wav' parameter; decreasing the dimensionality multiplier to 32 positively impacted the quality of results too.

Best results with SampleRNN have been achieved when modelling the 'Tone Rows' and 'Lower Register' datasets. Shared aspects of these two datasets are significant repetition of musical content and consistency of content. SampleRNN's time-based approach to modelling raw audio signals effectively give it an almost opposite set of attributes to WaveGAN. Where WaveGAN realistically models musical content, SampleRNN realistically models instrumental timbre and requires additional modifications to the dataset in order for it to model at least somewhat plausible musical content; repetition of musical content within the dataset clearly helps, as does time-stretch augmentation. Augmentations should be kept subtle given its sensitivity to even minor timbral variations in the dataset. Best results tended to be with parameters of 3 RNN layers and LSTM cells; skip connections reliably extended training lengths when used and occasionally improved results. Again given the focus on timbre afforded by SampleRNN, there is justification for training at 44100Hz, which resulted in a noticeable improvement in the realism of my generated samples, though I would still only advocate this if the generated samples themselves are intended to appear in the creative output, given the significant additional energy consumption.

Across both model architectures, clarity of recorded signal in the dataset has proved to be vitally important. As noted at the beginning of Chapter 3, at the outset of my research I had initially thought that a single microphone attached to the bell was a reasonable basis for achieving this clarity. However, there was significant disparity in quality of SampleRNN outputs between the lower register and higher registers of the tenor saxophone. WaveGAN exhibited the opposite behaviour, tending to generate a clearer signal in the middle and upper registers, suggesting it more successfully models higher-frequency content. In my future work with these models I will adopt a dual-microphone approach to capture a fuller picture of the tenor saxophone's projection of sound. Similarly, while I initially thought that pitch shifting for data augmentation was a reasonable choice, it was not clear to me at the data pre-processing stage that this technique had significantly altered my timbre, yet in generated SampleRNN outputs some phrases had a nasal quality that seemed to be a result of using pitch shift. This outcome resulted in my switching to time stretch augmentation when working with SampleRNN in later experiments.

This points to an additional affordance of these model architectures: they return useful information about the dataset itself. Their tendency to exaggerate aspects of the input data in their generated outputs makes them a useful teacher of flaws in our datasets and of how to create datasets more likely to yield improved outcomes.

Chapter 6

Symbolic-Domain Melodic Prediction in Practice: 'i prompt u', 'Strange Loops', 'Taps'

6.1 Introduction

This chapter presents an inquiry into creative musical applications to instrumental practice of symbolic-domain deep learning. The problem of accurately transcribing saxophone melodies in an automated fashion is addressed by means of the custom parser for offline pitch-based onset detection first introduced in Chapter 4. This method serves as a useful tool for text-domain dataset creation from several of the raw audio data sets introduced in Chapter 3. A character-level recurrent neural network (or 'Char-RNN') is trained on this data. The trained model is prompted to generate strings of raw melodic material which are automatically tokenized into Lilypond[1] code and rendered to notation. The resulting notated outputs are then curated and used as the basis for 'prompts' for human solo improvisation and form the basis of the portfolio outputs in this chapter. Outputs of this work are three semi-notated, effects-augmented, structured improvisations: 'i prompt u', utilising granular pitch shifting and delay; 'Strange Loops', created using a custom looping environment; and 'Taps', which incorporates multi-tap delay. The computational processes described in this chapter can be found in the accompanying folder or at https://github.com/markhanslip/PhD_Ch6_Char_RNN.

6.2 Rationale

Development of this work initially came about as a result of frustration with the unreliability of using onset detection and frequency analysis in combination for transcription of saxophone inputs. Owing to their real-time implementations and the difficulty of reliable onset detection for woodwind instruments, methods such as SuperCollider's Tartini.kr[2] and Onsets.kr[3] resulted in inaccuracies such as undetected onsets and frequency errors, leading to an informal personal review of offline, open source pitch tracking methods of which

[1]Kastrup, David, Werner Lemberg, Han-Wen Nienhuys, Jan Nieuwenhuizen, Carl Sorensen, Janek Warchoł, et al, *Lilypond*, version 2.24.1, computer software, lilypond.org

[2]McLeod, Philip and Geoff Wyvill, 'A Smarter Way to Find Pitch', proceedings of ICMC 2005: 31st International Computer Music Conference 2005, Barcelona, Spain, Sep 4-10, 2005, 138-141.

[3]Stowell, Dan and Mark Plumbley, 'Adaptive whitening for improved real-time audio onset detection', proceedings of ICMC 2007: 33rd International Computer Music Conference 2007, Copenhagen, Denmark, Aug 27-31 2007.

speech analyser Praat[4] and convolutional deep learning-based method CREPE[5] proved the most reliable. Discovery of accurate methods of tracking saxophone pitches then allowed me to build a tool to extract the most musically salient pitches by means of pitch-based onset detection, a method first proposed by Collins[6]. Having crafted this tool, I then began investigating the possibility of using machine learning to model the resulting symbolic data and generate streams of melodic data from the model for creative ends.

While early applications of this work involved mapping the generated outputs to banks of one-shot saxophone samples, I found the robotic, staccato character of the resulting music unenjoyable to both listen to and interact with. A much more fruitful application has been in notating outputs from the trained RNN. Doing so provides an alternative basis for practising, in which I learn new phrases from notated outputs (as opposed to learning generated samples by ear). As with outputs generated from raw audio, these strike a balance between novelty - as defined by that which I would not necessarily think to play myself - and relation to the ground truth data.

6.3 Technical Processes

6.3.1 Dataset Pre-Processing

In this section I present a process for extracting melodically-relevant pitch data from my solo saxophone datasets.

First, frequency content is extracted from the file using Praat-Parselmouth[7]. Advantages of using this library are its interoperability with my other Python-based processes, allowing for process automation, great accuracy of frequency estimation on tenor saxophone, presumably due to its proximity in timbre and register to the adult male human voice. Praat also returns NaN values where no pitch is detected, making the resulting data much easier to work with than data containing poor estimations. It also runs very quickly thanks to its C++ code base.

Frequencies are extracted at a rate of one per 10 milliseconds. The resulting values are linearly scaled through conversion to MIDI note format, allowing for straightforward comparisons of pitches. These pitches are then analysed for melodic-rhythmic salience through the use of a rule set. The rule set for determining melodic-rhythmic salience was defined in Chapter 4 in the context of interactivity, so a recap will suffice here:

- A pitch deviates from its predecessor by more than a specified margin (roughly half a semitone), or;
- A pitch is preceded by silence.

This analysis results in an array of 1s and 0s, effectively representing musical onsets. This array is then multiplied element-wise by the pitch array, effectively discarding all pitches lacking melodic-rhythmic salience. The intervening 0s can then be either filtered out to return a dataset of melodically significant pitches, or counted and interleaved with

[4]Boersma, Paul and David Weenink. *Praat*, version 6.3.09, computer software, https://www.fon.hum.uva.nl/praat/.

[5]Kim, Joon Wook, Justin Salamon, Peter Li, Juan Pablo Bello, 'CREPE: A Convolutional Representation for Pitch Estimation' (conference paper, ICASSP 2018: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, Calgary TELUS Convention Centre, Alberta, Canada, Apr 15-20, 2018).

[6]Collins, Nick, 'Using a Pitch Detector for Onset Detection' (conference paper, ISMIR 2005: 6th International Conference on Music Information Retrieval, London, UK, Sep 11-15, 2005).

[7]Jadoul, Jannick, Bill Thompson and Jan de Boer, 'Introducing Parselmouth: A Python Interface to Praat', *Journal of Phonetics* 71 (2018): 1-15.

the pitches to give a set of pitch-duration pairs. In order to be consumed by the language-domain RNN I use, the resulting data is then recast to Python strings.

Figure 6.1 shows the above-described process.

Adapting this Work for Other Instruments

If applying this process to other instruments, use of a different pitch tracker would be necessary. To this end, the deep learning model CREPE[8], which is empirically shown to generalise to more instruments than the existing state-of-the-art methods, would be appropriate. The process for extracting pitches using CREPE is different from a more conventional and established approach such as Praat. First, unless using a GPU with very large memory capacity, it is first necessary to segment the input data as long audio files tend to result in GPU memory errors. Second, the data frames outputted by CREPE need to be concatenated, and their contents filtered for frequencies with a high (90%+) confidence score, with the remainder set to NaN or 0. The frequency column can then be extracted and the remaining data pre-processing steps outlined above can be applied in the same way. While there would be a compromise on speed and efficiency compared with non-deep learning-based f0 estimation methods, CREPE is remarkably accurate and generalises to a greater range of instruments than previous methods.

6.3.2 Training the Model

The text-format dataset created in the previous steps is passed through a shallow, character-level recurrent neural network. While it was originally a side project of engineer Andrej Karpathy[9] and has long been the province of NLP ('Natural Language Processing', the statistical modelling of large bodies of natural language data) tutorials rather than being considered a serious architecture for language modelling, Char-RNN has a strong precedent for symbolic music generation, having formed the basis of Bob Sturm's folk-RNN project[10].

One major advantage of using RNNs for symbolic music generation over more well-established methods such as Markov chains is that RNNs are capable of generating predictions of arbitrary length from minimal inputs. Markov chains require their input to be of equivalent length to the output, whereas with RNNs a minimal input, such as a single pitch value in numeric form, can be used irrespective of the length of string being generated (a small caveat to this is that prediction lengths should not exceed the 'chunk_len' training parameter, less the length of the initial prompt). This makes the process of prompting a trained RNN to generate outputs trivial to automate.

The process of training and prompting the model in a single pass is shown in Figure 6.2.

6.4 Discussion of Raw Outputs

Similarly to samples generated from SampleRNN models in Chapter 5, generated outputs from this RNN often relate to dataset inputs only ambiguously. Dataset-output relationships in this context are additionally difficult to discern given the large composite dataset used and its reduction to simply a sequence of pitches in text format. Where a

[8]Kim, Joon Wook et al, 'CREPE: A Convolutional Representation for Pitch Estimation'.

[9]Karpathy, Andrej, 'The Unreasonable Effectiveness of Neural Networks'.

[10]Sturm, Bob et al, 'Folk Music Style Modelling by Recurrent Neural Networks with Long Short Term Memory Units'.

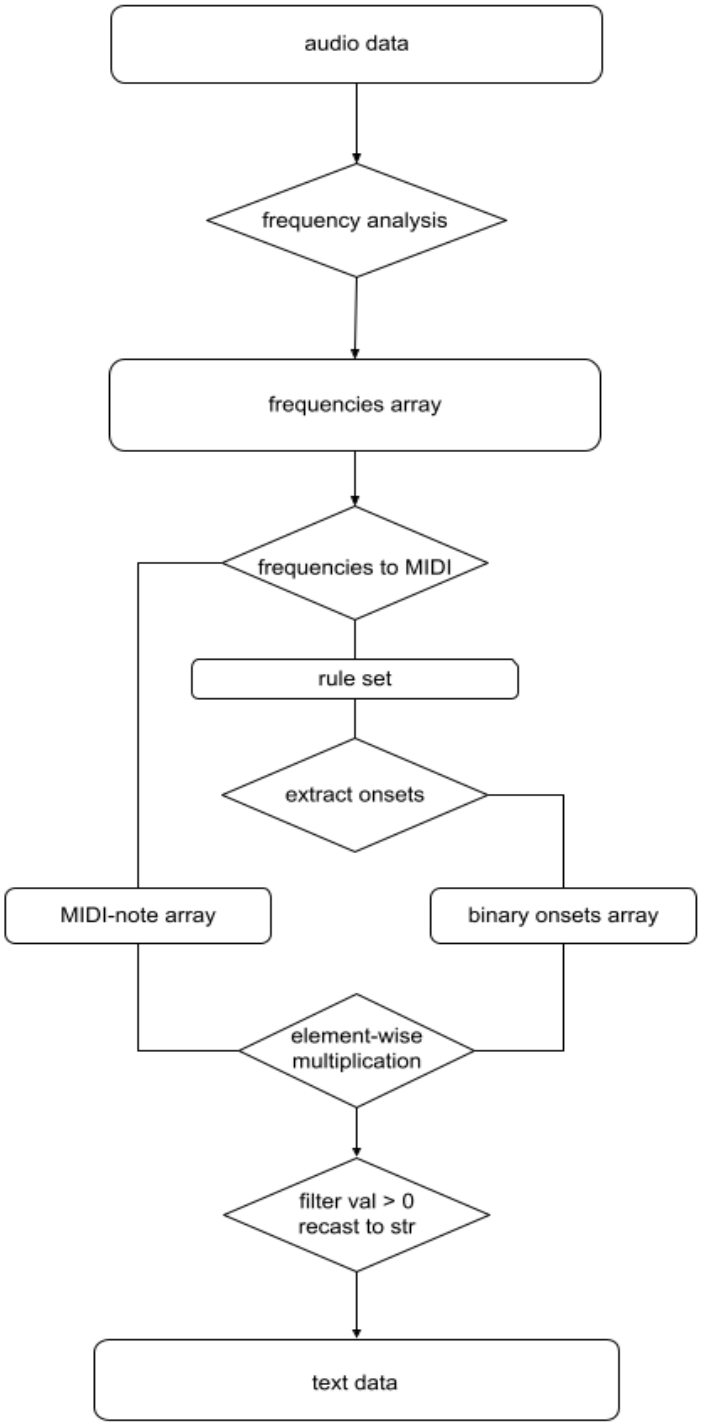

Figure 6.1: Flow diagram showing all stages of pre-processing the training data to text format.

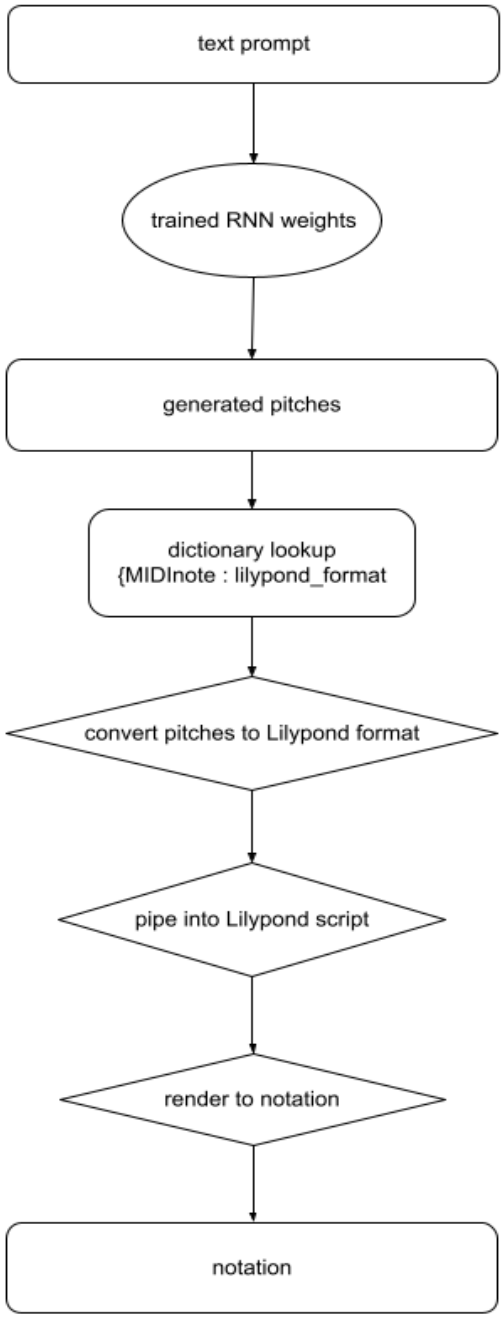

Figure 6.2: Flow diagram showing the process of training and prompting the model in a single pass.

strong relationship can be clearly discerned however is in the presence of intervallic structures clearly derived from the Tone Rows dataset. The first three examples below, taken directly from the raw model outputs, show material clearly derived from the 3rds (figure 6.3), 4ths (figure 6.4) and 7ths (figure 6.5) tone rows first illustrated in Chapter 3; the fourth example below is derived from the parts of the dataset where the tone rows had been scrambled (figure 6.6), with some containing additional displacement of octaves:

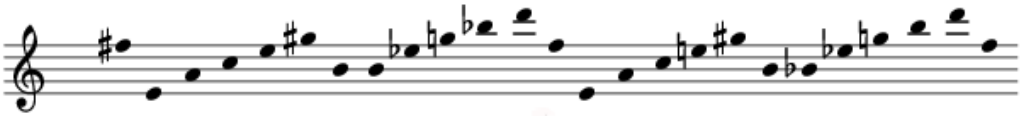

Figure 6.3: A raw Char-RNN output showing clear influence of the '3rds' tone row from the Tone Rows dataset.

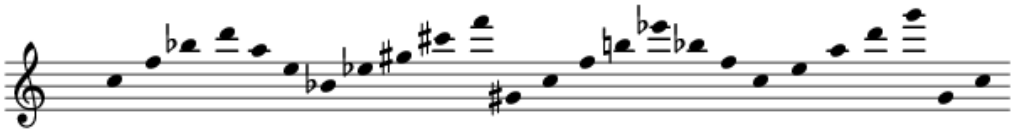

Figure 6.4: A raw Char-RNN output showing clear influence of the '4ths' tone row from the Tone Rows dataset.

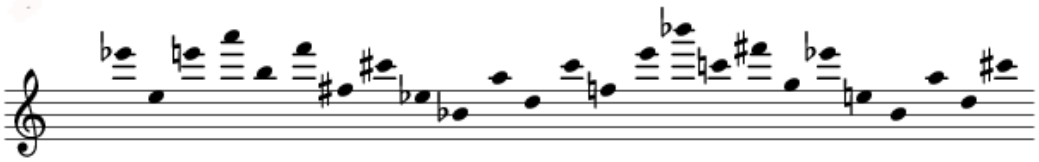

Figure 6.5: A raw Char-RNN output showing clear influence of the 'major 7ths' tone row from the Tone Rows dataset.

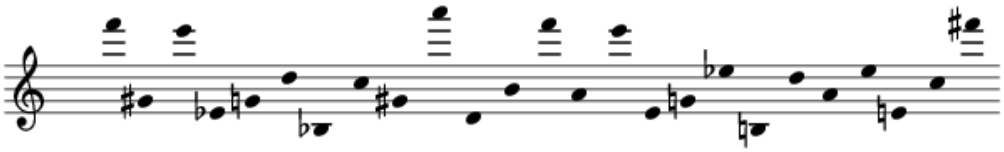

Figure 6.6: A raw Char-RNN output showing clear influence of randomised tone rows from the Tone Rows dataset.

This effectively demonstrates a similar behaviour on the part of Char-RNN to SampleRNN: dataset content such as that contained in the 'Tone Rows' dataset where material is subject to repetition or close variation tends to appear more clearly in the outputs. Undesirable 'over-fitting' (where the model simply generates content identical to the dataset) is still avoided owing to a combination of the RNN's mode-covering behaviour and the abundance of other material in the dataset.

Interestingly, some of the raw Char-RNN outputs also contained instances of single-note repetition, which is generally not something I do much in my practice. This tendency might be attributed to the presence of triple-tongued exercises in the Tone Rows dataset, where each note of the row is effectively played three times before moving onto the next. This is the clearest data-centred explanation I can find for why consecutive repeated notes should be considered a strong statistical probability when predicting new material. While

Yin et al [11] discovered a similar tendency in their transformer-based modelling of MIDI representations of Mozart and Haydn and interpreted it as under-fitting (a tendency in machine learning for the model to inadequately fit to the training data, resulting in vague outputs), I made the choice to exploit it creatively in one of the phrases in portfolio output 'i prompt u', discussed below, and throughout 'Taps'.

6.5 Musical Applications

6.5.1 Compositions for Solo Improvisation and Effects

The notated outputs from my Char-RNN model proved to be useful source material for structured improvisation. These outputs' very minimal specification, free of expressive markings or rhythm, had the effect of affording me a greater sense of freedom to decide how to apply them than the products of previous chapters' work. I experienced a greater sense of permission here than in previous chapters to imagine what additional inputs would bring these raw materials to life. It seemed an ideal opportunity to put into practice some effects I had been working on in SuperCollider. As a result, each piece here contains augmentation of my acoustic sound through effects, building on the use of looping in 'Major Piece' in Chapter 5, as well as on the use of compression in SoloSoloDuo and chaotic noise and distortion in 'Duo with WaveGAN'.

'i prompt u'

In 'i prompt u', I hand-curated melodies generated from the composite model to form a progression from tonal to atonal. The first two melodies are clearly suggestive of closely related key centers of C major and D minor. They then diverge to more specific tonalities, the third and fourth phrases suggesting an altered F-sharp dominant and Bmi713. The latter phrase could also be conceived of purely as a four-note cell of 'G-sharp-C-sharp-D-B'. Phrase five introduces chromaticism before giving way to explicity atonal phrases in the manner of serialist classical music and of course the Tone Rows dataset.

In this piece my acoustic sound is augmented through the use of SuperCollider's PitchShift.ar unit generator; effects in this piece were added post-recording as the results were better than in real-time. During the more tonal passages only the 'windowSize' and 'timeDispersion' parameters are used to create a somewhat randomised delay effect. Window size was set to the size of the entire section plus the maximum randomised time offset which is also passed to the timeDispersion parameter. The very large window sizes created a clarity in the effect that I found pleasing and that was problematic to recreate in real time due to the large window size creating a significant pause before the effect kicked in.

As the piece progresses, pitch-based effects are added. At phrase five, where chromaticism is first introduced, the 'pitchRatio' parameter is increased to shift the pitch of the effect by a whole tone. In later phrases, 'pitchDispersion', a random deviation from the pitch ratio, is added, bringing in highly unpredictable transpositions of the source material that I found both amusing and well-suited to these later phrases' more angular character.

Improvisation in my performance piece was restricted to variations on the written phrases as opposed to 'outright' improvisation. There are more variations on the more tonal phrases towards the beginning of the piece. This was largely a matter of taste and practicality - I felt that variation was easier in the earlier phrases owing to the possibilities offered by their implied tonal center. I also felt that the delay effect used on earlier phrases lended itself more comfortably to variation than the more chaotic effects used later in the

[11] Yin, Zongyu, et al, 'Measuring When a Music Generation Algorithm Copies Too Much: The Originality Report, Cardinality Score, and Symbolic Fingerprinting by Geometric Hashing'.

piece: given the density and unpredictability of activity created by the use of random pitch dispersion it seemed sufficient to stick to the written melody.

'Taps'

In the process of curating outputs for 'i prompt u', I noticed a tendency for the raw Char-RNN outputs to converge to a single note that would then be repeated. As noted in Discussion of Raw Outputs, a probable reason for the presence of repeated consecutive pitches in the predicted outputs from this model is the abundance of exercises in the 'Tone Rows' dataset in which each note is triple-tongued. The structure of 'Taps' exploits this tendency. I first deliberately selected phrases with this characteristic and what I deemed to be an interesting preceding phrase. I then developed the characterising aesthetic and structure for the piece; staccato written phrases interspersed with equally short, improvised sounds that are explicitly derived from the final, repeated notes of each composed phrase. In the final output, improvisation is kept very sparse and restricted to a single conceptual space of exploring degrees of timbral variation that can be derived from associating a sound with a single pitch and/or the fingering of that pitch on the instrument.

A multi-tap delay effect is used throughout, with transitions from composed melody to improvisation emphasised by changes in the effect parameters, with additional delay taps in the improvised sections. These changes are triggered by the use of a foot pedal. These short improvised sounds frequently involve extended techniques such as multiphonics and unconventional manipulation of the breath, articulation and embouchure.

'Strange Loops'

In 'Strange Loops', I selected outputs that implied a shared tonal center of E harmonic major. In the first section, single notes and short melodic fragments are looped with crossfade in order to build up texture and establish the tonality. Occasional granulation of the live input is added for sonic interest. This section gives way to a second section in which longer composed and improvised phrases are looped and allowed to interact in unpredictable ways, creating a loose polyphony. The third section is more textural, building layers of multiphonics (specified in the score with both notated pitches and key combination charts) whose constituent pitches are representative of the tonal center. Eventually, this texture breaks down into shorter multiphonic sounds which further deteriorate towards the end of the piece.

Further Reflection

Creating new compositions from these predicted pitch sequences was a relatively comfortable experience, and a freeing one compared with the work in previous chapters. While of course these sequences of notated pitches preserve almost none of the expressive quality of the generative audio outputs of the previous chapter, the sense of ownership afforded by training on one's own data remains. The fact of the outputs being based on my data led to a more direct association between what was on the page and what I would play than might be the case if, say, the outputs had been created through some more abstract generative process such as a 12-tone system. This sense of ownership and connection to my playing, combined with the very minimal specification of the notation, created a kind of 'open playing field' on which I felt a noticeable sense of freedom and agency compared with the work in previous chapters.

This sense of ease might also be in part a function of my background and training. The previous chapter's outputs relied heavily on my ear training whereas this chapter's outputs relied on my ability to interpret notation. While I like to think I have a 'good ear', a significant proportion of my formative years was spent sight-reading difficult music

in ensembles, and as such I might simply be more comfortable working with notation than by ear. I would also contend however that working by ear is inherently more difficult than interpreting simple notation.

Could go into implied scales in 'i prompt u' e.g. hira-joshi scale in first phrase, minor pentatonic in second, whatever the fourth one is called, dominant bebop scale in the first transposed phrase, Bartok-y-ness of the semitoney shifting phrase, implied 12-tone stuff later on.. again, a preference for spaciousness, repetition, clarity revealing itself. Mix of very ordered and more chaotic sections in Strange Loops and i prompt u, reflect tendency to improvise very carefully with composed material and play more experimentally / chaotically when no compositional material present.

6.6 Conclusion

Generation of notated pitches at first seemed to me to be an unlikely source of ideas for creation of new music in my creative practice. Compared with the model outputs in Chapter 4, which often have their own expressive character, the generated outputs used in this chapter have a relative absence of expressive or idiomatic quality beyond what I, as their effective co-author, infer. This proved to be unexpectedly freeing - the lack of specification in the raw outputs allows me the creative agency to bring them to life however I wish. This agency shows through the range of character in the creative outputs described in this chapter as well as in the enjoyment I experienced in making them.

Thinking deeper than these tacit notions of freedom or agency, I now reflect that a creative *associativity* drives the outputs created in this chapter. In 'i prompt u', the associations of fragments of melody with tonal centers and interval structures and the corresponding assignment of pitch shift effects are at the core of the creative impulse behind the piece. In 'Taps' I associate repetition of single notes in the generated outputs with the timbral possibilities of those notes, creating a structure for improvisation and corresponding structure for assignment of the tap delay effect used in the process. In 'Strange Loops' I associate common pitch groupings in the raw outputs with a specific tonal center and further associated certain multiphonics with this tonality; effects were then deployed with vertical (harmonic) and horizontal (melodic) emphases on this tonality in mind.

While one could argue that a more commonly-specified pitch-generating algorithm such as a 12-tone row generator could do the same job as my RNN, the sense of authorship afforded by training such an algorithm on ones own data outweighs mere random number generation. Furthermore, my experiments with both SampleRNN and Char-RNN have demonstrated their tendency to respond positively to repetition of content in the dataset. This tendency of RNNs could be further exploited by creators by adding more of the content they wish to see in the outputs to their datasets and creating uncertainties by adding contrasting material, making them more flexible and experimental than rules-based algorithms.

Chapter 7

Conclusions

7.1 Introduction

In this chapter I will discuss the insights generated through the research and outputs presented in this thesis; I will then outline and discuss core themes that have emerged as a result of this work.

7.2 Summary

Through this thesis and accompanying portfolio, I explored creative applications of three machine learning tasks - classification, generation and prediction - using deep learning model architectures to model original data drawn from my practice as an improvising saxophonist. This was done to answer questions pertaining to the following:

- The practicalities of modelling recordings of systematic instrumental practice on the saxophone with deep learning for applications in creative music practice;
- Potential creative applications of the resulting models to instrumental improvisation, instrumental practice and electroacoustic improvisation and composition;
- The *extent* of these models' usefulness for these applications;
- Behaviours and qualities of different deep learning architectures with respect to the datasets on which they are trained.

In the following sections I will summarise and reflect upon the core research chapters before more fully addressing my initial research questions in light of the work presented.

7.2.1 Datasets

In Chapter 3 I presented a number of original audio datasets created to serve as the bases of the deep learning models trained to do this work. In keeping with the initial motivation of exploring 'black-box' modelling of systems of musical information to creative ends, the datasets are a reflection, to varying degrees, of ways in which I categorise my improvisation practise. They range from, at their most systematic, strict exercises (as seen in the 'Scales and Arpeggios' and 'Tone Rows' datasets), to, at their most loosely defined, improvisation within bounds (as seen in the 'Melodic Improvisation' and 'Timbral Improvisation' datasets); the 'Register' and 'G Major' datasets occupy a space in between these two poles, somewhere between strict exercises and improvisation.

7.2.2 Classification of Spectrograms

In Chapter 4, an exploration of audio classification using constant-Q spectrograms, I presented two versions of 'SoloSoloDuo', a structured improvisation for saxophone and classifier, the format of which is a reflection of the process of creating audio datasets, training a classifier on them and using the classifier to mediate interactions between the live input and sample sets derived from phrase-based segmentation of the first two solo sections. The first version of 'SoloSoloDuo' relied heavily on the classifier for mediation of interactions, using only a counter method to mitigate false classifications and amplitude filtering to help distinguish played inputs from environment sounds. The second version featured a classifier re-trained on salience spectrograms for additional robustness to differences in acoustic conditions between training data and playing situation, and additional input differentiation in the form of pitch and onset analysis to make system responses more appropriate and less jarring to play with. The resulting pieces have an angular character that reflects the sharp contrasts in improvisatory approaches between the first two solo sections, the need to 'speak clearly' to the classifier and the interruptive dynamic created by interactions with the trained classifier.

7.2.3 Unconditional Raw Audio Generation

In Chapter 5, I presented numerous applications of the outputs of deep learning models trained using two model architectures for unconditional raw audio generation, WaveGAN and SampleRNN.

The first outputs, 'b.io' and 'Duo with WaveGAN', represent a continuation of the interactive work started in Chapter 4, albeit without the use of classification to mediate interactions. In 'b.io', interaction with pre-generated samples from SampleRNN models of the 'Lower Register' and a composite of the 'Middle-' and 'Upper Register' datasets was mediated by a straightforward on-the-fly pitch analysis of the live input, used to determine which register of the tenor saxophone the input predominantly occupied to determine which sample set to select a response from. Of particular interest in this piece is the way in which the gestural, vague character of the SampleRNN samples affects my playing, opening up a subtly different mode of expression. In 'Duo with WaveGAN', a WaveGAN generator model trained on the 'Melodic Improvisation' dataset is placed in the interactive loop and prompted on the fly. Its generated samples are reordered according to pitch and onset analysis of both the live input and the samples themselves, concatenated and played back, mitigating what would otherwise be a jarringly random virtual duo partner. In practice, the value of this setup was more as a kind of sparring partner for working on the agility and flexibility of one's improvised responses. I could however see more direct creative potential in interacting in this way with WaveGAN models trained on audio of a different instrument such as bass. The piece also highlights WaveGAN's generation speed which is such that it is suitable for real-time applications, an affordance that would surely endear it to other practitioner-researchers.

'Workshop I' and 'Workshop II' are compositions that represent the process of using pre-generated and curated WaveGAN and SampleRNN samples as material for ear training and technical practice. I made these pieces to highlight what I strongly feel to be a valuable affordance of generative models of raw audio to instrumental practitioners. Defining features of this affordance are the way novelty of content presents a challenge to the ears and technique, and the way familiarity of sound and expression afford enjoyment, assistance in learning and a sense of ownership: my experience was of finding it significantly easier and more enjoyable to execute and absorb this unfamiliar content because it had been modelled in my own sound and expression. I would strongly encourage advanced instrumental practitioners to explore this application of generative models in their own

practice, albeit tempered with a note of caution about mitigating the environmental impact of wider uptake of this method - I would encourage users to downsample their audio data and downgrade the model dimensionality multiplier to save on computation time.

'Lrrning' (and its computer-generated counterpart 'Fake Lrrning'), 'Gander' (and its similarly fake counterpart 'Fake Gander'), 'The Lows' and 'Major Piece' are compositions for improvisation based on transcription of generated and curated SampleRNN and WaveGAN samples. A feature of all of these pieces is the degree to which their phrases are loosely connected by similarity (each piece consists of samples generated from the same model, which in turns corresponds to a single dataset) but also quite varied in content, suggesting counterintuitive musical structures. A feature of all of the performances of these pieces is the degree to which the improvisation follows the implied structure of the compositions. Across all four pieces, close attention is paid to the implied tonality or character of each phrase and to the implied structure the combination of these phrases generates.

'Lrrning', for example, contains phrases that are clearly suggestive of atonality and others that strongly suggest tonal centers, despite all being ultimately derived from the 12-tone 'Tone Rows' dataset. 'Fake Lrrning' is significant in that it represents arguably the most successful sample generation of the project, a result of keeping the dataset at the original sample rate, and the use of time stretch data augmentation. Sample quality was also consistently high (this cannot be said for other models, outputs of which required significant curation), presumably the result of a more appropriate dataset-model size combination. This suggests that it would have been a better choice to downgrade the dimensionality of the model architecture when training on smaller datasets, and was borne out by more consistent results when I did the same with the WaveGAN models.

7.2.4 Symbolic Prediction

In Chapter 6, I explored the use of a language-domain RNN adapted for symbolic representations of pitch, using appropriate prompts to predict strings of pitch information from a model trained on a composite of all of the datasets besides 'Timbral Improvisation'. This audio dataset was pre-processing using the same process as that used for pitch-based onset detection for managing interactions in 'SoloSoloDuo' in Chapter 4 and 'b.io' and 'Duo with WaveGAN' in Chapter 5. The strings outputted by the model - melodic predictions, effectively - were converted to Lilypond code and rendered to music notation. I then curated the outputs, identifying which phrases were of most interest to me; this process was governed by a combination of playing through identified notations, considering their relationship to the dataset and imagining what can be done with them creatively.

This model's outputs afforded me more creative agency than I had experienced in any of the previous chapters; I now reflect that this was due to a combination of their minimalism, the rich suggestiveness of certain phrases' similarity to aspects of the dataset, and because interfacing with conventional notation (as opposed to interactive systems or audio samples) is something I have done regularly since childhood - this was 'home territory', in a way. They also provided a secure basis for some rewarding experimentation with digital effects such as multi-tap delay (in 'Taps'), looping and granulation (in 'Strange Loops') and granular pitch shifting (in 'i prompt u'). The resulting compositions and recordings are now a valuable basis for more improvisatory experiments with a live solo saxophone-plus-digital-effects setup.

7.2.5 Recap of Research Questions

To address my initial research question, **'What are the practical implications of using recordings of systematic instrumental practice on the saxophone as train-**

ing data for deep learning models for applications in creative music practice?', the answers uncovered in each chapter do vary to an extent depending on the data domain the recordings are being translated to, but before I address these specificities I will highlight some themes that recurred across all attempts to model my recordings.

The first is that it is crucial that the audio data being modelled is recorded in a consistent manner and in as noise-free an environment as is practical; this is especially pertinent when modelling the raw signal as in Chapter 5, but really applies to any audio intended for further analysis and modelling. The most practical solution here is to use a microphone with strong off-axis rejection, such as one with a supercardioid pickup pattern or one with a figure-of-eight pattern with a soundproofing buffer to prevent the rear side from picking up room sounds.

The second practical consideration is that deep learning models of audio generally require a quantity of audio that is not straightforward to acquire; time and energy must be fairly abundant if an instrumental practitioner is to record themselves practicing for the hours required, especially after taking into account post-recording edits such as removing silences and material one wouldn't wish to be modelled. To this end, the data augmentation techniques explored in this thesis, which vary according to task and model architecture, are a straightforward way of balancing the time and energy demands of recording. Indeed, in the case of SampleRNN, data augmentation proved necessary to address the balance of data quantity and consistency of musical content required for modelling to be successful. A potential method for musical practitioners seeking to accumulate their own training data not explored in this thesis is the use of source separation; a musician could potentially isolate their instrument from the mix of an existing album they played on and use the resulting stems as training data. This method would be made more challenging by the inevitable audio artifacts (source separation almost never results in completely 'clean' stems) and by the restrictions imposed by the data existing source separation models were trained on, but this remains a plausible approach.

Specific to the work in Chapter 4, when using machine listening to extract features for classification tasks, the *choice* of feature(s) is an important practical consideration. A musically-motivated time-frequency reprentation such as the CQT turned out to be an appropriate choice for 'SoloSoloDuo', particularly because I was classifying a large (1.4 seconds) window of content, but for others this vary according to what is being categorised. For example, if one is seeking to differentiate the timbres of individual sounds then the Mel Frequency Cepstral Coefficient (MFCC) would be a more approriate input representation. This example highlights a further practical consideration of what model architecture to use - while an image classifier was largely appropriate for the categories of spectrogram data I was seeking to differentiate, a simpler neural network architecture such as an MLP is typically sufficient for classifying lower-level features such as MFCCs.

Specific to the work in Chapter 5, a practical consideration is access to the required hardware for training generative deep learning models of raw audio, the RAM and GPU RAM requirements and training times of which make them unsuitable to be run on readily-available and affordable consumer hardware. At the time of writing, free cloud solutions such as Google's Colab are a good and user-friendly option. The classification architecture explored in Chapter 4 and the predictive text model presented in Chapter 6 can be trained and run in reasonable time in a consumer-level CPU.

A further practical consideration that applies across all of the work presented here is whether or not the source data itself being modelled is appropriate for the intended task and its inverse consideration of whether or not the task you wish to engage with is appro-

priate for the data one is working with'. The specificities of these considerations of course vary according to data and task: in Chapter 4, it was important that the data would be representative of what the trained classifier model would be presented with in practice, and recordings of improvisation were used as training data; in Chapter 5, it became evident that timbrally diverse data and small datasets resulted in poor generated outputs, therefore it was important to mode data with consistency of timbral character and to use significant data augmentations to get better results; in Chapter 6, non-melodic recordings were inappropriate to the data representation.

The question that follows on from this is **'To what creative ends can these models be applied in the context of improvisation, instrumental practice and composition for improvisation? To what extent do they contribute to these applications and in what specific ways?'** This project has demonstrated diverse creative ends to which these models can be applied. These have been evidenced thoroughly throughout Chapters 4, 5 and 6 and the accompanying portfolio, but to recap, these include the generation of novel material for technical practice, ear training and use in improvisation (as showcased in 'Workshop I and II'), generation of compositional ideas for improvisation (as shown in the structure of 'SoloSoloDuo' and the content and structure of 'Lrrning', 'Gander', 'The Lows', 'Major Piece', 'i prompt u', 'Taps' and 'Strange Loops') and incorporation into interactive systems (as showcased in 'SoloSoloDuo', 'b.io' and 'Duo with WaveGAN').

The *extent* to which these models were found to contribute to instrumental practice, improvisation and composition varied and should be addressed one at a time. In the domain of instrumental practice, there was a clear benefit from using outputs from generative models of raw audio as material for ear training and technical practice on the instrument. In the domain of improvisation, again ear training was a tangible benefit, as was being forced to find non-perfunctory responses to generated outputs when improvising with them in an interactive loop. The absorption of the contrasting expressive characters conferred by outputs of SampleRNN and WaveGAN models into my playing was an additional benefit to both instrumental practice and improvisation. Discriminative, generative and predictive models each offered significant usefulness in the compositional process: the process of training and interacting with a classifier defined the structure of 'SoloSoloDuo'; using curated outputs from generative models as the basis for compositions offered a way to effectively outsource aspects of the compositional process, crucially in a way that did not undermine a sense of authorship or achievement; similarly predictive modelling of text-based representations of pitch created a fast, efficient workflow that led to three additional compositions being written in a relatively short space of time. Additionally, the way in which generated outputs were used within the audiovisual project 'Gandering' suggests that experimentation with these models can also open up alternative domains of creative work.

The third question, **'How do specific classes of model architecture 'behave' musically with respect to the data on which they are trained?'**, needs to be answered on a case-by-case basis.

CNN

A classifier's behaviour is, on the surface, straightforward to define owing to its simplicity of function - all it does is analyse an unseen input and output a predicted class or category to which the input most likely belongs based on the pre-categorised data on which it was trained, often accompanied by the degree of confidence in its prediction. More interesting is the question of how it behaves with respect to musical data and whether, in the case of the convolutional neural network used in this thesis, it is a suitable tool

for audio applications. Classification of audio spectrograms is fairly commonplace in audio research and industry applications, examples being the popular birdsong identification app 'BirdNET', Algonaut's AI-augmented drum sequencer software 'Atlas' and Musiio's genre recognition technology: this approach clearly 'works' quite well. But convolutional models per se are explicitly designed to identify features of natural images, not spectrograms, making it not straightforward to reason about what is actually being learned. Convolutional models are also intensive for real-time applications even when stripped back as I did for Chapter 4. I circumvented this in 'SoloSoloDuo' further by only periodically recording the input stream, but other practitioners may well want a more immediately responsive application.

For these reasons it is my feeling that the lower level approach favoured by Flucoma in their machine learning tooling, in which classification is handled by, for example, a multilayer perceptron, which despite its name is a much simpler model architecture than any convolutional model, and input representations with lower computational overhead such as MFCCs, spectral centroids and such are favoured, are going to be a more appropriate choice for the majority of practitioners.

WaveGAN

The most striking aspects of WaveGAN's behaviour when modelling and generating audio signals are its ability to grasp musical content, its speed of sample generation and, unfortunately, the noisiness of the generated samples.

Its ability to model content is a clear function of its discriminator-generator architecture and adversarial training process: the generator's effective priority is to create plausible content. This 'mode-seeking' behaviour has downsides: given a diverse dataset, such as 'Timbral Improvisation' and the ones I externally sourced during the 'Gandering' project, the generator can end up simply seeking the easiest path to plausibility, which can result in swathes of the dataset being ignored and the generator converging on a small number of areas of the dataset; conversely, if the dataset contains repetitions of material as is the case with the 'Tone Rows' dataset, the learning task becomes too easy for a GAN and the generated samples, while of improved fidelity, are too often direct regurgitations of the dataset. This might be fine if you only wish to generate, say, one-shot samples with a bit of timbral variation, but for generation of novel phrases it doesn't cut it. The implication for creating datasets for modelling with WaveGAN - with a view to generating novel melodic phrases - is that a balance needs to be struck between variety (or avoidance of repetition) and cohesiveness. For this reason, if data augmentation is necessary, use of high-quality pitch shifting is more appropriate. When this balance is achieved, as was the case with the 'Melodic Improvisation' and 'G Major' datasets, the results are more novel and compelling.

On a more purely technical level, WaveGAN's speed of sample generation means it can be used in real-time interactive contexts, assuming the use of a GPU (sample generation with WaveGAN is a little slower but still reasonably fast in a CPU), as shown in 'Duo with WaveGAN'. This affordance has already been exploited in commercial audio plugin 'Tensorpunk'. I believe that for this combination of generation speed and ability to model content in novel ways, WaveGAN is an underrated creative tool compared with SampleRNN or RAVE; I suspect reasons for its relative lack of popularity are the lack of availability of an easy-to-use implementation (the reference version is written in a legacy version of Tensorflow that is no longer supported in Googe Colab, usually a first port-of-call for training models in the cloud) and the noisiness of its samples.

SampleRNN

I find SampleRNN's defining behaviours to be its capacity for novelty (showcased most clearly in the Workshop pieces and in 'Lrrning' and 'The Lows') and its ability to model the complexities of instrumental timbre to a realistic-sounding degree (heard most clearly in 'Fake Lrrning'). While it naturally models instrumental timbre easily, it struggles to model content; for this reason it clearly benefits from repetition of content in the dataset and from data augmentation through the use of time stretching. The benefits of this can again be heard in 'Fake Lrrning'. Timbral was realism was further improved by keeping the sample rate at 44100Hz as opposed to downsampling to 22050Hz (again, the benefit of this can be heard in 'Fake Lrrning'), with the trade-off that doing so increased training time significantly.

Char-RNN

Char-RNN exhibits a similar 'mode-covering' behaviour to SampleRNN, seeking a probability distribution that covers the full span of the dataset's contents. This was borne out in the outputs of the model I trained on a composite dataset consisting of all bar 'Timbral Improvisation' and can be clearly seen in the phrases that make up 'i prompt u' (suggestive of a range of inputs from tonal to atonal), 'Taps' (clearly derived from the 'Tone Rows' dataset) and 'Strange Loops' (clearly suggestive of tonal centers). Compared with the language-focused original version of this model architecture, the drastically reduced 'vocabulary' (limited to 12 characters - numbers 0 to 9 plus commas and spaces) and strictly uniform data format simplified the learning task considerably. Strong loss statistics, encouraging training-in-progress outputs and clear modelling of aspects of the data shown in generated outputs suggest this model architecture is well-suited for modelling musical information, better than it is for language modelling; this would also be borne out, of course, by Sturm et al's appropriation of the architecture for Folk-RNN [1]. More generally, Char-RNN is fast to train and predict, and can be trained and inferred upon in the CPU, making it a technologically accessible option for more practitioners.

7.3 Core Themes

In this section I enumerate and reflect upon themes and ideas that have recurred consistently throughout this project.

7.3.1 Dataset Creation and Manipulation

In Chapter 3, I presented and discussed several original audio datasets. Following the statistical modelling and creative work in Chapters 4-6, the work in Chapter 3 retrospectively reveals some considerations when creating datasets for AI modelling. These concerns tend to be specific to the class of model architecture and/or the intended application or outcome.

For example, the work on audio classification in Chapter 4 revealed a need for the dataset to be both representative of the unseen material it would classify and at least somewhat robust to natural variation such as changes to the local acoustic, changes of reed, variations in ones physicality (the same saxophone player will sound subtly different from one day to the next depending on how they feel physically) and microphone position and gain. My solution to the inherent sensitivity of CQT spectrograms to these subtle differences, differences which affect model performance, was to use salience modelling as a noise reduction tool. Doing so allowed me to use a model trained on data recorded 3 years

[1] Sturm, Bob et al, 'Folk Music Style Modelling by Recurrent Neural Networks with Long Short Term Memory Units'.

prior to classify inputs played on a different saxophone and reed-mouthpiece combination (I changed instrument in mid-2022). I make no claims of empirical proof here, but it is at least noteworthy that I achieved tolerable classification performance that enabled me to make version 2 of 'SoloSoloDuo' on this basis.

The work on applied generative modelling in Chapter 5 emphasised the need for clarity and strength of recorded signal when creating datasets for this purpose and the importance of timbral cohesion in the dataset; this finding held true for both WaveGAN and SampleRNN models. Beyond this finding, best results with SampleRNN were achieved when the raw dataset contained an amount of repetition of musical content (as was the case with the 'Tone Rows' and 'Lower Register' datasets) and had been augmented in size with additional time-stretched copies of itself (effectively providing more of this repetition of content without actually duplicating the data). With WaveGAN it proved necessary to avoid repetition lest the generated samples simply replicate the dataset; here it was better to emphasise *consistency of character* while avoiding actual repetition, as is the case with the 'G Major' and 'Melodic Improvisation' datasets.

When creating the dataset used in Chapter 6, experiments with data augmentation in the symbolic domain did not seem to improve outcomes: the generated material tended to be more plausible from models trained on data to which no augmentations had been applied, whereas I felt that the generated outputs from models trained on augmented data lost their connection to the original data (which even in the best case was already somewhat tenuous). However, the reductive nature of the data preprocessing meant that it was necessary to combine all datasets bar 'Timbral Improvisation' to simply have sufficient data to train the model. It was then interesting to inspect the raw generated outputs to try and parse which outputs were influenced by which parts of the dataset, with the 'Tone Rows' dataset proving most influential, presumably again due to the amount of repetition of content it contains. This is in keeping with SampleRNN's fondness for repetition of content and no doubt a function of RNNs' *mode-covering* behaviour.

7.3.2 Data Ownership = Creative Authorship; AI as Personalised Tool

An important aspect of this work is the fact that all the data on which the models used for creative work were trained was self-authored and created specifically for this project. New high-profile developments in AI such as large language-prompted generative models are trained on large bodies of externally-sourced data and as such they offer very little in the way of authorship to the creative musician. This is to say nothing of the often insufficient transparency around to how this data was sourced and whether artists and copyright holders were compensated or even consulted. While recording at length, preprocessing the resulting audio data and modelling it with deep learning are obviously much more onerous tasks than entering a text prompt to an API, the reward is a highly personalised and reusable creative tool, one that confers, as I hope has been made very clear through this work, multiple potential applications. Engagement with deep learning models at the level of training them on one's own data equips the practitioner with useful tools for the generation of raw materials to use in their practice in the form of the trained model; proficiency in the process of creating new datasets and training new models equips the practitioner to more easily create more new tools. Such tools carry a particular sense of ownership and authorship having been created through modelling one's own data.

Lacking a software development background, the idea of developing my own tools for musical creativity would have seemed fanciful before my engagement with deep learning. By learning about how these model architectures work and behave, practitioners can expand their practice outwards as I have done. The work in this thesis makes such an engagement a more accessible prospect.

7.3.3 AI as Teacher

A recurrent theme throughout this work has been the notion of AI as teacher. For example, development of the interactive pieces 'SoloSoloDuo', 'b.io' and 'Duo with WaveGAN' felt akin to practicing with the interactive equivalent of a jazz playalong. Doing so felt difficult, but had the benefit of forcing me out of my comfort zone in numerous ways: by thwarting any attempts on my part at linear development, setting up a more interruptive dynamic than I was initially comfortable with and sharpening my active listening abilities.

This theme continued in Chapter 5 with the Workshop pieces: while technically demanding, these represented a more comfortable and enjoyable practicing situation than the interactive duets, demanding that I simply learn to execute phrases from the generated samples by ear. While I would consider the duet pieces to be more successful as creative outputs than the Workshop pieces, I felt great enjoyment and benefit to my technique from the process of learning the phrases and from further private engagement with the same. This application of generative models of raw audio feels significant to me.

A less obvious way in which deep learning models can serve as a teacher is in the information they return about our datasets and music. Engagement with the process of training generative models of raw audio is almost self-improving, since the generated samples tend to exaggerate any flaws present in the dataset. On a musical level, the process of creating the datasets and modelling them led to re-evaluation of the distinctions between them. I have questioned my long-held belief that there was an obvious distinction between the 'timbral' and 'melodic' aspects of my practice: while this distinction forms the structural basis of SoloSoloDuo, developing the duo section encouraged me to combine these areas more liberally; this loosening of my boundaries can also be heard in the improvisations in 'Gander' and 'The Lows' and seen in the compositional structures of 'Strange Loops' and 'Taps'. The recogisable influence of the Tone Rows dataset on the raw Char-RNN outputs in Chapter 6 led me to treat the 12-tone *style* (as distinct from actually striving to improvise according to strict 12-tone rules) as a distinct structural region of 'i prompt u'; similarly pitch groupings more suggestive of tonal centers occupied their own distinct area of the same piece and of 'Strange Loops'.

7.3.4 AI as Creative Assistant

The use of the generative models discussed in Chapter 5 and the predictive model in Chapter 6 as providers of substantial source material for compositional ideas and improvisational 'prompts' has at multiple points throughout this work raised the notion of deep learning models acting as a creative assistant. This can be seen most clearly in the compositions for solo improvisation in Chapter 5 'Gander', 'Lrrning', 'The Lows' and 'Gander', which I can safely say contain ideas I would not have written left to my own devices! While the connections between the generated notations and the final composed outputs in Chapter 6 ('i prompt u', 'Strange Loops' and 'Taps') are slightly less explicit, with some additional inputs such as effects, inferences about tonal centers and associations with multiphonic pitches being brought to bear, they are still a further example of this notion. Again, I can say with certainty that these pieces contain ideas that I would not have come up with without the model outputs. Less explicitly again, without the trained classifier used in Chapter 4, the idea for the structure of 'SoloSoloDuo' might never have been considered, although it was borne as much from the entire process of training the classifier as it was from the trained model itself.

The use of deep learning as a source of material for practicing, as represented in the pieces 'Workshop I and II' in Chapter 5, seems to me, on reflection now, an especially

abundant creative application of generative models. While I used to write my own exercises or adapt my exercises from sources such as Slonimsky's 'Thesaurus of Scales and Melodic Patterns'[2], I feel a deeper sense of ownership of the outputs generated by deep learning models and a greater sense of enjoyment and fun when learning to execute them on mhy instrument.

7.3.5 Associativity

Another theme that has arisen throughout this work has been associativity. At multiple points throughout this thesis, particularly during disussion of creative outputs, it has been noted that association between a trained model's behaviour or output with a pre-existing musical idea or entity was a driver of bringing these abstracted data models to life in the creative realm. This notion of associativity differs slightly from psychological models in which it is connected to the idea of semantic distance, the idea being that the more semantically distant the association, the more creatively interesting the association is[3]. In my work these associations have been highly domain-specific, such as associating generated audio samples with the work of an existing composer-improviser, generated pitch cells with tonal centers, one improvised idea with the next, and the current improvised idea to a point in a compositional structure. These associations have felt a necessary counterbalance to the emphasis on process required for creating the deep learning models that underpin this work.

7.3.6 Explainability

While not explicitly referred to during this thesis, the notion of *explainability* of AI has been present throughout. As with associativity, the notion of explainability this thesis engages with differs from existing research definitions; for example, through DARPA's Explainable Artificial Intelligence Program, Gunning et al propose that Explainable AI should be able to explain to its user why it made a certain choice [4]. The version of explainability I am proposing practice-researchers and musicians consider when engaging with AI stops short of this ambition (it is worth noting that ChatGPT[5] was designed with this feature in mind, but in reality its explanations fall short of what users might reasonably have in mind!) but is a valuable and realistically attainable framework for an intelligent engagement that delves deeper than the end-user paradigm. I propose the following assumptions:

- An AI's output is more explainable the more familiar you are with the contents of the dataset;
- An AI's output is more explainable when you understand the expected 'behaviours' of the model architecture you are working with;
- An AI's output is more explainable when you can reason about *why* these behaviours are expected.

This feels important for trying to engage creative practitioners with deep learning; approaching this technology with an 'end-user' mindset where the practitioner expects magic to happen without putting the necessary thought and effort into dataset content, choice of model architecture, model size with respect to dataset size (or vice versa) and choice of augmentation technique with respect to data representation and model architecture,

[2]Slonimsky, Nicolas, *'Thesaurus of Scales and Melodic Patterns'*

[3]Kennet, Yoed N., 'What can quantitative measures of semantic distance tell us about creativity?' in *Current Opinion in Behavioral Sciences* Vol. 27, Jun 2019, 11-16

[4]Gunning, David, David W. Aha, 'DARPA's Explainable Artifical Intelligence Program', AI Magazine, 40(2), https://doi.org/10.1609/aimag.v40i2.2850, 44-58.

[5]OpenAI, 'GPT-4 Technical Report, unpublished paper, Mar 2023, arXiv:2303.08774.

then efforts are likely to fail and the user will likely disengage. This in turn would add to the narrative that only people with the necessary technical expertise can engage with AI, which I emphatically do not believe to be the case. An intelligent approach that takes into account the above prerequisites is more likely to lead to a personalised artistic engagement with AI and is more likely to succeed.

7.4 Future Work

An approach to creation of datasets for classification I had not considered when undertaking this work is a *bag-of-frames* approach: by using overlapping windows rather than fixed-length, non-overlapping consecutive audio segments as the basis for classification data it would be possible to considerably embiggen the dataset, give the classifier a greater range of plausible input examples and hopefully improve classification performance.

When recording solo saxophone datasets for generative modelling in future I will certainly use a dual-microphone setup in order to capture a fuller impression of those regions of the instrument than a single clip microphone is able to. I will also consider the compatibility between the dataset and model architecture carefully, effectively creating bespoke datasets intended for use with specific model architectures. I am especially keen to further develop my use of SampleRNN in this regard, creating new datasets with that model architecture in mind and creating models of them. I also intend to begin creating datasets for modelling with the RAVE and Catch-A-Waveform architectures. The former is attractive for its state-of-the-art output quality and the latter for its ability to generate variations on small datasets. I am especially attracted to the idea of an album of 'fake' saxophone solo pieces generated from SampleRNN and other models. I will also continue to use generated samples as sources of material for practice, composition and improvisation. I also see applications of audio generation in the field of music education research: recording instrumental practice sessions, training models of the resulting audio data and using it to generate new material for instrumental practice and new ideas for composition and improvisation.

I am adapting the SuperCollider patches created for the work in Chapter 6 for live performance with two distinct setups - one with laptop, audio interface and USB foot pedals, the other a more compact setup with the Bela embedded platform[6], foot pedals and touch sensors. I very much intend to take this work into the live performance space. I also intend to adapt 'SoloSoloDuo' for live performance - ideally this would take the form of porting of some of the Python scripts (for phrase-based segmentation in particular) to SuperCollider for ease of live use; doing so would also necessitate use of the Flucoma[7] toolkit for the classification aspect of the piece. It has recently been proved possible to run this toolkit on Bela hardware[8].

Another adaptation of this work to live performance I am keen to investigate is converting the visualisations used for online presentation of 'Gandering 1', 'SoloSoloDuo' and 'b.io' for real-time purposes. Since real-time traversal and visual rendering of a GAN's weight space would be computationally very expensive in an already compute-intensive domain, image sequences created through linear interpolation of the weight space would be pre-generated; the smooth speed-up and slow-down effects created by the Lucid Sonic

[6]McPherson, Andrew P. and Victor Zappi, 'An Environment for Submillisecond-Latency Audio and Sensor Processing on BeagleBone Black' (conference paper, AES: 138th Convention of the Audio Engineering Society, Sofitel Victoria Hotel, Warsaw, Poland, May 7–10, 2015).

[7]Tremblay, Pierre Alexandre, Owen Green, Gerard Roma, Alexander Harker, 'From collections to corpora: Exploring sounds through fluid decomposition' (conference paper, ICMC 2019: 45th International Computer Music Conference, Elmer Holmes Bobst Library, MORE, New York University, USA, Jun 16-23, 2019.

[8]Armitage, James, 'flucoma-bela', Feb 27, 2023, https://github.com/jarmitage/flucoma-bela

Dreams program would be mimicked by skipping images in the sequence according to real-time amplitude or other audio analysis of the live signal. I have begun to implement this work in Python though it may be necessary to use TouchDesigner or similar video processing software for the end result to be suitable for live performances.

I will continue to use my Char-RNN-based models as a means of fast generation of musical ideas. To this end I would like to experiment with adding artificially-created data and externally-sourced data to my own. I would also like to develop an effective data augmentation technique for this context and experiment with adding symbolic representations of a greater number of musical parameters than is currently the case.

Moving slightly away from work that direct flows from this thesis and towards other machine learning tasks for audio, I am especially interested in the possibilities offered by real-time timbre transfer, as made possible by Google Magenta's DDSP[9] and RAVE[10], and audio source separation, as exemplified by the Demucs[11] and Wave-U-Net[12] architectures. I see potential applications of source separation in particular in the field of music education research: using source separation to remove specific instruments from existing tracks to provide students with customised backing tracks, for example, and usig separation as an aid to transcription for instrumental study and analysis.

[9]Engel et al, 'DDSP: Differentiable Digital Signal Processing'.

[10]Caillon et al, 'RAVE: A variational autoencoder for fast and high-quality neural audio synthesis'.

[11]Rouard et al, 'Hybrid Transformers for Music Source Separation'.

[12]Stoller, Daniel, Sebastian Ewert, Simon Dixon, 'Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation', paper presented at ISMIR 2018: 19th International Society for Music Information Retrieval Conference, IRCAM, Paris, Sep 23-27, 2018, arXiv:1806.03185.

Bibliography

[1] Abadi, Martin, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 'TensorFlow: Large-scale machine learning on heterogeneous systems'. 2015. Computer software. https://tensorflow.org.

[2] Alafriz, Mikael. 'Introducing "Lucid Sonic Dreams': Sync GAN Art to Music with a Few Lines of Python Code!', *Towards Data Science*, Mar 13, 2021, https://towardsdatascience.com/introducing-lucid-sonic-dreams-sync-gan-art-to-music-with-a-few-lines-of-python-code-b04f88722de1

[3] Armitage, James. 'flucoma-bela'. Code repository. Feb 27, 2023. https://github.com/jarmitage/flucoma-bela.

[4] BBC Sound Effects. Online data repository. https://www.sound-effects.bbcrewind.co.uk.

[5] Bittner, Rachel, Brian McFee, Justin Salamon, Peter Li and Juan P. Bello. 'Deep Salience Representations for F0 Estimation in Polyphonic Music'. Proceedings of ISMIR 2017: 18th International Society for Music Information Retrieval Conference, Suzhou, China. Oct 23-27, 2017.

[6] Hamiet Bluiett, Jr. *Birthright: A Solo Blues Concert*. India Navigation, 1977. LP.

[7] Boersma, Paul and David Weenink. *Praat*, version 6.3.09. Computer software. https://www.fon.hum.uva.nl/praat/

[8] Anthony Braxton. *For Alto*. Delmark Records, 1971. LP.

[9] John Butcher. *Bell Trove Spools*. Northern Spy, 2012. CD and Digital. https://johnbutcher.bandcamp.com/album/bell-trove-spools-2.

[10] Bute, Mary-Ellen. 'Dada'. YouTube video. Iploaded by 'Musica Visual', Jun 13 2022. https://youtu.be/ihhJxrY3Vig?si=GQR4H5oOsBJ0kIfO.

[11] Cahuantzi, Roberto, Xinye Chen and Stefan Güttel. 'A comparison of LSTM and GRU networks for learning symbolic sequences'. Unpublished paper, Sep 2019. arXiv:2107.02248.

[12] Caillon, Anton and Philippe Esling. 'RAVE: A variational autoencoder for fast and high-quality neural audio synthesis'. Unpublished paper. ArXiv:2111.05011.

[13] Baptiste Caramiaux, Montecchio, Nicola, Atau Tanaka, Atau and Bevilacqua, Frédéric. 'Adaptive Gesture Recognition with Variation Estimation for Interactive Systems'. In *ACM Transactions on Interactive Intelligent Systems*, Volume 4, Issue 4. 1–34.

[14] Carr, CJ and Zach Zukowski. 'Generating Albums with SampleRNN to Imitate Metal, Rock, and Punk Bands'. Paper presented at MUME 2018 : The 6th International Workshop on Musical Metacreation. University of Salamanca, Spain. Jun 25-26, 2018.

[15] Carr, C.J. and Zack Zukowski. 'OUTERHELIOS - Free Jazz - neural generated - Coltrane'. YouTube video. Posted by Dadabots, Jan 28, 2020. https://youtu.be/C0dOin79Hm0.

[16] Carr, C.J. and Zack Zukowski. 'RELENTLESS DOPPELGANGER'. YouTube video. Posted by Dadabots, Sep 4, 2019. https://www.youtube.com/live/MwtVkPKx3RA?feature=share.

[17] Creswell, Antonia, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta and Anil A. Bharath. 'Generative Adversarial Networks: An Overview'. *IEEE Signal Processing Magazine* 35, Issue:1. IEEE, Jan 2018. 53-65.

[18] Cheuk, Kin Wai, Hans Anderson, Kat Agres and Dorien Herremans. 'nnAudio: An on-the-Fly GPU Audio to Spectrogram Conversion Toolbox Using 1D Convolutional Neural Networks' in *IEEE* Vol. 8. Aug 24, 2020. 161981-162003.

[19] Guilherme Coelho. 'AIMC 2021 | DDSP études for Tenor Saxophone & Violin'. Vimeo video. Posted by Guilherme Coelho, Jul 2021. https://vimeo.com/677268564/709378c6cf.

[20] Collins, Nick, 'Using a Pitch Detector for Onset Detection'. Paper presented at ISMIR 2005: 6th International Conference on Music Information Retrieval, London, UK. Sep 11-15, 2005.

[21] Collins, Tom, Alex Gonzalez, Jemily Rime, Jack McNeill and Mark Hanslip. 'Nobody New'. YouTube video. Uploaded by 'G-Zone', Jun 16 2022. https://youtu.be/PwBpW6LYue8.

[22] Tom Collins, Hanslip, Mark, Maloney, Liam, Quek, Lynette, Rime, Jemily, Zongyu (Alex) Yin. 'Circus'. Entry to AI Song Contest 2021. https://www.aisongcontest.com/participants/theelephantsandthe-2021.

[23] Dean, Roger T. and Jamie Forth. 'Towards a Deep Improviser: a prototype deep learning post-tonal free music generator'. In *Neural Computing and Applications* 32, 2020. 969–979.

[24] Dhariwal, Prafulla, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford & Ilya Sutskever. 'Jukebox: A Generative Model for Music'. Unpublished Paper. ArXiv:2005.00341.

[25] Dieleman, Sander. 'Generating Music in the Waveform Domain', *Latest Posts - Sander Dieleman*, March 24, 2020, https://sander.ai/2020/03/24/audio-generation.html.

[26] Jorrit Dijkstra. *30 Micro-Stems*. TryTone Records, 2002. CD and Digital. https://jorritdijkstra.bandcamp.com/album/30-micro-stems.

[27] Jorrit Dijsktra. *Never Odd or Even*. Driff Records, 2015. CD and Digital. https://jorritdijkstra.bandcamp.com/album/never-odd-or-even.

[28] Eric Dolphy. 'Miss Ann' on *Far Cry*. New Jazz Records, 1962.

[29] Donahue, Chris. ‘wavegan’. Online code repository. Posted by ‘chrisdonahue’, Apr 2018. https://github.com/chrisdonahue/wavegan/blob/master/train_wavegan.py

[30] Donahue, Chris, Julian McAuley and Miller Puckette. ‘Adversarial Audio Synthesis’. Paper presented at ICLR 2019: Seventh International Conference on Learning Representations. Ernest N. Morial Convention Center, New Orleans. May 6-9, 2019.

[31] Douwes, Constance, Philippe Esling and Jean-Pierre Briot. ‘Energy Consumption of Deep Generative Audio Models’. Proceedings of ICASSP 2022: EEE International Conference on Acoustics, Speech and Signal Processing. Sand Expo and Convention Centre, Singapore, 22-27 May 2022.

[32] Unknown author. Webpage. Posted May 2022. https://www.dpamicrophones.com/mic-university/source-dependent-proximity-effect-in-microphones

[33] Eck, Douglas and Jurgen Schmidhuber. ‘A First Look at Music Composition using LSTM Recurrent Neural Networks’. Technical report. Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale, Mar 2002.

[34] Ellery Eskelin. *Solo Live at Snug’s*. hatOLOGY, 2015. CD.

[35] el-Raeby, Mostafa. ‘pytorch-wavegan’. Code repository, Jun 11, 2019. https://github.com/mostafaelaraby/wavegan-pytorch.

[36] Engel, Jesse, Lamtharn (Hanoi) Hantrakul, Chenjie Gu, Adam Roberts. ‘DDSP: Differentiable Digital Signal Processing’. Paper presented at ICLR 2020: Eighth International Conference on Learning Representations, Online. Apr 26 - May 1, 2020.

[37] Fell, Mark. *Structure and Synthesis: The Anatomy of Practice*. Urbanomic, 2021. 20.

[38] Floros Floridis. *F.L.O.R.O. IV - Future Learning of Radical Options*. To Pikap Records, 2019. Vinyl and Digital. https://topikaprecords.bandcamp.com/album/f-l-o-r-o-iv-future-learning-of-radical-options.

[39] Franceschelli, Giorgio and Mirco Musolesi. ‘Copyright in generative deep learning’. *Data and Policy* 4. Cambridge University Press, 25 May 2022.

[40] Sam Gendel. *Pass If Music*. Leaving Records, 2018. Cassette and Digital. https://leavingrecords.bandcamp.com/album/pass-if-music.

[41] Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville and Yoshua Bengio, ‘Generative Adversarial Nets’. Paper presented at NIPS 2014: 28th Conference on Neural Information Processing Systems, Palais des Congrès de Montréal, Canada. Dec 8-13, 2014.

[42] Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville and Yoshua Bengio. ‘Generative adversarial networks’. *Communications of the ACM* 63, No. 11. ACM Digital Library, Nov 2020. 139-144.

[43] Unknown author. ‘Say hello to the "Hello, World" of machine learning’. Webpage. https://developers.google.com/codelabs/tensorflow-1-helloworld.

[44] Unknown author. Webpage. https://colab.google/.

[45] Greshler, Gal, Tamar Rott Shaham and Tomer Michaeli. ‘Catch-A-Waveform: Learning to Generate Audio from a Single Short Example’, *DeepAI*. Blog post, Jun 11, 2021. https://deepai.org/publication/catch-a-waveform-learning-to-generate-audio-from-a-single-short-example.

[46] Griffin, D.W and J. S. Lim. 'Signal estimation from modified short-time Fourier transform' in *ASSP* 32, no.2. EEE Trans., Apr, 1984. 236-243.

[47] Grimes. *elf.tech*. Web service. https://elf.tech/connect.

[48] Gunning, David and David W. Aha. 'DARPA's Explainable Artifical Intelligence Program' in *AI Magazine* Vol. 40(2). June 2019. https://doi.org/10.1609/aimag.v40i2.2850. 44-58.

[49] Hanslip, Mark. 'b.io'. Webpage. Dec 6 2022. https://ismir2022program.ismir.net/music_346.html

[50] Hanslip, Mark. 'Gandering 1' in *mjf digital originals: Gandering 1 - Mark Hanslip*. YouTube video, 3:48. Posted by 'Manchester Jazz Festival'. Jan 18 2022. https://youtu.be/5dIxUWNGndc.

[51] Hanslip, Mark. 'SoloSoloDuo' in *Interaction & Improvisation #2*. YouTube video, 34:43. Posted by 'AI Music Creativity 2022'. Sep 20 2022. https://youtu.be/Nin8GIIZW-4.

[52] Hanslip, Mark. 'mjf digital originals: Gandering 1 - Mark Hanslip'. YouTube video. Uploaded by 'Manchester Jazz Festival', Jan 18 2022. https://youtu.be/5dIxUWNGndc.

[53] Mark Hanslip. 'Monobrow', *Revival Room*. EfPi Records, 2021. CD and Digital. https://revivalroom.bandcamp.com/album/revival-room.

[54] Mark Hanslip. 'Spiders', *The Adding Machine*. Babel Label, 2011. CD and Digital. https://babel-label.bandcamp.com/album/the-adding-machine.

[55] Mark Hanslip. 'Spiders', *Outhouse*. Babel Label, 2008. CD and Digital. https://babel-label.bandcamp.com/album/outhouse.

[56] Hantrakul, Lamtharn and Zachary Kondak. 'GestureRNN: A neural gesture system for the Roli Lightpad Bloc'. In Proceedings of NIME 2018: New Interfaces for Musical Expression, Campus of Virginia Tech, Blacksburg, Virginia, USA, June 3-6 2018.

[57] Eddie Harris. 'Freedom Jazz Dance' on *The In Sound*. Atlantic Records, 1965.

[58] Harris, Louise. *Composing Audiovisually: Perspectives on Audiovisual Practices and Relationships*. Routledge, 2022. 43.

[59] Coleman Hawkins. 'Picasso', *The Verve Story - Disc One: 1944-53*. Verve Records, 1994. CD.

[60] He, Kaiming, Xiangyu Zhang, Shaoqing Ren and Jian Sun. 'Deep Residual Learning for Image Recognition'. Unpublished paper. ArXiv:1512.03385.

[61] Hofstadter, Douglas. *Goedel, Escher, Bach: An Eternal Golden Braid (Twentieth Anniversary Edition)*. Basic Books, 1999. 8.

[62] Emily Howard. 'shield for String Quartet'. Musical score, 2022. https://www.editionpeters.com/product/shield/ep73579.

[63] Huang, Gao, Zhuang Liu, Laurens van der Maaten and Kilian Q. Weinberger. 'Densely Connected Convolutional Networks'. Unpublished paper. arXiv:1608.06993.

[64] Huzaifah, Muhammad. 'Comparison of Time-Frequency Representations for Environmental Sound Classification using Convolutional Neural Networks'. Unpublished paper. ArXiv:1706.07156.

[65] Jadoul, Jannick, Bill Thompson and Jan de Boer. 'Introducing Parselmouth: A Python Interface to Praat'. *Journal of Phonetics* 71 (2018): 1-15.

[66] Kahl, Stefan, Connor M. Wood, Maximilian Eibl and Holger Klinck,'BirdNET: A deep learning solution for avian diversity monitoring' in *Ecological Informatics* 61, Mar 2021, ScienceDirect, 101236.

[67] Karpathy, Andrej. 'The Unreasonable Effectiveness of Recurrent Neural Networks', *Andrej Karpathy Blog*, May 21 2015, http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

[68] Karras, Tero, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen and Timo Aila. 'Training Generative Adversarial Networks with Limited Data'. Unpublished paper. ArXiv:2006.06676.

[69] Kastrup, David, Werner Lemberg, Han-Wen Nienhuys, Jan Nieuwenhuizen, Carl Sorensen, Janek Warchoł, et al. *Lilypond*, version 2.24.1. Computer software. https://lilypond.org

[70] Kennet, Yoed N.. 'What can quantitative measures of semantic distance tell us about creativity?' in *Current Opinion in Behavioral Sciences* Vol. 27. Jun 2019. 11-16.

[71] Kientzy, Daniel. *Les sons multiples aux saxophones : pour saxophones sopranino, soprano, alto, ténor et baryton'*. Salabert, 1982.

[72] Kim, Joon Wook, Justin Salamon, Peter Li, Juan Pablo Bello. 'CREPE: A Convolutional Representation for Pitch Estimation'. Paper presented at ICASSP 2018: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, Calgary TELUS Convention Centre, Alberta, Canada. Apr 15-20, 2018.

[73] Kosakowski, Piotre, Katarzina Kanzka, Joachim Rishaug. 'samplernn-pytorch'. Code repository, Nov 19, 2017. https://github.com/deepsound-project/samplernn-pytorch.

[74] Kwan Lam, Siu, Antoine Pitrou and Stanley Seibert. 'Numba: a LLVM-based Python JIT compiler'. LLVM '15: Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC. ACM Digital Library, Nov 2015. Article 7, 1-6.

[75] Lacoste, Alexandre, Alexandra Luccioni, Victor Schmidt, Thomas Dandres. 'Quantifying the Carbon Emissions of Machine Learning'. Unpublished paper. ArXiv:1910.09700.

[76] Steve Lacy. *Clinkers*. HatHut, 1978. CD.

[77] Steve Lacy. *November*. Intakt Records, 2010. CD and Digital. https://steve-lacy.bandcamp.com/album/novemberjorrit.

[78] Robert Laidlow. 'Robert Laidlow (2022): Silicon'. YouTube video. Posted by 'RNCM PRiSM', Mar 18 2023. https://youtu.be/3xmpywK0ACA.

[79] David Liebman. *Colors*. Hatology, 2003. CD.

[80] Lipton, Zachary C., John Berkowitz and Charles Elkan. 'A Critical Review of Recurrent Neural Networks for Sequence Learning'. Unpublished paper, May 2015. arXiv:1506.00019.

[81] Liu, Pei, Xuemin Wang, Chao Xiang and Weiye Meng. 'A Survey of Text Data Augmentation'. *2020 International Conference on Computer Communication and Network Security (CCNS)*. IEEE, 2020.

[82] Cecilia Lopez & Ingrid Laubrock. *Maromas*. Relative Pitch Records, 2023. CD and Digital. https://relativepitchrecords.bandcamp.com/album/maromas.

[83] Lye, Len. 'Free Radicals'. YouTube video. Uploaded by 'Musica Visual', Jan 19 2023. https://youtu.be/t5ych1ikDfI?si=CUlUEZXmYmOlFpg2.

[84] Marafioti, Andrés, Nicki Holighaus, Nathanaël Perraudin and Piotr Majdak. 'Adversarial Generation of Time-Frequency Features with Application in Audio Synthesis'. Paper presented at ICML 2019: 36th International Conference on Machine Learning, Long Beach Convention Centre, Long Beach, California, USA. Jun 9-15, 2019.

[85] Charles P. Martin and Toressen, Jim. 'An Interactive Musical Prediction System with Mixture Density Recurrent Neural Networks'. In Proceedings of NIME 2019: New Interfaces for Musical Expression, Universidade Federal do Rio Grande do Sul Porto Alegre, Brazil, 3-6 June, 2019.

[86] McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 'Librosa: Audio and Music Signal Analysis in Python.'. Proceedings of the 14th Python in Science Conference, pp. 18-25. 2015.

[87] Bill McHenry. *Solo*. Underpool, 2018. CD.

[88] McLaren, Norman. 'Blinkity Blink'. YouTube video. Uploaded by 'NFB', May 17 2015. https://youtu.be/q3YeWgUgPHM?si=5gcnmCgy5WBRVPgU.

[89] McLeod, Philip and Geoff Wyvill. 'A Smarter Way to Find Pitch'. Proceedings of ICMC 2005: 31st International Computer Music Conference 2005, Barcelona, Spain, Sep 4-10 2005. 138-141.

[90] McPherson, Andrew P. and Victor Zappi. 'An Environment for Submillisecond-Latency Audio and Sensor Processing on BeagleBone Black'. Paper presented at AES: 138th Convention of the Audio Engineering Society, Sofitel Victoria Hotel, Warsaw, Poland. May 7–10, 2015.

[91] Mehri, Soroush, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville and Yoshua Bengio. 'SampleRNN: An Unconditional End-to-End Neural Audio Generation Model'. Paper presented at ICLR 2017: Fifth International Conference on Learning Representations. Palais des Congrès Neptune, Toulon, France. April 24-26, 2017.

[92] Melen, Christopher and Sam Salem. 'PRiSM SampleRNN', Jan 12, 2020, https://github.com/rncm-prism/prism-samplernn.

[93] Ted Moore & Kyle Hutchins. 'shadow'. YouTube video. Posted by 'Ted Moore', 10 June 2021. https://youtu.be/CgALDzMYcbc?si=QfCjxILzPrl9lJnq.

[94] OpenAI. 'GPT-4 Technical Report. Unpublished paper, Mar 2023. arXiv:2303.08774.

[95] Guillaume Orti & Olivier Sens. *Reverse*. Quoi de neuf docteur, 2009. CD.

[96] Evan Parker. *Monoceros*. Incus Records, 1978. LP (since reissued).

[97] Evan Parker. *Chicago Solo*. Okkadisk, 1997. CD.

[98] Evan Parker, 'Evan Parker'. Interviewed by Frances-Marie Uitti. *Contemporary Music Review* 25, Issues 5-6. Oct-Dec 2006. 411-416.

[99] Parkins, Andrea. 'Nothing To Be Scared Of'. In *Grounds For Possible Music: On Gender, Voice, Language and Identity*. Edited by Julia Eckhardt. Errant Bodies, 2018. 132.

[100] Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai and Soumith Chintala. 'PyTorch: An Imperative Style, High-Performance Deep Learning Library'. *Advances in Neural Information Processing Systems* 32. Curran Associates, Inc., 2019. 8024-8035.

[101] Pelinski, Teresa, Victor Shepardson, Steve Symons, Franco Santiago Caspe, Adan L Benito Temprano, Jack Armitage, Chris Kiefer, Rebecca Fiebrink, Thor Magnusson and Andrew McPherson. 'Embedded AI for NIME:Challenges and Opportunities'. In Proceedings of NIME 2022: New Interfaces for Musical Expression, Waipapa Taumata Rau, Aotearoa, University of Auckland, New Zealand and online, 28 Jun - 1 Jul 2022.

[102] Perraudin, N., P. Balazs and P.L. Søndergaard. 'A fast Griffin-Lim algorithm' in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. Oct, 2013. 1-4.

[103] Pressing, Jeff. 'Improvisation: Methods and Models'. In *Generative Processes in Music: The Psychology of Performance, Improvisation, and Composition*, edited by John Sloboda, 129-156. Oxford University Press, 2001.

[104] Radford, Alex, Luke Metz and Soumith Chintala, 'Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks'. Poster presentation at ICLR 2016: 4th International Conference of Learning Representations. Caribe Hilton, San Juan, Puerto Rico. May 2-4, 2016.

[105] Rawat, Waseem and Zenghui Wang. 'Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review'. *Neural Computation* 29, Issue:9. MIT Press. Sep 2017. 2352-2449.

[106] Rojas, Raúl. 'The Backpropagation Algorithm'. *Neural Networks*. Springer, 1996. 149–182.

[107] Rott Shaham, Tamar, Tali Dekel, Tomer Michaeli. 'SinGAN: Learning a Generative Model from a Single Natural Image'. Unpublished paper. arXiv:1905.01164.

[108] Rouard, Simon, Francisco Massa & Alexandre Défossez. 'Hybrid Transformers for Music Source Separation'. Unpublished paper. ArXiv:2211.08553.

[109] Franziska Schroeder & Federico Reuben. 'AI Music Improvisation by Franziska Schroeder and Federico Reuben using RAVE model/Stable Diffusion'. YouTube video. Posted by 'freuPinta', 22 April 2024. https://youtu.be/tI6BMrEf4jU?si=nbw8anNr1XBG6ieG.

[110] Ivana Shishoska & Kiril Trbojevikj. 'echoes from the distance'. Entry to AI Song Contest 2023. https://www.aisongcontest.com/participants-2023/ki.

[111] Shorten, Connor and Taghi M. Khoshgoftaar. 'A survey on Image Data Augmentation for Deep Learning'. *Journal of Big Data* 6:60. SpringerOpen, 2019.

[112] Simon, Ian and Sageev Oore. 'Performance RNN: Generating Music with Expressive Timing and Dynamics'. Magenta Blog, 2017. https://magenta.tensorflow.org/performance-rnn.

[113] Slonimksy, Nicolas. 'Twelve-Tone Patterns' in *Thesaurus of Scales and Melodic Patterns*. Scribner, 1947. 173-175.

[114] Chintala, Soumith. 'NIPS 2016 Workshop on Adversarial Training'. YouTube video. Uploaded by 'David Lopez-Paz', Feb 17 2017. https://www.youtube.com/watch?v=X1mUN6dD8uE.

[115] Stability.AI. *Stable Audio*, version 2.0. Web service. https://stability.ai/stable-audio.

[116] Christian J. Steinmetz, Reiss, Joshua. 'auraloss: A collection of audio-focused loss functions in PyTorch'. Code repository, Dec 15, 2020. https://github.com/csteinmetz1/auraloss

[117] Phillipp Stolberg & Edgar Eggert. 'Noise to Water'. Entry to AI Song Contest 2022. https://www.aisongcontest.com/participants-2022/aiphex-twins.

[118] Stoller, Daniel, Sebastian Ewert, Simon Dixon. 'Wave-U-Net: A Multi-Scale Neural Network for End-to-End Audio Source Separation'. Paper presented at ISMIR 2018: 19th International Society for Music Information Retrieval Conference, IRCAM, Paris, Sep 23-27, 2018. arXiv:1806.03185.

[119] Stowell, Dan and Mark Plumbley. 'Adaptive whitening for improved real-time audio onset detection'. Proceedings of ICMC 2007: 33rd International Computer Music Conference 2007, Copenhagen, Denmark, Aug 27-31 2007.

[120] Sturm, Bob, Joao Felipe Santos and Iryna Korshunova. 'Folk Music Style Modelling by Recurrent Neural Networks with Long Short Term Memory Units'. Paper presented at ISMIR 2015: 16th International Society for Music Information Retrieval Conference. Hotel NH Malaga, Malaga, Spain. Oct 26-30, 2015.

[121] Taylor, Luke and Geoff Nitchke. 'Improving Deep Learning with Generic Data Augmentation'. *IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018.

[122] Tremblay, Pierre Alexandre, Owen Green, Gerard Roma, Alexander Harker. 'From collections to corpora: Exploring sounds through fluid decomposition'. Paper presented at ICMC 2019: 45th International Computer Music Conference, Elmer Holmes Bobst Library, MORE, New York University, USA. Jun 16-23, 2019.

[123] Luca Turchet. 'Dialogues with Folk-RNN: Smart Mandolin performance at NIME 2018'. YouTube video. posted by Luca Turchet, Jul 3 2018. https://youtu.be/VmJdLqejb-E.

[124] Udio. *AI Music Generator*, beta version. Web service. https://www.udio.com/.

[125] van den Oord, Aaron, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior and Koray Kavukcuoglu. 'WaveNet: A Generative Model for Raw Audio', *Deepmind*. Blog post, Sep 8, 2016. https://www.deepmind.com/blog/wavenet-a-generative-model-for-raw-audio.

[126] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin. 'Attention is All You Need'. Unpublished paper. arXiv:1706.03762.

[127] Jack Walker. 'Power Trio'. YouTube video. Posted by 'AI Music Creativity 2022', Sep 20 2022. https://youtu.be/U7S8quow0_U.

[128] Wang, Qi, Yue Ma, Kun Zhao and Yingjie Tian. 'A Comprehensive Survey of Loss Functions in Machine Learning'. *Annals of Deep Learning* 9. Springer, 2022. 187–212.

[129] Wright, Alex, Vesa Välimäki. 'Perceptual Loss Function for Neural Modelling of Audio Systems'. Conference Paper. Proceedings of ICASSP 2020: 45th International Conference on Acoustics, Speech, and Signal Processing, Online/Barcelona, May 4-8 2020.

[130] Yin, Zongyu, Federico Reuben, Susan Stepney and Tom Collins, 'Measuring When a Music Generation Algorithm Copies Too Much: The Originality Report, Cardinality Score, and Symbolic Fingerprinting by Geometric Hashing' in *SN Computer Science* 3:340, SpringerLink, Jun 2022.

[131] John Zorn. *The Classic Guide to Strategy: VOLUMES ONE AND TWO*. Tzadik Records, 1996. CD (originally released as separate volumes in 1983 and 1986).