Stage 1 Report

Automatic Music Score Transcription with Deep Neural Networks

Lele Liu

Supervisors:
Dr. Emmanouil Benetos
Dr. Veronica Morfi
Prof. Simon Dixon

Independent Assessor:
Dr. Marcus Pearce

December 8, 2020

Centre for Digital Music

Contents

Chapter 1

Introduction

1.1 Automatic Music transcription

Automatic Music Transcription (AMT) is a core problem in the field of Music Information Retrieval (MIR), it is the process of converting music audio into human or machine-readable music scores using computer algorithms [5]. The use of AMT systems is not limited to creating music notation from music recordings, but goes to a wider field of music-related tasks. Music transcription results can be further used in music source separation or hand separation tasks, the design of music recommendation/search systems, the analysis of music improvisation, error detection in music education, etc.

Research in AMT date back to the 1970s, when the problem was solved using signal processing methods [46]. In recent years, various methods have been used for AMT. Two of the most widely used methods are Non-negative Matrix Factorization (NMF) and deep learning methods. Since the introduction of deep learning [40], there has been a large body of research developing AMT systems using deep learning methods. In recent years, Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) are widely used in AMT research [14, 75, 19, 44]. Other deep learning methods such as multi-task learning (MTL) [36, 41] and using sequence-to-sequence models [64] are applied to AMT tasks.

However, due to the complexity of music signals, especially the frequent concurrent sound events, the problem of polyphonic automatic music transcription is still challenging. Besides, current AMT systems usually give output in a piano-roll or note sequence format. There are limited research on *complete transcription* where the systems can give score-level transcription results.

1.2 Research Question & Aims

In our project, we aim to explore deep learning methods for *complete music transcription*. More specifically, we are interested in transcribing music audio signals directly into a score-level music representation, which can be simply decoded into a machine-readable music format (e.g. musicXML, MEI). We are going to mainly focus on polyphonic piano music, and further extend our methods to more instruments.

1.3 Current outputs

Below are our outputs so far:

- Lele Liu and Emmanouil Benetos, "Automatic music accompaniment with a chroma-based music data representation," DMRN+14: Digital Music Research Network One-day Workshop, London, Dec 2019.
- Lele Liu and Emmanouil Benetos, "From audio to music notation," AI+Music Handbook, accepted.
- Adrien Ycart, Lele Liu, Emmanouil Benetos and Marcus Pearce, "Investigating the perceptual validity of evaluation metrics for automatic piano music transcription," Transactions of the International Society for Music Information Retrieval (TISMIR), 2020.
- Lele Liu, Veronica Morfi and Emmanouil Benetos, "Joint Piano-roll and Score Transcription for Polyphonic Piano Music," DMRN+15: Digital Music Research Network One-day Workshop, London, Dec 2020.
- Lele Liu, Veronica Morfi and Emmanouil Benetos, "Joint Multi-pitch Detection and Score Transcription for Polyphonic Piano Music," submitted to ICASSP 2021.

1.4 Report structure

The rest of the report is as follows. In Chapter 2, we provide a background research on AMT, mostly from our stage 0 report[1]. Our current project progress is described in Chapter 3, including experiments on time-frequency representations, score representations and joint multi-pitch detection and score transcription system. Chapter 4 covers our proposed future work, and Chapter 5 provides our time and publication plans during the PhD project. Appendix includes additional information on Covid-19 impact, training courses, and attached documents including stage 0 report, personal development plan, and some of our current research outputs.

[1] https://cheriell.github.io/documents/report/stage_0_report.pdf

Chapter 2

Literature Review

In this section, we review the general process of AMT and deep learning methods that are used in AMT or the wider fields of sequential problems such as Natural Language Processing (NLP), Automatic Speech Recognition (ASR) and Sound Event Detection (SED) in literature. We look into the theories of the methods and their applications in related fields. Finally, we look into the limitations and challenges in current AMT research.

2.1 Problem Definition

As mentioned in Chapter 1, AMT aims to get music notation from music audio signals [5, 6]. It is related with some other tasks in the field of MIR, such as onset detection, source separation and audio-to-score alignment. AMT is more complex than onset detection, whose result contains note onset information. On the other hand, onset detection and AMT results can be combined to achieve better system performance [36]. The same applies to AMT and source separation when it comes to multi-instrument music, where the results are related and can be jointly used in a multi-task manner [53].

In the wider fields of artificial intelligence, AMT is considered as a similar problem to automatic speech recognition (ASR) in music [5]. AMT creates a link between the acoustic and symbolic domain in the music fields, similar to what ASR does in the speech field. Besides, there are some similarities between AMT and the wider field of natural language processing, where music language models are used to describe musical grammar [91].

In most AMT systems, music signals are first converted into some time-frequency representations through algorithms such as Short-Time Fourier Transform (STFT) [67, 68] or Constant-Q Transform (CQT) [75, 42]. The system output may be in various formats depending on systems, from low-level results in piano-roll format, to high-level symbolic music formats such as musicXML, Lilypond and MEI.

An essential subtask in the AMT process is *pitch detection* or *multi-pitch detection* for polyphonic music. It is also called *multiple fundamental frequency estimation* (multiF0 estimation) in many cases, due to the close correlation between pitch and fundamental frequency. By separating music signals in short time frames, we obtain an output in the piano-roll format, indicating pitch activation in each time frame. However, the prediction result in this

step can be noisy with many extra or missing pitches in frames. MultiF0 estimation is widely explored in the literature and a large amount of existing work lies in this step [36, 75]. This subtask is evaluated as the "MultiF0 estimation" task in the annual Music Information Retrieval Evaluation eXchange (MIREX)[1].

After the first step of MultiF0 estimation, the next step called *note tracking* aims to predict notes from the results of MultiF0 estimation. It usually generates note sequences with three components - pitch, onset and offset. The note sequences can be converted into MIDI format for re-synthesizing, which makes it much easier for listeners to have a direct perception of how well the system performs. Various methods are used in this step, from median filtering [76, 60] to deep learning methods [16]. Besides, music language models are found useful in improving note tracking results [91]. Note tracking is evaluated annually in the MIREX "note tracking" task[1].

In most cases, polyphonic music can be divided into different parts (e.g. melody/accompaniment or soprano/alto/tenor/bass) or different instruments. This is when *multi-pitch streaming* (or *timbre tracking*, *instrument assignment* for multi-instrument music) works to separate notes into streams [47]. There are some other tasks in the wider field of MIR that are related to multi-pitch streaming, including music source separation [79] and instrument recognition [39].

The final step of AMT is creating music notation from the previous obtained output, we call it *score typesetting*. The system output is usually defined in a symbolic music representation. Related tasks include rhythm quantization [60] and note value recognition [61]. Notes are organised into meter-based timings instead of real time, and further typesetting will transcribe the music into music scores. It is worth mentioning that the AMT process is not always divided into the above steps. Some AMT systems use approaches that can skip some middle stage output [19, 64]. Regardless of the choice of approaches, we refer to the process of converting music audio into music staff notation as *complete music transcription*.

2.2 Datasets and Evaluation Metrics for AMT

2.2.1 Datasets

As there is an increasing amount of exploration on deep learning methods for AMT, people are using larger datasets to train and evaluate the systems they developed. There are several datasets that are commonly used for AMT problems in literature, such as the RWC dataset [32], MIDI Aligned Piano Sounds (MAPS) dataset [28], Bach10 [27], MedleyDB [9], and MusicNet [78]. Two recently proposed datasets are MAESTRO [37] and Slakh [54].

Although there are plenty of choices of AMT datasets, there are relatively more datasets for piano transcription (given the ease in automatically exporting MIDI annotations from acoustic pianos when using specific piano models such as Disklavier or Bösendorfer), but much less for other instruments, especially non-Western instruments. The biggest challenge of collecting AMT datasets is that annotating music recordings requires a high degree of music expertise, and is very time-consuming. Also, there might not be enough music pieces

[1] https://www.music-ir.org/mirex/wiki/MIREX_HOME

and recordings for some less popular traditional instruments when a large dataset is needed. Moreover, human-annotated transcription datasets are not guaranteed to have a high degree of temporal precision, which makes them less suitable for model evaluation on frame and note level. In [76], Su and Yang proposed four aspects to evaluate the quality of a dataset: generality, efficiency, cost and quality. They suggest that a good dataset should be not limited to a certain music form or recording conditions, should be fast-annotated, should be as low-cost as possible, and be accurate enough. Because of the difficulty in collecting large human-transcribed datasets, researchers have used electronic instruments or acoustic instruments with sensors that can directly produce annotations (e.g. electronic piano, MAESTRO dataset), or synthesised datasets (e.g. Slakh) instead of real recordings. The use of synthesized recordings greatly speed up dataset collection, but on the other hand, could introduce some bias in model training, limiting generality of the developed AMT system.

2.2.2 Evaluation Metrics

Despite collecting datasets, model evaluation is another important process in developing methodologies for AMT problems. Evaluating a music transcription can be difficult since there are various types of errors, from pitch errors to missing/extra notes, and each has a different influence on the final evaluation of results. Currently, common evaluation metrics for AMT systems focus mainly on frame/note level transcriptions [8, 4, 15, 42, 36]. Much less work has been done on stream and notation level transcriptions [56, 57, 58]. In the 2019 annual Music Information Retrieval Evaluation eXchange (MIREX), there are three subtasks[2] for music transcription for pitched-instruments - multiple fundamental frequency estimation on frame level, note tracking, and timbre tracking (multi-pitch streaming).

Common multiple fundamental frequency estimation methods [4] calculate frame-wise *precision*, *recall* and relevant *F-measure* values. The three scores are defined as:

$$precision = \frac{TP}{TP + FP} \tag{2.1}$$

$$recall = \frac{TP}{TP + FN} \tag{2.2}$$

$$F\text{-}measure \ = \frac{2 \times precision \times recall}{precision + recall} \tag{2.3}$$

the TP, FP and FN values correspond to *true positives*, *false positives* and *false negatives* respectively, and are calculated from all pitch values and time frames in the piano roll. There are also other methods for evaluating frame-wise transcription, such as separating different types of errors (e.g. missed pitches, extra pitches, false alarm) in multiple F0 estimation. A type-specific error rate is calculated in [65], where the authors defined a frame-level transcription error score combining different error types. Separating different error types can lead to a better interpretation on music transcription evaluation.

Note tracking problems usually define transcription results as sequences of notes, characterized by a pitch, onset and offset. A *tolerance* is defined to allow small errors in onset times since it is difficult to estimate exact time when building an AMT dataset as well as

[2] https://www.music-ir.org/mirex/wiki/2019:Multiple_Fundamental_Frequency_Estimation_%26_Tracking

transcribing music with an AMT system. A common *tolerance* is 50ms, which is used in the MIREX note tracking subtask. Some times offset time is also considered in evaluation, such as using a tolerance of the max between 20% of note length or 50ms for offset time. For any of the above scenarios, note-level precision, recall and F-measure are calculated for a final evaluation. Similar to frame-level F0 estimation, researchers have attempted to include error types in evaluation metrics (see e.g. [58]).

There are less publications on *multi-pitch streaming*. The evaluation for *multi-pitch streaming* uses similar metric like precision and recall. Gomez and Bonada [31] proposed a simple method of calculating accuracy and false rate to evaluate voice streaming applied to A Capella transcription. In 2014, Duan et al [26] used a similar evaluation method to calculate a more general multi-pitch streaming accuracy. The accuracy is defined as:

$$accuracy = \frac{TP}{TP + FP + FN} \tag{2.4}$$

Another work by Molina et al [58] proposed to include types of errors in streaming process, and used a standard precision-recall metric.

Recent years has seen some introduction of evaluation metrics for *complete music transcription* given a recent increase in methods that directly transcribes audio to music scores. Some methods proposed include [23, 57, 56]. A recent approach for evaluating score transcriptions is proposed by Mcleod and Yoshii [56], which is based on a previous approach [57] called *MV2H* (representing Multi-pitch detection, Voice separation, Metrical alignment, note Value detection, and Harmonic analysis). According to this metric, a score is calculated for each of the five aspects, then the scores are combined into a joint evaluation following a principle of one mistake should not be penalised more than once.

While most of evaluation metrics are based on music theory and simple statistical analysis, there are some metrics that contain some considerations on human perception of music transcriptions. In 2008, Daniel et al [24] explored the difference of some error types in AMT from the aspect of human perception, and proposed a modified evaluation metric that weights different error types.

2.3 Deep Learning in AMT

Current literature for AMT includes a mixture of deep learning and matrix decomposition approaches, with deep learning currently being used in the majority of scenarios. Neural networks tend to outperform matrix decomposition methods when there are enough training data, and are commonly used in AMT systems as a supervised learning model for a framewise or notewise output, and a smaller portion of methods are designed for higher levels such as rhythm transcription or typesetting. In the following, we discuss on neural network structures commonly used in AMT, and state-of-the-art methods in AMT systems. A short summary of AMT systems based on deep learning is in Table 2.1.

Table 2.1: State of the art deep learning AMT systems

AMT system	Dataset(s)	Transcription performance by stages (F-measures in percentage)				Comment
		Framewise	Notewise (onset only)	Voice separation	Score	
Sigtia et al. 2016 [75]	MAPS	74.45	67.05	-	-	CNN acoustic model + language model
Kelz et al. 2016 [42]	MAPS	79.33	-	-	-	Improved based on [75]
Bittner et al. 2017 [11]	Bach10, Su, MedleyDB	59% accuracy on Bach10	-	-	-	Proposed HCQT and 'deep salience' representation
Ycart et al. 2019 [91]	MAPS	69.3	71.7	-	-	Blending acoustic model and language model
Howthorne et al. 2018 [36]	MAPS	78.30	82.29	-	-	'Onset and Frame' by Google research
Kim and Bello 2019 [45]	MAESTRO	91.4	95.6	-	-	Improved 'Onset and Frame' by adversarial learning
Kelz et al. 2019 [41]	MAESTRO	89.58	95.38	-	-	Multitask learning (combining onset, offset, pitch, sustain pedal and velocity information)
Nakamura et al. 2018 [60]	MAPS-ENSTDkCl	72.8	-	-	21.4% error rate	Complete transcription step by step
Carvalho and Smaragdis 2017 [19]	music generated by MIDI synthesizers	-	-	-	P(sub) = 0.6% P(ins) = 0.1% P(del) = 0	End-to-end complete transcription for monophonic music
Román et al. 2019 [68]	synthesized Chorales dataset & Quartets dataset	-	-	-	CER = 18.10% for Chorales CER = 13.53% for Quartets	End-to-end complete transcription for four part music

2.3.1 Neural networks for AMT

Research in AMT has increasingly been relying on deep learning models, which use feedforward, recurrent and convolutional layers as main architectural blocks. An early example of a deep neural model applied to AMT is the work of Nam et al [62], which uses a deep belief network (DBN) in order to learn representations for a polyphonic piano transcription task. Resulting learned features are then fed to a support vector machine (SVM) classifier in order to produce a final decision. Another notable early work that made use of deep neural architectures was by Böck and Schedl [14], where the authors used a bi-directional recurrent neural network (RNN) with Long Short-Term Memory (LSTM) units, applied to the task of polyphonic piano transcription. Two points are particularly worth mentioning for the work of [14]: (i) the use of two STFT magnitude spectrograms with different window sizes as inputs to the network, in order to achieve both a "good temporal precision and a sufficient frequency resolution"; (ii) The output is a piano-roll representation of note onsets and corresponding pitches, and does not include information on note durations/offsets.

A first systematic study towards the use of various neural network architectures for AMT was done by Sigtia et al in [75]. The study compared networks for polyphonic piano transcription that used feedforward, recurrent, and convolutional layers (noting that layer types were not combined), all using a constant-Q transform (CQT) spectrogram as input time-frequency representation. Results from [75] showed that networks that include convolutional layers reported the best results for the task, which is also in line with other results reported in the literature, and with current methodological trends related to neural networks for AMT. The ability of convolutional neural networks (CNNs) to function well for tasks related to multi-pitch detection and AMT stems from the useful property of shift-invariance in log-frequency representations such as the CQT: a convolutional kernel that is shifted across the log-frequency axis can capture spectro-temporal patterns that are common across multiple pitches.

Following the work of [75], Kelz et al [42] showed the potential of simple frame-based approaches for polyphonic piano transcription using an architecture similar to [75], but making use of up-to-date training techniques, regularizers, and taking into account hyper-parameter tuning.

An influential work that used CNNs for multiple fundamental frequency estimation in polyphonic music was the *deep salience* representation proposed by Bittner et al [11]. Contrary to most methods in AMT that produce a binary output, the model of [11] produces a non-binary time-pitch representation at 20 cent pitch resolution, which can be useful for both AMT applications but also for several downstream applications in the broader field of MIR. A particular contribution of this work was the use of a harmonic constant-Q transform (HCQT) as input representation; the HCQT is a three-dimensional representation over frequency, time and the harmonic index, produced by computing several versions of the CQT by scaling the minimum frequency used by a harmonic.

The ability of CNNs in learning features in time or time-frequency representations keeps them still active in the AMT literature. This includes the work of Thickstun et al [78] that was carried out as part of the MusicNet dataset, and compared feedforward and convolutional networks learned on raw audio inputs, as opposed to having a time-frequency representation as input. It is worth noting however that convolutional, and more broadly neural networks,

when trained for AMT as a multi-label classification task, face the issue that they appear to learn combinations of notes exposed to them during training, and are not able to generalise unseen combinations of notes – the so-called *entanglement problem* as discussed in [43].

2.3.2 Music language models

Inspired by work in the field of speech processing, where many systems for automatic speech recognition (ASR) benefit from language models that predict the occurrence of a word or phoneme, researchers in MIR have recently attempted to use music language models (MLMs) and combine them with acoustic models in order to improve automatic music transcription performance. While the problem of polyphonic music prediction using statistical machine learning models (such as n-grams and hidden Markov models) is not trivial, the emergence of neural network methods for high-dimensional sequence prediction has enabled the use of MLMs for polyphonic music.

One of the first works to use neural network-based MLMs for polyphonic music prediction and combine them with multi-pitch detection, was carried out by Boulanger-Lewandowski et al [16]. The MLM was based on a combination of a recurrent neural network with a Neural Autogressive Distribution Estimator (NADE). The same RNN-NADE music language model was also used in [75], which was combined with a CNN as the acoustic model, showing that the inclusion of an MLM can improve transcription performance.

It was shown however that the MLMs which operate at the level of a small time frame (e.g. 10 msec) are only able to produce a smoothing effect in the resulting transcription [88]. More recently, Wang et al [85] used an LSTM-RBM language model as part of their proposed transcription system, but each frame corresponds to an inter-onset interval as opposed to a fixed temporal duration, resulting in improved transcription performance when using note-based metrics. Finally, Ycart et al [91] combined an LSTM-based music language model with a feedforward neural blending model which combines the MLM probabilities with the acoustic model probabilities. In line with past observations, the blending and language models work best when musically-relevant time steps are used (in this case, time steps corresponding to a 16th note).

2.3.3 Multi-task learning

Recent research in machine learning has focused on *multi-task learning* [69], where multiple learning tasks are addressed jointly, thus exploiting task similarities and differences. In the context of AMT, multi-task learning has been shown to improve transcription performance in certain cases. Tasks related to AMT such as note level transcription, onset detection, melody estimation, bass line prediction and multi-pitch detection (also sharing similar chroma and rhythm features) can be integrated into one model that would exploit task interdependencies.

In the 'Onsets and Frames' system by Hawthorne et al [36], which is currently considered the benchmark in automatic piano transcription, the authors used a deep convolutional and recurrent neural network (CRNN) to jointly predict onsets and multiple pitches. The onset detection results are fed back into the model for further improving frame-wise multi-pitch predictions. The Onsets and Frames model was further improved in the work of Kim and

Bello [45], which addresses the problem of expressing inter-label dependencies through an adversarial learning scheme.

Bittner et al [10] proposed a multi-task model that jointly estimates outputs for several AMT-related tasks, including multiple fundamental frequency estimation, melody, vocal and bass line estimation. The authors show that the more tasks included in the model, the higher the performance, and that the multi-task model outperforms the single-task equivalents. In another recent work [41], the authors designed a multi-task model with CNNs which enables four different transcription subtasks: multiple-f0 estimation, melody estimation, bass estimation, and vocal estimation. Results on the method of [41] showed an overall improvement in the multi-task model compared to single task models.

2.3.4 Complete music transcription

Recent works have paid attention to *complete transcription*, where systems are developed to convert music audio into a music score. There are two common ways in designing a complete transcription system. A traditional way is by using a combination of several methods and subtasks of AMT to form a system that can transcribe music audio to a notation level, which usually involves estimating a piano-roll representation in an intermediate process [60]. Another way which has become increasingly popular is designing an end-to-end system that directly converts input audio or a time-frequency representation into a score level representation such as textual encoding, without having a piano-roll or similar intermediate representation in the pipeline. In this scenario, a deep learning network is used to link the system input and output. A challenge in designing an end-to-end system is that the input and output of the system cannot be aligned directly (one is a time-based representation and the other is a representation in terms of metres or symbolic encoding). As a result, research has focused on encoder-decoder architectures [19, 64] which do not rely on framewise aligned annotations between the audio and music score.

A work worth mentioning which combined subtasks to build a transcription system is by Nakamura et al [60]. In this work, the authors divided a whole transcription system into a stream of subtasks: multi-pitch analysis, note tracking, onset rhythm quantization, note value recognition, hand separation, and score typesetting. The final system reads a spectrogram calculated from music audio, and outputs readable music scores. Offering the whole system structure, the authors did not focus on integrating algorithms for all the subtasks, but optimized methods for multi-pitch detection and rhythm quantization. The improved subtask performance ends up adding to the final performance of the system.

Encoder-decoder mechanisms have also been used for AMT in recent years, with the advantage in creating complete transcription systems without estimating and integrating complicated subtasks. Recent works have showed the potential of encoder-decoder methods, although their performance on polyphonic music transcription remains less explored in the literature. In 2017, Carvalho and Smaragdis proposed a method for end-to-end music transcription using a sequence-to-sequence architecture combined with CNNs and RNNs [19]. The developed system can output a textual music encoding in Lilypond language from an input audio waveform. However, the work focused mainly on monophonic music (which showed high-level performance), but only a simple scenario of polyphonic music was tested (with two simultaneous melodies within a pitch range of two octaves). Another exploration

on singing transcription by Nishikimi et al [64] also used a sequence-to-sequence model. A point worth mentioning is that they applied an attention loss function for the decoder, which improved the performance of the singing transcription system. The work, still, focused only on monophonic singing voice.

Using an encoder-decoder architecture is a simple way of designing end-to-end AMT systems, but there are also other works using Connectionist Temporal Classification (CTC). A recent example is by Román et al [68], in which the authors combined the use of a Convolutional Recurrent Neural Network and a CTC loss function. The CTC loss function enables the system to be trained using pairs of the input spectrogram and output textual encoding. In that work, a simple polyphonic scenario is considered where four voices are included in a music piece (in string quarters or four-part Bach chorales). The problem of end-to-end complete music transcription with unconstrained polyphony is still open.

2.4 Challenges and Limitations in AMT Research

Although AMT is still very active as a topic within MIR, the performance of current AMT systems is still far from satisfactory, especially when it comes to polyphonic music, multiple instruments, non-Western music, and 'complete' transcription. There are plenty of challenges in this area where further exploration is required. In this section, we summarise current challenges and provide potential further directions.

2.4.1 Datasets

The lack of annotated datasets is an aspect that limits the development of AMT systems. Due to the difficulty in collecting and annotating music recordings, there is still a lack of data for most music transcription tasks, especially for non-Western music and certain musical instruments. Apart from the lack of large datasets, current datasets for AMT also have some limitations. For example, there is limited instrument and cultural diversity in existing AMT datasets, most datasets are for piano/guitar and western music. The temporal precision of annotations for some datasets with real recordings is not always satisfactory - which is also a reason that most AMT systems set a relatively large onset/offset *tolerance* for note tracking tasks. Also, dataset annotations are typically limited to note pitch, onset and offset times, and sometimes note velocity. Additional annotations are needed for a more comprehensive transcription, such as rhythm, key information, and expressiveness labels.

Recently, an increasing number of datasets has been released, which are based on synthesizing MIDI files. MIDI files provide a good reference for multi-pitch detection since they provide temporally precise note annotations, but there are also limitations, since MIDI files do not provide annotations for score level transcription. Another limitation for synthesized data is that they might not reflect the recording and acoustic conditions of real-world audio recordings and can cause bias during model training.

2.4.2 Evaluation metrics

Current evaluation metrics mainly focus on frame-wise and note-wise evaluations, where transcription results are provided in a piano-roll representation or note sequences. Benchmark evaluation metrics also do not model different error types beyond measuring precision and recall. For example, an extra note may be more severe than a missing note in a polyphonic music, on-key notes may be less noticed than off-key ones, and an error in a predominant voice may be more obvious compared to a similar error in a middle voice. Besides, much less work can be found in evaluating complete transcription systems.

There is also a lack of perceptual considerations in commonly used evaluation metrics. Some work [65, 60] has attempted to create different types of errors, however these metrics still do not account for human perception. Deniel et al provided an early work on perceptually-based multi-pitch detection evaluation [24], but is not widely used in the community. In addition, there is still no work on perceptually-based evaluation metrics for score-level transcription.

2.4.3 Non-Western music

Most AMT methods aim specifically at modelling Western tonal music, but there is much less work done on automatically transcribing music beyond Western tonal music, such as world, folk and traditional music (see a related work in [7]). This results in AMT systems not being able to accurately or adequately transcribe non-Western music.

Differences between Western and non-Western music cultures that can affect the design of AMT systems include but are not limited to pitch space organisation and microtonality, the presence of heterophony (versus homophony or polyphony occurring in Western tonal music), complex rhythmic and metrical structures, differences in tuning and temperament, differences in musical instruments, and differences in methods for expressive performance and music notation amongst others. Despite the above differences, the lack of large annotated datasets is another limitation for music transcription research for non-Western music cultures.

2.4.4 Complete transcription

Although research in AMT has increasingly been focusing on complete transcription in recent years, current methods and systems are still not suitable for general-purpose audio-to-score transcription of multi-instrument polyphonic music. Some systems for complete transcription rely on typesetting methods as a final step (e.g. [60]), but most typesetting methods assume a performance MIDI or similar representation as input and are not designed to take noisy input into account. In addition, when many tasks are combined into a whole system for complete transcription, the errors in each step can accumulate and worsen the system's performance. As for end-to-end transcription methods, current research is still limited to monophonic music [19, 67] and special cases for polyphonic music [68], mostly using synthetic audio. There is still a large room for further work towards the development of systems for complete music transcription.

2.4.5 Expressive performance

Most AMT systems transcribe music into a defined framework of note pitch, onset and offset in a metre constrained format, but cover little expressive labels such as note velocity, speed symbols, as well as expressive playing techniques. Including expressive performance annotations is another challenge in current AMT research. It is currently hard to predict such information in automatic music transcription, although MIR research has been focusing on specific problems within the broader topic expressive music performance modelling (e.g. vibrato detection). How to incorporate the estimation and modelling of expressive performance into AMT systems remains an open problem.

2.4.6 Domain adaptation

Due to the increasing use of synthesized datasets, or due to the mainstream use of piano-speficic datasets for AMT, the ability of such models to generalize to real recordings, different instruments, acoustic recording conditions or music styles has become a problem worth considering. There is currently no research focusing on this question in the context of AMT, although the broader problem of domain adaptation has been attracting increasing interest in MIR and the broader area of machine learning. For example, tasks in MIR such as music alignment and singing voice separation were explored in a recent paper [50] using domain adaptation methods based on variational autoencoders. We believe that similar domain adaptation methods can be applied to automatic music transcription tasks to solve existing problems such as the lack of data for some less popular instruments and dealing with the differences between synthesised and real-life recorded datasets or different recording conditions.

2.5 Deep Learning for Sequential Data

In this section we discuss a little more about the use of sequence-to-sequence models and attention mechanisms in sequential problems. We will now cover the basic sequence model structures such as RNNs and TCNs.

2.5.1 Sequence-to-sequence models

Recurrent neural networks are limited to pre-aligned input and output sequences, but in a more general situation of sequence transduction, we need to deal with non-aligned input/output sequence pairs, and sometimes even free-length output sequence. To solve this problem, people have proposed different approaches. One of the early solutions was proposed by Graves et al. [34] and later improved in [33] as a sequence "Transducer", which uses connectionist temporal classification (CTC) to label non-aligned sequence. A null symbol is used in an intermediate stage to link an aligned output to the final non-aligned output sequence. Thus, this method is limited to labelling monotonic sequence, and the output sequence cannot be longer than the original aligned sequence.

As another solution to fit a more general situation of free-length sequence transduction,

Sutskever et al. proposed the use of a sequence-to-sequence (seq2seq) model [77]. Seq2seq model uses two RNN networks, one as an encoder, another as a decoder, to first map an input sequence into a fixed-dimensional latent vector, and then decode the vector into an output sequence. It is proved to be useful in many applications, such as machine translation [77, 1, 51], automatic speech recognition [2], syntactic constituency parsing [83], image captioning [93] and automatic music transcription [80, 64]. It is worth mentioning that Sutskever et al. found by reversing the order of input sequence, the system's performance can improve markedly, since the operation introduces many short term dependencies between the input and output sequences.

On top of seq2seq models, people have used attention mechanism to improve model performance [20, 82] (more details about attention mechanism to be discussed in the following section). Bahar et al. proposed the use of a two-dimensional sequence-to-sequence model as an alternative of the attention mechanism to model the dependency between the input and output sequences, and tested it in machine translation [1] and automatic speech recognition [2] tasks.

2.5.2 Attention mechanism

Another step towards better sequential modelling is the use of attention mechanism. While the idea of "attend" to some parts of data is not limited to sequential problems (see an use in computer vision in [92]), it is proved to be helpful and commonly used in today's sequential systems including neural machine translation [3], automatic speech recognition [21, 22], sentiment analysis [74], and automatic music transcription [64].

A early use of attention mechanism in automatic speech recognition is in [21], where Chorowski et al. used attention mechanism as an alignment between the input and output sequences. In order to add a preference for monotonic alignment, they apply a penalty to the alignments that map to pre-considered inputs. In a further work in [22], the authors proposed a method to deal with long utterance in speech recognition by extending the original content-based attention mechanism to be aware of location, or more precisely, to be able to take into account the alignment in the previous time step. In the same year, Chan et al. [20] proposed a "Listen, Attend and Spell" model that uses a pyramid structured bi-LSTM encoder (Listener) and an attention-based decoder (Speller) for a character-wise speech recognition.

A general attention mechanism for modelling the dependencies between input and output sequences usually uses (query, key, value) weights, output RNN hidden state, and input sequence vectors to calculate a context vector that implies an attention weights that the current output should pay on each of the input. This is further extended into a concept of *multi-head self-attention* in [82] in 2017. In their network architecture called the "Transformer", attention mechanism is not only used between the decoder's input and output, but also within the encoder and decoder as *self-attention* layers combined with feed forward networks to build dependencies within sentences. At the same time, they choose to calculate not only one set of attention weights, but 8 within each self-attention, to allow a word to "attend" to different words in a sentence. The idea of multi-head self-attention is further explored in works including [74, 71, 87, 84].

Chapter 3

Current Progress

3.1 Data Collection

We create a synthesized dataset using scores collected from the MuseScore website[1]. We do this as a starting point and because there is a lack of AMT datasets that provide score ground truth on both physical and musical time. A dataset that best fits this task is the recently published ASAP dataset [29], which will be investigated as future work. We collect scores in MusicXML format, convert them to MIDI files, and synthesize audio files using all the four piano models in the Native Instruments Kontakt Player[2]. The scores we collect cover various key and time signatures, tempos, modes and polyphony levels, but do not contain grace notes, triplets, arpeggios, trios or other complex playing techniques. Some statistics on the dataset are in Table 3.1 and Figure 3.1.

Table 3.1: Dataset Statistics. For polyphony levels, the numbers out of brackets are calculated without adding piano pedals, and the numbers in brackets are calculated with piano pedals.

Number of music pieces	210
Total hours	9.62 $\times$ 4 piano models
Total notes	222,219
Use of piano pedal	29% (61 pieces)
Maximum polyphony level	13 (26)
Average polyphony level	2.87 (3.21)
Time signatures	4/4, 3/4, 5/4, 6/8, 9/8, etc.
Key signatures	all 12 key signatures

3.2 Time-frequency representation

Like in most AMT systems, we use as input a time-frequency representation of the audio signal. We compare commonly used time-frequency representations – Short-Time Fourier

[1] https://musescore.com/hub/piano

[2] https://www.native-instruments.com/en/products/komplete/samplers/kontakt-6-player/

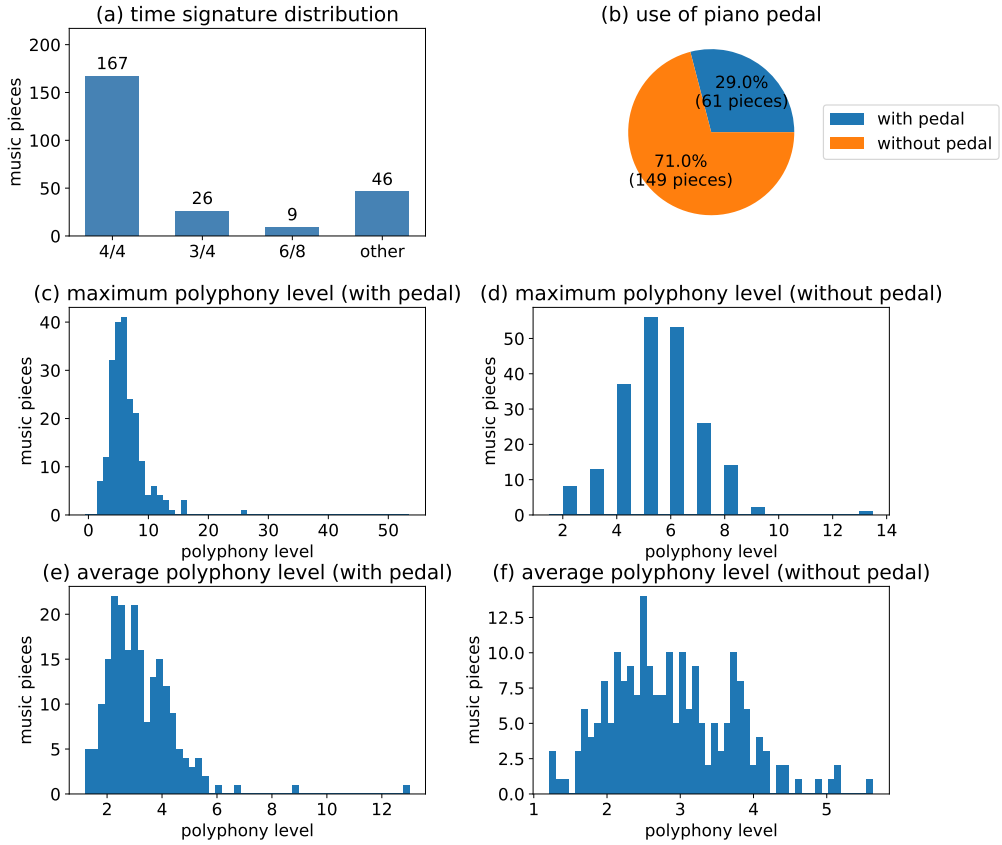


Figure 3.1: Statistics of the dataset. (a) Distribution of time signatures. There may be more than one time signature used in one music piece. (b) Distribution of the use of piano pedals in the music pieces. (c) Distribution of maximum polyphony level per piece (taking into account the use of piano pedals). (d) Distribution of maximum polyphony level per piece (without the use of piano pedals). (e) Distribution of average polyphony level per piece (with sustain pedals). (f) Distribution of average polyphony level per piece (without pedals). Note: (c)-(f) are distributions over the dataset, not subsets from (b).

Transform (STFT), Mel spectrogram, Constant-Q Transform (CQT), Harmonic Constant-Q Transform (HCQT), Variable-Q Transform (VQT) and their different parameters.

The performance is evaluated on a multi-pitch detection task with a Convolutional RecurrentNeural Network (CRNN) network architecture. That is, we assume the best input representation for the piano-roll task will also perform well on score prediction, since the two tasks are highly related.

Experiment All representations are log-valued and the signals are resampled to ensure the hop size of every spectrogram being equal to 10ms, which means equal length in the model input. The time-frequency representations we compare are:

- *STFT* - Magnitude spectrogram from the Short Time Fourier Transform with a Hanning window and FFT window length in {1024, 2048}. Signal sampling rate is 44.1kHz.

- *Mel Spectrogram* - Mel Spectrogram with different FFT window length in {1024, 2048} and different number of Mel bands in {128, 192, 256}. Signal sampling rate is 44.1kHz.
- *Constant-Q Transform (CQT)* - Spectrogram obtained from the Constant-Q Transform [17], with bins per octave in {12, 24, 36, 48, 60}, number of octaves in {7, 8} and lowest frequency equal to pitch A0=27.5 Hz, which is the lowest pitch in piano. Signals are resampled at 25.6 kHz to fit a hop length of 256.
- *Harmonic Constant-Q Transform (HCQT)* - Spectrogram from the Harmonic Constant-Q Transform proposed in [11], which is a 3-dimensional spectrogram with CQTs based on shifted harmonics. The parameters we select from are bins per octave in {36, 60}, number of octaves in {5, 6} and number of harmonics in {4, 5, 6}. Signals are resampled at 25.6kHz.
- *Variable-Q Transform (VQT)* - Spectrogram calculated from Variable-Q Transform proposed in [72]. We select γ values in {10, 20, 30}, number of bins per octave in {36, 60} and number of octaves in {7, 8}. Signals are resampled at 25.6kHz.

All the spectrograms are calculated using librosa [55]. The piano-rolls are calculated using pretty_midi [66], we use the default velocity-valued piano-rolls. We pad the input spectrograms and output piano rolls into the same length (equal to 4s), and map the frame length of the spectrograms to the frame length in the piano rolls. As a default, we use a hop size equal to 10ms for the spectrogram calculation.

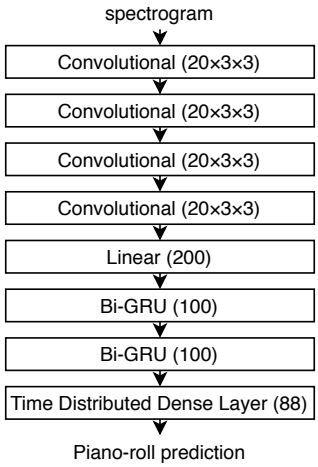


Figure 3.2: Model architecture for input representation experiment with layer sizes.

The structure of the model for multi-pitch prediction is in Figure 3.2 . For simplicity, we start with a simple architecture with little parameter tuning. During training, we leave the padded values out of the calculation of loss functions by adding a mask to filter out losses from the paddings.

Results We compare the performance of different input representations using the benchmark frame-level and note-level onset-only, and onset-offset evaluation from MIREX [4]. We use a onset tolerance of 50ms, but keep the offset tolerance to be 20% of the note duration, or

a 50ms, which ever is larger. The model performance trained in this experiment is shown in Table 3.2. To obtain a binary piano roll from the velocity-valued piano rolls, we threshold the output with a velocity threshold of 30 (velocity under 30 is not audible). No post-processing steps (e.g. smoothing, minimum duration pruning, gap filtering) is applied to the model output, since the RNN already does similar work. Note sequences are directly obtained from the binary piano rolls.

Table 3.2: F-measure of piano-roll prediction on different input representations and parameters. N_w: window length, N_m: number of mel bands, N_b: number of bins per octave, N_o: number of octaves, N_h: number of harmonics. P_f: framewise precision, R_f: framewise recall, F_f: framewise F-score, P_{on}: notewise onset only precision, R_{on}: notewise onset only recall, F_{on}: notewise onset only F-score, P_{onoff}: notewise onset and offset precision, R_{onoff}: notewise onset and offset recall, F_{onoff}: notewise onset and offset F-score.

Input representations			P_f	R_f	F_f	P_{on}	R_{on}	F_{on}	P_{onoff}	R_{onoff}	F_{onoff}
STFT:											
N_w											
1024			90.09	87.42	87.73	86.20	76.64	78.74	61.95	56.83	58.09
2048			**90.36**	**90.36**	**89.46**	**89.51**	**77.76**	**80.99**	**66.24**	**59.86**	**61.73**
Mel Spectrogram:											
N_w	N_m										
1024	192		88.41	87.32	86.88	83.13	75.81	76.89	59.87	56.51	56.89
2048	128		90.90	87.32	88.18	85.94	78.32	79.73	62.54	58.85	59.53
2048	192		90.77	85.72	87.20	85.70	76.47	78.49	62.71	58.02	59.10
2048	256		**91.60**	**88.19**	**88.98**	**90.48**	**78.65**	**82.12**	**67.38**	**60.80**	**62.95**
CQT:											
N_b	N_o										
12	7		90.73	88.81	88.91	88.41	77.96	80.76	65.57	60.11	61.68
12	8		90.92	88.50	88.69	89.48	78.00	81.33	66.02	59.70	61.69
24	8		93.02	90.35	90.91	92.67	80.72	84.39	70.79	63.86	66.15
36	8		93.39	90.67	91.27	92.99	80.91	84.71	70.86	63.94	66.29
48	8		**93.89**	90.44	91.44	**93.45**	81.31	85.14	**72.20**	65.15	**67.56**
60	8		93.79	**91.21**	**91.85**	93.25	**81.96**	**85.43**	71.82	**65.18**	67.40
HCQT:											
N_b	N_o	N_h									
36	5	4	91.47	89.43	89.76	91.43	79.79	83.19	67.76	61.55	63.52
60	5	4	91.85	88.96	89.55	90.88	78.88	82.48	66.79	60.47	62.57
60	6	4	92.17	89.88	90.24	90.89	79.55	82.74	67.07	61.11	62.93
60	6	5	**92.97**	**90.49**	**90.95**	**91.81**	**81.06**	**84.14**	**69.27**	**63.53**	**65.31**
60	6	6	91.60	89.03	89.43	88.67	79.48	81.68	64.49	59.93	61.09
VQT:											
N_b	N_o	γ									
36	7	10	92.87	90.64	91.01	92.54	80.60	84.24	70.75	63.87	66.17
60	7	10	93.22	90.69	91.14	92.33	80.37	83.94	71.41	64.49	66.71
60	8	10	94.00	90.93	91.75	**93.94**	82.08	**85.76**	72.63	65.53	67.94
60	8	20	**94.22**	**91.04**	**91.93**	93.81	**82.11**	85.70	**73.07**	**66.25**	**68.53**
60	8	30	94.15	91.01	91.85	93.91	82.03	85.70	73.00	66.05	68.37

Among the five spectrogram types, VQT shows the best performance, with a γ value of 20, and 8 octaves $\times$ 60 bins per octave in the frequency axis. We discover some trend in how the parameters influence model performance. For example, a window length of 2048outperforms 1024 for STFT and Mel Spectrogram. Larger number of frequency bins tend to result in

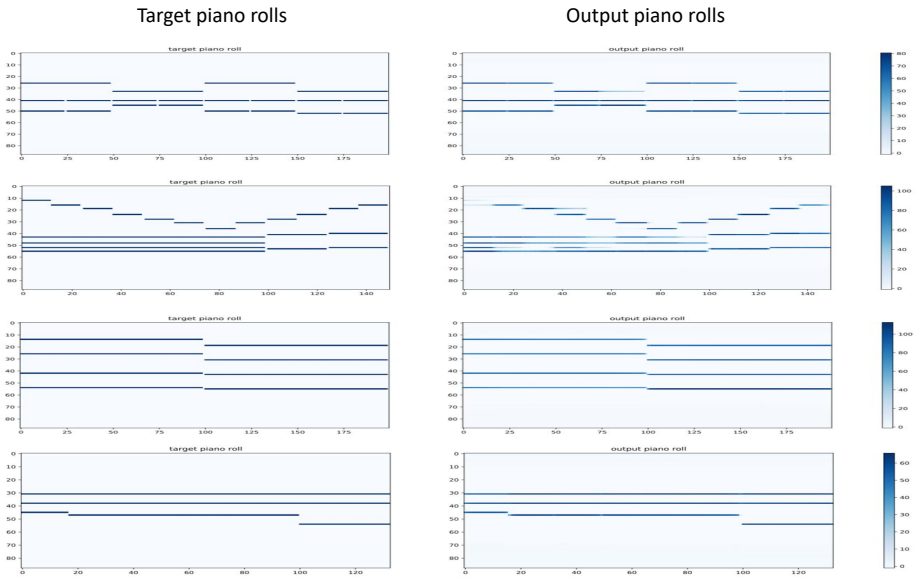


Figure 3.3: Example velocity-valued piano roll targets and outputs from model with the best input representation. Left column: ground truth piano rolls, Right column: transcribed piano rolls.

higher performance for Mel Spectrogram, CQT, HCQT and VQT.

Figure 3.3 includes example ground truth and transcribed piano rolls using the best input representation. Generally, the pitches are correctly predicted. However, the model is not very good at predicting short gaps between repeated notes, which results in merged note errors.

3.3 Score representation

One of the major challenges in developing an end-to-end A2S system is selecting an output representation that can support polyphonic music and includes various cues present in music scores. Compared to sentence outputs in ASR tasks, music notation is much more structured and complex. Some of the most commonly used symbolic music score encoding formats are MusicXML, **Kern, LilyPond, ABC and PAE [59]. However, MusicXML is a verbose music encoding, and the **Kern format only supports monophonic music per voice. All the other three formats support polyphonic music and encode music scores into strings. Here, we use the LilyPond [63] format as a base representation for our score data representation.

Although much more concise than the MusicXML format, LilyPond encoding is still complex with hierarchical structures such as parts and voices. To make the transcription task simpler, we assume that there are only two hand parts in piano music and only one voice per hand, where each voice can have multiple concurrent notes. We consider the left hand part and right hand part scores as two outputs predicted jointly. In this way, we discard the hierarchical structure of the LilyPond format and keep the most essential information in two strings. We assume our model to predict one bar at a time, this means that we only

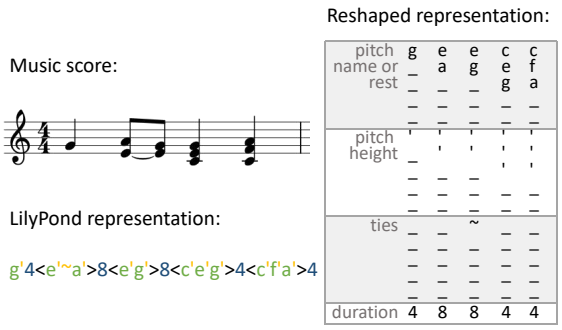


Figure 3.4: Example music score and corresponding LilyPond and Reshaped representation.

take into account the notes and rests in our score data representation, no barline or key/time signature symbols are included. We do not consider adding complex playing techniques such as arpeggios, trios, vibratos nor complex rhythm structures such as triplets, quintuplets. The symbols we use in LilyPond format are:

- *Pitch* - Combined with pitch chroma (e.g. 'c' for C, 'cis' for $C\sharp$ and 'ces' for $C\flat$) and pitch height (e.g.' $'$ ' for higher octave and ' , ' for lower octave, duplicate e.g. ' $''$ ' for double octaves).
- *Duration* - We use numbers to represent durations, e.g. '8' for an 8th note duration (duration symbol can be omitted for a 4th note duration). The same duration representation is used for chords and rests. ' . ' is added for dotted notes - resulting in e.g. '4.'.
- *Rest/Note/Chord* - A rest is represented as 'r' followed by its duration symbol. A note/chord is represented by its pitch(es) and duration. Chord pitches are grouped by brackets (e.g. pitches for a C major starting with a middel C is '$\langle\ c'\ e'\ g'\ \rangle$')
- *Tie* - Ties are represented using '$\sim$', added to its start note, such as '$c4 \sim c8$' for a tied note c, or '$\langle c \sim\ e\ g\rangle 4 \langle c\ f\ a\rangle 2$' for a tied c in chord.

Experiment Based on the above defined musical symbols, we compare the following two score data representations; an example for the two score representations can be seen in Figure 3.4.

- *LilyPond representation* - A representation based on LilyPond encoding by removing extra symbols and keeping only the described necessary symbols to reconstruct a musical score.
- *Reshaped representation* - Considering the length of a LilyPond score representation and the difficulty in learning structural information, we propose a Reshaped data representation based on the LilyPond representation that describes a score in a 2D matrix of symbols. We assume a maximum of five concurrent notes per hand, one for each finger, and the 2D matrix of symbols is indexed by symbol index and time, where each time step consists of (5+5+5+1=)16 symbols corresponding to five symbols for each one of

pitch names or rest, pitch heights, ties and one symbol for duration. Each column of the matrix can reconstruct a rest, note or a chord in a music score.

We separate piano solo music into two hand parts, and consider the prediction of right hand part and left hand part as separate tasks in a multitask learning model. The scores for are jointly predicted using a model with a shared CNN stack and two sequence-to-sequence models with attention mechanism originally used in machine translation in [3]. Detailed model architecture is in Figure 3.5.

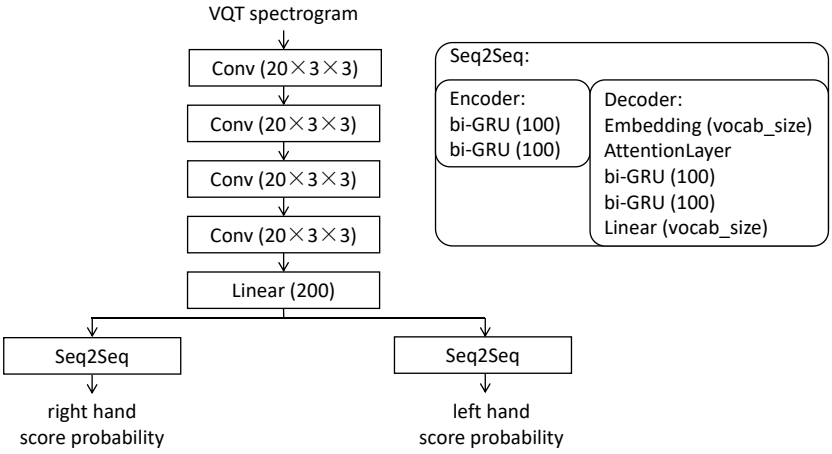


Figure 3.5: Model architecture with layer sizes.

We use the ground truth downbeat times to cut audio recordings into bars. Zero padding is added to the input audio spectrograms. For score representations, we split them into lists of symbols and consider the symbols as tokens in a sentence like tokens in natural language processing problems [30]. SOS, EOS and PAD symbols are added to the sentences. Symbols are encoded by one-hot encoding. For the Reshaped representation, the symbols are separately one-hot encoded, that is, we use separate index dictionaries for pitch name, pitch height, tie and duration symbols.

A negative log-likelihood loss is used for the score representation probabilities. A 50% teacher forcing ratio [30] is used in the training process. This means the models use the ground truth in the decoding process with 50% chance, otherwise use the most probable output symbol. During inference, a 0% teacher forcing ratio is used, and we simply adopt a greedy decoding to obtain the predicted output sequence. To further obtain a full score transcription from the models, we combine the predicted scores for all bars, and post-process the scores by obtaining the most probable time signature, adding missing rests and removing extra rests. The final score representation can be directly decoded to MusicXML format.

Results Since there is no existing standard A2S evaluation metric, we use the following two metrics as an indication of the system's performance:

- *Word error rate (WER)* of the LilyPond representation, adopted from neural machine translation tasks [30]. For the Reshaped representation, we first reconstruct the output to the LilyPond format and then calculate the word error rate to the ground truth LilyPond representation.

- *MV2H metric* proposed in [57] for complete AMT evaluation, composed of five sub-metrics: multi-pitch detection accuracy (F_p), voice separation accuracy (F_{voi}), metrical alignment accuracy (F_{met}), note value detection accuracy (F_{val}). Harmonic analysis is not included since we do not include key detection and chord estimation. The overall accuracy of this metric (F_{MV2H}) is the average over the four sub-metrics. In this work, we use the v1.0 of this metric, assuming our transcription and the input audio are time-aligned.

Table 3.3: Word error rates and MV2H results in percentage for different models. LilyPond: Model with LilyPond representation; Reshaped: Model with Reshaped representation. Both models are trained on three pianos and evaluated on four pianos in the synthesized dataset.

WER	wer_{right}		wer_{left}		wer
LilyPond	38.0		39.0		38.5
Reshaped	**37.8**		**34.5**		**36.2**
MV2H	F_p	F_{voi}	F_{met}	F_{val}	F_{MV2H}
LilyPond	66.7	**90.3**	94.8	93.2	86.3
Reshaped	**69.6**	89.7	94.8	**93.7**	**86.9**

Table 3.4: Time and space used in training (or inference) with LilyPond representation and Reshaped representation.

Score representation	Time	Memory (MB)
LilyPond	1194m47s (682m27s)	10817 (2555)
Reshaped	**156m19s (112m39s)**	**5043 (1677)**

Table 3.3 shows the WER evaluation and MV2H evaluation results on score prediction for the two score representations. All inputs are VQT spectrograms with 8×60 frequency bins and a γ value equal to 20. Results show the Reshaped representation is slightly better than the LilyPond representation in both metrics. The Reshaped representation also outperforms the LilyPond representation in terms of the time and memory resources required. Table 3.4 shows the time and memory resources required by the two score representations. Training time/memory is measured as average values per epoch with a batch size of 8, and inference time/memory is for predicting scores for the whole test set. Score post-processing time is not included in the measurement. We notice that the Lilypond representation inference time is larger than the duration of the music recording, and the Reshaped representation inference time is smaller than the duration of the music recording.

3.4 Joint piano-roll and score transcription

3.4.1 Multitask learning model

We combine the tasks of multi-pitch detection and score transcription based on the models we developed in Section 3.2 and Section 3.3 into a multitask learning model. The multitask end-to-end model is composed of convolutional layers, recurrent layers and sequence-to-sequence

models with an attention mechanism for A2S. It is, to our knowledge, the first holistic model that transcribes polyphonic piano music into both a piano-roll format (corresponding to a descriptive notation of the music audio) and a score in Western staff notation (corresponding to a prescriptive notation of the musical audio).

Experiment Using the best input and score representation from experiments in Section 3.2 and Section 3.3, we train a "Joint Model" that simultaneously predicts a piano-roll transcription and a symbolic score transcription. Here, we refer to the model used in time-frequency representation comparison experiment (described in Section 3.2) as "Piano-roll Only Model"; the model used in score representation comparison experiments (described in Section 3.3) as "Score-only Model". We compare the F-measures on piano-roll prediction between the Piano-roll Only Model and the Joint Model, and the WER and MV2H evaluation results between the Score-only Model and the Joint Model to see how the two tasks influence each other.

The model architecture for the Joint Model is in Figure 3.6. The convolutional layers and sequence-to-sequence models in the Joint Model are identical to the ones in the Score-only Model, and the piano-roll prediction branch follows the model structure of the Piano-roll Only Model.

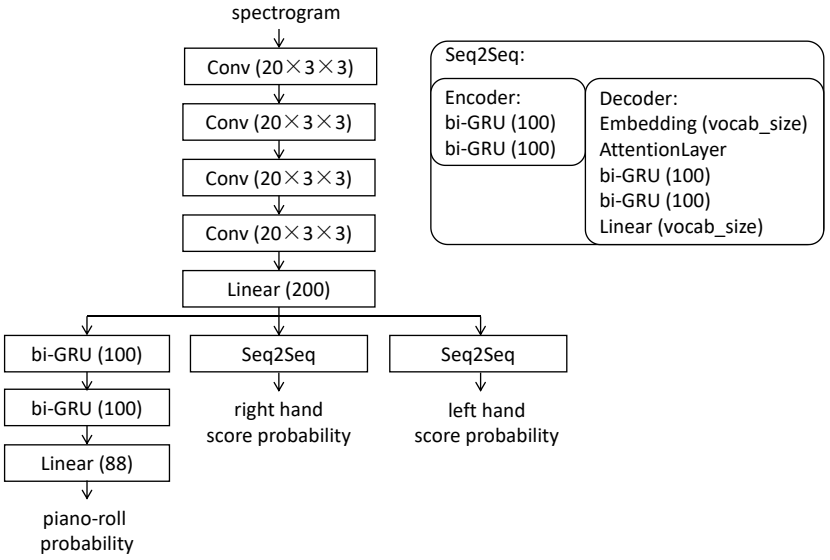


Figure 3.6: Model architecture for the multitask learning model.

Results Table 3.5 shows the model performances on multi-pitch detection and score transcription for single-task and multi-task models. The Joint Model performs better than the single-task models in terms of F-measure on piano-roll prediction and MV2H metric on score transcription. Although there is no large difference in terms of word error rate in score transcription, we still believe the multi-pitch detection task and the score transcription task helps each other. Since MV2H metric is designed specifically for music transcription, and is more accurate than WER which is adopted from natural language processing problems.

One example ground truth score and its transcription from the Joint Model is in Figure 3.7. In the example, the model does well in predicting meters and harmonies and can

Table 3.5: Performances on single-task and multi-task models.

<table>
<tr><td>F-measure</td><td colspan="2">F_f</td><td colspan="2">F_{on}</td><td>F_{onoff}</td></tr>
<tr><td>Piano-roll Only</td><td colspan="2">86.4</td><td colspan="2">67.6</td><td>52.0</td></tr>
<tr><td>Joint</td><td colspan="2">88.0</td><td colspan="2">66.7</td><td>53.6</td></tr>
<tr><td>WER</td><td colspan="2">wer_{right}</td><td colspan="2">wer_{left}</td><td>wer</td></tr>
<tr><td>Score-only</td><td colspan="2">37.8</td><td colspan="2">34.5</td><td>36.2</td></tr>
<tr><td>Joint</td><td colspan="2">37.6</td><td colspan="2">35.3</td><td>36.5</td></tr>
<tr><td>MV2H</td><td>F_p</td><td>F_{voi}</td><td>F_{met}</td><td>F_{val}</td><td>F_{MV2H}</td></tr>
<tr><td>Score-only</td><td>69.6</td><td>89.7</td><td>94.8</td><td>93.7</td><td>86.9</td></tr>
<tr><td>Joint</td><td>71.1</td><td>90.8</td><td>94.9</td><td>94.4</td><td>87.8</td></tr>
</table>

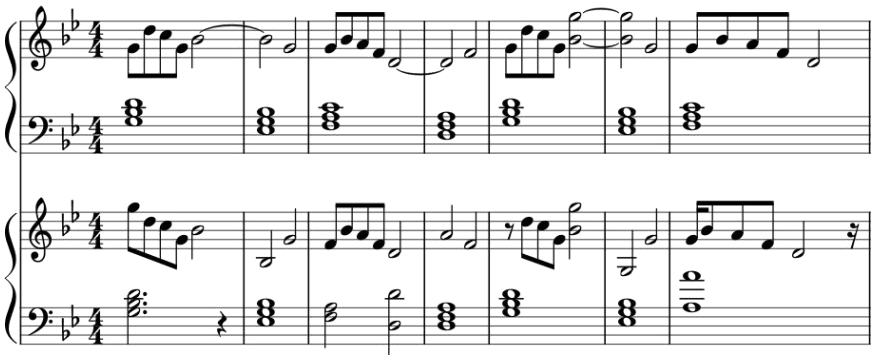


Figure 3.7: Example transcription output. Upper grand staff: ground truth. Lower grand staff: transcription output, ground truth key signature is used for visualisation purposes.

transcribe the melodies in general. Main errors include octave errors, note duration errors, extra/missing note errors and voice separation errors.

3.4.2 Data pre-processing

Based on our joint model developed in Section 3.4.1, we further explore the effect of data pre-processing techniques. More specifically, the use of:

- Waveform normalisation before calculating time-frequency representations;
- Spectrogram bin-wise normalisation.

Experiment We use the same train/valid/test split ratio of 8:1:1 with the synthesized dataset we collected. Among the four piano models in the dataset, we use three of them for training/validation, and all four of them for testing. We further fine tune and evaluate the model on MAPS-ENSTDkCl dataset.

We keep the model architecture the same as the Joint Model we used previously, except for changing the piano-roll output to a binary valued matrix. We re-scale music signal amplitude to the range of [-1, 1] for waveform normalisation, and use a 2-dimensional batch normalisation layer after the model's input layer for spectrogram normalisation. We compare:

- **NoNorm** - model with no pre-processing;
- **Norm1** - model with waveform normalisation;
- **Norm2** - model with both waveform normalisation and spectrogram normalisation.

Table 3.6: Upper: Model performance on the MuseScore synthesized dataset, all three models are trained for 15 epochs on the synthesized dataset. Bottom: Model performance on the MAPS-ENSTDkCl dataset, models are further fine tuned for 2 epochs on the MAPS-ENSTDkCl dataset.

WER	wer_{right}	wer_{left}	wer
NoNorm	37.6	35.3	36.5
Norm1	38.2	35.3	36.7
Norm2	**36.5**	**32.7**	**34.6**
NoNorm	72.4	75.2	73.8
Norm1	63.1	70.3	66.7
Norm2	**60.9**	**66.9**	**63.9**

MV2H	F_p	F_{voi}	F_{met}	F_{val}	F_{MV2H}
NoNorm	**67.7**	**90.4**	94.8	93.9	**86.7**
Norm1	65.6	89.9	94.8	93.9	86.1
Norm2	66.7	90.1	94.8	**94.1**	86.4
NoNorm	8.6	52.8	85.3	48.0	48.7
Norm1	28.1	**82.9**	85.3	79.8	69.0
Norm2	**30.1**	82.4	85.3	**82.2**	**70.0**

Results Model performances on the two datasets (synthesized and MAPS-ENTDkCl datasets) and different normalisation methods are shown in Table 3.6. From the results we can see that adding normalisation (both the waveform normalisation and the spectrogram normalisation) largely improves model's Generalisability when adapted to a new dataset, but didn't have large impact when testing on the datasets used in model training.

Chapter 4

Proposed Future Work

4.1 Study 1: More experiments on joint piano-roll and score transcription

4.1.1 Downbeat & beat tracking

Our current system is not yet fully-automatic since we use the ground truth downbeat times to separate music audio input into bars, and transcribe one bar per time. To further achieve a fully-automatic transcription system, we plan to make use of a downbeat tracking algorithm such as [12], [13], [18] to predict downbeat times from music signals.

To test the performance of the downbeat tracking algorithm, we plan to use the ASAP dataset [29] instead of the synthesized dataset we create. This is because the synthesized dataset has very little tempo variations, while the ASAP dataset is composed of real recordings of classical music pieces, which is closer to our end task.

Combining the result of beat tracking may also be helpful for note duration estimation in scores. [13] provides an implementation of joint beat and downbeat tracking. We may consider to use the beat estimates as an additional input to help our score transcription system. In Study 2, we will try to integrate downbeat and beat tracking together with score transcription into an end-to-end system.

4.1.2 Model training and evaluation on real-world data

It is necessary to train and evaluate our model on some real-world data, apart from our currently used synthesized data. This is not just real-world audio (e.g. the MAPS dataset [28]), but also real-world performances which do not have perfect tempo (e.g. the ASAP dataset [29]). The MAPS dataset is created from score MIDI files, so does not have natural performance characteristics. This is where other datasets with either MIDI performances or fully real-world human performances and recorded audio would be useful.

4.1.3 Model architecture variations

We plan to try some variations on model architectures for the joint piano-roll and score transcription model, including adding a feedback skip connection between the piano-roll output and testing the effect of using different numbers of shared convolutional layers. Feeding the piano-roll predictions to the model's score transcription part will probably help the model to better capture pitch values. On the other hand, our current system uses a fully shared convolutional stack, assuming the piano-roll transcription task and the score transcription task are highly related and share the same high-level features. However, the assumption is not ensured to be true and there exist probabilities that piano-rolls and scores need some different high-level features. By testing different numbers of shared convolutional layers, we can check whether our current design is the best.

4.1.4 Extend and optimize score representation

For now, our system does not deal with complex playing techniques such as arpeggios, trios, vibratos nor complex rhythm structures such as triplets, quintuplets. We leave the problem of playing technique labelling to Study 4. However, it is necessary to extend our current score representation to enable some complex rhythm structures (e.g. triplets) that frequently appear in classical piano music.

On the other hand, it will be helpful if we include some additional information in the score representation output that can map the scores with the piano-roll predictions note-by-note. In this way, the model performs some form of automatic alignment between the score transcriptions and the input signals, which can be helpful in model evaluation and analysis.

4.1.5 Score decoding algorithm optimization

Currently, we are simply using greedy decoding when getting the score outputs. That is, for each step, we use the best notes in the previous onset time to predict their following notes. The decoding algorithm can be improved into some form of matrix-based beam search to get better score transcription results.

4.2 Study 2: Exploring learning methodologies

The system we are going to develop in Study 1 can be seen as a preliminary work towards end-to-end complete transcription, where the "complete" means the system can transcribe a music from audio to music scores. In this study, we aim to go further in polyphonic music transcription by improving the model by trying more model architectures.

More detailed experiment proposals are described in the following.

4.2.1 Optimised attention mechanism & transformers

The attention mechanism we use in the beginning is the additive attention proposed in [3]. For each step of the score output, the attention layer fully connect each output with all inputs

and the model learns the attention weights over the whole input sequence. However, when doing transcription, we usually use the neighbouring notes before and after as a reference, instead of going through the whole song before starting to record scores. Thus, the fully connected attention layer is not necessary in an AMT system. Instead, we can use some form of multiplicative attention [52] that attends to only part of the input sequence. It can make the model more effective, unlike the additive attention in [3] who trains and predicts extremely slowly. Some variations in the implementation of multiplicative attention will be necessary. For example, size and location of the attention window of a score transcription system is not always linear with the inputs and we need to also adjust the attention window during training.

We also plan to try some more complex attention methods such as self attention and Transformers [82]. We assume this can be helpful especially in the model's decoder part since it helps the model to better learn the dependencies within the note sequences, which can result in better music grammar in score outputs.

4.2.2 Modelling long-term dependencies

Simply combining a downbeat tracker and a bar-based score transcription system does not end up in a holistic transcription model. Besides, the model's accuracy in transcribing metre will be highly dependent on the downbeat tracker used, and the transcription model will lose the chance to learn long-term dependencies in the music signal (e.g. phrase and section). In future work, we plan to integrate downbeat tracking into our score transcription system, and explore methods for the model to learn longer-term dependencies.

A significant feature in music signal is the relative timing grouped in beats and metre. In the design of music language model, metre-based time steps are usually better than frame-based time steps [89]. To take this musical feature into account, we can use temporal convolutional networks (TCNs) [48] or multi-head self attention [82], or a relation-aware self-attention [73] which is also used in [38] for music generation, to better capture long-term time dependencies in music. Another possible alternative of modelling the input and output sequence dependencies is by using 2-dimensional LSTMs as a decoder of sequence-to-sequence model, like in [1, 2]. We leave it as an open choice of experiment. We treat the use of TCNs our first choice since it is proved to be working well on dealing with very-long sequence problems [81, 25, 12].

4.2.3 Design of new loss functions

Another choice of model architecture is using a CTC loss function [68] instead of an encoder-decoder model. It can also model unequal input/output length, but we didn't use it due to its limit that the length of the output is always smaller than the length of the input (which is not valid for LilyPond representation). CTC loss is commonly used on automatic speech recognition problems [33], that share a lot of similarities with AMT. Furthermore, by using our proposed Reshaped score representation, we can ensure that the the score representation is strictly shorter than the music audio input. Thus, it's worth trying using a CTC loss function with a model structure such as a Transducer [33], and comparing the model's ability with the performance of the encoder-decoder architecture we previously used. We will need

to expand the CTC loss to support polyphonic music, or design a new loss function based on CTC loss for our system.

4.2.4 Modelling pitch dependencies

People have attempted various approaches to model the time axis during automatic music transcription. However, most AMT systems try to solve a multi-label classification problem, and consider pitches as independent classes. This is not true in a music context. Music pitches are ordered in frequencies and there exist some special relations among them, such as octaves, harmonies. When we design AMT systems as a multi-label classification task, we are actually ignoring these musical dependencies between pitches.

We propose to do some research on including pitch dependency modelling in an AMT system. A simple solution can be designing a new musically meaningful loss function that gives different penalties for classification mistakes. For example, we can use perceptual pitch distance based on chroma mapping. However, it can be hard to define a good loss function. Also, a perceptual based loss function can potentially increase the probability for AMT systems to make some specific pitch errors. As a better solution, we consider the use of multi-dimensional networks - which can model time and frequency axis at the same time.

One potential solution is using 2-dimensional LSTMs [35], similar application in automatic speech recognition can be found in [49] and [70]. Also, CNN is another straight forward network structure for modelling 2D data. In [45], Kim and Bello used a CNN-based discriminator for adversarial learning to model both time and frequency domain. We could also try similar approaches, or design other methods to jointly model time and frequency.

4.3 Study 3: Investigating evaluation metrics

As we discussed in Section 2.2.2 and Section 2.4.2, the evaluation of an transcription result is not a simple question of correct or wrong. There are a lot of music-related considerations that can influence human's perception of whether a music is transcribed nicely or badly. For example, an extra note in tone with other notes may be a less severe error compared to an extra out-of-tone note (it could also be tricky on determining if a note is in tone or not, for example, a semitone between E and D happens much more frequently than a semitone between D# and D in C major). A missing intermediate note in a chord may be less noticeable than a missing note in the main melody. Things can be more complex when it comes to sheet music. While the offset of a note is usually less noticeable when we listen to a music, it becomes very obvious when we look at a printed music (e.g. consider a situation when a piano pedal is used).

Some works investigating the perceptive aspects in AMT systems are [65, 24, 60, 90], but they are either working on frame-level or note-level evaluations. Current evaluation metrics for complete music transcription (see [23, 57, 56]) does not take human perceptual aspects into account. We would like to investigate the influence of different musical aspects in the evaluation of AMT systems by doing a human listening test, and if time permits, try to design a new evaluation metric for complete transcription that is more musically reasonable.

4.4 Study 4: Dealing with more music information

We plan to explore methods to deal with more music information in score transcription, and try to get a more "complete" transcription system. Some of the basic information include key and time signatures, which can be easily inferred from the score transcription outputs. In the following, we discuss two proposals of dealing with more complex music information. Due to time limits, we set a priority to the first one.

4.4.1 Multiple instrument music

Some works already covered multi-instruments transcription [26, 10, 53] (different types of instruments, rather than same type but more than one number of instrument), by using a multitask method or combining source separation and music transcription. However, they are both predicting a piano-roll representation, and didn't provide a score-level output. Similar methods may be combined with a score-level AMT system to generate music scores for different instruments.

4.4.2 Dynamics & performance techniques

Current transcription systems seldom provide dynamic changes and performance technique labels. However, these labels are an essential part of scores in western staff notation and other notations. Classical piano music, for example, usually have a lot of dynamic changes (e.g. p, f) and performance technique labels (e.g. glissando, vibrato, piano pedalling). Including those annotations can be helpful for those musicians who pay attention to more musical information except for the basic notes and rests.

For dynamics, we can consider adding an additional output label for note velocities. While for playing techniques, we may need to combine/integrate some playing technique recognition methods to obtain some inference. One problem of adding dynamics and performance technique annotations is that there may not exist a standard evaluation metric to check the model's performance. We leave it a question for further exploration.

4.5 Study 5: Domain shift in music data

Today, many people are using synthesized dataset for developing deep learning models due to easy access. However, synthesized datasets are usually very biased and can make models to learn the exact audio generation pattern in the synthesizer. Models' performance tend to drop a lot when applied to real recordings. Besides, datasets themselves have original distributions due to e.g. synthesizers, recording conditions, composers, music styles etc. This limit the potential of AMT systems within the dataset/similar pieces it was trained on.

We proposed to use domain adaptation methods to solve this problem. A simple choice could be using different encoders to map the domain specific data into some latent space, and then use a shared decoder to extract information about music transcription from the latent spaces (see [86] a similar approach). We could also use an adversarial network to force the discriminators to learn a shared feature space.

Chapter 5

Research Plan

5.1 Time plan

Gantt charts for my short and long-term plan can be found in Figure 5.1.

Short-term plan Before the stage 2 milestone in March of 2022, I plan to work on Study 1-3. Details listed below.

- **Study 1** - Work on more experiments on joint piano-roll and score transcription, including add downbeat tracking, test model architecture variations, extend and optimize score representations, and optimize score decoding algorithm;
- **Study 2.1** - Tryout optimized attention mechanism and transformers;
- **Study 2.2** - Explore methods to model long-term dependencies;
- **Study 2.3** - Compare encoder-decoder structure and using a CTC loss function;
- **Study 2.4** - Try methods for modelling pitch dependencies;
- **Study 3** - Investigate evaluation metrics for score transcription.

Long-term plan We aim to work on the proposed studies detailed in Chapter 4. Followed by a writing up stage in the final year. Detailed time plans together with milestones and potential publication plans are in Figure 5.1.

5.2 Publication targets

We plan to publish related research to conferences such as WASPAA, ISMIR, ICASSP, MLSP. Other choices including Music Encoding Conference, Machine Learning for Music Discovery Workshop (ICML workshop), European Signal Processing Conference (EUSIPCO), International Workshop on Machine Learning and Music, International Conference on Digital Libraries for Musicology (DLfM) and machine learning conferences such as NeurIPS, ICLR and ICML. In the short term, we aim to submit a paper to ISMIR, deadline in April next year.

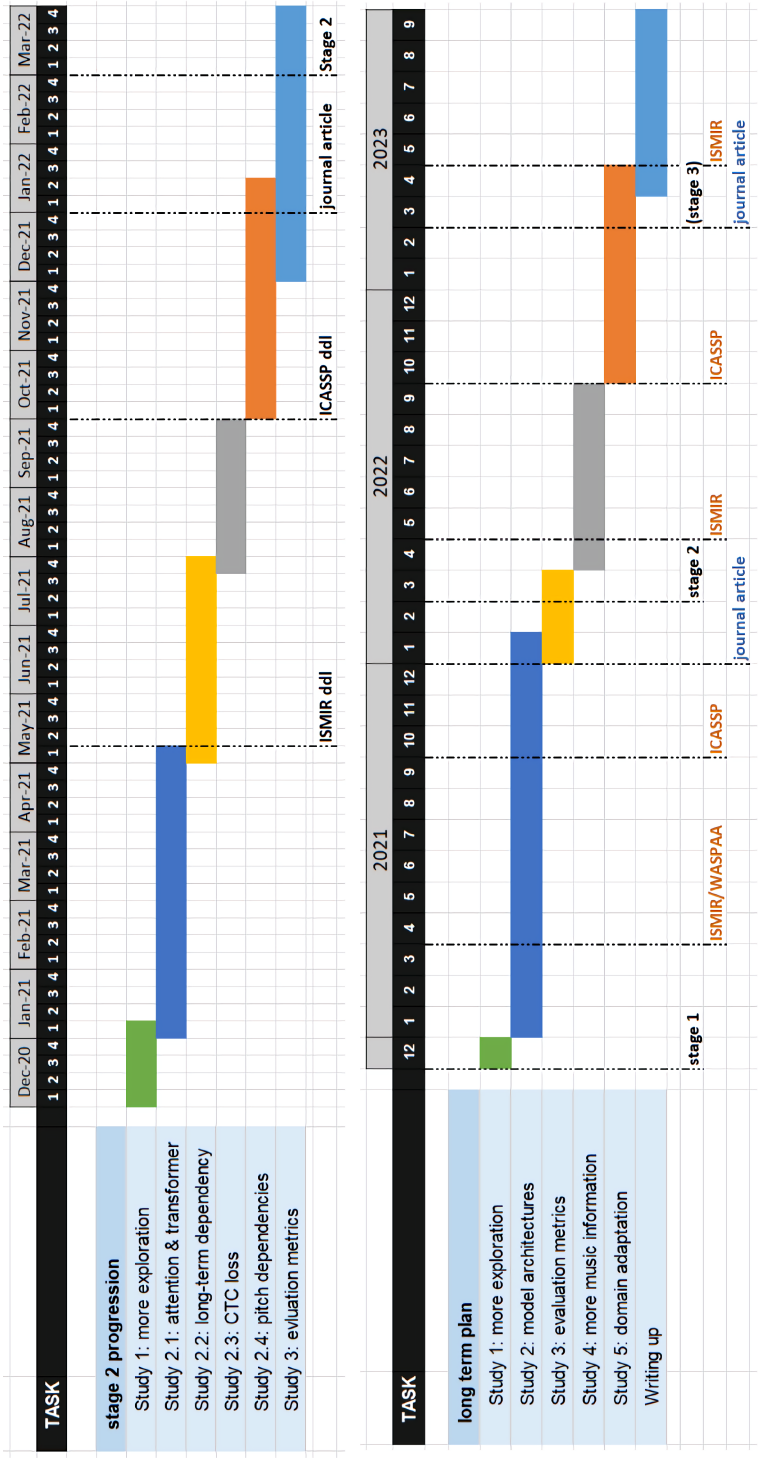


Figure 5.1: Gantt chart for research plan.

Towards the end of the PhD and some subtasks, we aim to submit relevant journal papers. Target journals include TISMIR, IEEE/ACM TASLP, JNMR and EURASIP JASMP.

References

[1] Parnia Bahar, Christopher Brix, and Hermann Ney. Towards Two-Dimensional Sequence to Sequence Model in Neural Machine Translation. *arXiv preprint*, pages 3009–3015, 2018.

[2] Parnia Bahar, Albert Zeyer, Ralf Schluter, and Hermann Ney. On Using 2D Sequence-to-sequence Models for Speech Recognition. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2019-May:5671–5675, 2019.

[3] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–15, 2015.

[4] Mert Bay, Andreas F. Ehmann, and J. Stephen Downie. Evaluation of multiple-F0 estimation and tracking systems. In *ISMIR, International Society for Music Information Retrieval Conference*, pages 315–320, 2009.

[5] Emmanouil Benetos, Simon Dixon, Zhiyao Duan, and Sebastian Ewert. Automatic Music Transcription: An Overview. *IEEE Signal Processing Magazine*, 36(1):20–30, 2019.

[6] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. Automatic music transcription: Challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, 2013.

[7] Emmanouil Benetos and Andre Holzapfel. Automatic Transcription of Turkish Makam Music. *ISMIR*, 2600:0–2, 2013.

[8] Rachel Bittner and Juan J Bosch. Generalised Metrics for Single-F0 Estimation Evaluation. *ISMIR*, pages 738–745, 2019.

[9] Rachel Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Bello. MedleyDB: A multitrack dataset for annotation - intensive mir research. *International Society for Music Information Retrieval Conference*, (Ismir):155–160, 2014.

[10] Rachel M. Bittner, Brian McFee, and Juan P. Bello. Multitask Learning for Fundamental Frequency Estimation in Music. *arXiv preprint arXiv:1809.00381*, pages 1–13, 2018.

[11] Rachel M. Bittner, Brian McFee, Justin Salamon, Peter Li, and Juan P. Bello. Deep salience representations for F0 estimation in polyphonic music. In *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017*, pages 63–70, 2017.

[12] Sebastian Böck and Matthew E P Davies. Deconstruct, Analyse, Reconstruct: How To Improve Tempo, Beat, and Downbeat Estimation. In *ISMIR, International Society for Music Information Retrieval Conference*, 2020.

[13] Sebastian Böck, Florian Krebs, and Gerhard Widmer. Joint beat and downbeat tracking with recurrent neural networks. *Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR 2016*, pages 255–261, 2016.

[14] Sebastian Bock and Markus Schedl. Polyphonic Piano Note Transcription with Recurrent Neural Networks. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 121–124, Kyoto, Japan, 2012.

[15] Juan J. Bosch, Ricard Marxer, and Emilia Gómez. Evaluation and combination of pitch estimation methods for melody extraction in symphonic classical music. *Journal of New Music Research*, 45(2):101–117, 2016.

[16] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, 2(Cd):1159–1166, 2012.

[17] Judith C. Brown. Calculation of a constant Q spectral transform. *Journal of the Acoustical Society of America*, 89(1):425–434, 1991.

[18] Chris Cannam, Emmanouil Benetos, Matthias Mauch, Matthew E. P. Davies, Simon Dixon, Christian Landone, Katy Noland, Dan Stowell, and Matthew Davies. Mirex 2015: Vamp Plugins From the Centre for Digital Music. *ISMIR, International Society for Music Information Retrieval Conference*, pages 0–3, 2015.

[19] Ralf Gunter Correa Carvalho and Paris Smaragdis. Towards End-to-End Polyphonic Music Transcription: Transforming Music Audio Directly to A Score. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, volume 2017-Octob, pages 151–155, 2017.

[20] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. Listen, Attend and Spell. pages 1–16, 2015.

[21] Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. End-to-end Continuous Speech Recognition using Attention-based Recurrent NN: First Results. pages 1–10, 2014.

[22] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-Based Models for Speech Recognition. *Advances in Neural Information Processing Systems*, 2015-Janua:577–585, jun 2015.

[23] Andrea Cogliati and Zhiyao Duan. A Metric for Music Notation Transcription Accuracy. *ISMIR*, pages 407–413, 2017.

[24] Adrien Daniel, Valentin Emiya, and Bertrand David. Perceptually-based evaluation of the errors usually made when automatically transcribing music. *ISMIR*, (May):550–555, 2008.

[25] Matthew E.P. Davies and Sebastian Böck. Temporal convolutional networks for musical audio beat tracking. *European Signal Processing Conference*, 2019-Septe, 2019.

[26] Zhiyao Duan, Jinyu Han, and Bryan Pardo. Multi-pitch streaming of harmonic sound mixtures. *IEEE Transactions on Audio, Speech and Language Processing*, 22(1):138–150, 2014.

[27] Zhiyao Duan, Bryan Pardo, and Changshui Zhang. Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions. *IEEE Transactions on Audio, Speech and Language Processing*, 18(8):2121–2133, 2010.

[28] Valentin Emiya, Roland Badeau, and Bertrand David. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE Transactions on Audio, Speech and Language Processing*, 18(6):1643–1654, 2010.

[29] Francesco Foscarin, Andrew Mcleod, Philippe Rigaux, Florent Jacquemard, Francesco Foscarin, Andrew Mcleod, Philippe Rigaux, Florent Jacquemard, Masahiko Sakai Asap, Francesco Foscarin, Andrew Mcleod, and Philippe Rigaux. ASAP : a dataset of aligned scores and performances for piano transcription. In *ISMIR, International Society for Music Information Retrieval Conference*, 2020.

[30] Yoav Goldberg. *Neural Network Methods for Natural Language Processing*. Morgan & Claypool, 2017.

[31] Emilia Gomez and Jordi Bonada. Towards Computer-Assisted Flamenco Transcription: An Experimental Comparison of Automatic Transcription Algorithms as Applied to A Cappella Singing. *Computer Music Journal*, 37(4):10–23, 2013.

[32] Masataka Goto and Hiroki Hashiguchi. RWC Music Database: Popular, Classical, and Jazz Music Databases. *ISMIR*, pages 1–2, 2002.

[33] Alex Graves. Sequence Transduction with Recurrent Neural Networks. *arXiv preprint arXiv:1211.3711*, 2012.

[34] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. *ACM International Conference Proceeding Series*, 148:369–376, 2006.

[35] Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. Multi-dimensional recurrent neural networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4668 LNCS(PART 1):549–558, 2007.

[36] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck. Onsets and frames: Dual-objective piano transcription. In *ISMIR, International Society for Music Information Retrieval Conference*, pages 50–57, 2018.

[37] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset. *ICLR*, pages 1–12, 2019.

[38] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music Transformer: Generating Music with Long-Term Structure. *ICLR*, pages 1–14, 2019.

[39] Eric J. Humphrey, Simon Durand, and Brian McFee. OpenMIC-2018: An open dataset for multiple instrument recognition. *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018*, pages 438–444, 2018.

[40] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016.

[41] Rainer Kelz, Sebastian Bock, and Cierhard Widnaer. Multitask learning for polyphonic piano transcription, a case study. In *Proceedings - 2019 International Workshop on Multilayer Music Representation and Processing, MMRP 2019*, pages 85–91, 2019.

[42] Rainer Kelz, Matthias Dorfer, Filip Korzeniowski, Sebastian Böck, Andreas Arzt, and Gerhard Widmer. On the Potential of Simple Framewise Approaches to Piano Transcription. *ISMIR*, 2016.

[43] Rainer Kelz and Gerhard Widmer. An experimental analysis of the entanglement problem in neural-network-based music transcription systems. *Proceedings of the AES International Conference*, 22-24-June:194–201, 2017.

[44] Rainer Kelz and Gerhard Widmer. Towards Interpretable Polyphonic Transcription with Invertible Neural Networks. In *ISMIR*, 2019.

[45] Jong Wook Kim and Juan Pablo Bello. Adversarial Learning for Improved Onsets and Frames Music Transcription. *ISMIR*, 2019.

[46] Anssi Klapuri and Tuomas Virtanen. Automatic Music Transcription. *Computer Music Journal*, 1(4):24–31, 1977.

[47] Chih-yi Kuan, Li Su, Yu-hao Chin, and Jia-ching Wang. Multi-Pitch Streaming of Interwoven Streams. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2017*, pages 311–315, 2017.

[48] Colin Lea, René Vidal, Austin Reiter, and Gregory D. Hager. Temporal convolutional networks: A unified approach to action segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9915 LNCS:47–54, 2016.

[49] Jinyu Li, Abdelrahman Mohamed, Geoffrey Zweig, and Yifan Gong. LSTM time and frequency recurrence for automatic speech recognition. *2015 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2015 - Proceedings*, pages 187–191, 2016.

[50] Yin-Jyun Luo and Li Su. Learning Domain-Adaptive Latent Representations of Music Signals Using Variational Autoencoders. *ISMIR*, pages 653–660, 2018.

[51] Minh Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, (c):1–10, 2016.

[52] Minh Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, 2015.

[53] Ethan Manilow, Prem Seetharaman, and Bryan Pardo. Simultaneous Separation and Transcription of Mixtures with Multiple Polyphonic and Percussive Instruments. *arXiv preprint*, 2019.

[54] Ethan Manilow, Gordon Wichern, Prem Seetharaman, and Jonathan Le Roux. Cutting Music Source Separation Some Slakh: A Dataset to Study the Impact of Training Data Quality and Quantity. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2019.

[55] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa/librosa: 0.8.0 (Version 0.8.0), 2020.

[56] Andrew Mcleod. Evaluating Non-Aligned Musical Score Transcriptions with MV2H. *ISMIR Late-Breaking/Demo*, pages 1–2, 2019.

[57] Andrew Mcleod and Mark Steedman. Evaluating Automatic Polyphonic Music Transcription. In *ISMIR, International Society for Music Information Retrieval Conference*, pages 42–49, 2018.

[58] Emilio Molina, Ana M. Barbancho, Lorenzo J. Tardón, and Isabel Barbancho. Evaluation framework for automatic singing transcription. *ISMIR*, pages 567–572, 2014.

[59] Meinard Müller. *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*. Springer, 2015.

[60] Eita Nakamura, Emmanouil Benetos, Kazuyoshi Yoshii, and Simon Dixon. Towards Complete Polyphonic Music Transcription: Integrating Multi-Pitch Detection and Rhythm Quantization. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2018-April, pages 101–105. IEEE, 2018.

[61] Eita Nakamura, Kazuyoshi Yoshii, and Simon Dixon. Note Value Recognition for Piano Transcription Using Markov Random Fields. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 25(9):1542–1554, 2017.

[62] Juhan Nam, Jiquan Ngiam, Honglak Lee, and Malcolm Slaney. A classification-based polyphonic piano transcription approach using learned feature representations. In *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011*, number November 2014, pages 175–180, 2011.

[63] Han-Wen Nienhuys and Jan Nieuwenhuizen. Lilypond, a System for Automated Music Engraving. In *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003)*, pages 167–171, 2003.

[64] Ryo Nishikimi, Eita Nakamura, Satoru Fukayama, Masataka Goto, and Kazuyoshi Yoshii. Automatic Singing Transcription Based on Encoder-decoder Recurrent Neural Networks with a Weakly-supervised Attention Mechanism. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, volume 2019-May, pages 161–165, 2019.

[65] Graham E. Poliner and Daniel P.W. Ellis. A discriminative model for polyphonic piano transcription. *Eurasip Journal on Advances in Signal Processing*, 2007, 2007.

[66] Colin Raffel and Daniel P. W. Ellis. Intuitive Analysis, Creation and Manipulation of Midi with pretty _ midi. *Proceedings of the International Society for Music Information Retrieval Conference*, 2014.

[67] Miguel A Román, Antonio Pertusa, and Jorge Calvo-Zaragoza. An End-To-End Framework for Audio-To-Score Music Transcription on Monophonic Excerpts. In *ISMIR, International Society for Music Information Retrieval Conference*, pages 34–41, 2018.

[68] Miguel A Román, Antonio Pertusa, and Jorge Calvo-zaragoza. A Holistic Approach to Polyphonic Music Transcription with Neural Networks. In *ISMIR, International Society for Music Information Retrieval Conference*, pages 731–737, 2019.

[69] Sebastian Ruder. An Overview of Multi-Task Learning in Deep Neural Networks. *arXiv preprint arXiv:1706.05098*, 2017.

[70] Tara N. Sainath and Bo Li. Modeling time-frequency patterns with LSTM vs. convolutional architectures for LVCSR tasks. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 08-12-Sept:813–817, 2016.

[71] Julian Salazar, Katrin Kirchhoff, and Zhiheng Huang. Self-attention Networks for Connectionist Temporal Classification in Speech Recognition. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2019-May:7115–7119, 2019.

[72] Christian Schörkhuber, Anssi Klapuri, Nicki Holighaus, and Monika Dörfler. A Matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution. In *Proceedings of the AES International Conference*, pages 232–239, 2014.

[73] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-Attention with Relative Position Representations. *arXiv preprint arXiv:1803.02155*, 2018.

[74] Tao Shen, Jing Jiang, Tianyi Zhou, Shirui Pan, Guodong Long, and Chengqi Zhang. Disan: Directional self-attention network for RnN/CNN-free language understanding. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 5446–5455, 2018.

[75] Siddharth Sigtia, Emmanouil Benetos, and Simon DIxon. An end-to-end neural network for polyphonic piano music transcription. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 24(5):927–939, 2016.

[76] Li Su and Yi Hsuan Yang. Combining spectral and temporal representations for multipitch estimation of polyphonic music. *IEEE Transactions on Audio, Speech and Language Processing*, 23(10):1600–1612, 2015.

[77] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 4:3104–3112, 2014.

[78] John Thickstun, Zaid Harchaoui, and Sham Kakade. Learning Features of Music from Scratch. *ICLR*, pages 1–14, 2017.

[79] Stefan Uhlich, Marcello Porcu, Franck Giron, Michael Enenkl, Thomas Kemp, Naoya Takahashi, and Yuki Mitsufuji. Improving music source separation based on deep neural networks through data augmentation and network blending. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, (October):261–265, 2017.

[80] Karen Ullrich and Eelco van der Wel. Music Transcription With Convolutional Sequence-to-Sequence Models. *International Conference on Learning Reprsentations (rejected)*, pages 1–9, 2018.

[81] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. *arXiv preprint*, arXiv: 160:1–15, 2016.

[82] Ashish Vaswani, Google Brain, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *Advances in neural information processing systems*, (Nips):5998–6008, 2017.

[83] Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. Grammar as a foreign language. *Advances in Neural Information Processing Systems*, 2015-Janua:2773–2781, 2015.

[84] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. *arXiv preprint arXiv:1905.09418*, 2019.

[85] Qi Wang, Ruohua Zhou, and Yonghong Yan. Polyphonic piano transcription with a note-based music language model. *Applied Sciences (Switzerland)*, 8(3), 2018.

[86] Wei Wei, Hongning Zhu, Emmanouil Benetos, and Ye Wang. A-CRNN: A Domain Adaptation Model for Sound Event Detection. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2020.

[87] Baosong Yang, Longyue Wang, Derek F. Wong, Lidia S. Chao, and Zhaopeng Tu. Convolutional Self-Attention Networks. pages 4040–4045, 2019.

[88] Adrien Ycart and Emmanouil Benetos. A study on LSTM networks for polyphonic music sequence modelling. *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017*, pages 421–427, 2017.

[89] Adrien Ycart and Emmanouil Benetos. Polyphonic Music Sequence Transduction with Meter-Constrained LSTM Networks. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2018-April:386–390, 2018.

[90] Adrien Ycart, Lele Liu, Emmanouil Benetos, and Marcus T. Pearce. Investigating the Perceptual Validity of Evaluation Metrics for Automatic Piano Music Transcription. *Transactions of the International Society for Music Information Retrieval*, 3(1):68–81, 2020.

[91] Adrien Ycart, Andrew Mcleod, Emmanouil Benetos, and Kazuyoshi Yoshii. Blending Acoustic and Language Model Predictions for Automatic Music Transcription. *ISMIR*, 2019.

[92] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:12744–12753, 2019.

[93] Lixia Zhao, Lingyan Zhu, Shuyan Zhao, and Xinxin Ma. Sequestration and bioavailability of perfluoroalkyl acids (PFAAs) in soils: Implications for their underestimated risk. *Science of the Total Environment*, 572:169–176, 2016.

Appendix I

Covid-19 impact Covid-19 did have a non-trivial impact on my PhD life. One of the most important aspect is the change to working from home. It takes me about one month to get used to the new situation. It is much harder for me to keep concentration at home than in the office. Another big problem I and many other peers found is that due to the leave of the IT people, our school's servers went down extremely frequently during the summer, which greatly slows down our experiment progress.

Training courses Courses I currently have taken or am now taking are listed in Table 5.1.

Table 5.1: Training courses so far

Code	**AIM Modules**	**Grade**
ECS7007	Research Methods and Responsible Innovation	78.9
ECS7002	Artificial Intelligence in Games	72.2
ECS7013	Deep Learning for Audio and Music	72.4
ECS7001	Neural Networks and NLP	68.1
ECS764	Applied Statistics	73.3 (currently)
Code	**CPD Training Courses**	**Status**
RD101	Getting Started with Your PhD	✓
RD100	Working with Your Supervisor	✓
RD208	Making a Poster Presentation	✓
RD105	Making the Most of Your First Academic Conference	✓
RD005	Speed-reading for Researchers	✓
PHD107	Becoming Writers - A six-step guide to more effective writing	✓

Appendix II:
Personal development plan

Centre for Doctoral Training in AI and Music
Personal Development Plan
This is a living document. It should be updated each year throughout the PhD.
Each time this document is updated, please upload it to your Supervision Log on MySIS.

Name: Lele Liu

Primary Supervisor: Dr. Emmanouil Benetos
Secondary Supervisor: Prof. Simon Dixon
Supervisor: Dr. Veronica Morfi
Independent Assessor: Dr. Marcus Pearce

Date: 15th Nov, 2019

Brief overview of PhD research project and major accomplishments expected:
(about a half page)

Proposed Research Topic:
Towards Complete Automatic Music Transcription with Deep Neural networks

Automatic Music Transcription (AMT) aims to convert music signal into some form of music annotations (e.g. piano-rolls, symbolic music scores). AMT is considered to be one of the most difficult problem in Music Information Retrieval (MIR). Common methodologies for AMT include Non-negative Matrix Factorization (NMF) methods and Neural Network (NN) methods. Since deep learning showed potential for a lot of AI related problems, there are more and more explorations on NN methods for AMT problem. State-of-the-art systems with NN methods have shown good performance on piano-roll prediction. However, for a long time, research in AMT has largely focused on multi-pitch detection; there is limited discussion on how to obtain a machine- or human-readable score transcription.

In our project, we aim to mainly focus on deep learning methods for complete AMT, paying special attention to end-to-end transcription. We plan to explore methods to obtain a readable score-format transcription output. We will first explore end-to-end methods for score transcription, and then extend our focus to a wider field of complete transcription (e.g. evaluation metrics, dynamics in score, non-western music, domain adaptation for different recording conditions/sound fonts, multi-instrument transcription).

We plan to do our project by exploring the following aspects:
- ✓ **Datasets** - create datasets with mapped music audio and scores.
- ✓ **End-to-end AMT system** - explore methodologies (input/output representations, model architectures) for an end-to-end model that converts music signal into music scores.
- ✓ **Evaluation metrics** - explore new evaluation metrics for complete AMT, including perceptual metrics to distinguish between different types of errors (e.g. missing/extra notes, semitone/octave errors, voice separation errors, duration errors, etc.)
- ✓ **Domain adaptation and model transferability** - test the fitness of domain adaptation methods on challenges such as different recording conditions/instruments.
- ✓ **Non-western music** - experiment methods for transcribing pitched non-western music, when it is impossible to collect large dataset for the specific type of music.
- ✓ **More music symbols** - add dynamics/tempos/playing techniques marks in transcription to make the scores more “complete”.
- ✓ **Multi-instrument score transcription** - extend the single-instrument system to multi-instrument music transcription.

Long-term (5+ year) career objectives:

(2-3 paragraphs)

I want to stay in academia (doing postdoc research) after finishing my PhD. According to the regulation of CSC, I can stay outside China for maximum 2-years of postdoc research, after that, I will go back China. I would like to spend the 2 years after graduation in postdoc research work. I would also like to continue research work after that.

Regarding the research area, I want to continue with some music-related or more generally, machine learning related topics. I currently have a slight preference on working in an academia institute, but leave it an open choice depending on further experience. I'll try to know more about research institutes related to my research area during my PhD.

What skills or training would help you achieve these long-term career objectives?
(1 paragraph)

Skills:

- ✓ **Critical Thinking** - I need to improve my critical thinking skills, which is essential for my PhD and further research career.
- ✓ **Research insight** - I need to do more literature reading and go to seminars to discuss and know more about current research.
- ✓ **Reading and writing skills** - I need to improve my reading and writing skills. Practice deliberately to better understand ideas and discover limitations in papers, and write in a more professional and well-organized style.
- ✓ **Project management** - I need better project/time management skills to keep project in progress as expected.
- ✓ **Communication skills** - improving my skills to presenting and communicating my work to audiences and peers (whether in seminars, conferences, to the public or in daily discussions) would be important. I also want to develop some skills on presenting my research to the wider public and schools (public engagement).

Training:

- ✓ **Summer schools** - probably some summer schools on specific topics such as deep learning, music processing or data science.
- ✓ **QMUL courses and essential courses** - attending the required courses and research skills development courses at QMUL.
- ✓ **Internships** - I would like to do some internships during my PhD (the ByteDance internship, which is planned to be half-year part-time from June 2021).
- ✓ **Research exchange or visits** - I would like to spend some time in another research institute in my third/fourth year. It could be quarter-year long. It's good to know other people in the field and more about how different research group works.

<u>Short-term (1-2 year) objectives and training</u>

Expected research results: publications; conferences, workshops or seminars to attend:
(1-2 paragraphs)

I have submitted a paper to ICASSP 2021, and will be presenting the same work at DMRN this December. Early next year, I plan to submit a paper to ISMIR 2021 around April.

In general, I will submit to conferences such as WASPAA, ISMIR, ICASSP, MLSP. Other choices including Music Encoding Conference, Machine Learning for Music Discovery Workshop (ICML workshop), European Signal Processing Conference (EUSIPCO), International Workshop on Machine Learning and Music and International Conference on Digital Libraries for Musicology (DLfM). Towards the end of this thesis (or some subtasks), we aim to submit relevant journal papers. Target journals could be TISMIR, IEEE/ACM TASLP, JNMR and EURASIP JASMP.

New technical skills or expertise to acquire for the PhD research:
(1-2 paragraphs)

- ✓ **Signal processing** - learn more about signal processing methodologies, getting more familiar with signal processing libraries in python (e.g. librosa). Learn more about music signal processing.
- ✓ **Symbolic music** - learn about music symbolics, especially stuff related to notation level transcription and related toolboxes.
- ✓ **Deep learning** - learn and get more familiar with deep learning and machine learning fundamentals, state-of-the-art deep learning methodologies, and deep learning pipeline.
- ✓ **Programming skill** - getting more familiar with programming for large projects, as well as building large deep learning networks with pytorch. It's also necessary to get familiar with reproducible machine learning experiments.
- ✓ **Statistics** - gaining more in-depth understanding about statistical analysis.

Other skills or professional training:
(1-2 paragraphs)

- ✓ **Communication skills** - practice communication skills to present my research ideas and discuss with other researchers. I would like to present at informal seminars (e.g. internal AIM seminars, mini-group meetings, music informatics/machine listening theme meetings, C4DM seminars, and seminars outside QMUL). I also plan to attend the CPD course "[QMA201] Presenting your research to an audience".
- ✓ **Organisational skills** - I am joining DMRN organisation this term, and am the AIM PhD representative on the EECS side. I would like to learn more about organising a conference and how to deal with students' needs in the school level.

Expected selection of taught modules
You must take 6 modules during years 1-2 of the PhD, typically 4 modules in Year 1 and 2 modules in Year 2. Please see the Module Selection Guide for more detail. List module codes and intended semesters below:

Year 1 Semester 1:

- ECS7007 Research Methods and Responsible Innovation (78.9)
- ECS7002 Artificial Intelligence in Games (72.2)

Year 1 Semester 2:

- ECS7013 Deep Learning for Audio and Music (72.4)
- ECS7001 Neural Networks and NLP (68.1)

Year 2 Semester 1:

- ECS764P Applied Statistics (currently 73.3)

Year 2 Semester 2:

- ECS7022 Computational Creativity

Expected selection of training courses
Please see the AIM QMplus page (https://qmplus.qmul.ac.uk/course/view.php?id=12594) for a list of recommend courses. Further information can be found on the CAPD website (https://academicdevelopment.qmul.ac.uk/bookings/). Some of the courses are compulsory for AIM CDT students (see QMplus). Please indicate a further 2 courses of potential interest this year.

Year 1:

- **[done] Getting Started with Your PhD (RD101)**
- **[done] Working with Your Supervisor (RD100)**
- Critical Thinking (RD104)

- [done] Making a Poster Presentation (RD208)
- [done] Speed-reading for Researchers (RD109)
- [done] Making the Most of Your First Academic Conference (RD105)

Year 2:

- **Understanding Research Impact (DC200)**
- **Presenting Research to an Audience (RD201)**
- **IP & Patent Training**
- **Critical Thinking (RD104)**
- [done] Becoming Writers - A six-step guide to more effective writing
- Managing Your Time and Workload (RD204)
- [booked] Social Enterprise and Open Research (RI010)
- [booked] Introduction to Impact (RI004)
- [booked] Planning for an academic career (RD301)
- How to Promote Research to the Media (RS012)
- Unconscious Bias (PD172)
- Additionally, I am planning to go to some courses in the Turing's Connection Programme, including Research software Engineering (booked), joining some interest groups, and potentially others not open yet.

Year 3:

- **Academic Career Planning for PhD Student (RC202)**
- **Public Engagement Masterclass (RS450)**
- **Commercialisation and Entrepreneurship**
- Evaluating Public Engagement and Impact (RD400)
- Reading Strategically and Analytically (RD203)

Year 4:

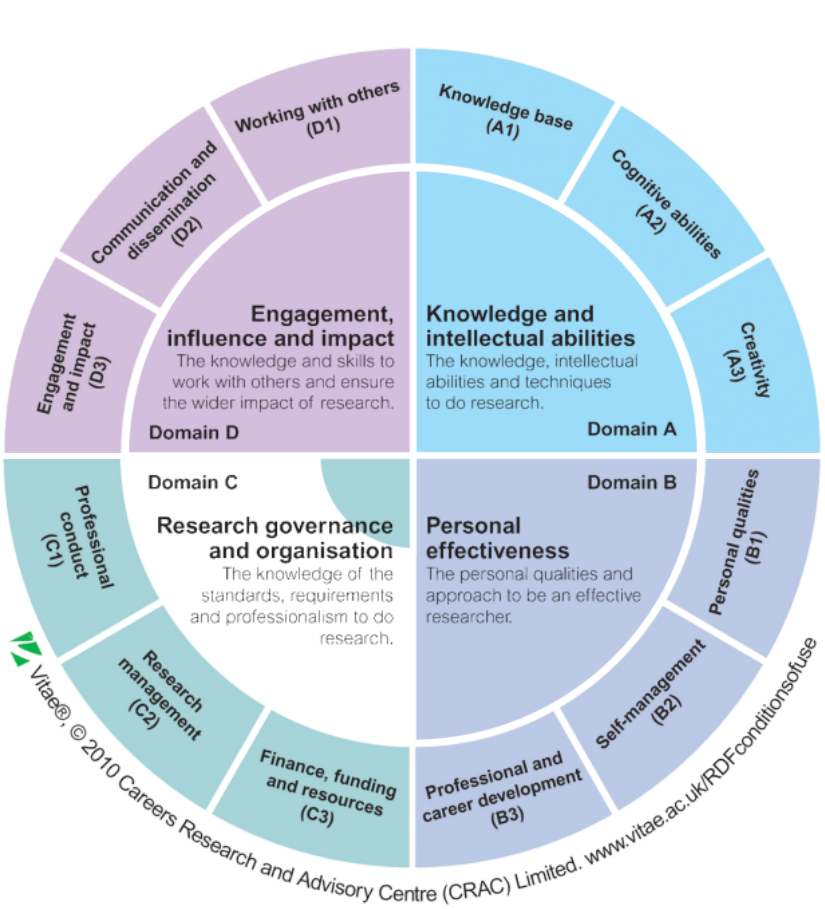
- **Writing Your Thesis (RD301)**
- **Preparing for Your Viva (RD302)**
- Interview Skills for PhD Students (RD326)
- Preparing Your CV for Fellowship Applications (RS003)
- Research Finance, Impact & Leadership

Are there any other resources or training that you would like to see the CDT provide?

I would like to have some internship opportunities (e.g. the ByteDance one starting next summer), as well as some research exchange/visit opportunities in the third/fourth year (to research groups such as Alicante, Spain, and McGill, Canada and Kyoto, Japan). Additionally, I would be very happy about some external research training opportunities like the Turing's connection programme, summer schools, etc.

Miss L Liu (130800488)

Progress



Personal Details

Full Name:	Lele Liu	**Gender:**	Female
Username:	acw487	**Email:**	lele.liu@qmul.ac.uk
Telephone:	+447421464123	**Mobile:**	+8618301559812
Enrolment Status:	R-E-E	**Programme:**	RRPF-QMCOMT3 PhD FT Computer Science with a Taught Component 5 years (EPSRC)
Course Name:	PhD FT Artificial Intelligence and Music	**Award Code:**	RP
Start Date:	19-Sep-2019	**Expected End Date:**	19-Sep-2024
Route:	RSAIM		
Faculty:	Science and Engineering	**School:**	School of Electronic Engineering and Computer Science
Department:	School of Electronic Engineering and Computer Science - Department of Computer Science		

Supervisors

Title	Given Names	Last Name	Telephone	Email	Active
Dr	Marcus Thomas	Pearce		marcus.pearce@qmul.ac.uk	true
Prof	Simon Edmund	Dixon		s.e.dixon@qmul.ac.uk	true
Dr	Emmanouil	Benetos		emmanouil.benetos@qmul.ac.uk	true
Ms	Gnostothea-Veroniki	Morfi			true

Points Summary

Year	Type	Pts: A	B	C	D	Total	Cap: A	B	C	D	Total
0	**Course/event attendance sub-total**	**260.0**	**0.0**	**0.0**	**0.0**	**260.0**					
	Teaching/demonstrating/marking/preparation	0.0	15.0	0.0	15.0	30.0					
	Teaching sub-total	**0.0**	**15.0**	**0.0**	**15.0**	**30.0**					
	Year 0 Total (with caps applied)	260.0	15.0	0.0	15.0	290.0					
1st	*Doctoral College event/course*	0.0	3.0	4.0	3.0	10.0					
	Researcher Development Course	0.0	7.0	1.0	7.0	15.0					
	Course/event attendance sub-total	**0.0**	**10.0**	**5.0**	**10.0**	**25.0**					
	Conference Presentation (Poster)	3.0	3.0	0.0	4.0	10.0					

Year	Type	Pts:	A	B	C	D	Total	Cap:	A	B	C	D	Total
	Giving presentations sub-total		**3.0**	**3.0**	**0.0**	**4.0**	**10.0**						
	Refereed Publication (Journal Paper, Book chapter, not abstract) acceptance		2.0	0.0	0.0	8.0	10.0						
	Written publications sub-total		**2.0**	**0.0**	**0.0**	**8.0**	**10.0**						
	Year 1 Total (with caps applied)		**5.0**	**13.0**	**5.0**	**22.0**	**45.0**						
2nd	*Researcher Development Course*		0.0	4.0	0.0	2.0	6.0						
	Course/event attendance sub-total		**0.0**	**4.0**	**0.0**	**2.0**	**6.0**						
	Year 2 Total (with caps applied)		**0.0**	**4.0**	**0.0**	**2.0**	**6.0**						
Total	*Core research knowledge or methods course (e.g. LTCC, IALS courses, masters lectures)*		100.0	0.0	0.0	0.0	100.0		100.0				100.0
	Doctoral College event/course		0.0	3.0	4.0	3.0	10.0						
	Researcher Development Course		0.0	11.0	1.0	9.0	21.0						
	Course/event attendance sub-total		**100.0**	**14.0**	**5.0**	**12.0**	**131.0**						
	Conference Presentation (Poster)		3.0	3.0	0.0	4.0	10.0		9.0	9.0		12.0	30.0
	Giving presentations sub-total		**3.0**	**3.0**	**0.0**	**4.0**	**10.0**						
	Teaching/demonstrating/marking/preparation		0.0	15.0	0.0	15.0	30.0			15.0		15.0	30.0
	Teaching sub-total		**0.0**	**15.0**	**0.0**	**15.0**	**30.0**						
	Refereed Publication (Journal Paper, Book chapter, not abstract) acceptance		2.0	0.0	0.0	8.0	10.0		4.0			16.0	20.0
	Written publications sub-total		**2.0**	**0.0**	**0.0**	**8.0**	**10.0**						
	Total (with caps applied)		**105.0**	**32.0**	**5.0**	**39.0**	**181.0**						
	Grand Total (with caps applied)		**105.0**	**32.0**	**5.0**	**39.0**	**181.0**						
	Target		60.0	20.0	15.0	30.0	210.0						

Pending Activities

Nothing found to display

Activity Record

Type	Code	Title	Provider	From	To	Hours	A	B	C	D	Total
Teaching/demonstrating/marking/preparation		demonstrator machine learning	School of EECS	16-Sep-2019 00:00	09-Dec-2019 00:00	30.0	0.0	15.0	0.0	15.0	30.0
Core research knowledge or methods course (e.g. LTCC, IALS courses, masters lectures)	ECS7001	Neural Networks and NLP	School of EECS	16-Sep-2019 00:00	31-Jan-2020 00:00	60.0	60.0	0.0	0.0	0.0	60.0
Core research knowledge or methods course (e.g. LTCC, IALS courses, masters lectures)	ECS7013	Deep Learning for Audio and Music	School of EECS	16-Sep-2019 00:00	31-Jan-2020 00:00	60.0	60.0	0.0	0.0	0.0	60.0
Core research knowledge or methods course (e.g. LTCC, IALS courses, masters lectures)	ECS7002	Artificial Intelligence in Games	School of EECS	16-Sep-2019 00:00	31-Jan-2020 00:00	60.0	60.0	0.0	0.0	0.0	60.0
Core research knowledge or methods course (e.g. LTCC, IALS courses, masters lectures)	ECS7007	Research Methods and Responsible Innovation	School of EECS	16-Sep-2019 00:00	31-Jan-2020 00:00	80.0	80.0	0.0	0.0	0.0	80.0
Doctoral College event/course	DC102	International PhD Student Welcome	Doctoral College	30-Sep-2019 13:30	30-Sep-2019 17:00	0.0	0.0	1.0	1.0	1.0	3.0
Doctoral College event/course	DC100	PhD Induction	Doctoral College	03-Oct-2019 09:30	03-Oct-2019 17:00	0.0	0.0	2.0	3.0	2.0	7.0
Researcher Development Course	RD100	Working With Your Supervisor	Researcher Development	08-Nov-2019 10:00	08-Nov-2019 13:00	0.0	0.0	4.0	0.0	2.0	6.0
Researcher Development Course	RD101	Getting Started with your PhD	Researcher Development	18-Nov-2019 10:00	18-Nov-2019 13:00	0.0	0.0	1.0	1.0	1.0	3.0
Conference Presentation (Poster)		DMRN+14 poster presentation & conference attendence	C4DM, School of EECS, Queen Mary University of London	17-Dec-2019 00:00	17-Dec-2019 00:00	6.0	3.0	3.0	0.0	4.0	10.0
Researcher Development Course	RD208	Making a Poster Presentation	Researcher Development	10-Feb-2020 10:00	10-Feb-2020 13:00	0.0	0.0	0.0	0.0	3.0	3.0
Researcher Development Course	RD105	Making the Most of Your First Academic Conference	Researcher Development	03-Mar-2020 13:30	03-Mar-2020 16:30	0.0	0.0	2.0	0.0	1.0	3.0
Refereed Publication (Journal Paper, Book chapter, not abstract) acceptance		TISMIR paper	ISMIR	20-Apr-2020 00:00	20-Apr-2020 00:00	0.0	2.0	0.0	0.0	8.0	10.0
Researcher Development Course	RD-QMA-005	Speed-reading for Researchers (2-part course)	Researcher Development	01-Oct-2020 10:00	02-Oct-2020 13:00	0.0	0.0	3.0	0.0	0.0	3.0
Researcher Development Course	PHD-QMA-107	Becoming Writers - A six-step guide to more effective writing	Researcher Development	07-Oct-2020 10:00	07-Oct-2020 13:00	0.0	0.0	1.0	0.0	2.0	3.0

Not Applicable Activities

Nothing found to display