Stage 0 Report

Automatic Music Transcription with Deep Neural Networks

Lele Liu

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

Independent Assessor:
Dr. Marcus Pearce

Centre for Digital Music

Contents

1 Introduction

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 [50]. 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 [43], 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 [12, 77, 17, 47]. Other deep learning methods such as multi-task learning (MTL) [36, 44] and using sequence-to-sequence models [66] 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.

In our proposed project, we aim to explore deep learning methods for *complete transcription*. We will develop systems that can extract information from music audio and generate human-readable music sheets. More specifically, we are going to explore possible methods for generating music notation and deep learning methods that can be applied to AMT, such as sequence-to-sequence models and attention mechanisms.

The structure of the rest of this report is as follows. In Section 2, we review recent research in AMT and possible methods that can be applied to AMT, focusing on deep learning methods. We also point out some limitations in current AMT systems. After the literature review, we describe our research question and aims in Section 3, and our proposed work in detail in Section 4. We include an explanation of our preliminary work in Section 5,

followed by our long-term and short-term research plan in Section 6.

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 Section 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 [55].

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 [94].

In most AMT systems, music signals are first converted into some time-frequency representations through algorithms such as Short-Time Fourier Transform (STFT) [69, 70] or Constant-Q Transform (CQT) [77, 45]. The system output may be in various format 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, 77]. 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 [**?**, 60] to deep learning methods [14]. Besides, music language models are found useful in improving note tracking results [94]. 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 [51]. There are some other tasks in the wider field of MIR that are related to multi-pitch streaming, including music source separation [81] and instrument recognition [41].

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 [63]. 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 [17, 66]. Regardless of the choice of approaches, we refer to the process of converting music audio into music staff notation as *complete music transcription*.

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

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 [29], MIDI Aligned Piano Sounds (MAPS) dataset [27], Bach10 [26], MedleyDB [9], and MusicNet [80]. Two recently proposed datasets are MAESTRO [37] and Slakh [56].

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 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 [78], 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, 13, 45, 36]. Much less work has been done on stream and notation level transcriptions [57, 58, 59]. 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:

$$precision = \frac{TP}{TP + FP} \tag{1}$$

$$recall = \frac{TP}{TP + FN} \tag{2}$$

$$F\text{-}measure \ = \frac{2 \times precision \times recall}{precision + recall} \tag{3}$$

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 [68], 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 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. [59]).

There are less publications on *multi-pitch streaming*. The evaluation for *multi-pitch streaming* uses similar metric like precision and recall. Gomez

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

and Bonada [28] proposed a simple method of calculating accuracy and false rate to evaluate voice streaming applied to A Capella transcription. In 2014, Duan et al [25] used a similar evaluation method to calculate a more general multi-pitch streaming accuracy. The accuracy is defined as:

$$accuracy = \frac{TP}{TP + FP + FN} \tag{4}$$

Another work by Molina et al [59] 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 [21, 58, 57]. A recent approach for evaluating score transcriptions is proposed by Mcleod and Yoshii [57], which is based on a previous approach [58] 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 [23] 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 1.

Table 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 [77]	MAPS	74.45	67.05	-	-	CNN acoustic model + language model
Kelz et al. 2016 [45]	MAPS	79.33	-	-	-	Improved based on [77]
Bittner et al. 2017 [11]	Bach10, Su, MedleyDB	59% accuracy on Bach10	-	-	-	Proposed HCQT and 'deep salience' representation
Ycart et al. 2019 [94]	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 [49]	MAESTRO	91.4	95.6	-	-	Improved 'Onset and Frame' by adversarial learning
Kelz et al. 2019 [44]	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 [17]	music generated by MIDI synthesizers	-	-	-	P(sub) = 0.6% P(ins) = 0.1% P(del) = 0	End-to-end complete transcription for monophonic music
Romn et al. 2019 [70]	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 [64], 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 [12], 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 [12]: (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 [77]. 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 [77] 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 [77], Kelz et al [45] showed the potential of simple frame-based approaches for polyphonic piano transcription using an architecture similar to [77], 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 [80] 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 [46].

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 [14]. 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 [77], 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 [91]. More recently, Wang et al [87] 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 [94] 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* [71], 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 [49], 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 [44], 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 [44] 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 [17, 66] 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 [17]. 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 [66] 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 Romn et al [70], 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 [68, 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 [23], 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 [17, 69] and special cases for polyphonic music [70], 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 [53] 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 More on Deep Learning in Sequential Problems

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. [31] and later improved in [30] as a sequence "Transducer", which uses connectionist temporal classification (CTC) to label non-aligned sequence. A null symbol in 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 [79]. 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 [79, 1, 54], automatic speech recognition [2], syntactic constituency parsing [85], image captioning [96] and automatic music transcription [82, 66]. 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 model improve model performance [18, 83] (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 [95]), it is proved to be helpful and commonly used in today's sequential systems including neural machine translation [3], automatic speech recognition [19, 20], sentiment analysis [76], and automatic music transcription [66].

A early use of attention mechanism in automatic speech recognition is in [19], 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 [20], 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. [18] 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, out-

put 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 [83] 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 [76, 73, 90, 86].

3 Research Question & Aims

Our main aim is 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. We are going to mainly focus on polyphonic piano music, and further extend our methods to more instruments.

4 Proposed Work

In this section, we describe four studies we propose during the PhD and the resources and tools we are going to use in our project. We propose to first focus on building an end-to-end system for score-level transcription, and then improve our model by further research on sequential models, evaluation metrics, system generalization and including more musical information.

4.1 Study 1: Design of an end-to-end system for score-level polyphonic piano music transcription

We propose to design a score-level data representation for polyphonic piano transcription. We will build a model that jointly predicts a score-level music data representation and a piano-roll representation, as well as create a new synthesized dataset with matched music audio and scores for training complete music transcription networks.

We aim to finish this work as part of stage 1 milestone, and submit a paper about this work to ICASSP 2021 (deadline around October this year).

4.1.1 Data representation

Compared to the large amount of work on AMT in frame and note level, there is less work focusing on score-level music transcription. As we discussed above in Section 2.3.4, although there are research on complete music transcription using end-to-end systems, they are whether working on monophonic transcription [69, 66], or a constrained condition of polyphonic music [70, 17]. One of the factors that prevent systems from predicting more common polyphonic music scores is the limitation in score-level music data formats. Although existing representations such as LilyPond and musicXML can be simply modified to fit into neural network systems for monophonic music, they become complex for polyphonic music and are not suited for direct use in neural networks.

We propose our first step in the PhD as designing a score-level data representation for polyphonic piano music fit neural networks. The designed data representation should fit for a complete music transcription system, and can be adaptable for other instruments and music styles. The designed data representation will be used as an output of a deep sequential model for AMT, and can be further decoded into a machine-readable music format.

Current literature covered different designs of symbolic music representations. Apart from the direct use of LilyPond format in [17] for monophonic transcription and two-melody polyphonic music, there are some other designs for more complex polyphonic music. For example, Román et al. used a **kern format-based symbolic representation in Bach chorales and four part strings transcription tasks. In some other music generation works, people have used text-based symbols with key, time, and beat information [35] or serialized midi pitch numbers [39, 40] to describe Bach chorales, and MIDI-like event-based representation [67, 40] to describe polyphonic piano music with expressive timing and dynamics in performance.

We will learn from existing symbolic music data representations and draw a new design of representation for symbolic polyphonic piano music. The designed data representation will not be very complete to include all the details in a music sheet, but cover the most essential music information to create readable music notations. More specifically, we plan to cover the following information in our designed data representation:

- note information (pitch, onset and offset) - pitch will be described in MIDI pitch numbers, re-scaled into 88 values for piano music, and onset and offset times will be described in beats.

- rest information (times and duration) - also described in beats. Probably, we will include only the rest duration in the representation, since rest start time can be inferred from note offsets. It can be difficult to predict rest when there are multiple voice parts in the music, but currently, we do not plan to cover voice separation in our work.

- beat and barlines information - they are related to time signatures, and may be hard to predict when there are changes in tempo and time signatures. Basically, we plan to include symbols for barlines in the representation, and beat will be described together with the onset/offset/rest duration times.

On the contrary, we are not going to include more detailed music information, listed below:

- clefs - it can be simply inferred from scores. Besides, using a different clef usually does not result in a wrong music sheet.

- time signatures - it can be inferred from the predicted metre and barlines information

- key signatures - it can be hard to predict a correct key signature when the music piece is not long enough, or when the key signature changes. Also, predicting key signature from music scores can be much easier compared to from music audio. Thus, we leave the prediction of key signatures as a further step in converting the data representation into a full music score, where we can make use of simpler methods (e.g. using chroma histograms).

- voice/hand-part information - voice separation can sometimes be very subjective. Since in this pilot study, we only want to provide a baseline to predict the most necessary information in a music score, we leave the prediction of voice as a future work.

- fermatas and arpeggios - they are of less importance in music scores, and some music sheets doesn't provide these information.

- speed/tempo information - there are different ways in describing speed and tempo information in a music sheet, such as providing bps values and and giving text descriptions. In this study, we omit tempo information, since it can be easier inferred from beat information.
- dynamics - they are precisely defined in, e.g. MIDI formats, but usually not very clearly defined in music scores. Symbols such as f, p, crescendo and decrescendo are subjective evaluations. Thus, we do not take dynamics into account here.
- other performance related information (e.g. slurs, articulation marks, grace notes, ornaments, emotions)

One challenge in mapping music audios with music scores is dealing with the use of piano pedals in music performance. Actual note duration in scores can be hard to infer from audios when the piano pedal is down. Unlike in piano-roll representation, where extended note offsets can be useful for reconstructing music audio, music scores usually do not have extended note offsets. Thus, in our designed data representation, we plan to keep the original note durations and not to extend them according to piano pedalling events. To make it easier, we can also use some pieces that do not contain many piano pedal events (e.g. Bach and Mozart) in our dataset.

4.1.2 Data collection

Despite the variable choices of AMT datasets, including but not limited to the MAPS dataset [27], Bach10 dataset [26], and large-scale MAESTRO dataset [37], most dataset for AMT tasks provide only audio and MIDI (or similar) format, and cannot be used for complete music transcription. The lack of music scores, especially matched music audio and machine-readable music scores, is a big limitation in developing deep learning-based end-to-end score-level AMT systems.

In order to access score information for music pieces, we make use of two small collections of matched scores in musicXML format - 10 pieces from the Bach10 dataset and 30 pieces from the MAPS dataset (or extract note, rhythm and key annotations from the A-MAPS dataset [92]).

Apart from making use of existing labelled data, we plan to collect a larger dataset with music scores and synthesized music audios. The dataset should contain more than 200 music pieces to enable training of deep neural

networks. We plan to collect music scores from MusicScore[3] (with pdf, MIDI, musicXML, and mp3 formats) and the Mutopia Project[4] (with pdf, MIDI, and LilyPond format). We are going to use Kontakt 6 player and sampled piano library to synthesize music audios from the collected music scores. We can also try using different synthesizers with room conditions.

4.1.3 Model architecture

We will then use our designed music data representation in a deep end-to-end model for automatic music transcription. The system will take an input music audio spectrogram x (possibly using CQT, Mel spectrogram, HCQT or VQT, depending on performance) and give two outputs y_1 and y_2, where y_1 will be a probability distribution for our designed symbolic music data representation, and y_2 will be a probability distribution for a framewise piano-roll representation. By designing the model in a multitask manner, the system can both give a prediction of music notation for the input audio, and provide some interpretation on the temporal variations in music performance. Ideally, by including both outputs in the model, they can help each other to improve the prediction accuracy.

We plan to build the model that uses a shared feature extraction network, which is followed by two separated branches for the two tasks. More specifically, we plan to use a shared CNN network to extract a feature space from input music audio spectrogram, and then add two sequential models to predict the two outputs individually. Possible choices of network structure for the sequential models include GRUs, LSTMs and TCNs. We leave the choice of network structure to be an experiment in our study. We will first train separate models for single tasks, and then train a multitask model that combines the two tasks. So that we can see if there is any improvement by combining the two tasks in one system. A further work on the model architecture can be trying to feed the piano-roll output back to the separate branch to predict music scores. The diagram for our model structure is shown in Figure 1.

Although piano-roll representation can be simply aligned with audio spectrogram in the time direction, our designed data representation will cover additional music information (e.g. barlines) that results in variable lengths in the system output. We can either use sequence-to-sequence models or connectionist temporal classification (CTC) to deal with it.

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

[4]https://www.mutopiaproject.org/index.html

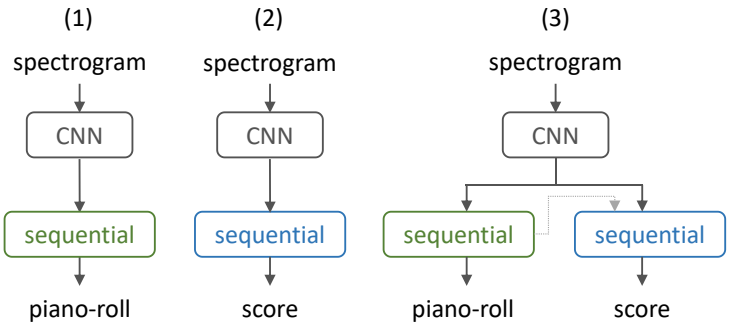


Figure 1: Model structure. The CNNs have similar structure, and the green and blue sequential networks are for prediction of music score and piano-roll respectively.

Another problem in sequence processing in our task is both sequence-to-sequence models and CTC decoding are limited to dealing with short sequence segments. So, we need to do some work to divide long music pieces into small processible segments. Simply cutting a music piece into a specific time length can result in separated notes/bars. One choice is to cut music into bars, which is also used in [17] and [69]. In this way, the system only need to process one bar at a time. We can use beat tracking methods for pre-processing to get beat and bar times. Due to the variable lengths in the cut music segments, we may need to do some zero padding in the model input. We consider this choice as a solution to an offline AMT system. For further work, we can try using the online CTC proposed in [42] that can be used for long and variable sequence length.

4.1.4 Result evaluation

There is not yet a standard metric for score-level music transcription. Some evaluation methods in the literature are:

- jointly considering pitch error rate, missing note rate, extra note rate, onset and offset time error rate, by Nakamura et al. in 2018 [60].
- MV2H evaluation metric proposed in 2018 [58] and adapted for non-aligned transcription in 2019 [57], which jointly considers multi-pitch detection, voice separation, metrical alignment, note value detection,

and harmonic analysis, and uses a DTW-based alignment algorithm to allow non-aligned transcriptions.

- using character/word error rate (CER or WER) commonly used in natural language processing tasks, by Román [69, 70], for a score representation based on text.

Here, we propose to use the first evaluation method in the list. Since it is more related to music concept compared to the third one, and simpler than the second one (especially when we are not including voice separation in our task). We can directly use the evaluation tool for analysing musicXML files by Nakamura.

Another choice, not yet in literature,is we could use the evaluation metric currently in development with Adrien Ycart, which takes perceptional evaluation into account.

Some systems working in polyphonic music transcription can be used for a performance comparison with our system, such as [17] using end-to-end method with sequence-to-sequence modesls, [60] combining multi-pitch detection, note tracking, rhythm quantization and score typesetting, and ScoreCloud[5], a software for music transcription. To be more specific on polyphonic piano transcription, we plan to compare our system with [60] proposed by Nakamura et al..

4.2 Study 2: Deep sequential models for complete music transcription

We suppose this will be the main part of the PhD research. 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 in different ways and including more music information in our transcription results.

More detailed experiment proposals are described in the following.

[5] https://scorecloud.com

4.2.1 Optimizing input/output data representation

A first choice for better transcription is to optimize the preliminary system by experimenting with more suitable input/output data representations and improve the model structure.

Input data representation We prefer a time-frequency input data representation rather than directly use a signal input. Two of the main reasons are 1) we will need a complex model structure to do feature extraction when dealing with audio input, and 2) transforming audio signal into a time-frequency representation is similar to what cochlea do during hearing, when it separates the frequencies of incoming wave.

There are various types of time-frequency based data representations that have been used for audio signal processing, such as Mel spectrogram, Constant-Q Transform (CQT), Variable-Q Transform (VQT), Harmonic Constant-Q Transform (HCQT). Which representation is better for AMT is hard to define. Mel spectrogram uses a Mel-frequency scaling, which is designed to simulate the way human ear works. It is commonly used in audio signal processing tasks [24, 38, 16, 66]. On the other hand, CQT [15, 84] uses logarithmic-scaled filters spaced in frequency, which is more close to how pitch frequencies are defined. Works using CQT as a input representation for AMT including [7, 82]. A general problem in using CQT is that in low frequencies, the time window will become very long, which causes problem in predicting low pitches in music. In order to solve the problem, Schörkhuber et al. proposed the use of VQT [74], where the Q-factor is modified to be smoothly changed to preserve the time resolution in low frequencies. More recently, HCQT is used in the "deep salience" representation [11] and later used in other AMT systems [10, 89, 22]. HCQT is a 3-dimensional representation indexed by harmonic, frequency and time, and is calculated based on CQT where for each harmony h, a time-frequency representation is calculated by re-scale the CQT's frequency index by harmonic values (a CQT is kept the same for the first harmonic). There are also other time-frequency based transformations worth trying (e.g. ERBlet transform [65]).

Output data representation Another aspect could be changes on the designed output representation. In study 1, we will mainly focus on getting some basic results on end-to-end polyphonic transcription, and only plans to include simple note, rest, beat and barlines information in our representation. As a starting up step, we may limit the work to a constrained condition.

For example, for music pieces without very complex rhythm patterns (e.g. triplets, change of time signatures).

To work towards more general polyphonic music, we want the music data representation to be able to deal with more complex music pieces, and extend it to cover various note lengths and rhythm patterns.

4.2.2 Experiment with more model architectures

In study 1, we have planned to try different sequence models combined with a multitask model. Here, we could also try different model architectures, such as including different auxiliary tasks (e.g. pitch salience, onsets?), adding a feedback input from the auxiliary task to the music score branch, or use soft parameter sharing rather than a hard parameter sharing in the CNN network. We can also do further parameter tuning to see if we can achieve better results.

Apart from these above, we are also planning to try more on model structures. Listed below:

Modelling long-term dependencies We first propose to use attention mechanism in our AMT system. Attention mechanism have been proved helpful in many sequential problems (previously discussed in Section 2.5.2). In our system designed in study 1, we are planning to use whether sequence-to-sequence models or TCNs. We can try both basic attention and (multi-head) self-attention, depending on our final model architecture choice, and see if we can get better results.

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 [93]. To take this musical feature into account, we can use a relation-aware self-attention [75], which is also used in [40] 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.

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 content. Mu-

sic pitch 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 AMT system. A simple solution can be design a new musically meaningful loss function that give 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 [32], similar application in automatic speech recognition can be found in [52] and [72]. Also, CNN is another straight forward network structure for modelling 2D data. In [49], 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 [68, 23, 60] (and one we are currently working on), but they are whether working on frame-level or note-level evaluations. Current evaluations metrics for complete music transcription (see [21, 58, 57]) 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, and try to design a new evaluation metric for complete transcription that is more musically reasonable.

4.4 Study 4: More exploration

We propose a list of more explorations that we are considering to include throughout the PhD project. There may not be enough time to cover all of them, but we put our general ideas here and make it a reference for our long-term choices.

4.4.1 Dealing with 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 use 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 [88] a similar approach). We could also use an adversarial network to force the discriminators to learn a shared feature space.

4.4.2 Dealing with more music information

We could also try to cover more music information in our AMT system. We won't have time to be very complete and cover very detailed performance annotations, but towards the end of this PhD, we want to cover more music information in our AMT system, including instruments, clefs, time signatures, keys, voice information.

We can choose from some of the ideas below:

More instruments Some works already covered multi-instruments transcription [25, 10, 55], 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.

Hand separation/Voice separation Voice separation is a subtask that is commonly ignored in most current AMT systems. Early research used stochastic local search [48] and Kalman filtering [33] to track hand information in MIDI data. The methods are later replaced by HMMs and RNNs [61, 62, 34] for higher accuracy. Hand separation in piano music can be considered whether as a problem of symbolic music, or it can be combined into the overall process of AMT in an end-to-end method. We have no idea which method will be better, but it may be worth a try. Also, current research showed a preference on using RNN networks for hand separation only [34], and a preference on using HMMs for hand separation combined with piano fingering [62]. We can also explore methods to improve current prediction accuracy, especially for real-time systems.

4.5 Resources and Tools

We are going to use the following resources and tools:

- Collecting music pieces from MuseScore sheet music[6]
- Synthesize music audio using Native Instrument Contakt Player piano sampler[7]
- Reaper[8] as digital audio workstation
- Pytorch[9] framework for deep learning

5 Preliminary Work

For now, we have worked on a literature review on neural network methods for AMT (book chapter submitted), presented a DMRN poster, joined

[6] https://musescore.com/
[7] https://www.native-instruments.com/en/
[8] https://www.reaper.fm/
[9] https://pytorch.org/

Adrien Ycart with a AMT evaluation project (paper submitted to TISMIR) and worked a bit on my proposed Study 1. Related outputs are added to the Appendix. In the following, we'll describe the work we have done so far for Study 1.

5.1 Dataset generation

Due to the lack of scores in commonly used AMT datasets, we decided to collect a new set of music scores and generate a synthesized dataset. We have currently collected 472 music scores (in musicXML format) from MuseScore[6]. Statistics on the dataset are in Table 2 and Figure 2.

Table 2: Dataset Statistics

Total hours	26.52 * 4 piano models
Total notes	679,228
Maximum polyphony level (with piano pedal)	52
Maximum polyphony level (without piano pedal)	13
Average polyphony level (with piano pedal)	3.59
Average polyphony level (without piano pedal)	3.02

We used the Batch Convert plugin[10] in MuseScroe to convert the collected scores to MIDI files, and synthesized music audio using Kontakt Player[7] in Reaper[8]. We synthesized each music piece using four piano fonts to get better generalization ability, and added some reverb effects to make the audio recordings more 'real'. In order to allow batch generation, we used ReaScript[11] and reapy[12] to allow batch rendering with python. Related software settings description and scripts are included in supporting materials.

5.2 Data representation for scores

As a starting point, we consider a simplified condition for polyphonic piano music. We separate music pieces into bars, and assume a 4/4 time signature. For now, we do not take into account complex rhythm structures such as triplets and arpeggios. Among all the different symbolic music formats, we choose to use a modified version of LilyPond format for its simplicity and

[10]https://musescore.org/en/project/batch-convert
[11]https://www.reaper.fm/sdk/reascript/reascript.php
[12]https://python-reapy.readthedocs.io/en/latest/

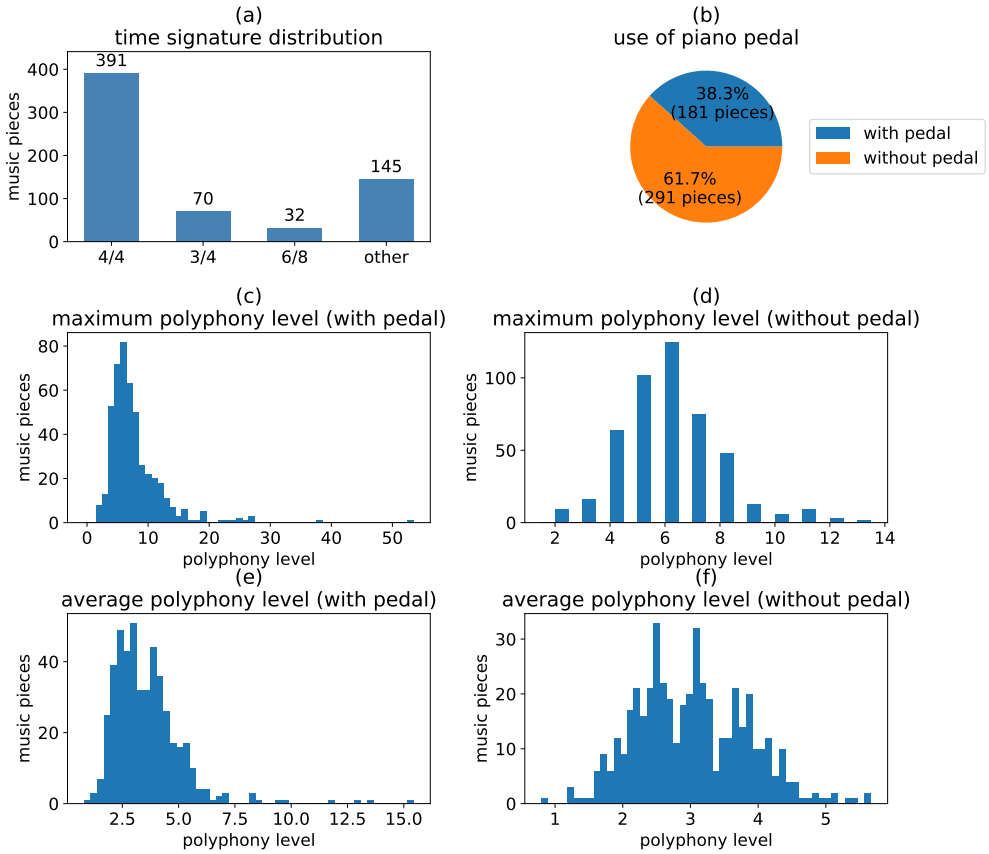


Figure 2: Statistics of the dataset. (a) Distribution of time signatures. There may be more than one time signature used in one music piece. (b) Distribution of the use of piano pedals in the music pieces. (c) Distribution of maximum polyphony level per piece (taking into account the use of piano pedals). (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 whole dataset, not subsets from (b).

better fit for neural networks. We represent polyphonic music as a single string, which will be one-hot encoded to fit for the model's output layer. Below are the symbols we use for different music components:

- pitch - combined with pitch chroma (e.g. 'c' for C, 'is' for ♯ and 'es' for ♭) and pitch height (e.g.' $'$ ' for higher octave and ' , ' for lower octave, duplicate e.g. ' $''$ ' for double octaves). The encoding is the same as in LilyPond.

  When using the one-hot encoding, pitch chromas are combined into 12 values (e.g. 'cis' is equal to 'des' and they are considered as a whole rather than separated as 'c' and 'is'. Similarly, duplicate octaves (e.g. ' $'''$ ') are considered as a whole.

- rest - rest is represented as 'r'.

- note duration - we use numbers to represent note durations, e.g. '4' for a 4th note. The same duration representation is used for rests. ' . ' is added for dotted notes - resulting in e.g. '4.'.

- concurrent notes (polyphony) - concurrent notes are grouped by brackets (e.g. pitch for a C chord starting with a middle C is '$\langle\ c'\ e'\ g'\ \rangle$').

  We are not currently separating music into different hands and voices.

- continuing note (tie) - ties are represented using '$\sim$'. Unlike in LilyPond where ties are added to the starting note, here, we add it to the continuing note. e.g. 'c4.' is equal to '$c4\ \sim c8$'

Based on the above design, an example score data representation is shown in Figure 3.

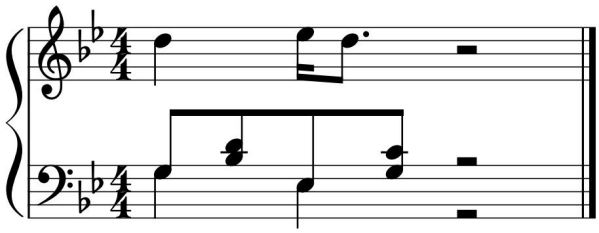
6 Research Plan

6.1 Time plan

Grantt charts for my short and long-term plan are in Figure 4.

Short-term plan Before the stage 1 milestone in this December, I will spend most of the time for Study 1 described in Section 4.1 (see Gantt chart in Figure 4). Below are the tasks I plan to do:

- **Collecting/creating new datasets** for score-level AMT.



<g d''>8 <~g bes d' ~d''>8 <ees ees''>16 <~ees d'>16 <~ees g c ~d'>8 r2

Figure 3: Example score data representation. Delimiters ‘ < ’ and ‘ > ’ enclose concurrent notes.

- **Designing score-level data representation** for deep sequential models in complete music transcription.
- **Implementing models** for data representation experiments.
- **Reproducing some baseline AMT systems** for result comparison in experiments.
- **Running experiments** to evaluate the suitability of designed symbolic data representation.
- **Writing paper** for ICASSP 2021
- **Staring experiment** with study 2

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

6.2 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

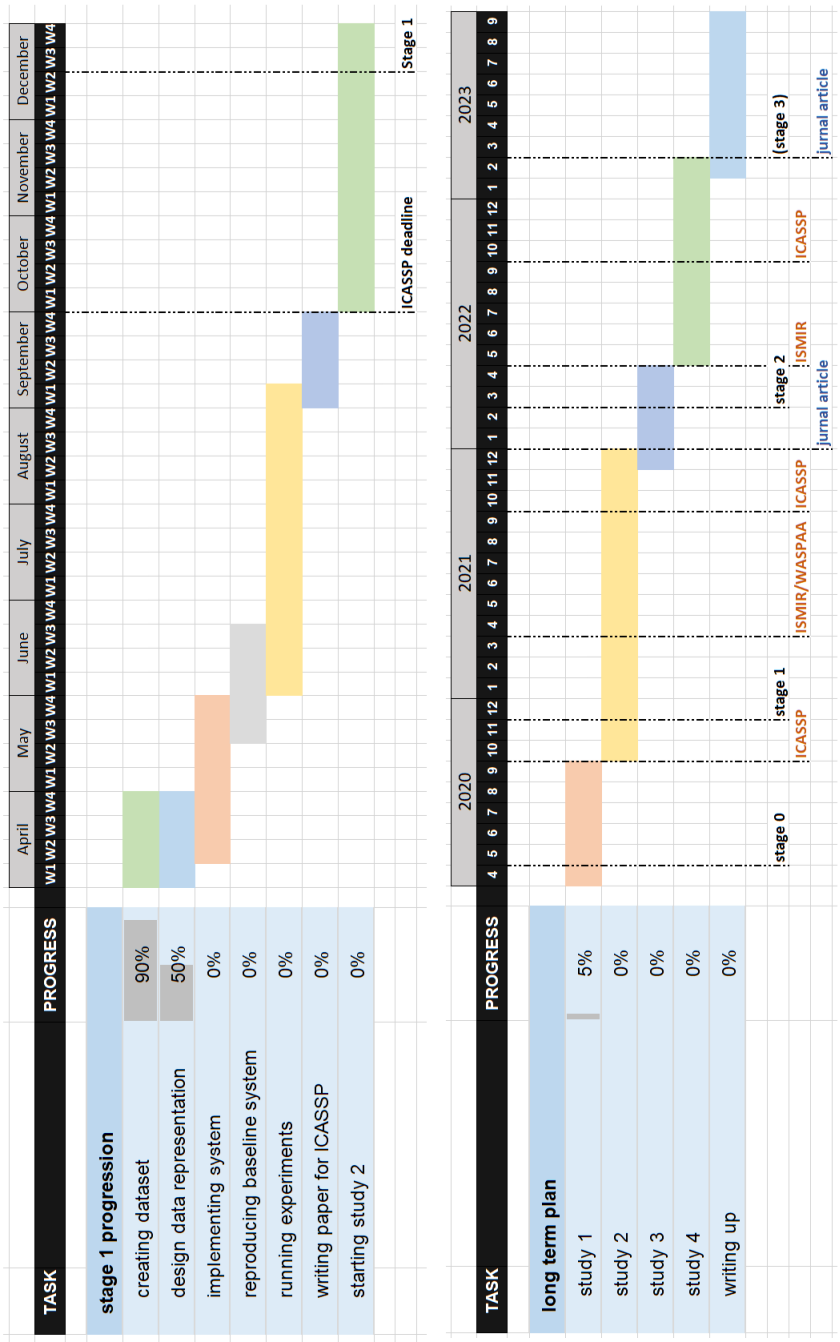


Figure 4: Gantt chart for research plan.

for Musicology (DLfM) and machine learning conferences such as NeurIPS, ICLR and ICML. In the short term, we aim to submit a paper to ICASSP, deadline in October.

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. *3rd 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. *ISMIR*, (Ismir):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. *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017*, pages 63–70, 2017.

[12] 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.

[13] 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.

[14] 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.

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

[16] Emre Cakir, Giambattista Parascandolo, Toni Heittola, Heikki Huttunen, and Tuomas Virtanen. Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 25(6):1291–1303, 2017.

[17] 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.

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

[19] 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.

[20] 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.

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

[22] Helena Cuesta, Emilia Gómez, and Pritish Chandna. A Framework for Multi-f0 Modeling in SATB Choir Recordings. 2019.

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

[24] Sander Dieleman and Benjamin Schrauwen. End-to-end learning for music audio. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 6964–6968, 2014.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

[30] Alex Graves. Sequence Transduction with Recurrent Neural Networks. 2012.

[31] 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.

[32] 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.

[33] Aristotelis Hadjakos and Fraois Lefebvre-Albaret. Three methods for pianist hand assignment. *Proceedings of the 6th Sound and Music Computing Conference, SMC 2009*, (July):321–326, 2009.

[34] Aristotelis Hadjakos, Simon Waloschek, and Alexander Leemhuis. Detecting Hands in Piano MIDI Data. *Mensch und Computer 2019-Workshopband*, pages 543–546, 2019.

[35] Gaëtan Hadjeres, François Pachet, and Frank Nielsen. DeepBach: A steerable model for bach chorales generation. *34th International Conference on Machine Learning, ICML 2017*, 3:2187–2196, 2017.

[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. *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] Shawn Hershey, Sourish Chaudhuri, Daniel P.W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson. CNN architectures for large-scale audio classification. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 131–135, 2017.

[39] Cheng Zhi Anna Huang, Tim Cooijmans, Adam Roberts, Aaron Courville, and Douglas Eck. Counterpoint by convolution. *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017*, pages 211–218, 2017.

[40] 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.

[41] 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.

[42] Kyuyeon Hwang and Wonyong Sung. Online Sequence Training of Recurrent Neural Networks with Connectionist Temporal Classification. pages 1–16, 2015.

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

[44] 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.

[45] 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.

[46] 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.

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

[48] J Kilian and HH Hoos. Voice separationa local optimisation approach. *Proceedings of the Third Annual International Symposium on Music Information Retrieval*, pages 39–46, 2002.

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

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

[51] 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.

[52] 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.

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

[54] 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.

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

[56] 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.

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

[58] Andrew Mcleod and Mark Steedman. Evaluating Automatic Polyphonic Music Transcription. *ISMIR*, pages 42–49, 2018.

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

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

[61] Eita Nakamura, Nobutaka Ono, and Shigeki Sagayama. Merged-output HMM for piano fingering of both hands. *Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR 2014*, (Ismir):531–536, 2014.

[62] Eita Nakamura, Yasuyuki Saito, and Kazuyoshi Yoshii. Statistical learning and estimation of piano fingering. *Information Sciences*, 517:68–85, 2020.

[63] 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.

[64] 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.

[65] T. Necciari, P. Balazs, N. Holighaus, and P. L. Sondergaard. The ERBlet transform: An auditory-based time-frequency representation with perfect reconstruction. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 498–502, 2013.

[66] 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.

[67] Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, and Karen Simonyan. This time with feeling: learning expressive musical performance. *Neural Computing and Applications*, pages 1–24, 2018.

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

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

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

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

[72] 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.

[73] 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.

[74] Christian Schörkhuber, Anssi Klapuri, Nicki Holighaus, and Monika Dörfler. A Matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution. *Proceedings of the AES International Conference*, (March 2015):232–239, 2014.

[75] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-Attention with Relative Position Representations. pages 464–468, 2018.

[76] 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.

[77] 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.

[78] 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.

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

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

[81] 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.

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

[83] 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.

[84] Gino Angelo Velasco, Nicki Holighaus, Monika Dörfler, and Thomas Grill. Constructing an Invertible Constant-Q Transform with Nonstationary Gabor Frames. *Porc. of the 14th Int. Conference on Digital Audio Effects (DAFx-11), Paris, France, September*, pages 93–99, 2011.

[85] 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.

[86] 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. pages 5797–5808, 2019.

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

[88] 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.

[89] Yu Te Wu, Berlin Chen, and Li Su. Polyphonic Music Transcription with Semantic Segmentation. *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2019-May:166–170, 2019.

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

[91] 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.

[92] Adrien Ycart and Emmanouil Benetos. A-MAPS : AUGMENTED MAPS DATASET WITH RHYTHM AND KEY ANNOTATIONS. *19th Interna- tional Society for Music Information Retrieval Conference, Late Breaking Demo*, pages 2–3, 2018.

[93] 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.

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

[95] 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.

[96] 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.

7 Appendix

7.1 Training courses

Courses I currently have taken or am now taking are listed in Table 3.

Table 3: 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	-
ECS7001	Neural Networks and NLP	-
Code	**CPD Training Courses**	**Status**
RD101	Getting Started with Your PhD	Finished
RD100	Working with Your Supervisor	Finished
RD208	Making a Poster Presentation	Finished
RD105	Making the Most of Your First Academic Conference	Finished

7.2 Current outputs

Below are the 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, submitted.
- 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), under review.

7.3 Attached Documents

In the remaining, I attach

- my AIM personal development plan,

- the DMRN+14 poster abstract,
- the submitted AI+music handbook chapter.