Title of your thesis

Your Name

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Your Department

Your Advisor, Chair
First Committee
Second Committee
Third Committee
Last Committee

December 4, 2020
Blacksburg, Virginia

Keywords: Some Keywords, Subject matter, etc.

Title of your thesis

Your Name

ABSTRACT

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Dedicated to Virginia Tech.

Acknowledgments

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

- Introduce the idea of musical research with computers. Talk about the illiac suite [23] and Music Information Retrieval.
- Significance of machine learning on the field
- Introduce idea of expressive musical performance. Brief conversation about the different performance components (articulation, dynamics, timing).
- Using Transformer architecture which hasn't been done in the field.
-

  Richard: report results

Chapter 2

Background

Richard: Consider creating two backgrounds sections that are split into musical information and modeling information. The musical information can contain related work as well as background and context. The same can go for modeling

.

There are two major research components that this project is based on. The first is the problem domain of expressive musical performance (EMP), and the second is the ML modeling domain of Transformers. We will introduce both of these components and provide context for what makes them interesting as a research project and why they are worth exploring together. We start first with an overview and definition of EMP, and then a summary of the Transformer.

2.1 Expressive Musical Performance

EMP is a subset of the research field of Music Information Retrieval (MIR) [1] whose purpose is to use computational information to study, interpret, and gain a better understanding of the *essence* of music itself [26]. Perhaps the most well known MIR application is that of a

[1] Widmer [26] points out that MIR itself does not encompass the entire scope of computer music research, but that it is a good proxy to use when referring the field as a whole. We will operate under the same assumption

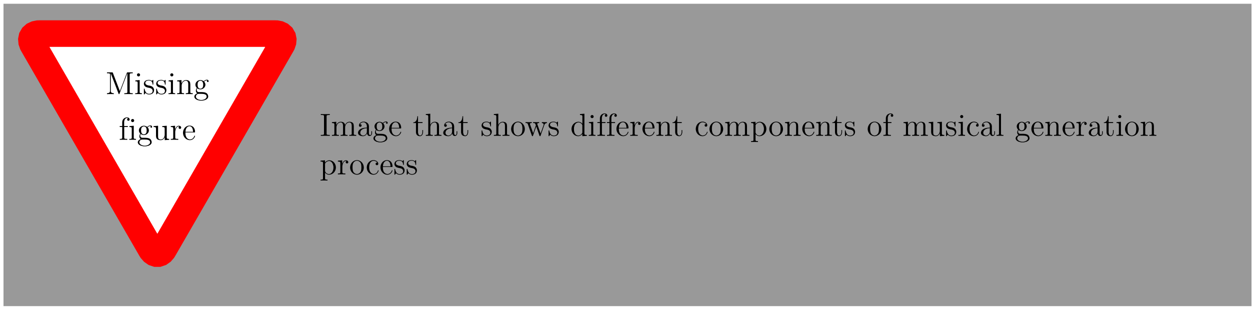

Figure 2.1: The first step of musical generation is composition, shown as a score in the figure. The second is performance, which is our area of interest. The third is the production of sound. Each different agent: composer, performer, instrument, and listener, can be thought of as a separate computational model in the generation process

musical recommendation system used by streaming services such as Spotify [2] to provide a personalized and unique experience for each user. However, as Widmer [26] suggests, there are a number of other non-trivial problems that face the field and will require significant effort from the research community to properly understand. A proper understanding of musical performance is one of them.

MIR tasks can be broadly categorized in two ways - the first as computational methods for music analysis, and the second as computational methods for music generation. We are interested in the latter and its application in musical performance. In order to study how musical performance generation (and more particularly *expressive* musical performance generation) models work, it is necessary to gain a proper understanding of the entire computational musical generation process as a whole. Ji et al. [17] break the process down into 3 different components, with 4 different roles or agents that interact with that process. Figure 2.1 shows each step in the process as well as the agents that participate

Richard: Try to get permission to reproduce the image in the paper

.

[2] spotify.com

An EMP model is analogous to the performer as show in 2.1, who takes as input a musical composition and produces as output a performance. It is the phenomena of musical expression that makes the performance generation process interesting. Musical expression can be thought of as the performers' interpretation of a composition codified into different performance parameters that are intended to increase the quality of the musical experience by the final listener. Because the quality of a musical experience is highly subjective, there is no definition of what makes for a "correct" interpretation of a given composition [3]. The subjective nature of EMP generation makes it a difficult problem to understand from a computational perspective. However, it also makes it a highly intriguing research topic given that a clear understanding of the problem from a computational perspective will no doubt further our understanding of what exactly it is that makes music so subjective in the first place, and bring us one step closer to understanding music itself.

To properly understand exactly what it is that constitutes expression in musical performance, it is necessary to provide a detailed description of the first two components of the generation process - namely, scores and performances. We refer the reader to appendix A.1 which provides some basic terminology and concepts that will be useful for grasping the following section [3]. Due to the constraint of our data we focus only on western classical piano music.

Richard add reference

2.1.1 Scores

A musical score is a symbolic representation of a musical composition. The symbolic notation used to create musical scores can be thought of as a language used to express musical ideas and information. It presents this information in a hierarchical structure with different levels of musical detail at each level. The lowest level contains information about the pitch and

[3] Most of the appendix material may seem elementary to those who already have a background in music or musical notation. However, we feel that is necessary to include if for no other reason than to provide a clear definition for our descriptions both in general and at detailed mathematical level

timing of every single note, as well as optional information about how the note should be played. This can include information specific to instruments such as the bow direction of a violin, but for our purposes (dealing only with piano) we will consider this to be the articulation of each note, usually indicated by legato or staccato

Richard: Make sure to have some background information on articulation in the appendix

The middle level contains information related to certain substructures within the musical composition, which are usually expressed within a grouping of notes or measures. The most common score annotations at this level are dynamic markings which indicate whether to play a grouping of notes loud (Forte), soft (Piano), or to gradually increase or decrease the volume (crescendo or decrescendo). Although dynamic markings are the most common at this level, it is also possible to see score markings for all other musical features, such as local tempo or articulation of a certain substructure. Perhaps the most important score marking at this level is that of a phrase, which is a marking that indicates that a group of notes should be interpreted as belonging to a singular musical idea and that each note should fit within the context of the phrase as a whole. A phrase can be expressed through all of the different aforementioned musical features, including the tempo, timing, dynamics, and articulation of the notes.

The highest level contains meta-information that relates to the entire composition as a whole. This information typically includes the key signature and time signature, as well as the global tempo for the entire piece, most commonly represented as BPM.

2.1.2 Peformance

An expressive musical performance contains most of the same musical information as does a score, but with one key difference; that is, that an expressive performance will deviate (or interpret) from the exact information that is presented in the score. For example, although a score may indicate a tempo of 120 BPM, it is highly unlikely that a given performer will perfectly adhere to this tempo throughout the entirety of the piece. This is even more apparent if the score indicates a change in tempo somewhere in the composition. If a score indicates that the performance should speed up over a series of notes, there is no telling at what rate the tempo should increase. Some performers may choose to speed up at a fast rate and over a short period of time. Others may choose to increase the tempo at a slow rate and over a longer period of time. A single accelerando (a score indication to pick up the tempo) can result in either of these outcomes.

With that being said, a performance contains most of the same features related to a score, which include pitch, tempo, timing and articulation. Each of these expressive features will be measurable and absolute, whereas the score markings of these features can be viewed more as a suggestion than a rule. There a few additional features that are present in performances which are not in scores. The first we will refer to as deviation which is heavily related to timing. It is typically represented as a numerical number which represents how far off the timing of a particular note deviates from it's "correct" position in the score. These micro-timing deviations present in musical performances are an essential part of expression. Without them, indicating that each note onset and offset is exactly in line with its marking in the score, performances sound robotic and mundane .

Richard

add a reference, graphic, and sam-

The other important feature of performance that is not always present in a score applies specifically to the piano, and is the presence of a piano pedal. There are several different types of piano pedals, but the most common are the sustain pedal, which prolongs the duration of every note of the piano when activated, and the soft pedal which softens the sound of the entire piano. Although the effects of these pedals are directly related to the articulation and dynamics of the performance, their presence (or lack of) can be seen as a crucial component of piano performance. It is common for the sustain pedal to see active use in almost all modern piano performance, even when there doesn't exist any score marking indicating it's use.

Richard: Add section and reference to the specifics of feature engineering related to both the score and the performance in the methods section

.

2.1.3 Data

The data required for EMP generation includes some digital form of representation of a score as well as a corresponding performance. Scores are typically given in the form of MusicXML, which is a text-based representation of a score. Performances could be directly be rendered as audio which is the process used by human performers with the use of an acoustic instrument. Instead of audio however, an intermediate data form, MIDI, is used to represent the performance. This better aligns with the generation process outlined in 2.1. In the full generation process, a separate model would be used to take the performance data in MIDI and synthesize that into raw audio which would be presented to the listener. Both data formats contain all of the required information to represent all of the musical components of both a score and a performance, including pitch, tempo, timing, articulation, deviation, and pedal. See appendix A.2 for more information on both MusicXML and MIDI.

To build an EMP generation model, it is necessary to run both the score and performance

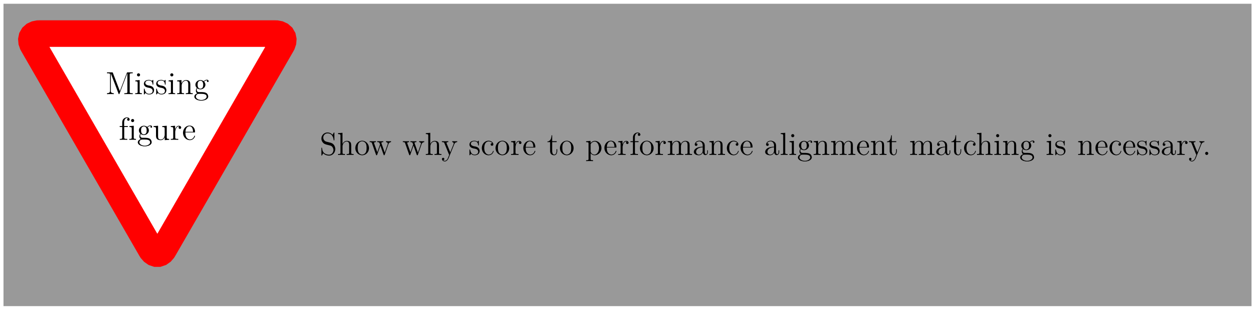

Figure 2.2: Two performances of the same score can vary wildly in their tempo and timing. This makes it necessary to have a score to performance alignment for every performance.

through a data alignment process in which every note of the performance is mapped to it's corresponding position in the score. Given the highly dynamic nature of musical performance, it is a non-trivial task to run this alignment process for a set of scores and performances, especially if the task is performed by manual human annotation. There exist methods for both manual and automatic alignment. Due to the time-consuming nature of manual alignment and the need for large data sets to build higher quality models, automatic alignment algorithms are an active area of research.

Richard: Add reference to section which gives relevant research

2.1.4 Features

A common challenge facing any application of data-driven and ML-based research is to find the correct representation of data that a model interacts with. The choice of these data representations (or features in the ML terminology) have a large impact on the results of any EMP task, irregardless of the model.

Score Features

There are some score features which are required for EMP models, which include the musical features at the lowest level of a score as explained in section 2.1.1. These are pitch and timing, and the duration of the notes. Mid-level features include concepts at the local level and have some music theoretic concepts, such as downbeat information of a given measure according to the time signature, or the tonality of a chord (tonic, dominant, etc). High-level features represent advanced music theoritic concepts that are more global to the entire piece, including abstract properties of the piece such as the emotion the piece should convey and how different sections of the piece relate to each to tell a complete story [5].

Both the mid-level and high-level features are not necessarily required for every EMP model as the lower-level features are, and are not consistent across all EMP models. It still remains an open question as to which features should be extracted from the data that the model can learn from. The lack of consistency in these features is one of the reasons that evaluation of EMP generation models is so difficult, as explained in section 2.1.5.

Richard: determine if this is the right place or not to outline the mathematical definition of the score features. Belongs either here or in the relevant work section

.

Performance Features

For western classical solo piano music, performance features are relatively simple compared to the score features as well as to other instruments. Most EMP models use the different aspects of a piano performance as explained in section 2.1.2 for their data features, including the pitch, tempo, timing (or timing deviation), articulation, and pedal. Although at an abstract level the features are the same, there are different numerical methods used to describe each of the different aspects. These are presented in

Richard: Add reference to relevant work section that goes over the different elements. This information may belong better here and then referenced in the relevant work section

.

2.1.5 Performance Evaluation

One of the most important components of any computational model performing a task is that of evaluation. Evaluation is used to determine the quality of a model, and serves as a benchmark to compare different models used in the same task. Due to the inherently subjective nature of music and musical performance discussed in 2.1, evaluation is notioursly difficult to understand and perform correctly for EMP generation models [3].

Evaluation for computational models, specifically for EMP models, is typically categorized in two ways, quantitative evaluation and qualitatiative evaluation. Quantitative evaluation methods involve using numerical metrics which are computationally generated and deterministic. Qualitiative evaluation methods usually involve some form of human feedback and judgement presented in some standardized statistical measures. The key difference between quantitative and qualitiative is that qualitative methods are not as consisten and much more difficult to reproduce, given the reliance on the subjective feedback of human listeners. Traditionally, quantitative methods are preferred because of their consistency and reliability, In the case of EMP models however, qualitative evaluation methods may be even more important in gaining an understanding of what makes one model better than another. Finding good methods of evaluation is an active area of research in EMP [3].

Quantitative

This method of evaluation is standard for ML models in general. There are a number of different metrics which are used in the evaluation process, all of which are specific to type of data and problem domain the model fits inside of. We will briefly cover the most common quantiative evaluation method that applies to our data and modeling domain, which is regression .

Richard
Add reference that discusses the feature engineering

The two common metrics used for evaluation and regression are Mean-Squared-Error (MSE) and the Pearson Correlation Coefficient, usually denoted as the R^2 score. MSE is used to measure the difference between a prediction and an actual observed target value, and can be denoted as $MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$, where Y_i is the observed value at time step i, and $\hat{Y}_1$ is the predicted value. R^2 is a probablistic measure of the linear correlation between variables X and Y, and is denoted as $\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$ where cov indicates the covariance and σ indicates the standard deviation. [4]

One of the problems with using quantiative, or "objective" evaluation methods, is that it usually invovles comparing a generated performance $\hat{Y}$ with a human performance Y. Given that no performance (or interpretation) of a can objectively been seen as better than another, this method of evaluation is also biasing the quality of a model towards some subjective view of the "correct" interpretation. Of course, a "correct" interpretation doesn't exist, which is what makes evalution methods for this particularly problem difficult.

[4] See wikipedia for more information on MSE, covariance, standard deviation, and the correlation coefficient

Qualitative

Richard: Need to conduct more research before I can write this section. Haven't done so because I won't be performing a qualitative evaluation myself in the paper. However it is still worth mentioning

2.2 Transformers

Richard: Change the outline and sections to deal with general sequence modeling. Start by talking about LSTMS and then introduce Transformers in the context of machine translation

To properly understand the significance of Transformers and their involvement in our work, it is necessary to provide context about the domain in which the Transformer was first introduced and give an overview of the existing work in that domain that the Transformer built on. We'll then provide some detail about the Transformer itself as well as adaptations of the original architectures and their results.

2.2.1 Natural Language Processing and Machine Translation

One of the most commonly studied fields in Machine Learning and Artificial Intelligence is Natural Language Process (NLP), which (similarly to MIR) uses computation to ascertain a better understanding of human language as well as build technological tools that are useful in performing common language tasks. One such task is that of machine translation, which involves using computation alone to translate text from one language to another. NLP research usually invovles building sequence-based models (which explore the individual elements of an ordered set of items) due to the inherently sequential nature of language, as opposed to a non-sequential model which doesn't account for sequential data, such as a single image. Machine translation falls under the category of sequence-to-sequence (seq-2-seq) modeling problems, which involve the mapping and relationship of one sequence to another. This is typically in the form of translating a single sentence from one language (English) to another (French).

More specifically, machine translation (and other seq-2-seq tasks) can be defined as taking an input sequence $\mathbf{x} = \{x_1, x_2, x_3, ...x_m\}$ of size m and producing an output sequence $\mathbf{y} = \{y_1, y_2, y_3, ...y_n\}$ of size n such that $M(\mathbf{x}) = \mathbf{y}$, where M can be any machine translation model. In some seq-2-seq tasks, $m = n$ are the same, implying that the input and output sequence are the same length. As is often the case in language translation, the input sentence and output sentence are of varying lengths, so we can assume that $m \neq n$.

It is common to use an encoder-decoder architecture for M, where there is an encoder E which takes in the input data and outputs and finds some hidden representation $E(\mathbf{x} = \mathbf{z})$. This hidden representation is given as input to the decoder, and the decoder uses it to produce the final output, $D(\mathbf{z}) = \mathbf{y}$. We can then define an encoder-decoder seq-2-seq model as $M(x) = D(E(\mathbf{x})) = \mathbf{y}$. Historically, a Long-Short-Term-Memory neural network (LSTM)[5] has been used for both E and D, where the hidden representation $\mathbf{z}$ has been a fixed length vector .

Richard
Add reference

One of the limitations of such a model is that it has to compress all of the information of the input data into the fixed-length vector $\mathbf{z}$ which causes the network to potentially lose important information, particularly in the case where an input sentence is given to the network which is longer than any present in the training data. Bahdanau et al. [2] present the attention mechanism which, used in conjunction with an RNN based encoder, allows for

[5] An LSTM is a common variant of a Recurrent Neural Network (RNN) which is the most standard deep learning model used for sequence modeling. See https://en.wikipedia.org/wiki/Long_short-term_memory

the hidden representation to itself be a sequence $\mathbf{z} = \{z_1, z_2, z_3, ..., z_m\}$ of size m (the same size as the input sequence). Each z_i element in the sequence contains information about the whole input sequence, with an emphasis on the elements closest to the i-th element. This allows the hidden representation to encode any relationship that one element in the sequence has with another. The decoder then uses this information to "pay attention" to words in the output sequence that have a relationship with words in the input sequence, given the context that is encoded in the hidden representation at a particular time step i. The attention mechanism and model that uses it achieved state of the art results in the machine translation task, due in part to the fact that hidden representation is not limited to a fixed-size vector. The original attention mechanism presented by Bahdanau et al. [2] and its adaptations have since been used in tandum with recurrent models to improve the state of the art in several sequence modeling tasks . One of the limitations with standard recurrent networks is their inability to retain information across long sequences - attention provides a way to create additional context and better memory across these longer sequences which has led to the increase of performance in attention-based models. .

Richard find reference

Richard find reference

2.2.2 Attention is All You Need

In the seminal paper, Vaswani et al. [24] introduce the Transformer. The Transformer is an encoder-decoder seq-2-seq modeling neural network architecture that relies solely on the use of attention and cuts out any semblance of a recurrent architecture. The Transformer was the first architecture to make use of attention by itself, and by doing so pushed the state of the art in machine translation even further than it had been with attention-based recurrent models.

The Transformer architecture consists of a stack of N layers, all of which use a combination of a self-attention (attention that applies only within a single input sequence and not between an input and a output sequence)

Richard: Explore different ways to describe self-attention. May not even be necessary at all to mention

mechanism along with a standard pointwise fully connected feed-forward neural network (FFNN). Both the encoder and decoder comprise of these attention based stacked layers. For a full description of the architecture see [24].

2.2.3 Transformer Adaptations: BERT and GPT

Of particular interest in the new Transformer modeling domain is powerful adaptations of the original architecture which have been applied to many other NLP tasks besides machine translation. On such architecture, BERT (which stands for Bidirectional Encoder Representations from Transformers), uses what can be referred to as an ”encoder only” Transformer model.

The original Transformer was built with machine translation in mind, but there are several other NLP tasks that could possibly benefti from using an attention only architecture. Some of these tasks include standard text classification, textual entailment, sentiment analysis, question answering, and many more . BERT was introduced as an encoder only transformer model that could generalize to all of these tasks. The method which it made use of was pre-training the model on a massive data set, with the intuition that by feeding the model so much data that it would learn a general representation of language that could then be applied to several different tasks. BERT is effectively a massive encoder for language in general, and can be used in conjuction with other models as simple decoders to perform these tasks. See the original paper[4] for the full architecture and details.

Richard

Find reference

Similarly to BERT, the Generative Pre-trained Transformer (GPT) architecture[22] is an adaptation of the original Transformer. The GPT architecture can be seen as a ”decoder only” transformer, and is used as a general Language Model (LM). The task of a LM is simple; to predict the next word in a sequence of given words. Given that GPT is a generative model, it employes the decoder side of the Transformer, which is responsible for actually generating the text as part of the machine translation taks. Similarly to BERT, GPT models are pre-trained on massive amounts of data to learn a general representation of language, and used in conjuction with other models to perform various tasks.

Both BERT and GPT have significatly pushed the state of art in NLP and sequence modeling in general. Their success in the domain of language presents questions about their effectiveness in other related domains, such as music.

Chapter 3

Related Work

Given the understanding of both EMP and Transformers presented in section 2, we'll

Richard: Try not to use contractions. Use we will instead of we'll

now give an overview of the existing relevant research from which we will build upon. This will include a variety of different EMP models, as well as applications of the Transformer to MIR related problems.

Richard: Put the data section in a different place

.

3.1 Existing Expressive Musical Performance Generation Models

EMP generation models fit into one of two categories, rule-based and data-based. Rule-based systems are built using a set of hardcoded rules which are derived using pre-existing musical knowledge and empirical studies involving human cognition. Data-driven models rely on probabilistic and machine learning methods to take an existing dataset of both scores and performances and use the performance data as a guide to learn the mapping between score features and performance features.

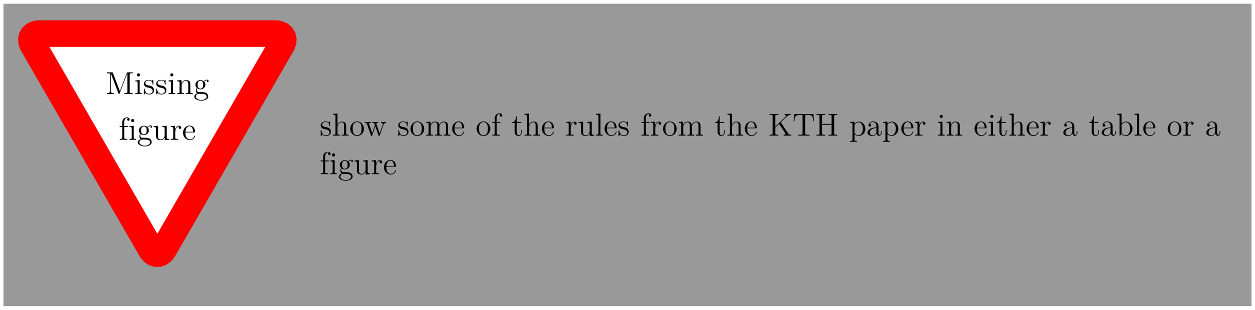

Figure 3.1: The left column shows the name of the rule, and the right column provides a language description of that rule. These are the rules that we might expect a data-based system to learn.

3.1.1 Rule Based

The KTH system [8] sits at the center of rule-based EMP models. Development of the KTH started in the 1980s and continued well into the 21st century. The initial idea behind the KTH system was to define a set of rules relating to the structure of a musical composition and how they affect a resulting performance, specifically with singing synthesis. The first set of rules was developed related for use in singing synthesis, and these rules were then later adapted to general musical performance.

Since then there have been two general methods in the continued development of the KTH rule system. The first is that of *analysis-by-synthesis*, which involved using the rules to synthesize musical performances that were presented to human listeners (both professional and non-professional), gathering listening feedback, and then using this feedback to modify the rules where needed. The second was an *analysis-by-measurement* method. This method uses direct computation to analyze the result of a computational generated performance with an existing real performance [1]. Example rules from the KTH system are found in figure 3.1
.

Richard

May need more exploration in

[1]This falls more in line with the data-driven approaches. However, data-driven models use the performance data to directly build the model, whereas the use of real performance data in the KTH system is for evaluation purposes only. Any further updates to the model still rely on a hardcoded set of rules

To our knowledge, the KTH rule-based system is the first sophisticated computational model for generating expressive performance, and its methods form the basis for much of the research that has been conducted since then. The explicitly defined rules in the KTH system can be thought of as the rules we might expect a data-based model to learn. Widmer [25] shows that data-driven methods do in fact learn some of the same rules as the KTH system, but also can learn rules that are the opposite of KTH rules. As has already been discussed, the difficult nature of model evaluation may describe this phenomenon, as there is no telling which rule is more "correct" than another. Nevertheless, the KTH rule system has been an important milestone in the development of EMP models in general.

3.1.2 Data Based

State of the art EMP generation models rely on existing data of actual human performance to learn the mapping between score and performance. The state of the art models are generally based either on sequential probabilistic or non-linear neural network methods[3], although there has been previous work with linear and non-sequential modeling. A complete overview of all relevant EMP generation models is presented in [3] and we will not iterate them here. Instead we will describe a few models and frameworks which are relevant to our work

Basis Function Models

The first of these is a complete computational and mathematical framework for exploring EMP, and is known as the Basis Model (BM) framework[5]. The BM framework for EMP describes the full end-to-end process involved both the generation and analysis of musical performance, starting with a set of Basis Function Models which are used to provide score features. The BM framework also defines *expressive parameters*, which are analogous to our definition of performance features as outlined in 2.1.2. Given score features which are defined by a set of basis functions as well a set of expressive parameters used to numerically define a performance, the BM framework then defines a model which can map between the score features and expressive parameters.

Richard: This idea needs more cohesion with the rest of the thesis. Try to provide our own mathematical definition of EMP (similarly to the way we did with neural machine translation). We could actually use the BM framework as this definition, although it may be more mathematical than we need

.

Eduardo [5] outlines the full mathematical definition of the BM framework, as well as the evolution of the framework and its application with specific feature and model definitions. BM models first started as simple linear non-sequential models which learned the linear relationship between a set of defined basis functions (or score feature) and a single expressive parameter, such as MIDI velocity. This version of the BM models each expressive parameter independently from all others, and implies that the interpretation one expressive parameter will not have an effect on the other . [2]. Both standard least squares regression and a probabilistic Bayesian approach are used to model the linear relationship.

Richard add more information about score feature

Richard Verify that footnote is correct

As the BM framework progressed, both non-linear and sequential models were introduced in the form of deep neural networks. The non-linear model was implemented first in the form of Feed-Forward Neural Network (FFNN) was implemented first and showed an increase in goodness-of-fit as well as predictive accuracy over the standard linear models. After the FFNN came a standard RNN and was used in conjunction with the FFNN with features where time-dependent and the sequential nature of music was relevant. The recurrent non-

[2] Although this is not necessarily the case in actual performance, it is a simplifying mathematical trait that makes the development and interpretation of the models simpler. All of the BM models operate under this same assumption

linear model performed the best relative to all other models.

virtuosoNet

Richard: Change virtuosoNet heading to look better. Also look into creating a macro for virtuosoNet to create a typeset so that the name stands out

. Similarly to the BM framework, the development of virtuosoNet is gradual. The first version of the model presented in [13] uses a recurrent hierarchical attention network (HAN) along with a novel encoder-decoder architecture specific to the EMP domain. No quantitative or qualitative evaluation results are presented at this point. The next iteration of virtuosoNet uses a similar encoder-decoder architecture but introduces an iterative sequential graph-based neural network (ISGN) that relies on the score representation as a graph data structure [15]. The latest version presented in [14] returns to the HAN architecture, but does so with a larger dataset as well as additional more abstract hierarchical models that are hypothesized to create better structure at the metrical level and preserve patterns across mid-level structures of the composition, in addition to learning them at the low-level.

Both the ISGN[15] and HAN[14] version of virtuosoNet are trained on the same dataset (which we will describe in section) and evaluated quantitatively using MSE and qualitatively using listening tests. In terms of quantitative evaluation, both the ISGN and HAN perform better than baseline models which remove some of the architecture complexity related to hierarchical layers. The final version of HAN reports better MSE metrics than ISGN. The qualitative evaluation with listening tests shows that both ISGN and HAN perform better than baseline models as well as better than the "deadpan" performance, which is a performance model that is statically computed using a simple set of rules and gives a somewhat robotic-sounding performance

Richard add section

Richard: Provide more explanation for the deadpan recording. May be worth it to mention in the qualitative evaluation section

. The final HAN version's qualitative evaluation includes a comparison between the HAN and the publicly available version of the BM framework model [3].

The results in [14] show that the HAN performs better than the BM model. There are many plausible reasons that may explain the difference in results other than the HAN being a superior model to the BM, including differences in the training data for both models, bias of the qualitative method towards the HAN, and the fact that the opinion of the members of the listening test doesn't necessarily imply one model being "superior" to another. However, given the results presented by Jeong et al. [14], we will assume that this version of the HAN represents the current "state of the art" in the field, if such a thing even exists.

Richard: Add section that talks about the features used for virtuosoNet

3.2 Datasets

One of the problems facing EMP and MIR in general is the lack of high quality and high scale datasets[3]. This is in large part due to the fact that scope of possible data to collect related to music data is large, compared to other domains. As has already been discussed, there are different stages in the musical process, and each of them contain different possibilities for the representation of music. For example, composition can contain largely the same amount of information in at least three forms. The first and most common is the symbollic representation in the form of a data format like MusicXML. A musical performance also contains within it information about the composition itself, and performances can represented

[3] The website for the BM model can be found https://basismixer.cp.jku.at/static/app.html. At the time of this writing, the website is currently unavailable

in an intermediate format such as MIDI, or in the form of raw audio. The same can be said for other fields such as NLP, which deals mostly with textual data, and Speech Processing, which deals mostly with language in the form of spoken word. However, the two fields are seen as distinct from each other and each come with more standardization in both research methods and data formats. Musical data and information has not seen the same rigour in the literature.

Another inherent problem with getting high-quality musical datasets is that most of the readily available musical data comes in the form of audio, which is much more difficult to process than symbolic (MusicXML) or intermediate (MIDI) forms given that it contains large amounts of noise and does not necessarily compress musical information. In contrast, NLP and Computer Vision directly deal with text and image data respectively, which are both readily available at a large scale due to the internet.

There are normally 3 required components for a EMP dataset.

1. Scores (usually in the form of MusicXML)
2. Performances (usually in the form of MIDI)
3. Metadata about the matching alignment between the score and performance.

Score data is usually gathered by finding readily available MusicXML files from open source software projects which contain music that is in the public domain (which all western classical music is) [4], or by using Optical Music Recognition (OMR) to auotomatically scan paper sheet music into a digital form followed by manual corrections where needed. Because the relevant performance features are difficult to extract from raw audio, performance data usually comes in the form of MIDI. To gather MIDI data of professional performance, it is necessary for the

[4] MuseScore is the most common. Also see the International Music Score Library Project

performers to play on a computer-controlled piano which can record performances in MIDI form, as well as automatically play back recorded performances which allow the complete reproducibility of any existing performance. Both the Yamaha Disklavier [5] and the older Bosendorfer CEUS system have this capability.

There is no standardized method for score-to-performance alignment methods and data representations. Each dataset presents in own alignment method as well as the metadata that represents the alignment.

To provide context for the progression of data used in EMP generation, we'll start by touching an older dataset used in older EMP research, the Magaloff Corpus, and then describing a much larger scale dataset, the Piano-e-competition, which has recently been adapted for use in EMP generation. A full overview of datasets for EMP generation can be found in [3]

Magaloff Corpus

Nikita Magaloff was a Russian pianist known for his performance cycles of Chopin's entire works for the solo piano. In one of his final cycles of performances recorder in 1989, he played on a Bosendorfer SE computer-controlled piano. The Magaloff Corpus [6] presented the recorded performances were converted to the standard MIDI format[5], thus making available full performance data of all of Chopins compositions for solo piano. Score data was obtained using OMR with manual corrections where needed. The alignment method presented in [9] was used to produce the note-matching annotations, along with manual correction. The dataset contains over 10 hours of playing, 150 compositions, and over 320,000 performed notes. The corpus however, is not publicly available, and has only been used in research by Flossmann et al. [6] and colleagues [5].

[5]https://usa.yamaha.com/products/musical_instruments/pianos/disklavier/index.html

Piano-e-competition

As has been discussed, there is a large push in modern MIR to produce high-quality large datasets. At the heart of this research in MIR is the Piano-e-competition. Started in 2002, it is an international piano competition which attracts some of the promising up and coming musicians at both the senior and junior level [1]. Every performance from the competition is played on a Yamaha Disklavier. Every performance from the competition dating back to 2002 is recorded in both MIDI and audio. Hawthorne et al. [11] introduce the MAESTRO dataset, which presents both MIDI and audio data from the Piano-e-competition in a canonical and easily accessible form. The dataset was first used to build a full musical analysis and generation process framework named wav2midi2wave, which includes a musical transcription process [10] from raw audio to midi (wav2midi), a direct musical composition and performance generation model [12] [6] (can be seen as the midi or midi2midi part of the wav2midi2wav framework), and a synthesis model that takes MIDI and generates raw audio[21] (midi2wav).

The Piano-e-competition also forms the basis for the data used to train virtuosoNet. The Piano-e-competition dataset itself does not provide any data about the scores of the music being performed; this data was collected by Jeong et al. [14] mostly from MuseScore. On top of gathering the score data for all performances in the Piano-e-competition, Jeong et al. [14] also run the automatic score-to-performance alignment algorithm presented in [19] to provide metadata about the alignment between each score and performance. Score-to-performance alignment is error-prone (especially in the case of performance mistakes) and as result, there are some performance notes which are not aligned to those in a score. Jeong et al. [14] also perform additional manual and heuristic corrections to the alignment where needed. The

[6] This model directly generates MIDI files without using scores. It simultaneously generates a composition and performance. This task can been seen as a merging of the two separate tasks, composition and performance, as show in figure 2.1

major difference between the Piano-e-competition and the Magaloff corpus is that it contains multiple performances for the same score, whereas the Magaloff Corpus has a 1-1 mapping between a score and performance. The dataset presented in [14] has 226 scores across 16 different composers, roughly 660,000 score notes, and around 3,500,000 performance notes. The number of matched performance notes is ten times larger than the Magaloff Corpus, and all data is made publicly available [7].

The Aligned Scores and Performances (ASAP) dataset [7] is a recent adaptation of both the dataset presented by Jeong et al. [14] and the MAESTRO dataset. It uses the MusicXML files from Jeong et al. [14], audio from the MAESTRO dataset, and MIDI files from both sources which are extracted from the common source of the Piano-e-competition. It provides additional alignment metadata for both MIDI and audio fields, as well as more manual correction in the MusicXML score files. Although the purpose of the ASAP dataset is for Automatic Music Transcription (AMT), which is the task of transcribing a score from a performance (either in audio or MIDI form) [8], it is just as equally useful for EMP generation. To our knowledge it has not been used in any EMP generation task. Although it is largely similar to the dataset used to train virtuosoNet, the implications of it's extensions have yet to be determined in EMP.

[7]The dataset can be found at https://github.com/mac-marg-pianist/chopin_cleaned

[8]AMT can be seen as the "opposite" of EMP, given that transcription does the reverse process of performance by mapping a performance to a score

Chapter 4

Methods and Experiments

Given the relevant background research and knowledge base, we will now describe the experiments we ran and the reason behind our experimental methods. Given the powerful advances in NLP due to the Transformer discussed in 2.2, our general goal was to investigate the results of the Transformer in application to EMP generation, which to our knowledge has never been done. Because both language and music are highly sequential and hierarchical in nature, our intuition was that because the Transformer does a good job of learning the general structure of language, that it can do the same of for music. We use the general framework for a complete end to end performance generation system which is proposed by virtuosoNet. In its simplicity, the initial purpose of this project was to determine if a Transformer based model can improve upon virtuosoNet, given the same data, features, and evaluation metrics

Richard: Make sure to add a section about feature engineering with virtuosoNet

. However, due to the highly ambiguous and subjective nature of EMP generation, there was no clear way to know if we confidently answer this question given our results. As such, we modified our research direction to providing additional insight and intuition about the nature of EMP generation itself and how this intuition can guide future work.

4.1 Data and Features

The reasons for the adoption of the virtuosoNet system are twofold: the first being that the dataset used to develop virtuosoNet was the largest publicly available dataset used in EMP generation, and the second being that the code and models of virtuosoNet are open-sourced [1] and contain all of the necessary data processing. This system also somewhat represents the ”state of the art” in EMP generation, so it provides a natural starting place to use for comparison against any further model development. The virtusoNet system uses handcrafted features for both scores and performances. Score features contain low-level information (pitch and timing), high-level information such as the key and metric information, as well as more detailed information such as the duration of rests, articulation markings (legato and staccato), and the distance from the closest preceding tempo and dynamics directions, slur, and beam status. The performance features include all of the standard performance features: tempo expressed as BPM, note onset deviation, MIDI velocity, articulation, and different features related to the onset and offset times of the pedal. A full outline of the features is given in [16].

4.2 Model

In the virtuosoNet system, there is a 1:1 mapping between notes in scores and performances. The original Transformer as an encoder-decoder model was designed as a seq-2-seq model where the sequences have different lengths, which adds additional complexities into the model to account for this difference. To keep our system simple, our model is conceptually similar to BERT, and acts as an encoder-only Transformer model. It contains a simple fully connected linear layer on top which will learn the final mapping between the Transformer encoding

[1] https://github.com/jdasam/virtuosoNet

and the actual score features. We use the standard absolute positional embedding which is concatenated with the score features as input to the model. The performance output features of the model can be used to construct a MIDI file, allowing for the system to performance full performance generation given a score in MusicXML form.

4.3 Experiments and Model Evaluation

virtuosoNet is built as a regression model and uses MSE as both its loss function and evaluation metric. It uses an 8-1-1 train/valid/test data split, and Jeong et al. [14] present MSE results for each different parameter of the performance features on the test set. We follow the same method for our quantitative evaluation. Most models were trained at 50 epochs, and the best model parameters were selected according to the lowest validation evaluation score. We used the software Neptune AI [20] to manage our experiments and report the metric feedback. Data for all of the experiments we ran including the model hyperparameters and metrics can be found online[2].

We ran experiments using the same data for several different model configurations. Similarly to our Transformer encoder model, we build an LSTM baseline model with 3 recurrent layers which acts as an encoder and a simple fully connected linear on top to perform the final mapping between the LSTM encoding and the output features. The LSTM baseline is 3 layers with a hidden size of 256, and is used for comparison purposes only.

[2]http://ui.neptune.ai/richt3211/thesis

4.3.1 Quantitative Evaluation

We use several different model configurations for the Transformer. Our Transformer baseline has 6 layers, 6 attention heads, and a hidden size of 256. We chose this as a base configuration because it closely matches the size of the original Transformer [24], except for the hidden size of the feed-forward layer. We chose a smaller hidden size of 256 for our base layer hidden size due the relatively small dimensionality, 78, of the input data to the model. Our initial goal was to find the optimal model configuration according to a quantitative evaluation, which meant finding the lowest total MSE loss. To keep our modeling honest, we withheld from running the final test evaluation until all models had been trained. As such, we needed a way to compare against existing virtuosoNet models without using the final MSE evaluation metrics reported by Jeong et al. [14]. To do this, we trained from scratch the Iterative Sequential Graph Network (ISGN) and the HAN baseline (HAN-BL) models as reported in [15] and [14] respectively, and used the metrics from the validation data set to guide our own model development before we ran the final evaluation. We ran a fairly exhaustive search of a single dimension of the hyperparameters at a time. The full results of these models are show in table 5.1 .

Richard add table reference

4.3.2 Qualitative Evaluation: Identifying Training Problems

During our model development we conducted our own listening tests to subjectively determine the quality of each model. It was apparent from the start that there was a mismatch between the quantitative results of the model and it's quality in a listening test. For example, the Transformer model with N_{id} 125 (which we will denote as $T_{N_{125}}$, see Table 5.1) had much worse validation MSE metrics than almost every other model Transformer model. However, a listening test revealed that the absolute tempo for smaller models, such as the

Transformer baseline $T_{N_{147}}$, was much faster and sounded worse (to the point where the performances are almost 'unlistenable') than $T_{N_{125}}$, even though it presented better quantitative metrics on the validation test set. The potential disconnect between the 'quality' of the model as determined by quantitative and qualitative evaluation led us to investigate potential problems with the training methods used by Jeong et al. [14]. See section for a more in-depth discussion of this evaluation. Richard Add ref

One of the first potential problems we identified was the method used to calculate and interpret the loss and evaluation. The output features of virtuosoNet are represented by a sequence of vectors with a length of 11. The first 4 features are values that correspond to a single expressive parameter, and are tempo, velocity, deviation, and articulation, respectively. The last 7 features are all different numbers that correspond to information about the pedal [16]. Jeong et al. [14] present MSE metrics for five different expressive parameters, which include all of those previously mentioned, as well as the pedal. This means that when we refer to the pedal MSE, it is an aggregation of the 7 different features that contain pedal information. The original MSE which was used to train virtuosoNet assumed that every feature of the output vector contributed equally to the final output and corresponding loss optimization. Given that there is 7 times more information for the pedal parameter than all others, we can think of this loss function as placing much more importance for the pedal than every other expressive feature. To combat this, we came up with a new weighted MSE loss function that allows for the optimization of some features over another.

We define the output vector as an 11 dimensional vector $\mathbf{v} = \{t, v, d, a, p_0, p_1, p_2, p_3, p_4, p_5, p_6\}$ where t, v, d, and a represent tempo, velocity, deviation, and articulation respectively, and p_i represents a single component of the pedal. For a predicted output vector $\mathbf{v}$ and the target output vector $\widehat{\mathbf{v}}$, standard MSE loss is $MSE(\mathbf{v}, \widehat{\mathbf{v}}) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{v}_i - \widehat{\mathbf{v}}_i)^2$. This can also be rewritten as $MSE(\mathbf{v}, \widehat{\mathbf{v}}) = \frac{1}{11}[(\mathbf{v}_t - \widehat{\mathbf{v}}_t)^2 + (\mathbf{v}_v - \widehat{\mathbf{v}}_v)^2 + (\mathbf{v}_d - \widehat{\mathbf{v}}_d)^2 + (\mathbf{v}_a - \widehat{\mathbf{v}}_a)^2 + \sum_{i=1}^{7}(\mathbf{v}_{p_i} - \widehat{\mathbf{v}}_{p_i})^2]$.

We introduce 5 different weight values: $\alpha_t, \alpha_v, \alpha_d, \alpha_a$ and α_p. Our weighted MSE loss is defined as $W_{MSE}(\mathbf{v}, \widehat{\mathbf{v}}) = \frac{1}{\alpha_t+\alpha_v+\alpha_d+\alpha_a+\alpha_p}[\alpha_t(\mathbf{v}_t - \widehat{\mathbf{v}}_t)^2 + \alpha_v(\mathbf{v}_v - \widehat{\mathbf{v}}_v)^2 + \alpha_d(\mathbf{v}_d - \widehat{\mathbf{v}}_d)^2 + \alpha_a(\mathbf{v}_a - \widehat{\mathbf{v}}_a)^2 + \alpha_p \sum_{i=1}^{7}(\mathbf{v}_{p_i} - \widehat{\mathbf{v}}_{p_i})^2]$. The original MSE can be seen as the weighted MSE with $\alpha_t, \alpha_v, \alpha_d, \alpha_a = 1$, and $\alpha_p = 7$. If we conceptualize the loss optimization in this way, we can view the original model optimization as placing much more importance towards accuracy in the pedal than any other feature of expression. The MSE was used not only as the loss function to optimize the model, but also as the actual metric to evaluate the model with. This evaluation means that models with an emphasis in pedal accuracy will be preferred over those without it. This presents the question of determining whether or not this is the right way to conceptualize a 'good' model. Would a different configuration of the expressive feature weights lead to a better outcome? These answers are non-trivial and this further emphasizes the importance of having better ways to both optimize and evaluate EMP generation models.

We also changed the way in which the loss was calculated for the articulation feature. As discussed in section 3.1.2, virtuosoNet uses an alignment algorithm which presents metadata about the alignment between the score and performance of every single note. The notes that are not aligned are included in the input data to the model, but are excluded from the loss calculation. A similar method is used with notes relating to the articulation feature and the pedal. Jeong et al. [14] say "Since the articulation is largely affected by the sustain pedal, we reduced the weight for the articulation loss to 0.1 for notes with the sustain pedal pressed at the offset". In the actual data generation code the weights for the articulation loss calculation are slightly more complicated than what is presented in the paper [3], but the intuition behind changing the loss for notes used in combination with pedal is the same. For some experiments we change this loss calculation for articulation to fall in line with all

[3] See github

other performance features which involves using alignment data only to exclude notes from the loss. Again, this type of optimization and evaluation is subjective and it is hard to say if one is more correct than another.

Changing the loss function (as well as the evaluation function) in such ways alters the interpretability of the metric and invalidates the direct comparison to the metrics reported for virtuosoNet. However, we can still compare model outputs qualitatively. Due to time and resource constraints, no sophisticated qualitative evaluation method was used to conduct this comparison. Our qualitative evaluation relied mostly on the authors own listening tests and internal discussions about the quality of the performances and potential places for error. The listening tests were conducted by comparing performances of 6 different compositions for each model both audibly and visually using the Digital Audio Workspace (DAW) software Logic Pro X. No numerical or statistical observations are reported given the fact that all evaluation was done by the author and represents an inherent bias which and cannot be seen as robust or reliable for further analysis. We do however provide some observations related to our qualitative evaluation along with our own intepretation of them, if for no other reason than to guide the intuition behind more robust methods for future work. This analysis is given in section .

Richard

Add reference

Chapter 5

Results

5.1 Quantitative Evaluation Results

5.2 Qualitative

Give personal qualitative report.

Model Configuration								Results in MSE					
N_{id}	Model	L	d_{hid}	D	LR	C	H	Total	Temp	Vel	Dev	Art	Pedal
123	LSTM	3	256	0.1	0.1	0.5		y	-	-	-	-	-
147	T-BL	6	256	0.1	0.00003	0.5	6	y	-	-	-	-	-
169			128					n	-	-	-	-	-
128			528					y	-	-	-	-	-
133			1024					y	-	-	-	-	-
118		12						y	-	-	-	-	-
—		24						n	-	-	-	-	-
132							13	y	-	-	-	-	-
171				0.2				n	-	-	-	-	-
173					0.01			n	-	-	-	-	-
—							26	n	-	-	-	-	-
134		12	528					y	-	-	-	-	-
—		12					13	n	-	-	-	-	-
135		12	528				13	y	-	-	-	-	-
125		24	528					y	-	-	-	-	-
73	HAN-BL	-	-	-	-	-	-	y	-	-	-	-	-
69	ISGN	-	-	-	-	-	-	y	-	-	-	-	-

Table 5.1: Some Caption

Chapter 6

Discussion

Richard: The following are some interesting discussion ideas that have come up so far. There is no telling if these will be in the final paper or not after conducting more experiments.

- Transformer performs worse according the quantitative metrics. This could be because it doesn't build in a specific hierarchical layer that is specific the problem. It is a much more generic model. There is a lot of room for exploration into experimenting with different architectures based on the Transformer to better fit the problem domain.

  Richard: Add more discussion based on more results

- Transformer appears to be a more dynamic model than the recurrent virtuosoNet model that makes more "mistakes". Does this mean that it is more "human".

Richard

Add discussion of uncanny valley

- Pedal in performance is messy. Could be because of problems in the feature and modeling, or could just be because it is a difficult problem to model.

-

  Richard: Discussion on qualitative results

Appendices

Appendix A

Appendices I

A.1 Musical Concepts and Terminology

A.1.1 Pitch

The first and most basic component in music is pitch. Pitch is a perceptual property of sounds that relates to the physical frequency of a sound vibration [18]. It is what determines whether or not a sound can be though of as "high" or low". The most commonly known way to conceptualize pitch is the 88 different keys on a piano keyboard, where each key represents a different patch value. Pitch is most commonly labeled using scientific pitch notation, which couples a range of letters (A to G) with a range of numbers (zero to eight) that correspond to different octave ranges [1]. The most well known pitch is C4, or "middle C", and lays in the very center of a standard 88 key piano.

Richard

Create or find visualization

A.1.2 Tempo and Timing

Tempo in music describes the rate at which notes are played, and timing describes when a particular note should be played relative to the start of the composition. They are best explained in the context of modern western musical notation introduces the idea of note

[1]https://en.wikipedia.org/wiki/Scientific_pitch_notation

durations, time signatures, measures, and beats [2].

Richard: Find a more intuitive way to explain this. The piano roll explanation and visualization may work better

Each composition is broken down into a sequence of measures, and the time signature defines how many beat exist per measure, as well as the duration of a single beat. For example, a 4/4 time signature indicates that there are 4 beats per measure (the top half of the time signature), and that the duration of each beat is represented by a quarter note. A 3/4 time signature would indicate only 3 beats per measure, with the beat duration represented by a quarter note. The timing of a note would refer to it's measure, beat, and note duration. Tempo is most commonly given in beats per minute (BPM). A composition with a 4/4 signatue and a 120 BPM would mean that after one minute, 30 measures of the composition should have been played so far.

Richard create or find visualization

A.1.3 Dynamics

Dynamics can simply be thought of as how loud or soft a note should be played (or has been played).

A.2 Data Representation

A.2.1 MusicXML

A.2.2 MIDI

[2] See https://en.wikipedia.org/wiki/Musical_notation#Modern_staff_notation for a more detailed explanation

Bibliography

[1] 2002 international piano-e-competition. URL https://www.yamahaden.com/midi-files/item/2002-international-piano-e-competition.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] Carlos E Cancino-Chacón, Maarten Grachten, Werner Goebl, and Gerhard Widmer. Computational models of expressive music performance: A comprehensive and critical review. *Frontiers in Digital Humanities*, 5:25, 2018.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Carlos Eduardo. *Computational modeling of expressive music performance with linear and non-linear basis function models*. PhD thesis, JOHANNES KEPLER UNIVERSITY LINZ, 2018.

[6] Sebastian Flossmann, Werner Goebl, and Gerhard Widmer. The magaloff corpus: An empirical error study. *Proceedings of the 11th ICMPC. Seattle, Washington, USA*, 2010.

[7] Francesco Foscarin, Andrew Mcleod, Philippe Rigaux, Florent Jacquemard, and Masahiko Sakai. Asap: a dataset of aligned scores and performances for piano transcription. In *ISMIR 2020-21st International Society for Music Information Retrieval*, 2020.

[8] Anders Friberg, Roberto Bresin, and Johan Sundberg. Overview of the kth rule system for musical performance. *Advances in Cognitive Psychology*, 2(2-3):145–161, 2006.

[9] Maarten Grachten et al. *Expressivity-aware tempo transformations of music performances using case based reasoning*. Universitat Pompeu Fabra, 2006.

[10] 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. *arXiv preprint arXiv:1710.11153*, 2017.

[11] 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. *arXiv preprint arXiv:1810.12247*, 2018.

[12] 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. *arXiv preprint arXiv:1809.04281*, 2018.

[13] Dasaem Jeong, Taegyun Kwon, and Juhan Nam. Virtuosonet: A hierarchical attention rnn for generating expressive piano performance from music score. In *NeurIPS 2018 Workshop on Machine Learning for Creativity and Design*, 2018.

[14] Dasaem Jeong, Taegyun Kwon, Yoojin Kim, Kyogu Lee, and Juhan Nam. Virtuosonet: A hierarchical rnn-based system for modeling expressive piano performance. In *ISMIR*, pages 908–915, 2019.

[15] Dasaem Jeong, Taegyun Kwon, Yoojin Kim, and Juhan Nam. Graph neural network for music score data and modeling expressive piano performance. In *International Conference on Machine Learning*, pages 3060–3070, 2019.

[16] Dasaem Jeong, Taegyun Kwon, Yoojin Kim, and Juhan Nam. Score and performance features for rendering expressive music performances. In *Proc. of Music Encoding Conf*, 2019.

[17] Shulei Ji, Jing Luo, and Xinyu Yang. A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions. *arXiv preprint arXiv:2011.06801*, 2020.

[18] Anssi Klapuri. Introduction to music transcription. In *Signal processing methods for music transcription*, pages 3–20. Springer, 2006.

[19] Eita Nakamura, Kazuyoshi Yoshii, and Haruhiro Katayose. Performance error detection and post-processing for fast and accurate symbolic music alignment. In *ISMIR*, pages 347–353, 2017.

[20] neptune.ai. Neptune: experiment management and collaboration tool, 2020. URL https://neptune.ai.

[21] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[22] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[23] Orjan Sandred, Mikael Laurson, and Mika Kuuskankare. Revisiting the illiac suite–a rule-based approach to stochastic processes. *Sonic Ideas/Ideas Sonicas*, 2:42–46, 2009.

[24] Ashish Vaswani, 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*, 30:5998–6008, 2017.

[25] Gerhard Widmer. Machine discoveries: A few simple, robust local expression principles. *Journal of New Music Research*, 31(1):37–50, 2002.

[26] Gerhard Widmer. Getting closer to the essence of music: The con espressione manifesto. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(2):1–13, 2016.