ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Piano Music Generation with Deep Learning Transformer Models

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Άγγελος Κ. Στάης

Επιβλέπων : Γεώργιος Στάμου

Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2023

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Piano Music Generation with Deep Learning Transformer Models

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Άγγελος Κ. Στάης

Επιβλέπων : Γεώργιος Στάμου

Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 13η Μαρτίου 2023

.....	.....	.....
Γ. Στάμου	Α. – Γ. Σταφυλοπάτης	Α. Βουλόδημος
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Επ.Καθηγητής Ε.Μ.Π.

.....

Άγγελος Κ. Στάης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Δήλωση Μη Λογοκλοπής και Ανάληψης Προσωπικής Ευθύνης

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

Περίληψη

Ο William Wordsworth έγραψε στο "The Solitary Reaper": "Τη μουσική στη καρδιά μου την έφερνα, πολύ μετά που δεν ακουγόταν πια". Ο συναισθηματικός αντίκτυπος της μουσικής, ιδίως της μουσικής για πιάνο, ξεπερνά τα πολιτισμικά σύνορα και αγγίζει ανθρώπους από όλα τα μονοπάτια της ζωής εδώ και αιώνες. Τα τελευταία χρόνια, η σύνθεση μουσικής με τη βοήθεια της τεχνητής νοημοσύνης έχει αναδυθεί ως επιστημονικός τομέας έχοντας συγκεντρώσει σημαντική προσοχή. Μεταξύ των πολλαπλών μοντέλων βαθιάς μάθησης που προτείνονται, το Transformer αποτελεί μια εξέχουσα αρχιτεκτονική για τη δημιουργία μεγαλύτερης διάρκειας συνθέσεων πιάνου. Η παρούσα εργασία εμβαθύνει στις δυνατότητες σύνθεσης συμβολικής μουσικής για πιάνο με δύο αξιοσημείωτα μοντέλα Transformer, το Music Transformer και το Perceiver-AR. Διερευνούμε την εκπαίδευση αυτών των μοντέλων σε έξι διαφορετικά σύνολα δεδομένων διαφόρων μεγεθών και μουσικών ειδών, παράγουμε ένα μεγάλο αριθμό συνθέσεων για κάθε εκπαιδευμένο μοντέλο και τα αξιολογούμε αντικειμενικά και υποκειμενικά. Εξετάζουμε την επίδραση των συνόλων δεδομένων, των ειδών μοντέλων και των συγκεκριμένων εκπαιδευμένων μοντέλων στη διαδικασία σύνθεσης μουσικής. Υπολογίζουμε επίσης τις συσχετίσεις των αντικειμενικών και υποκειμενικών μετρικών αξιολόγησης και προτείνουμε ένα σύνολο πρωταρχικών υποκειμενικών δεικτών ποιότητας για τη παραγόμενη μουσική από μοντέλα τεχνητής νοημοσύνης. Τέλος, συζητάμε για πιθανές βελτιώσεις και πεδία μελλοντικής έρευνας.

Λέξεις Κλειδιά: Τεχνητά Νευρωνικά Δίκτυα, Βαθιά Νευρωνικά Δίκτυα, Βαθιά Μάθηση, Μουσική, Συμβολική Μουσική, Μοντέλα Transformer, Πιάνο, Σύνθεση Μουσικής, Παραγωγή Μουσικής, Υποβοήθηση Σύνθεσης, Αναπαράσταση Μουσικής, Music Transformer, Perceiver AR

Abstract

William Wordsworth famously wrote in "The Solitary Reaper": "The music in my heart I bore, long after it was heard no more". The emotional impact of music, particularly piano music, has transcended cultural barriers and touched people from all walks of life for centuries. In recent years, the composition of music with the assistance of artificial intelligence has become a notable area of interest that has garnered significant attention. Among multiple deep learning models proposed, the Transformer has been a prominent approach for generating longer piano performances. This thesis delves into the capabilities of symbolic piano music generation with two noteworthy Transformer models, Music Transformer and Perceiver-AR. We explore training these models in six different datasets of various sizes and musical genres, generate large-scale number of outputs for each trained model and evaluate them objectively and subjectively. We investigate the impact of training datasets, model types and trained models on the music generation process. We also examine the correlations of objective and subjective evaluation metrics and propose a set of primary subjective quality indicators for music generated by artificial intelligence models. Finally, we suggest possible improvements and areas for future research.

Keywords: Artificial Neural Networks, Deep Neural Networks, Deep Learning, Music, Symbolic Music, Transformer Models, Piano, Music Synthesis, Music Generation, Composition Assistance, Music Representation, Music Transformer, Perceiver AR

Table of Contents

List of Tables & Figures

Introduction

Background & Motivation

Music composition with the assistance of artificial intelligence is a rapidly growing field that has garnered attention from researchers, musicians, and industry professionals recently. These systems can broadly be divided into two categories according to their objectives: autonomous music-making systems and systems that assist human musicians, as (J. P. Briot, 2021; J.-P. Briot et al., 2020) discuss.

While the idea of autonomous music-making systems is intriguing, it is important to consider the role of human creativity and intuition in the music-making process. Humans have a unique capacity for musical expression that is rooted in our emotional experiences and cultural heritage. Music that is generated purely by artificial intelligence models may lack the emotional depth and complexity that comes from human experience.

Conversely, composition assistance systems involve musicians composing music gradually with the aid of a music generation environment that offers suggestions, completion, and complementation. These systems are designed to provide people with the means to compose music more efficiently, enabling collaboration and experimentation, and promoting exploration of new musical horizons.

This project research aim is to create a composition assistance system for symbolic piano music generation that would support amateur musicians in their musical pursuits. Our inspiration was rooted in the desire to provide musicians in the music field with an accessible and intuitive platform for musical expression and creativity. Our goal is to develop a system that would empower amateur musicians to unleash their musical potential and create unique and original pieces of music, making the music creation process more accessible but also more enjoyable, thereby bringing our musical visions to life.

Objectives & Scope

To achieve the goal of the project, the following objectives have been identified:

- Train Music Transformer and Perceiver-AR models on six datasets of various sizes and musical genres.
- Generate a large number of outputs for each trained model.
- Conduct objective and subjective evaluation of the training datasets and model outputs.
- Investigate how the attributes of training datasets, model types, and trained models impact the quality of musical outputs generated.
- Estimate the correlations between evaluation metrics and explore possible causal relationships.
- Propose a set of primary subjective quality indicators for music generated by artificial intelligence models.

Structure of the Thesis

This thesis is divided into three main parts of a total of 11 chapters, each of which focuses on a particular aspect of the research.

Part I, which includes three chapters, is a Literature Review that provides an introduction to music theory elements such as basic musical elements, texture in music, and emotion in music. It also covers deep learning music generation principles, including deep learning models, music representation methods, model training, and evaluation methods. In addition, it provides an overview of the Music Transformer and Perceive-AR models.

Part II, which consists of five chapters, is titled Methods and Results. Chapter 5 discusses the training datasets and music representation methods, while Chapter 6 covers the training of the models, including their hyperparameters, architecture, and parameters, as well as the training process. It also presents the produced learning curves, and trained models selected. Chapter 7 focuses on output generation, while Chapter 8 and Chapter 9 cover objective and subjective evaluation, respectively. Each chapter in this part includes an implementation section.

Part III, Conclusions and Discussion, includes two chapters. Chapter 10 summarizes the conclusions regarding training datasets, model types, trained models, and evaluation metrics correlations. Chapter 11 is a discussion of the problems faced during the research and potential areas for improvement, as well as limitations and future directions in the field.

Part IV is an extended summary of the thesis in Greek, with the same structure as the original text, including a summary of each chapter and sub-chapter.

Finally, the implementation code for the project is available on the GitHub repository https://github.com/aggelostais/piano-music-generation and is discussed in separate implementation sub-chapters in Part II. Additionally, the listening test page for the subjective evaluation is https://ai-music-generation-survey.vercel.app/, and its code is provided on the GitHub repository https://github.com/aggelostais/ai-music-generation-survey.

Part I: Literature Review

Chapter 1: Music Theory Elements

The art of music is intricate and multifaceted, incorporating many different elements and principles. Understanding the principles of music requires a deep understanding of the many different elements that make up music, and how they interact with one another. Some of these principal concepts include pitch, rhythm, harmony, timbre, form and melody, being often interconnected and overlapping in practice. For example, the rhythm of a piece of music can be closely tied to its melody, and the harmony of a piece can be influenced by its rhythm and melody.

Due to these facts, studying music requires an understanding of not only the theoretical foundations of music but also how those fundamentals interact with one another and with other fields of study. Additionally, music is a form of art that is constantly evolving. New styles and trends emerge all the time, and this leads to changes in how music is perceived and made. Furthermore, the way people perceive music is also complex, being influenced by cultural, social, personal and historical factors, which makes it difficult to establish and define universal music principles. As a result, the fundamentals of music are not set in stone but rather undergo continuous change, adding to their complexity.

In music theory, a number of fundamental concepts of music are frequently examined. We intend to focus on few fundamental elements that are widely accepted and would be helpful throughout our work.

Basic Musical Elements

Melody: **The progression of musical pitches that the listener perceives as a single entity or pattern.** It is considered the horizontal aspect of music, as it moves through time. Also, is the part of the song that is most frequently recognized and remembered in memory.

Melodiousness: **The perceived quality of a melody or tune as being pleasing and harmonically rich**. A melody is considered melodious if it flows smoothly, is easy to sing or hum, and has a harmonically rich sound. **Melodiousness refers primarily to the perceived quality of a melody, but it can also be influenced by the accompanying harmony.** Harmonious chords and chord progressions can enhance melodiousness, while dissonant chords can detract from it.

Harmony: **The combination of pitches played simultaneously that create chords and chord progressions.** It is considered the vertical aspect of music, as it is based on the simultaneous playing of several notes. It gives music its sense of stability and structure and establishes a sense of tonality producing tension and release.

Harmonicity: **The extent pitches played simultaneously in a chord or melody sound pleasing and in tune.** This relationship determines the level of consonance or dissonance in a chord or melody, with consonant sounds being harmonically pleasing and dissonant sounds being harmonically unstable.

Rhythm: **The pattern of strong and weak beats and the way they are grouped in time.** It is the organization of time in music and refers to the way in which musical sounds are organized and structured over time. It gives music its sense of movement and flow, and is what motivates us to clap, dance, or tap our feet to the music.

Rhythmicity: The perceived quality of rhythm as being consistent, coherent and smooth, contributing to the overall musical experience.

Naturalness: **The perception of musical sounds, melodies, rhythms, or phrases as being in line with the listener's expectation based on their prior musical experiences and cultural background.** For example, certain chord progressions, melody lines, or rhythmic patterns are considered natural or intuitive because they align with the listener's musical expectations.

Tempo: **Overall pace or speed of the song.** It is typically measured in beats per minute (bpm). It helps create a sense of mood, energy and atmosphere in a piece of music being used to build up tension, let it out, or reflect the feelings expressed in a piece of music. It is closely related to rhythm and meter.

Form: **The structure of a piece of music**, being the way elements of a piece, such as melody, harmony, rhythm, and timbre, are put together to build a coherent whole. It includes elements such as repetition, contrast, and development.

Pitch: **The highness or lowness of a sound tone determined by the frequency of its waveform.** It is typically measured in hertz (Hz).

Timbre, Tone Color: **The quality of a sound that distinguishes it from other sounds with the same pitch and loudness**. It is defined by the harmonic spectrum of a sound, being the difference between same notes played on different musical instruments.

Dynamics: **Loudness or softness of a musical sound.** It adds emotional intensity to a musical piece.

Texture: **The way harmonies, melodies, rhythms, and timbres relate to create the overall effect of a piece of music** (Feezell Mark, 2013).

Texture in Music

This chapter is based on (J.-P. Briot et al., 2020; Gotham et al., 2021).

In music, the number of "voices" refers to the number of textures or individual melody lines that are played or sung at the same time. In music theory, there are four main categories that vary in terms of the number of music voices: monophony, heterophony, homophony, and polyphony.

Monophony, Melody: A single unaccompanied melody. It is the simplest and most exposed of all musical textures. Simple piano pieces, such as a single line melody, would typically be considered monophonic. The first movement of Cello Suite no. 1 in G Major (1717) by Johann Sebastian Bach is an example of a monophonic texture.

Heterophony: Multiple voices that play variations of the same melody simultaneously. These variations can range from small embellishing tones to longer runs in a single voice, as long as the melodic material stays relatively constant, and can be subtle or significant, but the overall effect is a rich and complex texture. While the multiple instruments play different embellishments, they present essentially the same melodic material.

Homophony: Multiple voices moving together harmonically at the same pace. It is a common texture where a dominant melody line is accompanied by supporting harmonies. In homophonic piano pieces, the harmony supports and reinforces the main melody. Homophony is sometimes further divided into two subcategories:

- **Homorhythm: All voices move in an extremely similar or completely unison rhythm**. This is most often seen in chorale-like compositions, where the melody and harmonies move together in block chords or when a melody stands out from the texture, but the other voices still moving in rhythmic unison. An example would be Six Horn Quartets: no. 6, Chorale (1910), written by Nikolai Tcherepnin.
- **Melody & Accompaniment**: **A distinct melody and supporting voices, which are called an accompaniment.** Often the melody will have a different rhythm from the supporting voice(s). This is the most common form of homophony.

Polyphony: Multiple voices with independent melodic lines and rhythms that blend together harmonically. This type of music features multiple voices or melody lines that are played at the same time. Polyphonic piano pieces often have a more complex and layered structure, with each voice contributing to the overall harmony. In Western classical music, polyphony is commonly heard in fugues, such as Fugue no. 5 in D Major (1951–1952), written by Dmitri Shostakovich.

When it comes to piano music the number of voices in a piece of piano music can greatly impact its overall sound and structure. Understanding the different types of textures in music and piano music specifically is important to identify the interactions between the different voices. **In our work, we aimed to integrate the full potential of polyphonic music by incorporating multiple voices into our compositions**. This would allow us to generate and complement rich and intricate compositions that would be impossible with just a single voice.

Emotion in Music

Our understanding of the role of emotion in music in general and piano music was greatly influenced and inspired by the works of (Juslin, 1993; Juslin & Västfjäll, 2008; Koelsch, 2010, 2014; Levitin, 2006; Sloboda, 1985) who have written extensively on these subjects.

A common definition **of emotion in music is the feelings, sentiments or moods that a piece of music evokes in the listener, referred to as an emotive or perceptual quality of music**. Despite not being regarded as a structural component of music like melody and harmony, it is still thought to be a crucial element of music that can significantly affect the emotional and psychological wellbeing of listeners. There are various ways to categorize musical emotions, including fundamental emotions (happiness, sadness, fear, surprise etc.) and more complex ones (nostalgia, longing, etc.).

One of the most significant ways in which music evokes emotion is through its use of melody, harmony, and rhythm. These musical components combine to produce tension and release, which can elicit emotions such as happiness, sadness, or excitement. For instance, a melody that moves up in pitch can convey optimism, whereas a melody that moves down in pitch can convey melancholy. Similar to how an unresolved chord progression can keep you feeling anxious, a resolved chord progression can cause you to feel relieved. Another example combining these structural elements would be a fast tempo and a major key conveying a sense of happiness, while a slow tempo and a minor key conveying a sense of sadness.

Emotion plays a crucial role in piano music too, as the piano's wide range of dynamics and expressiveness allows for a powerful evocation of emotions in the listener. The piano can produce a wide variety of sounds, from gentle and delicate to strong and thunderous, enabling composers and performers to write music in a wide range of genres or styles that are rich in emotion and embrace a variety of moods and sentiments. The piano also enables musicians to express their feelings through their playing, using methods like vibrato and rubato to further stimulate the listener's emotions and make the piano a powerful instrument for emotional expression.

As mentioned, emotion in music is a complex and multi-faceted aspect that is conveyed through a variety of musical elements, including melody, harmony, rhythm, timbre, and dynamics. However, it's worth emphasizing that other subjective aspects, such as cultural background, individual experiences, and the musical environment in which the music is heard, also have a role in shaping the emotional impact of music.

Chapter 2: Deep Learning Music Generation

In this chapter, we will delve into the fundamentals of deep learning for music generation. We will explore the core concepts that form the foundation of deep learning practices used to generate music. From recurrent neural networks to different evaluation techniques, we will examine the key elements and parameters that enable deep learning algorithms to produce and complement music that resembles human-created music. By providing an in-depth look at these basic elements, we aim to provide a solid understanding of the underlying principles and mechanisms behind deep learning-based music generation.

Deep Learning Models

In this sub-chapter, which is based in (Hernandez-Olivan, Hernandez-Olivan, et al., 2022; Hernandez-Olivan & Beltrán, 2023; Ji et al., 2020), **we will focus on some of the most recent and best-performing deep learning models for piano polyphonic music generation** we have come across. While our coverage will not be exhaustive, we will highlight some of the key models and architectures that are currently being used in this field.

Music composition has been attempted through deep learning by using various architectures, including Generative Models like VAEs and GANs, Recurrent Neural Networks (RNNs) like LSTMs, Neural Autoregressive Distribution Estimators (NADEs), as well as models commonly used in Natural Language Processing, such as Transformers.

Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are a type of artificial neural network designed for handling sequence or time series data. They are distinctive for their memory cells, which enable them to store information and model long-term sequences. Two popular RNN models are Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs). **In the field of music generation, these were the first types of networks used to generate music with long-term dependencies.**

As presented by (Hernandez-Olivan, Hernandez-Olivan, et al., 2022; Hernandez-Olivan & Beltrán, 2023), a breakthrough in the field of music generation using deep learning began with the creation of models that generated short-term note sequences using RNNs and semantic models based on unit selection. **RNNs become the most commonly used models in the field as they are among the simplest models for learning sequence data. However, some first models worked only for short sequences** and as the interest in creating longer sequences grew, other models that combined RNNs with probabilistic methods emerged, such as Google's Magenta MelodyRNN (J. Wu et al., 2017), Anticipation-RNN (Hadjeres & Nielsen, 2017) models and DeepBach (Hadjeres et al., 2017). MelodyRNN model enhanced RNN's capacity to learn long-term structures, while Anticipation-RNN, a novel model, enabled the application of user-defined positional restrictions. Both proved capable of generating melodies and chords that followed melody lines. On the other hand, DeepBach composed of an RNN and using Gibbs sampling, was one of the first models to achieve

counterpoint, combining melodies by following certain rules, and was considered among the state-of-the-art for music generation due to its ability to generate 4-voice chorales in the style of Bach.

Another model introduced, as discussed in (Ji et al., 2020), was TonicNet (Peracha, 2019), a GRU-based model that can predict the chords and notes of each voice at a given time step. TonicNet can be conditioned on salient features extracted from the training dataset or trained to predict these features as additional components of the modeled sequence. The model employs ancestor sampling to generate musical scores and predicts continuous markers in a purely autoregressive manner. Unlike other models, TonicNet does not require any preset information about length or phrase, which has led to state-of-the-art results on the JSB dataset currently.

Generative Networks (GANs, VAEs)

Generative models such as VAEs and GANs have been gradually applied to polyphonic music generation, trying to address the problem of generating new music ideas with a high level of creativity from scratch. Regarding GANs as (Ji et al., 2020) mention, although they are considered very powerful generative models, they can be challenging to train and typically aren't used for sequential data. However, recent research by (L. C. Yang et al., 2017) and (H. W. Dong et al., 2017) has shown that CNN-based GANs can be used for music generation. (L. C. Yang et al., 2017) developed a GAN-based MidiNet that generates melody one bar at a time and incorporates a new conditional mechanism based on chords. (H. W. Dong et al., 2017) created the MuseGAN model, which is the first model capable of generating multi-track polyphonic coherent music. More GAN-based models followed, including CNN-GAN (H. W. Dong & Yang, 2018) with binary neurons.

Variational Autoencoder (VAE) was firstly applied to game song generation. **MusicVAE** (Roberts et al., 2018)**, a VAE-based model, was proposed by Magenta in 2018 to interpolate in the VAE latent space to compose short music fragments from 2 to 16 bars**. The model was trained with approximately 1.5 million songs from the Lakh MIDI Dataset (LMD) and can generate polyphonic melodies for 3 instruments: melody, bass, and drums. **It remains one of the best-performing models to generate motifs or short melodies** from 2 to 16 bars. Later, Researchers have also proposed models like MahlerNet (Lousseief & Sturm, 2019) and Music FaderNets (Tan & Herremans, 2020), which can manipulate low-level attributes and infer high-level features through semi-supervised clustering. However, these models still have limitations in generating new and creative music ideas. After the creation of MusicVAE model along with the birth of new neural network architectures in other fields, the necessity and availability of new deep learning models that could create longer melodies grew.

Transformers

Recently, **Transformer models have shown great potential in polyphonic music generation as the music generation field aimed to generate longer sequences or melodies guided by the excellent capabilities of the new Transformer-based models in text generation.** (C.-Z. A. Huang et al., 2018) introduced the Music Transformer, a model combining the relative attention mechanism into the Transformer architecture. This model is capable of generating music structures with ~2,000 tokens in scale, building continuations that flow logically from a given motif, and creating

accompaniments in a seq2seq framework. It marks the first successful application of the Transformer architecture in generating music with long-term structure and one of the earliest models to utilize attention mechanism for polyphonic music generation. Music Transformer was initially trained on the JSB Chorales and the maestro dataset for virtuosic piano. Given these important capabilities, it is among the reasons why we selected this model for further training and music generation for our project.

(Donahue et al., 2019) introduced the idea of utilizing the Transformer model to generate music with multiple instruments, and proposed a transfer learning-based pre-training approach. They used this technique to enhance the performance of their Transformer-based music generation model by involving pre-training the model on one dataset (Lakh MIDI) and then fine-tuning it on another dataset (NES-MDB) to incorporate information from the latter dataset. New models that used pre-trained Transformers came up including **MuseNet proposed by OpenAI** (Payne Christine, 2019). This model, based on GPT-2, has the capability to produce 4-minute music compositions that incorporate a diverse range of styles and feature 10 different instruments. **One of the factors that contribute to its ability to remember the medium and long-term structure of music segments is its full attention mechanism within the context of 4,096 tokens.**

In order to enhance the musicality of the generated musical pieces and improve the attention learned, (N. Zhang, 2020) introduced a new Transformer model, the Adversarial Transformer, that utilizes adversarial learning to generate pieces of music with higher musicality. The combination of self-attention architecture and generative adversarial learning results in a generation of lengthy sequences that are guided by adversarial objectives, thereby providing robust regularization and helping the Transformer to focus on both global and local structure learning.

As (Ji et al., 2020) mentions, (Y.-S. Huang & Yang, 2020) proposed the Pop Music Transformer, based on Transformer-XL (Dai et al., 2019), for improving the transformation of music scores into events. The new event set, referred to as REMI (REvamped MIDI-derived events), was a novel music representation offering a metrical context to model music rhythm patterns, and thus making the model more aware of the beat, bar, and phrase hierarchy in music. It also provides a harmonic structure for controlling chord progression and coordinating different tracks in a music piece, such as piano, bass, and drum. The results implied that the music generated by the Pop Music Transformer had a more reasonable rhythm structure compared to the Music Transformer

Meanwhile in 2020, (S.-L. Wu & Yang, 2020) made an innovative attempt to generate jazz music through the development of the Jazz Transformer, which is also based on the Transformer-XL model. This model is among the latest in the family of attention-based models. **Also, one of the latest and noteworthy models to mention is the Perceiver AR** by (Hawthorne et al., 2022a)**. This model, specifically, utilizes cross-attention and has a context of 4,096 tokens, having been trained using the extensive maestro dataset.** It seems to have the capability to learn and produce longer musical sequences, being also among the reasons why we selected it for further training and music generation for our project.

Model Combinations

Combining VAEs and Transformers or other Neural Networks has given birth to new models that aim **to overcome the problem of losing sense of the music in the previous bars or the main motifs, faced in in Transformer-based models, when generating longer sequences.** Models like the TransformerVAE (Jiang et al., 2020) and PianoTree (Wang et al., 2020) are examples of such models that perform well in polyphonic music and can generate music phrases. Regarding PianoTree VAE it was designed to learn polyphonic music, being the first effort to generate polyphonic counterpoint in the context of music representation learning. The model uses a tree structure to reflect the hierarchical nature of music concepts. Despite these attempts, no model has yet achieved the goal of generating a long music piece with high level structure coherence following the sense of a certain direction.

Music Representation

Music representation format is a crucial aspect of deep learning in piano music generation. The way in which musical content is represented greatly affects the whole process of training the models and generating music. As discussed by (Ji et al., 2020), the representation of music can be categorized into two types:

- **Symbolic Representation**: **Consists of discrete variables capturing important aspects of music,** such as timing, pitches and dynamics. Symbol representation is often customized for specific instruments, which limits its versatility and requires extensive effort to adapt to new instruments.
- **Audio Representation**: **Consists of continuous variables retaining all music-related information** and captures rich acoustic details, such as timbre and articulation, making it more universal and applicable to any musical or non-musical audio signals. The symbol abstraction and precise performance control information hidden in the audio digital signal become less evident in this representation.

As (J.-P. Briot et al., 2020) argue, the representation of musical content is connected to the configuration of the input and output of the deep learning architecture. **The choice of representation and its encoding affects the learning process and the quality of the generated content.** The raw material for audio and symbolic representations is significantly different and the techniques used for processing and transforming the initial representation vary as well. These representations correspond to distinct scientific and technical fields, such as signal processing and knowledge representation. However, despite these differences, the deep learning architecture for processing these two types of representations is largely the same. As a result, the architecture for generating audio and symbolic music may be quite similar.

This project will concentrate on symbolic representations and deep learning methods for generating symbolic music. This choice is motivated by several reasons:

- **Most deep learning systems for music generation are based on symbolic representations.**
- **Symbolic datasets are more widely available and can be processed in a more efficient and computationally feasible way.**

- **The core of human music composition process is revealed through symbolic representations.**
- **Analyzing, processing and transformation of audio representations would require procedures out of the scale of this project.**
- **Regardless of whether audio or symbolic music is being generated, the principles of deep learning architecture and encoding techniques are largely the same** (J.-P. Briot et al., 2020)**.**

MIDI Format

This section is based on (J.-P. Briot et al., 2020; Hadjeres et al., 2017; C.-Z. A. Huang et al., 2019; Ji et al., 2020).

The MIDI (Musical Instrument Digital Interface) format is the most common symbolic representation for music generation to encode musical information, being the one we implemented too. Each musical piece consists of a series of event messages (or tokens) that describe the pitch, timing and velocity (how hard the key was struck) of individual notes played on an instrument.

The use of MIDI has several additional advantages over other forms of musical representation, such as audio or sheet music. Firstly, **MIDI data provides a level of abstraction that allows for more straightforward manipulation and transformation of musical elements, such as transposition or rhythm modification.** Furthermore, MIDI focuses on the underlying structure of the music, rather than the specific instrument that is being used to play the notes. This allows models to generate music that can be played on any instrument.

Despite the many advantages of MIDI as a representation for piano music generation, there are also some challenges that need to be addressed. For example, while MIDI encodes note timing and duration, it does not encode other music qualities such as timbre. Also, the choice of encoding techniques used to represent the MIDI data affects the quality of the generated music.

Model Training

Transformer-based models have been applied successfully in music generation. The performance of these models is highly dependent on their hyperparameters, which need to be carefully tuned for optimal results. Some of the key hyperparameters in Transformer music generation models include the following:

Learning Rate: The rate at which the model updates its weights during training. It controls the step size taken in the direction of the gradient, and if it is too high, the model may fail to converge, and if it is too low, the model may converge very slowly or get stuck in a local minimum.

Number of Epochs: The number of times the model will go through the entire training set. A smaller number of epochs can lead to underfitting, while a larger number of epochs can lead to overfitting.

Batch Size: The number of samples that the model processes in each training iteration. A larger batch size can lead to more efficient training, but may also require more memory, and smaller batch sizes may lead to a slower convergence.

Number of Layers: Number of layers in the model architecture. More layers can lead to more expressive power and better performance, but also increase the risk of overfitting and require more memory and computations.

Input, Sequence Length: The length of the input sequence that the model processes in each training iteration. Longer sequences may capture more context and result in more coherent generated music, but may also require more memory and increase the training time. It is typically set between 1,024 and 4,096.

Vocabulary Size: The number of unique tokens used to represent the input data. A larger vocabulary size can potentially allow the model to capture an extended range of musical complexity but can also increase the computational complexity and training time.

Embedding Dimension: The size of the embedding vectors that represent the input data. It might correspond to the number of features used to represent each musical event or note, with a higher value being able to capture more complex and subtle relationships between musical events but requiring more memory and computational resources.

Number of Heads: The number of attention mechanisms used in the self-attention layer. More attention heads can help the model capture more fine-grained relationships in the input, but also require more memory and computation.

Head Dimension: The dimensionality of the projection used to create the query, key, and value vectors in each attention head of the transformer.

Dropout Rate: The probability that a neuron will be randomly dropped out during training. Dropout can help prevent overfitting and improve generalization, but too high of a dropout rate can result in underfitting.

Loss Function: The objective function used to optimize the model during training. Common choices include cross-entropy, mean squared error, and binary cross-entropy.

Warmup Steps: The number of initial training steps during which the learning rate is increased gradually to its maximum value. Warmup steps can help the model avoid getting stuck in a suboptimal local minimum and converge faster.

Fine-tuning the hyperparameters of music generation models is often requires a significant amount of trial and error and experimentation, since they are specifically dependent to the dataset and model. Besides the general hyperparameters, some music generation models may have domain-specific hyperparameters, such as the structure of the music, that are tailored to the task.

Adam Optimizer

As (A. Zhang et al., 2021) discuss, Adam optimizer, introduced by (Kingma & Ba, 2014), is an optimization algorithm formed by a combination of several optimization techniques such as vectorization, combining previous gradients, per-coordinate scaling, and separating learning rate adjustment from scaling per coordinate. Adam optimizer can be combined with a learning rate scheduler to adjust the learning rate during training and improve the model's convergence. While

Adam has its default learning rate schedule, it may not always be optimal for a particular problem or model architecture. Therefore, a custom learning rate scheduler can be used to adjust the learning rate based on the specific needs of the problem. Another technique that can be used with Adam to help the model converge more quickly is warm-up steps. Warm-up steps gradually increase the learning rate from an initial value to a maximum value over a certain number of iterations.

Evaluation

Evaluation is an essential aspect of deep learning music generation. Objective and subjective metrics are employed to measure the quality and musicality of generated music. Objective metrics provide a quantitative measure of the performance of the model, while subjective metrics provide a more qualitative measure of the music's perceived quality. The choice of metric depends on the implemented models, the objectives of the work, as well as the musical domain in which the music generation model is operating. Objective evaluation metrics can be categorized according to whether they contain musical knowledge, as described by (Ji et al., 2020; L.-C. Yang & Lerch, 2020). The first category involves probabilistic measures without any musical domain knowledge like loss and the second category includes metrics that rely on music theory.

One of the most used metrics in machine learning as an objective function is loss. However, as explained in (Ji et al., 2020), while loss is an important metrics in evaluating the performance of machine learning models during training, it only reflects the ability of the model to process data and not the quality of the generated music. For this reason, additional selection of other musically meaningful metrics is crucial in evaluating the quality of generated music after completion of models training.

In summary, the use of a combination of objective and subjective metrics plays a crucial role in assessing the quality of music generated by deep learning models. It provides a way to quantify the quality of the generated music, assess its similarity to real music, and evaluate the model's ability to generate music that has several musical characteristics. It is a process that helps gain a comprehensive understanding of the strengths and weaknesses of music generation models and make informed decisions about how to improve the models and the generated music.

Chapter 3: Music Transformer Model

This chapter is mainly based on (C.-Z. A. Huang et al., 2018) original paper and (Tan, 2020) overview summary.

Music Transformer is the first successful application of Transformer models, originally developed in the NLP field, to the realm of symbolic music generation in order to create music with long-term structure. In music, relative timing is a critical factor, and the Music Transformer addresses this issue by incorporating a relative attention mechanism. It is able to generate minute-long music compositions of up to ~2,000 tokens, coherently elaborate on a given motif, and generate accompaniments in a seq2seq setting based on input melodies.

Model Architecture

It is built using decoder-only transformers, utilized to predict the next token at each time step using a relative attention mechanism and optimized with a cross-entropy loss function as the objective function.

Relative Attention

The idea of relative attention is brought forward by (Shaw et al., 2018). The objective is to describe relative position representations more efficiently, to allow attention to be informed by how far two positions are apart in a sequence. This is crucial in music because learning relative position representations enables to remember structure elements like scales and arpeggio patterns, repeated motifs, and call-and-response patterns.

Relative Attention Equation: $Attention(Q, K, V) = softmax(\frac{QK^T + S^{rel}}{\sqrt{d}})V$

Model Operation

Using a language model approach to generate symbolic music, Music Transformer treats each performance event as a token, a practice first proposed by (Oore et al., 2020), similar to a word token in a sentence. This allows the model to **learn the relationships between performance events through self-attention and generate music in an autoregressive manner,** like language models. The encoding used in the Music Transformer is specifically designed for piano pieces, but it can be adapted to encode multi-track music or leverage transfer learning with multi-track datasets.

More specifically, the model predicts the next musical event (token) iteratively based on the current input sequence. Firstly, it determines the importance of each input token via the attention scores for the input sequence. Next, the weighted sum of the embeddings of the input sequence is generated and passes through the layers of the model. Finally, a probability distribution for the next musical event is generated. In the training mode of operation, the difference between the predicted event distribution and actual next event is tried to be minimized, while in the generation mode the predicted next event is inserted into the musical piece event sequence.

Results

(C.-Z. A. Huang et al., 2018) demonstrate that using relative attention in the Music Transformer architecture leads to better NLL loss compared to other architectures like vanilla Transformers. Furthermore, incorporating timing and relational information results in improved results, with the generated music being more coherent and having a longer temporal structure.

Limitations & Improvements

(Hernandez-Olivan & Beltrán, 2023) claim that, although Music Transformer has the ability to produce longer and continuous melodies, **as the melody progresses, it tends to become increasingly random and deviate from the original musical sense of the piece**. They imply that this is justified by the lack of proper structure and organization in the composition process. As they explain, **although the harmony produced in the early stages of the melody is coherent because it follows a particular key, the lack of aesthetically pleasing transitions and connections between sections as the piece develops is a noticeable limitation.**

Music Transformer is trained to predict the next performance event at each time step using cross-entropy loss as the objective function. (Ji et al., 2020) argue that **this loss function does not necessarily reflect the quality of the generated music and serves only as a tool for the training process**. They express the view that using the teacher forcing training strategy may produce some good pieces, but **the generated music is often lacking in musicality, and the attention learned by the model is often messed and poorly structured.**

Chapter 4: Perceiver-AR Model

This chapter is mainly based on (Hawthorne et al., 2022a) original paper, (Hawthorne et al., 2022b) presentation overview and (Tiernan, 2022) overview summary.

Perceiver AR is a type of neural network architecture used for autoregressive generation of music and other types of data. It was originally developed by DeepMind and is based on the Perceiver (Jaegle et al., 2021) architecture, which is a type of transformer network that processes sequences of data of arbitrary length, **directly attending to over a hundred thousand tokens. This key advantage enables practical long-context density estimation without the need for hand-crafted sparsity patterns or memory mechanisms. It can be significantly effective for music generation as it could capture longer-term patterns and structures in music data.**

Model Architecture

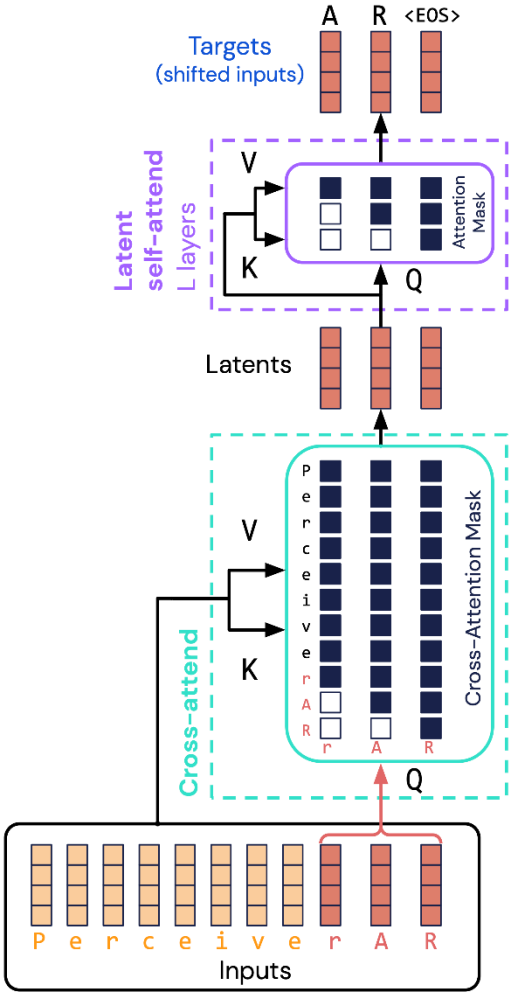

Figure 4.1: Perceiver-AR Model Functionality (Hawthorne et al., 2022a)

The use of autoregressive Transformers led to significant breakthroughs in generative modeling. These models generate samples one element at a time and have been effective in generating images, text, and audio. However, each layer of a Transformer becomes more expensive as more elements are used as input, limiting their effectiveness to sequences of no more than 2,048 elements. Perceiver models have been developed as an alternative and have shown excellent results on various real-world tasks with up to 100,000 elements. The Perceiver architecture is a flexible approach that can handle various types of input, including text, sound, and images.

Perceiver AR **can handle long input sequences of up to 50 times longer than standard Transformers, while deploying as widely and essentially as easily as standard decoder-only transformers.** This is achieved by passing both inputs and targets through a single, shared processing stack, allowing each target to learn how to use long-context and recent input as needed with minimal architectural restrictions. From this point of view, Perceiver AR can be viewed as an encoder-decoder architecture with no encoder layers.

It has an autoregressive aspect, as it uses its outputs as new inputs recursively, forming an attention map of how multiple elements relate to one another. Furthermore, as a Perceiver model it uses cross-attention to encode inputs into a latent space, separating the input's compute requirements from

model depth. However, its proposed architecture also aligns the latents one by one with the final elements of the input and masks the input carefully so that latents only see earlier elements. This ensures that processing happens one element at a time, aligning latent-space encoding and autoregressive generation.

The following Perceiver AR algorithm deals with the issue of large inputs by mapping inputs $X \in R^{MxC}$ (C is the number of channels) to a smaller number of latents $Z_1 \in R^{MxC}$:

- $Z_1 \leftarrow CrossAttend(X, Z_0)$

It is possible for more self-attention modules to process the latent array Z1 because it is frequently short (N < M):

- $Z_{l+1} \leftarrow SelfAttend(Z_l, Z_l)$

The operation described does not rely on the number of input points, and by choosing an appropriate value for N, it can be repeated for multiple layers without incurring excessive computational costs. This results in an architecture with a complexity of O(MN) + $O(LN^2)$ due to the cross-attention and the latent self-attention stack.

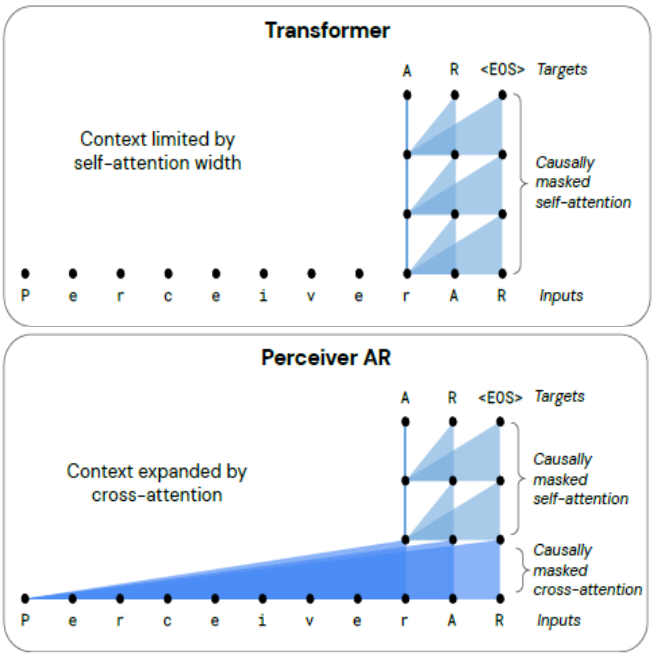

Figure 4.2: Perceiver-AR - Decoder-only Transformer Comparison (Hawthorne et al., 2022a)

However, reducing the number of input points from M to N means that the causal dependency between all input and output points used in Transformers for autoregressive modeling cannot be established. To address this issue, Perceiver AR introduces causal masking to both the input cross-attention and the latent self-attention layers and assigns one latent to each of the final N input points with the largest number of antecedents. **The influence of inputs that come after a given latent is masked at the cross-attention and all self-attention layers, preserving the autoregressive ordering of the targets throughout the network.** This technique can be applied to any ordered input as long as masking is used.

Results

Perceiver AR was originally trained on music data and has been shown to generate outputs with clear long-term coherence and structure according to (Hawthorne et al., 2022a). The samples obtained from the large transcription dataset exhibit stylistic and structural coherence that spans several minutes, containing repeating musical themes, chord patterns, arpeggios, and even ritardandos.

These results for symbolic music generation were highly promising, as the model could effectively handle long sequences of data and was capable of generating stylistically and structurally coherent music that exhibits creativity and novelty. It outperforms existing models such as Music

Transformer, achieving a lower negative log-likelihood even without the benefit of data augmentation.

Limitations & Improvements

As (Hernandez-Olivan, Hernandez-Olivan, et al., 2022) discuss, generating music that follows a high-level structure has been a long-standing challenge in the field of music generation. The difficulty lies in the fact that models need to comprehend a high level of musical understanding not only learning how rhythm, melody, and harmony are combined, but also being able to recall music events that happened many bars or minutes before. Even though the authors recognize that **Perceiver AR could be a possible solution to generate structured music, its capability to generate structured music have not been evaluated in terms of the music principles.**

Conclusion

In conclusion, the Perceiver-AR architecture is a promising architecture for generating music and other applications that require processing long sequences of data. The original Perceiver architecture's innovation of consuming all kinds of input, including text, sound, and images in a flexible form, paved the way for better generative modeling techniques. Perceiver-AR's proposed solution addresses the limitations of autoregressive Transformers, offering a more flexible and efficient approach to processing long sequences of data. Despite being a relatively new architecture, ongoing research efforts are dedicated to exploring its capabilities and improving its performance.

Part II: Methods & Results

Chapter 5: Training Datasets & Music Representation

Training Datasets

As discussed by (J.-P. Briot et al., 2020) the selection of a dataset is crucial for producing high-quality music outputs as it directly affects the quality and diversity of the generated music, making it essential to choose a dataset with a sufficient number of samples and size. In case of a low-quality or inadequately diversified datasets, the music generated by even the most advanced models may not accurately represent the genre or type of music it was trained on.

The relationship between the size and diversity of a dataset and its coherence is a significant challenge in creating deep generative models, as (Hadjeres, 2018) argues. A diverse dataset can challenge the model to differentiate between subgenres or musical styles and generate diverse outputs. On the other hand, even if a dataset samples have minimal differences, the models are required to generate musically appealing results. The size of the dataset needed for a model to perform well depends on the complexity and diversity of the music to be generated, but typically a dataset with thousands of examples is necessary.

To address these challenges, we prioritized specific characteristics of the openly available datasets in the process of choosing our training datasets. **Music genre was one of the most important factors we considered, ensuring a variety of genres to make relative comparisons between our generation outputs in respect to this factor. We also considered the scale of the dataset, intending to collect larger datasets to enhance the generalization capabilities of our models.**

In the following table, we provide an overview of the datasets we used in the training process of our models, including information about the above characteristics we considered. The factor of dataset size is represented in different perspectives by the number of songs and size parameter.

Dataset	Music Genre	Songs	Size
maestro-3.0.0	classical piano	1,276	85 MB
GiantMIDI-Piano	classical piano	10,855	281 MB
ailabs1k7	pop piano	1,748	25 MB
adl-piano-midi	misc	11,086	187 MB
Rock-Piano-MIDI	rock piano	8,761	136 MB
Los-Angeles-MIDI-segment	misc	46,997	1.3 GB

Table 1: Training Datasets

maestro-3.0.0

Maestro-3.0.0 (Hawthorne et al., 2018) dataset includes about 200 hours of virtuosic piano performances captured with fine alignment between note labels and audio waveforms. Several datasets of paired piano audio and MIDI had previously been published before maestro, allowing significant advancements in automatic piano transcription and related topics. However, from our

perspective **maestro has several advantages over existing classical piano datasets including its sufficient scale, quality and musical information included**. As a result, it has been one of the most used datasets in music information retrieval and generation tasks being referenced by hundreds of papers since its publication.

GiantMIDI-Piano

GiantMIDI-Piano (Kong et al., 2020) is a large-scale dataset of 10,855 solo classical piano pieces composed by 2,786 composers. The solo piano recordings were transcribed into MIDI files using a high-resolution piano transcription technology. GiantMIDI-Piano contains 90% live performance MIDI files and 10% sequence input MIDI files. The difference between other classical piano datasets including maestro and **its competitive advantage is its scale being one of the largest symbolic classical piano datasets so far.** This can also be confirmed by its measured size being the second largest out of the six datasets we have chosen.

ailabs1k7

Ailabs1k7 (Hsiao et al., 2021) includes audio files of 1,748 pop piano pieces from the Internet. The average song length is about 4 minutes. All the songs are in 4/4 time signature and were converted into a symbolic sequence, following a transcription, synchronization, quantization and analysis process. We used the midi files produced from the analysis step of the symbolic conversion process (midi_analyzed dataset folder). From our perspective **it is probably the biggest widely available pop piano solo dataset including sufficient symbolic information and also having already been used by the creators for music generation purposes with transformer architectures**. As mentioned, we wanted to have a variety of music genres on our training datasets, and also see the effect of training on a dataset relatively smaller both in total songs duration and in size on the training process and outputs produced.

ADL Piano MIDI

ADL Piano MIDI (Ferreira et al., 2020) is a collection of 11,086 piano pieces from various genres. This dataset is based on the Lakh MIDI dataset with only the tracks with instruments from the piano family being extracted. We considered ADL Piano MIDI Dataset a suitable dataset for our task, as first of all **is based on various genres being an important difference with most of the other datasets. Also, it is based on one of the biggest music datasets available, the Million Song Dataset, as a result having the required scale and size for our task.**

Rock Piano MIDI

As presented by (Lev, 2022)**, Rock Piano MIDI is a piano-drums midi dataset collected for music information retrieval and music generation purposes**. The author has made available many code implementations of music generation models and datasets used for his projects with the particular collection **being one of the biggest openly available for the rock genre having sufficient scale in duration and size for the purpose of our task.** We have extracted the drums part of the symbolic representation in the preprocessing phase of our training process as it is not used in our task.

Los Angeles MIDI segment

As also presented by (Lev, 2023), the original Los Angeles MIDI Dataset includes 600,000 completely unique MIDI files featuring a variety of music genres. However, with its the scale being extremely high for our computational resources and also incomparable to the scale of the other datasets, we chose to randomly take a smaller segment from the dataset approximately 1/20 of the original featuring 46,997 songs, 1.3 GBs size. **This dataset is referred as Los Angeles MIDI segment in this work being a relatively bigger dataset than the others**, however computationally feasible for our resources and comparable to the others. Using this dataset, we have the chance to conclude the effect of another multiple genre dataset and also to compare the effect of its larger size to the other misc. dataset ADL Piano MIDI.

Other Datasets Considered

Dataset	Music Genre	Songs
Los Angeles MIDI Dataset	misc	939,948
EMOPIA Dataset	pop piano	387 (1,078 clips)
Lakh MIDI Dataset	misc	174,533
JSBach Chorale Dataset	classical piano	382
classic-piano	classical piano	329

Table 2: Not Selected Datasets

In the above table we present some of the other datasets we come across during the process of our research which we also considered for the training of our models. They were not chosen for a number of reasons including their genre, for example the last two ones are classical piano having already enough bigger scale classical piano datasets, their size, for example EMOPIA dataset was not big enough for our purpose and at last for not being solo piano oriented like Lakh MIDI Dataset requiring additional preprocessing procedures to extract the piano tracks from the songs.

Music Representation & Preprocessing

Music Transformer

In the Music Transformer implementation, MIDI musical information is represented in the form of a sequence of discrete events based on the preprocessing algorithm suggested on (C.-Z. A. Huang et al., 2018), where each event is represented by an integer and is categorized into one of four types: note on, note off, time shift, and velocity. So, MIDI note events are converted into a sequence from the following set of vocabulary that are referred as Type of Event (Total Type's Events Number):

- **Note On (128)**: Integer value that corresponds to **the pitch of the note being played.** This value is within the range of MIDI notes numbers.
- **Note Off (128):** Integer value indicating **the end of a note's duration.**
- **Time Shift (100)**: Integer value that indicates **the amount of time elapsed since the previous event**. The value of the first-time shift event in the sequence represents the amount of time elapsed since the start of the sequence.

- **Velocity (32):** Integer value that corresponds to **the velocity or loudness of the note being played**. It affects the volume of the sound produced.

Perceiver-AR

In the Perceiver-AR implementation, the MIDI musical information representation, proposed by (Lev, 2022a), is in the form of a sequence of discrete events, with the type of events and the range of values in this case being slightly different. The encoding is performed by mapping each MIDI event to a unique integer. The set of vocabulary referred as Type of Event (Total Type's Events Number) is as follows:

- **MIDI Pitch (127):** Integer value that corresponds to the **pitch of the note being played**, where 1 represents the lowest possible pitch and 127 represents the highest possible pitch. The actual pitch value is calculated by subtracting 256 from the encoded value.
- **Duration (127):** Integer value that corresponds to the **duration of the current event/note**, where 1 represents the shortest possible duration and 127 represents the longest possible duration. The actual duration value is calculated by subtracting 128 from the encoded value.
- **Time Shift (127)**: Integer value that indicates **the amount of time elapsed since the previous event**, where 0 represents no time shift and 126 represents the maximum time shift value.
- **MIDI Velocity (127)**: Integer value that corresponds to the **velocity or loudness of the note being played,** where 1 represents the softest possible velocity and 127 represents the loudest possible velocity. The actual velocity value is calculated by subtracting 384 from the encoded value.
- **Compositions Separator/Intro/Zero Sequence (4):** Specific sequence of values that **indicates the start of a new composition or the end of the current composition,** where the values are [127, 255, 256, 383] respectively. This sequence is used as a separator between compositions or as an introduction to the first composition.

Representation Differences

Considering the two representations used we can conclude that **they have significant resemblance at their mapping as their event types have similar functionalities**. However, **one major difference is the range of values of Time Swift and Velocity types of events.** In Music Transformer, velocity events are represented by an integer value in the more limited range of 0-31, while in Perceiver-AR in the range of 1-127. Moreover, time swift events are represented by an integer value in the range of 0-99, while in Perceiver-AR in the range of 1-127.

The differences in the representation of velocity and time shift events between the Music Transformer and Perceiver-AR models could have implications for their expressive abilities. Velocity is an important aspect of music performance affecting the perceived loudness and intensity of a musical passage. In the Music Transformer model, the reduced range of possible velocity values, could limit the model to a relatively narrow range of dynamic levels. Similarly, time shifts are important in music to indicate the timing and rhythm of a piece. The Perceiver-AR's extended range of possible time shift values could allow for a wider range of timing variations and potential for more rich and nuanced pieces.

Implementation

Music Transformer

The midi preprocessing code is included on /music-transformer/preprocess.py file and was based on code from (K. Yang, 2021a) preprocess.py file. Firstly, the function preprocess_midi_files_under is used to preprocess all the MIDI files under the given directory. All the files with .mid or .midi extensions are searched for under that directory and for each MIDI file, the preprocess_midi function is called to encode the MIDI file. Each MIDI file is read using the encode_midi function from the midi_processor module (K. Yang, 2021b) and the encoded data is returned as an array. Error handling statements have been added for KeyboardInterrupt, EOFError, IOError, and KeySignatureError. The encoded data for each MIDI file is then saved in a pickle file in the directory specified in the save_dir parameter.

Perceiver-AR

The midi preprocessing code is based on (Lev, 2022a) and was added to the training notebook /perceiver-ar/Perceiver-Solo-Piano-Maker.ipynb as “Preprocess Training Data” cell. Firstly, the MIDI files are loaded and are added to a list. The script processes the MIDI files by looping through the files. For each file, the note information is extracted using the TMIDIX library (Lev, 2021) and stored in a list called events_matrix. All note events with a drum type are filtered out (if event[3] != 9). Next, the events are sorted by the start time, and the start time is recalculated based on a time resolution of 10ms. Similarly, the note duration is recalculated based on a duration resolution of 20ms. The processed data is stored in a list called train_data1. After processing all the MIDI files of the dataset, the train_data1 list corresponding to the whole dataset is written to a binary file using the TMIDIX Tegridy_Any_Pickle_File_Writer function, and the binary file is saved.

Chapter 6: Models Training

Models Hyperparameters

In this section, we will present the hyperparameters of two models, Music Transformer and Perceiver-AR, examining the differences in the values used in each model. This analysis will allow us to gain a deeper understanding of some distinguishing factors of these models in terms of performance. In the table below, we present most of the hyperparameters of the analyzed models which are critical to the models' performance.

Hyperparameter	Description	Music Transformer	Perceiver-AR
Layers		6	24
Batch Size	Number of training examples in one forward/backward pass	2	1
Input/Sequence Length	Maximum number of tokens processed in each input sequence	2,048	16,384
Vocabulary Size	Total number of unique event tokens	388	512
Embedding Dimension	Size of each embedding vector used to represent musical events	256	1,024
Attention Heads		4	16
Head Dimension	Embedding Dimension / Heads	64	64
Loss Function		Categorical Cross-Entropy	
Optimizer	Optimization algorithm	Adam	
Learning Rate		Custom Scheduler	Initial 2e-5
Warmup steps	Steps before the optimizer fully starts	4000	-
Dropout	Probability of an element to be zeroed in dropout	0.2	0.5

Table 3: Models Hyperparameters

Batch Size

Batch size forms one of the most important hyperparameters during the training of machine learning models. In our task we considered that both our models would require smaller batch sizes between 1-4 for multiple reasons with the most important ones being that smaller batch sizes offer a regularizing effect due to the noise they add to the learning process and also that generalization error is often best for lower batch sizes.

Also, we had to also take into account the amount of memory scaling with the batch size though becoming a limiting factor for us in batch size. In our case, training our models in maximum 12 GB memory graphics cards would enforce a **maximum batch size of 2 for Music Transformer model**

and 1 for Perceiver-AR model. We decided to keep batch size of 2 for Music Transformer as it was also the recommended value by (K. Yang, 2021a) implementation. Also, after experimenting with batch size 1, we would conclude that it would create much more fluctuations during training that would result in compromising models performance.

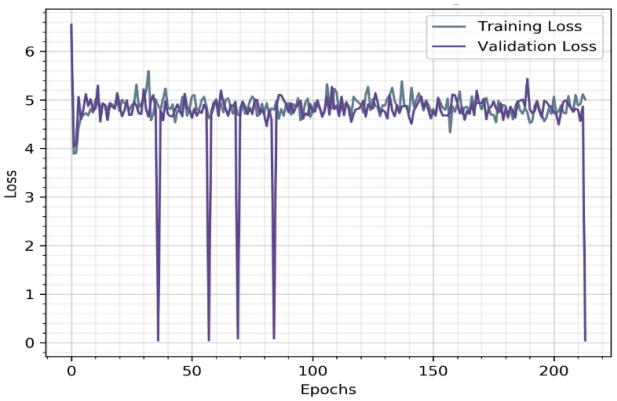

Figure 6.1: Music Transformer maestro-3.0.0 Training-Validation Loss, Batch Size = 1

To further support our claim we present the training-validation loss learning curve for maestro-3.0.0 dataset . Compared to the learning curves for batch size 2 in Learning Curves having all the other training hyperparameters same, we can observe that the batch size 1 model achieves relatively lower performance having unpredictable fluctations in validation loss.

Learning Rate

For Music Transformer implementation we used the recommended hyperparameters by (K. Yang, 2021a) for the initialization of Adam algorithm with:

- **Learning Rate**: Decay using custom learning rate schedule.
- **Beta 1 = 0.9, Beta 2 = 0.98**: Coefficients used for computing running averages of gradient and its square.
- **Epsilon = 1e-9**: Term added to the denominator to improve numerical stability.

We attempted to use the Adam algorithm without decay and with a very low learning rate to ensure that the loss did not start to diverge after decreasing to a certain point. We set the initial learning rate to both 1e-4 and 1e-5 but these models did not have an acceptable performance sticking in relatively high error values for a long time of epochs.

Regarding Perceiver-AR model, (Lev, 2022b) implementation recommended batch size of 4. However, having the restriction to train with batch size 1, as described above, required a small learning rate to maintain stability because of the high variance in the estimate of the gradient. For the above reason we decided to **lower the initial learning rate in Adam to 2e-5 from the initial 2e-4 value of** (Lev, 2022b) **implementation**, **demonstrating convergence in fairly better loss values but also increasing performance fluctuations and training time to an acceptable margin.**

Models Architecture & Parameters

In this section, we present the architecture and parameters of our two models being depicted in model diagrams, providing detailed description of the distinct layers of each model.

Music Transformer

Layer (Type: Depth – Index)	Number of Parameters
Music Transformer Decoder	
└ Embedding: 1-1	3,557,888
└ Positional Encoding: 1-2	0
└ Dropout: 1-3	0
└ Module List: 1-4	-
└ Decoder Layer: 2-1	1,262,976
└ Decoder Layer: 2-2	1,262,976
└ Decoder Layer: 2-3	1,262,976
└ Decoder Layer: 2-4	1,262,976
└ Decoder Layer: 2-5	1,262,976
└ Decoder Layer: 2-6	1,262,976
└ Linear: 1-5	3,557,888
Total Parameters	14,423,616
Trainable Parameters	14,423,616

Table 4: Music Transformer Model Diagram

The description of the distinct layers depicted in the Music Transformer model diagram is presented below:

- **Embedding**: Converts the input sequence of integers into a dense vector representation. The size of this layer is determined by the vocabulary size and the embedding dimension.
- **Positional Encoding**: Adds positional information to the input embeddings, allowing the model to learn the relative positions of the tokens in the input sequence.
- **Dropout**: Randomly drops out some units of the input based on the dropout parameter, reducing overfitting and improving generalization.
- **Module List**: This is a list of Decoder Layers, with each layer containing a multi-head relative attention mechanism and a feed-forward neural network, which are applied to the input in sequence.
- **Linear**: Maps the output of the last Decoder Layer to a vector of the same size as the input embeddings. This output is used to compute the final probabilities for each token in the output sequence.

Perceiver-AR

Layer (Type: Depth – Index)	Number of Parameters
Perceiver-AR: 1-1	
└ Embedding: 2-1	524,288
└ Embedding: 2-2	16,777,216
└ Rotary Embedding: 2-3	-
└ Module List: 2-4	-
└ Module List: 3-1	12,590,080
└ Module List: 2-5	-

└ Module List: 3-2	12,587,008
└ Module List: 3-3	12,587,008
└ Module List: 3-4	12,587,008
└ Module List: 3-5	12,587,008
└ Module List: 3-6	12,587,008
└ Module List: 3-7	12,587,008
└ Module List: 3-8	12,587,008
└ Module List: 3-9	12,587,008
└ Module List: 3-10	12,587,008
└ Module List: 3-11	12,587,008
└ Module List: 3-12	12,587,008
└ Module List: 3-13	12,587,008
└ Module List: 3-14	12,587,008
└ Module List: 3-15	12,587,008
└ Module List: 3-16	12,587,008
└ Module List: 3-17	12,587,008
└ Module List: 3-18	12,587,008
└ Module List: 3-19	12,587,008
└ Module List: 3-20	12,587,008
└ Module List: 3-21	12,587,008
└ Module List: 3-22	12,587,008
└ Module List: 3-23	12,587,008
└ Module List: 3-24	12,587,008
└ Module List: 3-25	12,587,008
└ Linear: 2-6	524,288
Total Parameters	332,504,064
Trainable Parameters	332,504,064

Table 5: Perceiver-AR Model Diagram

The description of the distinct layers depicted in the Perceiver-AR model diagram is presented below:

- **Embedding**: Convert the input tokens into a continuous high-dimensional space.
- **Rotary Embedding**: Applies a unitary transformation to the input sequence, designed to make it easier for the model to learn spatial relationships between tokens. This layer has no parameters.
- **Module List**: Contains instances of another module list, with each instance being a stack of self-attention and cross-attention layers, and each layer in the stack having multiple attention heads.
- **Linear**: This is a linear layer that decodes the output of the self-attention and cross-attention layers into a sequence of tokens.

Comparing the two models, we see **they have a difference by an order of magnitude in the number of parameters.** This difference could have a significant impact on the performance of the models during training and the generated output quality, and it was one distinction we wanted to highlight. On one hand, having a larger number of parameters could make Perceiver-AR more

complex and expressive, capturing more intricate patterns and relationships in the input data, possibly being able to generate more diverse musical sequences. However, the larger number of parameters could also make it more prone to overfitting, potentially producing more noise or dissonant sequences. On the other hand, having a smaller number of parameters could make Music Transformer less expressive, potentially limiting it to produce simpler and more repetitive music pieces. However, it could be also less prone to overfitting, resulting in more creative sequences.

Training Batches

Dataset	Songs	Size	Music-Transformer Training Batches	Perceiver-AR Training Batches
maestro 3.0.0	1,276	85 MB	1,276	1,719
GiantMIDI Piano	10,855	281 MB	10,855	9,451
ailabs1k7	1,747	25 MB	1,747	556
Rock Piano MIDI	8,761	187 MB	8,761	2,321
ADL Piano MIDI	11,086	136 MB	11,086	2,888
Los Angeles MIDI ssegment	46,997	1.3 GB	46,997	25,179

Table 6: Training Batches across Training Datasets-Models

Training Batches: Total training inputs presented at each epoch of training.

For Music Transformer, we consider each musical piece as a training batch and the number of batches for a dataset is equal to the number of songs in that dataset. Thus, during the training process, the model is given an entire song as input at each step. The choice of training batch depends on the model's specialization.

For Perceiver-AR, we split the dataset into constant size input batches, and the total number of these batches is not equal to the number of songs in the dataset. This approach ensures a balanced training batch as each batch contains the same amount of musical information for all instances of the Perceiver-AR model.

Determining the number of input batches for each dataset in Perceiver-AR case also provides some information about the duration and musical information of the songs at each dataset. For example, for maestro-3.0.0 with longer classical piano performances training batches number is larger than songs number, in contrast with the other datasets where the songs duration is smaller so each song is a portion of a training batch and the training batches number is smaller than songs number.

Training Process

Objective Function: We used the loss function as an objective function. We tried to minimize the validation loss and the difference between training-validation loss.

Validation: It was performed periodically at the end of each epoch for Music Transformer and every 100 training steps, more frequently, for Perceiver-AR.

Early Stopping

In our implementation of both models, we applied an early stopping procedure to determine the optimal step to stop training and save our models:

- **Our model would train for a certain amount of time to reach an established validation loss required value depending on the model's type.** We set those limits based on the capacity and scale of each model and also our practical experience in training the models.
    - For Music Transformer: Validation Loss < 2
    - For Perceiver-AR: Validation Loss < 1
- **After the models had reached the established performance, if the validation checkpoint established a new optimal validation loss value, we would save the checkpoint.** We would also plot the learning curves of the model's state to be able to evaluate the model training performance later on.
- **If the model had not established a new optimal validation loss value for a certain number of training steps, we would stop the training.** Again, in this case we set those limits based on the capacity and scale of each model and our practical experience in training the models. Setting smaller step limits would also probably produce the expected results. The number of steps for the two models were:
    - For Music Transformer: $\frac{100 * Training\ Batches}{Batch\ Size}$ steps
    - For Perceiver-AR model: 1,000 steps

Trained Models Selection

During the training process, we saved the checkpoints of all models corresponding to multiple validation loss values for each of the 12 different model-dataset combinations. We then analyzed the learning curves and selected the best-performing checkpoint for each model-dataset combination based on the following criteria:

- **Achieving minimal validation loss values.**
- **Minimizing the difference between training and validation loss.**
- **Achieving similar performance with fewer training steps to avoid overfitting.**

Learning Curves

In this section we present the learning curves of loss for each combination of model-training dataset, for our 12 different trained models combinations.

maestro 3.0.0

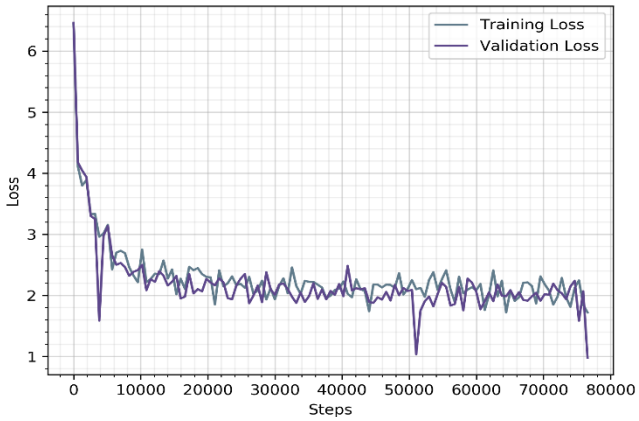

Figure 6.2: Music Transformer MAESTRO 3 Training-Validation Loss Learning Curve

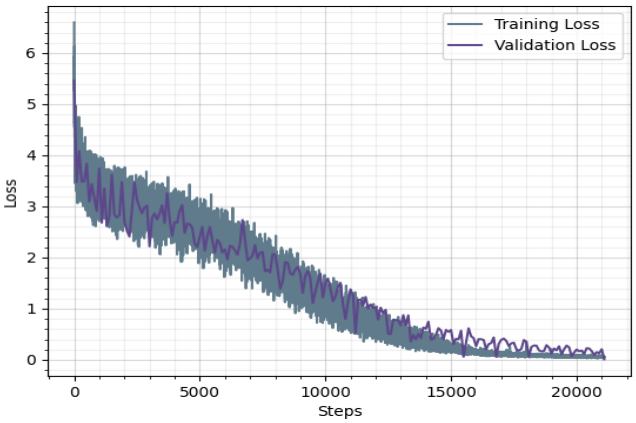

Figure 6.3: Perceiver-AR MAESTRO 3 Training-Validation Loss Learning Curve

GiantMIDI-Piano

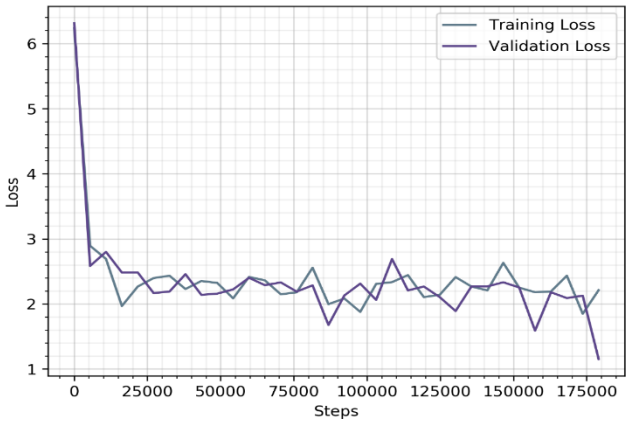

Figure 6.4: Music Transformer GiantMIDI-Piano Training-Validation Loss Learning Curve

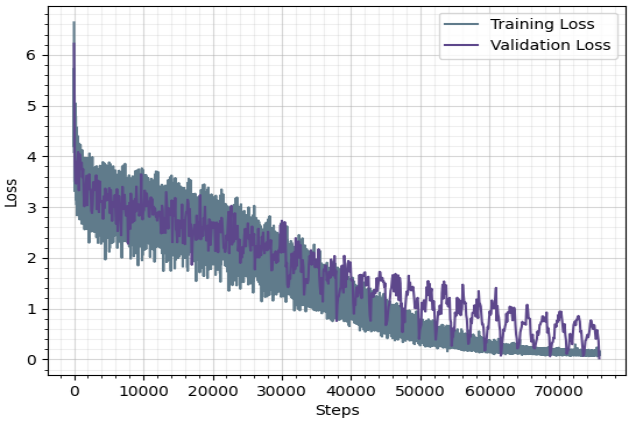

Figure 6.5: Perceiver-AR GiantMIDI-Piano Training-Validation Loss Learning Curve

Ailabs1k7

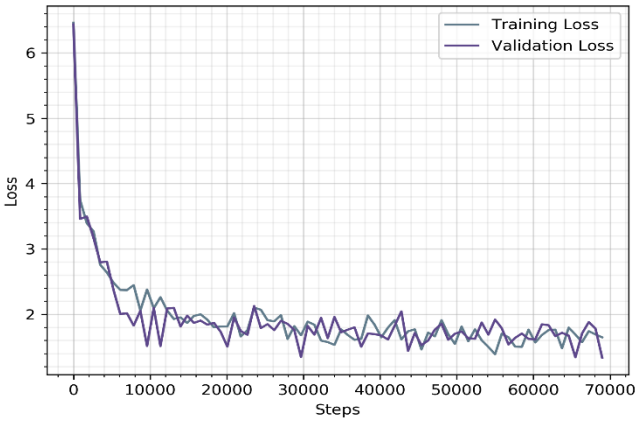

Figure 6.7: Music Transformer Ailabs1k7 Training-Validation Loss Learning Curve

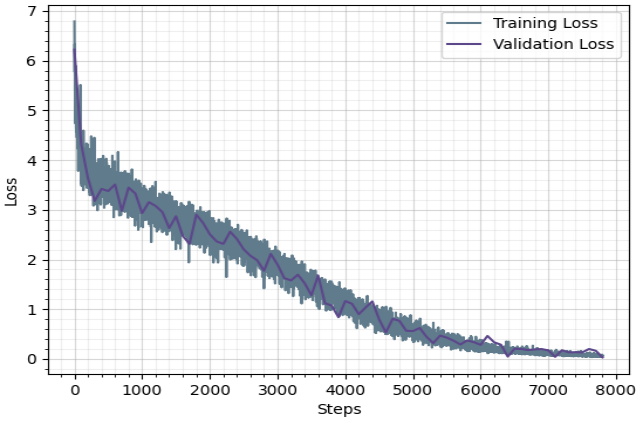

Figure 6.6: Perceiver-AR Ailabs1k7 Training-Validation Loss Learning Curve

Rock Piano MIDI

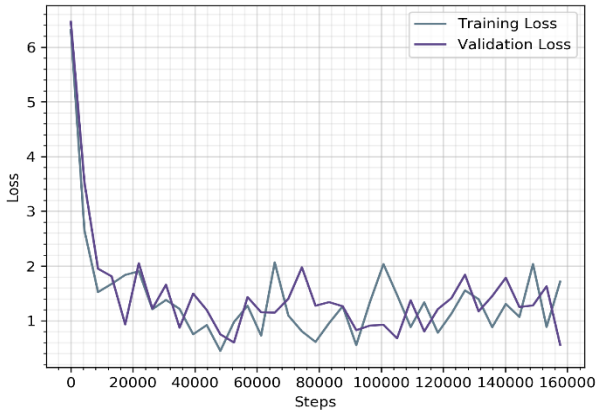

Figure 6.9: Music Transformer Rock Piano MIDI Training-Validation Loss Learning Curve

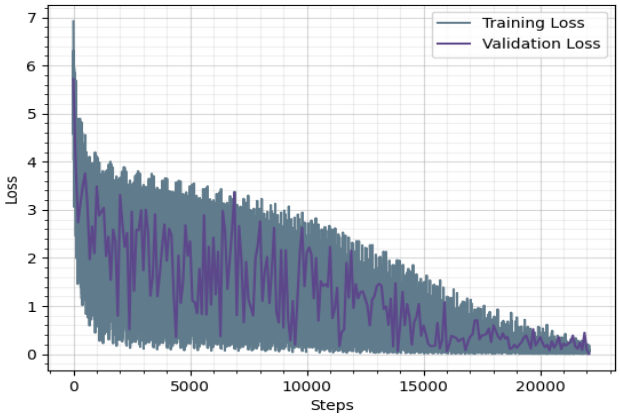

Figure 6.8: Perceiver-AR Rock Piano MIDI Training-Validation Loss Learning Curve

ADL Piano MIDI

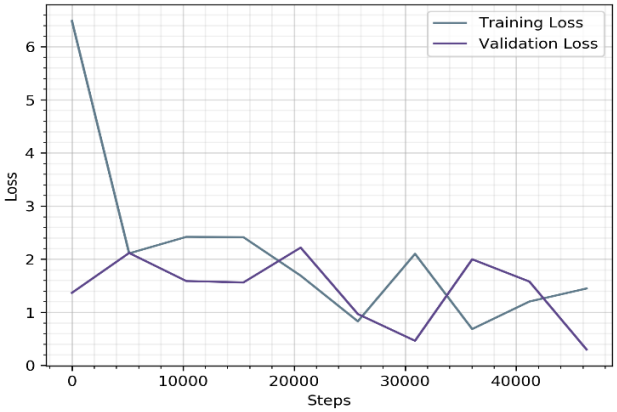

Figure 6.11: Music Transformer ADL Piano MIDI Training-Validation Loss Learning Curve

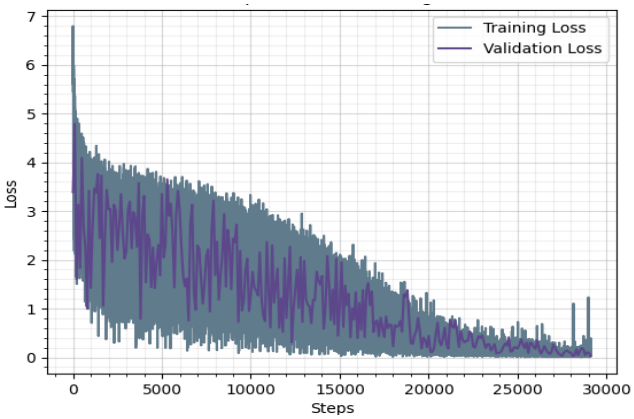

Figure 6.10: Perceiver-AR ADL Piano MIDI Training-Validation Loss Learning Curve

Los Angeles MIDI segment

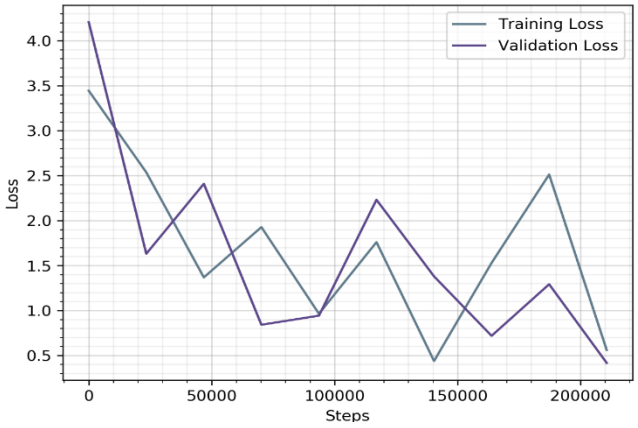

Figure 6.12: Music Transformer Los Angeles MIDI segment Training-Validation Loss Learning Curve

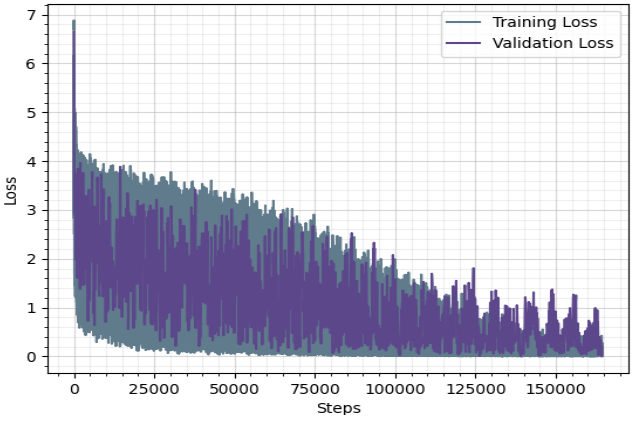

Figure 6.13: Perceiver-AR Los Angeles MIDI segment Training-Validation Loss Learning Curve

Trained Models

We present the selected models for each of the 12 different model-dataset combinations in the following tables. Along with the previously presented characteristics of each dataset, we include new metrics such as training batches, training steps, training epochs, and validation loss for the selected models. These metrics provide insight into the performance of each model during the training process.

Training Batches: Number of training inputs presented at each epoch of training. We note that the representation of training batches varies across our Music Transformer and Perceiver-AR model implementations.

Training Steps: Total number of training batches for each selected model. While not directly comparable for the two types of models, it provides a measure of the total number of training inputs required for each model to achieve optimal performance.

Training Epochs: **Approximation of the number of times each model was presented with the entire training dataset before achieving the required performance**, calculated as the saved model steps divided by the training batches. We note that the number of epochs needed to achieve satisfactory performance may vary depending on the size of the dataset and that is not directly comparable for the two types of models.

Validation Loss: Validation loss at the point each model was saved.

Music Transformer

Dataset	Songs	Size	Training Batches	Training Steps	Training Epochs	Validation Loss
maestro 3.0.0	1,276	85 MB	1,276	76,560	≈60	0.9793
GiantMIDI Piano	10,855	281 MB	10,855	179,091	≈17	1.1543
Ailabs1k7	1,747	25 MB	1,747	68,967	≈40	1.3368
Rock Piano MIDI	8,761	187 MB	8,761	157,680	≈18	0.5572
ADL Piano MIDI	11,086	136 MB	11,086	46,332	≈5	0.2999
Los Angeles MIDI segment	46,997	1.3 GB	46,997	210,672	≈5	0.4161

Table 7: Music Transformer Trained Models

Perceiver-AR

Dataset	Songs	Size	Training Batches	Training Steps	Training Epochs	Validation Loss
maestro 3.0.0	1,276	85 MB	1,719	21,100	13	0.0098
GiantMIDI Piano	10,855	281 MB	9,451	75,800	9	0.0192
Ailabs1k7	1,747	25 MB	556	7,800	14	0.0313
Rock Piano MIDI	8,761	187 MB	2,321	22,100	10	0.0063
ADL Piano MIDI	11,086	136 MB	2,888	29,100	11	0.0259
Los Angeles MIDI segment	46,997	1.3 GB	25,179	164,400	7	0.0037

Table 8: Perceiver-AR Trained Models

Implementation

Music Transformer

The training code is included on /music-transformer/train.py file and was based on code from (K. Yang, 2021a) train.py file further developed with additional features and functionalities, including saving training and evaluation metrics, plotting metrics graphs, checking for best performance metric values, implementing early stopping etc. It uses TensorFlow 2.0.0 and it was run on NVIDIA Tesla K40m GPU at ARIS GRNET supercomputer on December 2022 requiring about 10-11 GB GPU memory for the selected hyperparameters and a total amount of time 1-2 days for each model associated to the size of the training dataset.

At first, arguments such as the learning rate, batch size, dataset name, maximum sequence length, number of epochs, etc., are set. The directories for saving the trained model and its training and evaluation outputs are then defined, the dataset is loaded using a custom Data class, and the Music Transformer model is set up using the MusicTransformerDecoder class. The Adam optimizer from the TensorFlow library and the custom loss function defined in the callback module are used. Then, arrays are initialized to store the training and validation losses, as well as a variable for tracking the best validation loss.

The main training process is composed of two nested for-loops, with the outer loop iterating over the specified number of epochs, and the inner loop iterating over the number of batches in the dataset. The dataset is iterated over in batches, and the model parameters are updated using backpropagation and the defined optimizer. The validation metrics are being calculated every 100 batches. The arrays storing the training and validation losses, as well as the best validation loss, are also updated in the loop.

Every 100 batches, the current state of the model is evaluated. The resulting loss metric, along with a histogram of the source and target data for this batch, and an attention image summary for each weight in the model, are saved. If the current validation loss is better than the best seen so far (as determined by the thresholds or the last best seen value), the best value is updated and the current state of the model is saved. New plots of the training and validation loss, respectively, are also created. The best performing model and associated plots are updated and saved whenever a new best value is achieved as the code continues to run through all epochs and batches.

Next, the procedure that implements early stopping checks if the current epoch number is greater than a minimum 50 and if the last validation loss is worse than the best value seen so far. In addition, it is checked whether the epoch when the best validation loss was last seen is more than 100*batch_range steps away from the current epoch, indicating that the validation loss has not been improved for a significant number of epochs, and the training should be completed.

After completing the training process, training and validation value arrays are saved as NumPy (Harris et al., 2020) files for later analysis or plotting and training-validation loss learning curve is plotted, using Matplotlib (Hunter, 2007), to visualize the overall performance of the model.

Perceiver-AR

The training code is included on /perceiver-ar/Perceiver-Solo-Piano-Maker.ipynb notebook and was based on code from (Lev, 2022b) Perceiver-Solo-Piano-Maker.ipynb further developed with additional features and functionalities, including saving training and evaluation metrics, checking for best performance metric values, implementing early stopping etc. It uses PyTorch 1.12.1 and was run on NVIDIA GeForce GTX 1080 Ti and NVIDIA TITAN Xp GPU at Pink-Floyd server of Artificial Intelligence and Learning Systems Laboratory (AILS) ECE NTUA between November – December 2022 requiring about 10-11 GB GPU memory and a total amount of time 2-4 days for each model associated to the size of the training dataset.

In the first cells, before “Preprocess Training Dataset” cell, the directory paths for the training dataset are set up and several libraries and module TMIDIX (Lev, 2021) are imported. Next, the dataset name, basic directory, and current time variables are set. Several directories are created, such as the dataset directory, the training outputs directory, and the saved models’ directory.

After the “Load Training Data” cell, the preprocessed training data are loaded from the single pickle file and are concatenated into a single tensor. Then, the model is initialized and constants are defined to set up the model's parameters, such as sequence length, batch size, and learning rate. The PerceiverAR class is used to instantiate the model. The model's hyperparameters, such as the number of tokens, the dimensions, the number of layers, and the number of attention heads, are set. The data is prepared for training by creating a MusicDataset object that samples the data sequentially and then is loaded into train and validation loaders using the DataLoader class. The optimizer is instantiated with the Adam optimizer, and its learning rate is set.

The training loop consists of an outer loop over the number of batches to process, and a gradient accumulation loop runs within that, where the gradients are calculated for each batch of the data loader. The model parameters are updated using the optimizer, and the gradients are clipped to a maximum norm of 0.5 before the optimizer step is taken. The model is evaluated on a validation dataset every 100 steps, and the current state of the model is saved if the current evaluation loss is better than the best seen so far. New learning curves of the training and validation loss are also created. The best performing model and associated plots are updated and saved whenever a new best value is achieved.

Next, the early stopping procedure checks if the last validation loss is worse than the best value seen so far. In addition, it is checked whether the number of steps when the best validation loss was achieved is more than 1,000 steps away from the current step, indicating that the validation loss has not been improved for a significant number of steps, and training should be completed.

In “Save stats graphs and value arrays” cell, after completing the training process, training and validation value arrays are saved as NumPy (Harris et al., 2020) files for later analysis or plotting and training loss, validation loss, training-validation loss graphs are plotted using Matplotlib (Hunter, 2007).

Chapter 7: Output Generation

Number of Outputs

When generating MIDI outputs using our models, the number of outputs to generate per trained model is a crucial consideration. One factor to consider is the size of the dataset used to train the model. For small datasets, generating a larger number of outputs may result in more diverse and interesting outputs. On the other hand, for large datasets, generating fewer outputs may still result in a diverse set of outputs, but with less redundancy. Another factor to consider is the specific goals of the project. In our case, having datasets ranging from 1,000-47,000 songs approximately we aimed to enhance the advantages of generating a feasibly large number of outputs being able to showcase the full range of possibilities of our models.

Advantages of generating a larger number of outputs include the potential for increased diversity, which can lead to more creative outputs, that are less related to random factors. Additionally, a larger number of outputs may make it easier to identify patterns in the outputs, demonstrating possible improvements or differences between the models. However, there are also some potential disadvantages to consider with the main one being that it may be more difficult to evaluate a large number of outputs and determine which ones are of high quality. Additionally, a large number of outputs may result in a higher computational cost and longer generation time.

Aiming to compromise between all the above factors and mitigate possible disadvantages **we decided to produce 600 outputs for each trained model.** We considered this number to be relatively large compared to those found in literature. Additionally, from the perspective of computational time required, it was feasible corresponding to our available resources.

Primers Selection

Another consideration that had to be made was the origin and distribution of the input sequences (primers) our models would be provided to generate the outputs. **We decided to use a fixed set of 600 total sequences with 100 random pieces from each training dataset** for a number of reasons. We produced two types of each output based on each selected primer, one including the primer and one without the primer in order to be able to evaluate our outputs in all possible ways objectively and subjectively.

One advantage of using the same number of primers from each dataset is that it ensures that models are trained and evaluated on a similar range of musical genres and patterns. This is especially important in our case where most of our training datasets differ significantly in terms of musical genre, style, complexity and number of pieces. By using the same number of input sequences from each dataset, **the models are exposed to a similar range of musical material, possibly improving their ability to generate diverse and creative outputs. Another advantage is ensuring a balanced representation of each dataset in the generated outputs.** In our case where

some datasets are significantly larger than the others, using the same number of input sequences from each dataset can help prevent bias towards any dataset in the generated outputs.

We acknowledge that our approach may not be optimal for all datasets due to their unique characteristics. Specifically, **some datasets may require different amounts of input primers from each training dataset to achieve a satisfactory diversity of outputs.** However, determining the ideal proportions of inputs from each dataset is a challenging and complex problem that is beyond the scope of our work.

Primer & Output Number of Tokens

Choosing the number of primer input event tokens and output generated tokens is another important decision in our music generation task, as it can have a significant impact on the quality and diversity of the generated sequences. In our case, we experimented with different combinations of primer and output token numbers to determine the optimal combination we would use for both models.

One primary factor we considered was the length of the generated sequences and aimed to generate music pieces with a duration of 10 to 60 seconds. This approach ensured that the majority of the generated sequences would be appropriate for subjective evaluation purposes.

Another factor we considered was that **Perceiver-AR model performed best with a primer size of 512 and an output size of 512 tokens, according to our experiments.** In contrast, other combinations we tested, including 128-128, 256-128, 256-256, 256-512 token pairs were limited to lower-quality generated sequences. **Regarding Music Transformer models, we did not observe any significant variation in the quality of the generated sequences for different primer tokens and output tokens configurations.** Although we only tested a few configurations for this model, we concluded that **using a primer size of 512 and an output size of 512 tokens produced satisfactory outputs.** Nonetheless, it's worth noting that this conclusion is based on limited testing, and further experimentation may be required to thoroughly assess the effects of different primer and output token numbers.

Overall, considering the above factors **we decided to implement the combination of a primer size of 512 and an output size of 512 tokens** to ensure the generated sequences are of high quality, as diverse as possible, and fit for the intended evaluation purposes. However, it is essential to recognize that setting these specific parameters may have potential drawbacks. For instance, setting this fixed primer size could limit the ability of the models to generate longer-term patterns in the music data. Therefore, while these parameters can be useful in producing outputs of acceptable quality, it's also important to explore other options throught more extensive experimentation.

Perceiver-AR Generation Mode

In Perceiver-AR model implementation there are included two generation modes by (Lev, 2022b). The first mode named "Single Continuation Block Generator" takes the prime sequence of MIDI

tokens as input and generates the specified number of tokens (musical events) at once as a single generation block. The second mode named "Auto-Continue Custom MIDI" takes the prime sequence of MIDI tokens as input and generates the specified number of notes (not music tokens) at multiple loops by generating 4 tokens each time and adding the generated output as primer at each loop until it reaches the required number of notes.

We decided to use the "Single Continuation Block Generator" Perceiver-AR generation mode due to high similarity with Music Transformer generation process, generating all the required tokens at once. This allowed us to have consistency in the generation process. **It also provides more control over the generation process, as the length of the output can be adjusted by specifying the number of tokens to generate**. This helped ensure that the generated outputs were within the desired length range of 10-60 second. In contrast, in the second mode, although we had the capability of generating longer pieces of music, it was more difficult to control the length of the outputs.

Implementation

Primer Selection

The primer selection code is included on /primer-selection.ipynb notebook. A random subset of 100 files is selected from each training dataset to be included in the primer list. The directories of the training datasets are defined and the location to save the primer pickle list is specified. For each dataset, the main loop of the code iterates through and selects a random subset of 100 MIDI files from the directory, checking also if each file can be processed by the pretty_midi (Raffel & Ellis, 2014) and muspy (H.-W. Dong et al., 2020) libraries. If a file can be processed, it is added to the primer dictionary, otherwise it is skipped. Finally, the primer files list is saved to a pickle file.

Output Generation: Music Transformer

The implementation of model outputs generation is included on /music-transformer/batch_generate.py file and was based on code from (K. Yang, 2021a) generate.py file further developed to include batch generation functionality. The output generation code was run on NVIDIA GeForce GTX 1080 GPU at Ironman server of Artificial Intelligence and Learning Systems Laboratory (AILS) ECE NTUA between December 2022 and January 2023 requiring about 6-7 GB GPU memory.

Firstly, the MusicTransformer model and custom layers, are imported. The required and optional parameters for the script are set, such as the maximum sequence length, mode of generation (encoder-decoder or decoder only), beam size, length of generated output, and directory paths for model loading and MIDI output saving. The token number to be generated is set to 512 and the total length of the output to 1,024 with the mode being decoder only, as described above.

The model and directories for saving generated MIDI files are set up, and the list of primers is loaded from the pickle file. The script loops through each primer to generate a MIDI file. For each primer, the script encodes the MIDI input, creates two MIDI files for the generated sequence with

and without the primer, and generates MIDI outputs using the specified mode and beam size. The script generates every MIDI output using the generate method of the MusicTransformer object once. The method returns a list of integers representing the generated output sequence, which is then decoded back into a MIDI file using the decode_midi method from the midi_processor module. If an error occurs during MIDI generation, the script retries the generation until it succeeds.

Output Generation: Perceiver-AR

The implementation of model outputs generation is included on /perceiver-ar/perceiver-solo-piano-batch.py file and was based on code from (Lev, 2022b) Perceiver_Solo_Piano.ipynb notebook further developed to include batch generation functionality. The output generation code was run on NVIDIA GeForce GTX 1080 GPU at Ironman server of Artificial Intelligence and Learning Systems Laboratory (AILS) ECE NTUA between December 2022 and January 2023 requiring about 6-7 GB GPU memory.

First of all, the Perceiver-AR model, the Autoregressive wrapper for the model, and the TMIDIX module that provides tools for reading and manipulating MIDI files, are imported. Next, the environment is set up, setting up the basic directories for the input and output files. The list of input MIDI sequences (primers) is loaded from the pickle file, followed by the pre-trained Perceiver-AR model checkpoint. The model hyperparameters, including the number of tokens, the dimensionality of the model, the number of layers, and the size of the attention heads, are also set.

The "augment" function is defined, which takes an input sequence of MIDI notes and returns a set of augmented sequences. Each augmented sequence is created by shifting each note up or down by a random amount within a predefined range. The purpose of this function is to increase the diversity of the input data and improve the quality of the generated output.

Finally, batch generation parameters are set and a loop is defined by the code to generate new MIDI sequences based on each input MIDI sequence. For each input sequence, the code generates a set of augmented sequences using the "augment" function, then uses the Perceiver model to generate a new sequence of tokens based on each augmented sequence, as a single generation block. There are generated two kinds of sequences, one including the primer input sequence and another one with the primer tokens being trimmed. The resulting sequences are then written to MIDI files and saved to the output directory.

Chapter 8: Objective Evaluation

Objective Metrics

After producing our midi outputs for every trained model, we implemented two evaluation strategies to determine the quality of our generated music and make required comparisons. One of these strategies was objective evaluation with music objective metrics, to obtain some analysis about the characteristics of our generated musical pieces. These objective metrics can be either absolute metrics providing insights into properties of generated or training data or relative metrics comparing two sets of data, e.g., training and generated. For the objective evaluation part of our project, we selected some absolute musical metrics implemented by (H.-W. Dong et al., 2020) Muspy and (Raffel & Ellis, 2014) pretty_midi modules suitable for our comparison purposes, with this section mainly based in their work.

As (H.-W. Dong et al., 2020) states, Muspy implements a number of objective measures that have been suggested in literature. These objective metrics could be utilized to assess a music generating system by analyzing the statistical difference between the training data and the generated samples. These metrics are separated on two categories according to musical information they provide: pitch-related metrics and rhythm-related metrics. Also, according to (Raffel & Ellis, 2014), there are multiple functions in the PrettyMIDI class instances for analyzing musical data, with some of them being estimate tempo, get beats, get chroma etc.

In the following sections we present the objective metrics we selected in our objective evaluation, including a description of each metric. We explain in the according section the reasons and objectives of each choice.

Selected Objective Metrics

Metric	Related to	Implementation	Description
Pitches Used	Pitch	Muspy	Number of different pitches within a sample.
Pitch Range	Pitch	Muspy	Subtraction of the highest and lowest used pitch in semitones.
Pitch Classes Used	Pitch	Muspy	Unique pitch classes used.
Polyphony	Pitch	Muspy	Average number of pitches played concurrently, when at least one pitch is on.
Polyphony Rate	Pitch	Muspy	Ratio of time steps where multiple (2) pitches are on.
Scale Consistency	Pitch	Muspy	Largest pitch-in-scale rate over all major and minor scales. Pitch-in-scale rate is the ratio of the number of notes in a certain scale to the total number of notes.
Pitch Entropy	Pitch	Muspy	Shannon entropy of the normalized note pitch histogram.
Pitch Class Entropy	Pitch	Muspy	Shannon entropy of the normalized note pitch class histogram.
Empty Beat Rate	Rhythm	Muspy	Ratio of the number of empty beats (where no note is played) to the total number of beats.

Tempo Estimation	Rhythm	Pretty-Midi	Best tempo estimation from empirical estimate of tempos.

Table 9: Selected Objective Metrics

Used Metrics Equations

- $polyphony = \frac{\#(pitches\_when\_at\_least\_one\_pitch\_is\_on)}{\#(time\_steps\_where\_at\_least\_one\_pitch\_is\_on)}$
- $polyphony\_rate = \frac{\#(time\_steps\_where\_multiple\_pitches\_are\_on)}{\#(time\_steps)}$
- $scale\_consistency = \max_{root,mode} pitch\_in\_scale\_rate(root, mode)$
- $pitch\_entropy = -\sum_{i=0}^{127} P(pitch = i)\log_2 P(pitch = i)$
- $pitch\_class\_entropy = -\sum_{i=0}^{11} P(pitch\_class = i) \times \log_2 P(pitch\_class = i)$
- $empty\_beat\_rate = \frac{\#(empty\_beats)}{\#(beats)}$

In the process of deciding the objective evaluation metrics we had to consider many contradicting factors. First of all**, one of the most important factors was the available musical information in our datasets.** Really useful and enlightening features like key signatures, or time signatures of a song could not be used simply because this information is not provided in our training datasets and generated outputs. According to our knowledge, it would require several research, being the scope of another project, to implement the extraction of such information from musical pieces. Furthermore, **we wanted to conduct a general statistical analysis on our output datasets not focusing on single output files** that could yield to misleading considerations. On the account of this fact, metrics like Pitch Class Histogram, Tempo Histogram, Average Pitch Interval, could not be viewed through the scope of a dataset, only referring to a single musical piece.

Furthermore, **we tried to exclude metrics directly correlated with musical piece length as our training datasets have varying music pieces lengths and our outputs are limited to approximately one minute duration.** Considering this fact, metrics like notes used or number of beats would not provide reliable information on the quality of the generation process of our models. However, **we could not totally exclude the piece length factor from all our metrics, with metrics like pitches used, pitch range and pitch classes used, being indirectly related to piece length**. Excluding these metrics would conceal important information about pitch characteristics we intend to evaluate on our generated outputs.

Objective Metrics Statistics

For the objective statistical analysis of our training and output datasets we generated a descriptive statistic table including each metric and dataset combination. Our results for each metric included mean, median and standard deviation (std). We included the median as, unlike the mean, it is not affected by extremely large or extremely small outlier values.

	Pitch Range	Pitches Used	Pitch Classes	Polyphony	Polyphony Rate	Scale Consistency	Pitch Entropy	Pitch Class Entropy	Empty Beat rate	Tempo Estimation
adl-piano-midi: Training Dataset										
mean	46.8	32.35	9.92	3.141	0.506	0.923	4.189	2.894	0.117	80.8
median	46.0	30.00	10.00	3.038	0.516	0.947	4.185	2.881	0.030	81.3
std	14.1	13.74	1.89	0.991	0.275	0.078	0.598	0.309	0.194	24.7
adl-piano-midi: Music Transformer Outputs										
mean	28.9	11.19	5.86	2.902	0.448	0.952	2.549	1.878	0.021	90.3
median	31.0	10.00	6.00	2.786	0.479	1.000	2.706	2.074	0.000	96.8
std	15	7.52	2.82	1.514	0.343	0.136	1.184	0.837	0.097	28.9
adl-piano-midi: Perceiver-AR Outputs										
mean	40.9	23.09	9.6	1.829	0.185	0.91	3.863	2.831	0.026	88.1
median	40.0	23.00	10.00	1.725	0.146	0.935	3.919	2.842	0.000	90.4
std	12.6	6.82	1.72	0.634	0.175	0.085	0.594	0.343	0.074	23.7
ailabs1k7: Training Dataset										
mean	54.5	36.73	9.26	4.728	0.86	0.962	4.53	2.793	0.008	90.4
median	53.0	35.00	9.00	4.604	0.895	0.987	4.504	2.742	0.007	90.0
std	9	10.45	1.83	1.399	0.112	0.058	0.352	0.221	0.006	13.2
ailabs1k7: Music Transformer Outputs										
mean	45.8	24.37	8.96	4.278	0.744	0.938	3.947	2.662	0.004	93.4
median	46.0	25.00	9.00	4.228	0.836	0.967	4.180	2.720	0.000	92.9
std	9.9	7.66	2.02	1.551	0.234	0.072	0.813	0.463	0.012	14.6
ailabs1k7: Perceiver-AR Outputs										
mean	55.1	27.38	10.12	2.843	0.539	0.926	4.001	2.694	0.005	100
median	55.0	27.00	10.00	2.800	0.542	0.942	4.081	2.711	0.000	101.4
std	10.8	5.12	1.41	0.568	0.169	0.062	0.438	0.312	0.015	11.8
GiantMIDI-Piano: Training Dataset										
mean	66.7	58.73	11.73	5.019	0.689	0.826	5.124	3.232	0.038	96.3
median	69.0	60.00	12.00	4.875	0.739	0.840	5.164	3.257	0.026	99.6
std	11.2	14.82	0.86	1.958	0.206	0.096	0.469	0.235	0.045	14.7
GiantMIDI-Piano: Music Transformer Outputs										
mean	45	22.17	8.79	3.604	0.495	0.9	3.687	2.567	0.042	92.9
median	45.5	23.00	9.00	3.206	0.506	0.917	3.961	2.741	0.000	98.5
std	14.5	9.71	2.55	1.97	0.315	0.093	0.961	0.638	0.092	22.9
GiantMIDI-Piano: Perceiver-AR Outputs										
mean	44.1	27.22	10.06	2.395	0.325	0.896	4.169	2.919	0.008	102.3
median	43.0	27.00	10.00	2.124	0.260	0.921	4.252	2.937	0.000	104.4
std	11.2	7.22	1.6	1.158	0.274	0.084	0.589	0.331	0.02	15.4
Los-Angeles-MIDI-segment: Training Dataset										
mean	49.4	35.25	9.84	3.966	0.637	0.932	4.202	2.818	0.016	99.4
median	53.0	35.00	10.00	3.879	0.766	0.957	4.315	2.817	0.007	101.1
std	18.6	16.72	2.09	1.939	0.339	0.076	0.737	0.341	0.033	21.4
Los-Angeles-MIDI-segment: Music Transformer Outputs										
mean	27.7	10.25	5.67	2.759	0.403	0.891	2.297	1.757	0.019	84.3
median	30.0	9.00	6.00	2.503	0.382	0.986	2.524	1.975	0.000	91.1
std	17	7.62	3.19	1.745	0.367	0.235	1.316	0.983	0.095	35.4
Los-Angeles-MIDI-segment: Perceiver-AR Outputs										
mean	40.5	22.73	9.09	2.245	0.298	0.922	3.835	2.753	0.04	92.6
median	41.0	23.00	9.00	1.966	0.194	0.952	3.958	2.776	0.000	95.6

std	12.9	8	1.95	1.135	0.291	0.08	0.716	0.406	0.105	22.3
maestro-3.0.0: Training Dataset										
mean	68.1	66	11.99	2.441	0.331	0.792	5.376	3.345	0.029	102.4
median	71.0	69.00	12.00	2.444	0.319	0.793	5.445	3.361	0.023	103.3
std	11.2	12.13	0.12	0.39	0.128	0.079	0.355	0.146	0.022	8.1
maestro-3.0.0: Music Transformer Outputs										
mean	44.3	23.59	9.55	3.659	0.507	0.884	3.888	2.711	0.02	96.2
median	44.0	24.00	10.00	3.281	0.518	0.897	4.040	2.817	0.000	100.8
std	13.4	8.05	2.01	1.836	0.258	0.09	0.761	0.501	0.06	19.7
maestro-3.0.0: Perceiver-AR Outputs										
mean	52.5	31.92	11.01	2.326	0.317	0.856	4.44	3.006	0.012	94.2
median	52.0	32.00	11.00	2.324	0.317	0.862	4.461	3.026	0.000	99.2
std	9.8	5.37	1.07	0.607	0.187	0.078	0.369	0.268	0.023	22.7
Rock-Piano-MIDI: Training Dataset										
mean	39.3	27.8	10.04	3.007	0.406	0.905	3.906	2.897	0.03	104.4
median	39.0	26.00	11.00	2.910	0.389	0.923	3.974	2.914	0.012	106.5
std	16.1	13.05	2.04	1.409	0.288	0.083	0.72	0.379	0.057	15.1
Rock-Piano-MIDI: Music Transformer Outputs										
mean	26.9	8.91	5.3	2.234	0.303	0.889	2.116	1.658	0.032	86.6
median	30.0	7.00	5.00	2.000	0.183	0.990	2.252	1.842	0.000	97.1
std	17	7.13	3.17	1.31	0.329	0.241	1.288	0.97	0.122	37
Rock-Piano-MIDI: Perceiver-AR Outputs										
mean	40.4	21.5	9.66	1.899	0.198	0.899	3.743	2.797	0.009	102.6
median	40.0	22.00	10.00	1.687	0.130	0.920	3.862	2.830	0.000	104.1
std	13.3	6.6	1.69	0.898	0.215	0.083	0.624	0.383	0.021	17.1

Table 10: Objective Metrics Statistics Table

Objective Metrics Correlations

We have also included the objective metrics Kendall's correlation heatmap to measure monotonic relationships between the objective metrics. Our metrics are not normally distributed with some of them also not being continuous. Also, we would like to explore generally monotonic relationships between them and not necessarily linear. For the above reasons Kendall's correlation was the most robust and generally preferred method (Zinda, 2021).

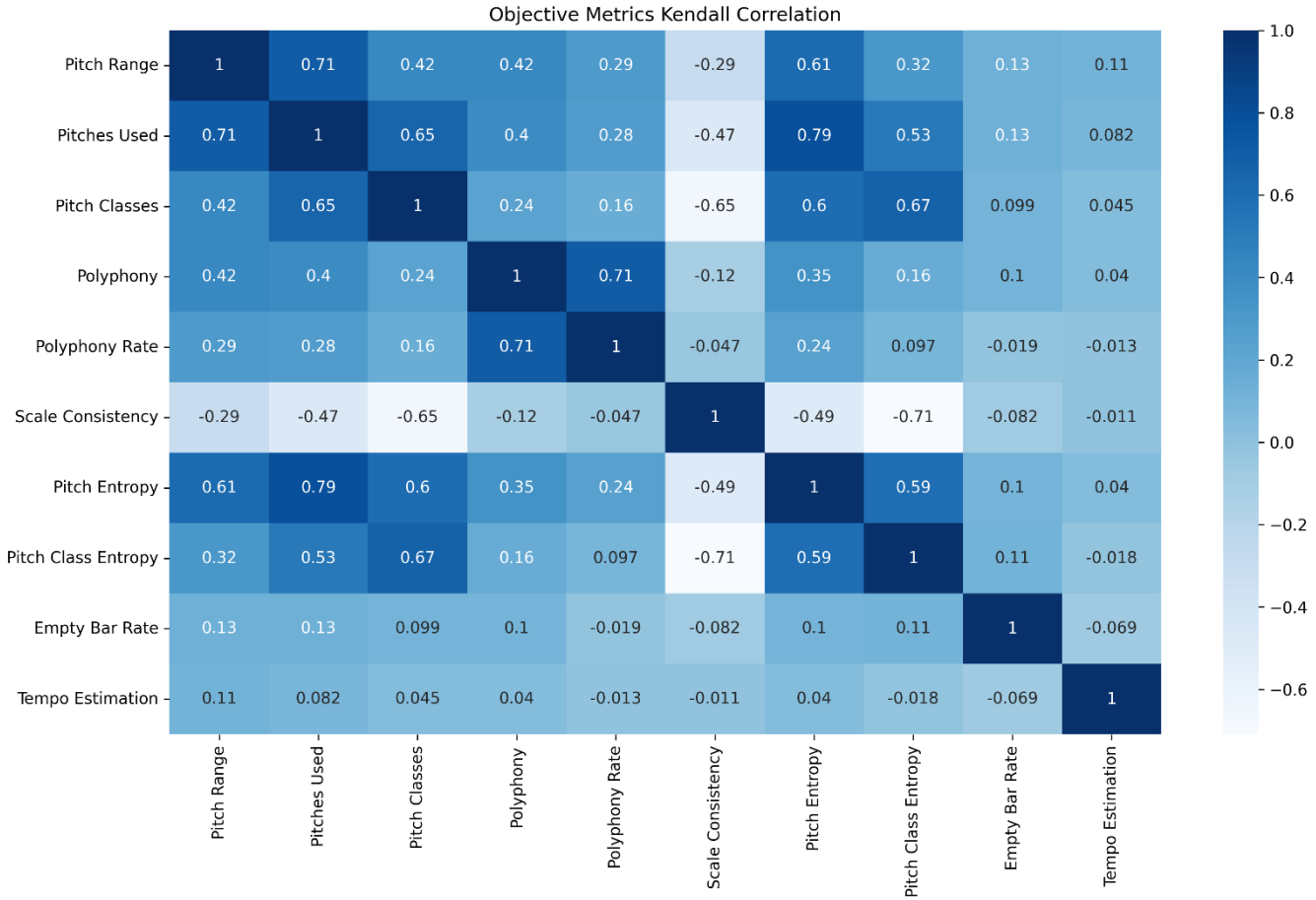

Figure 8.1: Objective Metrics Kendall Correlations Heatmap

Objective Metrics Plots

After calculating the statistical values for our objective metrics, we wanted to compare the distributions of all the datasets in a graphical way. For this purpose, we chose two types of graphs, the bar-plot and the violin-plot. In a bar-plot, each rectangle reflects an estimate of central tendency for a numerical variable, with the error bars showing the degree of uncertainty surrounding that estimate, providing a simpler evaluation and comparison of the datasets results. As error calculation method we used standard deviation (std) error. On the other hand, the violin-plot displays the distribution of quantitative data across a number of levels of one (or more) categorical variables comparing their distributions and including their kernel density estimations. It was appropriate for a more complex presentation of the distributions of each dataset. However, a slight we had to consider was the fact that the estimation procedure is affected by the sample size, and violins for small samples were probable to appear deceptively smooth.

In the following pages we present the two kinds of plots for all objective music metrics.

Pitches Used

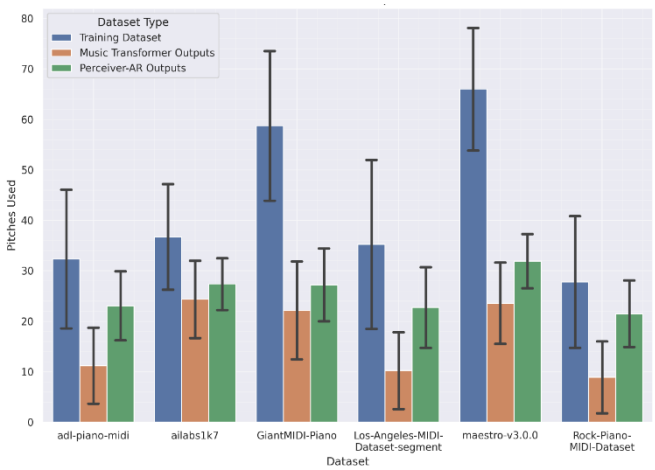

Figure 8.3: Pitches Used Bar-plot

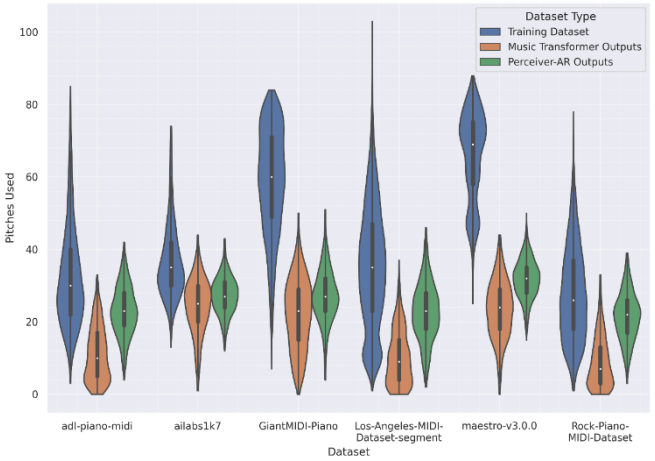

Figure 8.2: Pitches Used Violin-plot

Pitch Range

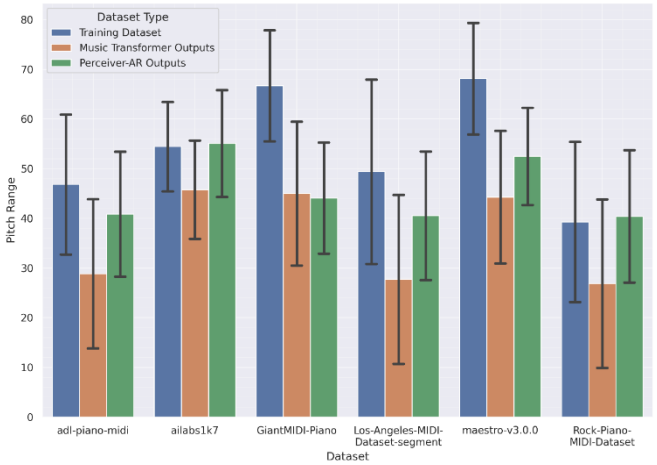

Figure 8.4: Pitch Range Bar-plot

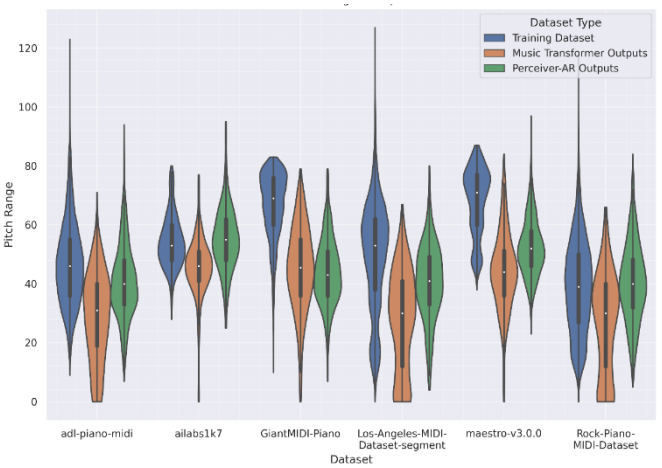

Figure 8.5: Pitch Range Violin-plot

Pitch Classes

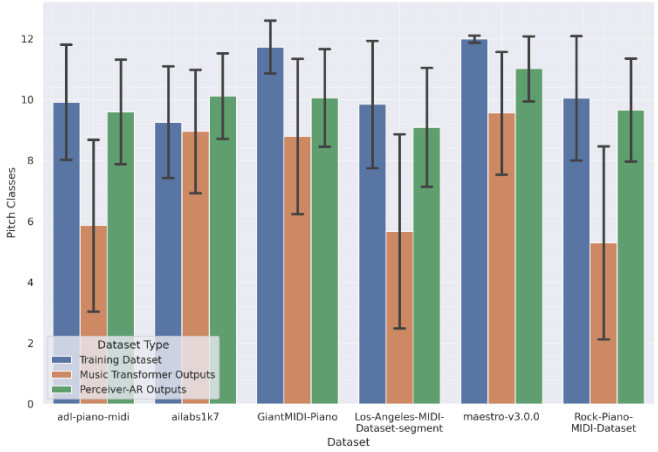

Figure 8.7: Pitch Classes Bar-plot

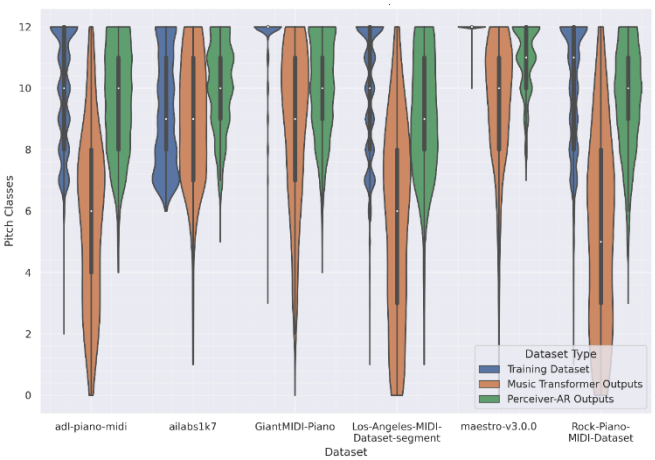

Figure 8.6: Pitch Classes Violin-plot

Pitch Entropy

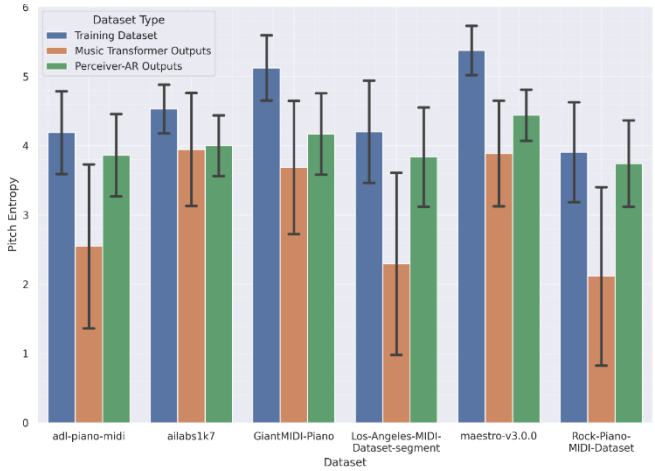

Figure 8.8: Pitch Entropy Bar-plot

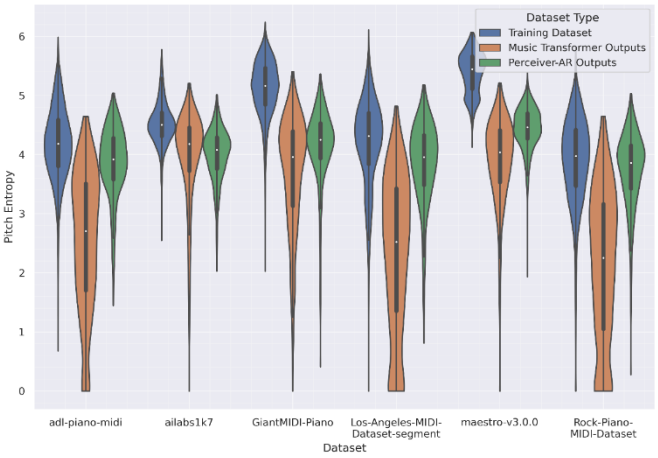

Figure 8.9: Pitch Entropy Violin-plot

Pitch Classes Entropy

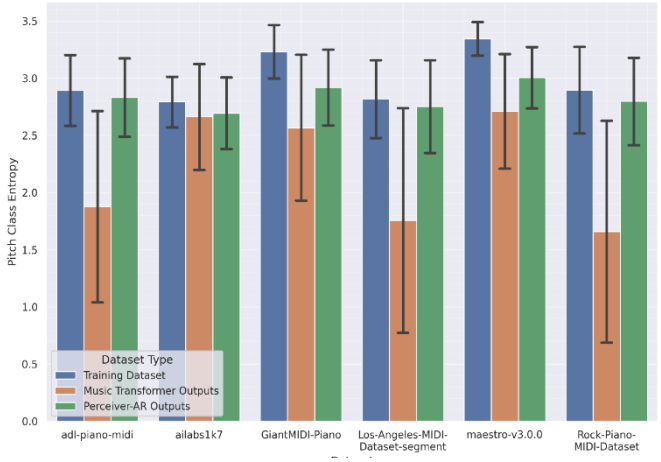

Figure 8.11: Pitch Class Entropy Bar-plot

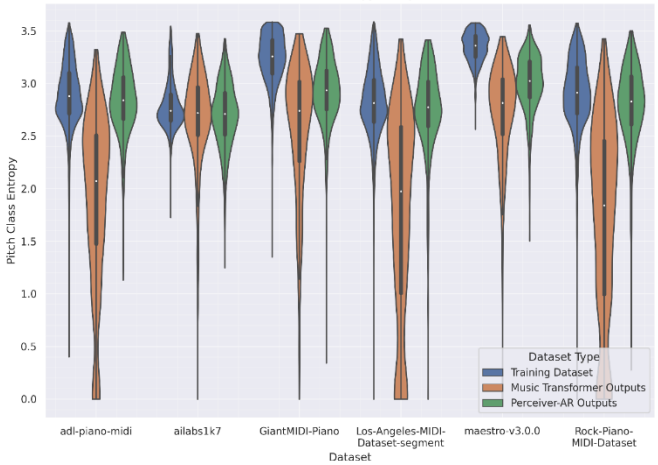

Figure 8.10: Pitch Class Entropy Violin-plot

Scale Consistency

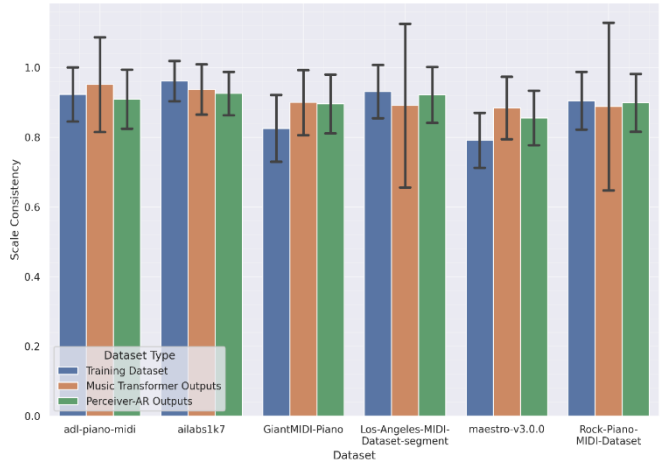

Figure 8.13: Scale Consistency Bar-plot

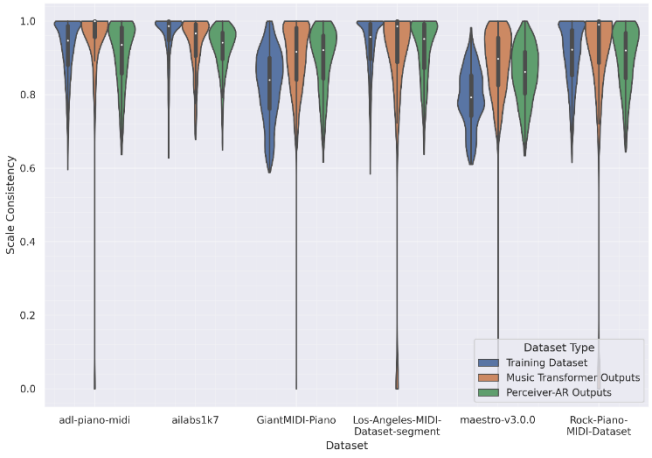

Figure 8.12: Scale Consistency Violin-plot

Polyphony

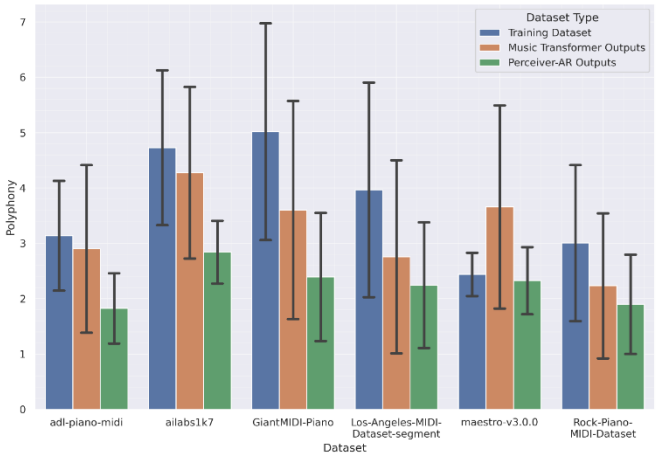

Figure 8.14: Polyphony Bar-plot

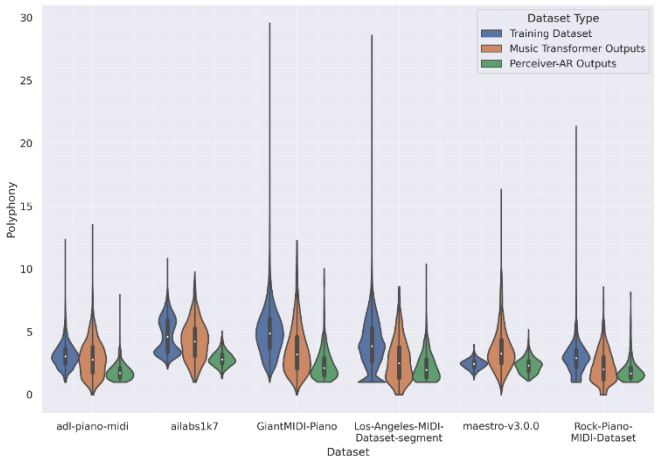

Figure 8.15: Polyphony Violin-plot

Polyphony Rate

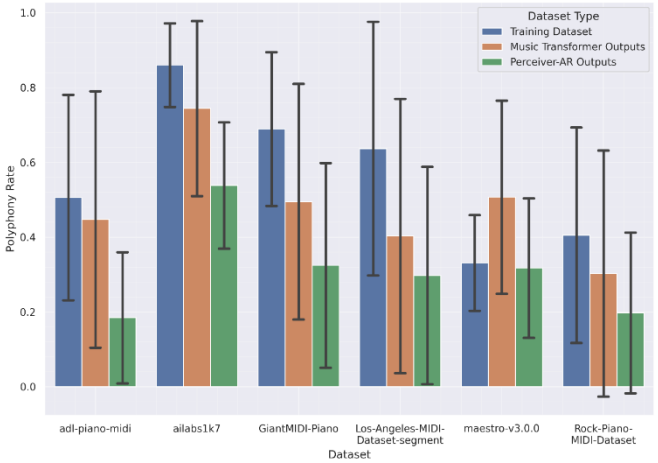

Figure 8.17: Polyphony Rate Bar-plot

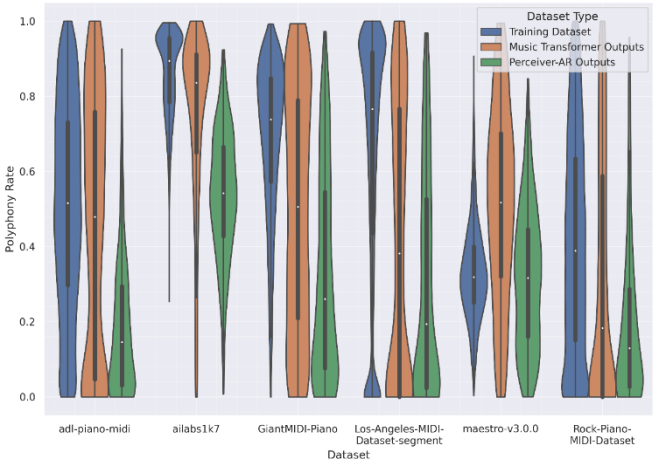

Figure 8.16: Polyphony Rate Violin-plot

Empty Beat Rate

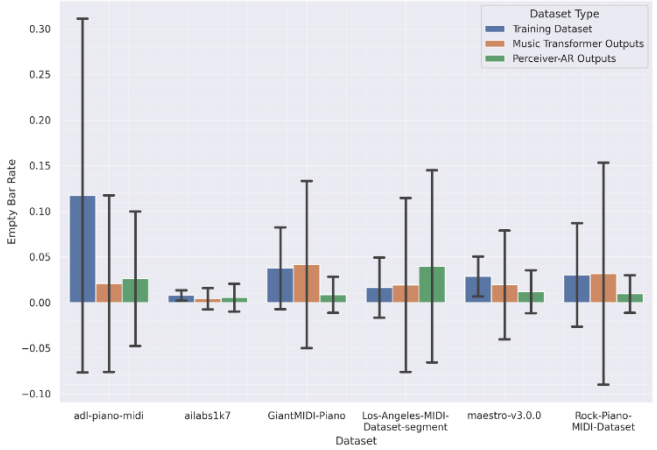

Figure 8.19: Empty Beat Rate Bar-plot

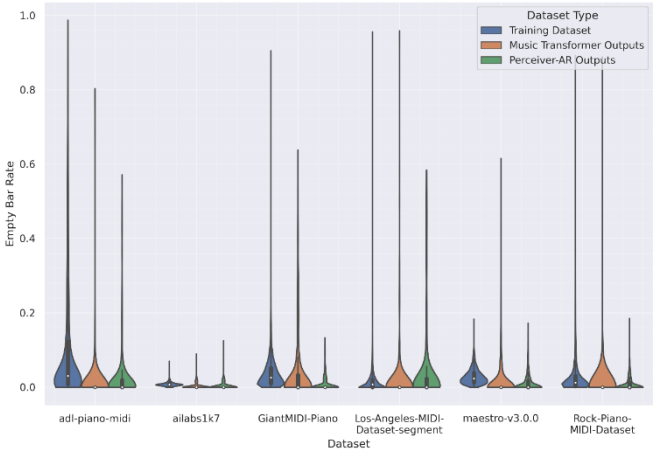

Figure 8.18: Empty Beat Rate Violin-plot

Tempo Estimation

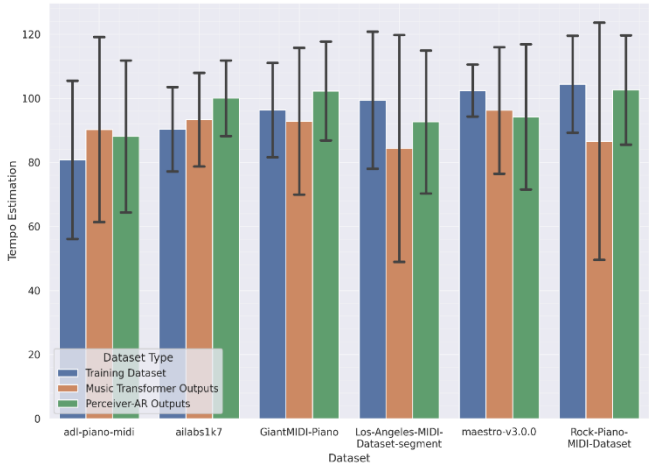

Figure 8.21: Tempo Estimation Bar-plot

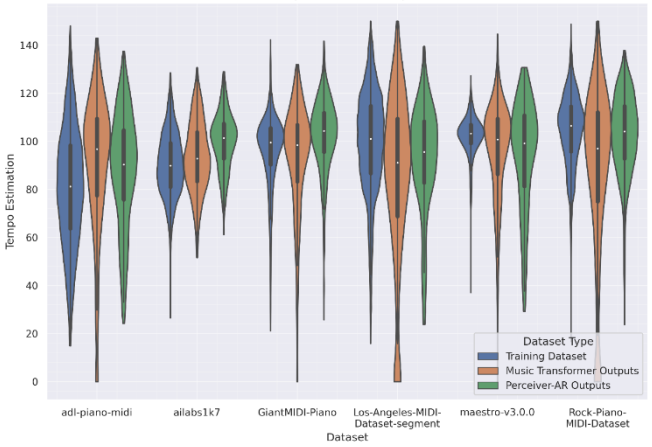

Figure 8.20: Tempo Estimation Violin-plot

Objective Metrics Analysis

We have created a summary table based on the metrics table and plots. We provide some comparative descriptions to help readers understand the degree of similarity between the output datasets and their corresponding training datasets.

If a models' output datasets are similar to the training datasets, we use the label "Similar." If there are some exceptions but the majority of the datasets are still similar to the training datasets, we use "Mostly similar." If the statistical metric shows that the output datasets are closer to the corresponding training datasets than the other models' outputs, we use "Closer." However, it's important to note that being closer doesn't necessarily mean being similar to the training datasets.

Furthermore, we want to mention that the differences in the pitches used metric may be partially due to variations in the duration of the music pieces in both the training and output datasets.

Objective Metric		**Training Datasets**	**Music Transformer Outputs**	**Perceiver-AR Outputs**
Pitch Range	**Mean**	Varying	Lower	Slightly lower, closer
	Std	Moderate to high	High	Moderate, similar
	Distribution	Symmetrical or skewed towards higher values	Wider, symmetrical towards lower values	Symmetrical, mostly similar
	Outliers	Several high value	Frequent low value	Few low and high value
Pitches Used	**Mean**	Varying	Lower	Lower, closer
	Std	Moderate to high	Moderate	Low
	Distribution	Symmetrical or skewed towards higher values	Symmetrical or skewed towards lower values	Symmetrical, more concentrated
	Outliers	Several high value	Few	Few
Pitch Classes	**Mean**	Concentrated	Significantly lower	Slightly lower, closer
	Std	Low to moderate	High	Low to moderate, similar

	Distribution	Multimodal, skewed towards high values	Wide stretching through the whole range or skewed towards high values	Multimodal or skewed towards high values
	Outliers	Several low value	Many low value	Few low value
Polyphony	**Mean**	Varying	Lower, closer	Significantly lower
	Std	High	High	Moderate
	Distribution	Concentrated, symmetrical or bimodal	Symmetrical, wider, more similar	Symmetrical, more concentrated towards lower values
	Outliers	Several high value	Few high value	Rare high value
Polyphony Rate	**Mean**	Varying	Lower, closer	Significantly lower
	Std	Moderate to high	High	High
	Distribution	Wide stretching through the whole range or skewed towards high values	Wide stretching through the whole range or skewed towards low values	Symmetrical or skewed towards low values
	Outliers	Few low value	Many low and high value	Several high value
Scale Consistency	**Mean**	Concentrated, high value	Similar	Similar, closer
	Std	Low	Moderate to high	Low, similar
	Distribution	Skewed towards high values of symmetrical	Skewed towards high values	Skewed towards high values, similar
	Outliers	Few low value	Several low value	Few low value, similar
Pitch Entropy	**Mean**	Varying	Significantly lower	Slightly lower, mostly similar
	Std	Low to moderate	Moderate to high	Low to moderate
	Distribution	Symmetrical, concentrated	Wide, symmetrical	Symmetrical, concentrated, mostly similar
	Outliers	Several low value	Many low value	Several low value, similar
Pitch Class Entropy	**Mean**	Concentrated	Significantly lower	Slightly lower, similar
	Std	Low	Moderate to high	Low, similar
	Distribution	Symmetrical, concentrated	Wide, symmetrical	Symmetrical, concentrated, similar
	Outliers	Several low value	Many low value	Several low value, similar
Empty Beat rate	**Mean**	Varying	Close	Significantly lower
	Std	Moderate	High	Low to moderate
	Distribution	Skewed towards low values	Skewed towards low values, similar	Skewed towards low values, more concentrated
	Outliers	Several high value	Several high value, similar	Few high value
Tempo Estimation	**Mean**	Concentrated	Similar	More similar
	Std	Low to moderate	Moderate to high	Moderate, similar
	Distribution	Symmetrical	Symmetrical, wider, similar	Symmetrical, more similar
	Outliers	Few mostly low value	Several low value	Few mostly low value, similar

Table 11: Objective Metrics Model Type Comparison

Implementation

The implementation for this part of our work is included in obj_eval.ipynb Notebook file. First of all, we process the midi files and calculated the music metrics for each midi song in our training and output datasets. We save the metric values on arrays and perform a check on whether we had any empty values because of corrupted midi files or failures during the processing and calculation of the metrics. We remove the substandard files, being statistically insignificant regarding our datasets. We then create a data-frame using (McKinney, 2010; Pandas Development Team, 2020) Pandas framework gathering all our data in a unified structure.

Specifically, in the first section of the code "Calculating Metrics" the basic required directories are set both for the training datasets, models outputs and the directories for saving metrics files and plots. The number of MIDI files in the input directory is counted. A NumPy (Harris et al., 2020) array called mus_metr is created to store the calculated metrics, with one dimension for indexing files and another dimension for the ten metrics parameters. Additionally, a dictionary called num_name is initialized to store the mapping between file numbers and names. Next, the MIDI files in the input directory are iterated over attempting to read the MIDI file using the PrettyMIDI and Muspy modules. The objective metrics are calculated, while error handling for corrupted MIDI files that cannot be processed is also included. Once metrics have been calculated for all files, the array mus_metr is checked for any NaN values. Any midi file with a NaN metric value is then removed from the array. The cleaned mus_metr array is then saved. This section is executed for all training and outputs datasets.

In the "Creating Dataframe" section of the code, after the objective metrics have been calculated for all the datasets, a data-frame is initiated by initializing an empty list mus_datasets and an empty data-frame mus_datasets_df. A loop is used to iterate through each dataset with the corresponding metric files (metrics_[dataset].npy, metrics_music-transformer_[dataset].npy, and metrics_perceiver-ar_[dataset].npy) being added to both mus_datasets and mus_datasets_df, with the file name removed to only retain the metric values. Several data-frames (df0 through df17) are defined based on the data in mus_datasets_df. Each data-frame corresponds to one of the eighteen training or output dataset metrics files, with the dataset name and type (training dataset or model output) added as new columns. The resulting data-frames are then concatenated into a single data-frame df. Finally, the resulting data-frame is saved.

In the "Creating Metrics Tables & Plots" section of the code, the data-frame is first being read from the pickle file. Next, descriptive statistics are being computed for the data, and separate Excel files are created for the different datasets. The statistics are being extracted from the original data-frame based on the dataset name and type of dataset, with describe() function being applied to each subset to compute the statistical measures. Following that, for each metric, two types of plots are generated: Bar-plot and violin plot, using Seaborn (Waskom, 2021). The data for each plot is the original data-frame with different x, y, and hue arguments. Finally, the plots are being saved in a specified directory.

Chapter 9: Subjective Evaluation

Subjective Evaluation Principles

(Hernandez-Olivan, Puyuelo, et al., 2022; Ji et al., 2020; L.-C. Yang & Lerch, 2020) have been our main references at this chapter.

Subjective or human evaluation is widely recognized as the most reliable way to assess the quality of generated art because the goal of these models is to create something that meets human aesthetic standards. Our music generation models are no exception to this. Human evaluation is particularly important in assessing the musical quality of the output as it can provide insights into melody, harmony, rhythm, coherence, emotional intensity, expressiveness, and style of musical outputs, identifying areas for models' improvement. Human evaluators' musical experience and expertise allow them to judge these qualities effectively.

Furthermore, as we discussed in the previous chapter and (Ji et al., 2020) point out, **there hasn't been demonstrated in relative literature any correlation between quantitative evaluation metrics and subjective evaluation metrics.** Human feedback makes the most logical and convincing post-hoc evaluation technique. On account of these facts, subjective assessment becomes an essential part of evaluating every music generation model, being an approach to follow in this work too.

The most typical method to subjectively evaluate the quality of music generation models is through a listening test. However, as noted by (Ji et al., 2020), **creating a listening test requires considering many factors, including the choice and presentation of audio samples, the listening environment, the selection of participants, and the content of the questions.** Attempting to control all of these variables in a large-scale study may be challenging, and it may be difficult to eliminate biases or recruit sufficient numbers of qualified participants. Therefore, it was important for us to eliminate methodology errors that could compromise the validity and reliability of our results, while also being mindful of the resources available to us for the assessment experiment.

One of the most important factors of a subjective listening test is quantity and diversity of subjects involved, with these having a high probability of varying greatly. As (Ji et al., 2020) argue, it is required that **1) there should be enough listening subjects with sufficient diversity to offer statistically significant results and 2) the music knowledge level of subjects is evenly distributed, including both people who lack music knowledge and people having some basic musical knowledge**. (L.-C. Yang & Lerch, 2020) draw attention to the fact that using relatively small sample size, most likely as a result of resource constraints, may raise concerns about the confidence interval's range and the study's statistical significance being a symptom of inadequate scientific rigor.

Another important factor in conducting subjective evaluation is **having absolute measurements of quality and not relying on the preference of one model over another** as (L.-C. Yang & Lerch, 2020) argue. Subjective relative metrics comparing the generated music with original corpus or music generated by other models can be beneficial in measuring relative differences or

improvements among different models. However, they have the huge drawback of absence of a standard comparison or absolute reference. Absolute measurements provide a clear and more objective standard for quality allowing for direct comparison across different models or music pieces and making it easier to identify areas for improvement in music generation models.

One of the other factors that should be considered is that **listening fatigue could increase evaluation error**, as continuous listening can make listeners produce relatively unreliable judgments. It becomes critical to keep our listening test short in order to prevent listening fatigue and guarantee the reliability of the results. The optimal duration for a listening test might vary depending on a variety of elements, including the complexity of the music, the quantity of samples, and the listening environment. However, we concluded that generally limiting the test to under 15 minutes would keep our subjects engaged and concentrated.

Moreover, **it is of foremost importance to define the questions asked clearly and specifically, offering users comprehensive explanations, to achieve a more effective evaluation.** As claimed by (Chu et al., 2022) study, musically beginner users rely heavily on comprehensive explanations for the subjective evaluation metrics and corresponding questions regarding AI-generated music in order to understand their meaning. Participants appreciate thorough descriptions of each metric and question, to understand clearly the factors being assessed, avoiding misleading and confusing expressions that would prevent completing the evaluation in the required way.

Other issues that should be taken into account as (Ji et al., 2020) report, is the environment that the listening test is conducted with a controlled environment having specific acoustic characteristics and equipment being suggested and ensuring that each subject receives the same instructions and stimuli.

One of the major and simplest approaches for assessment with absolute reference is to **conduct an adapted version of a Turing test, asking the subjects to judge whether the music is human-composed or computer-generated**. Other approaches according to (Ji et al., 2020) include **ranking a set of music samples generated by different models based on several evaluation metrics, by answering several subjective questions or expressing agreement to objective judgements**, such as how pleasant the melody is or if the music instantiate emotions, answering based on the common 5-point Likert scoring rule. We think that these approaches can be used in parallel by combining both methods of evaluation to be able to provide a more comprehensive understanding of the performance of models and the quality of the generated music.

Listening Test Metrics & Questions

For the design of the questions of our listening test we were mainly based on the proposed structure and form by (Hernandez-Olivan, Puyuelo, et al., 2022) and result findings by (Chu et al., 2022).

As discussed before, one of the main suggested approaches to follow is to distinguish our users based on their level of musical knowledge, a factor that will influence our further analysis and the

reliability of our conclusions. On account of that fact, we decided to form two categories of subjects based on different musical knowledge levels:

- **Basic Subject (Basic): People with no musical knowledge**, not being able to recognize and distinguish fundamental musical principles like melody, harmony and rhythm.
- **Experienced Subject (Pro): People with at least some musical knowledge and background,** such as average level musicians, conservatory students, music professionals etc. Basically, subjects who understand and recognize fundamental elements of a musical piece like melody, harmony, rhythm and genre.

Regarding the content of the listening test questions, we wanted to assess our models based on the most effective subjective evaluation criteria, having crucial impact on users' satisfaction with AI-generated music. First of all, we included an **overall rating metric describing how aesthetically pleasing that piece would be, summarizing the perception of the subject for that piece**. Another fundamental metric proposed in literature is **Naturalness (or Humanness). We implemented this metric through an adapted Turing test, asking whether the musical piece is human-composed or computer-generated**. These two questions form some primary understanding of the music quality and provide basic comparison reference to other music generation models.

Furthermore, **we decided to review emotion and familiarity, two subjective metrics that play an important role in the music listening experience** for people listening AI-generated music, according to (Chu et al., 2022). Emotion is a complex musical aspect being based on a combination of various musical elements, such as melody, harmony, rhythm, timbre, and dynamics. As (Chu et al., 2022) survey results demonstrate, a criterion for evaluating the emotional impact of music is important because it helps determine subject's emotional state while listening to a song. **We decided to examine the intensity of emotions of any kind generated by our music and not recognize the type of them,** simplifying the question based on this criterion. Identifying even simple emotions would require multiple questions adding unnecessary complexity to our listening test, being out of scope of our work. With regards to familiarity, described as either prior exposure, recognition, or personal experience of a subject with the musical piece, the results of (Chu et al., 2022) survey pointed out that participants tend to enjoy songs more when they are familiar with them.

Considering other metrics, **we wanted to assess our pieces regarding some fundamental music principles, strongly related to musical quality**, that would also be feasible for musically experienced users to identify. Some of the most important of these music principles that satisfy the previous criteria were **melodiousness, harmonicity, rhythmicity and genre**, being the ones, we chose to implement. As (Chu et al., 2022) point out, melodiousness is the most effective subjective metric to the overall satisfaction of users as defined in their own research, assessing both melody and harmony of a musical piece. In the context of our definitions in Basic Musical Elements, we assess these two musical principles separately, corresponding to the metrics of melodiousness and harmonicity. Rhythmicity also is established and considered as an important measure of a musical piece's quality by the subjects (Chu et al., 2022).

Regarding genre, **having trained datasets on different genres and using also primers from different genres, we wanted to determine whether genre elements would be evident and clear in**

our music outputs and to which extent possible genre interpolations would be recognizable by the listeners. However, we didn't aim our subject to identify specific genres as that would require multiple questions adding unnecessary complexity to our listening test, being out of scope of our work. On account of these, **we included a question about the extent to which listeners can identify the genre of the piece,** with the goal of simplifying the question based on this criterion.

In accordance with the musical metrics described above, we present the questions and statements for assessment in our listening test both for basic and experienced subjects in the table below.

Subject Musical Experience	Question	Music Metric	Score Range
Basic	Does this music bring emotions in your mind?	Emotion	1-5
	Have you heard similar music before?	Familiarity	1-5
	Is this music composed by human or AI?	Naturalness	AI/HC (0-1)
	Music Piece Overall Rating	Overall	1-5
Pro	Does this music bring emotions in your mind?	Emotion	1-5
	The melody progresses smoothly.	Melodiousness	1-5
	The harmony is pleasant.	Harmonicity	1-5
	The rhythm is consistent and smooth.	Rhythmicity	1-5
	Can you identify the music genre of the piece?	Genre	1-5
	Is this music composed by human or AI?	Naturalness	AI/HC (0-1)

Table 12: Subjective Evaluation Listening Test Questions

Sample Selection & Processing

In this sub-section, we will delve into the aspect of sample selection and processing for conducting our music generation survey. As described in Output Generation Chapter we generated **600 output pieces from each trained model (7,200 total pieces)**, both with primer included and without primer. Having included a metric about naturalness of musical pieces, implemented through an adapted Turing test, **our samples in the listening test had to be provided without having primers.** That can be explained as musically experienced listeners could perhaps identify primers on ai-generated music recognizing that the continuation of the piece is not the original composed so determining ai-generated music pieces and being susceptible to a huge bias.

Conducting the adapted Turing test also required adding original human compositions to our samples. **We added the same 600 original pieces we used as primers from the training datasets (7,800 total pieces)**, as the initial population of "original outputs" (as they would be referred). In that way, we had equal distribution of music pieces across all six datasets while also being able to make additional comparisons between the pieces' original continuations and our models ai-generated continuations. We trimmed the initial 512 event tokens (the primer part) from these pieces and only kept the subsequent 512 tokens, following the same procedure for generating outputs from our models.

Next, we applied filtering operations to the 7,800 outputs, which included 600 original pieces and 600 outputs from each of the 12 trained models. **We eliminated any pieces shorter than 10 seconds or longer than 60 seconds, resulting in 6,106 pieces remaining**. The lower duration limit ensures our survey participants engage with musical pieces long enough to be able to evaluate them, while the upper limit keeps the survey completion time relatively short and ensures avoiding listening fatigue with the corresponding increasing evaluation error.

Furthermore, **we performed random sampling selecting 48 pieces from each model** with the number of pieces having primer from each dataset being equally distributed to 8. **Regarding the training datasets we selected 16 samples from each dataset original pieces**. Concluding, we have **672 total samples with 576 total output samples from all our models and 96 original samples**, with the percentage of original samples out of the total number being approximately 14.29%.

Finally, we converted our MIDI samples subset to the MP3 audio format. MIDI is not a universal audio format and requires specific software or plugins to play, and as a result **the playback of MIDI files on the listening test page would depend on the browser audio conversion and sound-font**. Different browsers could have different audio plugins or sound-fonts, which could result in our MIDI files being played differently. This could add unwanted variability to our study and bias to our results, as different subjects would have different stimuli during the listening test. Conversely, MP3 is a universally supported audio format not requiring any additional plugins. By converting our MIDI files to MP3 format, we could ensure that all of our participants had consistent listening stimuli.

Listening Test Page

To address the challenge of evaluating the large number of AI-generated music samples, the web page https://ai-music-generation-survey.vercel.app/ was created in the form of an online survey. **The page enables users to evaluate a small set of tracks, 6 in particular**, making the process more manageable and less time-consuming. **The number of 6 samples was selected to keep the survey completion time relatively short, approximately 5 minutes.**

First of all, the page welcomes the user to the survey, explaining about its aim and providing some initial guidance for users' participation. It also informs the participants about the expected completion time. The next step of the survey aims to **assess the user's musical expertise, in order to be assigned in one of the two groups**. The user is asked whether it has ever played a musical instrument or studied at a music conservatory, to determine its musical experience. They are also asked to identify if a key signature in a piece of music is in major on minor scale. This **simple question of fundamental musical knowledge is a small test so that we can assess the credibility of users claiming to be musically experienced.** Of course, If they do not answer the question correctly, they are classified as "basic" users.

The final introductory step of the page provides some more detailed descriptions and guidance to the users about the survey's main page. It includes a thorough description of the question about emotion, explaining clearly the factor being assessed, as from some early hands-on experience with the survey, the objective of this question without proper explanation was unclear to the users. Finally, it informs about some elements of the main page such as that the progress regarding the

completion of the survey will be displayed at the top of the page and that also each composition will be looped after finishing.

Regarding the main page, **the user is presented the questions and statements being assessed, according to the musical knowledge group she has been allocated**. To procced to the next musical piece, **she has to listen to a proportion of the piece above 80% and also complete all the questions. These requirements are included as baseline measures of ensuring reliability of our user evaluations, avoiding malicious users, automated or incomplete responses.** After the submission of the evaluation of each piece the progress bar is updated. The introductory modals can be always accessed by clicking the information button, in case the user needs to review our guidance. After finishing the required number of 6 pieces, the final modal appears thanking users for their participation providing them with the options to continue evaluating even more songs or to close the survey page.

Regarding the choice of online survey for our listening test, we aimed to implement a convenient and accessible platform for participants to provide their feedback, which also saving significant amount of manpower compared to traditional in-person testing. However as (Ji et al., 2020) points out, this method of evaluation also comes with some drawbacks, particularly in terms of ensuring the authenticity, validity and reliability of the results. Since participants can complete the task from the comfort of their own homes, there is a risk that they may hastily complete the task without giving proper consideration to their answers. To minimize this factor, we introduced the musical knowledge evaluation question, the requirements for submitting each evaluation and didn't provide any reward for participating. This way users would be motivated to participate with sincere intentions carrying out the evaluation more thoughtfully and responsibly. Despite the above possible limitations, online evaluation provided us with several benefits, such as to attract a larger pool of participants, making it easier to reach a wider audience. Additionally, it was the only cost-effective option in accordance with our resources. Searching and recruiting users was done exclusively through social media not using platforms such as Amazon Mechanical Turk.

Subjective Metrics Statistics

We obtained a total of 1,380 evaluations between January 21, 2023, and March 1, 2023, consisting of 674 evaluations from basic subjects and 706 evaluations from pro subjects. Concerning the Naturalness metric, basic subjects exhibited an accuracy of 0.556, whereas pro subjects an accuracy of 0.626.

For the subjective statistical analysis of our training and output datasets we generated a descriptive statistic table for each subject group including each metric and dataset combination. Our results for each metric included mean and standard deviation (std). The count column represents the total number of evaluations received for each model-dataset combination.

Basic Subject

	Dataset Type	Dataset	Count	Familiarity	Emotion	Naturalness	Rating
mean	Training Dataset	adl-piano-midi	24	3.00	2.79	0.38	2.71

std				1.02	1.02	0.49	0.91
mean	Music Transformer Outputs	adl-piano-midi	47	2.79	2.89	0.32	2.74
std				1.21	1.26	0.47	1.19
mean	Perceiver-AR Outputs	adl-piano-midi	51	3.08	3.20	0.47	3.00
std				1.16	1.22	0.50	0.96
mean	Training Dataset	ailabs1k7	15	3.13	3.20	0.47	3.33
std				0.92	1.01	0.52	1.11
mean	Music Transformer Outputs	ailabs1k7	49	3.16	3.06	0.45	2.96
std				1.28	1.07	0.50	1.14
mean	Perceiver-AR Outputs	ailabs1k7	51	3.27	3.35	0.49	3.06
std				1.06	1.04	0.50	0.81
mean	Training Dataset	GiantMIDI-Piano	16	3.00	2.56	0.44	2.75
std				1.26	0.89	0.51	1.00
mean	Music Transformer Outputs	GiantMIDI-Piano	50	3.24	3.10	0.48	3.12
std				1.04	1.04	0.50	0.96
mean	Perceiver-AR Outputs	GiantMIDI-Piano	43	2.98	3.07	0.44	2.79
std				1.03	1.01	0.50	1.04
mean	Training Dataset	Los-Angeles-MIDI-segment	15	3.20	3.00	0.33	2.67
std				1.08	1.13	0.49	1.29
mean	Music Transformer Outputs	Los-Angeles-MIDI-segment	45	3.16	3.09	0.47	3.04
std				1.15	1.24	0.50	1.09
mean	Perceiver-AR Outputs	Los-Angeles-MIDI-segment	46	2.89	2.89	0.37	2.76
std				1.25	1.18	0.49	1.06
mean	Training Dataset	maestro-3.0.0	13	3.31	2.92	0.54	2.92
std				1.25	1.44	0.52	1.38
mean	Music Transformer Outputs	maestro-3.0.0	50	2.90	2.80	0.48	2.90
std				1.16	1.09	0.50	1.09
mean	Perceiver-AR Outputs	maestro-3.0.0	46	2.85	2.85	0.30	2.67
std				1.01	0.87	0.47	0.87
mean	Training Dataset	Rock-Piano-MIDI	14	3.29	3.00	0.50	3.14
std				1.07	1.36	0.52	0.95
mean	Music Transformer Outputs	Rock-Piano-MIDI	43	3.14	3.14	0.40	2.86
std				0.97	1.01	0.49	1.06
mean	Perceiver-AR Outputs	Rock-Piano-MIDI	56	3.04	3.00	0.39	3.04
std				1.16	1.06	0.49	1.09

Table 13: Basic Subject Subjective Metrics Statistics

Pro Subject

	Dataset Type	Dataset	Count	Emotion	Melodiousness	Harmonicity	Rhythmicity	Genre	Naturalness	Rating
mean	Training Dataset	adl-piano-midi	13	3.08	2.92	3.15	2.62	3.00	0.38	2.85
std				1.26	1.38	1.34	1.19	1.47	0.51	1.34
mean	Music Transformer Outputs	adl-piano-midi	41	3.24	3.10	3.15	3.15	3.00	0.32	3.05
std				1.22	1.34	1.28	1.37	1.20	0.47	1.26
mean	Perceiver-AR Outputs	adl-piano-midi	56	3.38	3.13	3.23	3.05	2.82	0.45	3.14
std				1.29	1.36	1.31	1.35	1.34	0.50	1.24
mean	Training Dataset	ailabs1k7	24	3.88	3.33	3.42	3.58	3.42	0.42	3.42
std				1.03	1.17	1.10	1.21	0.97	0.50	1.21

mean	Music Transformer Outputs	ailabs1k7	48	3.40	3.17	3.29	3.15	2.90	0.25	3.04
std				1.14	1.15	1.29	1.38	1.28	0.44	1.20
mean	Perceiver-AR Outputs	ailabs1k7	53	3.28	2.92	3.13	2.94	2.72	0.32	3.00
std				1.29	1.34	1.37	1.38	1.31	0.47	1.32
mean	Training Dataset	GiantMIDI-Piano	13	3.69	3.08	3.62	3.00	2.92	0.46	3.23
std				0.85	1.19	1.19	1.29	1.50	0.52	1.01
mean	Music Transformer Outputs	GiantMIDI-Piano	55	3.07	2.82	3.11	2.78	2.96	0.29	2.82
std				1.14	1.28	1.20	1.27	1.14	0.46	1.20
mean	Perceiver-AR Outputs	GiantMIDI-Piano	51	3.18	2.82	3.06	2.96	2.73	0.24	2.90
std				1.21	1.32	1.27	1.25	1.15	0.43	1.20
mean	Training Dataset	Los-Angeles-MIDI-segment	22	3.05	2.64	2.50	2.55	2.59	0.32	2.64
std				1.21	1.18	1.14	1.26	1.14	0.48	1.26
mean	Music Transformer Outputs	Los-Angeles-MIDI-segment	44	3.43	3.05	3.18	3.02	2.68	0.30	3.07
std				1.21	1.28	1.32	1.37	1.43	0.46	1.35
mean	Perceiver-AR Outputs	Los-Angeles-MIDI-segment	53	3.45	3.19	3.25	3.19	2.92	0.40	3.06
std				1.05	1.14	1.16	1.29	1.24	0.49	1.22
mean	Training Dataset	maestro-3.0.0	20	3.15	3.15	3.15	2.95	2.55	0.35	2.95
std				1.27	1.46	1.35	1.47	1.50	0.49	1.39
mean	Music Transformer Outputs	maestro-3.0.0	50	3.12	2.74	2.88	2.84	2.94	0.30	2.84
std				1.02	1.17	1.15	1.23	1.17	0.46	1.22
mean	Perceiver-AR Outputs	maestro-3.0.0	48	3.33	3.08	3.19	3.13	2.77	0.44	2.98
std				1.12	1.32	1.25	1.20	1.13	0.50	1.19
mean	Training Dataset	Rock-Piano-MIDI	14	3.14	3.29	3.36	3.14	3.07	0.29	3.07
std				1.35	1.20	1.28	1.46	1.21	0.47	1.27
mean	Music Transformer Outputs	Rock-Piano-MIDI	48	3.21	2.90	2.92	2.88	2.94	0.25	2.75
std				1.05	1.32	1.16	1.28	1.31	0.44	1.38
mean	Perceiver-AR Outputs	Rock-Piano-MIDI	53	3.53	3.25	3.40	3.19	2.81	0.38	3.17
std				1.12	1.25	1.12	1.29	1.30	0.49	1.25

Table 14: Pro Subject Subjective Metrics Statistics

Subjective Metrics Weighted Mean across Dataset-Models

For an overall comparison of the training-outputs datasets subjective metrics performance, we present the table of weighted mean values of all metrics scaled to Likert 1-5 scale. We have excluded Genre metric as it is not applicable in multi-genre datasets. The "Weighted Mean" column takes into account the percentage of responses from each subject group and the number of metrics each subject group evaluates. This means that each metric for each subject group is given equal importance. In addition, in the final column, the weighted mean of model outputs also incorporates the percentage of responses from each model type.

Dataset Type	Dataset	Weighted Mean: Basic	Weighted Mean: Pro	Weighted Mean	
Training Dataset	adl-piano-midi	2.75	2.86	**2.80**	
Music Transformer Outputs	adl-piano-midi	2.68	2.99	2.85	**2.98**

Perceiver-AR Outputs	adl-piano-midi	3.04	3.12	3.09	
Training Dataset	ailabs1k7	3.13	3.38	**3.31**	
Music Transformer Outputs	ailabs1k7	2.99	3.01	3.00	**3.01**
Perceiver-AR Outputs	ailabs1k7	3.16	2.93	3.02	
Training Dataset	GiantMIDI-Piano	2.77	3.24	**3.03**	
Music Transformer Outputs	GiantMIDI-Piano	3.10	2.79	2.91	**2.88**
Perceiver-AR Outputs	GiantMIDI-Piano	2.90	2.81	2.84	
Training Dataset	Los-Angeles-MIDI-segment	2.80	2.61	**2.67**	
Music Transformer Outputs	Los-Angeles-MIDI-segment	3.04	2.99	3.01	**3.00**
Perceiver-AR Outputs	Los-Angeles-MIDI-segment	2.76	3.12	2.99	
Training Dataset	maestro-3.0.0	3.08	2.96	**2.99**	
Music Transformer Outputs	maestro-3.0.0	2.88	2.77	2.81	**2.86**
Perceiver-AR Outputs	maestro-3.0.0	2.65	3.08	2.91	
Training Dataset	Rock-Piano-MIDI	3.11	3.02	**3.06**	
Music Transformer Outputs	Rock-Piano-MIDI	2.93	2.77	2.83	**2.96**
Perceiver-AR Outputs	Rock-Piano-MIDI	2.91	3.17	3.06	

Table 15: Subjective Metrics Weighted Means across Dataset-Models

Subjective Metrics Correlations

We have also included the subjective metrics Kendall's correlation heatmap to measure monotonic relationships between the subjective metrics. The reasons for using Kendall's correlation are similar to that mentioned in the Objective Metrics Correlations section.

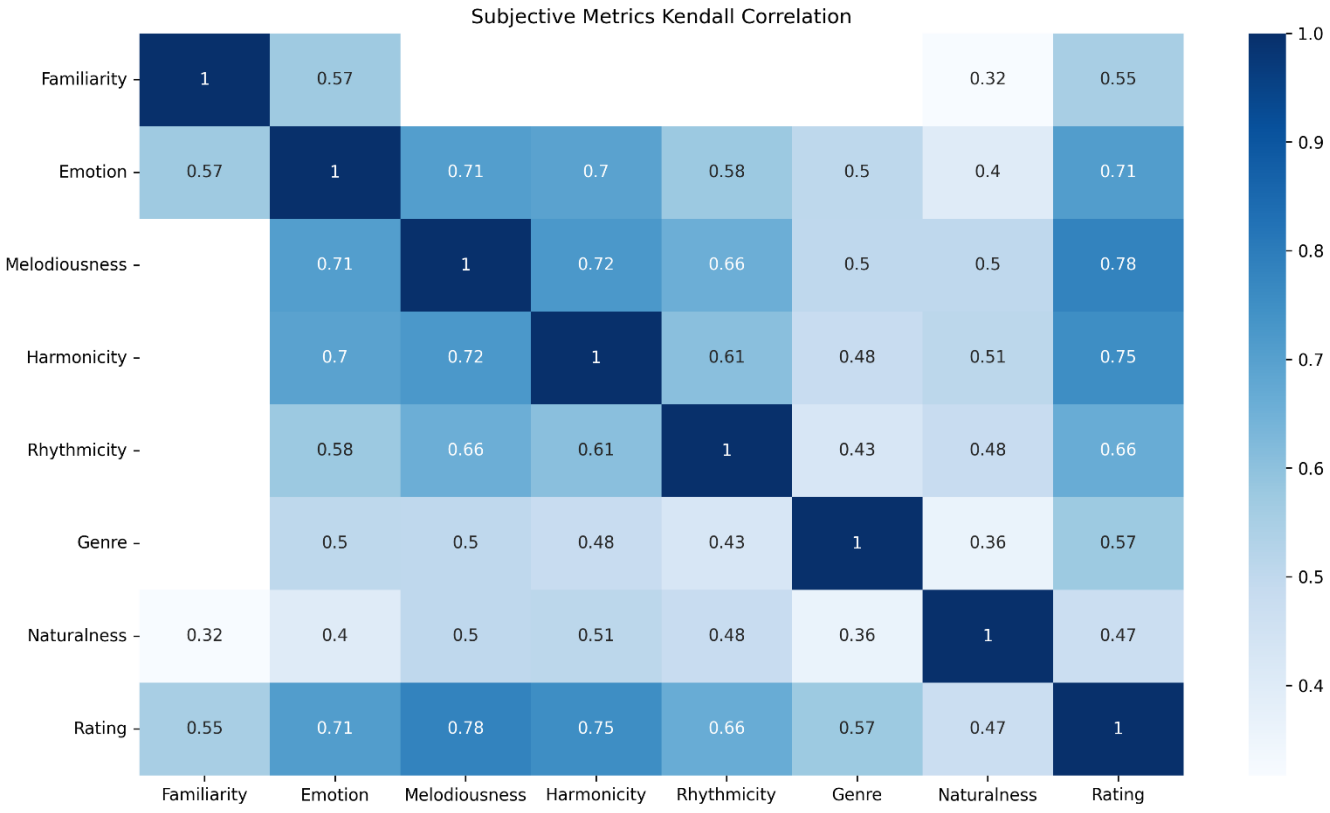

Figure 9.1: Subjective Metrics Kendall Correlation Heatmap

Subjective Metrics Plots

After calculating the statistical values for our objective metrics, we wanted to compare the mean values of all the datasets in a graphical way. For this purpose, we chose the bar-plot graph. As error calculation method we used standard deviation (std) error. In the following pages we present seven categories of plots differentiated by the specific focus of their analysis.

Subjective Metric across Datasets & Dataset Types

Basic Subject

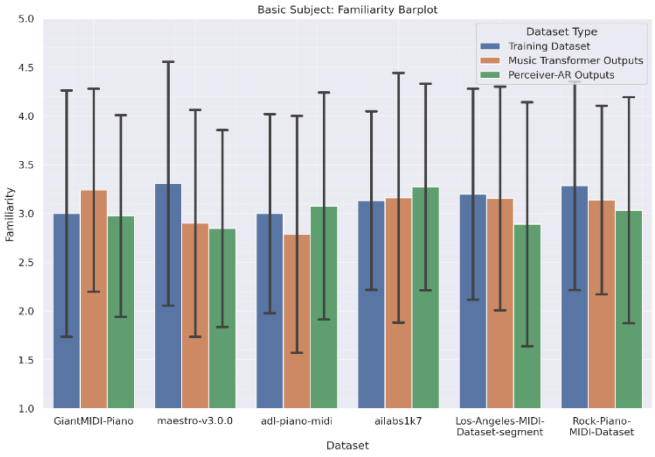

Figure 9.3: Basic Subject – Familiarity Bar-plot

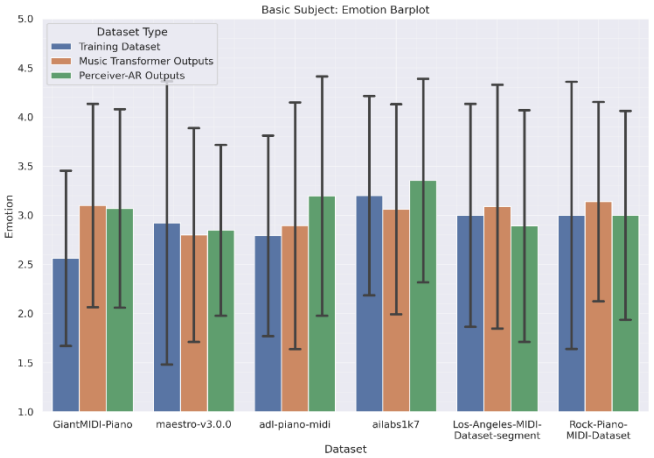

Figure 9.2: Basic Subject - Emotion Bar-plot

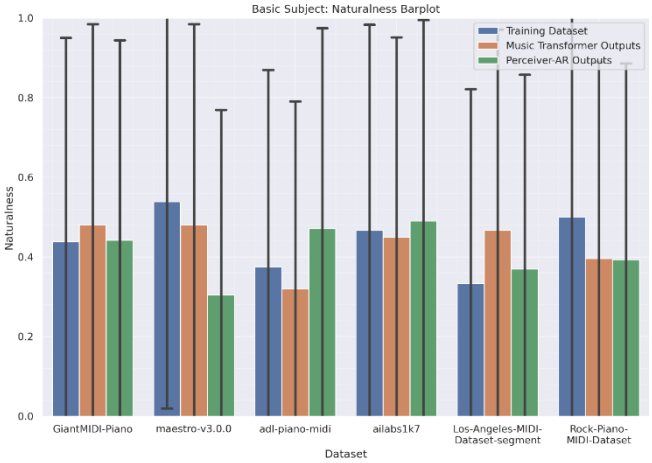

Figure 9.5: Basic Subject - Naturalness Bar-plot

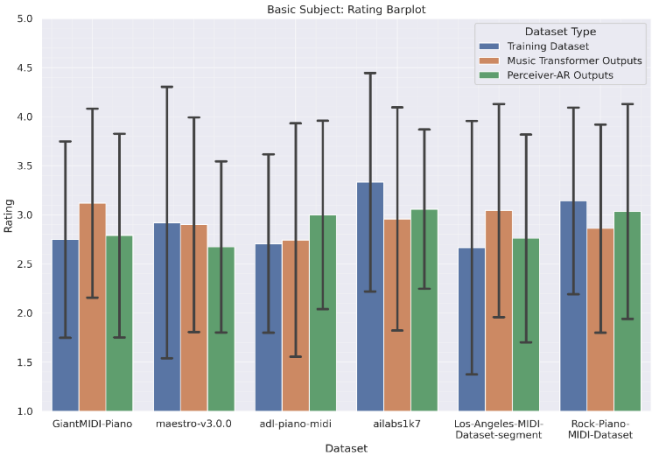

Figure 9.4: Basic Subject - Rating Bar-plot

Pro Subject

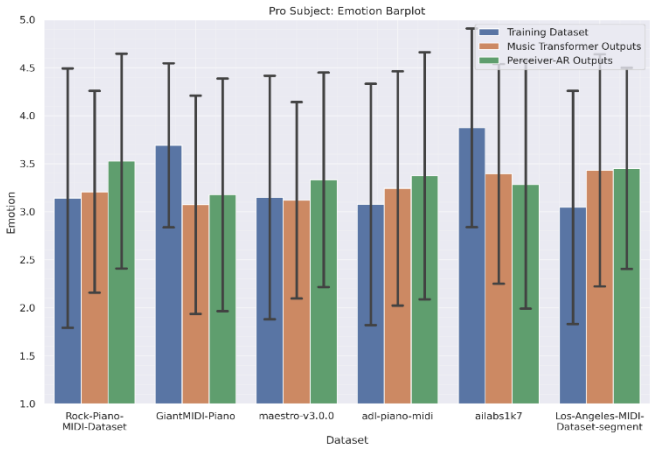

Figure 9.7: Pro Subject - Emotion Bar-plot

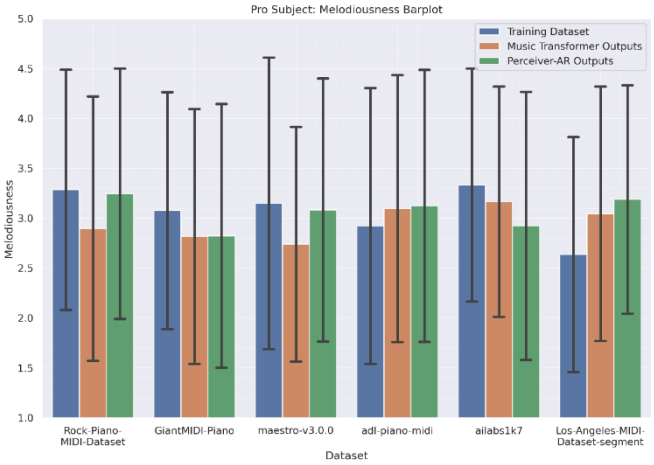

Figure 9.6: Pro Subject - Melodiousness Bar-plot

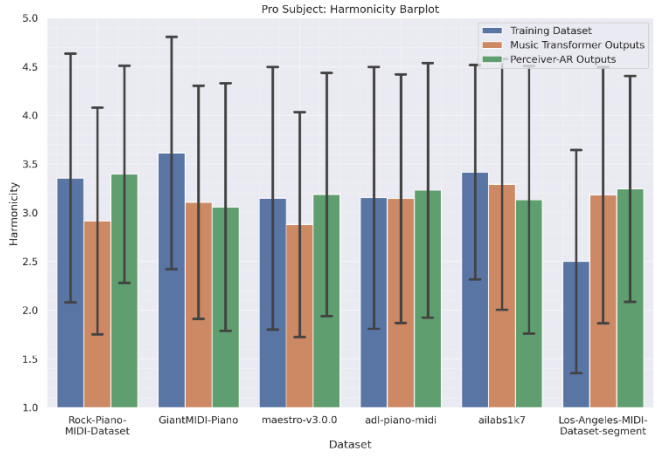

Figure 9.9: Pro Subject - Harmonicity Bar-plot

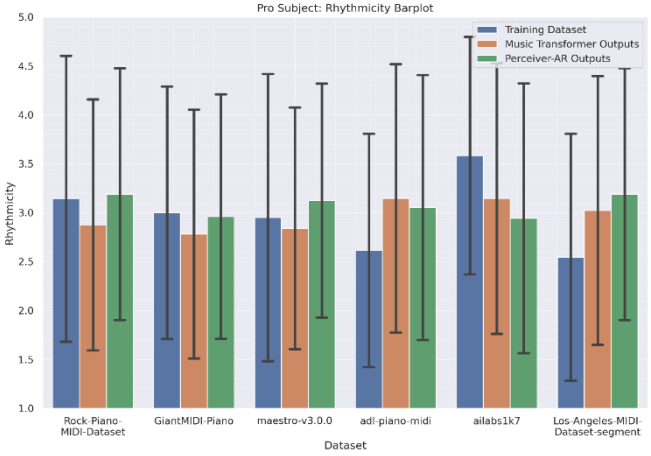

Figure 9.8: Pro Subject - Rhythmicity Bar-plot

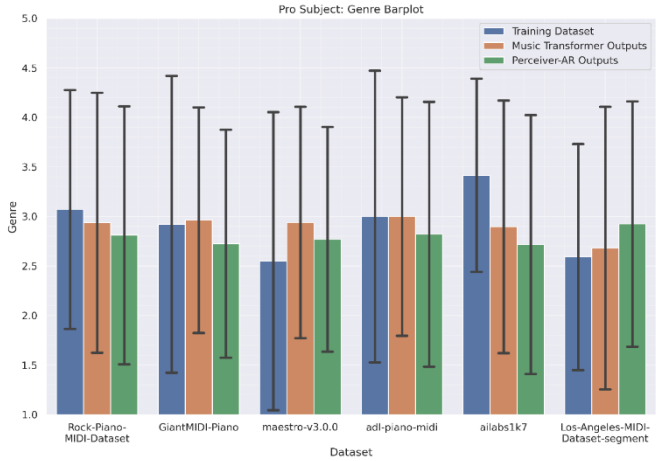

Figure 9.11: Pro Subject - Genre Bar-plot

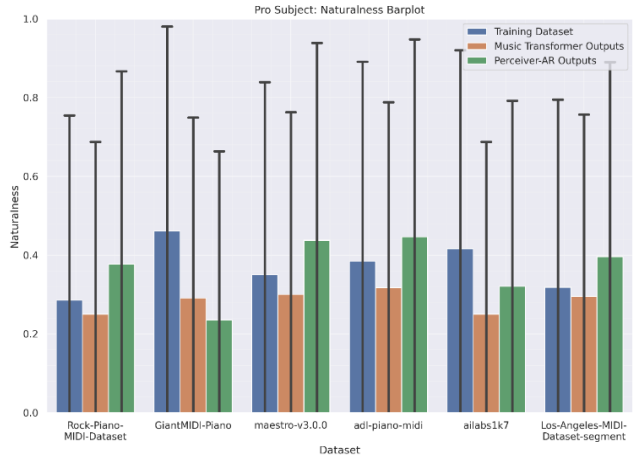

Figure 9.10: Pro Subject - Naturalness Bar-plot

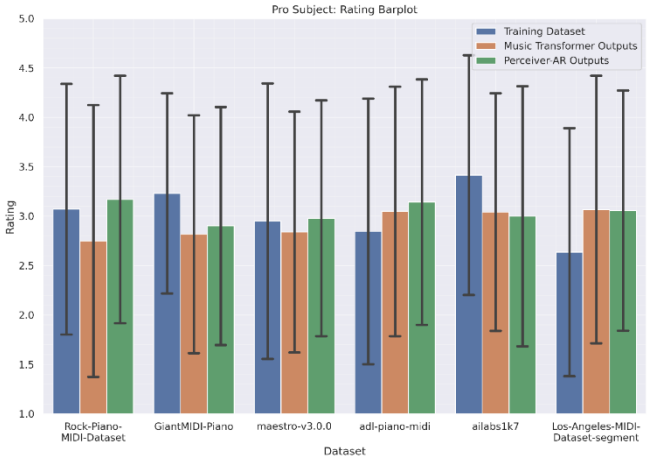

Figure 9.12: Pro Subject - Rating Bar-plot

Subjective Metrics across Dataset Types

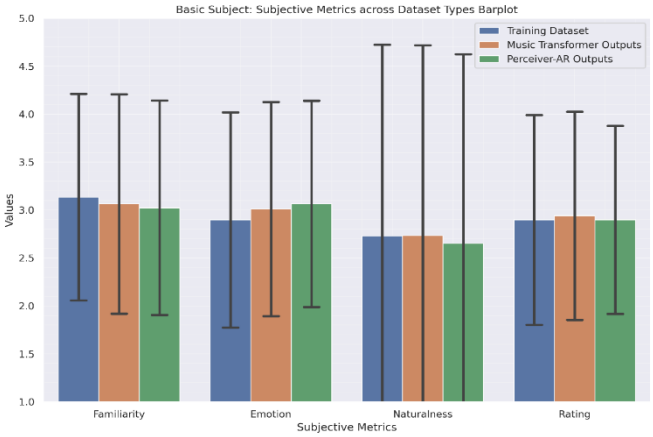

Figure 9.14: Basic Subject - Subjective Metrics across Dataset Types Bar-plot

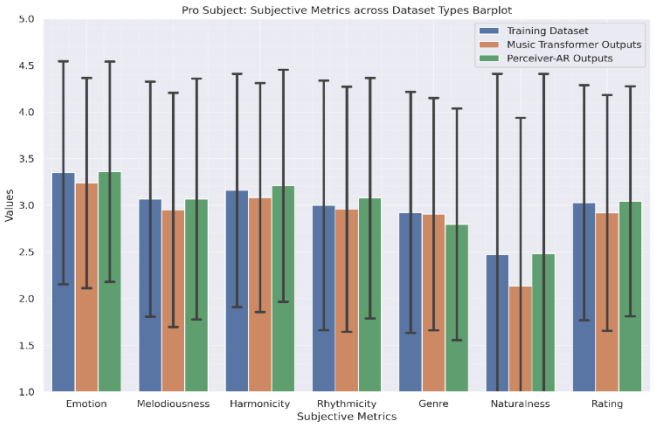

Figure 9.13: Pro Subject - Subjective Metrics across Dataset Types Bar-plot

Subjective Metrics across Datasets

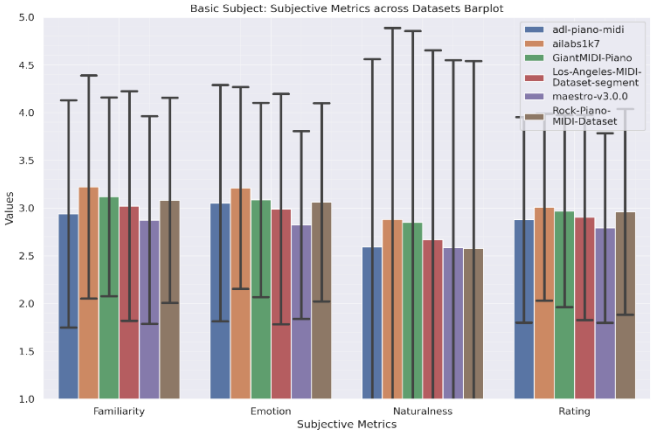

Figure 9.16: Basic Subject - Subjective Metrics across Datasets Bar-plot

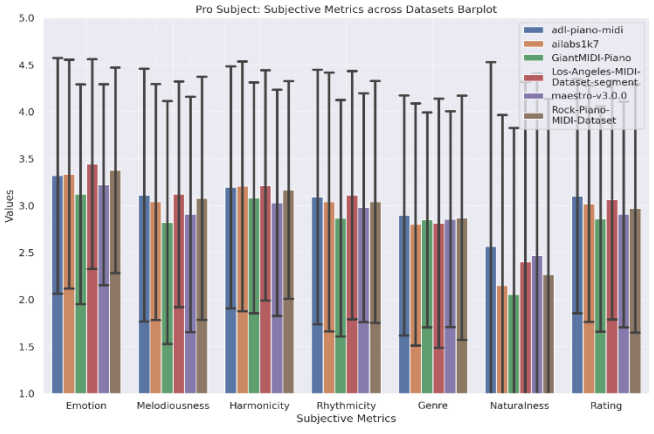

Figure 9.15: Pro Subject - Subjective Metrics across Datasets Bar-plot

Subjective Metrics across Primer Datasets

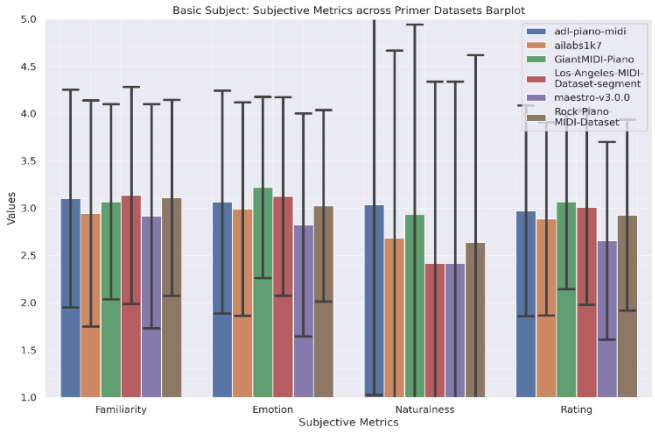

Figure 9.18: Basic Subject - Subjective Metrics across Primer Datasets Bar-plot

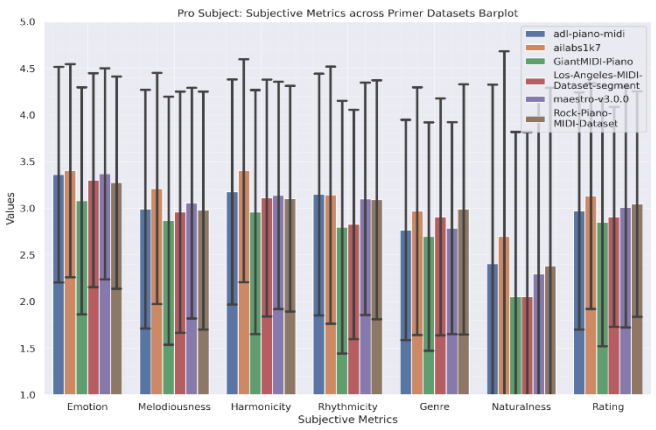

Figure 9.17: Pro Subject - Subjective Metrics across Primer Datasets Bar-plot

Dataset Types across Subjective Metrics

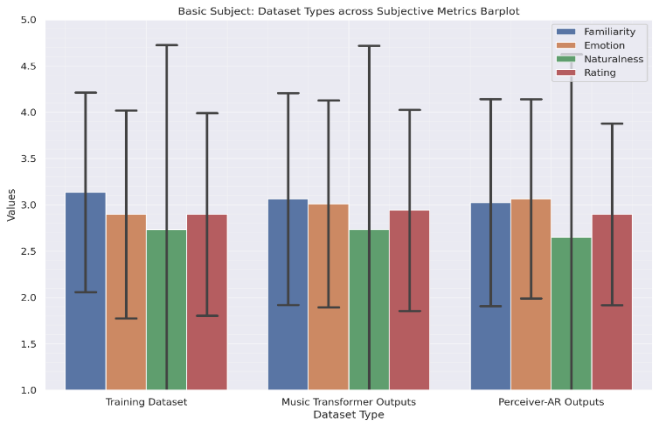

Figure 9.20: Basic Subject - Dataset Types across Subjective Metrics Bar-plot

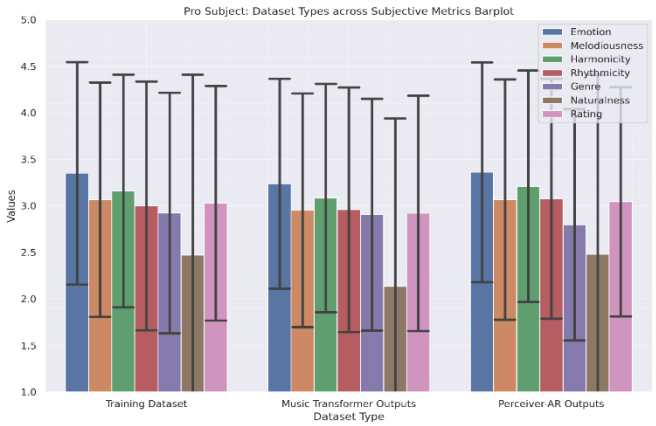

Figure 9.19: Pro Subject - Dataset Types across Subjective Metrics Bar-plot

Datasets across Subjective Metrics

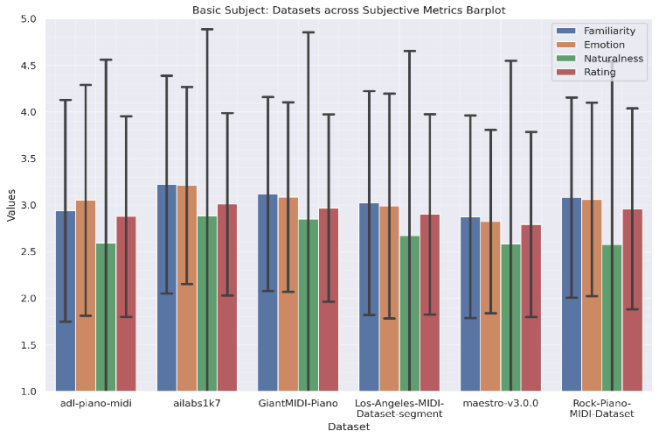

Figure 9.22: Basic Subject - Datasets across Subjective Metrics Bar-plot

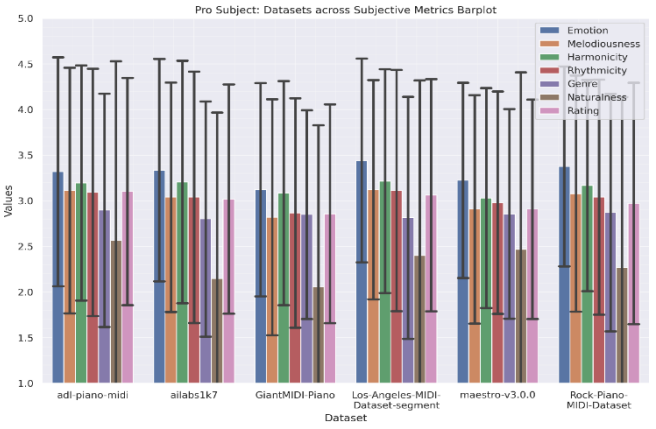

Figure 9.21: Pro Subject - Datasets across Subjective Metrics Bar-plot

Primer Datasets across Subjective Metrics

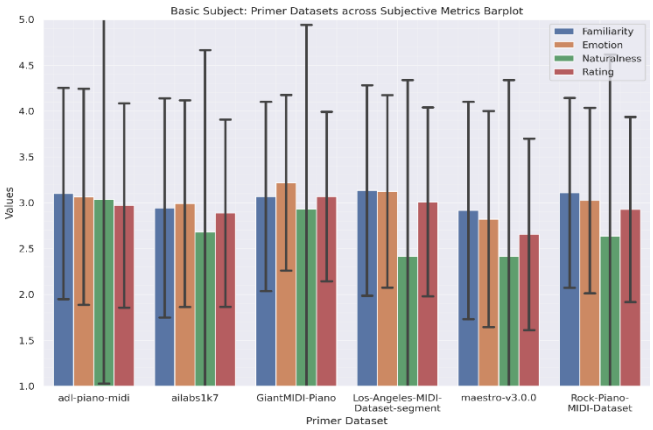

Figure 9.24: Basic Subject - Primer Datasets across Subjective Metrics Bar-plot

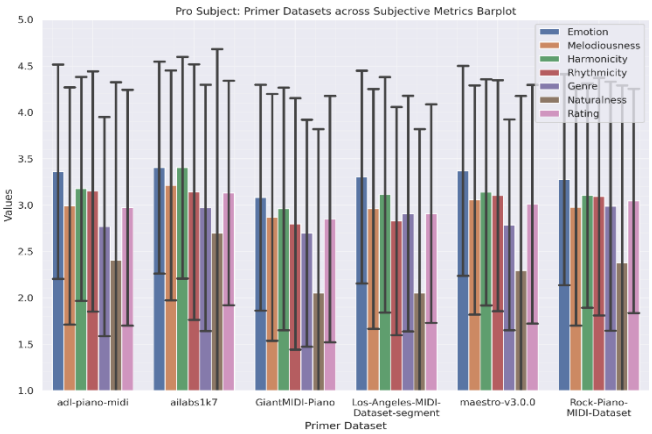

Figure 9.23: Pro Subject - Primer Datasets across Subjective Metrics Bar-plot

Implementation

Sample Selection & Processing

This part of our work is included in sub_eval.ipynb Notebook file “Sample Selection & Processing” cells. At first, the number of MIDI files in the midi output directories is counted and any files whose length is less than 10 or greater than 60 seconds are filtered. Next, the specified number of MIDI files is selected from the list of MIDI output directories. For each trained model directory, 48 files are randomly selected, with the number of pieces having primer from each dataset being equally distributed to 6. Each selected file is checked that it can be processed using the pretty_midi (Raffel & Ellis, 2014) and mido (Bjørndalen, 2023) libraries.

Next, the MIDI samples subset is converted to the MP3 audio format. Firstly, the files are encoded as mp3 with MusicBee application using the format option and GeneralUser GS v1.471 midi soundfont. The encoding is done with the minimum file size option, a bit rate of 87.0 kb/s and a sampling rate of 48.0 kHz. Next, because the total size of the files is relatively big for our data storage, the files are further compressed to a bit rate of 32.0 kb/s and a sampling rate of 16.0 kHz using the FormatFactory application.

Listening Test: Page

The listening test page is https://ai-music-generation-survey.vercel.app/ and its code is included in https://github.com/aggelostais/ai-music-generation-survey GitHub repository. It was built using code from (Filandrianos, 2019) further developed to include additional features and functionalities. The page is implemented using the Flask web framework in Python. The music samples and ratings collected are stored in Firebase Storage. Finally, the survey is deployed on Vercel as a Serverless Function.

The main flask application is included in ‘index.py’. Several routes that respond to HTTP requests are defined, and Firebase Storage is used to store and retrieve the rating files. Concurrency is managed using threading. The Firebase app and storage are initialized using the configuration data. The base route ('/') is called when a user navigates to the root URL, and it renders the 'vote.html' template. The function of '/get_melodies' route reads the list of samples file and returns it. The function of '/download_sample' route retrieves a sample filename and returns the URL link to download the sample from Firebase Storage. The route '/insert' function retrieves the required rating pickle file, saves the rating and stores the rating file back in Firebase Storage. It also includes some data validation and concurrency management. The app.run() function starts the Flask server.

The main page template is included in ‘/templates/vote.html’ file. The main body element contains the listening test title, the progress bar, and the listening test questions. The questions are in the form of radio buttons with corresponding images being the 5-point Likert scale. The play button is placed above the questions and when clicked, plays the randomly selected sample.

Three introductory modal dialog windows and one final modal are also included. The first modal dialog window "myModal1" contains a welcome message and is triggered when the page is first loaded. The second modal dialog window "myModal2" includes the questionnaire about the user musical background. The third modal "myModal3" is the final introductory modal containing some further survey guidance and the "Start" button to begin the survey. The last modal defined "myModal4" is displayed after the user has completed the survey, containing the thank you message, additional information about the project, "Finish" button that closes the survey window and "Continue" button that closes the modal to continue the survey.

Next, a JavaScript script is defined including the rating and musical knowledge variables and functions. Some of these are the AJAX calls to retrieve samples list, the functionality of modal windows buttons and helper functions for playing the music sample, checking user inputs, selecting radio buttons, etc. These functions are responsible for updating the progress bar, displaying the corresponding images of the Likert scale, and playing the samples.

The ‘/static/js/vote.js’ script consist of several additional functions that control the user interface of the survey page. The images and radio buttons are manipulated according to the selected scale values and the states of the HTML elements are changed by these functions. The make_enable() function, which is called every second by the setInterval() method, checks if 80% of a song has been played and if all the required questions have been completed. If all of these conditions are met, then the Next button for submitting the evaluation is enabled. When the Next button is clicked, the insert() function is called. If all the required fields have been filled, an AJAX request is sent to the server to save the user's single piece evaluation. If the request is successful the playbtn() function is called playing the next sample, and an AJAX request is sent to the server to download the next song. Then, the initial variables and the rating fields are reset. Finally, the current question number is incremented, and the progress bar is updated.

Listening Test: Data Storage

In sub_eval.ipynb Notebook file “Listening Test Page” cells, the basic and pro ratings are created as python dictionaries. Next, for all MP3 samples a key-value pair is created in both basic and pro

dictionaries, where the key is the sample name and the value is a list of 8 empty sub-lists, representing the subjective metrics. After all the MP3 files have been processed, the basic and pro ratings dictionaries are written to two pickle files, basic.pkl and pro.pkl.

A project is created in Firebase and Firebase Storage is used to store the ratings pickle files and the MP3 music samples. Reading and writing is set to be allowed in the security rules section. The Firebase configuration details are extracted, creating a config.json configuration file that is used by the main flask application in 'index.py'.

Listening Test: Deployment

The survey is deployed on Vercel using the (*Flask Hello World – Vercel*, 2023) template and the Web Server Gateway Interface (WSGI) with Flask to enable handling requests on Vercel with Serverless Functions.

Subjective Metrics Statistics, Correlation & Plots

This part of our work is included in sub_eval.ipynb Notebook file "Subjective Metrics Statistics", "Subjective Metrics Plots" and "Subjective & Objective Metrics Correlations" sections.

In "Subjective Metrics Statistics", the rating files are loaded and relevant information such as sample name, dataset type, dataset name, number, and ratings are extracted and stored in several lists. The extracted data is then used to create a pandas DataFrame. Next, the accuracy for Naturalness metric of each subject group is calculated by dividing the correctly rated samples by the total number of samples for each subject group. Then, for each dataset name, the original dataframe is split according to the three different type of datasets Training Dataset, Music Transformer Outputs and Perceiver-AR Outputs as well as according to the musical knowledge groups. Descriptive statistics are calculated for each split dataframe and some irrelevant columns are removed. Finally, the resulting statistics dataframes are written to two Excel files, "subjective_metrics_basic.xlsx" and "subjective_metrics_pro.xlsx".

In "Subjective Metrics Plots", the ratings dataframe is divided into two dataframes based on the subject group. Then, for each metric in each subject group, a bar plot is created, 11 in total. For the next plots, the "Naturalness" column is scaled between 1-5. Then, another 12 plots are created with 2 plot types for each subject group: all metrics for each dataset type, all dataset types for each metric, all metrics for each dataset, all datasets for each metric, all metrics for each primer dataset and all primer datasets for each metric. All graphs are created using Seaborn (Waskom, 2021).

In "Subjective & Objective Metrics Correlations", after making a copy of both subjective and objective evaluation dataframes, the correlations for each dataframe's columns are computed using the Kendall method. For the Objective-Subjective metrics correlations, the two dataframes are merged on common columns and the correlation matrix is computed for the merged dataframe using the Kendall method. To visualize the correlation matrices, heatmaps are generated using Seaborn (Waskom, 2021).

Part III: Conclusions & Discussion

Chapter 10: Conclusions

Training Datasets

Through the training and evaluation process, we were able to assess the training datasets on several metrics and identify specific attributes and characteristics of the datasets that had a considerable impact on the models' training process and the quality of generated musical outputs. Though our coverage regarding the determining factors of these derived conclusions and evaluation values cannot be exhaustive.

Models Training

Ailabs1k7 and maestro 3.0.0 models converge more stably through training compared to other datasets models, having less fluctuations in training and evaluation loss. This fact can be explained for a number of reasons including that these datasets are the smallest ones. Also, they are probably more coherent than the others only representing one musical genre with each dataset's music pieces being more musically close to the others of that dataset having less variations. This is much a more challenging condition for bigger datasets or datasets spreading through multiple genres.

Rock MIDI Piano models primarily and Los Angeles MIDI segment models secondly perform more unstably through the training process compared to the other datasets models, having more fluctuations in training and evaluation loss. This fact can be explained for a number of reasons including that these datasets are less coherent and calibrated or a bit less representative for our problem compared to the other datasets or include more noise factors, for example more empty music measures.

Smaller datasets, or in other words datasets with smaller number of training batches, convert in smaller numbers of training steps compared to datasets with bigger numbers of training batches. The difference is much more unambiguous on Music Transformer models with the smaller datasets ailabs1k7 and maestro 3.0.0 converging in relatively low number of steps. However, it is also noticeable on Perceiver-AR models too with these two datasets models converging in lower number of steps than any other dataset model.

Subjective Evaluation

Regarding Familiarity, training datasets exhibited moderate values mainly above the middle, failing to achieve higher values. These values can be attributed to multiple factors. Firstly, **datasets comprising a considerable number of incomplete or messy pieces with discontinuities, long pauses or noise influenced familiarity evaluation**, as subjects would compare lower-quality pieces they could encounter to virtuoso piano tracks they had experienced before. Furthermore, **in terms of single-genre training datasets it was expected that users would be more familiar with the musical pieces**, as these genres, classical, pop, and rock, are widely recognized. Contending this fact, the two highest values were observed in the classical maestro-3.0.0 and Rock-Piano-MIDI. **Conversely, datasets with mixed genres exhibited lower familiarity values as they also included**

unfamiliar genres, such as folk, reggae, religious, soundtracks etc., with adl-piano-midi dataset having the worst evaluation on this metric.

Regarding Emotion, training datasets showed moderate values around the middle. These values can partially be attributed to the fact that training datasets were not specifically designed for generating emotionally rich music, with the exception of ailabs1k7 which had better values in the pop genre with more emotional content. Conversably, adl-piano-midi dataset demonstrated the poorest performance specifically on this metric, with the presence of multiple music genres making it more difficult to produce good results.

In terms of Melodiousness, training datasets values were moderate, mostly above middle, though without achieving desired higher values. One of the contributing factors for this could be the presence of more melodically incomplete pieces in some training datasets, for example in the Los-Angeles-MIDI-segment which had significantly lower ratings on Melodiousness compared to other training datasets, such as ailabs1k7 and Rock-Piano-MIDI which had more preferable values.

Regarding Harmonicity, training datasets values were satisfactory and above the middle in most cases, though not achieving higher values. Training datasets that contained more lower quality harmonically inconsistent pieces, such as the Los-Angeles-MIDI-segment, produced very low values compared to other training datasets with much better results, such as GiantMIDI-Piano, ailabs1k7, and Rock-Piano-MIDI.

With regard to Rhythmicity, the training datasets exhibit moderate values around the middle. Relatively better values are observed in ailabs1k7 and Rock-Piano-MIDI with that being explained by the fact that these datasets' genres have more obvious and simpler rhythmicities. Conversably, lower values can be partially attributed to incomplete or messy tracks with rhythm discontinuities or long pauses which are more prevalent in certain datasets, such as Los-Angeles-MIDI-segment and adl-piano-midi, affecting the perception of Rhythmicity by the subjects.

Regarding Genre, it is not clearly identifiable in single-genre training datasets, exhibiting moderate values mostly below the middle. For our observations we mainly examined single-genre datasets, with the values of these datasets being attributed to multiple factors. **Firstly, genre recognition is a difficult challenge in piano music, where genre differences are not as clear** as in multi-organic music. Additionally, **even musically experienced subjects may have limited familiarity with piano music genres and their distinctive characteristics**. However, relatively better values are observed in the ailabs1k7 dataset, followed by the Rock-Piano-MIDI, but also some unexpected low values are found in datasets regarding well-known genres, such as classic maestro-3.0.0, which we expected that would be recognized to a more satisfactory extent.

On the metric of Naturalness, training datasets received the lowest values compared to all subjective metrics, exhibiting values below the middle. Multiple factors may account for these values. Datasets comprising a considerable number of incomplete or messy pieces with discontinuities, long pauses, noise or inappropriate midi encodings, such as Los-Angeles-MIDI-segment and adl-piano-midi, may have influenced the evaluations having the lowest naturalness values. Conversely, the best naturalness values were achieved by ailabs1k7 and GiantMIDI-Piano, though there were still not satisfactory.

In regard to Overall Rating, training datasets exhibit satisfactory values close to the middle. The highest evaluated training datasets were ailabs1k7 and Rock-Piano-MIDI. Lower evaluated datasets, like Los-Angeles-MIDI-segment, were associated with incomplete pieces or pieces with discontinuities, long pauses, etc.

Key Findings

Based on the weighted mean table**, among the training datasets ailabs1k7 (3.31), Rock-Piano-MIDI (3.06) and GiantMIDI-Piano (3.01) were subjectively evaluated better**, while adl-piano-midi (2.80) and Los-Angeles-MIDI-segment (2.67) were evaluated significantly lower. **In terms of model outputs ailabs1k7 (3.01), Los-Angeles-MIDI-segment (3.00) and adl-piano-midi (2.98) had the highest evaluations**, with the last two surpassing significantly their original training datasets evaluations.

In general, training datasets quality is below the expected standards. The model outputs have outperformed the datasets in many metrics, indicating the need for improvement in the training datasets. Even though training datasets consist of humans composed pieces, they were not evaluated as highly as expected, especially in terms of naturalness. This suggests that there are significant shortcomings in the quality of these training data for music generation purposes.

A considerable number of incomplete and disorganized pieces within a dataset can have a substantial negative impact on the quality of the generated pieces. The presence of discontinuities, long pauses, and noise in incomplete or messy pieces can result in similar issues in the generated pieces, leading to lower quality outputs. The subjective metrics most affected include melodiousness, harmonicity, rhythmicity, and naturalness. **In order to partially identify datasets with this characteristic, the Empty Beat Rate objective metric can be used.** Datasets with a higher prevalence of incomplete and unorganized pieces tend to have higher mean values or more high value outliers in the training or output datasets in Empty Beat Rate. For instance, adl-piano-midi and Los-Angeles-MIDI-segment are datasets that exhibit this characteristic more frequently, while ailabs1k7 having a significantly low Empty Beat Rate lacks this characteristic.

Diverse multi-genre datasets are more effective in enhancing the generalization capabilities of models compared to single-genre datasets. By incorporating different genres within a dataset, models are exposed to a wider variety of musical styles, structures, and features. This enables them to learn more generalized representations of music and avoid overfitting, making them more capable of generating high-quality outputs across different genres' primers. Furthermore, **larger multi-genre datasets can reduce the impact of issues like incomplete and disorganized pieces to some extent.** This is demonstrated by the adl-piano-midi and Los-Angeles-MIDI-segment models outputs, which were evaluated better than the original training datasets.

Large-scale datasets over 100 MB can improve the generalization capabilities of models. By exposing models to a greater volume of musical pieces, they can learn more generalized representations of the underlying patterns and structures of the music they are trained on. Furthermore, **larger scale can reduce the impact of issues like incomplete and disorganized pieces to some extent. Conversably, smaller scale limits the effect of other positive dataset**

characteristics. For instance, the ailabs1k7 models' outputs were unable to achieve similar levels of performance as the training dataset, although the training dataset is of the highest quality.

The Scale Consistency objective metric can help identify datasets that orientate towards emotionally rich music to some extent. Consistent scales and harmonies are often used in popular music genres, which can make listeners feel a more intense emotional experience and more familiar with the music. This is demonstrated by the ailabs1k7 models, which are evaluated better than other datasets on Emotion.

Model Types

During the training and evaluation process, we assessed the outputs of our two models, Music Transformer and Perceiver-AR on several metrics and identified specific attributes and characteristics of the models' that had a considerable impact on the training process and the quality of generated musical outputs. Though our coverage regarding the determining factors of these metrics' values could not be exhaustive.

Models Training

Perceiver-AR models converge to at least an order of magnitude lower evaluation loss values 0.004-0.03 compared to Music Transformer models 0.29-1.15. This can be explained by the increased model capacity and dimensions of Perceiver-AR models having higher capability of modeling the problem more sufficiently.

The number of epochs Perceiver-AR models converge have much less variations, 7-14 epochs, compared to Music Transformer models, 5-60 epochs. This observation demonstrates another consequence of the different model capacities, with the higher capacity model Perceiver-AR being more robust, independent of the training dataset.

Objective Evaluation

Model Type	Mean	Std	Distributions	Outliers	Similarity with Training Datasets
Common Characteristics					• Scale Consistency • Tempo Estimation
Music Transformer	Lower	High	Wide, diverse	Frequent low value	• Polyphony • Polyphony Scale
Perceiver-AR	Slightly lower	Small-Moderate	Concentrated	Rare	• Pitches Used • Pitch Range • Pitch Classes Used • Pitch Entropy • Pitch Class Entropy • Empty Beat Rate

Table 16: Objective Metrics Model Type Comparison

The comparison between Music Transformer outputs and Perceiver-AR outputs in terms of objective musical metrics provides several conclusions. Firstly, **both Music Transformer outputs and Perceiver-AR outputs generally have lower values for most metrics compared to the training datasets.** Furthermore, **the distributions of the metrics are often symmetrical**, indicating that the models are generating musical outputs with a balanced distribution of musical characteristics. For Scale Consistency and Tempo Estimation, both models have outputs with similar means and distributions close to the corresponding training datasets.

Music Transformer outputs tend to have smaller mean values, higher standard deviations, more low-value outliers and more wide and diverse distributions compared to the training datasets and Perceiver-AR outputs for most objective metrics. **Perceiver-AR outputs are more concentrated and similar to the training datasets and have fewer outliers compared to Music Transformer outputs in most metrics.** This possibly suggests that Perceiver-AR tend to generate musical outputs that are more consistent and have less diverse musical characteristics compared to the training data. This is particularly evident in pitch-related metrics. However, for polyphony-related metrics and Empty Beat Rate, Music Transformer outputs are closer to the datasets, while Perceiver-AR outputs are significantly lower.

Subjective Evaluation

Subjective Metric	Subject Group	Training Datasets	Music Transformer Outputs	Perceiver-AR Outputs
Familiarity	Basic	3.12	**3.06**	3.02
Emotion	Basic	2.89	3.01	**3.06**
	Pro	3.35	3.24	**3.36**
Melodiousness	Pro	3.07	2.95	**3.07**
Harmonicity	Pro	3.16	3.08	**3.21**
Rhythmicity	Pro	3.00	2.95	**3.08**
Genre	Pro	2.92	**2.91**	2.80
Naturalness	Basic	0.43	**0.43**	0.41
	Pro	0.37	0.28	**0.37**
Rating	Basic	2.90	**2.94**	2.90
	Pro	3.03	2.92	**3.04**
Weighted Mean	Basic & Pro Weighted	2.98	2.90	**2.99**

Table 17: Subjective Metrics Model Type Comparison

Both Music Transformer and Perceiver-AR output datasets generally have moderate values across most metrics, with better results for Harmonicity and Emotion. **Music Transformer outputs (2.90 weighted mean) have lower values than the training datasets (2.98 weighted mean). Perceiver-AR outputs (2.99 weighted mean) have similar values to the training datasets,** performing better than Music Transformer in most metrics.

Most output datasets show higher Emotion values compared to the training datasets, **highlighting the models' ability to generate emotionally rich music**. However, **the lower output values for Genre suggest that the genre characteristics of the training datasets are not clearly passed to the outputs**, with the pieces being influenced by the primers' genre as well.

On the metric of Naturalness, output datasets received the lowest values. One of the determining factors for this is the difficulty of generating music that accurately simulates every intricate detail of human composed music. Other **determining factors could be small discontinuities or tangles in the melody, abrupt changes in rhythm, and pauses**. Improvements are needed primarily on the training datasets.

Regarding the overall rating, the output datasets exhibit moderate results, mostly having higher values than the training datasets, indicating the potential of the models to generate better music.

Key Findings

Music Transformer

The Music Transformer model has lower modeling capacity for musical elements, resulting in greater fluctuations and more susceptibility to the training datasets characteristics during the training process. The outputs of the model are more diverse with more outliers and have lower subjective evaluations than the original training data.

The Music Transformer may be capable of producing more creative music than what is present in the training datasets. As (Ji et al., 2020) discuss, musical creativity involves extrapolating patterns beyond what is seen in the training data. Music Transformer has more diverse musical characteristic distributions than the training datasets with more outliers, suggesting that it may be better able to produce innovative music that differs from the characteristics of the training data. However, it is also possible that these statistical differences between the Music Transformer outputs and the training data are due to a higher level of randomness or model inaccuracies. Additional evaluation is required to determine if the Music Transformer is truly capable of producing more creative music.

Perceiver-AR

The Perceiver-AR model better captures intricate musical elements and patterns in the training data. It is more robust and performs consistently regardless of the training dataset. Its outputs are less diverse, more similar to the original training data, and are subjectively evaluated similarly or better than the training datasets.

Perceiver-AR may be able to mitigate limitations of the training datasets to some extent. As previously discussed, many training datasets don't have the required quality having been assessed lower than expected in a variety of subjective metrics. Nevertheless, Perceiver-AR was able to generate outputs that outperformed the training datasets in some of these metrics, suggesting that it has the ability to produce higher quality results despite the limitations of the training data.

Common Characteristics

The length of the input context has a substantial impact in the understanding of musical structure. Perceiver-AR can access longer musical sequences during training and music generation being more capable of remembering medium and long-term structures of a piece. Conversably, Music Transformer tends to repeat shorter musical patterns due to its limited context window. Therefore, for optimal results regarding this parameter, the models should be able to access the complete musical sequence of a track as a single entity.

The lack of clear representation of musical principles such as structure, melody, harmony and rhythm, has a significant negative impact in the modelling of music. Both models learn this information indirectly in a tangled and confused way**.** This is the main cause of the model outputs having in some extent discontinuities or tangles in the melody, abrupt changes in rhythm, and pauses, since the models lack an understanding that these characteristics are typically undesirable. By incorporating clear encoding of these principles, the models can understand the relationships between different elements of music and how they interact to create better musical pieces.

Trained Models

After assessing our two model architectures, we evaluated all the trained models using subjective metrics and identified **three best performing models: Perceiver-AR adl-piano-midi (3.09 weighted mean), Perceiver-AR Rock-Piano-MIDI (3.06 weighted mean) and Perceiver-AR ailabs1k7 (3.02 weighted mean)**. Overall, Perceiver-AR adl-piano-midi was found to be the most robust and best performing model, Perceiver-AR Rock-Piano-MIDI was the second best performing and best model according to pro subjects, and Perceiver-AR ailabs1k7 was the best model according to basic subjects.

Based on the subjective evaluation, Perceiver-AR adl-piano-midi outperformed the other models in Naturalness and Rating and also performed significantly well in Familiarity, Emotion, Melodiousness, Harmonicity and Genre. Perceiver-AR Rock-Piano-MIDI was found to be best in Emotion (for Pro Subjects), Melodiousness, Harmonicity, Rhythmicity and Rating and also performed significantly well in Genre. Perceiver-AR ailabs1k7 was the best model in terms of Familiarity, Emotion (for Basic Subjects), Naturalness (for Basic Subjects), Rating (for Basic Subjects) and also performed significantly well in Rating (for Pro Subjects). It should be noted that in cases where multiple models show similar level of favorable performance on a metric with very little deviation, they were reported as both performing best.

Evaluation Metrics Correlations

After analyzing training datasets and models through training and evaluation, **we estimate correlations derived from the objective and subjective metrics and examine possible causal relationships that explain some of these correlations**. It is important to note that a correlation does not imply causation, and these relationships may be influenced by other factors that are not accounted for in the analysis. We attempt to identify the significance of each subjective metric in

determining music quality and propose a set of primary quality indicators for music generated by artificial intelligence models.

Objective Metrics

Pitch related metrics are moderately correlated in most cases. This can be explained, as these metrics are often interdependent on each other. For example, the pitch used metric is correlated with the pitch range metric because, if more pitches are being used in a musical piece, it would naturally lead to more pitch range.

Scale consistency has a negative moderate correlation with pitch classes and pitch classes entropy. If a musical piece has high scale consistency, it means that the pitches in the piece are more likely to conform to a particular musical scale, resulting in fewer unique pitch classes in the piece, and lower pitch classes entropy. On the other hand, if a musical piece has low scale consistency, the pitches in the piece do not conform to a particular musical scale, resulting in a wider distribution of pitch classes and higher pitch classes entropy.

Subjective Metrics

Most subjective metrics are at least moderately correlated, as they all influence music quality to some extend and form possible causal relationships.

Familiarity is moderately correlated with Emotion (0.57), as people tend to have a stronger emotional response to music, they are familiar with. This is because familiar music is often associated with personal memories, experiences, and emotions, eliciting stronger emotional responses when similar music is heard again.

Melodiousness and Harmonicity highly influence Emotion in music, as evidenced by the strong correlation between these measures (0.71 and 0.70 respectively), as they are the primary aspects and key qualities of music pieces. Therefore, they contribute to a great extend to the perceived pleasantness and emotional impact of the music. Rhythmicity moderately contributes to Emotion, as indicated by its correlation (0.58), corresponding to the perceived energy of the pieces. **Melodiousness, Harmonicity and Rhythmicity are strongly positively correlated, indicating they are musical aspects that are strongly interrelated.**

Genre does not appear to be a determining factor of the music quality, as evidenced by its moderate correlations with other metrics. While the identification of a music genre can be useful to categorize music and its influences, it appears to not be so important for the perceived quality of music, as music listeners prioritize other aspects of music pieces.

Naturalness is related to the overall quality of the music to a lesser extent than other metrics, having weak to moderate correlations with all other subjective metrics. Naturalness represents how closely the music resembles natural, human composed music, which is a factor that contributes to the overall perceived quality of the music. However, its binary nature limits the amount of information that can be conveyed, leading to lack of granularity and weaker correlations with all other metrics.

Rating is a comprehensive measure of overall perceived quality of the music, as it has moderate to high correlations with all other subjective metrics. As expected, it reflects the combined influence of all other subjective metrics on the overall perception of music quality.

Melodiousness, Harmonicity, and Emotion are the subjective metrics that are most effective at evaluating the quality of music generated by AI models, forming the primary quality indicators for subjective music evaluation. These metrics have a high correlation with Rating (0.78, 0.75 and 0.71 respectively), suggesting they are the ones that mostly influence subjects regarding their evaluation of music quality. When evaluating the quality of generated music, it is essential to include and give the greatest weight to these metrics, as they provide the most reliable reflection of subjects' opinions of the music quality. Consequently, they should primarily be assessed during subjective music evaluation.

Objective-Subjective Metrics

We present the correlations heatmap for all the objective and subjective metrics regarding the musical pieces that were subjectively evaluated. As it is demonstrated, there is no statistical correlation between the objective and subjective metrics for the evaluated musical pieces

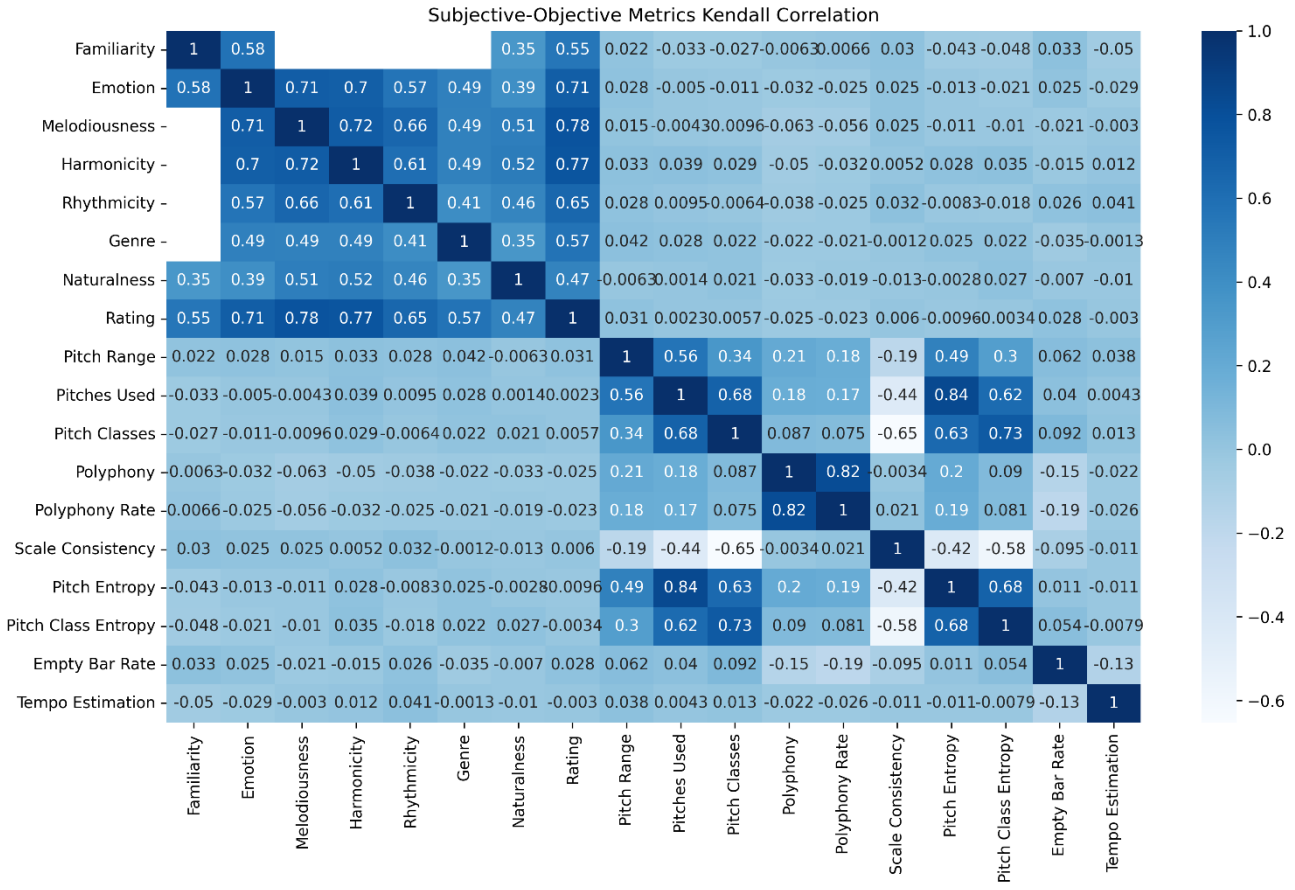

Figure 10.1: Objective-Subjective Metrics Kendall Correlations Heatmap

Chapter 11: Discussion

In this chapter, we explore the limitations and obstacles encountered during the research, along with possible improvement in our method and implementation. We also suggest some future directions for the field.

Problems & Improvements

Models Training

The validation frequency is not consistent between Music Transformer and Perceiver-AR models. Music Transformer is validated at the end of each epoch, while Perceiver-AR is evaluated every 100 training steps. The validation frequency of Music Transformer is determined by the total number of training batches, which varies depending on the dataset size. As a result, datasets with more training batches, such as ADL Piano MIDI and Los Angeles MIDI segment, have fewer validation points for Music Transformer, which limits the amount of information provided by its learning curves. **A possible implementation improvement would be to validate Music Transformer models at a constant number of training steps.**

Objective Evaluation

The duration of the musical pieces could be a useful objective metric to consider in the objective evaluation. This metric is correlated with pitch-related metrics to some extent and can help filter musical pieces for subjective evaluation. Additionally, analyzing the duration of generated pieces can provide insights into the outputs datasets by comparing their durations with those of the training datasets.

Subjective Evaluation

The original training samples used in the total subjective evaluation samples were quite limited, resulting in a small number of evaluations for the training datasets compared to the output datasets. Specifically, the training datasets were evaluated much less frequently than the output datasets, comprising only 14.7% of the total evaluations. It would be preferable to have a similar number of training datasets – model outputs evaluations as it would lead to more accurate and fair comparisons between them. **A possible improvement would be to increase the proportion of original samples in the total evaluation samples to approximately 33% in order to obtain a similar number of training datasets evaluations.**

In some cases, subjective evaluation results show discrepancies between the two groups of participants. One group would rate a dataset or a trained model highly, while the other would rate it very low. These variations might be attributable to differences in musical knowledge between the two groups, but the number of participants is another critical factor. If the subjective evaluation had a higher number of participants, the conflicting results would be minimized, and we could investigate possible causal relationships that could explain some of these inconsistencies.

The subjective evaluation did not include an assessment of the transitions between the input sequences and the resulting pieces. This means that the overall coherence and structure of the music pieces were not fully evaluated. Since the naturalness metric required examining the musical pieces without the input sequence, the smoothness of the transitions and the overall structure of the pieces were not evaluated. **A possible improvement would be to evaluate the coherence of the musical piece with the primer sequence after having completed the other subjective metrics evaluations without the primer.**

The subjective evaluation lacked qualitative analysis of the musical pieces, which could have provided insights into the characteristics of our training datasets, models, and metrics. **A potential solution could be including a question with predefined options for the listeners to describe some typical attributes of the pieces**, such as discontinuities, tangles in the melody, inaccuracies in harmony, abrupt rhythm changes, theme repetitions, pauses, sticking points, pleasing transitions and connections, familiar patterns, sudden closures, **or allowing them to provide descriptions of other attributes.**

Limitations & Future Directions

Training Datasets

The limited availability of datasets is a challenge to the advancement of complex models in our field. The lack of a variety of openly available datasets has posed a challenge for us in collecting training datasets for our models. The issue has been acknowledged by other researchers too, including (Cancino-Chacón et al., 2018), who emphasize the need for more comprehensive datasets to support the development of better music generation models. The lack of diverse and comprehensive datasets makes it difficult to train models that can capture the complexity and nuances of musical performances, leading to limitations in the models' ability to generate better quality music. Therefore, there is a need of creating more datasets.

Another primary limitation of music generation field is the quality of the training datasets. As highlighted by our conclusions, the training datasets used were generally below the expected standards. This results in lower quality outputs, which can be seen in many metrics evaluations with the most typical being naturalness. One factor that can significantly impact the quality of these datasets is the presence of incomplete and disorganized pieces with discontinuities, long pauses, and noise. Therefore**, it is essential to carefully curate and preprocess the training datasets with high quality annotations.** By doing so, it is possible to optimize the effectiveness of the music generation models and produce higher quality music.

The limited size and lack of musical genres diversity of the training datasets represents also one of the main restrictions of the field. Most available datasets are restricted mainly on classical genre, which makes most of the models focus on the same music styles. The training datasets should ideally represent the entire spectrum of musical genres, including different rhythms, chord progressions, melodies, harmonies, and dynamics. Also, as concluded by our research, large multi-genre datasets are more effective in enhancing the generalization capabilities of models compared

to single-genre datasets and can reduce the impact of issues like incomplete and disorganized pieces to some extent, improving the quality of the generated music. **This highlights the importance of continuing to develop and expand existing datasets, as well as creating new ones that meet the desired standards for scale and diversity.**

Music Representation

MIDI-like event representation is very limited to capture the complexity of musical language. The use of pitch, time shift, and velocity events, although commonly used, is insufficient in capturing the intricate nuances of music, as (Ji et al., 2020) also point out. **The development and implementation of new forms of more musically rich representations is crucial in effectively conveying the diversity of musical elements.** In addition to pitch and timing, elements such as structure, tempo, and chords, as well as higher-level concepts such as grooving and musical emotion, could be incorporated to musical tokens. By representing a wider range of musical elements, the models would have a better understanding of whole structure of the pieces, note directions, chord progressions and rhythmic patterns, thus enhancing their ability to produce more quality outputs. A promising effort in this direction is the REMI representation introduced by (Y.-S. Huang & Yang, 2020).

Music Generation Models

The computational resources needed for training Transformer models are very high. This poses a significant obstacle to their development and experimentation with these models in the field. To address this challenge, **researchers must focus on reducing the computational requirements of these models** during training. **A promising solution to this issue is transfer learning,** where a model is trained on one dataset and then fine-tuned on another. (Donahue et al., 2019) have introduced an approach in this direction that shows promise.

Models should be able to generate music with long-term structure and appropriate musical closure. Currently, our models have the ability to produce music with a fixed number of output events using a recursive generation process. Nevertheless, this approach does not allow control over when the music should end, leading to sudden and unsatisfying endings that lack the musical closure that is essential for genuine music.

Users can interact with our music generation models only by providing the initial sequence and the length of the output sequence. Since these models were not designed to be controllable, users are unable to engage with the models in a more extensive way, such as by controlling musical elements like scale, rhythm, piece structure, and theme. **To achieve more comprehensive interaction, several musical features need to be represented and adjustable within music generation models**. This would enable music generation models to function fully as composition assistant systems.

Music generation models should be able to produce more creative music that deviates from the training data. As also (Ji et al., 2020) emphasize that the output should not only statistically resemble the patterns learned from the training dataset, but also exhibit diverse musical

characteristics. By doing so, the models can explore and showcase new musical styles and genres, ultimately contributing to pure musical innovation.

Music generation models could be designed to generate emotionally rich music that reflects specific moods and emotions. This requires a deep understanding of the intricate relationship between music and emotion, which should be integrated into the music representation. This approach could result in producing music that effectively communicates a range of emotional states and can be used in a variety of applications, including film scoring, therapy, and entertainment.

Objective Evaluation

New musically informed objective metrics should be aligned with subjective evaluation metrics, forming a primary set of objective evaluation quality indicators. As (Ji et al., 2020) addresses, objective metrics used by different researchers are often diverse, leading to varying evaluation results for the same models. Moreover, there is a lack of correlation between quantitative metrics and subjective evaluation. **Future research should establish a strong correlation between quantitative metrics and subjective evaluation, and work towards automating the music evaluation process** by integrating complex music knowledge into the generation models through automatic feature extraction methods.

Subjective Evaluation

A higher number of evaluations would provide more reliable, robust and detailed results about our training datasets and models. First, it could increase the reliability of our results by reducing the impact of individual biases or random variations in the subjective judgments. With a scale larger number of evaluations, we could obtain a more accurate perception of the musical pieces' evaluations by a larger sample of listeners. Additionally, a more in-depth analysis of the training datasets and models would be made, identifying more potential areas of improvement for our models. Finally, some conflicting results would be minimized, and we could investigate possible causal relationships that could explain some of these inconsistencies. However, it was not feasible for us obtain a larger number of evaluations due to resource and time constraints.

Part IV: Εκτεταμένη Περίληψη

Εισαγωγή

Υπόβαθρο & Στόχος

Η σύνθεση μουσικής με τη βοήθεια της τεχνητής νοημοσύνης είναι ένας ταχέως αναπτυσσόμενος τομέας που έχει προσελκύσει πρόσφατα την προσοχή ερευνητών, μουσικών και επαγγελματιών της μουσικής βιομηχανίας. Τα συστήματα σύνθεσης μουσικής μπορούν σε γενικές γραμμές να χωριστούν σε δύο κατηγορίες: αυτόνομα συστήματα μουσικής σύνθεσης και συστήματα υποβοήθησης σύνθεσης, όπως αναφέρουν οι (J. P. Briot, 2021; J.-P. Briot et al., 2020).

Στόχος της παρούσας εργασίας είναι η δημιουργία ενός συστήματος υποβοήθησης σύνθεσης συμβολικής μουσικής για πιάνο, το οποίο θα υποστηρίζει ερασιτέχνες μουσικούς στις μουσικές τους αναζητήσεις. Θέλουμε να δοθεί η δυνατότητα στους μουσικούς να απελευθερώσουν τις μουσικές τους ικανότητες και να δημιουργήσουν μοναδικά και πρωτότυπα μουσικά κομμάτια, καθιστώντας τη διαδικασία σύνθεσης μουσικής πιο προσιτή αλλά και πιο ευχάριστη.

Αντικείμενο

Για την επίτευξη του σκοπού της εργασίας, το αντικείμενο της ορίζεται ως εξής:

- Εκπαίδευση των μοντέλων Music Transformer και Perceiver-AR σε έξι σύνολα δεδομένων διαφόρων μεγεθών και μουσικών ειδών.
- Σύνθεση μεγάλου αριθμού εξόδων για κάθε περίπτωση μοντέλου.
- Αντικειμενική και υποκειμενική αξιολόγηση των συνόλων εκπαίδευσης και των συνθέσεων των μοντέλων.
- Διερεύνηση της επίδρασης των χαρακτηριστικών των συνόλων εκπαίδευσης, των τύπων μοντέλων και των εκπαιδευμένων μοντέλων στην ποιότητα των παραγόμενων κομματιών.
- Εκτίμηση των συσχετίσεων μεταξύ των μετρικών αξιολόγησης και διερεύνηση πιθανών αιτιωδών σχέσεων.
- Πρόταση ενός συνόλου πρωτογενών υποκειμενικών δεικτών ποιότητας για τη παραγόμενη μουσική από μοντέλα τεχνητής νοημοσύνης.

Δομή

Η εργασία χωρίζεται σε τρία κύρια μέρη με συνολικά 11 κεφάλαια, καθένα από τα οποία επικεντρώνεται σε μια διαφορετική πτυχή της έρευνας μας.

Το Μέρος Ι, το οποίο περιλαμβάνει τρία κεφάλαια, είναι μια βιβλιογραφική επισκόπηση που παρέχει μια εισαγωγή σε στοιχεία της μουσικής θεωρίας. Καλύπτει επίσης τις αρχές της σύνθεσης μουσικής με βαθιά μάθηση, συμπεριλαμβανομένων των μοντέλων βαθιάς μάθησης, των μεθόδων αναπαράστασης μουσικής, της εκπαίδευσης μοντέλων και των μεθόδων αξιολόγησης. Επιπλέον, παρέχει μια επισκόπηση των μοντέλων Music Transformer και Perceive-AR.

Το Μέρος ΙΙ, το οποίο αποτελείται από πέντε κεφάλαια, έχει τίτλο Μέθοδοι και Αποτελέσματα. Το Κεφάλαιο 5 εξετάζει τα σύνολα εκπαίδευσης και τις μεθόδους αναπαράστασης μουσικής, ενώ το Κεφάλαιο 6 καλύπτει την εκπαίδευση των μοντέλων, συμπεριλαμβανομένων των υπερπαραμέτρων, της αρχιτεκτονικής και των παραμέτρων τους, καθώς και τη διαδικασία εκπαίδευσης. Παρουσιάζονται επίσης οι παραγόμενες καμπύλες μάθησης, καθώς και τα επιλεγμένα εκπαιδευμένα μοντέλα. Το Κεφάλαιο 7 επικεντρώνεται στην παραγωγή συνθέσεων, ενώ το Κεφάλαιο 8 και το Κεφάλαιο 9 καλύπτουν την αντικειμενική και υποκειμενική αξιολόγηση, αντίστοιχα.

Το Μέρος ΙΙΙ, Συμπεράσματα και Συζήτηση, περιλαμβάνει δύο κεφάλαια. Το Κεφάλαιο 10 συνοψίζει τα συμπεράσματα σχετικά με τα σύνολα εκπαίδευσης, τα είδη μοντέλων, τα εκπαιδευμένα μοντέλα και τις συσχετίσεις των μετρικών αξιολόγησης. Το Κεφάλαιο 11 είναι μια συζήτηση των προβλημάτων που αντιμετωπίστηκαν κατά τη διάρκεια της έρευνας και των πιθανών βελτιώσεων, καθώς και των περιορισμών και των μελλοντικών κατευθύνσεων στο πεδίο.

Το παρόν Μέρος IV είναι μια εκτεταμένη περίληψη της εργασίας στα ελληνικά, με την ίδια δομή με το πρωτότυπο κείμενο, που περιλαμβάνει περίληψη κάθε κεφαλαίου και υποκεφαλαίου.

Τέλος, ο κώδικας υλοποίησης της εργασίας είναι διαθέσιμος στο GitHub https://github.com/aggelostais/piano-music-generation και αναλύεται στα ξεχωριστά υποκεφάλαια υλοποίησης στο Μέρος ΙΙ. Επιπλέον, η σελίδα της έρευνας για την υποκειμενική αξιολόγηση είναι https://ai-music-generation-survey.vercel.app/ και ο κώδικάς της παρέχεται στο GitHub https://github.com/aggelostais/ai-music-generation-survey.

Μέρος Ι: Θεωρητική Επισκόπηση

Κεφάλαιο 1: Στοιχεία Μουσικής Θεωρίας

Η τέχνη της μουσικής είναι σύνθετη και πολυδιάστατη, ενσωματώνοντας πολλά διαφορετικά στοιχεία. Η κατανόηση των αρχών της μουσικής απαιτεί βαθιά κατανόηση των πολλών διαφορετικών στοιχείων που τη συνθέτουν και του τρόπου με τον οποίο αλληλεπιδρούν μεταξύ τους. Θα επικεντρωθούμε σε λίγα βασικά στοιχεία που είναι ευρέως αποδεκτά στη μουσική θεωρία και θα μας χρειαστούν στη συνέχεια της εργασίας.

- **Τονικό Ύψος (Pitch):** Το ψηλό ή χαμηλό ύψος ενός μουσικού τόνου που καθορίζεται από την ηχητική συχνότητα του. Προσδιορίζεται ισοδύναμα από το συνδυασμό νότας και οκτάβας σε ένα μουσικό όργανο.
- **Μελωδία (Melody):** Η εξέλιξη των μουσικών τόνων στο χρόνο ως ενιαία οντότητα ή μοτίβο.
- **Μελωδικότητα (Melodiousness):** Η αντιληπτή ποιότητα της μελωδίας ενός κομματιού.
- **Αρμονία (Harmony)**: Ο ταυτόχρονος συνδυασμός τόνων που δημιουργούν συγχορδίες και ακολουθίες συγχορδιών.
- **Αρμονικότητα (Harmonicity):** Η έκταση στην οποία οι νότες που παίζονται ταυτόχρονα σε μια συγχορδία ή ακολουθία συγχορδιών ακούγονται αρμονικά συνεπείς.
- **Ρυθμός (Rhythm)**: Ο τρόπος που είναι οργανωμένες στο χρόνο οι νότες δημιουργώντας γρήγορα ή αργά χρονικά μοτίβα.
- **Ρυθμικότητα (Rhythmicity):** Η αντίληψη του ρυθμού ως συνεπή, συνεκτικού και ομαλού.
- **Φυσικότητα (Naturalness, Humanness)**: Η αντίληψη των μουσικών μελωδιών, ρυθμών ή φράσεων ως φυσικές με βάση τις προσδοκίες του ακροατή.

Κεφάλαιο 2: Σύνθεση Μουσικής με Βαθιά Μάθηση

Βαθιά Νευρωνικά Δίκτυα

Σε αυτό το υποκεφάλαιο, το οποίο βασίζεται στους (Hernandez-Olivan, Hernandez-Olivan, et al., 2022; Hernandez-Olivan & Beltrán, 2023), θα επικεντρωθούμε σε μερικά από τα πιο πρόσφατα και αξιοσημείωτα μοντέλα βαθιάς μάθησης για τη σύνθεση πολυφωνικής μουσικής πιάνου. Η σύνθεση μουσικής έχει επιχειρηθεί με τη χρήση διαφόρων αρχιτεκτονικών νευρωνικών δικτύων όπως τα VAEs, τα GANs, τα Αναδρομικά Νευρωνικά Δίκτυα (RNNs), τα NADEs, τα Transformer καθώς και συνδυασμούς αρχιτεκτονικών.

Τα Αναδρομικά Νευρωνικά Δίκτυα (RNN) ήταν η πρώτη αρχιτεκτονική νευρωνικών δικτύων που χρησιμοποιήθηκε για τη δημιουργία μουσικής με μακροπρόθεσμη δομή, καθώς είναι από τα απλούστερα μοντέλα για ακολουθιακά δεδομένα. Μερικά παραδείγματα τέτοιων μοντέλων είναι το MelodyRNN (J. Wu et al., 2017) της Google Magenta και το DeepBach (Hadjeres et al., 2017). Ωστόσο, αυτά τα είδη μοντέλων παρήγαγαν μόνο σύντομες ακολουθίες και καθώς το ενδιαφέρον για τη δημιουργία μεγαλύτερων ακολουθιών αυξήθηκε, δοκιμάστηκαν και άλλες αρχιτεκτονικές.

Τα μοντέλα Transformer έδειξαν μεγάλες δυνατότητες στην παραγωγή πολυφωνικής μουσικής, καθώς ο τομέας στόχευε στη δημιουργία μεγαλύτερων ακολουθιών. Οι (C.-Z. A. Huang et al., 2018) παρουσίασαν το Music Transformer, που χρησιμοποιεί το μηχανισμό relative attention στην αρχιτεκτονική του Transformer αποτελώντας το πρώτο μοντέλο αυτής της αρχιτεκτονικής που εφαρμόστηκε με επιτυχία στο συγκεκριμένο πεδίο. Για αυτούς τους λόγους το επιλέξαμε ως ένα από τα μοντέλα που θα εκπαιδεύσουμε. Επίσης, προτάθηκε το MuseNet από την OpenAI (Payne Christine, 2019) που έχει τη δυνατότητα να παράγει μουσικές συνθέσεις διάρκειας 4 λεπτών και έως 10 διαφορετικών οργάνων. Τέλος, ένα από τα πιο πρόσφατα και αξιοσημείωτα μοντέλα είναι το Perceiver-AR από (Hawthorne et al., 2022a). Αυτό το μοντέλο, συγκεκριμένα, χρησιμοποιεί ένα μηχανισμό cross-attention και έχει context 4,096 tokens και είναι το δεύτερο μοντέλο που επιλέξαμε να εκπαιδεύσουμε αποτελώντας ένα από τα πιο πρόσφατα μοντέλα του πεδίου.

Ο συνδυασμός VAE και Transformers ή άλλων νευρωνικών δικτύων έχει γεννήσει νέα μοντέλα που έχουν ως στόχο να ξεπεράσουν το πρόβλημα της απώλειας συνοχής ή κύριου μοτίβου, που εμφανίζεται στα μοντέλα που βασίζονται σε Transformers, κατά τη δημιουργία μεγαλύτερων ακολουθιών. Μοντέλα όπως το TransformerVAE (Jiang et al., 2020) και το PianoTree (Wang et al., 2020) είναι παραδείγματα τέτοιων προσπαθειών.

Αναπαράσταση Μουσικής

Η αναπαράστασης της μουσικής είναι μια σημαντική πτυχή της σύνθεσης μουσικής που επηρεάζει σε μεγάλο βαθμό την όλη διαδικασία εκπαίδευσης των μοντέλων και παραγωγής μουσικής. Όπως συζητήθηκε από τους (Ji et al., 2020), αυτή μπορεί να γίνει με δύο τρόπους:

- **Συμβολική Αναπαράσταση**: Διακριτές μεταβλητές που αποτυπώνουν σημαντικές πτυχές της μουσικής, όπως τις χρονικότητες, τα τονικά ύψη και τις εντάσεις των νοτών.
- **Αναπαράσταση Ήχου**: Συνεχείς μεταβλητές που διατηρούν όλη την ηχητική μουσική πληροφορία και αποτυπώνουν όλες τις ακουστικές λεπτομέρειες.

Στην εργασία αυτή χρησιμοποιούμε τη συμβολική αναπαράσταση για τους παρακάτω λόγους:

- Τα περισσότερα μοντέλα βαθιάς μάθησης για την παραγωγή μουσικής βασίζονται σε συμβολικές αναπαραστάσεις.
- Τα συμβολικά σύνολα δεδομένων είναι ευρύτερα διαθέσιμα και μπορούν να υποβληθούν σε επεξεργασία και εκπαίδευση με πολύ πιο υπολογιστικά αποδοτικό τρόπο.
- Ο πυρήνας της ανθρώπινης μουσικής σύνθεσης εκφράζεται μέσω συμβολικών αναπαραστάσεων.
- Η επεξεργασία και ο μετασχηματισμός ηχητικών αναπαραστάσεων ξεφεύγει από το αντικείμενο του παρόντος έργου.
- Ανεξάρτητα από το αν παράγεται μουσική σε συμβολική ή ηχητική αναπαράσταση, οι αρχές της βαθιάς μάθησης και της αναπαράστασης της μουσικής είναι σε μεγάλο βαθμό οι ίδιες (J.-P. Briot et al., 2020).

Η μορφή MIDI (Musical Instrument Digital Interface) είναι η πιο συνηθισμένη συμβολική μορφή για την αναπαράσταση μουσικής, όντας αυτή που υλοποιήσαμε και εμείς. Κάθε μουσικό κομμάτι αποτελείται από μια ακολουθία γεγονότων (tokens) που περιγράφουν το ύψος, το χρόνο και την ένταση (πόσο δυνατά χτυπήθηκε το πλήκτρο) των νοτών που παίζονται.

Εκπαίδευση Μοντέλων

Τα μοντέλα Transformer έχουν εφαρμοστεί με επιτυχία στην σύνθεση μουσικής με την απόδοση τους να εξαρτάται σε σημαντικό βαθμό από τις υπερπαραμέτρους τους. Ορισμένες από τις βασικές υπερπαραμέτρους είναι οι εξής:

- **Ρυθμός Μάθησης**: Ο ρυθμός με τον οποίο το μοντέλο ενημερώνει τα βάρη του μοντέλου κατά τη διάρκεια της εκπαίδευσης.
- **Αριθμός Εποχών**: Ο αριθμός φορών που το μοντέλο θα εκπαιδευτεί σε ολόκληρο το σύνολο εκπαίδευσης.
- **Batch Size**: Ο αριθμός δειγμάτων που επεξεργάζεται το μοντέλο σε κάθε βήμα εκπαίδευσης.
- **Αριθμός Επιπέδων**: Ο αριθμός επιπέδων στην αρχιτεκτονική του μοντέλου.
- **Ακολουθία Εισόδου**: Το μήκος της ακολουθίας εισόδου που επεξεργάζεται το μοντέλο.
- **Μέγεθος Λεξιλογίου**: Ο αριθμός των μοναδικών γεγονότων (tokens) που χρησιμοποιούνται για την αναπαράσταση των δεδομένων εισόδου.
- **Embedding Dimension**: Το μέγεθος του διανύσματος αναπαράστασης στα επίπεδα embedding.
- **Attention Heads**: Μηχανισμοί attention, καθένας δυνητικά μπορεί να μαθαίνει διαφορετικά μουσικά χαρακτηριστικά.
- **Head Dimension**: Το μέγεθος της αναπαράστασης κάθε attention head.
- **Dropout Rate**: Η πιθανότητα ένας νευρώνας να μηδενιστεί τυχαία κατά τη διάρκεια της εκπαίδευσης.
- **Συνάρτηση Σφάλματος**: Η αντικειμενική συνάρτηση που χρησιμοποιείται για τη βελτιστοποίηση του μοντέλου κατά τη διάρκεια της εκπαίδευσης.
- **Βήματα Προθέρμανσης**: Ο αριθμός των αρχικών βημάτων εκπαίδευσης κατά τη διάρκεια των οποίων ο ρυθμός μάθησης αυξάνεται σταδιακά στη μέγιστη τιμή του.

Η λεπτομερής ρύθμιση των υπερπαραμέτρων των μοντέλων δημιουργίας μουσικής απαιτεί αρκετό πειραματισμό, δεδομένου ότι εξαρτώνται ειδικά από το σύνολο δεδομένων και το μοντέλο. Εκτός από τις γενικές υπερπαραμέτρους, ορισμένα μοντέλα στο πεδίο μπορεί να έχουν και ειδικές υπερπαραμέτρους, όπως η δομή της μουσικής για παράδειγμα.

Αξιολόγηση

Για τη αξιολόγηση της ποιότητας της παραγόμενης μουσικής από μοντέλα βαθιάς μάθησης χρησιμοποιούνται αντικειμενικές και υποκειμενικές μετρικές. Οι αντικειμενικές μετρικές παρέχουν ένα ποσοτικό μέτρο της απόδοσης των μοντέλων, ενώ οι υποκειμενικές μετρικές παρέχουν ένα πιο ποιοτικό μέτρο της εκλαμβανόμενης από τους ακροατές ποιότητας της μουσικής. Η συνάρτηση απώλειας είναι μια σημαντική αντικειμενική μετρική για την αξιολόγηση της απόδοσης των μοντέλων μηχανικής μάθησης κατά τη διάρκεια της εκπαίδευσης, αλλά αντικατοπτρίζει μόνο την ικανότητα του μοντέλου να εκπαιδεύεται στα δεδομένα. Η επιπρόσθετη επιλογή άλλων μετρικών με μουσική σημασία είναι κρίσιμη για την αξιολόγηση της ποιότητας της παραγόμενης μουσικής μετά την ολοκλήρωση της εκπαίδευσης των μοντέλων. Τέλος η χρήση ενός συνδυασμού αντικειμενικών και υποκειμενικών μετρικών βοηθά στην απόκτηση μιας ολοκληρωμένης κατανόησης για τα μοντέλα και τη παραγόμενη μουσική.

Κεφάλαιο 3: Μοντέλο Music Transformer

Το κεφάλαιο αυτό βασίστηκε στην αρχική δημοσίευση των (C.-Z. A. Huang et al., 2018) και την περίληψη του (Tan, 2020).

Αρχιτεκτονική Μοντέλου: Αποτελείται από decoder-only Transformers ενσωματώνοντας το μηχανισμό relative attention. Ο μηχανισμός αυτός είναι παραλλαγή του μηχανισμού self-attention και μοντελοποιεί σχετικές χρονικές σχέσεις που είναι σημαντικές στη μουσική.

Λειτουργία Μοντέλου: Προβλέπει το επόμενο μουσικό γεγονός (token) αναδρομικά με βάση την τρέχουσα ακολουθία εισόδου σύμφωνα με τη παρακάτω διαδικασία:

- Καθορίζεται η σημασία κάθε γεγονότος εισόδου για τη πρόβλεψη του επόμενου γεγονότος μέσω των attention scores για την ακολουθία εισόδου.
- Δημιουργείται ένα σταθμισμένο άθροισμα των embeddings της ακολουθίας εισόδου και προωθείται στα επίπεδα του μοντέλου.
- Παράγεται μια κατανομή πιθανοτήτων για το επόμενο μουσικό γεγονός.
- **Εκπαίδευση**: Ελαχιστοποίηση διαφοράς μεταξύ κατανομής προβλεπόμενου γεγονότος και του πραγματικού επόμενου γεγονότος.
- **Σύνθεση Μουσικής**: Το προβλεπόμενο επόμενο γεγονός εντάσσεται στην ακολουθία εξόδου.

Αποτελέσματα: Αποτέλεσε τη πρώτη επιτυχημένη εφαρμογή Transformers στη σύνθεση μουσικής με μακροπρόθεσμη δομή. Παράγει συνθέσεις της τάξης κάποιων λεπτών, 2,000 tokens.

Κεφάλαιο 4: Μοντέλο Perceiver-AR

Το κεφάλαιο αυτό βασίστηκε στην αρχική δημοσίευση των (Hawthorne et al., 2022a) και τις συνοπτικές παρουσιάσεις των (Hawthorne et al., 2022b) και (Tiernan, 2022).

Αρχιτεκτονική: Συνδυασμός μηχανισμών cross-attention, self-attention και casual masking

- **Μηχανισμός Cross-Attention**: Απεικόνιση της ακολουθίας εισόδου σε μια συμπιεσμένη λανθάνουσα αναπαράσταση και πραγματοποίηση λειτουργιών attention στις λανθάνουσες αναπαραστάσεις.
- **Μηχανισμός Casual Masking**: Η επίδραση εισόδων που έπονται ενός τμήματος της λανθάνουσας αναπαράστασης απαλείφεται.

Λειτουργία: Παρόμοια αναδρομική διαδικασία με το Music Transformer.

- Μεγαλύτερο context έως ~65,000 γεγονότα, δυνατότητα εκμάθησης πιο μακροπρόθεσμων μουσικών χαρακτηριστικών και δομών.
- Αποσύνδεση του μεγέθους ακολουθίας εισόδου από τους απαιτούμενους υπολογιστικούς πόρους.

Αποτελέσματα: Μεγαλύτερες συνθέσεις αρκετών λεπτών.

Μέρος ΙΙ: Μεθοδολογία & Αποτελέσματα

Κεφάλαιο 5: Σύνολα Εκπαίδευσης & Αναπαράσταση Μουσικής

Σύνολα Εκπαίδευσης

Η επιλογή ενός συνόλου δεδομένων είναι μια σημαντική πτυχή της παραγωγή μουσικών συνθέσεων, επηρεάζοντας την ποιότητα και την ποικιλομορφία της παραγόμενης μουσικής. Στη διαδικασία αυτή εστιάσαμε στο μουσικό είδος των συνόλων δεδομένων στοχεύοντας στη μέγιστη δυνατή ποικιλία ειδών για να κάνουμε σχετικές συγκρίσεις και στο μέγεθος των συνόλων θέλοντας να ενισχύσουμε τις δυνατότητες γενίκευσης των μοντέλων μας. Τα σύνολα δεδομένων που επιλέξαμε φαίνονται στο παρακάτω πίνακα:

Σύνολο Εκπαίδευσης	Μουσικό Είδος	Κομμάτια	Μέγεθος
maestro-3.0.0	classical piano	1,276	85 MB
GiantMIDI-Piano	classical piano	10,855	281 MB
ailabs1k7	pop piano	1,748	25 MB
adl-piano-midi	misc	11,086	187 MB
Rock-Piano-MIDI	rock piano	8,761	136 MB
Los-Angeles-MIDI-segment	misc	46,997	1.3 GB

Table 18: Σύνολα Εκπαίδευσης

Ορισμένα σημαντικά χαρακτηριστικά του κάθε συνόλου δεδομένων είναι:

- **maestro-3.0.0** (Hawthorne et al., 2018): Έχει επαρκές μέγεθος και ποιότητα στη κωδικοποίηση της μουσικής αποτελώντας ένα από τα πλέον χρησιμοποιούμενα σύνολα δεδομένων.
- **GiantMIDI-Piano** (Kong et al., 2020): Έχει μεγάλο μέγεθος αποτελώντας ένα από τα μεγαλύτερα MIDI σύνολα δεδομένων κλασσικού πιάνου.
- **ailabs1k7** (Hsiao et al., 2021): Είναι το μεγαλύτερο ανοιχτά διαθέσιμο σύνολο pop piano και χρησιμοποιήθηκε από τους δημιουργούς του για σύνθεση μουσικής με μοντέλα transformers.
- **adl-piano-midi** (Ferreira et al., 2020): Περιέχει πολλαπλά μουσικά είδη και βασίζεται στο Million Song Dataset, αποτελώντας ένα από τα μεγαλύτερα ανοιχτά διαθέσιμα σύνολα.
- **Rock-Piano-MIDI** (Lev, 2022): Είναι το μεγαλύτερο ανοιχτό σύνολο rock είδους και χρησιμοποιήθηκε από το δημιουργό του για σύνθεση μουσικής σε πολλές εργασίες.
- **Los-Angeles-MIDI-segment** (Lev, 2023): Είναι το μεγαλύτερο ανοιχτό σύνολο πολλαπλών μουσικών ειδών. Για λόγους υπολογιστικής απόδοσης πήραμε το 1/20 του αρχικού συνόλου.

Αναπαράσταση Μουσικής

Τα δύο μοντέλα αναπαριστούν τη μουσική πληροφορία με ακολουθίες διακριτών γεγονότων όπου κάθε γεγονός αντιστοιχεί σε μια κατηγορία γεγονότων. Οι κατηγορίες γεγονότων για τα δύο μοντέλα είναι οι εξής:

Music Transformer

- **Note On (128 γεγονότα)**: Το τονικό ύψος της νότας που παίζεται.
- **Note Off (128 γεγονότα)**: Η χρονική στιγμή λήξης μιας νότας.
- **Time Swift (100 γεγονότα)**: Ο χρόνος που πέρασε από το προηγούμενο γεγονός.
- **Velocity (32 γεγονότα)**: Η ένταση που παίζεται μια νότα.

Perceiver-AR

- **MIDI Pitch (127 γεγονότα):** Το τονικό ύψος της νότας που παίζεται.
- **Duration (127 γεγονότα)**: Η διάρκεια του τρέχοντος γεγονότος/τονικού ύψους.
- **Time Swift (127 γεγονότα)**: Ο χρόνος που πέρασε από το προηγούμενο γεγονός.
- **MIDI Velocity (127 γεγονότα)**: Η ένταση που παίζεται μια νότα.

Λαμβάνοντας υπόψη τις δύο αναπαραστάσεις, συμπεράνουμε ότι έχουν σημαντική ταύτιση, καθώς οι τύποι των γεγονότων τους έχουν παρόμοιες λειτουργικότητες. Ωστόσο, μια σημαντική διαφορά που παρατηρείται είναι το εύρος γεγονότων των ειδών Time Swift (32 γεγονότα στο Music Transformer, 127 γεγονότα στο Perceiver-AR) και Velocity (100 γεγονότα στο Music Transformer, 127 γεγονότα στο Perceiver-AR). Στο Perceiver-AR έχουμε πιο λεπτομερή αναπαράσταση με μεγαλύτερο εύρος που παρέχει μεγαλύτερες δυνατότητες έκφρασης της έντασης και δευτερευόντως των χρονικών αξιών σε ένα κομμάτι.

Υλοποίηση

Για το Music Transformer, ο κώδικας προεπεξεργασίας των midi αρχείων περιλαμβάνεται στο αρχείο /music-transformer/preprocess.py και βασίστηκε στον κώδικα από το αρχείο preprocess.py του (K. Yang, 2021a).

Για το Perceiver-AR, ο κώδικας προεπεξεργασίας των midi αρχείων βασίζεται στο κώδικα του (Lev, 2022a) και προστέθηκε στο σημειωματάριο εκπαίδευσης /perceiver-ar/Perceiver-Solo-Piano-Maker.ipynb ως κελί "Preprocess Training Data".

Κεφάλαιο 6: Εκπαίδευση Μοντέλων

Υπερπαράμετροι Μοντέλων

Σε αυτή την ενότητα, θα παρουσιάσουμε τις υπερπαραμέτρους των δύο μοντέλων, εξετάζοντας τις διαφορές στις τιμές τους και κατανοώντας βαθύτερα ορισμένους παράγοντες που διακρίνουν αυτά τα μοντέλα. Στον παρακάτω πίνακα, παρουσιάζουμε τις σημαντικότερες υπερπαραμέτρους των μοντέλων που είναι κρίσιμες για την απόδοση τους.

Υπερπαράμετρος	Music Transformer	Perceiver-AR
Αριθμός Επιπέδων	6	24
Batch Size	2	1
Ακολουθία Εισόδου	2,048	16,384

Μέγεθος Λεξιολογίου	388	512
Embedding Dimension	256	1,024
Attention Heads	4	16
Head Dimension	64	64
Συνάρτηση Σφάλματος	Categorical Cross-Entropy	
Συνάρτηση Βελτιστοποίησης	Adam	
Ρυθμός Μάθησης	Custom Scheduler	Initial 2e-5
Βήματα Προθέρμανσης	4000	-
Dropout	0.2	0.5

Table 19: Υπεπαράμετροι Μοντέλων

Batch Size: Οι τιμές 2 για το Music Transformer και 1 για το Perceiver-AR ήταν οι μέγιστες δυνατές βάσει των διαθέσιμων πόρων και γι' αυτό επιλέχθηκαν.

Ρυθμός Μάθησης

- **Music Transformer**: Υιοθετήσαμε τις προτεινόμενες τιμές για την αρχικοποίηση του αλγορίθμου Adam από την υλοποίηση του (K. Yang, 2021a).
- **Perceiver-AR**: Λόγω μείωσης του batch size της υλοποίησης του (Lev, 2022b), μειώσαμε το ρυθμό μάθησης στον αλγόριθμο Adam σε 2e-5 από την αρχική τιμή 2e-4 της υλοποίησης.

Αρχιτεκτονική & Παράμετροι Μοντέλων

Σε αυτή την ενότητα, παρουσιάζουμε την αρχιτεκτονική και τις παραμέτρους των δύο μοντέλων που απεικονίζονται στα διαγράμματα μοντέλων της αντίστοιχης ενότητας.

Για το Music Transformer έχουμε συνολικά 12,587,008 παραμέτρους ενώ για το Perceiver-AR 332,504,064 παραμέτρους. Συγκρίνοντας τα δύο μοντέλα, η διαφορά μιας τάξης μεγέθους στον αριθμό των παραμέτρων μπορεί να έχει σημαντικό αντίκτυπο στην απόδοση των μοντέλων. Για το Perceiver-AR, η ύπαρξη μεγαλύτερου αριθμού παραμέτρων θα μπορούσε να το καταστήσει πιο εκφραστικό έχοντας τη δυνατότητα να μοντελοποιήσει πιο λεπτομερώς περισσότερα μουσικά χαρακτηριστικά, αλλά και πιο επιρρεπές στην υπερπροσαρμογή, παράγοντας ενδεχομένως περισσότερο θόρυβο ή παράφωνες ακολουθίες. Για το Music Transformer, η ύπαρξη μικρότερου αριθμού παραμέτρων θα μπορούσε ενδεχομένως να το περιορίσει στην παραγωγή απλούστερων και πιο επαναλαμβανόμενων μουσικών κομματιών. Ωστόσο, θα μπορούσε επίσης να αποφύγει την υπερπροσαρμογή, με αποτέλεσμα πιο δημιουργικές ακολουθίες.

Training Batches

Σύνολο Εκπαίδευσης	Κομμάτια	Music-Transformer Training Batches	Perceiver-AR Training Batches
maestro 3.0.0	1,276	1,276	1,719
GiantMIDI Piano	10,855	10,855	9,451
Ailabs1k7	1,747	1,747	556

Rock Piano MIDI	8,761	8,761	2,321
ADL Piano MIDI	11,086	11,086	2,888
Los Angeles MIDI segment	46,997	46,997	25,179

Table 20: Training Batches Εκπαίδευσης Μοντέλων

Training Batches: Συνολικές είσοδοι εκπαίδευσης που παρουσιάζονται σε κάθε εποχή.

Για το Music Transformer, θεωρούμε κάθε μουσικό κομμάτι ως ένα training batch και ο αριθμός των συνολικών training batches για ένα σύνολο δεδομένων είναι ίσος με τον αριθμό των κομματιών σε αυτό το σύνολο δεδομένων.

Για το Perceiver-AR, χωρίζουμε το σύνολο δεδομένων σε training batches σταθερού μεγέθους και ο συνολικός αριθμός αυτών των training batches δεν είναι ίσος με τον αριθμό των κομματιών στο σύνολο δεδομένων.

Διαδικασία Εκπαίδευσης

Αντικειμενική Συνάρτηση: Χρησιμοποιήσαμε τη συνάρτηση σφάλματος. Στόχος να ελαχιστοποιήσουμε το σφάλμα επαλήθευσης και τη διαφορά μεταξύ των σφαλμάτων εκπαίδευσης-επαλήθευσης.

Επαλήθευση: Πραγματοποιήθηκε περιοδικά στο τέλος κάθε εποχής για το Music Transformer και κάθε 100 βήματα εκπαίδευσης για το Perceiver-AR.

Πρόωρη Διακοπή Εκπαίδευσης: Εφαρμόσαμε μια διαδικασία πρώιμης διακοπής για να καθορίσουμε το βέλτιστο σημείο για να σταματήσουμε την εκπαίδευσή και να αποθηκεύσουμε τα μοντέλα μας.

- Το μοντέλο μας εκπαιδεύεται μέχρι να φτάσει σε ένα απαιτούμενο σφάλμα επαλήθευσης, ανάλογα με τον τύπο του μοντέλου.
    - Music-Transformer: Σφάλμα Επαλήθευσης < 2
    - Perceiver-AR: Σφάλμα Επαλήθευσης < 1
- Αφού το μοντέλο έχει φτάσει την απαιτούμενη επίδοση, εάν στο συγκεκριμένο βήμα επαλήθευσης πετύχει ένα νέο βέλτιστο σφάλμα επαλήθευσης, αποθηκεύουμε το μοντέλο σε αυτό το σημείο.
- Εάν το μοντέλο δεν πετυχαίνει ένα νέο βέλτιστο σφάλμα επαλήθευσης για έναν ορισμένο αριθμό βημάτων εκπαίδευσης, διακόπτουμε την εκπαίδευση. Ο ορισμένος αριθμός των βημάτων για τα δύο μοντέλα ήταν:
    - Music Transformer: $\frac{100*Training\ Batches}{Batch\ Size}$ βήματα
    - Perceiver-AR: 1.000 βήματα

Επιλογή Εκπαιδευμένων Μοντέλων

Κατά τη διάρκεια της εκπαίδευσης, αποθηκεύσαμε πολλαπλά στιγμιότυπα όλων των μοντέλων για κάθε έναν από τους 12 διαφορετικούς συνδυασμούς μοντέλου-συνόλου δεδομένων. Στη

συνέχεια αναλύσαμε τις καμπύλες μάθησης και επιλέξαμε το εκπαιδευμένο μοντέλο με την καλύτερη απόδοση για κάθε συνδυασμό με βάση τα ακόλουθα κριτήρια:

- Επίτευξη ελάχιστων τιμών σφάλματος επαλήθευσης.
- Ελαχιστοποίηση της διαφοράς μεταξύ των σφαλμάτων εκπαίδευσης και επαλήθευσης.
- Επίτευξη παρόμοιων επιδόσεων με λιγότερα βήματα εκπαίδευσης για την αποφυγή υπερπροσαρμογής.

Καμπύλες Μάθησης

Στην αντίστοιχη ενότητα παρουσιάζουμε τις καμπύλες μάθησης των σφαλμάτων εκπαίδευσης-επαλήθευσης για τους 12 διαφορετικούς συνδυασμούς εκπαιδευμένων μοντέλων.

Εκπαιδευμένα Μοντέλα

Παρουσιάζουμε τα επιλεγμένα μοντέλα για κάθε έναν από τους 12 διαφορετικούς συνδυασμούς μοντέλου-συνόλου δεδομένων στους ακόλουθους πίνακες.

Σύνολο Εκπαίδευσης	Training Batches	Βήματα Εκπαίδευσης	Αριθμός Εποχών	Σφάλμα Επαλήθευσης
MAESTRO 3	1,276	76,560	≈60	0.9793
GiantMIDI Piano	10,855	179,091	≈17	1.1543
Ailabs1k7	1,747	68,967	≈40	1.3368
Rock Piano MIDI	8,761	157,680	≈18	0.5572
ADL Piano MIDI	11,086	46,332	≈5	0.2999
Los Angeles MIDI segment	46,997	210,672	≈5	0.4161

Table 21: Επιλεγμένα Εκπαιδευμένα Μοντέλα Music-Transformer

Σύνολο Εκπαίδευσης	Training Batches	Βήματα Εκπαίδευσης	Αριθμός Εποχών	Σφάλμα Επαλήθευσης
MAESTRO 3	1,719	21,100	13	0.0098
GiantMIDI Piano	9,451	75,800	9	0.0192
Ailabs1k7	556	7,800	14	0.0313
Rock Piano MIDI	2,321	22,100	10	0.0063
ADL Piano MIDI	2,888	29,100	11	0.0259
Los Angeles MIDI segment	25,179	164,400	7	0.0037

Table 22: Επιλεγμένα Εκπαιδευμένα Μοντέλα Perceiver-AR

Υλοποίηση

Για το Music Transformer, ο κώδικας εκπαίδευσης περιλαμβάνεται στο αρχείο /music-transformer/train.py και βασίστηκε στον κώδικα από το αρχείο train.py του (K. Yang, 2021a), το οποίο εμπλουτίστηκε περαιτέρω με πρόσθετα χαρακτηριστικά και λειτουργίες, συμπεριλαμβανομένης της αποθήκευσης των μετρικών εκπαίδευσης και επαλήθευσης, της απεικόνισης γραφικών παραστάσεων των μετρικών, του ελέγχου για καλύτερες τιμές των μετρικών επιδόσεων, της εφαρμογής της πρόωρης διακοπής κ.λπ. Χρησιμοποιεί το TensorFlow

2.0.0 και εκτελέστηκε σε GPU NVIDIA Tesla K40m στον υπερυπολογιστή ARIS GRNET τον Δεκέμβριο του 2022, απαιτώντας περίπου 10-11 GB μνήμης GPU για τις επιλεγμένες υπερπαραμέτρους και συνολικό χρόνο 1-2 ημέρες για κάθε μοντέλο σχετιζόμενο με το μέγεθος του κάθε συνόλου εκπαίδευσης.

Για το Perceiver-AR, ο κώδικας εκπαίδευσης περιλαμβάνεται στο σημειωματάριο /perceiver-ar/Perceiver-Solo-Piano-Maker.ipynb και βασίστηκε στον κώδικα από το (Lev, 2022b) Perceiver-Solo-Piano-Maker.ipynb που εμπλουτίστηκε περαιτέρω με πρόσθετα χαρακτηριστικά και λειτουργίες, συμπεριλαμβανομένης της αποθήκευσης των μετρικών εκπαίδευσης και επαλήθευσης, του ελέγχου για τις καλύτερες τιμές μετρικών απόδοσης, της εφαρμογής της πρόωρης διακοπής κ.λπ. Χρησιμοποιεί το PyTorch 1.12.1 και εκτελέστηκε σε NVIDIA GeForce GTX 1080 Ti και NVIDIA TITAN Xp GPU στον server Pink-Floyd του Εργαστηρίου Τεχνητής Νοημοσύνης και Συστημάτων Μάθησης (AILS) της ΗΜΜΥ ΕΜΠ μεταξύ Νοεμβρίου - Δεκεμβρίου 2022 απαιτώντας περίπου 10-11 GB μνήμης GPU και συνολικό χρόνο εκπαίδευσης 2-4 ημέρες για κάθε μοντέλο σχετιζόμενο με το μέγεθος του συνόλου εκπαίδευσης.

Κεφάλαιο 7: Παραγωγή Εξόδων

Αριθμός Κομματιών: Παράχθηκαν 600 κομμάτια για κάθε μια από τις 12 περιπτώσεις εκπαιδευμένων μοντέλων. Επιλέχθηκε ένας μεγαλύτερος αριθμός εξόδων για κάθε περίπτωση μοντέλου καθώς υπάρχει η δυνατότητα μεγαλύτερης ποικιλομορφίας, η οποία μπορεί να οδηγήσει σε πιο δημιουργικά κομμάτια και παράλληλα περιορίζεται η επίδραση τυχαίων παραγόντων. Επιπλέον, ένας μεγαλύτερος αριθμός εξόδων μπορεί να διευκολύνει τον εντοπισμό μοτίβων στα κομμάτια.

Αρχικές Ακολουθίες (Primers): Χρησιμοποιήθηκε ένα σταθερό σύνολο 600 συνολικά ακολουθιών με 100 τυχαία κομμάτια από κάθε σύνολο δεδομένων εκπαίδευσης. Με αυτό το τρόπο διασφαλίζεται δίκαιη αντιπροσώπευση όλων των συνόλων εκπαίδευσης ανεξαρτήτως μεγέθους και καλύτερες δυνατότητες σύγκρισης μεταξύ των εξόδων των μοντέλων, ενώ τα μοντέλα εκτίθενται σε ένα ευρύτερο μουσικό υλικό από πολλά σύνολα δεδομένων, βελτιώνοντας ενδεχομένως την ικανότητά τους να παράγουν ποικιλομορφία κομματιών.

Αριθμός Γεγονότων Αρχικής Ακολουθίας & Εξόδου: Χρησιμοποιήθηκε ο συνδυασμός 512 γεγονότων αρχικής ακολουθίας και 512 γεγονότων εξόδου. Επιλέχθηκε αυτός ο συνδυασμός καθώς, αρχικά, η επιθυμητή διάρκεια των μουσικών κομματιών ήταν 10 έως 60 δευτερόλεπτα ώστε να είναι κατάλληλα για την υποκειμενική αξιολόγηση. Επιπρόσθετα, σύμφωνα με τα πειράματά μας, το μοντέλο Perceiver-AR απέδωσε καλύτερα με μέγεθος primer 512 και μέγεθος εξόδου 512 tokens. Όσον αφορά το μοντέλο Music Transformer, δεν παρατηρήσαμε καμία σημαντική διαφοροποίηση για διαφορετικές διαμορφώσεις primer tokens και output tokens, με το συνδυασμό 512-512 να παράγει ικανοποιητικά κομμάτια.

Λειτουργία Παραγωγής Εξόδων: Για το Music Transformer η λειτουργία παραγωγής εξόδων ήταν προκαθορισμένη, ενώ για το Perceiver-AR επιλέχθηκε ο τρόπος λειτουργίας Single Continuation Block Generator. Σύμφωνα με αυτόν δίνεται ως είσοδος στο μοντέλο η αρχική ακολουθία και

παράγεται ο καθορισμένος αριθμός γεγονότων μονομιάς ως ένα ενιαίο σύνολο γεγονότων. Η επιλογή αυτή έγινε καθώς αυτός ο τρόπος λειτουργίας ήταν ίδιος με αυτόν που χρησιμοποιείται στο Music Transformer και ήταν επιθυμητό να υπάρχει συνέπεια στα δύο μοντέλα, ενώ παράλληλα εξασφαλίζει έλεγχο του αριθμού των γεγονότων εξόδου που θα παραχθούν.

Υλοποίηση

Ο κώδικας επιλογής αρχικών ακολουθιών περιλαμβάνεται στο σημειωματάριο /primer-selection.ipynb.

Η υλοποίηση της παραγωγής εξόδων για το Music-Transformer περιλαμβάνεται στο αρχείο /music-transformer/batch_generate.py και βασίστηκε στον κώδικα του generate.py του (K. Yang, 2021a) που εμπλουτίστηκε περαιτέρω ώστε να περιλαμβάνει λειτουργικότητα μαζικής παραγωγής εξόδων. Ο κώδικας παραγωγής εξόδων εκτελέστηκε σε GPU NVIDIA GeForce GTX 1080 στον server Ironman του Εργαστηρίου Τεχνητής Νοημοσύνης και Συστημάτων Μάθησης (AILS) της ΗΜΜΥ ΕΜΠ μεταξύ Δεκεμβρίου 2022 και Ιανουαρίου 2023, απαιτώντας περίπου 6-7 GB μνήμης GPU.

Η υλοποίηση της δημιουργίας εξόδων για το Perceiver-AR περιλαμβάνεται στο αρχείο /perceiver-ar/perceiver-solo-piano-batch.py και βασίστηκε στον κώδικα από το σημειωματάριο Perceiver_Solo_Piano.ipynb του (Lev, 2022b) που εμπλουτίστηκε περαιτέρω ώστε να περιλαμβάνει τη λειτουργικότητα μαζικής παραγωγής εξόδων. Ο κώδικας παραγωγής εξόδων εκτελέστηκε σε GPU NVIDIA GeForce GTX 1080 στον server Ironman του Εργαστηρίου Τεχνητής Νοημοσύνης και Συστημάτων Μάθησης (AILS) της ΗΜΜΥ ΕΜΠ μεταξύ Δεκεμβρίου 2022 και Ιανουαρίου 2023, απαιτώντας περίπου 6-7 GB μνήμης GPU.

Κεφάλαιο 8: Αντικειμενική Αξιολόγηση

Αντικειμενικές Μετρικές

Μετά τη παραγωγή εξόδων για κάθε περίπτωση μοντέλου, εφαρμόσαμε δύο στρατηγικές αξιολόγησης των κομματιών. Η μία από αυτές ήταν η αντικειμενική αξιολόγηση με μουσικές μετρικές, για να διερευνήσουμε τα χαρακτηριστικά των παραγόμενων μουσικών κομματιών μας και να κάνουμε αντίστοιχες συγκρίσεις. Οι μετρικές που χρησιμοποιήσαμε είναι υλοποιημένες στα (H.-W. Dong et al., 2020) Muspy and (Raffel & Ellis, 2014) pretty_midi.

Στον επόμενο πίνακα παρουσιάζουμε τις αντικειμενικές μετρικές που χρησιμοποιήσαμε στην αντικειμενική μας αξιολόγηση, συμπεριλαμβανομένης της περιγραφής κάθε μετρικής. Στην συνέχεια εξηγούμε τους λόγους και τους στόχους κάθε επιλογής.

Μετρική	Υλοποίηση	Περιγραφή
Pitches Used	Muspy	Αριθμός διαφορετικών τονικών υψών σε ένα κομμάτι.
Pitch Range	Muspy	Διαφορά του υψηλότερου και χαμηλότερου τονικού ύψους σε ημιτόνια.
Pitch Classes Used	Muspy	Αριθμός μοναδικών κλάσεων τονικών υψών.
Polyphony	Muspy	Μέσος αριθμός τονικών υψών που παίζεται ταυτόχρονα.

Polyphony Rate	Muspy	Ποσοστό του χρόνου που τουλάχιστον 2 τονικά ύψη παίζονται ταυτόχρονα.
Scale Consistency	Muspy	Μεγαλύτερο ποσοστό pitch-in-scale σε όλες τις μείζονες και ελάσσονες κλίμακες.
Pitch Entropy	Muspy	Εντροπία Shannon του κανονικοποιημένου ιστογράμματος τονικών υψών.
Pitch Class Entropy	Muspy	Εντροπία Shannon του κανονικοποιημένου ιστογράμματος κλάσης τονικού ύψους.
Empty Beat Rate	Muspy	Αναλογία των χρονικών παλμών που δε παίζεται καμία νότα προς το συνολικό αριθμό παλμών.
Tempo Estimation	Pretty-Midi	Βέλτιστη εμπειρική εκτίμηση του ρυθμού.

Table 23: Επιλεγμένες Αντικειμενικές Μετρικές

Κατά την επιλογή αντικειμενικών μετρικών λάβαμε υπόψη μας πολλούς παράγοντες. Πρώτα απ' όλα, ένας από τους σημαντικότερους παράγοντες ήταν οι διαθέσιμες μουσικές πληροφορίες στα σύνολα δεδομένων και εξόδων μας. Επιπλέον, θέλαμε να πραγματοποιήσουμε μια γενική στατιστική ανάλυση στα σύνολα δεδομένων και εξόδων και όχι να επικεντρωθούμε σε μεμονωμένα κομμάτια που θα μπορούσαν να οδηγήσουν σε παραπλανητικά συμπεράσματα για κάποιο σύνολο. Τέλος, προσπαθήσαμε να αποκλείσουμε τις μετρικές που σχετίζονται άμεσα με τη χρονική διάρκεια του κομματιού, καθώς τα σύνολα εκπαίδευσης έχουν ποικίλα μήκη μουσικών κομματιών ενώ οι έξοδοι μας περιορίζονται σε διάρκεια περίπου ενός λεπτού. Ωστόσο, δεν μπορέσαμε να αποκλείσουμε εντελώς τον παράγοντα της διάρκειας από όλες τις μετρικές μας.

Στατιστικές Τιμές Αντικειμενικών Μετρικών

Για την αντικειμενική στατιστική ανάλυση των συνόλων δεδομένων και εξόδων δημιουργήσαμε έναν πίνακα που παρουσιάζεται στην αντίστοιχη ενότητα και περιλαμβάνει τις τιμές κάθε μετρικής και συνδυασμού συνόλων δεδομένων. Τα αποτελέσματά μας για κάθε μετρική περιλαμβάνουν τη μέση τιμή (mean), τη διάμεσο (median) και την τυπική απόκλιση (std). Συμπεριλάβαμε τη διάμεσο καθώς, σε αντίθεση με τη μέση τιμή, δεν επηρεάζεται σε μεγάλο βαθμό από ακραίες τιμές.

Συσχετίσεις Αντικειμενικών Μετρικών

Δημιουργήσαμε επίσης τη γραφική παράσταση της συσχέτισης Kendall των αντικειμενικών μετρικών για την εύρεση μονοτονικών σχέσεων μεταξύ τους, η οποία παρουσιάζεται στην αντίστοιχη ενότητα. Οι μετρικές μας δεν έχουν κανονικές κατανομές, ενώ ορισμένες από αυτές είναι επίσης ασυνεχείς. Επίσης, θέλαμε να διερευνήσουμε γενικά μονοτονικές σχέσεις μεταξύ τους και όχι απαραίτητα γραμμικές. Για τους παραπάνω λόγους η συσχέτιση του Kendall ήταν η προτιμώμενη μέθοδος (Zinda, 2021).

Γραφικές Παραστάσεις Αντικειμενικών Μετρικών

Αφού υπολογίσαμε τις στατιστικές τιμές για τις αντικειμενικές μετρικές, συγκρίνουμε τις κατανομές όλων των συνόλων δεδομένων με δύο τύπους γραφικών παραστάσεων, το bar-plot και το violin-plot. Το διάγραμμα bar-plot παρέχει μια απλούστερη αξιολόγηση και σύγκριση των αποτελεσμάτων των συνόλων δεδομένων. Ως μέθοδο υπολογισμού του σφάλματος

χρησιμοποιήσαμε την τυπική απόκλιση (std). Από την άλλη πλευρά, το διάγραμμα violin-plot παρέχει μια πιο σύνθετη παρουσίαση των κατανομών κάθε συνόλου δεδομένων.

Στην αντίστοιχη ενότητα παρουσιάζουμε τα δύο είδη διαγραμμάτων για όλες τις αντικειμενικές μουσικές μετρικές.

Ανάλυση Αντικειμενικών Μετρικών

Σε αυτό το υποκεφάλαιο δημιουργήσαμε έναν συνοπτικό πίνακα ανάλυσης με βάση τον πίνακα μετρήσεων και τα διαγράμματα, ο οποίος παρουσιάζεται στην αντίστοιχη ενότητα. Για τη σύγκριση των διαφόρων συνόλων χρησιμοποιούνται περιγραφές του μέσου όρου (mean), της τυπικής απόκλισης (std), της κατανομής (distribution) και των ακραίων τιμών (outliers). Παρέχουμε ορισμένες συγκριτικές περιγραφές για να βοηθήσουμε στη κατανόηση του βαθμού ομοιότητας μεταξύ των συνόλων εξόδων και των αντίστοιχων συνόλων εκπαίδευσης.

Εάν τα σύνολα εξόδων ενός μοντέλου είναι παρόμοια με τα σύνολα εκπαίδευσης, χρησιμοποιούμε την ένδειξη "Similar". Εάν υπάρχουν κάποιες εξαιρέσεις, αλλά η πλειονότητα των συνόλων εξόδων εξακολουθεί να είναι παρόμοια με τα σύνολα εκπαίδευσης, χρησιμοποιούμε την ένδειξη "Mostly similar". Εάν η μετρική δείχνει ότι τα σύνολα εξόδων είναι πιο κοντά στα αντίστοιχα σύνολα εκπαίδευσης από ό,τι οι έξοδοι του άλλου μοντέλου, χρησιμοποιούμε την ένδειξη "Closer". Ωστόσο, είναι σημαντικό να σημειωθεί ότι το να είναι πιο κοντά δεν σημαίνει απαραίτητα ότι είναι παρόμοια με τα σύνολα εκπαίδευσης.

Επιπλέον, θέλουμε να αναφέρουμε ότι οι διαφορές στις μετρικές που σχετίζονται με τονικά ύψη (pitch-related) μπορεί να οφείλονται εν μέρει σε διαφορές στη διάρκεια των μουσικών κομματιών τόσο στα σύνολα εκπαίδευσης όσο και στα σύνολα εξόδων.

Υλοποίηση

Η υλοποίηση αυτού του μέρους της εργασίας μας περιλαμβάνεται στο αρχείο notebook obj_eval.ipynb.

Κεφάλαιο 9: Υποκειμενική Αξιολόγηση

Αρχές Υποκειμενικής Αξιολόγησης

Οι (Hernandez-Olivan, Puyuelo, et al., 2022; Ji et al., 2020; L.-C. Yang & Lerch, 2020) αποτέλεσαν τις βασικές αναφορές μας σε αυτό το υποκεφάλαιο.

Η υποκειμενική ή ανθρώπινη αξιολόγηση αναγνωρίζεται ευρέως ως ο πιο αξιόπιστος τρόπος αξιολόγησης της ποιότητας παραγόμενης μουσικής από μοντέλα τεχνητής νοημοσύνης, επειδή ο στόχος των μοντέλων αυτών είναι να συνθέσουν μουσική βάσει των ανθρώπινων αισθητικών κριτηρίων. Επιπλέον, όπως επισημαίνουν οι (Ji et al., 2020), δεν έχει αποδειχθεί στη σχετική βιβλιογραφία κάποια συσχέτιση μεταξύ μετρικών ποσοτικής αξιολόγησης και της υποκειμενικής αξιολόγησης.

Η πιο συνήθης μέθοδος για υποκειμενική αξιολόγηση είναι η έρευνα δοκιμής ακρόασης (listening test). Ωστόσο, όπως επισημαίνεται από τους (Ji et al., 2020), η διεξαγωγή μιας τέτοιας έρευνας απαιτεί την συνυπολογισμό πολλών παραγόντων, συμπεριλαμβανομένης της επιλογής και της παρουσίασης των δειγμάτων ήχου, του περιβάλλοντος ακρόασης, της επιλογής των συμμετεχόντων και του περιεχομένου των ερωτήσεων. Συγκεκριμένα απαιτείται: 1) να υπάρχουν αρκετοί συμμετέχοντες με επαρκή ποικιλομορφία για στατιστικά σημαντικά αποτελέσματα και 2) το επίπεδο μουσικών γνώσεων των συμμετεχόντων να είναι ομοιόμορφα κατανεμημένο, περιλαμβάνοντας τόσο άτομα που δεν έχουν μουσικές γνώσεις όσο και άτομα που έχουν κάποια μουσική εμπειρία και βασική γνώση μουσικής θεωρίας.

Ένας άλλος σημαντικός παράγοντας για τη διεξαγωγή υποκειμενικής αξιολόγησης είναι η υιοθέτηση απόλυτων μετρικών που δεν βασίζονται στην προτίμηση ενός μοντέλου έναντι ενός άλλου, όπως υποστηρίζουν οι (L.-C. Yang & Lerch, 2020). Ένας ακόμα παράγοντας είναι η κόπωση ακρόασης (listening fatigue) που μπορεί να αυξήσει το σφάλμα των αξιολογήσεων. Τέλος, είναι υψίστης σημασίας να ορίζονται οι ερωτήσεις της έρευνας με σαφήνεια και πληρότητα, επεξηγώντας το σκοπό και το περιεχόμενο τους όπου χρειάζεται.

Μια από τις σημαντικότερες και απλούστερες προσεγγίσεις για την υποκειμενική αξιολόγηση χρησιμοποιώντας απόλυτη μετρική είναι το προσαρμοσμένο τεστ Turing, με τους συμμετέχοντες να κρίνουν αν ένα κομμάτι είναι ανθρώπινη σύνθεση ή σύνθεση τεχνητής νοημοσύνης. Άλλες προσεγγίσεις σύμφωνα με τους (Ji et al., 2020) περιλαμβάνουν την αξιολόγηση μουσικών δειγμάτων που έχουν παραχθεί από διαφορετικά μοντέλα με βάση διάφορες μετρικές αξιολόγησης. Συνδυάζουμε αυτές τις δύο προσεγγίσεις για μια πιο ολοκληρωμένη αξιολόγηση της απόδοσης των μοντέλων και της ποιότητας της παραγόμενης μουσικής.

Μετρικές & Ερωτήσεις Έρευνας

Για το σχεδιασμό των ερωτήσεων της έρευνας βασιστήκαμε κυρίως στην προτεινόμενη δομή και μορφή από τους (Hernandez-Olivan, Puyuelo, et al., 2022) και στα ευρήματα των (Chu et al., 2022).

Αποφασίσαμε να ορίσουμε δύο κατηγορίες υποκειμένων με βάση τα διαφορετικά επίπεδα μουσικών γνώσεων: α) Βασικό υποκείμενο (basic), για άτομα χωρίς μουσικές γνώσεις, και β) Έμπειρο υποκείμενο (pro), για συμμετέχοντας με βασικές μουσικές γνώσεις και υπόβαθρο.

Όσον αφορά τις μετρικές που επιλέξαμε:

- **Συναίσθημα**: Αποτελεί σημαντικό παράγοντα για τους ακροατές για την αξιολόγησης της ποιότητας ενός κομματιού. Εξετάζουμε την ένταση των συναισθημάτων που δημιουργούνται και όχι στο ποια συγκεκριμένα συναισθήματα δημιουργούνται.
- **Μελωδικότητα, Αρμονικότητα, Ρυθμικότητα**: Αποτελούν θεμελιώδεις μουσικές πτυχές ενός κομματιού και μπορούν να αναγνωριστούν από μουσικά έμπειρους ακροατές.
- **Είδος**: Αναφέρεται στο βαθμό στον οποίο μουσικά έμπειροι συμμετέχοντες μπορούν να αναγνωρίσουν το μουσικό είδος του κομματιού. Έχοντας εκπαιδεύσει τα μοντέλα μας σε διαφορετικά μουσικά είδη και χρησιμοποιώντας επίσης αρχικές ακολουθίες διαφορετικών ειδών, εξετάζουμε αν στοιχεία είδους είναι αναγνωρίσιμα στα παραγόμενα κομμάτια.

- **Οικειότητα**: Αποτελεί σημαντικό παράγοντα για τους ακροατές για την αξιολόγησης της ποιότητας ενός κομματιού.
- **Φυσικότητα:** Υλοποιήσαμε αυτή τη μετρική μέσω του προσαρμοσμένου τεστ Turing.
- **Συνολική Αξιολόγηση**: Συνοψίζει την αντίληψη του υποκειμένου για το συγκεκριμένο κομμάτι.

Σύμφωνα με τις μουσικές μετρικές που περιγράφονται παραπάνω, παρουσιάζουμε στον παρακάτω πίνακα τις ερωτήσεις της έρευνας τόσο για τα βασικά όσο και για τα έμπειρα υποκείμενα.

Μουσική Γνώση Υποκειμένου	Ερώτηση	Μετρική	Εύρος Αξιολόγησης
Βασικό Υποκείμενο	Σου δημιουργεί αυτή η μουσική συναισθήματα;	Συναίσθημα	1-5
	Έχεις ξανακούσει παρόμοια μουσική;	Οικειότητα	1-5
	Έχει αυτή η μουσική συντεθεί από άνθρωπο ή από τεχνητή νοημοσύνη (AI);	Φυσικότητα	AI/HC (0-1)
	Συνολική Αξιολόγηση Μουσικού Κομματιού	Συνολική Αξιολόγηση	1-5
Έμπειρο Υποκείμενο	Σου δημιουργεί αυτή η μουσική συναισθήματα;	Συναίσθημα	1-5
	Η μελωδία εξελίσσεται ομαλά.	Μελωδικότητα	1-5
	Η αρμονία είναι ευχάριστη.	Αρμονικότητα	1-5
	Ο ρυθμός είναι συνεπής και ομαλός.	Ρυθμικότητα	1-5
	Μπορείς να αναγνωρίσεις το μουσικό είδος του κομματιού;	Μουσικό Είδος	1-5
	Έχει αυτή η μουσική συντεθεί από άνθρωπο ή από τεχνητή νοημοσύνη (AI);	Φυσικότητα	AI/HC (0-1)
	Συνολική Αξιολόγηση Μουσικού Κομματιού	Συνολική Αξιολόγηση	1-5

Επιλογή & Επεξεργασία Δειγμάτων

Σε αυτό το υποκεφάλαιο, θα εμβαθύνουμε στην επιλογή και την επεξεργασία των δειγμάτων που χρησιμοποιήθηκαν στην έρευνα υποκειμενικής αξιολόγησης.

Αρχικά, επιλέξαμε τα σύνολα εξόδων που παράχθηκαν χωρίς τις αρχικές ακολουθίες (7,200 συνολικά κομμάτια), καθώς λόγω του προσαρμοσμένου τεστ Turing οι συμμετέχοντες δε θα έπρεπε να μπορούν να αναγνωρίσουν γνωστές αρχικές ακολουθίες και να συμπεράνουν ότι τα παραγόμενα κομμάτια είναι τεχνητές συνθέσεις. Στη συνέχεια προσθέσαμε ανθρώπινες συνθέσεις απαραίτητες για το τεστ Turing (7,800 συνολικά κομμάτια), επιλέγοντας τις συνέχειες των κομματιών των συνόλων δεδομένων που χρησιμοποιήθηκαν για αρχικές ακολουθίες με ίδιο αριθμό 512 γεγονότων όπως και στις εξόδους των μοντέλων.

Επίσης, φιλτράραμε τα κομμάτια ανάλογα με τη διάρκεια τους κρατώντας κομμάτια 10 έως 60 δευτερολέπτων (6,106 συνολικά κομμάτια). Θέλαμε αφενός να έχουν επαρκή αλλά όχι και πολύ μεγάλη διάρκεια ώστε η έρευνα μας να ολοκληρώνεται σε ένα καθορισμένο μικρό χρονικό διάστημα. Στη συνέχεια πραγματοποιήσαμε τυχαία δειγματοληψία επιλέγοντας 48 κομμάτια από κάθε εκπαιδευμένο μοντέλο, με ίσο αριθμό 8 κομματιών με αρχική ακολουθία από το κάθε

σύνολο δεδομένων. Αντίστοιχα για τις ανθρώπινες συνθέσεις επιλέξαμε 16 κομμάτια από κάθε σύνολο δεδομένων. Συνοψίζοντας προέκυψαν 672 συνολικά δείγματα με 576 εξόδους μοντέλων και 96 ανθρώπινες συνθέσεις, και ποσοστό ανθρώπινων συνθέσεων 14.29%.

Τέλος, μετατρέψαμε τα δείγματα από τη μορφή MIDI σε αρχεία ήχου. Αυτό συνέβη καθώς η μορφή MIDI δεν είναι καθολικά υποστηριζόμενη μορφή ήχου στα προγράμματα περιήγησης, με αποτέλεσμα τα κομμάτια να παίζονται με διαφορετικό τρόπο λόγω διαφορετικών προσθέτων και προτύπων ήχου. Αντίθετα για τα αρχεία ήχου δεν υπάρχει αυτός ο ανεπιθύμητος παράγοντας.

Ιστοσελίδα Έρευνας

Για να αντιμετωπιστεί η πρόκληση της αξιολόγησης του μεγάλου αριθμού μουσικών δειγμάτων από πολλούς συμμετέχοντες, δημιουργήθηκε η ιστοσελίδα διαδικτυακής έρευνας https://ai-music-generation-survey.vercel.app/. Η σελίδα επιτρέπει στους συμμετέχοντες να αξιολογήσουν ένα τυχαίο σύνολο 6 κομματιών, με τον αριθμό των 6 δειγμάτων να επιλέγεται για να διατηρηθεί ο χρόνος συμπλήρωσης της έρευνας σύντομος στα περίπου 5 λεπτά.

Στη συνέχεια αναφέρουμε μερικά σημαντικά στοιχεία της ιστοσελίδας. Αρχικά για να προσδιοριστεί η μουσική εμπειρία του χρήστη, προκειμένου να κατανεμηθεί σε μία από τις δύο ομάδες, ο χρήστης ερωτάται αν έχει παίξει ποτέ μουσικό όργανο ή αν έχει σπουδάσει σε μουσικό ωδείο. Επίσης, του ζητείται να προσδιορίσει αν ένα μουσικό κομμάτι είναι σε μείζονα ή ελάσσονα κλίμακα. Αυτή η βασική ερώτηση μουσικής γνώσης αποτελεί ένα μικρό τεστ, ώστε να αξιολογήσουμε την αξιοπιστία των χρηστών που ισχυρίζονται ότι έχουν μουσική εμπειρία.

Στον χρήστη παρουσιάζονται οι ερωτήσεις ανάλογα με την ομάδα μουσικών γνώσεων του. Κάθε κομμάτι παρουσιάζεται ξεχωριστά και για να υποβληθεί η αξιολόγηση του κομματιού ο χρήστης πρέπει να ακούσει ένα ποσοστό του κομματιού πάνω από 80% και επίσης να συμπληρώσει όλες τις ερωτήσεις. Οι δύο αυτές απαιτήσεις αποτελούν βασικά μέτρα διασφάλισης της αξιοπιστίας των αξιολογήσεων, αποφεύγοντας κακόβουλους χρήστες, αυτοματοποιημένες ή ημιτελείς απαντήσεις.

Στατιστικές Τιμές Υποκειμενικών Μετρικών

Συγκεντρώσαμε συνολικά 1.380 αξιολογήσεις μεταξύ της 21ης Ιανουαρίου 2023 και της 1ης Μαρτίου 2023, οι οποίες αποτελούνται από 674 αξιολογήσεις από αρχάριους συμμετέχοντες και 706 αξιολογήσεις από έμπειρους συμμετέχοντες. Όσον αφορά τη φυσικότητα, οι αρχάριοι συμμετέχοντες παρουσίασαν ακρίβεια 0,556, ενώ οι έμπειροι συμμετέχοντες ακρίβεια 0,626.

Για την υποκειμενική στατιστική ανάλυση των συνόλων εκπαίδευσης και εξόδων δημιουργήσαμε έναν πίνακα στατιστικών τιμών για κάθε ομάδα υποκειμένων που περιλαμβάνει κάθε μετρική και συνδυασμό συνόλων δεδομένων και παρουσιάζεται στην αντίστοιχη ενότητα. Τα αποτελέσματά μας για κάθε μετρική περιλαμβάνουν τη μέση τιμή (mean) και την τυπική απόκλιση (std). Η στήλη count αντιπροσωπεύει τον συνολικό αριθμό των αξιολογήσεων που ελήφθησαν για κάθε συνδυασμό μοντέλου-συνόλου δεδομένων.

Επιπλέον, για μια συνολική σύγκριση των επιδόσεων των υποκειμενικών μετρικών των συνόλων εκπαίδευσης-εξόδων, παρουσιάζουμε στην αντίστοιχη ενότητα τον πίνακα με τις σταθμισμένες

μέσες τιμές όλων των μετρικών προσαρμοσμένες στην κλίμακα Likert 1-5. Από τον υπολογισμό των μέσων όρων εξαιρέσαμε τη μετρική είδους καθώς δεν είναι εφαρμόσιμη σε σύνολα δεδομένων πολλαπλών ειδών. Η στήλη "weighted mean" λαμβάνει υπόψη το ποσοστό των απαντήσεων από κάθε ομάδα υποκειμένων και τον αριθμό των μετρικών που αξιολογεί κάθε ομάδα υποκειμένων. Επιπλέον, στην τελευταία στήλη, ο σταθμισμένος μέσος όρος των αποτελεσμάτων των μοντέλων ενσωματώνει επίσης το ποσοστό των απαντήσεων από κάθε τύπο μοντέλου.

Συσχετίσεις Υποκειμενικών Μετρικών

Δημιουργήσαμε επίσης τη γραφική παράσταση της συσχέτισης Kendall των υποκειμενικών μετρικών για την εύρεση μονοτονικών σχέσεων μεταξύ τους, η οποία παρουσιάζεται στην αντίστοιχη ενότητα. Οι λόγοι για τη χρήση της συσχέτισης Kendall είναι παρόμοιοι με αυτούς που αναφέρθηκαν στην ενότητα Συσχετίσεις Αντικειμενικών Μετρικών.

Γραφικές Παραστάσεις Υποκειμενικών Μετρικών

Αφού υπολογίσαμε τις στατιστικές τιμές για τις υποκειμενικές μετρικές, συγκρίνουμε τις μέσες τιμές των συνόλων δεδομένων γραφικά με ραβδογράμματα. Ως μέθοδο υπολογισμού του σφάλματος χρησιμοποιήσαμε την τυπική απόκλιση (std). Στην αντίστοιχη ενότητα παρουσιάζουμε επτά κατηγορίες γραφημάτων που διαφοροποιούνται ανάλογα με την μεταβλητή εστίασης.

Υλοποίηση

Το τμήμα αυτό της εργασίας υλοποιείται στο αρχείο Notebook sub_eval.ipynb. Η ιστοσελίδα της έρευνας είναι https://ai-music-generation-survey.vercel.app/, με τον κώδικα της να βρίσκεται στο https://github.com/aggelostais/ai-music-generation-survey. Βασίστηκε στον κώδικα του (Filandrianos, 2019) ενώ αναπτύχθηκε περαιτέρω για να συμπεριλάβει πρόσθετα χαρακτηριστικά και λειτουργίες. Η σελίδα υλοποιήθηκε χρησιμοποιώντας το Flask σε Python. Τα δείγματα μουσικής και οι βαθμολογίες που συλλέγονται αποθηκεύονται στο Firebase Storage. Τέλος, η έρευνα αναπτύσσεται (deployment) στο Vercel ως Serverless Function.

Μέρος ΙΙΙ: Συμπεράσματα & Συζήτηση

Κεφάλαιο 10: Συμπεράσματα

Σύνολα Εκπαίδευσης

Μέσω της διαδικασίας εκπαίδευσης και αξιολόγησης, αξιολογήσαμε τα σύνολα δεδομένων σε διάφορες μετρικές και εντοπίσαμε συγκεκριμένα χαρακτηριστικά και ιδιότητες τους που έχουν σημαντικό αντίκτυπο στην εκπαίδευση των μοντέλων και στην ποιότητα των παραγόμενων κομματιών.

Εκπαίδευση Μοντέλων

Όσον αφορά την εκπαίδευση, αρχικά, τα μοντέλα ailabs1k7 και maestro-3.0.0 συγκλίνουν πιο σταθερά σε σύγκριση με τα μοντέλα των άλλων συνόλων, έχοντας λιγότερες διακυμάνσεις στα σφάλματα εκπαίδευσης και επαλήθευσης. Επίσης, τα μοντέλα Rock-MIDI-Piano κυρίως και τα μοντέλα Los-Angeles-MIDI-segment έχουν πιο ασταθή απόδοση σε σύγκριση με τα άλλα μοντέλα, έχοντας περισσότερες διακυμάνσεις στα σφάλματα εκπαίδευσης-επαλήθευσης. Τέλος, τα μικρότερα σύνολα δεδομένων συγκλίνουν σε μικρότερο αριθμό βημάτων σε σύγκριση με μεγαλύτερα σύνολα δεδομένων.

Υποκειμενική Αξιολόγηση

Όσον αφορά την Εξοικείωση (Familiarity), τα σύνολα δεδομένων παρουσίασαν μέτριες τιμές κυρίως πάνω από τη μέση. Οι τιμές αυτές μπορούν να αποδοθούν, πρώτον, σε σύνολα δεδομένων με σημαντικό αριθμό ατελών ή πρόχειρων κομματιών με ασυνέχειες, μεγάλες παύσεις ή θόρυβο. Επιπλέον, στα σύνολα δεδομένων ενός μουσικού είδους, αναμενόταν μεγαλύτερη εξοικείωση των χρηστών με τα μουσικά κομμάτια, καθώς αυτά τα είδη, κλασική, ποπ και ροκ, είναι ευρέως αναγνωρίσιμα. Αντίθετα, τα σύνολα δεδομένων με πολλαπλά είδη παρουσίασαν χαμηλότερες τιμές, καθώς περιλάμβαναν και λιγότερα γνωστά μουσικά είδη.

Όσον αφορά το Συναίσθημα (Emotion), τα σύνολα δεδομένων είχαν μέτριες τιμές γύρω από τη μέση. Αυτές οι τιμές μπορούν εν μέρει να αποδοθούν στο γεγονός ότι τα συγκεκριμένα σύνολα δεδομένων δεν προσανατολίζονται ειδικά για τη δημιουργία συναισθηματικά πλούσιας μουσικής.

Όσον αφορά τη Μελωδικότητα (Melodiousness), οι τιμές των συνόλων δεδομένων ήταν μέτριες, ως επί το πλείστον πάνω από τη μέση, χωρίς όμως να επιτυγχάνονται οι επιθυμητές υψηλότερες τιμές. Ένας από τους παράγοντες που συμβάλει σε αυτό είναι η παρουσία περισσότερων μελωδικά ατελών κομματιών σε ορισμένα σύνολα δεδομένων.

Όσον αφορά την Αρμονικότητα (Harmonicity), οι τιμές των συνόλων εκπαίδευσης ήταν ικανοποιητικές και πάνω από τη μέση στις περισσότερες περιπτώσεις. Τα σύνολα δεδομένων που περιείχαν περισσότερα κομμάτια χαμηλότερης ποιότητας με αρμονική ασυνέπεια είχαν χαμηλότερες τιμές.

Όσον αφορά τη Ρυθμικότητα (Rhythmicity), τα σύνολα εκπαίδευσης παρουσιάζουν μέτριες τιμές γύρω από τη μέση. Σχετικά καλύτερες τιμές παρατηρούνται στα ailabs1k7 και Rock-Piano-MIDI που αιτιολογείται από το γεγονός ότι τα είδη αυτών των συνόλων δεδομένων έχουν απλούστερα και πιο εμφανή ρυθμικά μοτίβα. Συνακόλουθα, οι χαμηλότερες τιμές μπορούν εν μέρει να αποδοθούν σε ατελή ή πρόχειρα κομμάτια με ασυνέχειες ρυθμού ή μεγάλες παύσεις.

Όσον αφορά το Είδος (Genre), δεν είναι σαφώς αναγνωρίσιμο σε σύνολα δεδομένων ενός είδους, παρουσιάζοντας μέτριες τιμές κυρίως κάτω από τη μέση. Η αναγνώριση είδους αποτελεί δύσκολη πρόκληση στη μουσική για πιάνο, όπου οι διαφορές είδους δεν είναι τόσο σαφείς όσο στην πολύ-οργανική μουσική. Επιπλέον, ακόμη και τα μουσικά έμπειρα υποκείμενα μπορεί να έχουν περιορισμένη εξοικείωση με τα είδη της μουσικής πιάνου και τα ιδιαίτερα χαρακτηριστικά τους.

Όσον αφορά τη Φυσικότητα (Naturalness), τα σύνολα εκπαίδευσης έλαβαν τις χαμηλότερες τιμές σε σύγκριση με όλες τις υποκειμενικές μετρικές, παρουσιάζοντας τιμές κάτω από τη μέση. Τέλος για τη συνολική αξιολόγηση (Overall Rating), παρουσιάζουν ικανοποιητικές τιμές κοντά στη μέση.

Κύρια Ευρήματα

Με βάση τον πίνακα σταθμισμένων μέσων όρων, μεταξύ των συνόλων εκπαίδευσης τα ailabs1k7 (3.31), Rock-Piano-MIDI (3.06) και GiantMIDI-Piano (3.01) αξιολογήθηκαν υποκειμενικά καλύτερα, ενώ τα adl-piano-midi (2.80) και Los-Angeles-MIDI-segment (2.67) αξιολογήθηκαν σημαντικά χαμηλότερα. Όσον αφορά τα σύνολα εξόδων των μοντέλων, τα μοντέλα που εκπαιδευτήκαν με τα ailabs1k7 (3.01), Los-Angeles-MIDI-segment (3.00) και adl-piano-midi (2.98) είχαν τις υψηλότερες αξιολογήσεις, με τα δύο τελευταία να ξεπερνούν σημαντικά τις αξιολογήσεις των αρχικών συνόλων εκπαίδευσης.

Γενικά, η ποιότητα των συνόλων εκπαίδευσης είναι κάτω από τις αναμενόμενες προδιαγραφές. Οι έξοδοι των μοντέλων έχουν καλύτερη απόδοση από τα σύνολα δεδομένων σε πολλές μετρικές, παρόλο που τα σύνολα εκπαίδευσης αποτελούνται από ανθρώπινες συνθέσεις, γεγονός που υποδεικνύει την ανάγκη βελτίωσης των συνόλων εκπαίδευσης.

Ένας σημαντικός αριθμός ατελών και ανοργάνωτων κομματιών σε ένα σύνολο δεδομένων έχει σημαντικό αρνητικό αντίκτυπο στην ποιότητα των παραγόμενων κομματιών. Η παρουσία ασυνεχειών, μεγάλων παύσεων και θορύβου σε ατελή ή πρόχειρα κομμάτια οδηγεί σε παρόμοια προβλήματα στα παραγόμενα κομμάτια. Οι υποκειμενικές μετρικές που επηρεάζονται περισσότερο από αυτό το παράγοντα περιλαμβάνουν τη μελωδικότητα, την αρμονικότητα, τη ρυθμικότητα και τη φυσικότητα. Για τον μερικό εντοπισμό συνόλων δεδομένων με αυτό το χαρακτηριστικό, μπορεί να χρησιμοποιηθεί η αντικειμενική μετρική Empty Beat Rate. Τα σύνολα δεδομένων με αυτό το χαρακτηριστικό τείνουν να έχουν υψηλότερες μέσες τιμές ή περισσότερες ακραίες υψηλές τιμές στα σύνολα εκπαίδευσης ή εξόδων στο Empty Beat Rate. Για παράδειγμα, τα adl-piano-midi και Los-Angeles-MIDI-segment είναι σύνολα δεδομένων που εμφανίζουν αυτό το χαρακτηριστικό πιο συχνά, ενώ το ailabs1k7 που έχει ελάχιστο Empty Beat Rate δεν το εμφανίζει καθόλου.

Τα σύνολα δεδομένων πολλαπλών ειδών ενισχύουν τη γενίκευση των μοντέλων σε σύγκριση με σύνολα δεδομένων ενός είδους. Με την ενσωμάτωση διαφορετικών ειδών σε ένα σύνολο δεδομένων, τα μοντέλα εκτίθενται σε μια ευρύτερη ποικιλία μουσικών στυλ και χαρακτηριστικών. Αυτό επιτρέπει στα μοντέλα να μαθαίνουν πιο γενικευμένες αναπαραστάσεις της μουσικής και να αποφεύγουν την υπερπροσαρμογή. Επιπλέον, τα μεγαλύτερα σύνολα δεδομένων πολλαπλών ειδών μπορούν να μειώσουν σε κάποιο βαθμό τον αντίκτυπο προβλημάτων όπως τα ελλιπή και ανοργάνωτα κομμάτια. Αυτό φαίνεται στα μοντέλα adl-piano-midi και Los-Angeles-MIDI-segment, τα οποία ήταν σε θέση να αποδώσουν καλύτερα από τα αρχικά σύνολα εκπαίδευσης.

Τα σύνολα δεδομένων μεγάλης κλίμακας άνω των 100 MB ενισχύουν τη γενίκευση των μοντέλων. Εκθέτοντας τα μοντέλα σε μεγαλύτερο όγκο μουσικών κομματιών, αυτά μαθαίνουν πιο γενικευμένες αναπαραστάσεις των υποκείμενων μοτίβων και δομών της μουσικής Επιπλέον, η μεγαλύτερη κλίμακα μπορεί να μειώσει σε κάποιο βαθμό τον αντίκτυπο προβλημάτων όπως τα ελλιπή και ανοργάνωτα κομμάτια. Από την άλλη μεριά, το μικρότερο μέγεθος περιορίζει την επίδραση άλλων θετικών χαρακτηριστικών των συνόλων δεδομένων. Για παράδειγμα, τα μοντέλα ailabs1k7 δεν πετυχαίνουν το ίδιο επίπεδο επιδόσεων με το σύνολο εκπαίδευσης, παρόλο που το σύνολο εκπαίδευσης είναι υψηλότατης ποιότητας.

Η αντικειμενική μετρική Scale Consistency μπορεί να βοηθήσει στον εντοπισμό συνόλων δεδομένων που προσανατολίζονται σε συναισθηματικά πλούσια μουσική. Η μεγαλύτερη συνέπεια στις κλίμακες είναι χαρακτηριστικό των πιο δημοφιλών ειδών μουσικής, τα οποία έχουν μεγαλύτερο συναισθηματικό αντίκτυπο στους ακροατές. Αυτό φαίνεται στα μοντέλα ailabs1k7, τα οποία έχουν καλύτερες επιδόσεις από άλλα σύνολα δεδομένων στο συναίσθημα βασιζόμενα σε ένα σύνολο δεδομένων ποπ.

Είδη Μοντέλων

Μέσω της διαδικασίας εκπαίδευσης και αξιολόγησης, συγκρίναμε τα αποτελέσματα των δύο ειδών μοντέλων μας, του Music Transformer και του Perceiver-AR, σε διάφορες μετρικές και εντοπίσαμε συγκεκριμένα χαρακτηριστικά και ιδιότητες των μοντέλων που είχαν σημαντικό αντίκτυπο στη διαδικασία εκπαίδευσης και στην ποιότητα των παραγόμενων κομματιών.

Εκπαίδευση Μοντέλων

Τα μοντέλα Perceiver-AR συγκλίνουν σε τουλάχιστον μια τάξη μεγέθους χαμηλότερες τιμές σφαλμάτων επαλήθευσης 0,004-0,03 σε σύγκριση με τα μοντέλα Music Transformer 0,29-1,15. Το γεγονός αυτό μπορεί να εξηγηθεί από τις αυξημένες δυνατότητες και διαστάσεις των μοντέλων Perceiver-AR που έχουν μεγαλύτερη ικανότητα μοντελοποίησης του προβλήματος.

Επιπλέον, ο αριθμός των εποχών που συγκλίνουν τα μοντέλα Perceiver-AR έχει πολύ μικρότερες διακυμάνσεις, 7-14 εποχές, σε σύγκριση με τα μοντέλα Music Transformer, 5-60 εποχές. Το γεγονός αυτό αποτελεί άλλη συνέπεια της διαφορετικής διαστατικότητας των μοντέλων, με το μοντέλο Perceiver-AR να είναι πιο ανθεκτικό, ανεξάρτητα από το σύνολο εκπαίδευσης.

Αντικειμενική Αξιολόγηση

Είδος Μοντέλου	Μέσος Όρος	Τυπική Απόκλιση	Κατανομές	Ακραίες Τιμές	Ομοιότητα με Σύνολα Εκπαίδευσης
Κοινά Χαρακτηριστικά					• Scale Consistency • Tempo Estimation
Music Transformer	Χαμηλότερος	Υψηλή	Ευρείες, ποικίλες	Συχνές Χαμηλές Τιμές	• Polyphony • Polyphony Scale
Perceiver-AR	Ελαφρώς Χαμηλότερος	Χαμηλή-Μεσαία	Συγκεντρωμένες	Σπάνιες	• Pitches Used • Pitch Range • Pitch Classes Used • Pitch Entropy • Pitch Class Entropy • Empty Beat Rate

Table 25: Αντικειμενική Αξιολόγηση Ειδών Μοντέλων

Η σύγκριση μεταξύ των εξόδων του Music Transformer και των εξόδων του Perceiver-AR στις αντικειμενικές μουσικές μετρικές παρέχει διάφορα συμπεράσματα. Πρώτον, οι έξοδοι και των δύο μοντέλων έχουν γενικά χαμηλότερες τιμές για τις περισσότερες μετρικές σε σύγκριση με τα σύνολα εκπαίδευσης. Επιπλέον, οι κατανομές των μετρικών είναι συχνά συμμετρικές, υποδεικνύοντας ότι τα μοντέλα παράγουν κομμάτια με ισορροπημένη κατανομή μουσικών χαρακτηριστικών. Οι έξοδοι του Music Transformer έχουν μικρότερες μέσες τιμές, υψηλότερη τυπική απόκλιση, περισσότερες ακραίες χαμηλές τιμές και πιο ευρείες και ποικίλες κατανομές σε σύγκριση με τα σύνολα εκπαίδευσης και τις εξόδους του Perceiver-AR για τις περισσότερες μετρικές. Τα κομμάτια του Perceiver-AR έχουν πιο συγκεκριμένα μουσικά χαρακτηριστικά, μεγαλύτερη ομοιότητα με τα σύνολα εκπαίδευσης και λιγότερες ακραίες τιμές. Αυτό ενδεχομένως υποδηλώνει ότι το Perceiver-AR παράγει κομμάτια που είναι πιο συνεπή και διαφοροποιούνται λιγότερο από τα δεδομένα εκπαίδευσης.

Υποκειμενική Αξιολόγηση

Υποκειμενική Μετρική	Σύνολο Υποκειμένων	Σύνολα Εκπαίδευσης	Συνθέσεις Music Transformer	Συνθέσεις Perceiver-AR
Οικειότητα	Αρχάριοι	3.12	**3.06**	3.02
Συναίσθημα	Αρχάριοι	2.89	3.01	**3.06**
	Έμπειροι	3.35	3.24	**3.36**
Μελωδικότητα	Έμπειροι	3.07	2.95	**3.07**

Αρμονικότητα	Έμπειροι	3.16	3.08	**3.21**
Ρυθμικότητα	Έμπειροι	3.00	2.95	**3.08**
Είδος	Έμπειροι	2.92	**2.91**	2.80
Φυσικότητα	Αρχάριοι	0.43	**0.43**	0.41
	Έμπειροι	0.37	0.28	**0.37**
Συνολική Αξιολόγηση	Αρχάριοι	2.90	**2.94**	2.90
	Έμπειροι	3.03	2.92	**3.04**
Σταθμισμένος Μέσος Όρος	Αρχάριοι & Έμπειροι Σταθμισμένα	2.98	2.90	**2.99**

Table 26: Υποκειμενική Αξιολόγηση Ειδών Μοντέλων

Τα σύνολα εξόδων έχουν γενικά μέτριες τιμές στις περισσότερες μετρικές, με καλύτερα αποτελέσματα στην αρμονικότητα και το συναίσθημα. Οι έξοδοι του Music Transformer (σταθμισμένος μέσος όρος 2.90) έχουν χαμηλότερες τιμές από τα σύνολα εκπαίδευσης (σταθμισμένος μέσος όρος 2.98). Οι έξοδοι του Perceiver-AR (σταθμισμένος μέσος όρος 2.99) έχουν παρόμοιες τιμές με τα σύνολα εκπαίδευσης, έχοντας καλύτερες επιδόσεις από το Music Transformer στις περισσότερες μετρικές.

Τα περισσότερα σύνολα εξόδων παρουσιάζουν υψηλότερες τιμές συναισθήματος σε σχέση με τα σύνολα εκπαίδευσης, αναδεικνύοντας την ικανότητα των μοντέλων να παράγουν συναισθηματικά πλούσια μουσική. Ωστόσο, οι χαμηλότερες τιμές εξόδων για το είδος υποδηλώνουν ότι τα χαρακτηριστικά του είδους των συνόλων εκπαίδευσης δεν μεταφέρονται σαφώς στις εξόδους, με τα κομμάτια να επηρεάζονται και από το είδος της αρχικής ακολουθίας.

Στη μετρική της φυσικότητας, τα σύνολα εξόδων έλαβαν τις χαμηλότερες τιμές. Ένας από τους καθοριστικούς παράγοντες γι' αυτό είναι η δυσκολία σύνθεσης μουσικής που προσομοιώνει με ακρίβεια κάθε λεπτομέρεια της ανθρώπινης συνθετικής διαδικασίας. Άλλοι καθοριστικοί παράγοντες θα μπορούσαν να είναι μικρές ασυνέχειες ή μπερδέματα στη μελωδία, απότομες αλλαγές στο ρυθμό και παύσεις. Απαιτούνται βελτιώσεις κυρίως στα σύνολα εκπαίδευσης.

Όσον αφορά τη συνολική βαθμολογία, τα σύνολα εξόδων παρουσιάζουν μέτρια αποτελέσματα, έχοντας ως επί το πλείστον υψηλότερες τιμές από τα σύνολα εκπαίδευσης, υποδεικνύοντας τη δυνατότητα των μοντέλων να παράγουν καλύτερη μουσική.

Κύρια Ευρήματα

Music Transformer

Το μοντέλο Music Transformer έχει χαμηλότερη ικανότητα μοντελοποίησης μουσικών στοιχείων, με αποτέλεσμα μεγαλύτερες διακυμάνσεις και μεγαλύτερη ευαισθησία στα χαρακτηριστικά των συνόλων εκπαίδευσης. Οι έξοδοι του μοντέλου είναι πιο ποικίλες με περισσότερες ακραίες τιμές και έχουν χαμηλότερες υποκειμενικές αξιολογήσεις από τα αρχικά δεδομένα εκπαίδευσης.

Το Music Transformer πιθανόν να είναι ικανό να παράγει πιο δημιουργική μουσική που διαφέρει από τα δεδομένα εκπαίδευσης. Όπως αναφέρουν οι (Ji et al., 2020), η μουσική δημιουργικότητα περιλαμβάνει την εξερεύνηση προτύπων πέρα από αυτά που παρατηρούνται στα δεδομένα

εκπαίδευσης. Το Music Transformer έχει κατανομές μουσικών χαρακτηριστικών που διαφέρουν σε κάποιο βαθμό από τα σύνολα εκπαίδευσης με περισσότερες ακραίες τιμές, γεγονός που υποδηλώνει ότι μπορεί να είναι ικανό να παράγει πιο δημιουργική μουσική. Ωστόσο, είναι πιθανό αυτές οι στατιστικές διαφορές μεταξύ των αποτελεσμάτων του Music Transformer και των δεδομένων εκπαίδευσης να οφείλονται σε υψηλότερο επίπεδο τυχαιότητας ή ανακρίβειες του μοντέλου. Απαιτείται πρόσθετη αξιολόγηση για να διαπιστωθεί αν το Music Transformer είναι πραγματικά ικανό να παράγει πιο δημιουργική μουσική.

Perceiver-AR

Το μοντέλο Perceiver-AR συλλαμβάνει καλύτερα περίπλοκα μουσικά μοτίβα στα δεδομένα εκπαίδευσης. Είναι πιο στιβαρό και συνεπές ανεξάρτητα από το σύνολο δεδομένων εκπαίδευσης. Τα κομμάτια του είναι λιγότερο διαφοροποιημένα, πιο κοντά στα αρχικά δεδομένα εκπαίδευσης και αξιολογούνται υποκειμενικά παρόμοια ή καλύτερα από τα σύνολα εκπαίδευσης.

Το Perceiver-AR πιθανόν μπορεί να μετριάσει σε κάποιο βαθμό περιορισμούς των συνόλων εκπαίδευσης. Όπως συζητήθηκε προηγουμένως, πολλά σύνολα εκπαίδευσης δεν έχουν την απαιτούμενη ποιότητα και έχουν αξιολογηθεί χαμηλότερα από το αναμενόμενο σε διάφορες υποκειμενικές μετρήσεις. Παρ' όλα αυτά, το Perceiver-AR ήταν σε θέση να παράγει κομμάτια που ξεπέρασαν τα σύνολα εκπαίδευσης σε ορισμένες από αυτές τις μετρικές.

Κοινά Χαρακτηριστικά

Το μέγεθος της εισόδου (context) που έχει πρόσβαση ένα μοντέλο έχει σημαντικό αντίκτυπο στην κατανόηση της μουσικής δομής. Το Perceiver-AR έχει πρόσβαση σε μεγαλύτερες μουσικές ακολουθίες κατά τη διάρκεια της εκπαίδευσης και της σύνθεσης μουσικής, όντας πιο ικανό να θυμάται τις μεσοπρόθεσμες και μακροπρόθεσμες δομές ενός κομματιού. Αντιθέτως, το Music Transformer τείνει να επαναλαμβάνει μικρότερα μουσικά μοτίβα λόγω του περιορισμένου πλαισίου (context). Συνεπώς, για βέλτιστα αποτελέσματα ως προς αυτό το παράγοντα, τα μοντέλα θα πρέπει να είναι σε θέση να έχουν πρόσβαση στην πλήρη μουσική ακολουθία ενός κομματιού ως ενιαία οντότητα.

Η έλλειψη άμεσης αναπαράστασης μουσικών στοιχείων, όπως η δομή, η μελωδία, η αρμονία και ο ρυθμός, έχει σημαντικό αρνητικό αντίκτυπο στη μοντελοποίηση της μουσικής. Και τα δύο μοντέλα μαθαίνουν αυτές τις πληροφορίες έμμεσα με έναν μπερδεμένο και συγκεχυμένο τρόπο. Αυτή είναι η κύρια αιτία που τα κομμάτια των μοντέλων έχουν σε κάποιο βαθμό ασυνέχειες ή μπερδέματα στη μελωδία, απότομες αλλαγές στο ρυθμό και παύσεις, καθώς τα μοντέλα δεν κατανοούν ότι αυτά τα χαρακτηριστικά είναι συνήθως ανεπιθύμητα. Με την ενσωμάτωση άμεσης αναπαράστασης για αυτά τα στοιχεία, τα μοντέλα θα μπορούν να κατανοήσουν τις σχέσεις μεταξύ των διαφόρων στοιχείων της μουσικής και τον τρόπο με τον οποίο αλληλεπιδρούν για τη δημιουργία πιο ποιοτικής μουσικής.

Εκπαιδευμένα Μοντέλα

Μετά την αξιολόγηση των δύο ειδών μοντέλων, αξιολογήσαμε όλες τις περιπτώσεις εκπαιδευμένων μοντέλων χρησιμοποιώντας υποκειμενικές μετρικές και εντοπίσαμε τα τρία μοντέλα με τις καλύτερες επιδόσεις: Perceiver-AR adl-piano-midi (3.09 σταθμισμένος μέσος όρος),

Perceiver-AR Rock-Piano-MIDI (3.06 σταθμισμένος μέσος όρος) και Perceiver-AR ailabs1k7 (3.02 σταθμισμένος μέσος όρος). Συνολικά, το Perceiver-AR adl-piano-midi είναι το πιο ισχυρό και με τις καλύτερες επιδόσεις μοντέλο, το Perceiver-AR Rock-Piano-MIDI είναι το δεύτερο καλύτερο και το καλύτερο για τους έμπειρους συμμετέχοντες, ενώ το Perceiver-AR ailabs1k7 είναι το καλύτερο μοντέλο για τους αρχάριους συμμετέχοντες.

Συσχετίσεις Μετρικών Αξιολόγησης

Σε αυτό το υποκεφάλαιο ερευνούμε τις συσχετίσεις που προκύπτουν από τις αντικειμενικές και υποκειμενικές μετρικές και εξετάζουμε πιθανές αιτιώδεις σχέσεις που εξηγούν ορισμένες από αυτές. Προσπαθούμε να προσδιορίσουμε τη σημασία κάθε υποκειμενικής μετρικής στον καθορισμό της ποιότητας της μουσικής και προτείνουμε ένα σύνολο πρωτογενών δεικτών ποιότητας για μουσική που παράγεται από μοντέλα τεχνητής νοημοσύνης.

Αντικειμενικές Μετρικές

Οι μετρικές που σχετίζονται με το τονικό ύψος (pitch-related) έχουν μέτριες συσχετίσεις μεταξύ τους στις περισσότερες περιπτώσεις. Αυτό εξηγείται καθώς αυτές οι μετρικές συχνά αλληλεξαρτώνται η μια από την άλλη. Επιπλέον, η μετρική scale consistency έχει αρνητική μέτρια συσχέτιση με τις μετρικές pitch classes και pitch classes entropy. Αυτό συμβαίνει καθώς εάν ένα μουσικό κομμάτι έχει υψηλή συνέπεια σε μια κλίμακα, χρησιμοποιούνται λιγότεροι τόνοι που ανήκουν αρμονικά σε αυτή κλίμακα άρα λιγότερες μοναδικές κλάσεις τόνων στο κομμάτι και χαμηλότερη εντροπία κλάσεων τόνων.

Υποκειμενικές Μετρικές

Οι περισσότερες υποκειμενικές μετρικές συσχετίζονται τουλάχιστον σε μέτριο βαθμό, καθώς όλες επηρεάζουν την ποιότητα της μουσικής και διαμορφώνουν πιθανές αιτιώδεις σχέσεις.

Η Εξοικείωση (Familiarity) συσχετίζεται μέτρια με το συναίσθημα (0.57), καθώς οι άνθρωποι τείνουν να έχουν ισχυρότερη συναισθηματική αντίδραση στη μουσική, με την οποία είναι εξοικειωμένοι. Αυτό οφείλεται στο γεγονός ότι η οικεία μουσική συνδέεται συχνά με προσωπικές αναμνήσεις, εμπειρίες και συναισθήματα, προκαλώντας ισχυρότερες συναισθηματικές αντιδράσεις.

Η Μελωδικότητα (Melodiousness) και η Αρμονικότητα (Harmonicity) επηρεάζουν σε μεγάλο βαθμό το Συναίσθημα (Emotion) στη μουσική, όπως αποδεικνύεται από την ισχυρή συσχέτιση μεταξύ τους (0.71 και 0.70 αντίστοιχα). Ως βασικές μουσικές πτυχές των κομματιών, συμβάλλουν σε μεγάλο βαθμό στον συναισθηματικό αντίκτυπο της μουσικής. Η Ρυθμικότητα (Rhythmicity) συμβάλλει μέτρια στο συναίσθημα με συσχέτιση (0,58). Η μελωδικότητα, η αρμονικότητα και η ρυθμικότητα συσχετίζονται έντονα μεταξύ τους, υποδεικνύοντας ότι αυτές οι μουσικές πτυχές είναι έντονα αλληλένδετες.

Το Είδος (Genre) δεν φαίνεται να αποτελεί καθοριστικό παράγοντα για την ποιότητα της μουσικής, όπως αποδεικνύεται από τις μέτριες συσχετίσεις του με άλλες μετρικές. Ενώ ο προσδιορισμός ενός μουσικού είδους μπορεί να είναι χρήσιμος για την κατηγοριοποίηση της μουσικής και των επιρροών της, φαίνεται ότι δεν επιδρά καθοριστικά στην ποιότητα της μουσικής.

Η Φυσικότητα (Naturalness) σχετίζεται με τη συνολική ποιότητα της μουσικής σε μικρότερο βαθμό από ό,τι άλλες μετρικές, με ασθενείς έως μέτριες συσχετίσεις με όλες τις άλλες υποκειμενικές μετρικές. Η φυσικότητα αντιπροσωπεύει το πόσο πολύ η μουσική μοιάζει με τη φυσική, ανθρώπινη μουσική, η οποία είναι ένας παράγοντας που μπορεί να συμβάλει στη συνολική αντιληπτή ποιότητα της μουσικής. Ωστόσο, ο δυαδικός χαρακτήρας της περιορίζει την ποσότητα της πληροφορίας που μεταφέρει, με αποτέλεσμα τις ασθενέστερες συσχετίσεις με τις άλλες μετρικές.

Η Συνολική Βαθμολογία (Overall Rating) είναι ένα ολοκληρωμένο μέτρο της συνολικής εκλαμβανόμενης ποιότητας της μουσικής, με μέτριες έως υψηλές συσχετίσεις με όλες τις άλλες υποκειμενικές μετρικές. Όπως αναμενόταν, αντανακλά τη συνδυασμένη επίδραση όλων των άλλων υποκειμενικών μετρικών στη συνολική αντίληψη της ποιότητας της μουσικής.

Η Μελωδικότητα (Melodiousness), η Αρμονικότητα (Harmonicity) και το Συναίσθημα (Emotion) είναι οι πιο καθοριστικές μετρικές στην αξιολόγηση της ποιότητας της παραγόμενης μουσικής και μπορούν να θεωρηθούν οι πρωταρχικοί δείκτες ποιότητας για την υποκειμενική αξιολόγηση της παραγόμενης μουσικής από τεχνητή νοημοσύνη. Αυτές οι μετρικές έχουν υψηλή συσχέτιση με την συνολική αξιολόγηση (0.78, 0.75 και 0.71 αντίστοιχα), γεγονός που υποδηλώνει ότι είναι αυτές που επηρεάζουν κυρίως τους ακροατές όσον αφορά την αξιολόγηση της ποιότητας της μουσικής.

Αντικειμενικές-Υποκειμενικές Μετρικές

Στην αντίστοιχη ενότητα παρουσιάζεται η γραφική παράσταση συσχετίσεων όλων των αντικειμενικών και υποκειμενικών μετρικών βάσει των μουσικών κομματιών που αξιολογήθηκαν υποκειμενικά. Όπως αποδεικνύεται, δεν υπάρχει στατιστική συσχέτιση μεταξύ των αντικειμενικών και υποκειμενικών μετρικών για τα μουσικά κομμάτια που αξιολογήθηκαν.

Κεφάλαιο 11: Συζήτηση

Προβλήματα & Βελτιώσεις

Εκπαίδευση Μοντέλων

Η συχνότητα επαλήθευσης δεν είναι συνεπής μεταξύ των μοντέλων Music Transformer και Perceiver-AR. Στο Music Transformer γίνεται επαλήθευση στο τέλος κάθε εποχής, ενώ στο Perceiver-AR κάθε 100 βήματα εκπαίδευσης. Ως αποτέλεσμα, στα σύνολα δεδομένων με περισσότερα training batches έχουμε λιγότερα σημεία επαλήθευσης για το Music Transformer, γεγονός που περιορίζει τις πληροφορίες των καμπυλών μάθησης. Μια πιθανή βελτίωση θα ήταν η επαλήθευση των μοντέλων του Music Transformer σε σταθερό αριθμό βημάτων εκπαίδευσης.

Αντικειμενική Αξιολόγηση

Η διάρκεια των μουσικών κομματιών θα μπορούσε να είναι μια χρήσιμη αντικειμενική μετρική. Αυτή η μετρική συσχετίζεται σε κάποιο βαθμό με τις μετρικές των τονικών υψών και μπορεί να βοηθήσει στο φιλτράρισμα των μουσικών κομματιών για την υποκειμενική αξιολόγηση. Επιπλέον, η ανάλυση της διάρκειας των παραγόμενων κομματιών μπορεί να παράσχει πληροφορίες για τα

σύνολα εξόδων συγκρίνοντας τη διάρκεια των παραγόμενων κομματιών με εκείνες των συνόλων εκπαίδευσης.

Υποκειμενική Αξιολόγηση

Τα αρχικά δείγματα εκπαίδευσης που χρησιμοποιήθηκαν στα συνολικά δείγματα υποκειμενικής αξιολόγησης ήταν αρκετά περιορισμένα, με αποτέλεσμα να υπάρχει μικρός αριθμός αξιολογήσεων για τα σύνολα εκπαίδευσης σε σύγκριση με τα σύνολα εξόδων. Συγκεκριμένα, τα σύνολα εκπαίδευσης αξιολογήθηκαν πολύ λιγότερο από τα σύνολα εξόδων, αποτελώντας μόνο το 14,7% των συνολικών αξιολογήσεων. Θα ήταν προτιμότερο να υπάρχει παρόμοιος αριθμός αξιολογήσεων, καθώς αυτό θα οδηγούσε σε πιο ακριβείς και δίκαιες συγκρίσεις. Μια πιθανή βελτίωση θα ήταν η αύξηση των αρχικών δειγμάτων σε περίπου 33% των συνολικών δειγμάτων, ώστε να επιτευχθεί παρόμοιος αριθμός αξιολογήσεων των συνόλων εκπαίδευσης-εξόδων.

Σε ορισμένες περιπτώσεις, τα αποτελέσματα της υποκειμενικής αξιολόγησης παρουσιάζουν αποκλίσεις μεταξύ των δύο ομάδων συμμετεχόντων. Αυτές οι αποκλίσεις μπορεί να οφείλονται σε διαφορές μουσικών γνώσεων, αλλά ο αριθμός των συμμετεχόντων είναι ένας άλλος κρίσιμος παράγοντας. Αν η αξιολόγηση είχε μεγαλύτερο αριθμό συμμετεχόντων, τα αντικρουόμενα αποτελέσματα θα ελαχιστοποιούνταν και θα μπορούσαμε να διερευνήσουμε πιθανές αιτιώδεις σχέσεις για αυτά.

Η υποκειμενική αξιολόγηση δεν περιελάμβανε αξιολόγηση των μεταβάσεων μεταξύ των αρχικών ακολουθιών και των κομματιών που προκύπτουν. Συνεπώς, η συνολική συνοχή και η δομή της μουσικής δεν αξιολογήθηκαν πλήρως. Δεδομένου ότι η μετρική της φυσικότητας απαιτούσε την εξέταση των μουσικών κομματιών χωρίς την αρχική ακολουθία, η ομαλότητα της μετάβασης και η συνολική δομή του κομματιού δεν αξιολογήθηκαν. Μια πιθανή βελτίωση θα ήταν να αξιολογηθεί η συνοχή του κομματιού με την αρχική ακολουθία, αφού πρώτα ολοκληρωθούν οι αξιολογήσεις των άλλων υποκειμενικών μετρικών χωρίς την αρχική ακολουθία.

Από την υποκειμενική αξιολόγηση απουσίαζε η ποιοτική ανάλυση των μουσικών κομματιών, που θα μπορούσε να δώσει πληροφορίες για τα χαρακτηριστικά των συνόλων εκπαίδευσης και των μοντέλων. Μια πιθανή λύση θα μπορούσε να είναι η προσθήκη μιας ερώτησης με προκαθορισμένες επιλογές για να επιλέξουν οι ακροατές ορισμένα τυπικά χαρακτηριστικά των κομματιών, όπως ασυνέχειες, μπερδέματα στη μελωδία, ανακρίβειες στην αρμονία, απότομες αλλαγές ρυθμού, επανάληψεις θέματος, παύσεις, κολλήματα, ικανοποιητικές μεταβάσεις, γνωστά μοτίβα, ξαφνικά κλεισίματα, ή να τους επιτραπεί να δώσουν περιγραφές άλλων χαρακτηριστικών.

Περιορισμοί & Μελλοντικές Επεκτάσεις

Σύνολα Εκπαίδευσης

Η περιορισμένη διαθεσιμότητα των συνόλων δεδομένων αποτελεί πρόκληση για την εκπαίδευση μοντέλων σύνθεσης μουσικής. Η έλλειψη μιας ποικιλίας ανοικτά διαθέσιμων συνόλων δεδομένων αποτελεί ένα σημαντικό περιορισμό καθώς καθιστά δύσκολη την εκπαίδευση μοντέλων που μπορούν να αποτυπώσουν την πολυπλοκότητα και τις αποχρώσεις των μουσικών εκτελέσεων,

οδηγώντας σε περιορισμούς στην ικανότητα των μοντέλων να παράγουν καλύτερης ποιότητας μουσική. Ως εκ τούτου, υπάρχει ανάγκη δημιουργίας περισσότερων συνόλων δεδομένων.

Ένας άλλος πρωταρχικός περιορισμός είναι η ποιότητα των συνόλων εκπαίδευσης. Όπως τονίζεται στα συμπεράσματά μας, τα σύνολα εκπαίδευσης που χρησιμοποιήθηκαν ήταν γενικά κάτω από τα αναμενόμενα πρότυπα. Αυτό έχει ως αποτέλεσμα χαμηλότερη ποιότητα συνθέσεων, η οποία φαίνεται σε πολλές μετρικές με πιο χαρακτηριστική τη φυσικότητα. Ένας παράγοντας που μπορεί να επηρεάσει σημαντικά την ποιότητα αυτών των συνόλων δεδομένων είναι η παρουσία ελλιπών και ανοργάνωτων κομματιών με ασυνέχειες, μεγάλες παύσεις και θόρυβο. Ως εκ τούτου, είναι απαραίτητη η προσεκτική επιμέλεια και προεπεξεργασία των συνόλων εκπαίδευσης.

Το περιορισμένο μέγεθος και η έλλειψη ποικιλίας μουσικών ειδών στα σύνολα δεδομένων εκπαίδευσης αποτελούν επίσης έναν από τους κύριους περιορισμούς. Τα περισσότερα διαθέσιμα σύνολα δεδομένων περιορίζονται κυρίως στο κλασικό είδος, γεγονός που οδηγεί τα μοντέλα να επικεντρώνονται στα ίδια μουσικά στυλ. Τα σύνολα δεδομένων εκπαίδευσης θα πρέπει ιδανικά να αντιπροσωπεύουν ολόκληρο το φάσμα των μουσικών ειδών. Επίσης, όπως κατέληξε η έρευνά μας, τα μεγάλα σύνολα πολλαπλών ειδών ενισχύουν τη γενίκευση των μοντέλων και μπορούν να μειώσουν την επίδραση θεμάτων όπως τα ελλιπή και ανοργάνωτα κομμάτια. Αυτό υπογραμμίζει τη σημασία της συνέχισης της ανάπτυξης και της επέκτασης των υφιστάμενων συνόλων δεδομένων, καθώς και της δημιουργίας νέων που πληρούν τα επιθυμητά πρότυπα για το μέγεθος και την ποικιλομορφία.

Αναπαράσταση Μουσικής

Η αναπαράσταση γεγονότων τύπου MIDI είναι πολύ περιορισμένη για να αποτυπώσει την πολυπλοκότητα της μουσικής γλώσσας. Η χρήση γεγονότων τόνου, χρονικής μετατόπισης και έντασης δεν επαρκεί για την αναπαράσταση της ποικιλομορφίας των μουσικών στοιχείων, όπως επισημαίνουν και οι (Ji et al., 2020). Εκτός από το τονικό ύψος και τις χρονικότητες, στοιχεία όπως η δομή, ο ρυθμός και οι συγχορδίες, καθώς και το grooving ή το συναίσθημα, θα μπορούσαν να αναπαρασταθούν από μουσικά γεγονότα. Με αυτό το τρόπο, τα μοντέλα θα έχουν καλύτερη κατανόηση ολόκληρης της δομής των κομματιών, των εξελίξεων των συγχορδιών και των ρυθμικών μοτίβων, ενισχύοντας έτσι την ικανότητά τους να παράγουν πιο ποιοτικά κομμάτια. Μια πολλά υποσχόμενη προσπάθεια προς αυτή την κατεύθυνση είναι η αναπαράσταση REMI που εισήγαγαν οι (Y.-S. Huang & Yang, 2020).

Μοντέλα Σύνθεσης Μουσικής

Οι υπολογιστικοί πόροι που απαιτούνται για την εκπαίδευση των μοντέλων Transformer είναι πολύ υψηλοί. Αυτό αποτελεί σημαντικό εμπόδιο για την ανάπτυξη τους και τον πειραματισμό με αυτά τα μοντέλα. Για την αντιμετώπιση αυτής της πρόκλησης, οι ερευνητές πρέπει να επικεντρωθούν στη μείωση των υπολογιστικών τους απαιτήσεων. Μια πολλά υποσχόμενη λύση σε αυτό το ζήτημα είναι το transfer learning, με τους (Donahue et al., 2019) έχουν παρουσιάσει μια προσέγγιση προς αυτή την κατεύθυνση που δείχνει πολλά υποσχόμενη.

Τα μοντέλα θα πρέπει να παράγουν μουσική με μακροπρόθεσμη δομή και κατάλληλο μουσικό κλείσιμο. Επί του παρόντος, τα μοντέλα μας παράγουν κομμάτια με σταθερό αριθμό μουσικών

γεγονότων χρησιμοποιώντας μια αναδρομική διαδικασία σύνθεσης. Παρ' όλα αυτά, η προσέγγιση αυτή δεν επιτρέπει τον έλεγχο του τέλους του κομματιού, οδηγώντας σε ξαφνικά και μη ικανοποιητικά κλεισίματα που δεν αντιπροσωπεύουν ανθρώπινες συνθέσεις.

Οι χρήστες μπορούν να αλληλεπιδράσουν με τα μοντέλα παραγωγής μουσικής μας μόνο παρέχοντας την αρχική ακολουθία και το μήκος του κομματιού που θα συντεθεί. Οι χρήστες δεν είναι σε θέση να αλληλεπιδρούν με τα μοντέλα ευρύτερα, ελέγχοντας μουσικά στοιχεία όπως η κλίμακα, ο ρυθμός, η δομή του κομματιού και το θέμα. Για να επιτευχθεί μια πιο ολοκληρωμένη αλληλεπίδραση, πρέπει να αναπαρασταθούν και να προσαρμόζονται διάφορα μουσικά χαρακτηριστικά μέσα στα μοντέλα σύνθεσης μουσικής. Αυτό θα τους επιτρέψει να λειτουργούν πλήρως ως συστήματα υποβοήθησης σύνθεσης.

Τα μοντέλα σύνθεσης μουσικής θα πρέπει να παράγουν πιο δημιουργική μουσική που αποκλίνει από τα δεδομένα εκπαίδευσης. Όπως τονίζουν και οι (Ji et al., 2020), οι συνθέσεις δεν θα πρέπει μόνο να μοιάζουν στατιστικά με τα μοτίβα που μαθαίνονται από το σύνολο εκπαίδευσης, αλλά να παρουσιάζουν και διαφορετικά μουσικά χαρακτηριστικά.

Τα μοντέλα σύνθεσης μουσικής μπορούν να σχεδιαστούν για να παράγουν συναισθηματικά πλούσια μουσική που αντανακλά συγκεκριμένες διαθέσεις και συναισθήματα. Αυτό απαιτεί βαθιά κατανόηση της σχέσης μουσικής-συναισθήματος και ενσωμάτωση της στη μουσική αναπαράσταση. Αυτή η προσέγγιση θα μπορούσε να οδηγήσει στην σύνθεση μουσικής για ένα εύρος συναισθηματικών καταστάσεων που μπορεί να χρησιμοποιηθεί σε ποικίλες εφαρμογές, όπως η μουσική επένδυση ταινιών, η θεραπεία και η ψυχαγωγία.

Αντικειμενική Αξιολόγηση

Θα πρέπει να αναπτυχθούν νέες αντικειμενικές μετρικές βασιζόμενες σε μουσικά χαρακτηριστικά, οι οποίες θα συσχετίζονται με τις υποκειμενικές μετρικές αξιολόγησης, σχηματίζοντας ένα πρωταρχικό σύνολο αντικειμενικών δεικτών ποιότητας αξιολόγησης. Η μελλοντική έρευνα θα πρέπει να αναζητήσει συσχετίσεις μεταξύ των ποσοτικών μετρικών και της υποκειμενικής αξιολόγησης και να εργαστεί προς την κατεύθυνση της αυτοματοποίησης της διαδικασίας αξιολόγησης της μουσικής με την ενσωμάτωση σύνθετης μουσικής γνώσης στα μοντέλα παραγωγής μέσω αυτόματων μεθόδων εξαγωγής χαρακτηριστικών.

Υποκειμενική Αξιολόγηση

Ένας μεγαλύτερος αριθμός αξιολογήσεων θα παρείχε πιο αξιόπιστα, ισχυρά και λεπτομερή αποτελέσματα σχετικά με τα σύνολα εκπαίδευσης και τα μοντέλα μας. Πρώτον, θα αύξανε την αξιοπιστία των αποτελεσμάτων μας μειώνοντας τον αντίκτυπο της τυχαιότητας στις υποκειμενικές κρίσεις. Με κλιμακωτά περισσότερες αξιολογήσεις, θα είχαμε μια πιο ακριβή εικόνα της συνολικής αξιολόγησης των μουσικών κομματιών από ένα μεγαλύτερο δείγμα ακροατών. Επιπλέον, θα γινόταν μια πιο εμπεριστατωμένη ανάλυση των συνόλων εκπαίδευσης και των μοντέλων, εντοπίζοντας πιθανές περιοχές βελτίωσης. Τέλος, θα ελαχιστοποιούνταν ορισμένα αντικρουόμενα αποτελέσματα και θα μπορούσαμε να διερευνήσουμε πιθανές αιτιώδεις σχέσεις που θα εξηγούσαν αυτές τις ασυνέπειες. Ωστόσο, δεν ήταν εφικτό να συγκεντρώσουμε μεγαλύτερο αριθμό αξιολογήσεων λόγω περιορισμών πόρων και χρόνου.

References

Bjørndalen, O. M. (2023). *Mido: MIDI Objects for Python* (1.2.10). GitHub. https://github.com/mido/mido

Briot, J. P. (2021). From artificial neural networks to deep learning for music generation: history, concepts and trends. *Neural Computing and Applications*, *33*(1), 39–65. https://doi.org/10.1007/S00521-020-05399-0/METRICS

Briot, J.-P., Hadjeres, G., & Pachet, F.-D. (2020). *Deep Learning Techniques for Music Generation*. Springer International Publishing. https://doi.org/10.1007/978-3-319-70163-9

Cancino-Chacón, C. E., Grachten, M., Goebl, W., & Widmer, G. (2018). Computational Models of Expressive Music Performance: A Comprehensive and Critical Review. *Frontiers in Digital Humanities*, *5*. https://doi.org/10.3389/fdigh.2018.00025

Chu, H., Kim, J., Kim, S., Lim, H., Lee, H., Jin, S., Lee, J., Kim, T., & Ko, S. (2022). An Empirical Study on How People Perceive AI-generated Music. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 304–314. https://doi.org/10.1145/3511808.3557235

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. v., & Salakhutdinov, R. (2019). Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 2978–2988. https://doi.org/10.48550/arxiv.1901.02860

Donahue, C., Mao, H. H., Li, Y., Cottrell, G., & McAuley, J. (2019). LakhNES: Improving Multi-instrumental Music Generation with Cross-domain Pre-training. *International Society for Music Information Retrieval Conference*.

Dong, H. W., Hsiao, W. Y., Yang, L. C., & Yang, Y. H. (2017). MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 34–41. https://doi.org/10.48550/arxiv.1709.06298

Dong, H. W., & Yang, Y. H. (2018). Convolutional Generative Adversarial Networks with Binary Neurons for Polyphonic Music Generation. *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018*, 190–196. https://doi.org/10.48550/arxiv.1804.09399

Dong, H.-W., Chen, K., McAuley, J., & Berg-Kirkpatrick, T. (2020). *MusPy: A Toolkit for Symbolic Music Generation*. https://doi.org/10.48550/arxiv.2008.01951

Feezell Mark. (2013). *LearnMusicTheory.net Music Theory Fundamentals High-Yield Music Theory, vol. 1*.

Ferreira, L. N., Lelis, L. H. S., & Whitehead, J. (2020). Computer-Generated Music for Tabletop Role-Playing Games. *Proceedings of the 16th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE 2020*, 59–65. https://doi.org/10.48550/arxiv.2008.07009

Filandrianos, G. (2019). geofila/site: Presentantion Site. In *GitHub*. GitHub. https://github.com/geofila/site

Flask Hello World – Vercel. (2023). https://vercel.com/templates/python/flask-hello-world

Gotham, M., Gullings, K., Hamm, C., Hughes, B., Jarvis, B., Lavengood, M., & Peterson, J. (2021). *Open Music Theory*. https://viva.pressbooks.pub/openmusictheory/

Hadjeres, G. (2018). *Interactive Deep Generative Models for Symbolic Music* [PhD]. Sorbonne Universite.

Hadjeres, G., & Nielsen, F. (2017). *Interactive Music Generation with Positional Constraints using Anticipation-RNNs*. https://doi.org/10.48550/arxiv.1709.06404

Hadjeres, G., Pachet, F., & Nielsen, F. (2017). DeepBach: A Steerable Model for Bach Chorales Generation. *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, 1362–1371.

Harris, C. R., Millman, K. J., der Walt, S. J. van, Gommers, R., Virtanen, P., David Cournapeau, Wieser, E., Taylor, J., Sebastian Berg, Smith, N. J., Kern, R., Hoyer, M. P. and S., van Kerkwijk, M. H., Matthew Brett, Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Pierre Gérard-Marchant, … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hawthorne, C., Jaegle, A., Cangea, C., Borgeaud, S., Nash, C., Malinowski, M., Dieleman, S., Vinyals, O., Botvinick, M., Simon, I., Sheahan, H., Zeghidour, N., Alayrac, J.-B., Carreira, J., & Engel, J. (2022a). *General-purpose, long-context autoregressive modeling with Perceiver AR*. https://doi.org/10.48550/arxiv.2202.07765

Hawthorne, C., Jaegle, A., Cangea, C., Borgeaud, S., Nash, C., Malinowski, M., Dieleman, S., Vinyals, O., Botvinick, M., Simon, I., Sheahan, H., Zeghidour, N., Alayrac, J.-B., Carreira, J., & Engel, J. (2022b, July 16). *Perceiver AR: general-purpose, long-context autoregressive generation*. https://www.deepmind.com/blog/perceiver-ar-general-purpose-long-context-autoregressive-generation

Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, C. Z. A., Dieleman, S., Elsen, E., Engel, J., & Eck, D. (2018). Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset. *7th International Conference on Learning Representations, ICLR 2019*. https://doi.org/10.48550/arxiv.1810.12247

Hernandez-Olivan, C., & Beltrán, J. R. (2023). *Music Composition with Deep Learning: A Review* (pp. 25–50). https://doi.org/10.1007/978-3-031-18444-4_2

Hernandez-Olivan, C., Hernandez-Olivan, J., & Beltran, J. R. (2022). *A Survey on Artificial Intelligence for Music Generation: Agents, Domains and Perspectives*. arXiv. https://doi.org/10.48550/ARXIV.2210.13944

Hernandez-Olivan, C., Puyuelo, J. A., & Beltran, J. R. (2022). *Subjective Evaluation of Deep Learning Models for Symbolic Music Composition*.

Hsiao, W. Y., Liu, J. Y., Yeh, Y. C., & Yang, Y. H. (2021). Compound Word Transformer: Learning to Compose Full-Song Music over Dynamic Directed Hypergraphs. *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, *1*, 178–186. https://doi.org/10.48550/arxiv.2101.02402

Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., Dai, A. M., Hoffman, M. D., Dinculescu, M., & Eck, D. (2018). *Music Transformer*. https://doi.org/10.48550/arxiv.1809.04281

Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Simon, I., Hawthorne, C., Shazeer, N., Dai, A. M., Hoffman, M. D., Dinculescu, M., & Eck, D. (2019). Music Transformer. *International Conference on Learning Representations*. https://openreview.net/forum?id=rJe4ShAcF7

Huang, Y.-S., & Yang, Y.-H. (2020). Pop Music Transformer: Beat-Based Modeling and Generation of Expressive Pop Piano Compositions. *Proceedings of the 28th ACM International Conference on Multimedia*, 1180–1188. https://doi.org/10.1145/3394171.3413671

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science and Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., & Carreira, J. (2021). *Perceiver: General Perception with Iterative Attention*. https://doi.org/10.48550/arxiv.2103.03206

Ji, S., Luo, J., & Yang, X. (2020). *A Comprehensive Survey on Deep Music Generation: Multi-level Representations, Algorithms, Evaluations, and Future Directions*.

Jiang, J., Xia, G. G., Carlton, D. B., Anderson, C. N., & Miyakawa, R. H. (2020). Transformer VAE: A Hierarchical Model for Structure-Aware and Interpretable Music Representation Learning. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 516–520. https://doi.org/10.1109/ICASSP40776.2020.9054554

Juslin, P. N. (1993). *Handbook of Music and Emotion: Theory, Research, Applications*. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199230143.001.0001

Juslin, P. N., & Västfjäll, D. (2008). Emotional responses to music: The need to consider underlying mechanisms. *Behavioral and Brain Sciences*, *31*(5), 559–575. https://doi.org/10.1017/S0140525X08005293

Kingma, D. P., & Ba, J. L. (2014). Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. https://doi.org/10.48550/arxiv.1412.6980

Koelsch, S. (2010). Towards a neural basis of music-evoked emotions. *Trends in Cognitive Sciences*, *14*(3), 131–137. https://doi.org/10.1016/j.tics.2010.01.002

Koelsch, S. (2014). Brain correlates of music-evoked emotions. *Nature Reviews Neuroscience*, *15*(3), 170–180. https://doi.org/10.1038/nrn3666

Kong, Q., Li, B., Chen, J., & Wang, Y. (2020). GiantMIDI-Piano: A large-scale MIDI dataset for classical piano music. *Transactions of the International Society for Music Information Retrieval*, *5*(1), 87–98. https://doi.org/10.48550/arxiv.2010.07061

Lev, A. (2021). *Tegridy-Tools: Symbolic Music NLP Artificial Intelligence Toolkit*. GitHub. https://github.com/asigalov61/tegridy-tools

Lev, A. (2022a). *GIGA-Piano-XL: SOTA Piano Transformer model trained on 4.2GB of Solo Piano MIDI music*. GitHub. https://github.com/asigalov61/GIGA-Piano-XL

Lev, A. (2022b). *Perceiver Music Transformer*. GitHub. https://github.com/asigalov61/Perceiver-Music-Transformer

Lev, A. (2022c). *Rock Piano MIDI Dataset*. GitHub. https://github.com/asigalov61/Rock-Piano-MIDI-Dataset

Lev, A. (2023). *Los Angeles MIDI Dataset: SOTA kilo-scale MIDI dataset for MIR and Music AI purposes*. GitHub. https://github.com/asigalov61/Los-Angeles-MIDI-Dataset

Levitin, D. J. (2006). *This is your brain on music: The science of a human obsession*. Penguin.

Lousseief, E., & Sturm, B. (2019). *MahlerNet: Unbounded Orchestral Music with Neural Networks*. http://www.mahlernet.se/files/SMC2019.pdf

McKinney, W. (2010). Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, 56–61. https://doi.org/10.25080/MAJORA-92BF1922-00A

Oore, S., Simon, I., Dieleman, S., Eck, D., & Simonyan, K. (2020). This time with feeling: learning expressive musical performance. *Neural Computing and Applications*, *32*(4), 955–967. https://doi.org/10.1007/s00521-018-3758-9

Pandas Development Team. (2020). *pandas-dev/pandas: Pandas*. Zenodo. https://doi.org/10.5281/zenodo.3509134

Payne Christine. (2019, April 25). *MuseNet*. OpenAI. https://openai.com/blog/musenet/

Peracha, O. (2019). *Improving Polyphonic Music Models with Feature-Rich Encoding*. https://doi.org/10.5281/zenodo.4245396

Raffel, C., & Ellis, D. (2014). Intuitive Analysis, Creation and Manipulation of MIDI Data with pretty_midi. *Proceedings of the 15th International Conference on Music Information Retrieval Late Breaking and Demo Papers*. https://colinraffel.com/publications/ismir2014intuitive.pdf

Roberts, A., Engel, J., Raffel, C., Hawthorne, C., & Eck, D. (2018). A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music. *35th International Conference on Machine Learning, ICML 2018*, *10*, 6939–6954. https://doi.org/10.48550/arxiv.1803.05428

Shaw, P., Uszkoreit, J., & Vaswani, A. (2018). Self-Attention with Relative Position Representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 464–468. https://doi.org/10.18653/v1/N18-2074

Sloboda, J. A. (1985). *The musical mind: The cognitive psychology of music*. Oxford University Press.

Tan, H. H. (2020). *Understanding Music Transformer - gudgud96's Blog*. https://gudgud96.github.io/2020/04/01/annotated-music-transformer/

Tan, H. H., & Herremans, D. (2020). *Music FaderNets: Controllable Music Generation Based On High-Level Features via Low-Level Feature Modelling*. https://doi.org/10.48550/arxiv.2007.15474

Tiernan, R. (2022, July 31). *DeepMind's Perceiver AR: a step toward more AI efficiency*. ZDNET. https://www.zdnet.com/article/deepminds-perceiver-ar-a-step-toward-more-ai-efficiency/

Wang, Z., Zhang, Y., Zhang, Y., Jiang, J., Yang, R., Zhao, J., & Xia, G. (2020). *PIANOTREE VAE: Structured Representation Learning for Polyphonic Music*. https://doi.org/10.48550/arxiv.2008.07118

Waskom, M. (2021). Seaborn: Statistical Data Visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/JOSS.03021

Wu, J., Hu, C., Wang, Y., Hu, X., & Zhu, J. (2017). A Hierarchical Recurrent Neural Network for Symbolic Melody Generation. *IEEE Transactions on Cybernetics*, *50*(6), 2749–2757. https://doi.org/10.48550/arxiv.1712.05274

Wu, S.-L., & Yang, Y.-H. (2020). *The Jazz Transformer on the Front Line: Exploring the Shortcomings of AI-composed Music through Quantitative Measures*. https://doi.org/10.48550/arxiv.2008.01307

Yang, K. (2021a). *Implementation of music transformer with tensorflow-2.0 (ICLR2019)*. GitHub. https://github.com/jason9693/MusicTransformer-tensorflow2.0

Yang, K. (2021b). *Midi Neural Processor*. GitHub. https://github.com/jason9693/midi-neural-processor

Yang, L. C., Chou, S. Y., & Yang, Y. H. (2017). MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation. *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017*, 324–331. https://doi.org/10.48550/arxiv.1703.10847

Yang, L.-C., & Lerch, A. (2020). On the evaluation of generative models in music. *Neural Computing and Applications*, *32*(9), 4773–4784. https://doi.org/10.1007/s00521-018-3849-7

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2021). Dive into Deep Learning. *ArXiv Preprint ArXiv:2106.11342*.

Zhang, N. (2020). Learning Adversarial Transformer for Symbolic Music Generation. *IEEE Transactions on Neural Networks and Learning Systems*, 1–10. https://doi.org/10.1109/TNNLS.2020.2990746

Zinda, Z. (2021). *Data Science Stats Review: Pearson's, Kendall's, and Spearman's Correlation for Feature Selection*. PhData. https://www.phdata.io/blog/data-science-stats-review/