POLITECNICO
MILANO 1863

Dipartimento di Elettronica, Informazione e Bioingegneria

Master Degree in Music and Acoustic Engineering

Modeling Harmonic Complexity in Automatic Music Generation using Conditional Variational Autoencoders

by:
Davide Gioiosa

matr.:
921336

Supervisor: Prof. Massimiliano Zanoni

Co-supervisor: Dr. Luca Comanducci

Academic Year
2019-2020

Abstract

Is it possible to use complexity as a parameter to automatically generate music? This is the question that motivates our research. In the area of automatic music composition, several neural network models have been implemented to generate music of a certain musical genre, e.g. rock, pop, jazz, or to capture and imitate the style of a composer. Recent studies in this area of research, focus on providing the ability not only to generate music, but also to be able to condition the creative process.

From previous researches we know that complexity is a parameter closely related to the amount of brain activity of the listener (the so-called "arousal potential"). It also affects a person's musical preferences. Given this close correlation with a listener's perceptions, we decide to explore the use of this parameter in music.

Complexity is present in each of the aspects in which the music can be divided, e.g. chords, rhythm, melody, etc. Among these we choose to focus on the harmony. In particular, in this work we explore harmonic complexity and its use as a parameter to condition the generation of chord sequences. For the automatic generation process we exploit two conditional neural network models both based on the Variational Autoencoder. We evaluated, through a perceptual test, the ability to generate chord sequences give a desired complexity values.

Sommario

È possibile utilizzare la complessità come parametro per generare musica automaticamente? Questa è la domanda che motiva la nostra ricerca. Nell'ambito della composizione musicale automatica, sono stati implementati diversi modelli di reti neurali in grado di generare musica di un certo genere musicale, e.g. rock, pop, jazz, o in grado di catturare ed imitare lo stile di un compositore. I recenti studi in quest'ambito di ricerca, si concentrano sul fornire la possibilità non solo di comporre musica, ma di poter condizionare il processo creativo.

Dalle precedenti ricerche sappiamo che la complessità è un parametro strettamente correlato alla quantità di attività cerebrale dell'ascoltatore (il cosidetto "arousal potential"). Inoltre condiziona le preferenze musicali di una persona. Data questa stretta correlazione con le percezioni di un ascoltatore, abbiamo deciso di esplorare l'uso di questo parametro in musica.

La complessità è presente in ognuno degli aspetti in cui la musica può essere divisa, ad esempio accordi, ritmo, melodia, ecc... Tra questi scegliamo di focalizzarci sull'armonia. In particolare, in questo lavoro esploriamo la complessità armonica ed il suo utilizzo come parametro per condizionare la generazione di sequenze di accordi. Per il processo di generazione automatica utilizziamo due modelli di rete neurale condizionati, entrambi basati sul Variational Autoencoder. Valutiamo, tramite un test percettivo, la capacità di generare sequenze di accordi scegliendo il valore di complessità desiderato.

Ringraziamenti

This thesis is the result of almost a year of work at the Image and Sound Processing Lab. First I would like to thank my supervisor, prof. Massimiliano Zanoni, for having been a guidance during this path. He allowed me to work on topics in which I am passionate about and helped me learning a lot. I would like also to thank Dr. Luca Comanducci for his constant support, in particular in the most difficult moment. It was really a pleasure working with both of you.

A big thank goes to my family and friends, for their constant support in these five years, it has been a long journey. I am lucky to have you by my side.

Davide

Contents

List of Figures

List of Tables

Introduction

Music is part of every person's life. Art, in all its forms, has always been an expression of creativity and aesthetics for humans, impossible to represent simply by using mathematical formulas or algorithms. In the last decade, however, this concept has been revolutionized. In the music field, the development of new techniques for encoding audio content has marked a clear passage of music from the physical (CD, vinyls) to the digital format. As a consequence, vast audio catalogs that can be consulted easily and is exponentially increasing the amount of data available online. Streaming services such as Spotify or Apple Music provide now to the users the possibility of searching, explore and create music collection, giving the possibility to listen to music in any place and time. Consequently, there has been a growing interest in the research field of Music Informational Retrieval (MIR), which deals with the extraction and the inference of meaningful features from music that describe its content using techniques of Digital Signal Processing, Machine Learning and Deep Learning. The tasks addressed by MIR are multiple, such as classification (e.g. genre-tag), Recommender Systems (web-based systems such as Spotify) and Automatic Music Generation.

As far as the level of abstraction is concerned, features can be distinguished according to the media format analyzed (e.g. audio or symbolic domain). In general we identify as low-level descriptors those that represent audio or musical content close-to-signal (e.g. pitch, frequency) while as high-level descriptors those that are close to human perception (e.g. emotions, expectations).

Several studies have been conducted to link features belonging to different levels of abstraction based on research in musical cognition and psychology fields, through the analysis and the use of mid-level features such as rhythm, harmony, melody, computed by algorithms or obtained by human meta-tags [6, 7]. High-level descriptors allow to define music as it is perceived by most listeners in a meaningful and intuitively recognizable way. Their use is of great interest because with this feature it is possible to model MIR and generative system's in relation to the musical experience of most users e.g. create playlists based on mood, classify music based on the perceived emotion or recommend new songs according to user's taste.

In our study we focus on **music complexity**, investigating the use of

this feature as a parameter in the generation process of automatic music composition systems. Complexity characterizes something we consider to be counterintuitive or unpredictable. Several studies have shown evidence of the 'inverted U-shaped' relationship between complexity perceived and arousal potential in different artistic fields (e.g. music, paintings). Furthermore, this parameter can influence the level of preference and interest, making it an important feature in the study of aesthetic and emotional human perception [8, 9]. In particular, we will analyze the musical harmonic complexity, focusing on the perception of the notes that compose chords and how they relate to each other in a progression over time, and we will use it to model chord sequences in the Automatic Music Generation domain. Manipulation of this parameter can be done in music by modeling the expectation (e.g sensory, cognitive) of a person while listening to a music piece. Many experienced composers often use techniques to confirm or alter the predictions made by a listener to provoke the desired emotions. Although the evaluation of musical complexity perceived may depend on several factors such as musical knowledge or familiarity, we assume that it is possible define a general perception of this concept, considering the possibility in distinguish complexity values in music by different users with western cultural background. For example in the context of harmony, the choice of perfect authentic cadence (V-I) is usually recognized in the western listener as a conclusive and stable chord transition [10].

Automatic Music Generation is an area of growth in recent years due to improved Deep Learning techniques, the availability of computational power and the availability of large number of data. These technological developments have made possible the creation of models capable of generating musical composition learning rules and structures directly from data. These models resulted to be more interesting and solid than the previous ones generated by Markov Chains or rule-based design. Several researches have been conducted regarding neural network models for the generation of new music compositions by learning long-term structures or specific musical styles (i.e. classic, jazz). In our study we mainly focus on the Conditional Architectures, which are models parameterized with some extra information (condition) of the data, such as class labels or outputs of other architectures. This type of neural networks allow the possibility to have a control over the data generation process [11, 12]. The starting point of our study is the annotated dataset obtained from the research of Di Giorgi, containing chord sequences associated with a perceptual complexity value [13]. We discuss two different conditional architectures based on Variational Autoencoder and using harmonic complexity as the conditioning class label. Then we explore the conditioned generation of new chord sequences given values of harmonic complexity. Finally, we evaluate the models by conducting a listening test collecting ratings on the outputs generated.

The use of perceptual features in automatic music composition is

innovative, especially in the use of complexity. We believe that the possibility of using this kind of conditioning in the generative process may open up new and interesting scenarios on music modeling.

Following we describe the structure of the thesis: in chapter 1 we present the state of the art on complexity in music, high-level features representations and automatic music generation. The required background on music theory, neural networks and Variational Autoencoder model is provided in chapter 2. Then, in chapter 3 we will present our two conditional neural networks models used in the study. In chapter 4 we will analyze the structure and the results obtained from a perceptual test on the chord sequences generated by our models. Finally we will discuss the conclusions on the work done and possible future developments in the chapter 5.

1 State of the Art

In this chapter we discuss the perceptual features in Music Information Retrieval (MIR) and then focus on the complexity, how it can be defined according to different points of view and how it relates to aesthetics and arousal perceptions. We present studies that analyze this concept in music focusing on harmony, showing the potential that this feature could have in the generative domain. Finally, we cover the topic of automatic music generation using Deep Learning techniques, presenting the state of the art and the strategies to condition the generative process.

1.1 Perceptual musical features

Following, we discuss the different types of features retrieved from audio or musical content analyzed in the Music Information Retrieval research area, focusing on the importance of high-level features. Then we describe complexity, first giving a general definition and then describing the research done on this feature in the music field. Finally we focus on harmonic complexity, the central theme of our research.

1.1.1 Features in MIR research

"Music Information Retrieval is a multidisciplinary research endeavor that strives to develop innovative content-based searching themes, novel interfaces, and evolving networked delivery mechanisms in an effort to make worlds vast store of music accessible to all" [14]. It involves several domains such as musicology, acoustics, psychoacoustics, signal processing and computer science. MIR deals with the extraction and the inference of meaningful features from musical and audio data in different formats

(e.g. wav in audio domain, MIDI in symbolic domain) that describe its content. This process is achieved using techniques of Digital Signal Processing (DSP), Machine Learning (ML) and Deep Learning (DL), combining algorithms of signal analysis with musicology knowledge.

According to the level of abstraction of the information, the extracted features can be divided mainly into three categories. The first are the low-level features, often based on short-time, frame-based measures, and include audio signal information such as spectral centroid, MFCC (Mel-Frequency Cepstrum Coefficients) or psychoacoustic measures such as loudness and sharpness. These informations are usually not considered meaningful for end-user. The seconds are mid-level features, based on a longer window of analysis. They describe musical concepts such as rhythms, harmony or melody. The latter are the high-level features, which are semantic descriptors closely related to human knowledge, e.g. emotions, expectations and complexity. The graphic representation of the different categories is shown in Figure 1.1.

The extracted features are used in the many applications of MIR research field. The classification task consists in assigning one or more labels to music content. Kim et al. [15] discuss methods for recognizing moods or emotions, using content-based approaches (e.g. feature extraction) and contextual text information (e.g. tags, lyrics or human annotations). Oramas [16] proposes a representation learning approach using Deep Learning techniques for music genre classification, combining multimodal data representations (e.g. audio, text, images). With the increase of digital content and the use of audio streaming platforms (e.g. Spotify, Apple Music) the field of Music Recommendation Systems has become more relevant. There are several applications in this domain, such as automatic search systems in large music libraries, systems capable of suggesting suitable songs to users [17]. Music recommendation systems are usually developed through content-based filtering (learning from the audio features) and collaborative filtering, collecting information about user-music interactions (e.g., number of listens, ratings), to identify similar users and tracks. Automatic Music Generation field implement systems capable of abstracting information about musical concepts and composition rules to create musical pieces [3].

In recent years there has been a growing interest in high-level features that can represent music content as it is perceived by the listener. These meaningful semantic descriptors allow the developing MIR systems that can relate directly to the end-user. Several studies have been conducted to extract and analyze the most relevant features in the human cognitive process. Celma et al. [18], propose a set of mid-level features (e.g. rhythm, harmony, structure) obtained from the combination of low-level descriptors and generalizations induced from manually annotated databases by through ML techniques. In Eerola and Juslin researches [19, 20] are analyzed the aspects of music that contribute to emotional expression. The results showed that few qualitative features

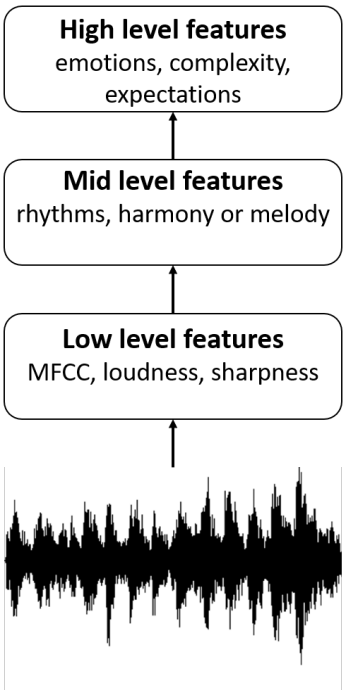

Figure 1.1: *The different levels of music features.*

are capable of describing a major part of the listener responses (e.g. happiness, sadness, fear rating), obtaining an explained variance amounted to nearly 90%. Friberg et al. [21] select a set of nine features (speed, rhythmic clarity, harmonic complexity) from psychology studies to describe overall properties of the music based on human perception. The results showed that this small number of dedicated features were superior to a "brute force" model using a large number of general audio features. Another interesting approach is the one described in [7], in which DL techniques are used to extract seven mid-level features (melodiousness, tonal stability, rhythmic complexity) related to human perception of music from an annotated dataset. As evidenced by the studies cited above, it is possible to extract and use high-level descriptors to understand and, consequently, to condition human perceptions. In our study, we focus on complexity. In particular we analyze its properties and relevance in music, then we explore the use of this descriptor in Automatic Music Generation field.

1.1.2 Complexity

Complexity is a concept used to describe what we consider unpredictable or counterintuitive. In general, the measure of complexity in a system is correlated to the knowledge that we have about it and our ability of decoding. Many kinds of research investigate the relation of this property with art in his different domains such as paintings, music, dance, highlighting the variation of aesthetic perception and hedonic

Figure 1.2: *Complete order, Chaos, Complete Disorder. Figure taken from [1].*

value (i.e. beauty, pleasantness) in response to different complexity stimuli [22, 23, 24, 25]. Streich proposes two perspectives in the analysis of complexity, the Formal view and the Informal View [26]. The first one is related to the approach in the scientific field, defining this property using a mathematical formula to describe the problem independently from the observer. The study in this direction, such as informational theory, relate complexity with the quantity of information transmitted by a source, derived from Shannon's theorem. In our case we focus on the informal view that it is more suitable in describing the everyday and in no-scientific understanding of cognitive problems.

Complexity research is usually defined somewhere in the space between "order and disorder" [27]. In defining an universal representation of this concept, Klamut describe the complexity as a function of system state based on the non-linearly transformed entropy [28]. He describes a full ordered state and a full disordered state as the extremes with no complexity. The first contains the maximum possible symmetry so it can not be defined as complex. At the opposite, the information is not long kept in the system because completely dissipated. He defines the system with the highest level of complexity relies "on the edge of the chaos" between these two states. [29]

This concept is well highlighted in Edmonds's experiment represented in figure 1.2 [1]. While the first image is considered simple due to the symmetry, the choice of the most complex is not an obvious task. Usually the second one is indicated instead of the third. This happens because usually the last image is perceived as random, and consequently, failing to code a structure or rules, not complex. A preliminary knowledge of the data representation language can modify the evaluation of complexity, allowing the distinction between noise and a very complex behavior. In fact, indicating the second image as a pattern contained in the third one, the latter will now instead be considered as the most complex. Consequently, Edmonds defines complexity as the difficulty in formulating the overall behavior of the system, even knowing almost completely the elements that compose it and their relationships.

1.1.3 Musical Complexity

Complexity in music can be view as a multidimensional construct in relation to the different hierarchical layers that compose music. Every aspect, starting from low level such as notes to high levels such as structure, is related to this propriety. This explains why the subjective music complexity evaluation is depending on several factors. For this reason, it has been investigated complexity at different layers, focusing the study on single aspects such as rhythms [13, 30], melody [31], harmony [13, 32] or combining more properties [26, 24]. It has been discuss in several studies that certain facets of complexity can be relevant in music perception for a listener. The inverted U-shaped curve behaviour express the relation of this property with arousal potential and other hedonic dimensions (e.g. pleasantness, familiarity) [33, 34, 25, 35]. In Figure 1.3 is represented the Wundt curve, which shows the proposed inverted-U relationship between preference and arousal proposed by Berlyne. Furthermore complexity is a valid parameter in discuss the preferences of a listener. Foscarin et al. and Güçlütürk et al. [36, 37] discuss the relationship between musical background of a listener and the complexity value preference in songs and chord sequences.

In our study we investigate on harmonic complexity and we explore the use of this property in automatic composition.

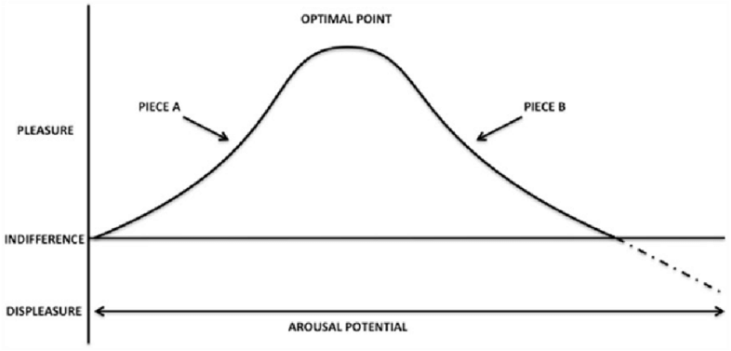

Figure 1.3: *Wundt inverted U-shaped curve, showing the relation between complexity and preference. Picture from [2].*

1.1.4 Harmonic Complexity

In music, harmony is defined as the process by which the composition of individual sounds, or superpositions of sounds, is analyzed by hearing. In our research we focus on the study of complexity in relation to this property. Harmonic complexity can be divided into three different layers: harmonic rhythm, harmonic dissonance, and harmonic evolution [38]. The first one refers to the rate of chords change over time within a meter. The second concerns the notes that compose a chord, whose intervals can be perceived consonant or dissonant, which is a measure

of how pleasant are sounds. Harmonic evolution, concerns the study of chord progression as a temporal sequence, modeling complexity as an expectation. During the listening of a musical piece, the expectation is the cognitive process by which a listener tries to predict what will happen next based on the context. It has been shown in several neural studies how the expectancy violation, generated by unexpected chord changes, is able to arouse an emotional reaction and neural activity on the part of the listener, increasing the perception of complexity [39, 40]. The choice of chords and their progression in time can therefore be considered a determining parameter in the composition of a song to condition a certain stimuli of arousal and emotions [41, 10].

In the literature, several researches has been conducted on possible models capable of representing this parameter. Lerdhal developed a hierarchical model for the perception of tonal tension and chord distances in experienced listeners [42, 43]. In his study he defines the measure of the Tonal Pitch Space Distance in order to identify the relationship between different chords and predict corresponding tension and release patterns in the progressions. Based on this study, Marvsik builds a new tonal model by measuring the harmonic complexity in relation to the deviation of a chord progression with respect to the Tonic- Subdominant-Dominant-Tonic sequence (I-IV-V-I) taken as a reference, and applying it to the measure complexity in different musical genres [32].

In our study we refer to harmonic complexity as a high-level feature expressed by Di Giorgi through a data-driven language model of cognitive harmonic expectations [13]. In his research, he propose a model that expresses the probability of a chord sequence as a descriptor of harmonic complexity, whose correlation is strongly confirmed by the results obtained through listening tests. Starting from an annotated dataset obtained by this model, we analyze through Deep Learning techniques the possibility of using harmonic complexity as a conditioning parameter in the generation of chord progressions.

1.2 Automatic Music Generation

In this section we discuss a brief overview about computer music generation. The first experiments in music generation through computer date back to the middle of the last century, exploring the use of stochastics models such as Markov Chains and rule-based systems. Nowadays, thanks to the large amount of data and improvements in the computational power, Deep Learning techniques have been revolutionizing this area of research, opening up to new generative scenarios never imagined before. The availability of large datasets containing information (from symbolic to audio formats) allows training models to learn the abstraction of concepts such as structures and rules in composition. These models were found more solid in the generalization and the description of systems than those previously implemented using analytical formulations or

manual brute force design [44, 3].

1.2.1 Generative Systems

The first experiments on algorithmic and automatic composition techniques date back to mid-20th century. Winograd [45] is one of the firsts to apply the rules of natural language in music, creating a grammar to define the relationships between notes in a sequence. Through this Rule-based system it is possible to determine which "valid" note will follow the previous one, as in the composition of sentences. Subsequently Keller et al. [46] proposes an approach to automatic generation using a probabilistic grammar, which associates a probability value to every choice. The connection between "common" sequences is represented by high values, the ones that are less used a low value. This language is able to generate improvised jazz solos over a chord progression.

Another popular model in algorithmic composition are Markov Chains, a Stochastic model in which data are modeled as sequences and the probability of the next state depends on a previous one. Based on the idea that most of the composer follows a set of rules, Bell [47] implemented a system using Markov Chains to choose the next pitch, rhythm, and chord in a composition. It also used a genetic algorithm to select the set of models that could generate the most ear-pleasing compositions. Although these models are able to generate pleasant music, they have obvious limitations. In fact, they are not able to obtain a deep knowledge of musical rules and composition techniques used by humans.

The use of neural networks has provided a major breakthrough in this field, obtaining more solid models in the generalization and the description of systems, e.g. capturing long-term dependencies, retrieve style of composing. Neural networks are trained on large datasets in symbolic or audio format. In the audio domain Google DeepMind [48] developed WaveNet, a Convolutional Neural Network (CNN) fully probabilistic and autoregressive in grado di generare raw audio, in which all previous audio samples condition the distribution of the next one. This model has been successfully applied in both text-to-speech and music generation.

In symbolic domain, Mozer [49] implemented one of the first Recurrent Neural Network (RNN) in music generation in his project CONCERT. The model generates melodies accompanied by a chord progression, producing a pitch, duration and chord at each time step. Eck et al. [50] implemented a RNN using Long Short Term Memory (LSTM) layers to generate blues chord sequences. The generation is performed giving in input a seed chord and, through the iterative feedforward of the model, the next one is produced, which will be used like new input of the network. The process is repeated until the entire sequence is created. Then, this model was extended also to generate melodies. The LSTM blocks of the chords are related to each others and to those of the melodies, so that their succession depends on both parameters. Brunner et al. [51]

implemented a model able to generate chord based on music theory concepts using LSTMs networks. The structure of the model is composed of two LSTMs networks, the first one generates chord progressions, which are given as input to the second network that generates polyphonic music in relation to the chords. MusicVAE is a Variational Autoencoder (VAE) developed by Google Magenta [52] able to model sequences with long-term structure using an encoder composed of bidirectional LSTMs layers and a hierarchical RNN for the decoder. The model can create new compositions by interpolating the sequences encoded into the latent space.

These models show the possibility of obtaining systems that can understand concepts of music theory and composition and use them to create new music.

1.2.2 Complexity as a parameter of generation

The goal of our study is to build a neural network able to model harmonic complexity in order to condition the generation of chord progressions. As previously discussed, the relationship of complexity with arousal and aesthetic perception can lead to interesting implications in music composition. Several studies today are addressing the possibility of a human-level control and interaction in the generation process, in order for the machine to compose according arbitrary constraints. Conditional Architectures are networks parameterized on the basis of an extra conditioning information, such as a class label or a feature vector. The goal of these types of architectures is to provide the ability to have a control over the data during the generation. In the architecture implementation, this extra information is usually fed into the network as an additional and specific input layer. Figure 1.4 illustrates the structure of a conditional architecture. The conditional layer is usually a simple input layer, e.g. a genre label, or an output of some network (the same or another one).

Meade at al. [11] explores conditioning-based controls used to influence the generation process in an RNN model formed by LSTM layers. Some of the controls analyzed are velocity, composer style, major-minor or even a combination of these features. The conditioning in generation of sequences of rhythms is discussed by Makris [53], which uses as extra input a bass line or a beat structure. The previously discussed Wavenet [48] model is made conditional in order to guide generation using an additional tag as conditional input, such as the musical instrument. MidiVAE is a VAE that can create polyphonic music with multiple instrument tracks, modeling the dynamics through note duration and velocity and giving the possibility of manipulate different genre style (e.g. classical, jazz) [54]. The model can apply style transfer and interpolate between short musical pieces, manipulating pitches, dynamics and instrumentation in the latent space between the two tracks.

Research on complexity-related generation is still poorly explored in

the literature. Among the main reasons is certainly the difficulty in finding datasets with numerous data annotated according to this property. As previously mentioned, to solve this problem we rely on data obtained by Di Giorgi et al [13]. In our study we implement two conditional architectures based on VAE, in the former we concatenate the conditional layer as input to the standard model, in the second as input to a Regressor.

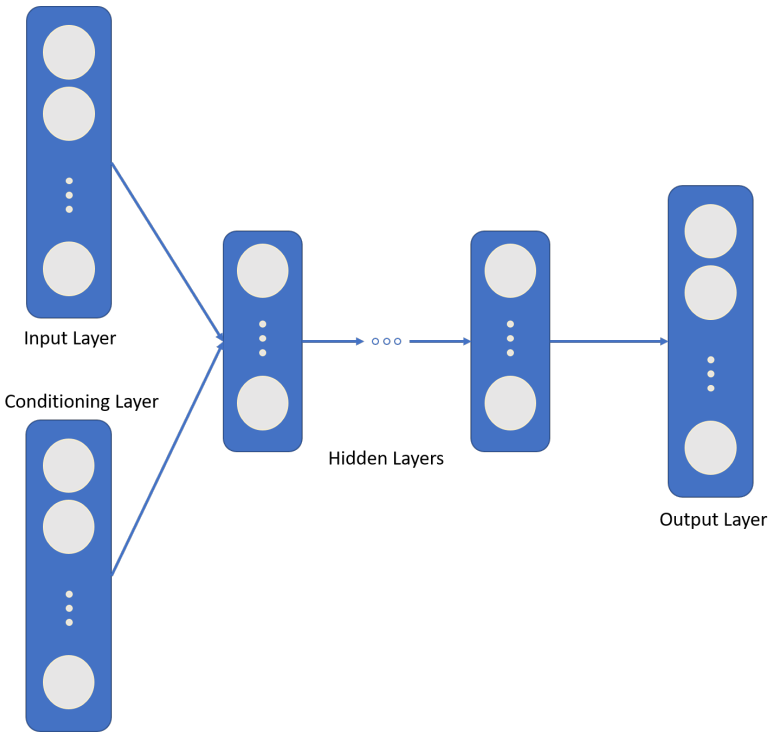

Figure 1.4: *Conditional Architecture. The conditioning layer provide the possibility to control the data generation process.*

2
Theoretical Background

In this chapter we discuss the theoretical background necessary for the understanding of the topics presented in this thesis. We describe some musical concepts and notation related to music and Tonal Harmony. Then, we give a general overview on neural networks, first presenting their functionality and then focusing on Variational Autoencoder (VAE), that is the starting generative model architecture of our study.

2.1 Musical Background

Following, we provide basic music knowledge such as pitch, chords composition and harmonic progression. Then, we describe several formats used to represent musical content.

2.1.1 Pitch, Intervals and Scale

Pitch is a perceptual quality of sounds that allows us their ordering on a frequency-related scale extending from low to high [55]. An octave is the distance between one musical pitch and an other that is double or half its frequency. In human perception of music two pitches at distance of one octave are perceived has the same pitch, with different high. For this reason, in music, their notations are defined with the same names. This phenomenon is defined as octave equivalence. In this study the terms pitch, tone and note are used as synonyms.

The equal temperament system divides each octave in 12 pitches, using this formula:

$$f_p = 2^{1/12} f_{p-1}, \tag{2.1}$$

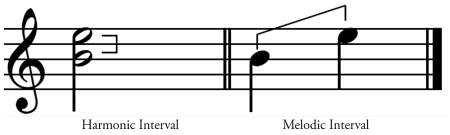

Figure 2.1: *Harmonic Interval (simultaneous tones) and Melodic Interval (consecutive tones).*

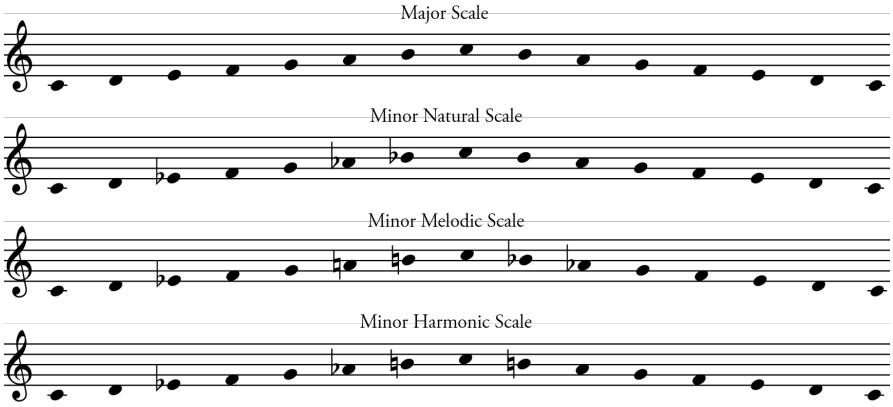

Figure 2.2: *The most used scales in Western music: major Scale and minor scale (natural, melodic and harmonic).*

where f_p is the frequency of a pitch. The interval is the distance between two notes and is defined by the frequency ratio (i.e. 2:1 for two notes an octave apart). The interval between to notes played simultaneously is defined as *harmonic interval*, while the *melodic interval* refers to two notes played consecutively. An example is provided in Figure 2.1.

A semitone or half-step is the smallest music interval. The interval composed of two semitones is called whole tone or step. Pitch can be labeled using a combination of letters and numbers, as in scientific pitch notation, assigning a frequency to each pitch in a specific octave (Table 2.1). In temperament system the frequency of the pitches is tuned relative to the standard reference: $A4 = 440Hz$.

Scales are an ordered set of musical pitches that covers the range of an octave. The different types of scales are classified according to intervals between successive notes composing it. In western music the most popular are major and minor (natural, melodic and harmonic) whose intervals are shown in Figure 2.2.

Intervals can be defined in reference to a certain scale and indicated as degrees, written with Roman numerals. The root is considered the first degree and starting from this the others are defined according to their distance.

2.1.2 Chords

A chord is a combination of three or more notes played simultaneously. Name and type of chords are classified by the number of notes and the

Pitch Class	Octave 2	Octave 3	Octave 4	Octave 5	Octave 6
C	66Hz	131Hz	262Hz	523Hz	1046Hz
C♯/D♭	70Hz	139Hz	277Hz	554Hz	1109Hz
D	74Hz	147Hz	294Hz	587Hz	1175Hz
D♯/E♭	78Hz	156Hz	311Hz	622Hz	1245Hz
E	83Hz	165Hz	330Hz	659Hz	1319Hz
F	88Hz	175Hz	349Hz	698Hz	1397Hz
F♯/G♭	93Hz	185Hz	370Hz	740Hz	1480Hz
G	98Hz	196Hz	392Hz	784Hz	1568Hz
G♯/A♭	104Hz	208Hz	415Hz	831Hz	1661Hz
A	110Hz	220Hz	440Hz	880Hz	1760Hz
A♯/B♭	117Hz	233Hz	466Hz	932Hz	1865Hz
B	124Hz	247Hz	494Hz	988Hz	1976Hz

Table 2.1: *The scientific pitch notation assigns a frequency to each pitch in a specific octave.*

intervals between them. Chords are build starting from the root tone adding intervals of a third. In pop music the main type of chord used are triads, which are composed by root, third and fifth. This chords can be classify in 4 categories depending on the intervals between the notes, as indicated in Tab 2.2. Triads can be extended by superimposing another third, obtaining quadriads defined as *seventh chord*. It is possible to create further chords with more note but in our study we focus at most the one composed by 4 notes. Furthermore we also consider *fifth chords* (or power chords) that are defined as bichords, composed only by root and fifth.

An important aspect of music composition it is *chord voicing*, that refers to how notes are arranged and ordered within a chord. One common operation is the *harmonic inversion* or simply *inversion*, that is the rearrangement of the top-to-bottom elements in an a chord, moving the notes in different positions with respect to the one in the root position (bass note). We provide an example of this technique in Figure 2.3. This technique is often used by composers to change the musical perception of a listener modifying the way he/she perceives the chord and how it relates to the others in a progression. The connection between note movements in a progression is defined as *lead voicing*.

Triad type	Lower Interval	Upper Interval
Major	4	3
Minor	3	4
Augmented	4	4
Diminished	3	3

Table 2.2: *Triads type depending on the intervals that compose the chord.*

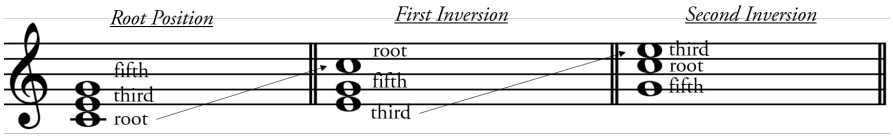

Figure 2.3: *The three possible inversions of a triads. Harmonic inversion rearrange the notes in a chord so that the original bottom note becomes an upper note.*

2.1.3 Harmonic progression

An harmonic progression is sequence of chords played one after the other in time. Chords in a progression can be defined using two types of notation:

- the pitch class of its root note and the type (e.g. major, minor) i.e. C - Am - G - F
- Roman numerals indicating the degree of the chord in relation to the root of the reference scale i.e. I - VI - V - IV

The first type of representation is explicit and indicates directly which are the chords that form the progression, while the second describes the chords in relation to the degrees of the scale on which they were built. This notation is therefore "absolute" and is easy to transpose in different keys, but requires musical knowledge to recognize the chord that is built on the specific root using the notes of the scale.

Harmonic functions describe the role that a particular chord has in the creating of a larger harmonic progression in relation to his degree tendencies. Tendencies are style/context specific and create expectation in the listener, such as tension or resolution. In common practice chord degrees are clustered in 3 groups based on their harmonic function, tonic (I-III-VI), pre-dominant (II-IV) and dominant (V-VII). Although there are many possibilities of create different chord sequences, these are usually limited to a few bars' lengths and with the use of certain chords. In general, progressions are repeated in a composition and identify its different structures, such as verse or chorus. In addition, the use of some chords progressions can identify certain musical genres (e.g. 12-bar blues) or a composer writing style.

2.1.4 Music Representation formats

Musical content can be express with different representation and encoding depending on the type of input and output of the architecture used. Despite the differences between the audio and symbolic representation, in many Deep Learning network the way the information is processed it is basically the same, as only numerical values and operations are considered.

In the audio domain the musical content can be represented by raw signal Waveform, indicating the amplitude of the signal in time, an example is provided in Figure 2.4. Usually in audio analysis are used its transformed representation such as spectrogram and chromagram. The first one describes the frequency and rhe intensity, while the second one the pitch classes and the intensity, all respect to time axis (Figures 2.5, 2.6).

In symbolic domain a very common format is MIDI, that is a technical standard describes a protocol of communication between various electronic musical instruments. This format carries real time event messages related to the data of the note played, in note on and note off events are:

- the channel number, indicates the instrument or the track. Represented by an integer in $0, 1, \ldots, 15$.
- the note number, indicates the pitch. Represented by an integer in $0, 1, \ldots, 127$
- the velocity, describes how loud the note is played. Represented by an integer in $0, 1, \ldots, 127$.

The duration is indicated in a data structure containing the delta-time value, which specifies time information and event types.

Piano Roll is a format inspired by the perforated roll of paper used for pianola and represent the list of notes contained in each time steps in two-dimensional plane. The figure 2.7 shows an example of this representation, in which the y-axis indicates the notes contained in the different time instants, described by the x-axis. In our research we use this format to encode the information about chord progressions. There are also other type of symbolic representation such as text, markup language, lead sheet that we not discuss here, but can be further explored in [3].

2.2 Deep Learning Background

Here we present an overview of the Neural Networks providing the basic concepts to understand their functioning and structure. We discuss the

Figure 2.4: *Example of waveform, the x-axis represents the time, the y-axis the amplitude of the audio signal (original from [3]).*

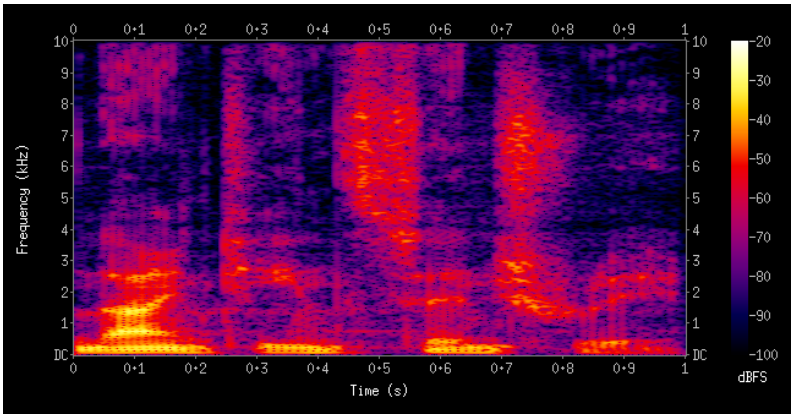

Figure 2.5: *Example of spectrogram, obtained from the audio signal via a Fourier transform. X axis: time (in seconds), Y axis: frequency (in kHz) and the third axis: intensity of the sound, expressed with color. Aquegg's original image at "https://en.wikipedia.org/wiki/Spectrogram"*

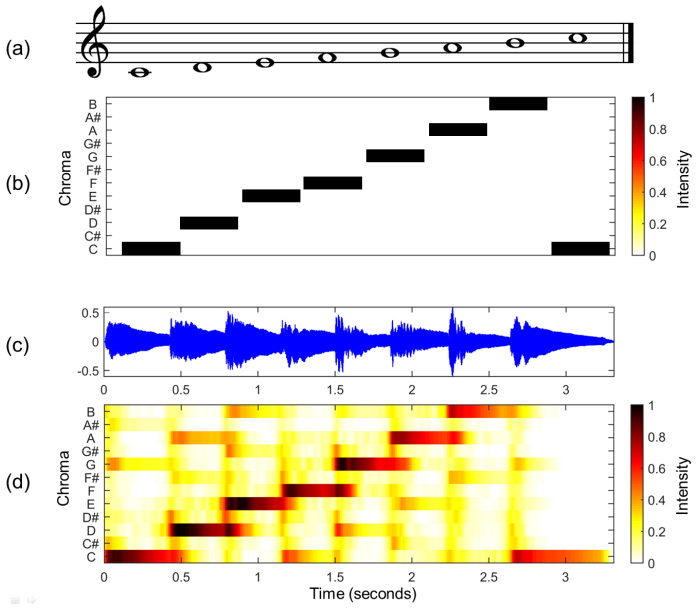

Figure 2.6: *Examples of chromagrams. (a) Musical score of a C-major scale. (b) Chromagram obtained from the score. (c) Audio recording of the C-major scale played on a piano. (d) Chromagram obtained from the audio recording. The image is reproduced from Meinard Mueller's original one at "https://en.wikipedia.org/wiki/Chroma feature" under a CC BY-SA 3.0 licence*

Figure 2.7: *Example of Piano Roll format. The x-axis represent the time, the y-axis the notes played.*

architecture of the base model and the process of training and testing of the network. Then, we describe the generative and the discriminative approaches. Finally, we focus on VAE.

2.2.1 Artificial Neural Networks

An Artificial Neural Network (ANN) or Neural Network is an interconnected structure of simple computational units, also defined as neurons or nodes, whose functionality is loosely inspired by the biological thinking of the human brain. Each connection can transmit a signal to other neurons, then the information is processed and sent to others units in the network. The signal is represented by a real number and the output of each neuron is obtained from the sum of the input signals using some non-linear function, defined as activation function.

The mathematical formulation is:

$$y = f(\sum_{i=1}^{n}(w_i x_i + b)), \tag{2.2}$$

where b is the bias, x represent the input signals multiply for its weights w and f indicates a type of activation function. The most commons function used in neural networks are ReLU, Sigmoid and Tanh whose mathematical formulas and graphs are described in Figure 2.8. We represent in Figure 2.9 a basic scheme of the artificial neuron. The processing ability of the network is stored in the inter-unit connection weights associated to neurons and signals, obtained by a process of adaptation to, or learning from, a set of training data [56]. Neurons are usually grouped in layers that may perform different transformations on their inputs. The basic architecture of the ANN is composed by input, hidden and output layer (Figure 2.10). In feed-forward networks, the signal crosses the layers from

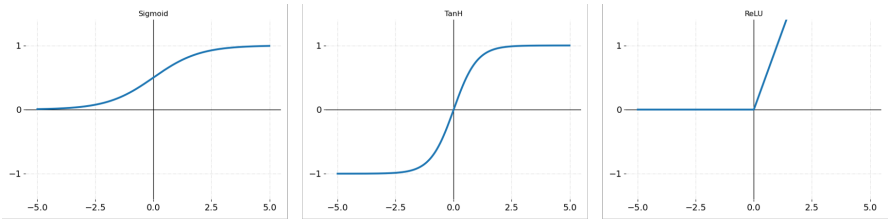

Figure 2.8: *Plots and formulas of the most used activation functions in Neural Networks: Sigmoid, TanH and ReLU.*

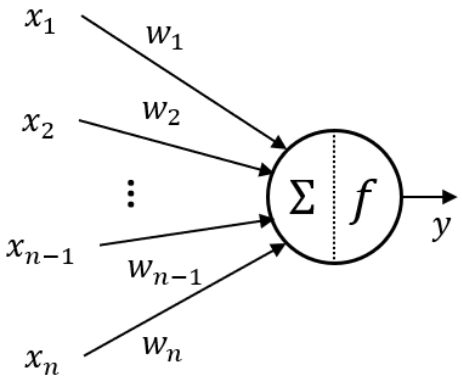

Figure 2.9: *The scheme of an artificial neuron in feed-forward network: in input receives from the previous layers different signals i, and each one has an associated weight w. The neuron calculates the sum of these inputs and passes the result through the activation function f, returning the output.*

the input to the output, through the n hidden layers. Neural Networks are mainly used for tasks such as classification (e.g. label spam and non-spam emails), prediction (e.g. weather forecasting) or generation (e.g. create musical compositions or paintings).

2.2.2 Training and Loss function

In order to objectively evaluate the network output and thus the quality of the model, we need to define loss function. This function calculates the distance between the output y obtained by the model and the desired one $\hat{y}$ (ground truth), evaluating the learning capabilities of the network. Among the various types of loss function, for the purpose of this thesis we focus on:

- Mean Squared Error (MSE):

$$\frac{1}{N}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{2.3}$$

  The formula calculates the average squared distance over the N elements between the estimated values $\hat{y}$ and the desired ones y.

- Cross Entropy:

$$CE = -\frac{1}{N}\sum_{i=1}^{n}(y_i log(\hat{p}_i) + (1 - y_i) log(1 - \hat{p}_i)), \tag{2.4}$$

  where N is the number of values in the model output, $\hat{p}_i$ is the probability of an event x_i to belong to the class 0 or 1, and y_i is the class that it actual belongs to.

In the construction of a neural network, the dataset is usually divided in three sets: training, validation and test. The first is composed of the data used in the training process, the procedure by which the network learns its parameters. The validation set is used to provide an unbiased of a model fit on the training dataset while tuning its hyperparameters (e.g. number of layers). The test set is used to provide an unbiased evaluation of the final model fit on training dataset.

During the training of the model is calculated the best set of weights for maximizing a neural network's objective. This is performed through the gradient descent optimization algorithm, which aims at finding the parameters of the network that minimize the loss function. The algorithm is based on moving iteratively in the opposite direction of the gradient of the function at the current position, which indicates the direction to the steepest descent. The learning rate η is an hyper-parameter that defines by how much the weights w have to be updated at each iteration k according to the equation:

$$w^{k+1} = w^k - \eta \nabla E, \tag{2.5}$$

with ∇E is the gradient of the cost function $E(w)$.

The *Stochastic gradient descent* is a variant of gradient descent used when the training data is large and the standard version may be inefficient and slow. In the calculation of the gradient, this method only considers one or a subset of samples from the training set. *Backprogation* is the standard method of estimating the gradients for a multilayer neural network and is based on the chain rule principle. It estimates the contribution of each weight on the final loss value of the neural network [57].

One of the fundamental issues of neural networks is their generalization ability on the problem, which is the ability of the model to perform well on unseen inputs. During the training process of the network, the

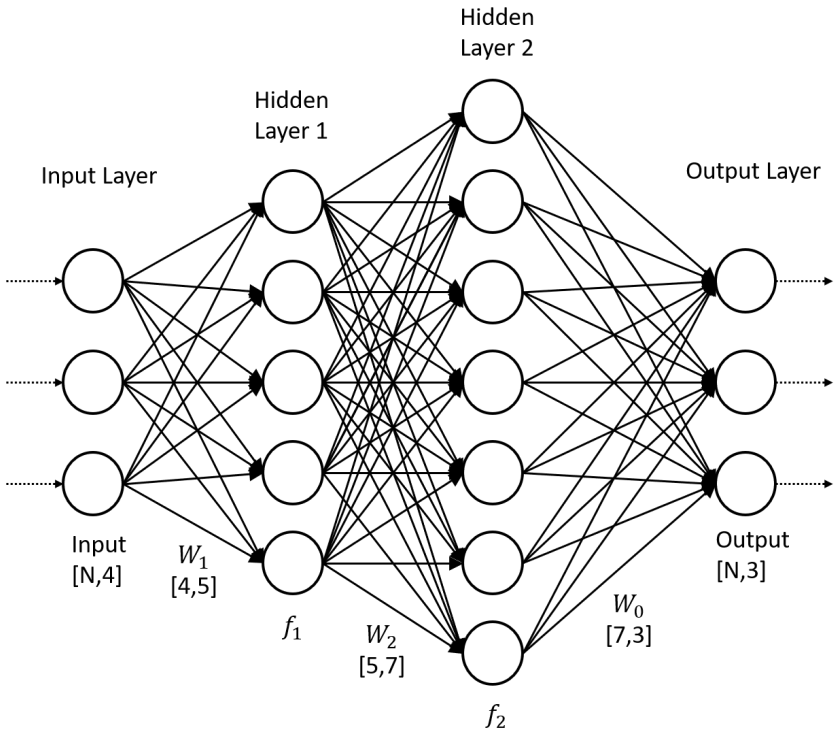

Figure 2.10: *Example of ANN, composed by 2 hidden full connected layers, respectively with 5 and 7 hidden units.*

error on the training data is reduced as an optimization process over the model parameters. The objective searched in the models is to obtain also a low testing error, that is the error on the unseen data, index of good generalization of the problem by the model. The two central problems on this issue are: *underfitting* and *overfitting*. The first occurs when the error measure on the training data is large, this happens when the model is not complex enough to accurately capture relationships between features and a target variable. The second occurs when the expected error on new data is large, this happens when the model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model of testing data. A graphic example is shown in Figure 2.11.

In order to face this topic it is not possible to define of the universal methods, but to modify the flexibility of the model (its ability to fit a variety of functions) so as to obtain the best structure in order to face a determined problem.

2.2.3 Discriminative and Generative Models

In ML and DL we can distinguish two different types of approaches to address a given task: the discriminative approach and the generative approach. The former models the decision boundary between classes in

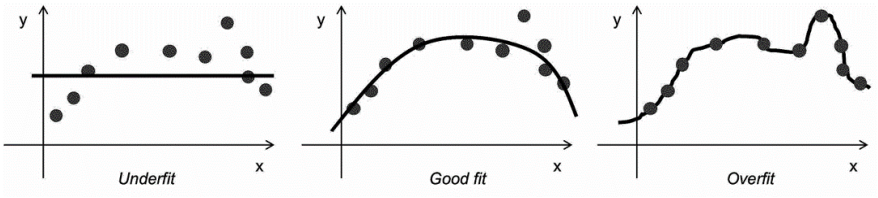

Figure 2.11: *Example of underfit, good fit and overfit of the data. In the first and the last case the network model does not perform correctly (original image from [3]).*

the dataset, the latter the distribution of classes, allowing new data to be generated.

Specifically, given x as the observed data and y as its target label, discriminative models capture the conditional probability $p(Y|X)$ and classifies different data instances to the correspondent class label, learning a direct map from inputs x to the y. The objective of these models is therefore to identify the decision boundary between classes identifying the parameters that allow to correctly label the data instances, using conditional probability. Some examples are the Logistic Regression or the Support Vector Machine.

Generative models instead capture the joint probability $p(X, Y)$, or just $p(X)$ if there are no labels, and learn to model the entire data distribution. The goal of these models is to find the parameters capable of explain all the data. Some examples in this field are the VAE, the Generative Adversarial Networks (GAN) and Auto Regressive models. This type of models can be used to generate new data sampling from the distribution $p(X, Y)$, also respect some property or the interpolation of data.

A further difference between the two types of models is the posterior probability $p(Y|X)$. In discriminative models it is inferred from a parametric model, where the parameters come from the training data and it is used to maintain the least expected loss, minimizing the misclassification results. In generative models, which focus on the joint probability, the class posterior possibility $P(Y)$ is considered as in Bayes' theorem [58]:

$$p(y|x) = \frac{p(x|y)p(y)}{\sum_i p(x|i)p(i)} = \frac{p(x|y)p(y)}{p(x)}. \tag{2.6}$$

Further information and comparisons between these two types of models are discussed in [59]. In our study we focus on generative models in music. In the rest of the section we describe the VAE, that is the starting model used in our research.

2.2.4 Variational Autoencoders

The VAE is a neural network model introduced by Kingma and Welling [60]. It learns a compact representation of the distribution of the data, allowing to sample from this distribution to create new data. The architecture of this model is based on Autoencoder (AE), a network model whose goal is to learn a compressed representation of data. It consists of the encoder, which compresses the input data (encoding), and the decoder, which attempts to reconstruct the original inputs as good as possible from their reduced representation (decoding). In VAE, instead of the AE, latent vectors should follow a unit Gaussian distribution. Defining x as a random input variable of data samples and z the set of latent variables, the generative model is defined by:

$$p(x, z) = p(x|z)p(z), \tag{2.7}$$

where $p(z)$ is the prior distribution of the latent variables, and $p(x|z)$ the conditional likelihood distribution.

The structure of the VAE is composed by the encoder $q_\phi(z|x)$, which approximates the posterior $p(z|x)$, and the decoder $p_\theta(x|z)$, which parameterizes the likelihood $p(x|z)$, where ϕ and θ are respectively the parameters of the encoder and decoder. Following the framework of Variational Inference, the optimization of encoder and decoder is done by maximizing the variational lower bound, also defined as Evidence Lower BOund (ELBO):

$$\mathcal{L}(\phi, \theta, \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{x})}\big(\log p_\theta(\mathbf{x}|\mathbf{z})\big) - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})) \leq logp(x). \tag{2.8}$$

The first term is the reconstruction loss, or expected log-likelihood, and the second one, D_{kl}, is the Kullback–Leibler divergence, that measures the difference between two probability distribution P and Q. In detail, $D_{kl}(Q||P)$ defines the amount of information lost when Q is used to approximate P [61]. It is defined as:

$$D_{kl}(p||q) = \sum_{i=1}^{n} p(x_i) \cdot \frac{\log(p(x_i))}{q(x_i)}, \tag{2.9}$$

where n is the number of observations. If the D_{kl} has value 0 the two distributions are identical. The calculation of the gradient through the ELBO is not possible during the training, due to the fact that z is randomly sampled, which makes it non-differentiable. This problem is usually addressed through the application of the reparametrization trick, considering $p(z)$ a diagonal-covariance Gaussian and replace $z \sim N(\mu, \theta I)$ with:

$$\epsilon \sim N(0, I). \tag{2.10}$$

$$z = \mu + \theta\epsilon. \tag{2.11}$$

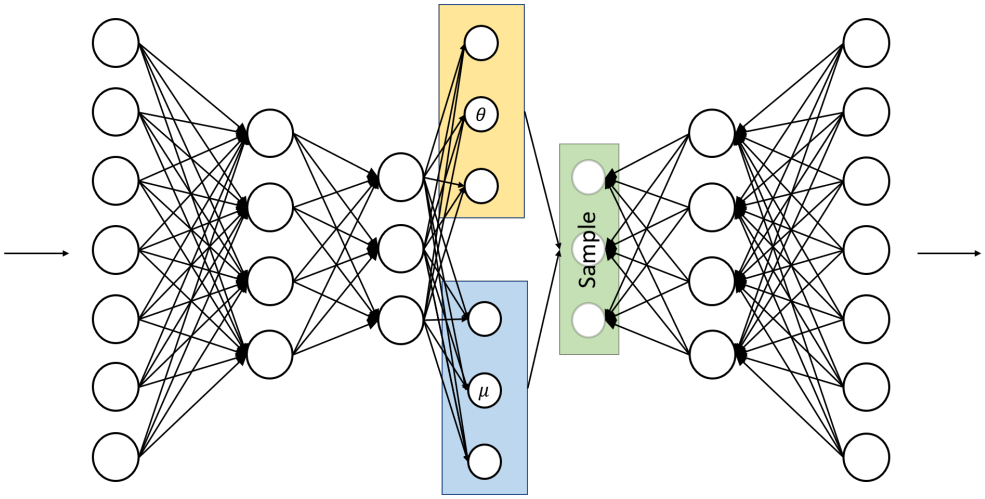

Figure 2.12: *Example of VAE architecture. The first part consists of the encoder, the middle part is the latent space, which is the reduced representation of the input, and the final part is the decoder.*

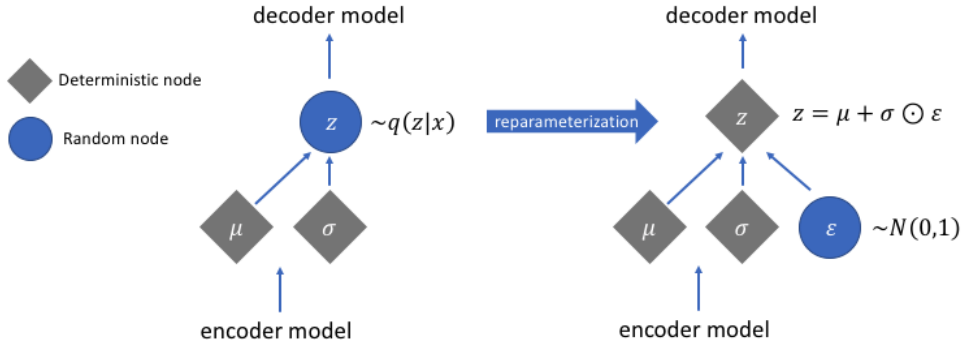

Figure 2.13: *Reparametrization trick: is used to allow the calculus of the gradient descent despite the random sampling in the VAE architecture.*

In this way z is expressed with μ and σ, that are two parameters that we learn during the training, plus a random fixed value, Figure 2.13.

3 Models

In this chapter we discuss the network models implemented in order to generate new chord progressions using harmonic complexity as a conditional parameter. We describe the initial approach to the problem, consisting of different standard Variational Autoencoder (VAE) architectures and show how data related to chord progressions are represented in the obtained latent space.

Then, we focus on the Conditional Architectures, which are models parameterized by a conditional information e.g. class label or a feature vector. The goal of these networks is to provide some control over the data generation process. We present the two implemented conditional models, denoted as Conditional Variational Autoencoder (CVAE) and Regressor Variational Autoencoder (RVAE), describing the architectures and the data distribution in the respective latent spaces. Finally, we analyze the possibility of generating new chord progressions using the implemented network models.

3.1 General Formulation

The goal of our study is to devise a neural network model capable of generating chord sequences using harmonic complexity as a generation parameter. As discussed in Sec. 1.2.2, there are several network models that are able to automatically generate music conditioned by a certain style or genre. However, to the best of our knowledge, the use of musical complexity as a generative parameter was not explored in the literature. In our study we focus on deep latent variable models such as the VAE. The advantage of these models is the ability to capture the pertinent characteristics of a given datapoint and disentangle factors of variation

C	C#	D	D#	E	F	F#	G	G#	A	A#	B
1	0	0	0	1	0	0	1	0	0	0	0

Figure 3.1: *The representation of C major chord (composed by the notes C,E,G) in chroma vector format.*

in a dataset in the latent space representation. We approach the problem first with the standard VAE model, then we proceed to explore conditioning architectures in order to condition the generation process according to the complexity.

3.1.1 Data Representation

As described in 2.1.4, audio or music content can be represented in different formats (e.g. MIDI, wav). In our study, the data consist of sequences of chords to which is associated an harmonic complexity class. The data are represented in the symbolic domain. Each chord is described by a multi-hot vector $\mathbf{x} \in \mathbb{R}^{1\times N_p}$, where N_p is the number of pitch classes$\{C, C\sharp, D, D\sharp, E, F, F\sharp, G, G\sharp, A, A\sharp, B\}$. The positions corresponding to the notes contained in the chord have value 1, the remaining are 0. This type of format is also defined as chroma vector. We show in Figure 3.1 the multi-hot vector representing the C major chord.

The chord sequences are encoded by combining the chroma vectors together, resulting in a representation of the data in a Piano Roll format. In our dataset the chord progressions are composed of M chords, therefore the inputs of our model are represented by a matrix $\mathbf{X} \in \mathbb{R}^{M\times N_p}$. In Figure 3.2 we present an example of chord progression matrix, where rows represents the chords and columns the pitch classes. Each sequence is associated with a class of harmonic complexity. This mapping is defined by Di Giorgi through a data-driven language model of cognitive harmonic expectations [13]. Di Giorgi defines the probability related to a certain chord sequence as a descriptor of its harmonic complexity. High probability values are associated to the "simplest" progressions, while low probability are associated to the progressions perceived as more complex.

This value is represented with an integer c or by using one-hot encoding vector $\mathbf{c} \in \mathbb{R}^{1\times N_c}$, where N_c is the number of complexity classes. We provide an example of this two formats in Figure 3.3.

3.1.2 Standard Variational Autoencoder

The first model we discuss is an implementation of the standard VAE. We define with $\mathcal{X}^{(train)} = \{\mathbf{X}^{(1)}, ...\mathbf{X}^{(n)}\}$ the training set containing chord sequences and $\mathcal{C}^{(train)} = \{c^{(1)}, ...c^{(n)}\}$ their harmonic complexity value. Each chord sequence $\mathbf{X}$ is associated with a latent representation $\mathbf{z} \in \mathbb{R}^k$, where k is the dimension of the encoding in latent space. Given

1	0	0	0	1	0	0	1	0	0	0	0
0	0	1	0	0	1	0	0	0	1	0	0
0	0	0	0	1	0	0	1	0	0	0	1
1	0	0	0	0	1	0	0	0	1	0	0
1	0	0	0	1	0	0	1	0	0	0	0

Figure 3.2: *The 5x12 matrix representing the chord progression composed by 5 chords. The sequence in this example is: C maj, D min, E min, F maj, C maj.*

3 , | 0 | 0 | 0 | 1 | 0 |

Figure 3.3: *Two types of encoding formats used for the conditional information. In the example we show the representation of class 3 out of 5 possible (0 to 4). On the left the label class is identified with an integer value, on the right using one-hot encoding.*

the set of chord sequences $\mathbf{X}^{(train)}$, the VAE uses an encoder to convert input $\mathbf{X}$ into a latent vector $\mathbf{z}$ of smaller dimension, that represents the distribution over the learned features of the data. Then, the second part of the architecture is composed of the decoder, which samples from this distribution and reconstructs the original chord sequences. We have discussed in detail the theory behind the VAE in 2.2.4.

In this initial approach we focus only on analyzing the distribution of data in the latent space. The original implementation of the VAE is unsupervised, in fact, the network receives only $\mathbf{X}$ as input data. Consequently, the model organizes the distribution of the data in the latent space according to the features that is able to infer from the data. We implement two VAE architectures to evaluate the possibility of obtaining a latent space clustered with regard to the harmonic complexity value of the chord sequences, without giving such information as input to the network

In the first model we implement a network composed of full-connected layers, we present its structure in Tab. 3.1. The number of neurons decreases in the dense layers that form the encoder, up to two, which is the latent space dimension k. The decoder is composed by the same layers of the encoder positioned in reverse order. In this second part of the network the number of neurons increases to reconstruct the input from the reduced representation.

The second model implemented is a Recurrent Neural Network (RNN) composed by Long Short Term Memory (LSTM) layers, in order to analyze the dependencies between the chords $\mathbf{x}$ in the sequence $\mathbf{X}$. The architecture of this network described in Tab. 3.2. As we described in

Layer	Number of Neurons	Activation Function Used
Dense	512	ReLU
Dense	256	ReLU
Dense	128	ReLU
Dense	32	ReLU
Dense (z mean)	2	Linear
Dense (z log var)	2	Linear
Lambda (z)	2	-
Dense	32	ReLU
Dense	128	ReLU
Dense	256	ReLU
Dense	512	ReLU

Table 3.1: *VAE architecture composed of full-connected layers*

section 3.1.1, the chord progression $\mathbf{X}$ are represented by 0 and 1 values, so the reconstruction loss function used in both models is the binary cross-entropy (section 2.2.2).

In Figure 3.4 we show the graphs representing the distributions of the sequences of chords in the latent spaces obtained from the full-connected model (a) and the RNN (b). The data are represented in relation to their harmonic complexity value. We analyzed the distribution of the data according to the value of harmonic complexity using this plots and, in both cases, not evident clusters according to the this feature. The RNN model is able to distinguish two clusters of data, which, as a result of analyzing the encoded progressions and decoding the points in latent space, represent the major and minor sequences.

Therefore, it is not possible to use these two implemented models to generate chord sequences based on a given value of harmonic complexity. The VAE model generates new data by sampling z from $\mathcal{N}(0, \mathbf{I})$ through the decoder network $p_\theta(\mathbf{X}|z)$, so since there are no evident data clusters it is not possible to control the generation process according to this parameter.

Following the results obtained with the standard VAE, we modified the network architecture, in order to condition it explicitly according to the feature of our interest. Specifically, we devised two models able to model the harmonic complexity value during the data generation process.

3.2 Conditional Variational Autoencoder

In this section we consider Conditional Architectures, which are networks characterized by the addition of the conditioning feature as an additional input layer to the network model. Specifically, we implement the CVAE, an extension of the VAE model which incorporates the conditioning information by concatenating the layer at the input of both the encoder

Layer	Number of Neurons	Activation Function Used
LSTM	64	ReLU
LSTM	32	ReLU
Dense (z mean)	2	Linear
Dense (z log var)	2	Linear
Lambda (z)	2	-
Repeat Vector	-	-
LSTM	32	ReLU
LSTM	64	ReLU

Table 3.2: *VAE architecture composed of LSTM layer*

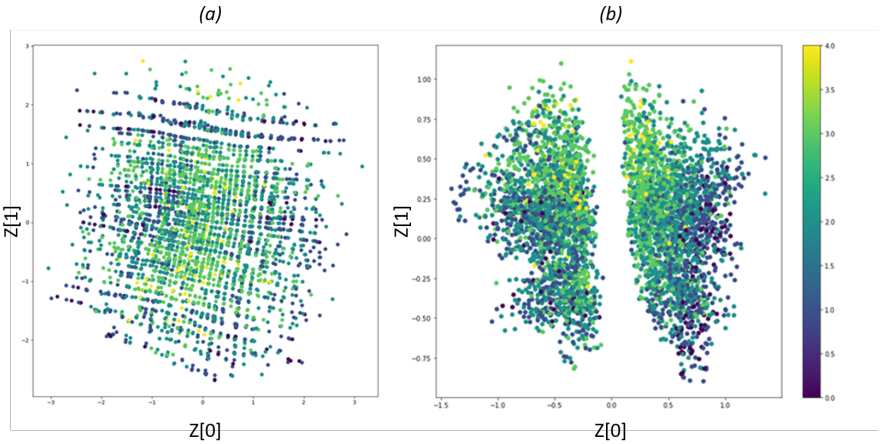

Figure 3.4: *Plot of the latent space of the two standard VAE architectures implemented. In (a) is represented the one obtained from the full-connected model, while in (b) is the one from the RNN. In the graph (b) the data are clustered in major and minor chord sequences.*

and the decoder. The architecture of the network in shown Figure 3.6. This type of model provide the ability to have a control over the data during the generation process through the conditioning with the target feature [11]. Then, using the model of CVAE it is possible to generate chord progressions using values of harmonic complexity as a conditional parameter.

Given the harmonic complexity value defined as one-hot vector $\mathbf{c} \in \mathbb{R}^{1\times N_c}$ and the latent variable z, our goal is to implement a model $p_\theta(\mathbf{X}|\mathbf{c}, z)$ capable of generating new chord progression $\mathbf{X}$ conditioned on $\mathbf{c}$ and z. We define this network as the generator model, parameterized by θ. The purpose of this model is learning the best parameter to maximize the conditional log-likelihood $\log p_\theta(\mathbf{X}|\mathbf{c})$.

As discussed by Sohn et al. [62], the CVAE can be trained with the Stochastic Gradient Variational Bayes (SGVB) framework by maximizing the variational lower bound (ELBO) of the conditional log-likelihood. We introduce an auxiliary distribution $q_\phi(z|\mathbf{X}, \mathbf{c})$, to approximate the true

posterior $p_\theta(z|\mathbf{X}, \mathbf{c})$. Then, we refer to the conditional log-likelihood as:

$$\log p_\theta(\mathbf{X}|\mathbf{c}) = D_{\mathrm{KL}}\big(q_\phi(z|\mathbf{X}, \mathbf{c})||p_\theta(z|\mathbf{X}, \mathbf{c})\big) + \mathcal{L}(\mathbf{X}, \mathbf{c}; \theta, \phi), \tag{3.1}$$

where the variational lower bound is expressed by [62, 63]:

$$\begin{aligned}\mathcal{L}(\mathbf{X}, \mathbf{c}, \theta, \phi) = \mathbb{E}_{q_\phi(z|\mathbf{X},\mathbf{c})}\big(\log p_\theta(\mathbf{X}|z, \mathbf{c})\big) \\ - D_{\mathrm{KL}}(q_\phi(z|\mathbf{X}, \mathbf{c})p_\theta(z|\mathbf{c})).\end{aligned} \tag{3.2}$$

The first term of the equation is the log-likelihood of samples, while the second is the Kullback-Leibler divergence, which regularize the distance between the prior $p_\theta(z|\mathbf{c})$ and the proposal distribution $q_\phi(z|\mathbf{X}, \mathbf{c})$. We refer the auxiliary proposal distribution $q_\phi(z|\mathbf{X}, \mathbf{c})$ the recognition model (encoder), while $p_\theta(\mathbf{X}|\mathbf{c}, z)$ is the generator (decoder), and defined as multivariate Gaussian distributions. The prior $p_\theta(z|\mathbf{c})$ is assumed to follow an isotropic multivariate Gaussian distribution. Following the implementation of the standard VAE model, as we described in 2.2.4, we apply the reparametrization trick in order to calculate the the gradient through the ELBO [60].

Finally, the generation process of chord sequences is described by simply two steps:

- Sample a random latent variable z from the prior distribution $p(z)$
- Concatenate the z variable with the $\mathbf{c}$ conditioning vector and generate a new data from $p_\theta(\mathbf{X}|\mathbf{c}, z)$

We provide in Figure 3.7 a simple representation of the generator model in the standard VAE and the CVAE. Then, we show in Figure 3.7 an example of generation of a chord progression using the CVAE.

3.2.1 Network Architecture

The architecture of the network is represented in Tab. 3.3 and consists of full-connected layers. In the encoder part, the number of neurons of the first layer is greater than the size of the matrix representing the chord sequences, then is reduced in the successive Dense layers, up to the dimension two in latent space. The choice of the initial number of neurons was made because it led to better results both in terms of reconstruction of input data and their distribution in the latent space. We introduced a Dropout layer between the first two full-connected layers in order to reduce the problems due to over-fitting. The structure of the decoder is composed of the same layers as the encoder, placed in reverse order (except the Dropout layer).

The encoder input consists of the concatenation of the chord sequence $\mathbf{X}$ with the conditional feature $\mathbf{c}$, while the decoder is formed by $\mathbf{c}$ concatenated with the latent variable z. As discussed for the standard VAE architectures, the reconstruction loss function used is the binary cross-entropy, as the input are matrices containing binary values.

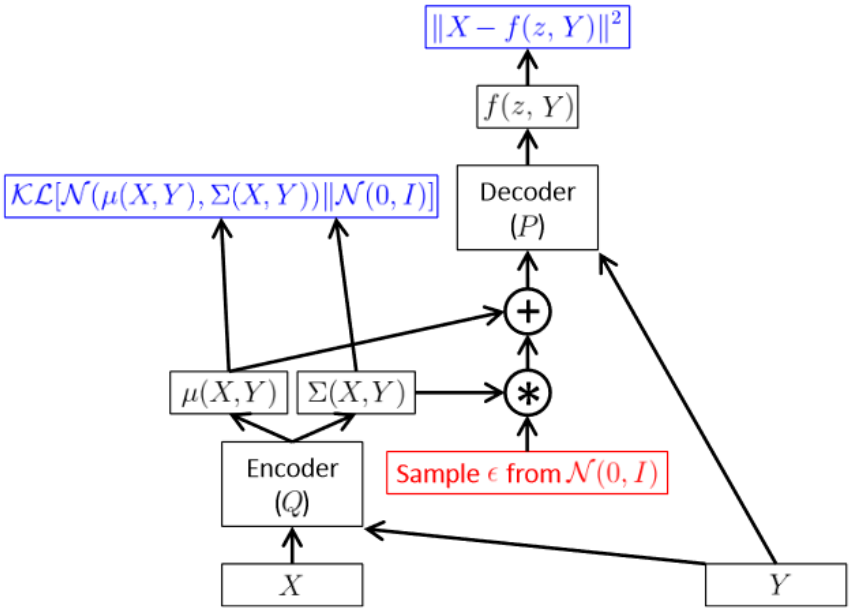

Figure 3.5: *Architecture of CVAE: X are the input data, Y are the labels, in our research we indicate Y=C [4].*

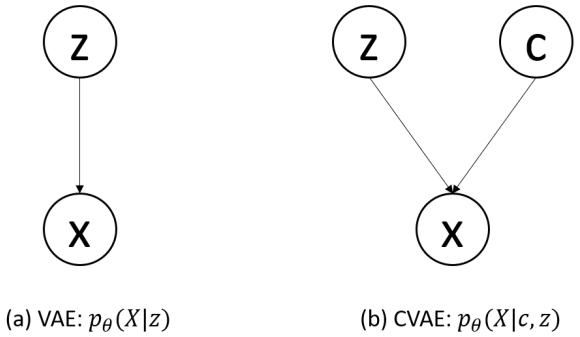

Figure 3.6: *Graphical model representations of standard VAE generator (a) and attribute-conditioned generator in CVAE (b).*

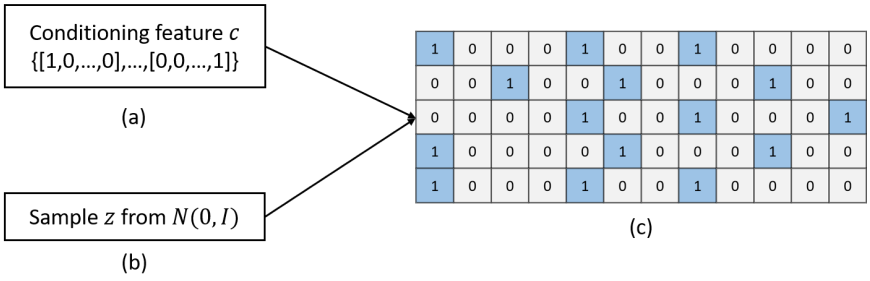

Figure 3.7: *Generator model of CVAE. Concatenating an harmonic complexity vector* **c** *(a) with a latent variable z (b), the network can generate a chord sequence* **X** *(c).*

Layer	Number of Neurons	Activation Function
Input (flat)	60	-
Input label	5	-
Concatenate	65	-
Dense	512	ReLU
Dropout	512	-
Dense	128	ReLU
Dense (z mean)	2	Linear
Dense (z log var)	2	Linear
Lambda (z)	2	-
Concatenate	7	-
Dense	128	ReLU
Dense	512	ReLU
Dense	60	ReLU

Table 3.3: *CVAE architecture. The input of the encoder and the decoder of the model are concatenated to the conditioning vector* **c**.

3.3 Variational Autoencoder and Regressor

In this section we discuss the second conditional VAE architecture implemented, which is based on learning disentangled representation of data. "A representation in the latent space is considered disentangled if the changes along one dimension are explained by a specific factor of variation, while being relatively invariant to changes in other factors" [5, 64]. This model allows us to explicitly condition the distribution of the data in the latent space with respect to the harmonic complexity, obtaining a dimension that encodes this property. This allows us to control the generation process using the harmonic complexity as a parameter by moving in latent space along the disentangled dimension.

The architecture of this model is composed by the combination of the standard VAE with a Regressor, and is based on the study conducted by Zhao et al. [5]. The structure of the network is shown in Figure 3.8. We refer to this model as RVAE. This network is characterized by the disentagled-dimension with regard to the harmonic complexity c, represented as an integer value, in the latent space. Such a result is obtained through the use of VAE combined with a Regressor, which has as input the attribute c that explicitly conditions the latent representation z of the data $\mathbf{X}$.

Given the training set of chord sequences $\mathcal{X}^{(train)}$ and the set of their harmonic complexity values $\mathcal{C}^{(train)}$, the model explicitly conditions the latent variable z on c so that $p_\theta(z|c)$ captures an attribute-specific prior on latent representation. This is defined as the latent generator, as it allows to sample a latent representation z for a given value of c from this distribution.

Zhao [5] assumes that the decoder network $p_\theta(\mathbf{X}|z)$ is able to cap-

ture the non-linearity of the generative model $p_\theta(z|c)$, then the latent generator can be parameterized with a linear model:

$$p_\theta(z|c) \sim \mathcal{N}(z; \mathbf{u}^\mathbf{T} c, ^2 \mathbf{I}), \mathbf{u}^\mathbf{T}\mathbf{u} = 1, \tag{3.3}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{u}$ is the disentangled dimension. The parameters of the RVAE can be estimated by maximizing the sum of the log likelihood $\sum_{i=1}^{n} \log p(x_i)$. This maximization is performed through the variational inference procedure and defining an auxiliary function $q_\phi(z, c|\mathbf{X})$ to approximate the true posterior $p_\theta(z, c|\mathbf{X})$. We rewrite $\log p(\mathbf{X})$ as:

$$\log p(\mathbf{X}) = D_{\mathrm{KL}}\big(q_\phi(z, c|\mathbf{X})||p_\theta(z, c|\mathbf{X})\big) + \mathcal{L}(\mathbf{X}), \tag{3.4}$$

where $\mathcal{L}(\mathbf{X})$ is the variational lower bound (or ELBO).

Based on mean-field theory, which considers that the behavior of a stochastic model can be approximated by the average value of the elements from which is composed, we assume $q(z, c|X) = q(z|X)q(c|X)$. Then, the ELBO is defined as:

$$\begin{aligned}\mathcal{L}(\mathbf{X}) = &-D_{\mathrm{KL}}\big(q(c|\mathbf{X})p(c)\big) \\ &+ \mathbb{E}_{(z|\mathbf{X})}\big[\log p(\mathbf{X}|z)\big] - \mathbb{E}_{q_\phi(c|\mathbf{X})}\big[D_{\mathrm{KL}}(q(z|\mathbf{X})p(z|c))\big].\end{aligned} \tag{3.5}$$

We define $q_\phi(c|x)$ as the probabilistic regressor and is formulated as an univariate Gaussian $q_\phi(c|x) \sim \mathcal{N}(c; f(x; \phi_c), g(x; \phi_c)^2)$, where ϕ_c are the parameters of the inference networks. The first term in Eq. 3.5 is the KL divergence, which regularizes the prediction of c with regard to a prior distribution. In our model the ground-truth of c is known for each training sample $\mathbf{X}$, so this term can be substituted by $\log q_\phi(c|\mathbf{X})$. The second term of Eq. 3.5 corresponds to the reconstruction loss, which promotes the proper reconstruction of the input data from the latent space, similarly to what is proposed in the standard VAE architecture. The last condition encourages the posterior $q_\phi(z|\mathbf{X})$ to resemble the feature harmonic complexity-specific prior $p(z|c)$. The expectation of the last two terms of Eq. 3.5 are maximized using the SGVB estimator through the the reparametrization trick [60].

We show the plot $q_\phi(z|\mathbf{X})$ in Figure 3.9, to represent the latent space obtained from the input data and the feature labels. As previously mentioned, there is a disentangled-dimension in the latent space that models the feature c, representing the harmonic complexity. It is possible to generate new chord sequences with a certain class of complexity by moving in latent space along the disentangled dimension, and in the other axis to obtain different sequences with the same value of c.

3.3.1 Implementation of RVAE

In Tab. 3.4 we present the layers that compose the model architecture. As in the model of CVAE, it is composed by full-connected layers, where

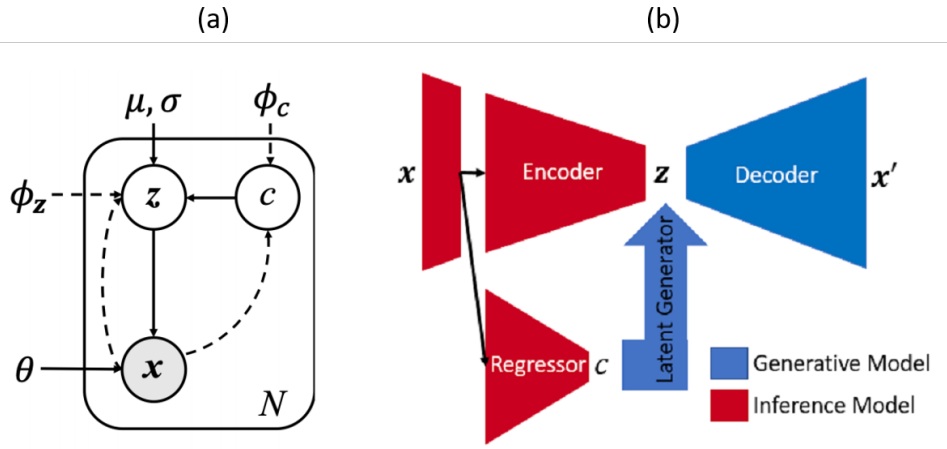

Figure 3.8: *Probabilist diagram (a) and graphical diagram (b) of RVAE. The architecture is composed by the standard VAE combined with a Regressor. Each input X is assumed to be generated from its representation z, which is dependent on complexity c. The inference model is composed by the probabilistic encoder, to obtain the latent representation, and the regressor, for predictic harmonic complexity. Picture from [5].*

in the encoder the number of neurons in the first layer is initially greater than the dimensions of the input, as this led to better results both in terms of reconstruction of input data and their distribution in the latent space. Then, is reduced in the successive Dense layers, up to the dimension two in latent space. We introduced two Dropout layers between between the first two, and between the second and third full-connected layers, in order to reduce the problems due to over-fitting. The decoder has the same layers of the encoder inversely ordered (except the Dropout layers). The structure of the regressor is represented in Tab. 3.5 and is composed by Dense layers. Also in this implementation the reconstruction loss is the binary cross-entropy.

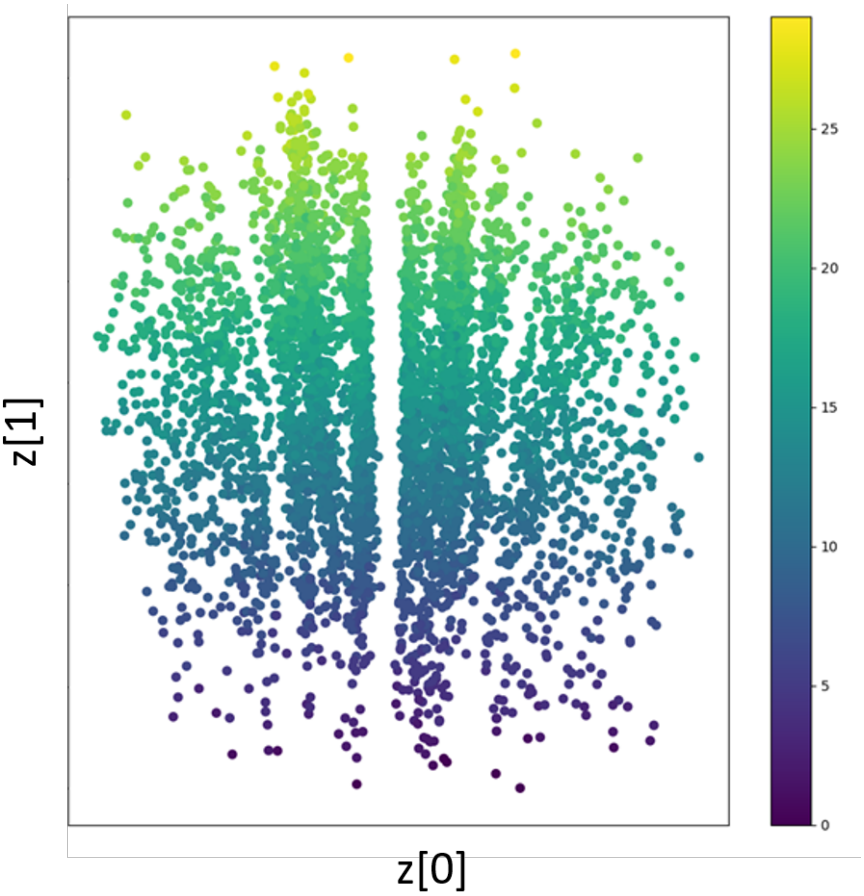

Figure 3.9: *Plot of the training data encoded in the latent space of Regressor VAE. The y-axis is the disentangled axis representing the complexity feature c.*

Layer	Number of Neurons	Activation Function Used
Input (flat)	60	-
Dense	512	ReLU
Dropout	512	-
Dense	256	ReLU
Dropout	256	-
Dense	64	ReLU
Dense (z mean)	2	Linear
Dense (z log var)	2	Linear
Lambda (z)	2	-
Dense	64	ReLU
Dense	256	ReLU
Dense	512	ReLU
Dense	60	ReLU

Table 3.4: *RVAE architecture: VAE*

Layer	Number of Neurons	Activation Function Used
Input	1	-
Dense (r mean)	1	Linear
Dense (r log var)	1	Linear
Lambda (r)	2	-
Lambda (pz mean)	2	Linear

Table 3.5: *RVAE architecture: Regressor*

4
Experimental Setup and Evaluation

In this chapter we discuss the structure of the dataset used in our models. Then, we evaluate the two implemented networks in reconstructing the input and then in generating new data. We evaluate the reconstruction by calculating the accuracy of the reconstructed chord sequences on the test data. Subsequently, we evaluate the generation of novel chord sequences given a desired harmonic complexity by human ratings in a listening test. Next, we describe the process of chord progression generation in the two models and the transition from symbolic to audio representation. We present the web-app developed to collect the user ratings. Finally, we describe the experiment we conducted and we analyze the results obtained from the analysis of human ratings.

4.1 Experimental Setup

In this section we describe the dataset used in our study. Then, we discuss the cleaning algorithm we implemented, in order to eliminate the blurring effect of the models' outputs. This problem is caused by the dalla data-dimension reduction of Variational Autoencoder (VAE). By applying the cleaning algorithm on the outputs we obtain a binary matrix. Finally, we evaluate our two network models in the reconstruction of input data, in particular, the chords and the individual notes.

4.1.1 Dataset

Our dataset is generated using a data-driven language model of cognitive harmonic expectations. Di Giorgi et al. [13] propose a model that expresses the probability of a chord sequence as a descriptor of harmonic

complexity, whose correlation is strongly confirmed by the results obtained through listening tests. The network implemented in [13] is a compound model composed of three different architectures: prediction by partial matching, hidden Markov model and deep recurrent neural networks. This model was trained on the Ultimateguitar.com dataset, composed by half a million annotated chord sequences. These data were first preprocessed by Di Giorgi, reducing the number of possible chords from 373 to 4. In particular, for each of the 12 class pitches the possible chord types are *maj*, *min*, 7 and (5).The first 3 types of chords were chosen because they were the ones most present in the annotations of the chord sequences, while the following was selected (5) because the mapping to either of the three previous types is not possible without any prior information about the tonality.

The data-driven language model implemented by in [13] is able to obtain sequences of chords estimating a probability value, which is related to the complexity perceived by people. The validity of these values was tested by performing perceptual tests. More information can be found in [13].

In our research we consider a dataset obtained from this model. The dataset has been already used by Foscarin [37] to analyze the correlation between the musical expertise of the listeners and the complexity of the chord sequences. It consists in 6311 sequences composed by 5 chords each, which are represented in Piano Roll format, as described in Sec. 3.1.1. Each sequence begins and ends with the same chord, which can be $Cmaj$ or $Cmin$. This limitation was chosen to reduce the chance that a chord progression would be perceived as random. We discussed the concept of random stimulus and perceived complexity in the 1.1.2 section. Each chord sequence is associated with a complexity value. These values are mapped to 30 bins. In this study we will refer to class,bin or value as synonyms when related to complexity. In Figure 4.1 are shown two histogram representing the number of chord progressions per complexity bin, divided by the major and minor progressions.

By analyzing the Tab. 4.1 it is possible to evaluate how the different bins reflect the complexity values using some rules of music theory. Analyzing, for example, the major sequences, we identify that progressions in the first bins (low complexity) contain harmonic transitions mainly between chords I-IV-V. This type of association is in agreement with Marvsik's study [32], where he identifies as "simple progressions" the ones formed by the transitions between these chords. Continuing in the subsequent bins, out-of-key chords are inserted, which increase the perceived complexity value.

4.1.2 Output cleaning

In generating chord progressions, both $\hat{\mathbf{X}}$ models produce "blurred" outputs due to the reduction of data dimensions in the latent space in VAE.

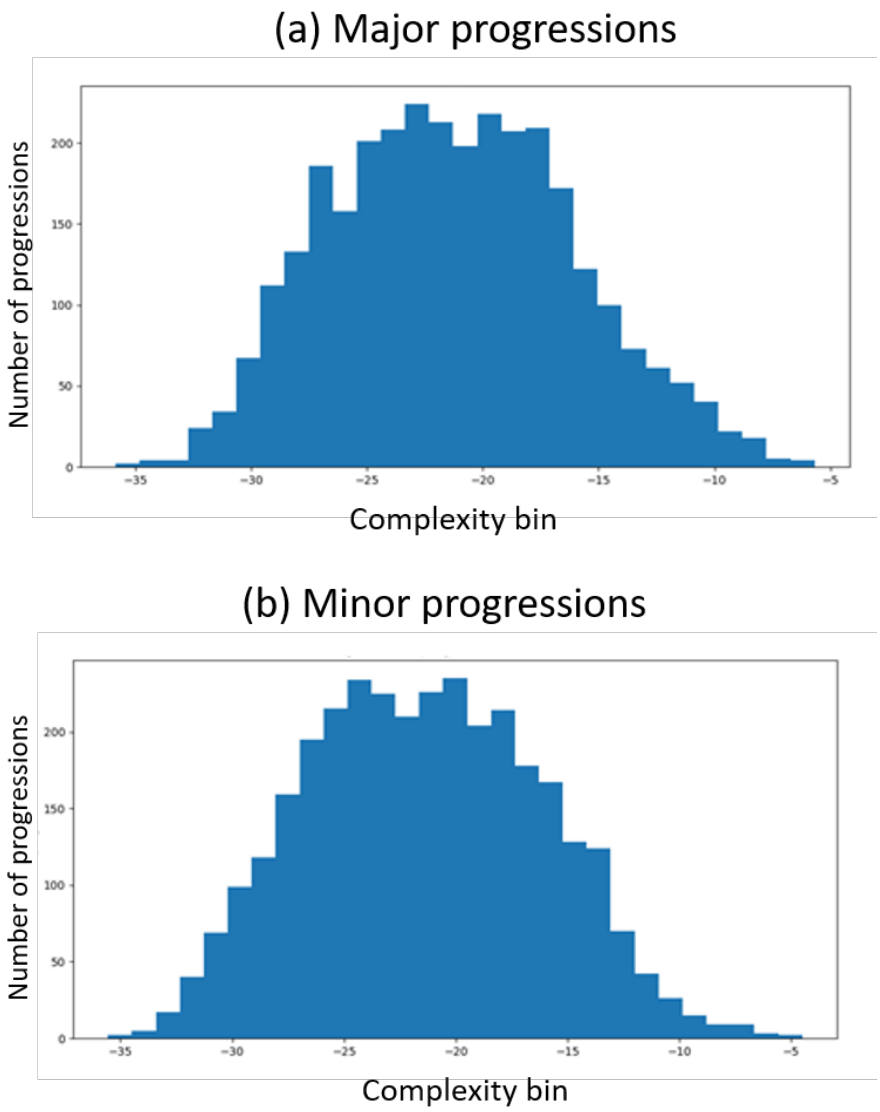

Figure 4.1: *The number of chord progressions in the dataset per complexity bin. The bins group the values of complexity in the dataset into 30 classes. In (a) the major sequences (b) the minor sequences. The first start and end with C maj, while the second with C min.*

Complexity Bin	Major Progressions	Minor Progressions
1	C G F G C C F C G C	Cm Fm A♯ D♯ Cm Cm F Cm F Cm
2	C G C G C C F Am G C	Cm G♯ Fm G Cm Cm A♯ G♯ Fm Cm
5	C E7 Am F C C G Dm Am C	Cm D♯ Cm Fm Cm Cm A♯ D♯ F Cm
10	C B A♯ A C C D♯m D♯ G7 C	Cm Dm G♯ Dm Cm Cm G7 Fm D♯ Cm
20	C A7 A♯7 D♯ C C G F♯7 A C	Cm C♯m G7 G♯ Cm Cm G♯7 A♯m D Cm

Table 4.1: *A subset of the dataset divided by different bins representing complexity.*

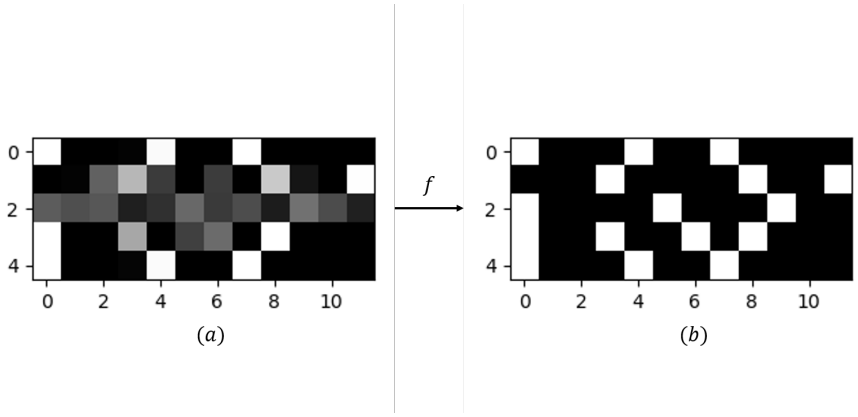

Figure 4.2: *(a) output of the models (b) "cleaned" version.*

We define an output as "blurred" when its composed of chords vectors whose values are between $\{0, 1\}$, instead of $[0, 1]$. In our study we want the outputs to be made of only binary values since each of the values of $\hat{\mathbf{X}}$ represents the notes of the chords in the sequence, and therefore the only values we consider are "on" and "off". Since we know what possible chords of a progression we can obtain as output in our model, we apply a "cleaning" algorithm.

For each chord $\hat{\mathbf{x}}$ belonging to a sequence generated by the $\hat{\mathbf{X}}$ models, we compute its cosine distance with respect to each of the 48 possible chords contained in a dictionary (12 pitch classes, for 4 types) in multi-hot vector format. We then select the chord in the dictionary that gave the best result and substitute it in place of the original model output, resulting in a $\hat{\mathbf{X}}$ progression consisting only of 1 and 0 values.

4.1.3 Evaluation of Encoder and Decoder

Here we evaluate the ability of the models to reconstruct the input chord sequences X, calculating the accuracy in the number of notes and chords

Model	Chord recognized	Notes recognized
CVAE	82.5%	80.77 %
RVAE	71%	70%

Table 4.2: *Reconstruction accuracy of the reconstructed input sequences in the two conditional models implemented. Is is evaluated the reconstruction of chords and single notes.*

	Mean Squared Error	R2 Variance
Regressor	3.002	87.9%

Table 4.3: *Regressor values in RVAE model*

correctly recognized. The comparison will be made between X and the outputs of the network $f(\hat{\mathbf{X}})$, where f is the cleaning function defined in the previous section. The architectures of the two models are described in Chapter 3. In both cases the dataset was divided into 70% training, 10% validation and 20% testing, balancing the division of the data for the different complexity classes. Both models use Adam optimization algorithm. In the Conditional Variational Autoencoder (CVAE) we reduce the mapping of 30 complexity classes to 5, in order to balance the number of examples for each bin. As previously shown in 4.1.1, the classes at the extremes values contain few example data. In Regressor Variational Autoencoder (RVAE) the 30 classes of the original dataset are instead preserved. This leads to better results in the regressor and consequently in the representation of the disentagled-dimension by the complexity in the latent space.

The results of chord and single note reconstruction of the two models are shown in Table 4.2. These results indicate that both models have learned a valid data distribution. They can encode the inputs in a lower-dimensional space and reconstruct them correctly. In particular, the CVAE exhibits significantly higher values in reconstructing chords and notes than the other model.

Furthermore, for the RVAE we evaluate the accuracy of the model calculating the mean squared error and R2 score. The results are shown in Table. 4.3. The R2 score is the coefficent of determination. Its value is very high, about 88%, which indicates a strong proportion of the variance in complexity that is predictable from our model. We also show in Figure 4.3 the plot of the predictions made by the regressor w.r.t the ground-truth values.

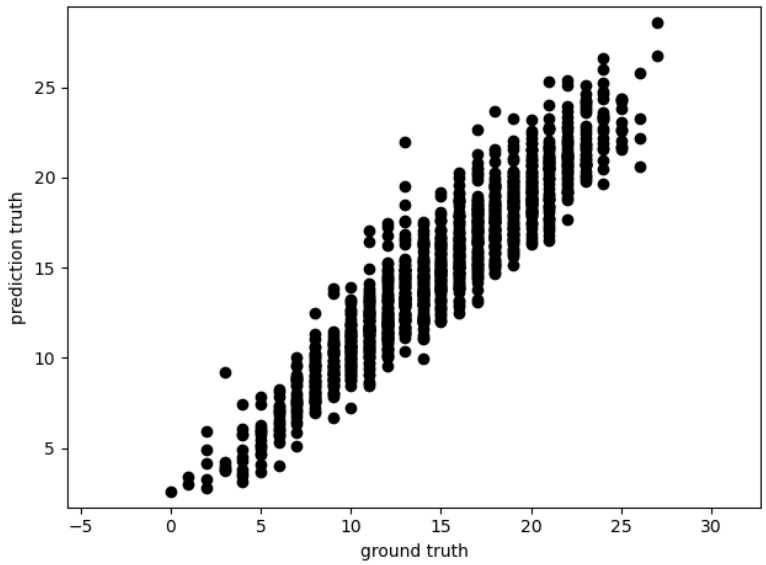

Figure 4.3: *Plot of the predicted complexity value by our model vs. ground-truth*

4.2 Evaluation of conditioned sequence generation

Following, we discuss the generating process of chord sequences for CVAE and RVAE. Then, we describe the conversion from symbolic to audio domain, discussing the choices made regarding tempo, instrumentation and chord voicing.

4.2.1 Chord Sequence Generation

For the perceptual test we generated 80 chords progressions, 40 per model, including 8 for each of the 5 possible complexity values. The 8 sequences are equally divided into major and minor sequences. As we described in 3.2, the generation process in CVAE was done by:

- Sample a random latent variable z from the prior distribution $p(z)$
- Concatenate the z variable with the $\mathbf{c}$ conditioning vector and generate a new data from $p_\theta(\mathbf{X}|\mathbf{c}, z)$

For the RVAE model, we can generate the chords sequences by moving along the disentagled dimension and give the latent variables as input of the decoder. To compute the latent variable associated with a complexity value in the disentagled dimension, we encode a sequence from the training set that has the desired complexity. From its coordinate along the disentagled dimension, we get the indication about the values of the latent variable to be used for generating chord sequences with similar or different complexity. Finally, in the choice of the samples to use in the listening test, we avoid to select the chord progressions generated by our

model that are equal to the ones in the training set used to train the two models.

4.2.2 Creation of audio excerpts

In order to perform the listening test, the generated chord sequences needed to be converted in audio format. In the transition from symbolic to audio domain we had to make decisions about tempo, instrument and chord voicing. We used the choices discussed in [13], aiming to restrict the subject's attention solely to the chord sequence. Specifically, regarding time, the chords change every 1.5 seconds, corresponding to 4 beats at 160 beats per minute (BPM) or 2 beats at 80 BPM. For the intrumentation and arrangement we use a classic grand piano. Starting from the MIDI format we used the sounds of a commercial grand piano sample library to render the chord progression in audio domain. Finally, we dealt with the chord voicing. This is a factor of great importance in the perception of a chord progression. Using chords only in their fundamentals makes progressions "artificial" and monotonous, due to the absence of a melodic interconnection between the chord notes. In order to obtained a more "musical" result, Di Giorgi et al. use a simple voice leading model. Given a progression, the model computes the list of possible voicings and their scores from each chord, based on music theory rules (e.g. basic counterpoint, voice leading rules). Then the optimal voicing sequence is obtained with Viterbi algorithm.

4.3 Web app

In this section we evaluate the ability of the models to generate new chord progressions by means of listening test. First we discuss the generation process in the two conditional models. Then, we describe the web-app we designed to collect ratings on the generated sequences. In the first part of the experiment, the participants are profiled based on their music background using the self-report questionnaire of the *Goldsmiths Musical Sophistication Index*. The second part is the perceptual test in which the participants were asked to express their level of agreement to the indicated complexity value provided for each chord progressions. The evaluation is expressed using the Likert scale scores from 0 to 4, where completely agree is the highest score and completely disagree the lowest one.

4.3.1 Gold MSI Test

To profile the musical skills and attitudes of the participants we use the Goldsmiths Musical Sophistication Index (Gold-MSI) self-report questionnaire. The test consists of 38 questions with seven-point scale answers for each question. The answers are then combined to form 5 sub-factors

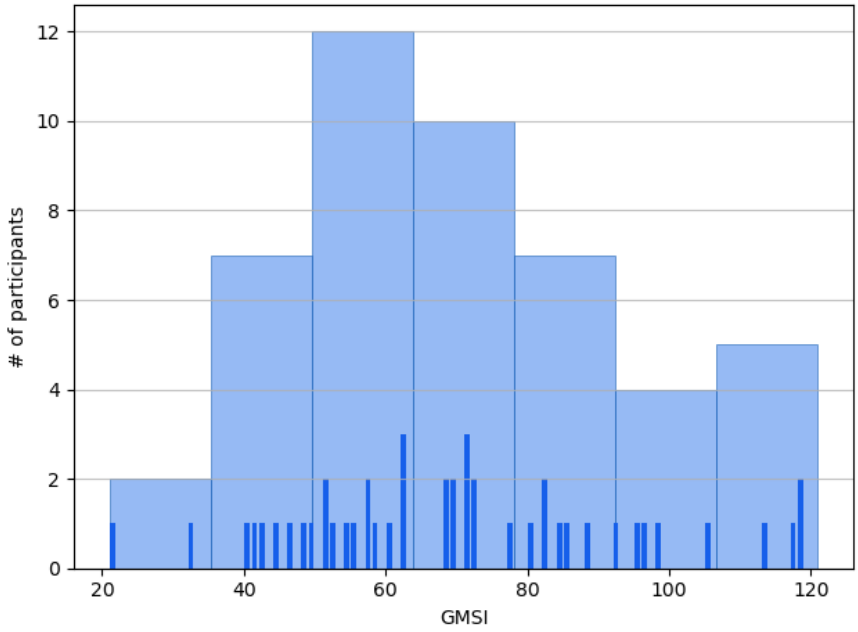

Figure 4.4: *The number of participants in the listening test grouped by musical expertise using the standard self-report questionnaire General Musical Sophistication Index (Gold MSI v1.0). High values of this measure indicate good musical skills and expertise of a user.*

(active engagement, perceptual abilities, musical training, singing abilities, emotions) and one general factor (general music sophistication).

In Figure 4.4 we show an histogram plot representing the GMSI of the participants.

4.3.2 Listening Test

In this section we describe the structure of the listening test. The web-app has been devoped using Flask, a web-framework written in Python and developed in AWS.

The test was composed by 80 questions, each consisting of a sequence of chords belonging to 5 possible complexity classes (1 to 5). The users, after having listening as many times as they wants each sequence, were asked to express a level of agreement on the complexity value proposed. This value was expressed using the Likert scale scores from 0 to 4, where the highest score (4) means completely agree, the lowest one (0) is completely disagree. If the rating of the user was less than 2, an extra question was shown. Asking the user to specify if the perceived complexity is greater than or smaller.

The initial two questions were used to familiarize with the user interface and are not recorded. The 80 sequences were initially shuffled, then shown using a Round Robin algorithm that sorts them by usage. The 20% of the proposed complexity values in the questionnaire are pur-

posely wrong. This was done by predicting that, on average over the 80 questions, a user would definitely have a tendency to not always express opinions of agreement.

The test was developed as a public web application. To avoid any friction and incentivate the user we have not added any form of authentication. Every user could perform the test in autonomy without a supervisor. This choices might introduce some noise in the results, since we do not have control on the behaviours of the users while conducting the test. In the following section we present some of the possible problems and we describe the choices made in order to minimize their occurrence and minimize their effect.

- **The participants are not able to understand the instructions of the test**. Before the start of the audio test we provided a precise description of the test structure and the questions that will have be asked to the user. The implementation of the GUI has been kept as simple and clean as possible, in order to make it very intuitive (Figure 4.6).

- **Participants do not understand the meaning of complexity classes**. Before starting the audio test we let the user listen to 6 chord sequences (3 per class) of complexity 1 and 5 (Figure 4.5). These sequences come from the training set of the dataset used by our models. This was done to provide some references of the two extremes of complexity values to the participant. Moreover, these audio references was available to be listened to during the test. This is chosen to prevent the user from assessing complexity only by looking for similarities among reference sequences instead of having a generic idea about complexity.

- **Users purposely fail the test**. By conducting the test online, it was not possible to supervise participants, who may behave unpredictably without any reason in particular. To avoid this problem we proposed the test mainly to the people interested in our research, such as university students, researchers and teachers of our degree course, in order to reduce any risk of receiving random tests.

- **User getting tired and starting to give answers randomly in order to complete the experiment**. We provided the possibility to send the results of the audio test after having done a minimum of 10 questions. This made the duration of the test variable (it lasts about 15-20min by answering all the questions) depending on the listener's willingness to continue or stop and send its ratings.

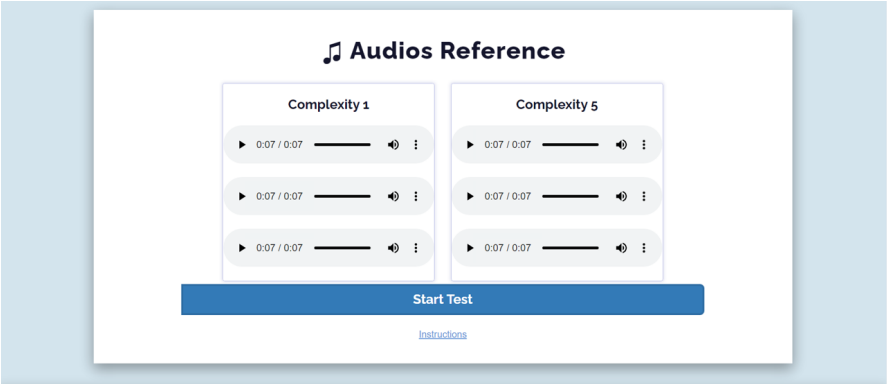

Figure 4.5: *The audio reference page of the web-app. Here are present* 3 *examples for chord sequences belonging to complexity class* 1 *(lowest) and* 5 *(highest)*

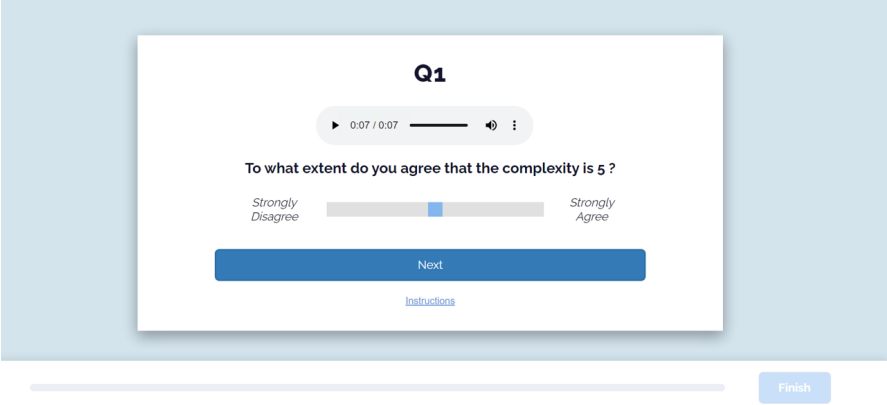

Figure 4.6: *The listening test of the experiment. The participants are asked to express their level of agreement to the indicated complexity value provided for each chord progressions. The slider is locked until the sequence has been completely played. Also the "next" button, to step to the following audio, is initially locked, until the rating is expressed with the slider. The instruction opens a pop-up with instructions for carrying out the test and two reference audios belonging to complexity classes* 1 *and* 5*.*

4.4 Results

In this section we discuss the data collected from the listening test in order to evaluate the two models in generating chord sequences using complexity as a conditioning parameter. Initially, we analyze the ratings made on the single samples, to detect possible non-valid results. Then, we evaluate the two models implemented by analyzing the users' ratings of the 80 sequences, 40 for each neural network. The listening test was taken by 47 participants and each chord sequence was evaluated about 30 times.

4.4.1 Data Cleaning

Initially, we analyzed participants' ratings on each of the 80 audio samples to identify possible invalid results. In particular, we focused on analyzing the ratings of disagreement with respect to the complexity value given in the questions. In the test, when a user expresses a value of disagreement with respect to the complexity estimated by our models, they needed to specify whether they believe the complexity was higher or lower than the one indicated.

Analyzing these values for the 80 chord progressions, some ambiguous samples were identified. We define ambiguous sample when is not present a clear consensus in the users that expressed disagreement whether the complexity given sample is higher or lower. To be more clear, in Figure 4.7 we provide two examples in which we analyse the answers of the users that disagre with the complexity value provided. In the first audio, about 90% of those users believe that the complexity shout have an higher value. We do not consider this case as ambiguous, since those users have a clear (90%) consensus of what should be the complexity according to them. In the second audio, about 48% of the users indicate that the complexity value should be minor, 52% instead higher. In this case we define the samples as ambiguous, because on average users express discordant opinions on the complexity of such audio and so there is not a clear consensus. This behavior happens because we asked the users to evaluate audio complexity, which as mentioned in the sections 1.1.2, is a perceptual parameter dependent on multiple factors.

The goal of our network models is to condition the generation of chord sequences according to the value of complexity, expressed as a "general" concept (and not specific to the subjectivities of each person). For this reason we identified in the dataset and excluded the audios that show in the users' disagreement ratings the ambiguity previously defined. To indicate a sample as dubious the delta between higher and lower must be less than 33%. By applying this criteria, 13 dubious sequences were identified and therefore cleaned up.

4.4.2 Data Analysis

We consider the ratings given by participants on the 67 samples using Likert scale values expressed from 0 to 4, shown in Figure 4.8. The data shows that more than 61% of the evaluations agreed with the complexity values expressed by our model, 31.8% agreed and 29.43% completely agreed. Approximately 9% are neutral ratings, indicated by the middle value on the scale. The remaining data is broken down into 24.6% disagreement and only 5.23% complete disagreement. The percentage of disagreement is "relatively high". However, this result is predictable since in this test we are evaluating the judgment of complexity, which as we discussed earlier, is not an absolute value. Therefore in the evaluation of the samples in the test we expected a significant presence of this value. From the results, however, is highlighted the clear difference between the values of disagreement and completely disagreement, in fact the latter much lower than the others. These data provide a positive rating of our models in generating chord sequences using complexity as a parameter.

We proceed to analyse the same data by splitting the ratings on the audios according to the model from which they were generated. We describe in Tab. 4.4 this evaluations dividing the sequences generated by the model of RVAE and CVAE. Analyzing the data, the first model shows slightly higher results than the second model, obtaining about 64% (32.57% agree and 31.20% completely agree) of positive evaluations compared to 59.3% (31.32% agree and 28% completely agree) of the CVAE.

We proceed in analyzing participants' ratings with respect to the complexity classes of the sequences generated by the two models. We represent the results in Tab. 4.5 for the CVAE and in tab. 4.6 for the RVAE. The results show that the RVAE model performs better in chord sequences with low complexity (1 to 3), in particular it reaches about 67.5% agreement in the class of 3 versus 20.1% disagreement (of which only 3.3% completely disagree). In contrast, the CVAE model performs better for high complexities (4 and 5). In particular for the highest complexity it obtains more than 78.6% agreement of the users.

Finally, we analyzed a possible correlation between a user's musical expertise expressed through the Gold-MSI questionnaire and the ratings expressed. We used the Pearson correlation and the Spearman correlation evaluating the musical level of a user in relation to the agreement values expressed in the perceptual test. We tested the correlation between the Gold-MSI and the evaluation of the samples in relation to the different complexity classes, and between the ratings given to the single sequences.

We want to evaluate if users with similar value of musical knowledge express similar judgments on the samples. However, the results show that this correlation is not present. In fact in the ratings of the audio, users with close values of Gold-MSI expressed discordant opinions (examples in Figure 4.9). This result is interesting and consistent with the discussions about the concept of complexity made in Section 1.1.2. In-

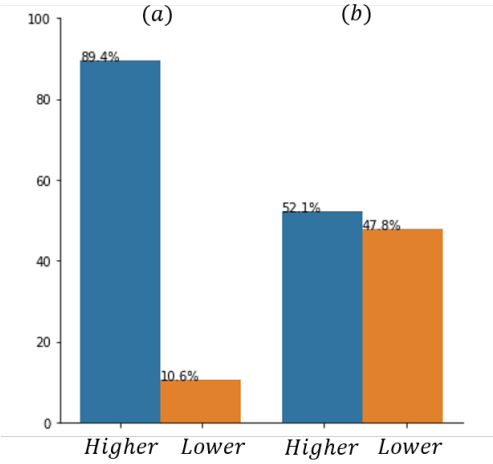

Figure 4.7: *Analysis of participants' negative ratings: we evaluate, among users who disagree, how many believe that complexity has a higher or lower value than the one we indicated in the test. In the first example (a) the choice is clear, in fact about the 90% believe that the complexity is higher. In the second (b), about 52% indicate that it has an higher value, 48% instead assign a lower complexity. The case described in (b) is defined as "doubtful" sample.*

User Ratings	CVAE	RVAE	Average
0	5.89 %	4.41 %	5.23 %
1	25.79 %	23.13 %	24.60 %
2	8.96 %	8.67 %	8.83 %
3	31.32 %	**32.57 %**	31.88 %
4	28.00 %	**31.20 %**	29.43 %

Table 4.4: *Ratings of the participants on the samples generated by the CVAE and RVAE models. The values are expressed using 5-value Likert scale, from 0 to 4. The percentages indicate the average number of user ratings for a possible value out of the total, where 0 means Completely Disagree and 4 is Completely Agree.*

deed, a subject's evaluation of this perceptual feature depends on several factors, such as familiarity, and thus is not solely related to a subject's musical knowledge.

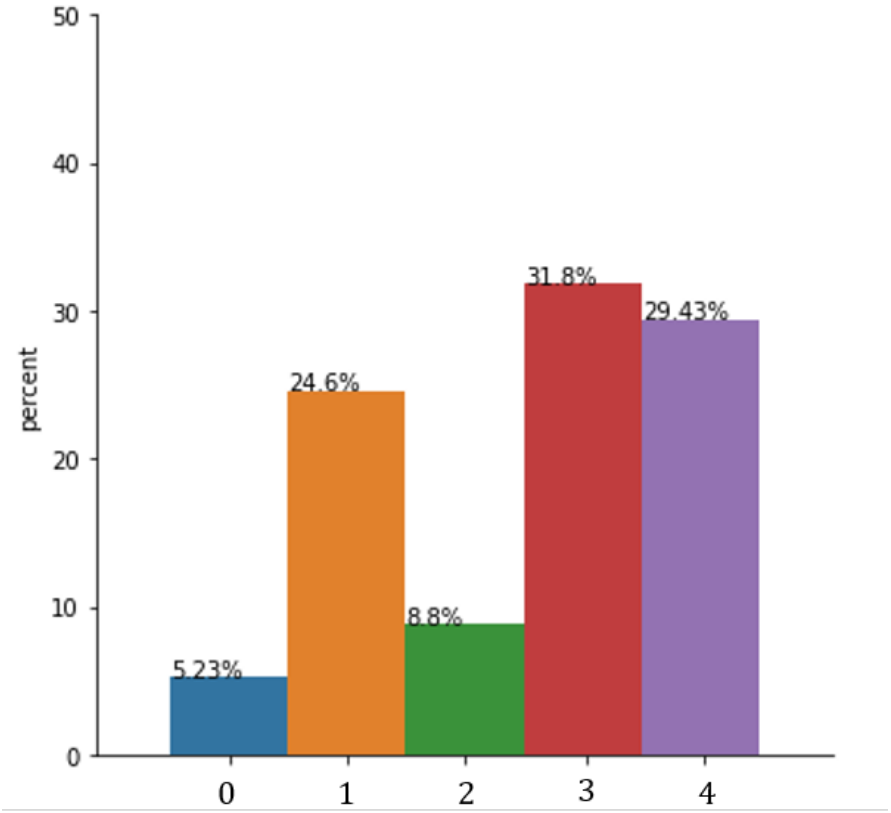

Figure 4.8: *Histogram of the Likert scale values expressed by the participants to evaluate the level of agreement with the complexity value we indicated in the listening test.*

CVAE

	Samples complexity				
User Ratings	1	2	3	4	5
0	9.60 %	7.18 %	4.49 %	3.10 %	4.58 %
1	26.55 %	35.32 %	32.02 %	19.87 %	11.45 %
2	9.04 %	7.79 %	12.36 %	9.32 %	5.34 %
3	20.34 %	31.14 %	30.34 %	40.37 %	36.65 %
4	34.47 %	18.57 %	20.79 %	27.34 %	41.98 %

Table 4.5: *Ratings of the participants on the 40 samples generated by the CVAE w.r.t. their complexity values. The percentages indicate the average number of ratings for each value of the Likert scale out of the total for each of the complexity classes.*

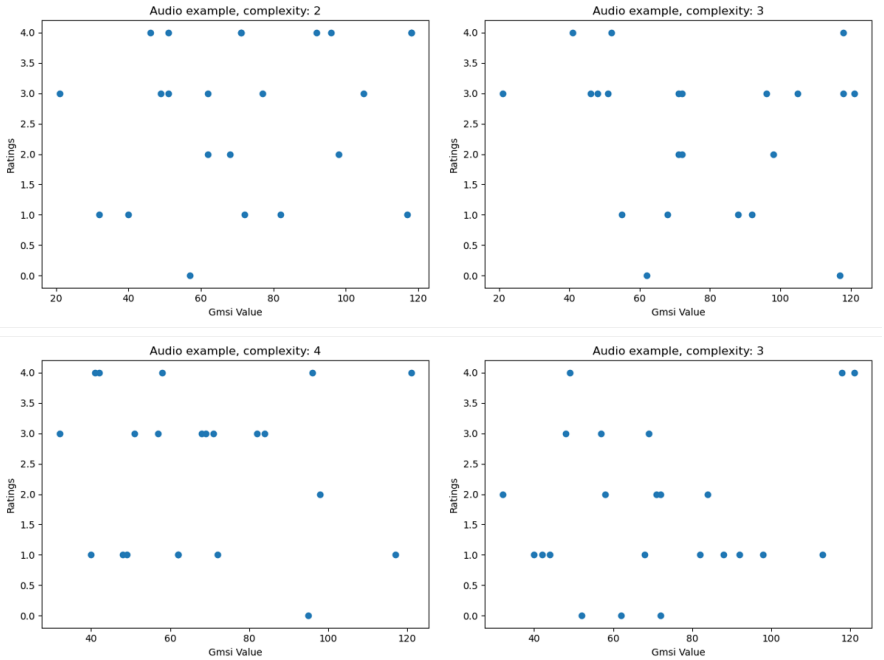

Figure 4.9: *Plots of the the users' ratings (y-axis) in relation to their Gold-MSI value (x-axis) for audio samples used in the test. No clear correlation has been highlighted between this 2 values.*

RVAE

	Samples complexity				
User Ratings	1	2	3	4	5
0	5.74 %	6.55 %	3.37 %	3.0 %	0.90 %
1	24.71 %	24.05 %	16.85 %	26.0 %	21.62 %
2	5.75 %	9.29 %	12.35 %	10.0 %	8.10 %
3	25.86 %	32.78 %	42.70 %	39.0 %	28.84 %
4	37.94 %	27.33 %	24.73 %	22.0 %	40.54 %

Table 4.6: *Ratings of the participants on the 40 samples generated by the RVAE w.r.t. their complexity values. The percentages indicate the average number of ratings for each value of the Likert scale out of the total for each of the complexity classes.*

5
Conclusions and Future Works

In this thesis we addressed the issue of automatic music generation of chord sequences, using complexity as a conditional parameter. We implemented and trained two Conditional Architectures based on the standard Variational Autoencoder model using an annotated dataset with chord sequences and complexity value.

Then we evaluated the models in two tasks: input reconstruction by using the data in the test set and the generation of new chord sequences by using complexity as a parameter. This second assessment is done through human ratings. They have been generated 80 chords progressions, 40 per model, including eight for each of the 5 possible complexity values.

We developed a web-app to collect the user ratings on the generated sequences through a listening test. In this test, the users were asked to evaluate the chord sequences indicating their level of agreement on the complexity value provided in the questions, expressing their accordance using a 5-value Likert Scale.

The goal of the test was to assess whether the complexity values of the sequences generated by the models correspond with the perception of complexity by the listener. This is definitely a complicated task, due to the difficulties in evaluating high-level descriptors. In addition, the dataset used has limitations, such as the number of sequences and the type of chords composing them.

The results obtained from user ratings on the generated samples show high agreement values with the complexity values given by our models. In particular, Conditional Variational Autoencoder (CVAE) performs better for sequences with high complexity values, reaching the 78.6% of the users' approval (more than 40% are completely agree ratings). The Re-

gressor Variational Autoencoder (RVAE) performs better for sequences with low complexity, reaching the 67.5% of agreement. Considering the assumptions about the difficulty of our objective and evaluating the results obtained, we can define both networks capable of modeling complexity as a parameter in chord generation.

In addition, we investigated the lack of correlation between a user's musical knowledge, expressed via Gold-MSI value, and the assessment on the complexity of the samples. This indicates that the complexity perceived by two people with similar levels of musical expertise can often be different, demonstrating that this depends on a combination of multiple factors, including subjective parameters.

5.1 Future Work

We obtained interesting results concerning the use of harmonic complexity as a generative parameter. This is, however, only one of the many facets of complexity in music. In future work we would like to explore other types of complexity (e.g. melodic, rhythmic) to manipulate the generative process. Another interesting scenario is how to combine harmonic complexity with other complexity facets e.g. harmony-rhythms, harmony-melody. This would open up new scenarios in the world of composition, allowing the creation of music to be conditioned according to a high-level feature closely related to human perception.

Bibliography

[1] B. Edmonds, "Complexity and scientific modelling," *Foundations of science*, vol. 5, no. 3, pp. 379–390, 2000.

[2] A. Chmiel and E. Schubert, "Back to the inverted-u for music preference: A review of the literature," *Psychology of Music*, vol. 45, p. 030573561769750, 04 2017.

[3] J.-P. Briot, G. Hadjeres, and F. Pachet, *Deep learning techniques for music generation*. Springer, 2020.

[4] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[5] Q. Zhao, E. Adeli, N. Honnorat, T. Leng, and K. Pohl, *Variational AutoEncoder For Regression: Application to Brain Aging Analysis*, vol. 11765, pp. 823–831. 10 2019.

[6] O. Celma, P. Herrera, and X. Serra, "Bridging the music semantic gap," 01 2005.

[7] A. Aljanaki and M. Soleymani, "A data-driven approach to mid-level perceptual musical feature modeling," *arXiv preprint arXiv:1806.04903*, 2018.

[8] T. Parmer and Y.-Y. Ahn, "Evolution of the informational complexity of contemporary western music," in *ISMIR*, 2019.

[9] Y. Güçlütürk, R. H. Jacobs, and R. v. Lier, "Liking versus complexity: Decomposing the inverted u-curve," *Frontiers in Human Neuroscience*, vol. 10, p. 112, 2016.

[10] E. Smit, F. Dobrowohl, N. Schaal, A. Milne, and S. Herff, "Perceived emotions of harmonic cadences," *Music Science*, vol. 3, p. 205920432093863, 07 2020.

[11] N. Meade, N. Barreyre, S. Lowe, and S. Oore, "Exploring conditioning for generative music systems with human-interpretable controls," in *ICCC*, 2019.

[12] J.-P. Briot and F. Pachet, "Music Generation by Deep Learning - Challenges and Directions." ArXiv, Dec. 2017.

[13] B. Di Giorgi, S. Dixon, M. Zanoni, and A. Sarti, "A data-driven model of tonal chord sequence complexity," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 11, pp. 2237–2250, 2017.

[14] J. S. Downie, "The scientific evaluation of music information retrieval systems: Foundations and future," *Computer Music Journal*, vol. 28, no. 2, pp. 12–23, 2004.

[15] Y. Kim, E. M. Schmidt, R. Migneco, B. G. Morton, P. Richardson, J. Scott, J. A. Speck, and D. Turnbull, "State of the art report: Music emotion recognition: A state of the art review," in *ISMIR*, 2010.

[16] S. Oramas, F. Barbieri, O. Nieto, and X. Serra, "Multimodal deep learning for music genre classification," *Trans. Int. Soc. Music. Inf. Retr.*, vol. 1, pp. 4–21, 2018.

[17] M. Schedl, "Deep learning in music recommendation systems," *Frontiers in Applied Mathematics and Statistics*, vol. 5, p. 44, 2019.

[18] O. Celma, P. Herrera, and X. Serra, "Bridging the music semantic gap," 01 2005.

[19] T. Eerola, A. Friberg, and R. Bresin, "Emotional expression in music: Contribution, linearity, and additivity of primary musical cues," *Frontiers in psychology*, vol. 4, p. 487, 07 2013.

[20] P. Juslin and E. LINDSTRÖM, "Musical expression of emotions: Modelling listeners' judgements of composed and performed features," *Music Analysis*, vol. 29, pp. 334 – 364, 10 2011.

[21] A. Friberg, E. Schoonderwaldt, A. Hedblad, M. Fabiani, and A. Elowsson, "Using listener-based perceptual features as intermediate representations in music information retrieval," *The Journal of the Acoustical Society of America*, vol. 60, pp. 1951–1963, 10 2014.

[22] J. L. Casti, "Chapter 2 - complexity and aesthetics: Is good art "complex" art?," in *Art and Complexity* (J. Casti and A. Karlqvist, eds.), pp. 21–29, Amsterdam: JAI, 2003.

[23] Y. Güçlütürk, R. H. A. H. Jacobs, and R. v. Lier, "Liking versus complexity: Decomposing the inverted u-curve," *Frontiers in Human Neuroscience*, vol. 10, p. 112, 2016.

[24] M. M. Marin and H. Leder, "Examining complexity across domains: relating subjective and objective measures of affective environmental scenes, paintings and music," *PloS one*, vol. 8, no. 8, p. e72412, 2013.

[25] M. M. Marin, A. Lampatz, M. Wandl, and H. Leder, "Berlyne revisited: Evidence for the multifaceted nature of hedonic tone in the appreciation of paintings and music," *Frontiers in human neuroscience*, vol. 10, p. 536, 2016.

[26] S. Streich, "Music complexity: a multi-faceted description of audio content," 2007.

[27] P. Grassberger, "Toward a quantitative theory of self-generated complexity," *International Journal of Theoretical Physics*, vol. 25, pp. 907–938, 01 1986.

[28] J. Klamut, R. Kutner, and Z. R. Struzik, "Towards a universal measure of complexity," *Entropy*, vol. 22, no. 8, p. 866, 2020.

[29] S. Kauffman, "The origins of order: Self-organization and selection in evolution," *emergence.org*, vol. 15, 08 1992.

[30] P. Vuust and M. A. Witek, "Rhythmic complexity and predictive coding: a novel approach to modeling rhythm and meter perception in music," *Frontiers in psychology*, vol. 5, p. 1111, 2014.

[31] T. Eerola, "Expectancy-violation and information-theoretic models of melodic complexity," *Empirical Musicology Review*, vol. 11, no. 1, pp. 2–17, 2016.

[32] L. Maršík, "Music harmony analysis: towards a harmonic complexity of musical pieces," 2017.

[33] D. E. Berlyne, "Novelty, complexity, and hedonic value," *Perception & psychophysics*, vol. 8, no. 5, pp. 279–286, 1970.

[34] D. E. Berlyne, *Studies in the new experimental aesthetics: Steps toward an objective psychology of aesthetic appreciation.* Hemisphere, 1974.

[35] C. R. Madan, J. Bayer, M. Gamer, T. B. Lonsdorf, and T. Sommer, "Visual complexity and affect: ratings reflect more than meets the eye," *Frontiers in psychology*, vol. 8, p. 2368, 2018.

[36] Y. Güçlütürk and R. van Lier, "Decomposing complexity preferences for music," *Frontiers in Psychology*, vol. 10, p. 674, 2019.

[37] F. Foscarin, "Chord sequences: Evaluating the effect of complexity on preference," 2017.

[38] A. Cohen, "D. temperley, the cognition of basic musical structures," *Psychology of Music*, vol. 32, pp. 105–117, 2004.

[39] N. Steinbeis, S. Koelsch, and J. Sloboda, "The role of harmonic expectancy violations in musical emotions: Evidence from subjective, physiological, and neural responses," *Journal of cognitive neuroscience*, vol. 18, pp. 1380–93, 09 2006.

[40] S. Leino, E. Brattico, M. Tervaniemi, and P. Vuust, "Representation of harmony rules in the human brain: Further evidence from event-related potentials," *Brain research*, vol. 1142, pp. 169–77, 05 2007.

[41] N. Steinbeis, S. Koelsch, and J. A. Sloboda, "The role of harmonic expectancy violations in musical emotions: Evidence from subjective, physiological, and neural responses," *Journal of cognitive neuroscience*, vol. 18, no. 8, pp. 1380–1393, 2006.

[42] F. Lerdahl, "Tonal pitch space," *Music Perception*, vol. 5, pp. 315–350, 01 1988.

[43] F. Lerdahl, *Tonal Pitch Space*, vol. 29. 01 2001.

[44] R. Fiebrink and B. Caramiaux, "The machine learning algorithm as creative musical tool," 11 2016.

[45] T. Winograd, *Linguistics and the Computer Analysis of Tonal Harmony*, p. 113–153. Cambridge, MA, USA: MIT Press, 1993.

[46] R. M. Keller and D. R. Morrison, "A grammatical approach to automatic improvisation," in *Proceedings, Fourth Sound and Music Conference, Lefkada, Greece, July."Most of the soloists at Birdland had to wait for Parker's next record in order to find out what to play next. What will they do now*, Citeseer, 2007.

[47] C. Bell, "Algorithmic music composition using dynamic markov chains and genetic algorithms," *Journal of Computing Sciences in Colleges*, vol. 27, pp. 99–107, 12 2011.

[48] A. oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," 09 2016.

[49] M. Mozer, "Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing," *Connection Science - CONNECTION*, vol. 6, pp. 247–280, 01 1994.

[50] D. Eck and J. Schmidhuber, "A first look at music composition using lstm recurrent neural networks," 2002.

[51] G. Brunner, Y. Wang, R. Wattenhofer, and J. Wiesendanger, "Jambot: Music theory aware chord based generation of polyphonic music with lstms," in *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 519–526, 2017.

[52] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *International Conference on Machine Learning*, pp. 4364–4373, PMLR, 2018.

[53] D. Makris, M. Kaliakatsos-Papakostas, I. Karydis, and K. L. Kermanidis, "Combining lstm and feed forward neural networks for conditional rhythm composition," in *International conference on engineering applications of neural networks*, pp. 570–582, Springer, 2017.

[54] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer, "Midi-vae: Modeling dynamics and instrumentation of music with applications to style transfer," *arXiv preprint arXiv:1809.07600*, 2018.

[55] A. Klapuri, *Introduction to Music Transcription*, pp. 3–20. 01 2006.

[56] K. Gurney, *An introduction to neural networks*. CRC press, 1997.

[57] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[58] I. Ulusoy and C. Bishop, "Comparison of generative and discriminative techniques for object detection and classification," vol. 4170, pp. 173–195, 01 2006.

[59] A. Jordan *et al.*, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," *Advances in neural information processing systems*, vol. 14, no. 2002, p. 841, 2002.

[60] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[61] S. Jørgensen, "Model selection and multimodel inference: A practical information—theoretic approach, second edition, kenneth p. brunham, david r. anderson, springer-verlag, heidelberg, 2002, 490 pages, hardbound, 31 illustrations," *Ecological Modelling*, vol. 172, p. 96–97, 02 2004.

[62] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems*, vol. 28, pp. 3483–3491, 2015.

[63] X. Yan, J. Yang, K. Sohn, and H. Lee, "Attribute2image: Conditional image generation from visual attributes," 12 2015.

[64] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.