



UNIVERSITY OF PADOVA

DEPARTMENT OF MATHEMATICS "TULLIO LEVI-CIVITA"

MASTER THESIS IN DATA SCIENCE

VARIATIONAL AUTOENCODERS AND THEIR USE FOR SOUND GENERATION

SUPERVISOR

PROFESSOR SERGIO CANAZZA
UNIVERSITY OF PADOVA

CO-SUPERVISOR

PROFESSOR ANTONIO RODÀ
UNIVERSITY OF PADOVA

MASTER CANDIDATE

CHIARA DE LUCA

ACADEMIC YEAR

2022-2023

TO MY DEAR COUSIN LIVIA,
WITH A MIND FULL OF KNOWLEDGE
AND THE BEAUTIFUL HEART
OF ONE WHO HAS TRULY KNOWN
HOW TO DEEPLY LOVE MUSIC.

Abstract

This thesis explores the use of Variational Autoencoders (VAEs) in the field of sound generation, with a particular focus on timbral diversity and the infinite possibilities of sound transformation. Sound generation is approached from two distinct angles: harmonic sounds and non-harmonic soundscapes. Several prior research studies have already demonstrated the ability of AutoEncoders to capture the primary features of a sound, creating a latent space that preserves these features and can subsequently generate similar sounds, characterized by a shared timbral quality or musical intent. This thesis will, therefore, scrutinize this sound generation system, conducting multiple experiments with mel-spectrograms as input.

Furthermore, the latent space of the models will be extensively explored, capable of mapping the characteristics of sound into a space from which it is then possible to easily manipulate timbres and sound changes, leading to the generation of smooth sound morphing.

A questionnaire was administered to some participants to assess crucial aspects of the generated sound, such as sound quality, sound classification, and the smoothness of the generated sound morphings. The results were very promising, indicating a good level of sound generation and a certain fluidity in sound transformation, both for harmonic and non-harmonic sounds.

This research has natural practical applications in the field of sound design and the creation of background music generation systems. With strong prospects for sound manipulation and exploration, the approach presented is a promising blend of deep learning and musical knowledge.

Contents

ABSTRACT	v
LIST OF FIGURES	ix
LIST OF TABLES	xi
LISTING OF ACRONYMS	xiii
1 INTRODUCTION	1
1.1 Music Generation	3
1.2 DL for sound generation	5
1.2.1 Autoencoder	6
1.2.2 Restricted Boltzmann Machine	6
1.2.3 Recurrent Neural Network	7
1.2.4 Generative Adversarial Network	8
1.3 Limitations and Future Trends	10
1.4 Related Works	11
2 DATA	13
2.1 Audio Representation for a DL approach	13
2.1.1 Waveform	13
2.1.2 Spectrogram	14
2.1.3 Mel-Spectrogram	15
2.1.4 Other representations	16
2.2 Dataset	16
2.2.1 Non-Harmonic Sounds: FoodSound Dataset	17
2.2.2 Harmonic Sounds: Good Sounds Dataset	18
2.2.3 Data Pre Processing	19
3 VARIATIONAL AUTOENCODER	23
3.1 Preliminaries: Autoencoder Model	23
3.2 Variational Autoencoder	24
3.2.1 Conditional Variational Autoencoder	28
3.2.2 Latent Space and Sound Variability	28
3.2.3 Advantages and Disadvantages	30

4	SOUND GENERATION BY MEANS OF CVAE	31
4.1	CVAE application	32
4.1.1	Non-Harmonic Sounds	33
4.1.2	Harmonic Sounds	36
4.1.3	Without conditioning: a convolutional VAE approach	37
4.2	Generation of sound morphing	39
5	EXPERIMENTAL EVALUATION	53
5.1	How to combine sounds	53
5.2	How to improve smoothness and variability	55
5.2.1	General Considerations	57
6	FINAL RESULTS	63
6.1	Survey	63
6.1.1	Participants	64
6.1.2	Structure	64
6.2	Results	66
6.2.1	Quality	66
6.2.2	Classification	69
6.2.3	Smoothness	72
7	CONCLUSION	75
7.1	Discussion	75
7.2	Future Directions	77
	REFERENCES	79
	ACKNOWLEDGMENTS	85

Listing of figures

2.1	Fast Fourier Transformation mechanism	14
2.2	Mel scale versus Hertz scale plot	15
2.3	Harmonic - non Harmonic comparison	17
2.4	Non Harmonic Sounds - Waveforms, Spectrograms and Mel-Spectrograms	20
2.5	Harmonic Sounds - Waveforms, Spectrograms and Mel-Spectrograms	21
3.1	AutoEncoders - General structure	24
3.2	Variational AutoEncoder - General structure	26
3.3	Reparameterization trick mechanism.	27
3.4	Regularized Latent Space behavior	29
3.5	T-SNE Algorithm.	30
4.1	Loss Plot - Non Harmonic	34
4.2	Spectrum: simple VS complex sound	36
4.3	Loss Plot - Harmonic	37
4.4	Conditional VAE Generation - CELLO Sound	38
4.5	Griffin Lim Algorithm	39
4.6	Sound Generation - Simpler non-Harmonic Sound	40
4.7	Sound generation - More complex non-harmonic Sound	40
4.8	Static sounds of the same class	41
4.9	Dynamic over time sounds of the same class	41
4.10	Original - Sound 0	42
4.11	Generated - Sound 0	42
4.12	Original - Sound 10	42
4.13	Generated - Sound 10	43
4.14	Original - Sound 16	43
4.15	Generated - Sound 16	43
4.16	Principal Component Analysis with 2 and 3 components	43
4.17	T-SNE for non-Harmonic sounds with 2 components	44
4.18	Sound Range for each musical instrument	45
4.19	Harmonic Generation - an example	46
4.20	Original VS Generated Instruments	46
4.21	Original VS Generated Sound	47
4.22	Original VS Generated Sound 2	47
4.23	Attack Issue in CVAE Generation	48

4.24	Latent Space Plots - 3D PCA and T-SNE	49
4.25	Woodwind Latent Space with AE	50
4.26	Mel-spectrograms distributed in Latent Space	51
5.1	Sound morphing process - pt.1	54
5.2	Preliminary procedures for sound morphing - fade in/fade out + overlap . . .	56
5.3	Perlin Noise Injection effect	57
5.4	Distance Matrix - Points in Latent Space	58
5.5	Sound Morphing according to distance, 15 -> 12 and 15 -> 3	59
5.6	Sound Morphing - Extreme distance points (12 -> 13 and 19 -> 20)	59
5.7	Sound Morphing Mechanism for Harmonic Sounds	60
5.8	Cello - Trumpet Sound Morphing	60
5.9	Cello - Violin Sound Morphing	61
5.10	Woodwind to String - Woodwind to Woodwind	61
6.1	Participants plots - Age, Gender and hearing problems	64
6.2	Sound Quality Evaluation	66
6.3	Verifying ANOVA assumptions: Shapiro-Wilk and Levene Tests	67
6.4	Kruskal-Wallis Test - Quality Scores	68
6.5	Dunn Test for significant differences - p-values table	68
6.6	Non Harmonic Sound Classification according to CVAE Latent Space	69
6.7	Harmonic Sound Morphing Classification Results	70
6.8	Chi Square - Contingency Tables for Sound Morphing items in the survey . .	72
6.9	Chi Square Test - Values and Interpretations	72
6.10	Smoothness Evaluation	73
6.11	Verifying ANOVA assumptions: Shapiro-Wilk and Levene Tests	73
6.12	Kruskal-Wallis Test - Smoothness Scores	73

Listing of tables

1.1	Sound generation: main characteristics	3
1.2	DL Music Generation Limits	10
4.1	Some Experimented Models	33
4.2	Harmonic Generation Evaluation	49
6.1	Survey aims and general structure	63
6.2	Harmonic Sound Morphing Results	71

Listing of acronyms

DL	Deep Learning
VAE	Variational AutoEncoder
CVAE	Conditional Variational AutoEncoder
LS	Latent Space
SM	Sound Morphing

1

Introduction

Music is generally and commonly defined as the science or art of ordering tones or sounds in succession, in combination, and in temporal relationships to produce a composition having unity and continuity [1]. However, music has been the subject of various attempts at definition, as it is something that everyone knows what it is but defining it is quite complex. Some, for example, emphasize the aspect of intentional sound or the contrast with noise, some focus on organized sound, some on aesthetic and pleasing sound[2]. One of the most interesting musicological perspectives on this matter is the Edgard Varèse approach[3], who argues that music is nothing more than organized sounds in space. This opens up a very interesting scenario for sound because it allows for the consideration of all types of manipulable sounds as music, removing the limitation to atonal or highly experimental music. This theory has, in turn, been the subject of various criticisms, especially directed towards the excessive abstraction of his viewpoint, which did not even consider the melodic or harmonic aspect of a sound.

Regardless of the interpretation of what is definable as music, it remains a fact that an integral part of what characterizes it, especially from an aesthetic point of view, is the presence of patterns and sound timbres, which are responsible for making music pleasing to the human ear. Music is created through imitation and creativity, using the art of musical composition, which encompasses the act of conceiving a piece of music, the art of creating music, and the final musical product [4].

The significant interest in computer-based composition began in the second half of the 20th century. Reviewing the history of computers and the programs that later led to the development of new possibilities and perspectives for computer music and electronic music, one must remember CSIRAC[5]. It originated in Australia in 1949 and was the world's first computer to reproduce digital music in 1951. Computer music further evolved in 1957 thanks to Max Mathews[6]. In that year, not only was the first computer music generator developed, functioning with the IBM 704, but also the MUSIC I programming language was created. It later evolved into MUSIC II the following year, allowing four-voice polyphony but requiring about an hour to generate one minute of music. From there, the history of computer music programming took off, especially thanks to significant inventions like the IBM 7090, GROOVE[7], and Fairlight CMI[8], remembered as one of the first digital workstations.

Much of computer music developed during that period, often starting from emerging research centers in computer music, which were engaged in discovering new languages and producing software and hardware for musical purposes. For example, during those years, the significant and pioneering contributions of CCRMA at Stanford University (founded by John Chowning and focused on the development of digital synthesizers and programming languages like Max/MSP), IRCAM (with contributions from Jean Claude Risset, emphasizing multidisciplinary and innovation), CNMAT at the University of California (known for its significant contributions in algorithmic composition and audio signal processing), CNUCE (with contributions from Pietro Grassi, specializing in sound manipulation techniques), and CSC at the University of Padua (founded by Debiasi and oriented towards developing systems for real-time synthesis and live electronic performance) should be remembered.

Additionally, it is worth mentioning groundbreaking musical works in this field, such as Lejaren Hiller's Illiac Suite[9] from 1956, one of the first compositions produced with the help of a computer (Illiac I), and various compositions by Max Mathews, such as Analog 1: Noise Study and Daisy Bell (with synthetization of human voices), both born in 1961. Also noteworthy are Experiments in Musical Intelligence (EMI) by David Cope and Analogiques A and B by Iannis Xenakis. In 2000, Koenig's Project 1 (PR1) marked a turning point as it utilized Markov chains for computer music generation.[10].

Since then, various computational techniques have been employed, including Generative Grammars, Cellular Automata, and Chaos Theory. The potential of generating music tracks

through algorithmic-mathematical control started to emerge. The development of these techniques eventually led to their integration with advanced Deep Learning methods, with pioneering examples like DeepBach [11] for the computational composition of Bach chorales. Neural networks have proven to be excellent tools for music generation, and in recent years, there has been a particular focus on developing increasingly effective Deep Learning techniques.

1.1 MUSIC GENERATION

Musical generation is a vast field that ranges from faithful sound reconstruction to computational creativity, using various algorithmic approaches and different types of sound input. What's even more crucial is the purpose of the sound one wishes to produce and the ways in which it will be suitably generated. The basic features of the musical generation process can be divided in the following manner [12]:

Characteristic	What does it mean	Example
Sound type	what do you want to produce?	melody, polyphony, timbre...
Sound destination	who is it intended for?	musician, user, new software...
Sound usage	how is the generated sound processed?	Music performance, audio file...
Sound mode	how is the sound generated?	Sheet music, audio file...
Sound style	What musical style is required?	classical style, polyphonic...

Table 1.1: Sound generation: main characteristics

Many of the generative characteristics of the sound are intrinsically chosen at the time of model development. Indeed, the style of the sound is intrinsically chosen by the input sounds used in the model's training, just as the mode or use of the sound is intrinsically linked to the type of output set by the model. It should also be remembered that the architecture of the model to be chosen is closely connected with the characteristics of the sound and with the generation objective. For instance, the generation of polyphonic pieces will require a multi-channel structure and more memory compared to what might be needed for the generation of ambient sounds.

The sound generation process undergoes several steps [13] common to all the different Deep Learning architectures that can be used:

- **Selection and collection of sound data.** This can take various forms; the essential thing is to derive a certain number of sounds from which the model can learn.

- **Choice of audio representation.** This choice is closely linked to the purpose of the generation and the chosen data (for instance, a chromatogram might be more suitable in cases of polyphony, and presumably ineffective in the case of inharmonic sounds).
- **Choice of the model configuration.** This step includes not only the conscious choice of the model most suited to our purpose but also the various trials for the choice of ideal parameters and hyperparameters.
- **Model training.** This can require more or less time, often linked to the complexity of the model and the choice of hyperparameters (example: the number of epochs).
- **Sound generation.** In essence, the step that deals with transforming what the model has learned into real sound.
- **Evaluation.** Sensory evaluation of the sound, namely the human ear's verification of how close the generated sound was to what was hoped for.

Another crucial issue in the field of music generation is the type of sound representation that will be provided to the model. This is divided into two broad categories: Audio and Symbolic [14]. This division can be intuitively seen as a categorization into continuous (audio) and discrete (symbolic) variables. For a detailed and comprehensive look at the audio representations used here, the reader is referred to the Audio Representation for a DL approach section in the following chapter. However, it is pertinent to examine the main characteristics of these two types of representations and their advantages and disadvantages:

Audio Representation: Direct encoding of sound waves without further conversions or transformations. The sound is processed as it is. Examples of audio representations are waveforms and spectrograms.

- **ADVANTAGES:**

1. Faithfully captures sound nuances and details
2. Quite consistent
3. Suitable for harmonic sounds and simple noises, ideal for ambient sounds and complex noises.

- **DISADVANTAGES:**

1. Requires more memory

2. Greater computational complexity
3. Less intuitive sound manipulation.

Symbolic Representation [15]: Translates sound into a set of symbols, typically MIDI sounds and musical notes.

- **ADVANTAGES:**

1. Easier sound manipulation and analysis
2. Reduces computational load

- **DISADVANTAGES:**

1. Captures fewer sound nuances.
2. Requires conversion before listening (possible information loss)
3. Not suitable/optimal for all sounds (e.g., ambient noises)

Basically, we can conclude that audio can provide greater fidelity, while symbolic offers enhanced interpretability and control.

1.2 DL FOR SOUND GENERATION

Deep Learning is a form of machine learning in which the computer can learn from experience. Specifically, through hierarchical operations, the machine is capable of learning complex concepts and then reworking and reconstructing them from simpler concepts[16]. A fundamental part of Deep Learning is Deep Neural Networks, which have evolved naturally from the Perceptron[17]. It is, in fact, the inefficiency of the Perceptron in classifying non-linearly separable domains that led to the creation of the neural networks as we know them and apply them today.

The history of neural networks is relatively recent (1980) but quite intense, experiencing continuous fluctuations in interest from the scientific community. Over time, increasingly complex neural architectures have been created, suitable for different types of data. What typically distinguishes a neural network, however, is a structure composed of three types of layers:

input, hidden, and output layers. Depending on the type of problem, these layers will take on different characteristics (for example, an output layer for a binary problem will be significantly different from that for a multi-class classification problem). What makes neural networks complex and powerful are undoubtedly the hidden layers. Of course, depending on the depth of the network, the model will be more complex and better at capturing important information from the data.

During the same period of development of the first neural networks, studies on sound generation through Deep Learning began. In the initial experiments, architectures that had already proven effective in similar fields, such as computer vision or NLP, were used. It should be noted that, depending on the type of data being analyzed, different architectures are appropriate. Music, in particular, is a type of data that can be treated differently depending on the representation chosen for it. Below is a list of the main Deep Learning architectures considered effective for music generation.

1.2.1 AUTOENCODER

An autoencoder is a type of neural network widely used in tasks such as classification and music generation, as well as in data dimensionality reduction. As the main subject of this thesis, the autoencoder, its functioning, and its limitations will be extensively discussed in the following chapter.

1.2.2 RESTRICTED BOLTZMANN MACHINE

A Restricted Boltzmann Machine (RBM) is a stochastic neural network capable of learning the probability distribution of the inputs it encounters. It owes its name to the Boltzmann distribution. Its operation is facilitated by the presence of two distinct layers, one visible and one hidden. While nodes within the same layer cannot be interconnected, every node in the visible layer is connected to every node in the hidden layer. The initiation of a stochastic process leads to the estimation of activation probabilities of the hidden layer, which will subsequently be used to estimate the activation probabilities of the visible layer. This process allows for the reconstruction of the original input. Subsequently, an algorithm named “contrastive divergence” is employed to adjust the weights between the nodes. With each iteration of this process, there will be progressively better learning from the data.

However, the process can be further summarized into two steps:

- Feedforward step: encodes the input into the hidden layer.
- Backward step: decodes the input. Specifically, during this operation, the aim is to re-generate the representation.

Each iteration corresponds to a different weight update: if the generated data is considered to be original, the connections remain unchanged; otherwise, the weights are updated. In practice, the RBM operates by adjusting the weights to enhance the representation capability of the generated data.

1.2.3 RECURRENT NEURAL NETWORK

Recurrent Neural Networks (RNNs) are widely used deep learning algorithms for sequential data. This makes them perfect for use in music generation[18], especially in the realm of symbolic music. In fact, music is essentially a language where dependencies between one sound and another are what create the coherence, linearity, and suggestiveness of a melody.

In practical terms, RNNs are feedforward neural networks with a recurrent component that captures sequences of information useful for calculating the new output. Like feedforward networks, recurrent networks consist of input layers, hidden layers, and output layers. However, while the feedforward network can be represented by the equation $X_t = Y_t$, in a recurrent network, the prediction equation[19] is:

$$(h_t, c_t) = f_n(h_{t-1}, x_t) \quad (1.1)$$

Where h_t is the new state, h_{t-1} is the previous state, x_t is the current input, and x_{t-1} is the previous input.

Their characteristic is, therefore, the ability to learn not only from current elements but also from previous ones. This is what makes these networks perfect for time sequences.

However, RNNs use Backpropagation Through Time (BPTT[20]) to estimate gradients. The underlying mechanism is the same as traditional backpropagation but specifically tailored for sequence data. The only difference is that in BPTT, errors are summed at each step. It is precisely because of this operation that RNNs face a significant training problem: gradient estimation difficulty. There are two main issues: vanishing gradients and exploding gradients[21]. Both problems are determined by the gradient's magnitude. The first problem occurs when the

gradient is too small, becoming so tiny during iterations that it vanishes. The second problem occurs when gradients are too large, leading to model instability and NaN values. Certainly, the problem can be mitigated by reducing the model's complexity, perhaps by reducing the number of hidden layers. However, the real trick to overcome this issue is to use a specific type of RNN, which has become increasingly popular lately: the Long Short-Term Memory (LSTM).

LSTM: OVERCOMING THE CHALLENGES OF RNN

The success of LSTM lies in learning how to manage its memory, removing it from the continuous flow of network operations. This is possible thanks to the addition of a key element of such a network: a cell gate. The cell is the foundation of LSTM, as it can regulate its memory. Specifically, it functions through these elements:

- Input Gate: chooses which information should be added to the cell's memory.
- Forget Gate: Chooses which information should be omitted and forgotten.
- Output Gate: selects the parts of memory contributing to the current output.

This, in brief, allows for the use of a recurrent network capable of capturing long sequences by capturing the right information.

1.2.4 GENERATIVE ADVERSARIAL NETWORK

The Generative Adversarial Attack (GAN) is a significant recent innovation (2014) in the field of music generation[22]. Their mechanism involves the simultaneous training of two different neural networks:

- **The Generator**: the generative model that transforms a noise sample into a sample resembling those taken from a distribution of real sound representation.
- **The Discriminator**: the discriminative model that calculates the probability that the generated sample comes from real data rather than from the generator.

Formula[23] is:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim P_{\text{Data}}} [\log D(x)] + \mathbb{E}_{z \sim P_z(z)} [\log(1 - D(G(z)))] \quad (1.2)$$

where $D(x)$ is the probability that x came from the real data, $D(G(x))$ is the probability that $G(x)$ came from the real data, $1 - D(G(x))$ is the probability that $G(x)$ didn't come from the real data. Moreover, $\mathbb{E}_{x \sim P_{\text{Data}}} [\log D(x)]$ represents the estimate of the probability that x is a real data point drawn from our set of real data. The algorithm aims to maximize this quantity, as it would mean that the algorithm is able to accurately recognize real data among all data points. And $\mathbb{E}_{z \sim P_z} [\log(1 - D(G(z)))]$ represents the expectation of the logarithm of $1 - D(G(z))$ with respect to $G(z)$ when G generates data from z in accordance with the probability distribution p_z . This term is used in the training of generative models like GANs to assess how effectively the generator G is deceiving the discriminator D . The objective is to maximize this expectation, which means that the generator is producing data that is difficult to distinguish from real data according to the discriminator.

In the case of music generation, the input typically used for the generator G is random noise. Currently, GANs are among the most advanced algorithms in terms of coherence, creativity, and the quality of the generated sound. Below is the presentation of their main advantages and disadvantages:

- **ADVANTAGES:**

1. High-quality sound generation.
2. Flexible generation, adaptable to all types of sound, music, style, and musical genre.
3. Versatile and open-ended generation: the created samples can be entirely different from the original ones while maintaining a certain similarity.

- **DISADVANTAGES:**

1. High computational complexity
2. Extended training time
3. Not immediately interpretable
4. Possibility of generating inconsistent sounds (if training is not optimal)
5. Tendency to overfit (and, consequently, to produce very static generation)

We can conclude that GANs, despite their many flaws and limitations, are so widely used because they are among the existing architectures with greater authenticity and quality in the generated sound.

1.3 LIMITATIONS AND FUTURE TRENDS

The sound generation capability is still very limited in many respects. The most significant limitation is undoubtedly the almost complete lack of creativity.

For the generation of good music, a certain level of song structuring is also required to make it linear and coherent for an engaging listening experience. The structure of a musical piece is what gives it a sense of direction in listening and is inherently connected to the genre or style of the piece itself. Going into specifics, highly competitive elements of a good piece include those related to the details of the composition, such as melodic passages, choice of key, specific alterations, and, in general, everything that contributes to the pleasantness and smoothness of a musical phrase. All of this requires a high degree of competitive reasoning, as well as continuous experimentation to find the best timbre or sequence of sounds. All these operations are extremely limited when using Deep Learning. They have also been categorized by Briot and Pachet[24] into four classes: control, structure, creativity, and interactivity. Below, the difference and limitation of Deep Learning compared to human composition[25] are outlined according to these classes.

Challenges	Human Composing[26]	DL Composing
Control	High. Quite natural depending on the theoretical and technical preparation of the composer.	Minimum. Generative algorithms operate as black boxes.
Structure	High. Fairly automatic depending on the composer's theoretical and cultural knowledge	Low. Requires additional constraints and specific data corpora for the intended objective.
Creativity	High. Depends on the composer's level of creativity. Sometimes it can be innate or extremely natural.	Minimal, very limited. The music created is only an intelligent regeneration of the input sounds.
Interactivity	Very high. The compositional process itself consists of continuous changes.	Low. The machine is automatic and does not involve interactivity.

Table 1.2: DL Music Generation Limits

Studies on music generation using Deep Learning are promising, but there is still a long way to go. The limitations discussed earlier can be seen as areas of scientific exploration that researchers will seek to investigate in the future. Pushing beyond the boundaries of limited control, structure, creativity, and interactivity in generated music will be a potential goal of future research. Other avenues of scientific exploration could involve combining different architec-

tures, which could lead to the creation of more dynamic and engaging musical material. It is also possible to explore how generation changes based on very diverse data, such as music of different styles or genres. Basically, the future of music generation is still open, and computer and sonic explorations are among the most diverse and promising[27]. In general, the future directions may lean more towards the development of generative algorithms to assist composers rather than as autonomous algorithms.

1.4 RELATED WORKS

Musical generation has been under scrutiny by the scientific community for several decades, although the use of Deep Learning in sound generation has been explored only in more recent decades. Studies such as [28] and [29] have provided a clear history of computer-based sound generation, while [30] has contributed to giving a general overview of AI usage in the field. Crucial for the development of more in-depth DL research are several synthesis articles, such as [13] and [31]. These general articles offer a perspective on the evolution of musical generation, ranging from the use of various models, such as VAE, GAN[32], or LSTM[33], which have been roughly outlined in articles like [34]. The most detailed and comprehensive study, covering essential aspects of musical generation from its inception, is [23].

Specifically, for the variational autoencoders, the focus of this thesis, various investigations have been conducted, both regarding their application in computer music [35] and in the analysis of sound timbres through latent space [36]. Finally, the future developments in the field of DL in musical composition, as well as its limitations and challenges, have been thoroughly outlined and described by Pachet in [24].

2

Data

2.1 AUDIO REPRESENTATION FOR A DL APPROACH

As mentioned in the previous chapter, when using a deep learning architecture, great care must be taken in how the audio is represented as input. Recalling that sounds can be represented through audio representation or symbolic representation, this paragraph will delve into the audio representations. For the project of this thesis, in fact, only audio representations were used. The reason for this choice lies in the very objective of the thesis, which is the timbral exploration of sound. In cases with more polyphonic or melodic purposes, a symbolic representation would certainly have been more appropriate. In this case, however, the sole audio representation allows preserving all the nuances of the sound, enabling its exploration with greater coherence and detail.

2.1.1 WAVEFORM

A waveform is the simplest and most direct audio representation, as it is a raw audio signal. The waveform is encoded using pulse code modulation, known as PCM. This allows for the generation of a continuous wave over time. Specifically, the x-axis represents time, while the y-axis represents the amplitude of the signal. The waveform is decoded as a sequence of numbers, where each number represents an amplitude sample at a given sampling frequency. The most common sampling frequencies are 22050 and 44100, which will respectively produce 22050

and 44100 samples (in the case of 1 second of audio playback).

Deep Learning architectures that allow the use of waveforms as input are somewhat limited and primarily encompass various end-to-end architectures.

2.1.2 SPECTROGRAM

In the description of the main audio representations, it is appropriate to start with the fundamental element of digital sound reproduction: the signal. It is nothing more than the variation in air pressure over time. Its measurement in samples (commonly 22050Hz or 44100Hz, but also) generates the so-called waveforms (already described previously). A waveform is what allows us to listen to a sound and verify its behavior. However, it captures only the amplitudes of the sound and is not suitable for all types of tasks. In fact, when dealing with neural networks, it is advisable to input much more substantial elements than a simple waveform. One possible solution is to decompose the signal in order to convert it from the time domain to the frequency domain. In this operation, the Fourier transform comes to our aid, capable of generating the signal's spectrum.

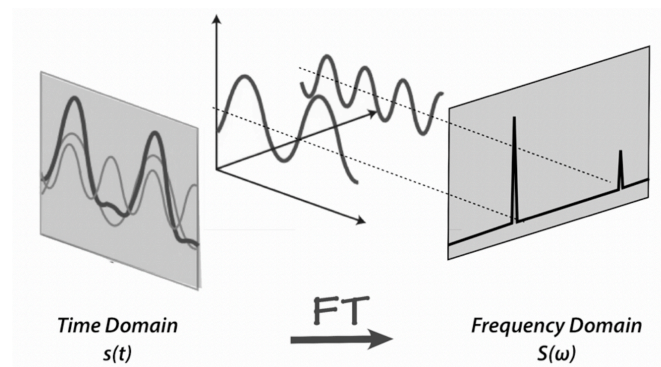


Figure 2.1: Fast Fourier Transformation mechanism

The transform is based on the Fourier theorem, which states that every signal can be decomposed into a set of sinusoidal or cosinusoidal waves that add up to the initial signal.

The algorithm that applies the Fourier transform is called Fast Fourier Transformation (FFT).

Completeness is achieved through the use of Short Time Fourier Transform (STFT). It represents the spectrum of signals as time varies. Essentially, it calculates different spectra by performing FFT on different windowed segments of the signal.

Stacking the generated FFTs on top of each other is what leads to the creation of a sound spectrogram. The x-axis represents time, and the y-axis represents frequency. It is also worth noting that the y-axis is converted to a logarithmic scale, and spectrograms have their own colors, which are used to indicate the intensity of sound (usually expressed in decibels).

2.1.3 MEL-SPECTROGRAM

The mel-spectrogram originates from a fundamental assumption: the human ear does not perceive frequencies linearly. It is here that the need arose to create a scale that represents the frequency content of sound as it is humanly perceived, rather than how it actually is. The scale created for this purpose is called the “mel” scale. It tracks the perception of sound by the human ear. Its shape reflects the fact that humans perceive differences in frequency of low-pitched sounds better than high-pitched sounds.

$$M(f) = 2595 \cdot \ln(1 + 700) \quad (2.1)$$

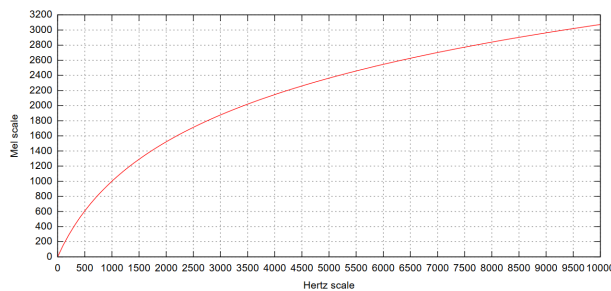


Figure 2.2: Mel scale versus Hertz scale plot

In conclusion, the mel-spectrogram is nothing more than a spectrogram in which frequencies are converted to the mel scale.

DATA PRE PROCESSING

For the creation of spectrograms consistent with our objective, the choice of certain parameters is important:

- **Hop length:** it determines the number of samples between consecutive frames. Its value is linked to the temporal resolution of the spectrogram.

- **Number of FFT (Fast Fourier Transform):** it is the number of points used for spectrogram computation. Its value is linked to the frequency resolution of the spectrogram.

For both parameters, common values include 128, 256, 512, 1024, and 2048. The choice depends on the trade-off sought between temporal and frequency resolution. It is advisable to avoid the number of FFT exceeding the hop length value. Specifically, it is preferable that $\text{hop length} \leq \text{num. FFT}/2$.

As final values, 1024 has been chosen as the number of FFT, and 256 as the hop length value for the spectrogram.

2.1.4 OTHER REPRESENTATIONS

There are several possible audio representations to choose from. We can list various types of acoustic features, which compress and contain certain sound aspects, as well as other variations of the spectrogram, such as chromatograms or MFCCs. The choice to use spectrograms and mel spectrograms exclusively in the experimental part was dictated both by their recognized effectiveness in similar tasks and by our initial data. Audio representations like waveforms or acoustic features would not have been an optimal choice for generating perceptually good sound or would have been limited and computationally expensive. Other representations derived from the spectrogram in the same way would not have been as suitable for the type of reference data. For example, MFCCs are better suited for speech data, while chromatograms are more suitable for data where melodic patterns play a significant role, whereas our data is audio-based, and even in harmonic sounds, the melodic component is limited, as there is at most one note for each audio clip.

Since the ultimate goal is to achieve good sound generation, it is reasonable to assume that the mel spectrogram is the winning choice, as it is the representation most capable of capturing the perceptible sound timbres to our ear and then generating similar ones.

2.2 DATASET

For this thesis, we have chosen to work on both harmonic and non-harmonic sounds.

- Harmonic Sounds: these are sounds whose spectral components follow a harmonic relationship with each other. Specifically, one can identify a fundamental frequency followed by a set of partials that are more or less evenly spaced, thus following a harmonic ratio. Examples of such sounds are musical instruments like the violin, piano, and flute,

where the sound is relatively clear, along with the perception of the fundamental frequency.

- Non-Harmonic Sounds: as the name suggests, these are sounds that do not relate to harmony. Their spectral components do not follow a harmonic frequency ratio, and their structure tends to be more complex. Non-harmonic sounds include dissonant sounds, noises, or even percussive sounds.

The following image illustrates this distinction clearly.

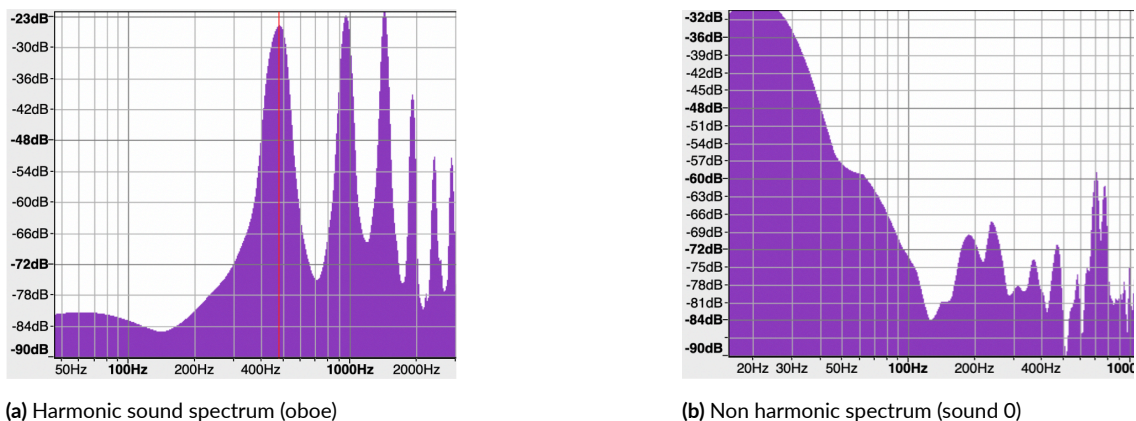


Figure 2.3: Harmonic - non Harmonic comparison

While in the sound of an oboe (a harmonic sound), there is a clear fundamental frequency around 500Hz (see the red line), with partials following a harmonic ratio spaced 500Hz apart, in the non-harmonic sound, there is no clear pattern, and the structure appears more complex.

2.2.1 NON-HARMONIC SOUNDS: FOODSOUND DATASET

The Food Sound dataset consists of 11,000 audio clips divided into 22 different types of sounds. The original dataset provides clips of varying duration (1, 2, or 4 seconds) while maintaining sound stability and coherence throughout the clip, meaning they do not modulate. All sounds are non-harmonic, characterized by a lack of clear structure, and are associated with specific sound effects, backgrounds, or noises. The sounds exhibit considerable diversity, with key sonic attributes including continuity and depth, contrasting with sizzling and synthetic sound types. The sounds are both continuous and exhibit variable pitch. The frequency varies, and there is a presence of glitchy and sci-fi-influenced elements. Below is a list of all types of sounds.

Sound	Type
0	Continuous glitchy elements with moving panorama
1	Continuous hum with pitch and frequency modulation
2	Continuous, deep and spatial, stable and rumbly
3	Continuous, deep, spatial and gloomy
4	Continuous, hollow, humming with ascending and descending pitch
5	Continuous, humming with regular low impulse
6	Continuous, rapid and glitchy elements with long pitch envelope
7	Continuous, rapid elements with slowly moving panorama
8	Continuous, rapid, glitchy buzzing elements
9	Continuous, rapid, sizzling elements
10	Continuous, scary sizzling cymbals with varying pitch
11	Continuous, soft noise. Alarm like effect
12	Continuously stridulating, buzzing and sizzling sound of processed crickets
13	Digital, continuous humming with varying pitch
14	Light, a tonal ring of shredded glassy elements like insect swarm
15	Sci-fi, science fiction, continuous, slightly sizzling elements
16	Sci-fi, science fiction, squishy humming
17	Spatial, continuous, glitchy element with modulation and panning movements
18	Spatial, deep, synthetic with continuous frequency modulation
19	Subtle, softly sizzling element with moving panorama
20	Synthetic sci-fi, science fiction with a sixth interval and slight moving panorama
21	Synthetic sci-fi, science fiction with glitchy, slowly varying pitch envelope and heavy low rumbling

2.2.2 HARMONIC SOUNDS: GOOD SOUNDS DATASET

The chosen dataset is GoodSounds [37]. It contains short recordings of variable duration (approximately 5 seconds per clip) of harmonic sounds. The sounds in the initial dataset belong to two different types: single notes and sustained notes (held) and scales. Only the first type of sounds was used, as our objective is the exploration of sound timbre, which requires the presence of sounds with clear and sustained timbre, without variations in time or frequency. Specifically, the sounds were performed by professional musicians using different microphone configurations (from one to four different microphones) and various recording devices. Each clip contains a sustained note from a musical instrument. The timbre of the instruments is

particularly sharp and clear, making the type of instrument easily recognizable even by non-experts. The chosen instruments are cello, violin, flute, clarinet, oboe, piccolo, and trumpet. These musical instruments share a sustained envelope. More percussive instruments such as piano or percussion would not be particularly suitable for continuous timbral exploration and sound morphing. Furthermore, it should be noted that the instruments are diverse, ranging from stringed instruments to wind instruments, and their musical ranges are also different. This will contribute to an even more interesting timbral analysis.

2.2.3 DATA PRE PROCESSING

For both reference datasets, the audio clips are in .wav format. Proper data preprocessing is crucial, especially in deep learning models, particularly in generative models, where the accuracy and coherence of the results are heavily linked to the quality of the audio used as input for model training. The inherent nature of the chosen datasets is highly advantageous, as all the audio clips are clear and free from noise. Operations such as normalization, filtering, feature extraction, equalization, and click reduction are common and widely employed in such applications. In this paragraph, I will present the main operations that have enabled the reference .wav files to become excellent training material for the model.

- **Audio Loading.** The audio should be loaded appropriately, accommodating its main characteristics such as duration, sample rate, and number of channels.
- **Duration Management.** When the audio is loaded, it will assume its own duration. However, it's important to note that when generating a uniform audio representation for all the samples used in training, it's essential to ensure the same duration for all sounds used as samples. This is closely related to the architectures that will be used, as having the same image shape for all spectrograms will allow the model to train. Therefore, a PADDING function will be applied to each audio representation to make all audio representations of the same duration.
- **Normalization.** Normalizing the values of an audio representation (which are notoriously different from each other) ensures uniformity and the absence of distortion while appropriately preserving all the nuances of the sound.

NON-HARMONIC SOUND

As a quick consideration of non-harmonic data, it becomes evident that they are rather diverse in nature. They are continuous over time, exhibit different frequencies (notice, for example, the low frequencies of sound number 3 compared to sound number 7), and have rather distinct connotative elements. This is evident from the very patterns in the spectrogram. Spectrograms with clear horizontal lines give rise to smoother sounds, while spectrograms with vertical lines give rise to sizzling or glitchy elements (see sound 17).

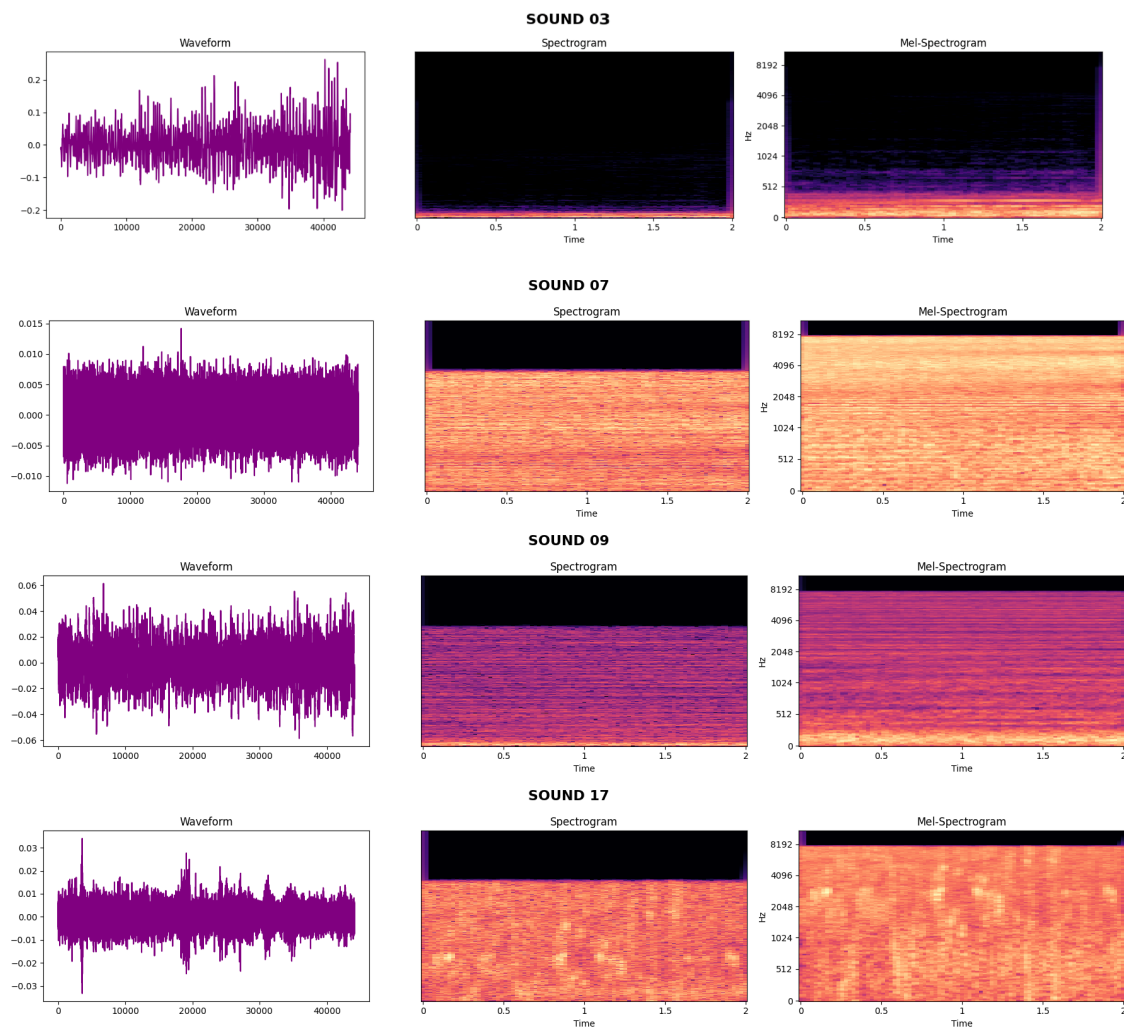


Figure 2.4: Non Harmonic Sounds - Waveforms, Spectrograms and Mel-Spectrograms

HARMONIC SOUND

As a quick assessment of harmonic data, it is evident that they possess an extremely recognizable timbre[38]. Within the various clips, the sound has a variable onset moment, typically within the first second of the clip. Furthermore, the sound exhibits a clear envelope and a certain pitch stability. Differences in instrument timbre can be discerned from the spectrogram, as we can clearly observe the relatively lower fundamental frequency of the cello compared to that of the piccolo. The onset frequency remains the same throughout the clip.

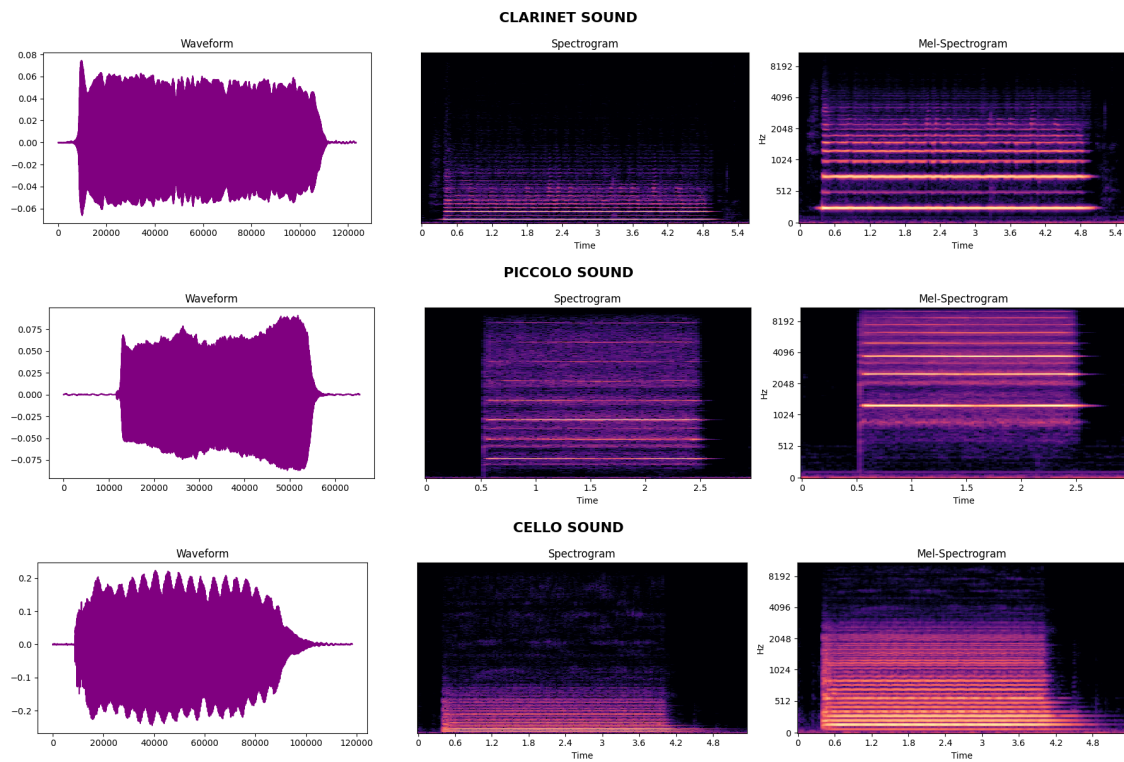


Figure 2.5: Harmonic Sounds - Waveforms, Spectrograms and Mel-Spectrograms

3

Variational AutoEncoder

3.1 PRELIMINARIES: AUTOENCODER MODEL

The autoencoder is a widely used and highly effective model for both classification tasks and the generation of new samples. It is a neural network consisting of several hidden layers, with the sole constraint of having the same number of nodes in input and output. This ensures its functionality, allowing, for example in music generation, to produce new samples more or less similar to those given as input. The strength of autoencoders lies in their ability to extract essential features from the input through a dimensionality reduction process. It falls under the supervised learning category. To delve deeper into its structure, it's essential to note that an autoencoder is composed of two main parts:

- **ENCODER:** the first part of the model's operation. It takes samples as input and then represents them in a latent space of arbitrary dimensionality using dimensionality reduction techniques.
- **DECODER:** takes the encoder's output (the latent space) and tries to reconstruct the original samples.

WHY AN AUTOENCODER ISN'T ENOUGH

The Autoencoder model has a tremendous capacity to capture fundamental details and represent them in a latent space, which is why this model is highly valued in scientific literature.

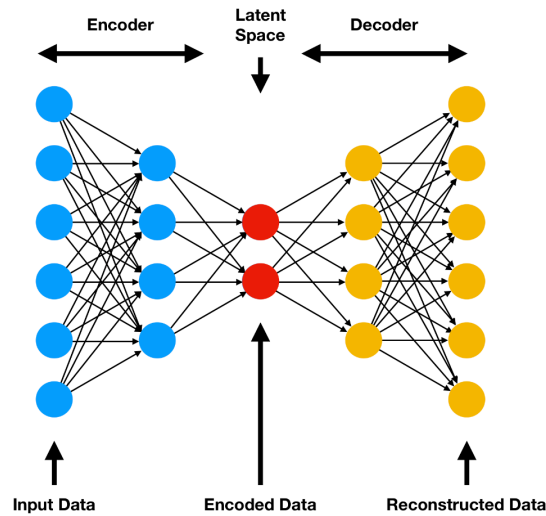


Figure 3.1: AutoEncoders - General structure

However, the model proves to be very effective in classification tasks and less so in generation tasks. The reason that makes this model less effective for sound generation tasks lies in the latent space. The standalone autoencoder, in fact, is not able to structure the latent space in a way that can generate consistent new samples. Without clear regularization, the points in the latent space may become nonsensical and incapable of congruent generation. Moreover, being a very unconstrained model, it tends towards overfitting, which further complicates the generation of suitable and diverse samples. The solution to the AE's challenges lies in the use of VAEs, which we will delve into in the next section.

3.2 VARIATIONAL AUTOENCODER

The Variational Autoencoder (VAE) is a type of deep learning model primarily focused on generation. It goes beyond creating a latent representation of data and decoding it; it aims to model the probabilistic distribution of the latent space. This allows transitioning from a deterministic model (autoencoder) to a probabilistic model (variational autoencoder).

Following the encoding process, the goal of VAE is to understand how the initial data (high-dimensional) can be coherently generated from the latent variables. The joint probability $p(x|z)p(z)$ enables this mechanism. $p(x|z)$ estimates how the data x is generated from the latent variables z , while $p(z)$ represents the latent variables.

The strength of VAE lies in the use of Variational Inference, which provides greater compu-

tational efficiency by drastically reducing the complexity of estimating $p(x)$ using traditional methods and sampling techniques. Variational Inference focuses on estimating an approximate distribution of $p(x)$ rather than the exact distribution. The approximate distribution is estimated from the family of distributions $q(z|x)$. This results in a distribution that provides a good approximation of the data while minimizing the complexity of manipulation and optimization operations.

Variational Inference is fundamentally based on two key concepts: the Kullback-Leibler Divergence (KLD) and the Evidence Lower Bound (ELBO).

The KLD measures the divergence between the approximate distribution $q(z|x)$ and the exact conditional distribution $p(z|x)$. In simple terms, it is essential for validating a good approximation of the data.

This leads to the formula:

$$q^*(z|x) = \arg \min \text{DKL}[q(z|x) \parallel p(z|x)] \quad (3.1)$$

At the end of this process, we can have confidence in a good latent representation of the data because the goal of finding the best approximate distribution $q(z|x)$ among the possible distributions in Q will have been achieved.

The goal to maximize, as an immediate consequence of estimating the KLD, is the ELBO, which aims to find the best approximation of the latent data distribution. In fact, the objective function of the VAE is [39]:

$$L(\theta, \phi) = \mathbb{E}_{q_\phi(z)} [\log p_\theta(x|z)] - \beta \text{DKL}(q_\phi(z|x) \parallel p_\theta(z)) \quad (3.2)$$

or, in other words, the difference between the likelihood and the KL.

It is possible to decompose the objective function into three essential blocks[40]:

- **Reconstructive block.** It focuses on a coherent reconstruction of the initial samples. In other words, it aims to minimize the reconstruction error.
- **Regularization block.** It deals with regularizing the approximation to q so that the approximate distribution does not deviate too much from the true one. In other words, it deals with the probabilistic estimation of the model.
- **The reconstruction term β .** The value of this parameter regulates the effect of the two blocks presented earlier. In particular, as β increases, greater emphasis will be given to regularization. Conversely, the model will favor the reconstruction aspect

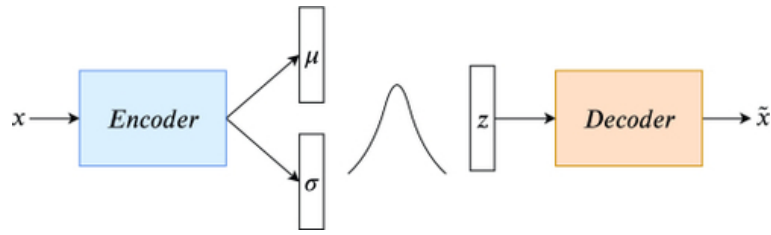


Figure 3.2: Variational AutoEncoder - General structure

So it's possible to intuitively visualize the structure and operation of a Variational Autoencoder (VAE) from the following image, where it's clear how the encoder, with the involvement of mean and variance values that allow regularization of the space, manages to create a bottleneck. This will serve as the starting point for the decoder to generate new observations.

REPARAMETRIZATION TRICK

The effective functioning of a neural network largely hinges on the backpropagation mechanism, which is crucial for optimizing the model's weights. This technique is rooted in the computation of the gradient of the loss function by retroactively propagating the error. This mechanism facilitates the updating of weights accordingly, typically through the employment of a gradient descent algorithm. Consequently, backpropagation enables the training of networks to be efficient, swift, potent, and to achieve optimal performance.

Variational Autoencoders also heavily rely on this method, but for its proficient implementation, it is imperative to employ the so-called reparametrization trick.

Indeed, the VAE harbors an intrinsic challenge: the estimation of the latent space necessitates sampling from a reference distribution, usually denoted as a multivariate normal. Such sampling gives rise to stochastic elements since extracting any sample from a probabilistic distribution entails a random process. However, the backpropagation algorithm cannot be directly executed through random nodes. It also involves the utilization of a gradient descent algorithm during sampling, which introduces a stochastic element, hence being non-differentiable.

To be precise, backpropagation pivots on computing gradients in deterministic operations, thereby necessitating exact gradients. Yet, sampling remains a random act and lacks defined gradients.

This would render the employment of backpropagation unfeasible.

This is where reparametrization steps in, with a mechanism that is inherently quite straightforward: it bifurcates the deterministic and the stochastic elements. Therefore, one wouldn't

sample from $N(m, \sigma^2)$, but would instead sample the error from the $N(0, 1)$ distribution, subsequently deriving the desired sample z .

The error component, epsilon, introduces randomness, transforming $\mu + \sigma\epsilon$ into a fully deterministic and differentiable entity[41].

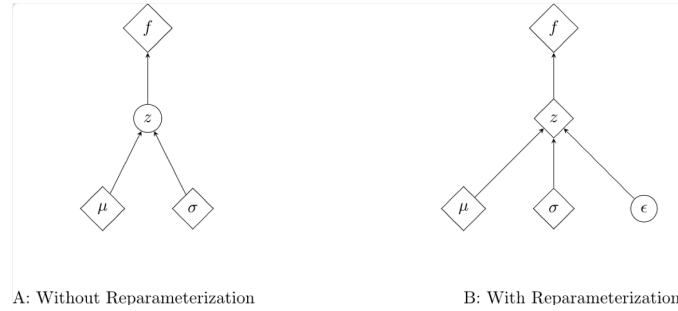


Figure 3.3: Reparameterization trick mechanism

In conclusion, reparameterization serves as the cornerstone that facilitates the use of back-propagation while preserving the model's stochastic essence.

LOSS FUNCTION

Loss functions are well-known components of a neural network that allow us to assess the model's performance during training. In fact, in supervised problems, the loss function is always closely related to the accuracy (or its variants) as it measures the effectiveness of a model in capturing information and making accurate predictions of output data. In our case, the goal is to minimize the loss, which means that in the training process, from input to output, the information loss is minimal. This leads to effective model optimization.

In this thesis, the loss function used will be the typical loss function of variational autoencoders, which is a combined loss composed of the sum of the reconstruction loss and the Kullback-Leibler Divergence (KLD).

$$\text{Combined Loss} = \text{RL weight} \times \text{RL} + \text{KLD} \quad (3.3)$$

The reconstruction loss measures the error between the input and the output reconstructed from the input. In the case of variational autoencoders, this measure is calculated using Mean Square Error or Binary Cross-Entropy, depending on the reference data. This loss aims to compress the input into a lower-dimensional space as efficiently as possible while ensuring consistency in regenerating the initial input. Formula is:

$$\text{Reconstruction Loss} = \frac{1}{N} \sum_{i=1}^N \left\| y_{\text{target}}^{(i)} - y_{\text{predicted}}^{(i)} \right\|_2^2 \quad (3.4)$$

The KLD is the function that transforms the latent space into a normally distributed space. This results in more agile data sampling and is suitable for generating coherent new data, as well as exploring and manipulating the latent space. Formula is:

$$\text{KL Divergence Loss} = -\frac{1}{2} \sum_{i=1}^N \left(1 + \log(\text{variance}^{(i)}) - \mu^{(i)2} - \exp(\text{variance}^{(i)}) \right) \quad (3.5)$$

The use of the combined loss leads to faithful input reconstruction, the ability to generate various similar timbres, and a sufficiently manipulable latent space.

3.2.1 CONDITIONAL VARIATIONAL AUTOENCODER

The Conditional Variational Autoencoder model (CVAE) originates as a variant of the VAE (Variational Autoencoder). It is characterized by the presence of an additional attribute that conditions the model's behavior. Similar to the VAE, the CVAE consists of both an inference network (encoder) and a generative network (decoder). Mathematically, the only difference lies in the estimation of $p(z|x)$.

A CVAE is a more complex model than a simple VAE, primarily because it requires a slightly more elaborate design (including the target variable y) and, most importantly, an appropriate dataset in which each observation is associated with a label y .

The significant advantage of a CVAE is its ability to incorporate additional information beyond what can be gleaned from the data alone. In this project, the CVAE will be employed because it offers a good potential for generating clean, high-quality sound that aligns well with the desired type of audio.

3.2.2 LATENT SPACE AND SOUND VARIABILITY

VAEs are designed to have a regularized latent space, aiming to lead to the generation of coherent samples. However, the latent space is not merely a passageway before generation; it is a remarkable mode of timbral exploration. The latent space allows us to manipulate certain sound characteristics and merge them[40]. This is because we are in a regularized space identified by

specific points, each representing a different sample. By projecting our latent space using some dimensionality reduction technique, it becomes visualizable, with closely related observations forming small clusters. If each point, based on its features, occupies a space in proximity or distance to other points, respectively closer or farther in terms of common characteristics, it means that we can thoroughly explore all timbral variations contained in our reference data.

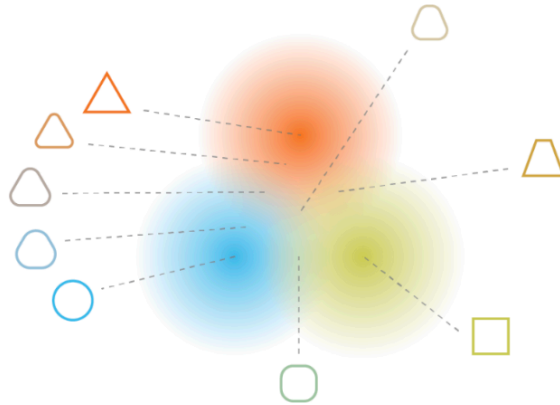


Figure 3.4: Regularized Latent Space behavior

In this study, we chose to use two dimensionality reduction techniques for space visualization: PCA and T-SNE.

PRINCIPAL COMPONENT ANALYSIS

Principal Component Analysis (PCA) is a linear dimensionality reduction technique. It starts with the variables that describe the original data. Using a covariance matrix, it estimates the relationships among the different variables. It then proceeds to calculate eigenvalues and eigenvectors, which allow the selection of principal components that capture the most data variance. In essence, it aims to choose dimensions that, despite reduced dimensionality, explain as much data variability as possible. At the end of the process, the reduced-dimensional data is ready to be projected and visualized, either in 2D or 3D, depending on the number of selected components.

T-DISTRIBUTED STOCHASTIC NEIGHBOR EMBEDDING

For the description of how T-SNE works, please refer to the algorithm outlined in the reference paper of this visualization technique for the latent space [42].

Algorithm 1: Simple version of t-Distributed Stochastic Neighbor Embedding.

Data: data set $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$,
cost function parameters: perplexity $Perp$,
optimization parameters: number of iterations T , learning rate η , momentum $\alpha(t)$.
Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, \dots, y_n\}$.

begin
 compute pairwise affinities p_{ji} with perplexity $Perp$ (using Equation 1)
 set $p_{ij} = \frac{p_{ji} + p_{ij}}{2n}$
 sample initial solution $\mathcal{Y}^{(0)} = \{y_1, y_2, \dots, y_n\}$ from $\mathcal{N}(0, 10^{-4}I)$
 for $t=1$ **to** T **do**
 compute low-dimensional affinities q_{ij} (using Equation 4)
 compute gradient $\frac{\delta C}{\delta \mathcal{Y}}$ (using Equation 5)
 set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t) (\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)})$
 end
end

Figure 3.5: T-SNE Algorithm

3.2.3 ADVANTAGES AND DISADVANTAGES

In general, the use of autoencoders and their variants [43] has been highly successful, as they are a very flexible type of model capable of capturing various types of information, both linear and nonlinear, from different types of data. They are also highly adaptable and customizable, providing access to a wide range of hyperparameters that can be chosen by the user. However, this is also a drawback of autoencoders, as they are indeed very sensitive to hyperparameters, requiring careful tuning by the programmer. Additionally, despite their flexibility, they can be quite computationally expensive.

Their use has proven to be highly effective in various types of tasks, including anomaly detection, feature extraction, sample generation, and denoising. Furthermore, their use is also quite effective in the field of music. Specifically in the context of generation, autoencoders are not only extensively studied in scientific literature [35] but also highly appreciated for their ability to generate sound that seamlessly combines coherence and variability of the desired sound.

4

Sound generation by means of CVAE

In this chapter, I will present the attempts at sound generation for the two reference datasets. Specifically, the objective is not limited to faithful sound generation from the input but also to the exploration of sound in its various aspects. In particular, harmonic and inharmonic sounds will be analyzed separately. It is expected that these two different types of sounds will be characterized by different aspects. For harmonic sounds, greater attention will be placed on the timbral resemblance of the represented instrument, while for non harmonic sounds, the ability to faithfully recreate the effects that are most present in the chosen sounds, such as the sense of depth or sizziness, will be the focus. A fundamental subject of study will also be sound exploration through latent space.

INPUT SOUND: A COMPROMISE BETWEEN DETAIL AND COST

In the preprocessing phase towards the mel spectrogram, as previously described in detail, the choice of the number of mel bands to represent the sound plays a fundamental role. This parameter affects two crucial aspects essential for good generation:

- Sound resolution, or how many sound details we choose to capture.
- Input shape, or how much computational effort the machine will need to train the data.

Specifically, the number of mel bands alters the length of the input image's axis. For example, a mel value of 64 results in a shape of 64 in one of the dimensions. This is why it is important

to choose this parameter carefully. For good sound generation, the sound must be more or less clear, sharp, and detailed, but at the same time, I do not want disproportionate computational complexity compared to the generative task's objective. The values used for parameters and hyperparameters for each dataset will be referenced in the following paragraphs.

CODE IMPLEMENTATION

The code implementation was carried out using various libraries, which can be divided into three categories:

- UTILITY: use of common Python libraries for data manipulation (numpy, pandas...) and visualization (matplotlib, sklearn...)
- DEEP LEARNING: I chose an approach with PyTorch, driven by the fact that it is a highly flexible and customizable library, highly appreciated by the scientific community, even in the musical field, for its computational efficiency.
- SOUND: for sound data processing (especially for input and output conversion processes), I preferred to use LIBROSA. Libraries like soundfile and sounddevice were also essential.

An overview of the code implementation can be found in the GitHub repository [44].

4.1 CVAE APPLICATION

A Conditional Variational Autoencoder (CVAE) has been chosen for the following reasons:

- The structure of the datasets used allows for it. In fact, both datasets used come with labels. For the Good Sounds dataset, the condition will be the musical instrument. For the Food Sounds dataset, the condition will be the type of sound.
- A CVAE can acquire more information about sound due to the presence of label y . This leads to:
 - Clearer and separable latent variables
 - Better manipulation of variables and, consequently, the generation of sound
 - When a specific sound is required, precise and accurate generation of that exact sound.

4.1.1 NON-HARMONIC SOUNDS

Non harmonic sounds are characterized by greater complexity. However, since the original dataset contains a wide variety of sounds, the models to be experimented with and the generated results may vary significantly. Several attempts were made, ranging from simpler to more complex models:

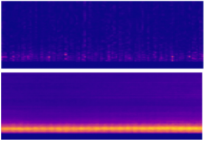

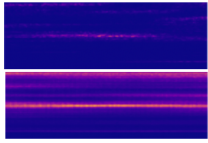
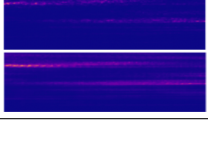


Name	Characteristic	Time per Epoch	Val Loss	Original VS Generated
Model 1	Three hidden layers (two in encoding, one in decoding) with 32 hidden units each. LS = 32.	6s	0.0010	
Model 2	Three hidden layers (two in encoding, one in decoding) with 256 and 128 hidden units. LS = 64.	5s	0.0010	
Model 3	Three hidden layers (two in encoding, one in decoding) with 256 and 128 hidden units. LS = 128.	5s	0.0010	
Model 4	Six hidden layers (three in encoding, three in decoding) with 1024 or 512 hidden units each.	20s	0.0008	
Model 5	Four Hidden Layers (two in encoding, two in decoding) with 256 hidden units each. LS = 256.	6s	0.0008	
Model 6	Four Hidden Layers (two in encoding, two in decoding) with 256 hidden units each. LS = 512.	9s	0.0010	

Table 4.1: Some Experimented Models

As for hyperparameters, various trials were conducted:

- Batch Size: 64, 128, 256
- Learning Rate: 0.0001, 0.0005, 0.001
- Optimizer: Adam
- Epochs: 50, 100, 250, 500, 1000

As activation functions, ReLU, TanH, and sigmoid have been experimented with. Specifically, sigmoid is used in the final layer of each model. In fact, the sigmoid activation function maps values from 0 to 1, which is perfectly compatible with the type of data I am dealing with, as the mel-spectrograms used as input are normalized between 0 and 1. Regarding the loss function used, please refer to subsection “Loss Function” in Section 3.2.

It’s important to note that as the number of epochs increases, the generated results do not deteriorate; rather, they become ineffective as the loss function stops decreasing around the 100th epoch. Below is a figure showing the trend.

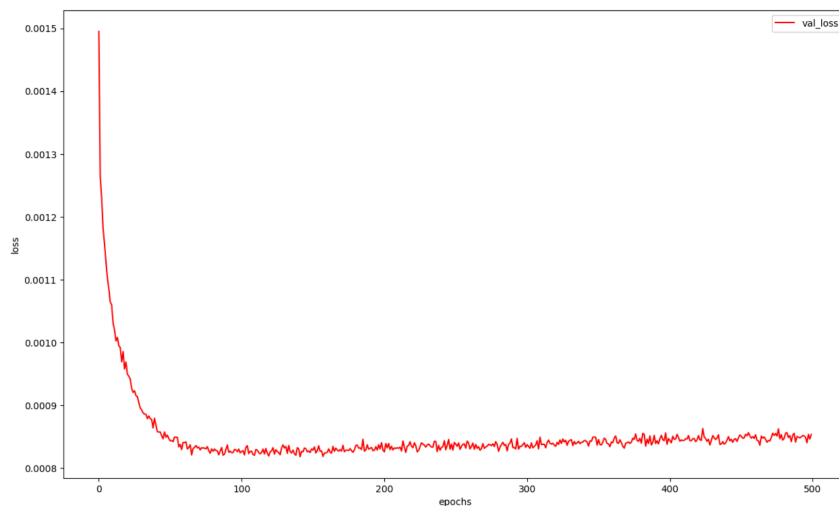


Figure 4.1: Loss Plot - Non Harmonic

As initial observations regarding the model’s performance, which will be further elaborated in the following paragraphs, it is possible to deduce that:

- Less is more: The use of fewer hidden layers resulted in greater similarity to the original sound and smoother output. Adding more hidden layers does not necessarily lead to improved performance, or at least not significantly. There is a stronger risk that the sound appears mechanical with more hidden layers.

- The number of hidden units is naturally linked to the model's complexity, and here again, less is more.
- The choice of activation function is crucial. Three types were tested: ReLU, Sigmoid, and TanH. Only ReLU is capable of achieving good results.
- The very low loss value is due to the type of image that is predominantly all blue with a few yellow lines to predict. It is important to consider this relatively (as a comparison between different models) rather than in absolute terms. The ultimate judgment of sound quality is, however, the perceptual evaluation by human listening.

At first glance at the spectrogram reproductions, the generation appears to be functioning, and the sound is similar to the original.

The most crucial factor for sound generation turned out to be the dimension of the latent space. While hyperparameters may provide only marginal improvements to generation, the latent space has the ability to completely alter the sound generation process.

SOUND COMPLEXITY AND LATENT SPACE DIMENSIONALITY: WHY THEY CAN CHANGE EVERYTHING

In general, it can be observed from the early generations that an excessively high-dimensional latent space generates sound that is overly mechanical, stuttering, and less evocative. On the other hand, an excessively low-dimensional latent space makes the sound unclear, intuitively because the latent space does not seem capable of capturing all the essential details of the audio clip. Consequently, the generated sound appears anonymous and devoid of its initial characteristics, while still preserving a certain level of listenability.

When dealing with non-harmonic sounds, the problem becomes more complex as the dataset comprises sounds with diverse characteristics. This implies that in the generation process, a significant distinction must be made between sounds in the dataset with a more complex and irregular structure compared to those that, although non-harmonic, exhibit a predominantly regular and orderly structure.

- Simpler sounds: continuous and deep [e.g., 3, 8]
- More complex sounds: sizzling, scratchy, or noise like background sounds [e.g., 2, 17]

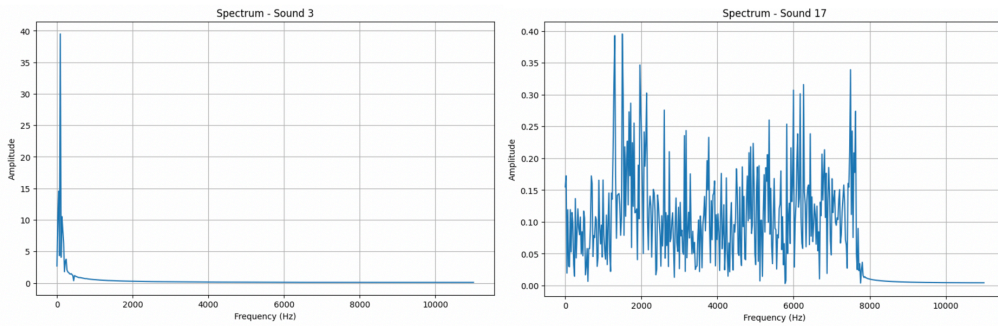


Figure 4.2: SPECTRUM: simple VS complex sound

The varying complexity of sound can be discerned not only in its audio representations, such as the mel spectrogram used as input but is also evident in the spectrum, as illustrated in the graph below.

Simpler models, such as those with fewer hidden units, prove to be very effective for continuous and deep sounds, perhaps because they possess very distinct characteristics that can be captured even by a simpler model. Their not entirely irregular structure allows for good compression in the latent space even with lower dimensions, such as 64 or 128. In contrast, more complex sounds, at the same latent space dimensionality, are generated in a bothersome and mechanical manner. Therefore, a high-dimensional latent space capable of capturing more information is required.

4.1.2 HARMONIC SOUNDS

For harmonic sounds, the same models were implemented, and experiments were conducted by varying all the parameters and hyperparameters used for non-harmonic sounds. Following a similar trial-and-error approach, good generation results were achieved. Here, the ease of training compared to non-harmonic sounds lies in the lower sound variety. In fact, as previously mentioned in the dataset description, what distinguishes the sounds is the timbre, not the sound type (presence of the timbre of a single musical instrument) or its nature (sustained sound, same frequency).

For this task, the models that were experimented with were generally the ones tested for non-harmonic sounds. In the case of harmonic sounds, a larger latent space was required. Among the models experimented with, we can mention the a model with three hidden layers (two in the encoder, one in the decoder), 512 hidden units, and a latent space of 256 as the best-performing

one, proved to be an excellent solution. With a time per epoch of 8 seconds and a loss of 0.0005, the model is simple and efficient, capable of capturing the main information. Unlike non-harmonic sounds, all the sounds benefit greatly from the same type of model, which is a direct consequence of the homogeneity of sound complexity.

All models were trained for many epochs, and below is the image of the loss plot for each epoch.

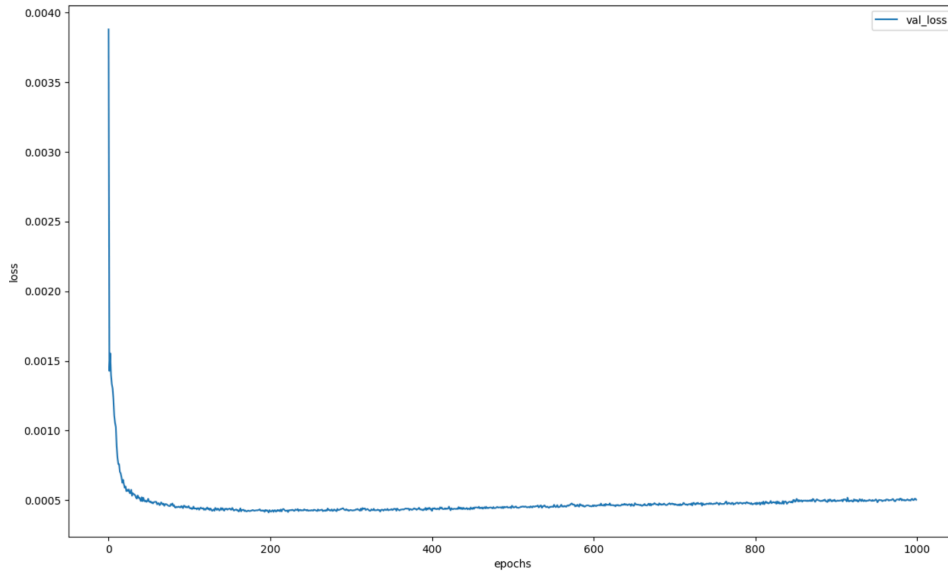


Figure 4.3: Loss Plot - Harmonic

It is evident that it behaves similarly to that for non-harmonic sounds, reaching a minimum around 100 epochs. Generally, the loss values are very close, fluctuating between 0.0005 and 0.0006. In terms of auditory perception, the sound does not appear mechanical at all but rather very fluid. The generated timbre is clearly recognizable and attributable to the original instrument.

4.1.3 WITHOUT CONDITIONING: A CONVOLUTIONAL VAE APPROACH

Among the various experiments conducted, I report the attempt to remove the conditional part from the model. This can be achieved by simply removing the labels from the reference dataset and adjusting the model training accordingly. In addition to implementing the models already presented in the previous paragraph (which also yielded unsatisfactory results), we considered trying the use of convolutional layers.

Using convolutional layers in variational autoencoders could yield significant benefits in certain specific tasks. Specifically, they are highly effective and widely used in classification algorithms, denoising operations, and capturing specific features. Convolutional layers have their natural application in computer vision and, in general, in tasks related to extracting valuable information from images. Since we are using spectrograms as input, which can be considered as “images” of sound, it was advantageous to employ this technique. In fact, a convolutional variational autoencoder is capable of extracting valuable information from spectrograms as well. However, this approach proves to be less suitable, both empirically and computationally, for sound generation objectives.

This is related to the lack of constraints in the latent distribution, leading to the generation of sounds that are less consistent with the original ones or otherwise unsatisfactory. At the computational level, the algorithm is less efficient, more complex, and with significantly longer execution times, taking 3 seconds per epoch. The non conditional convolutional model is not well-suited for sound generation. It is worth noting that this type of model is rarely used for musical and generative purposes in the scientific literature.

Below, we present a typical poor generation in the case of not conditioning the autoencoder, despite the implementation of convolutional layers.

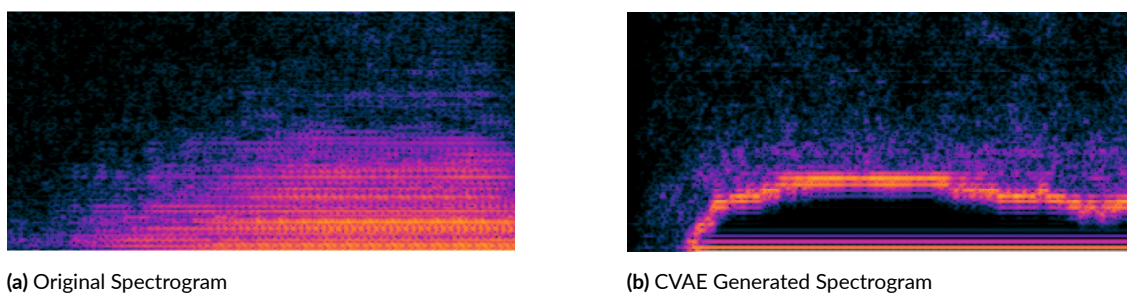


Figure 4.4: Conditional VAE Generation - CELLO Sound

In the case of cello sounds, you can observe the following characteristics of the generated sound:

- It is very noisy, and the sound intensity is significantly lower, making the audio clip almost distant in perception.
- Sometimes it reproduces only the fundamental frequency without capturing the timbral characteristics of the sound. This is in stark contrast to the primary goal, which aims for a purely timbral analysis.

4.2 GENERATION OF SOUND MORPHING

In general, variational autoencoders work quite well for sound generation. However, the assessment parameters for generation vary depending on whether the sounds are harmonic or non-harmonic. In harmonic sounds, the most significant parameter is the timbre of the instrument, which is the auditory recognizability of the instrument producing the sound. In non-harmonic sounds, the most significant parameter is how well the generated sound preserves some form of structure. For example, the evaluation includes assessing how well a continuous sound has preserved its continuity or how well a scratched sound has retained its original character.

These evaluations are based on two characteristic data comparisons:

- Original mel-spectrogram vs. generated mel-spectrogram. Even at a visual level, it is possible to determine if the reference pattern has been reproduced. Measures such as loss scores or Mean Squared Error (MSE) can also be helpful in this regard.
- Original sound vs. generated sound. For this assessment, only the human ear is taken into consideration.

A crucial step to proceed with sound evaluation is the conversion from a spectrogram to audio. This essentially requires the reverse process compared to the initial preprocessing. The most widely used and highly effective method is the Griffin-Lim[45] algorithm. Its mechanism is based on an initial estimation of the sound's phase. Then, through an iterative process until convergence, the audio signal is reconstructed using the inverse Fourier transform and information obtained from the spectrogram, such as magnitude. At the end of this process, the reconstructed signal should closely resemble the original audio. To achieve accurate reconstruction, it is essential to use the same parameters as those used in the input preprocessing, namely a hop length of 256 and a frame size of 1024. The functioning of the Griffin-Lim algorithm is described below:

Algorithm 1 Griffin-Lim Algorithm

```
1: Set:  $\angle X_0(i)$ 
2: Initialize:  $X_0(i) = |X(i)| \cdot e^{j\angle X_0(i)}$ 
3: for  $n = 1, 2, \dots, N$  do
4:    $X_n(i) = T(IT(|X(i)| \cdot e^{j\angle X_{n-1}(i)}))$ 
5: end for
6:  $\hat{x}(n) = IT(X_N(i))$ 
```

Figure 4.5: Griffin Lim Algorithm

Before to continue, it's important to note that the parameters for assessment may vary depending on whether the sounds are harmonic or non-harmonic, and these considerations are essential when evaluating the quality of sound generation.

NON-HARMONIC SOUND

As mentioned earlier, in the case of non-harmonic sounds, the evaluation of generation is based on how well it reflects its initial characteristics (see the table in section 2.2.1).

A fundamental consideration to make is that the quality of generation is inherently linked to the complexity of the sound. In fact, it is quite interesting to observe the difference between generating a simple sound and a more complex sound during the training phase. Below is an image of the generation of a simple sound after only 5 epochs.

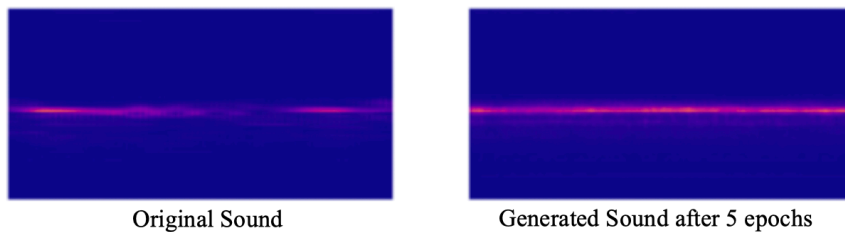


Figure 4.6: Sound Generation - Simpler non-Harmonic Sound

While even the simplest sound has not been perfectly generated, it is evident that its structure has been largely captured by the model, which will be able to refine the sound further with additional epochs.

On the contrary, for a complex sound, capturing the pattern requires many more epochs to achieve even a minimum resemblance to the original spectrogram. A complex sound, after 5 epochs, has not been understood at all and requires at least 100 epochs for a minimal pattern definition.

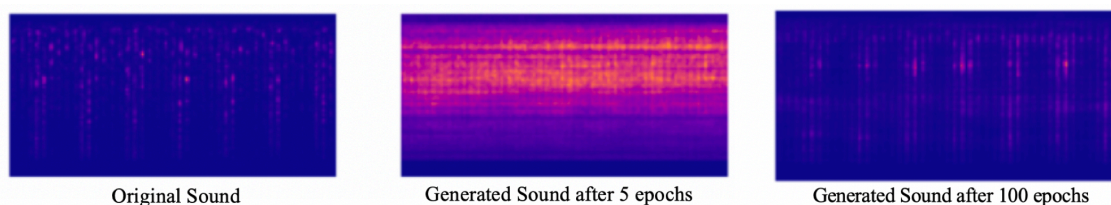


Figure 4.7: Sound generation - More complex non-harmonic Sound

While it may appear that the pattern has been recreated after 100 epochs, a closer look reveals that the generated sound is rigid. This results in a sound that, while retaining the pattern, is somewhat mechanical and less natural compared to the original.

This effect is a direct consequence not only of the types of sounds, whether more or less complex but also of the dataset itself. In fact, 500 clips have been provided for each sound type. For some sounds, the clips are very similar to each other, which becomes evident even upon initial listening. Special attention should be given to all those sounds with rapid variations over time. These variations are very rapid and highly variable from clip to clip. This could explain the greater difficulty in capturing the main features. In contrast, continuous or deep audio clips have minimal variations, both in terms of time and frequency. They are also very stable in each audio clip, which greatly aids the model's training. To obtain measurable confirmation of this, I randomly selected a certain number of audio clips from the same sound class for two different sounds: one with a clear dominant frequency, continuous and static, and one with rapid variations over time. When calculating the Mean Squared Error between samples of the same class from their original data, the continuous sound showed significantly lower MSE. Therefore, we can confirm that the internal variability between classes also contributes to a simpler or more complex generation by the model. Below are images of three sounds from the same class compared, one more static and one with rapid variations over time.

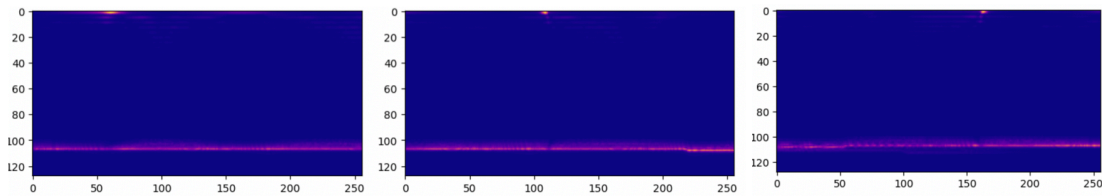


Figure 4.8: Static sounds of the same class

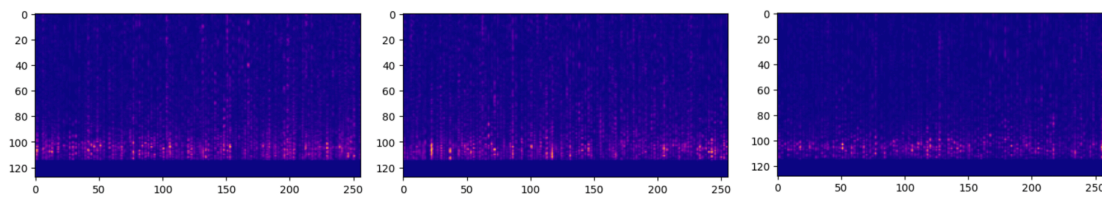


Figure 4.9: Dynamic over time sounds of the same class

There is a clear distinction between continuous sounds and more dynamic sounds, and the complexity of sound plays a fundamental role. However, in general, sound generation for non-

harmonic sounds is quite good. In addition to generating the main characteristics of the sound, other elements such as sound intensity and their development over the audio clip duration are correctly regenerated.

Below are three generated sounds, specifically sound 0, 10, and 16. Please note that the good reconstruction is evident not only from the mel spectrogram, which is the model's input but also from the spectrogram and the corresponding waveform.

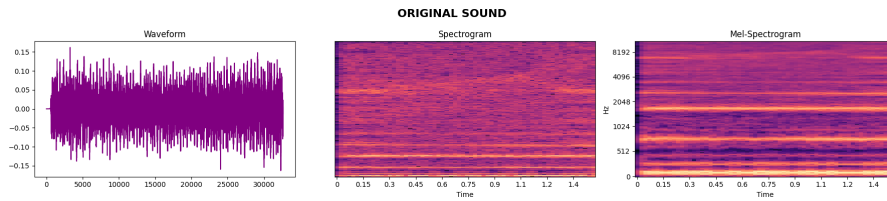


Figure 4.10: Original - Sound 0

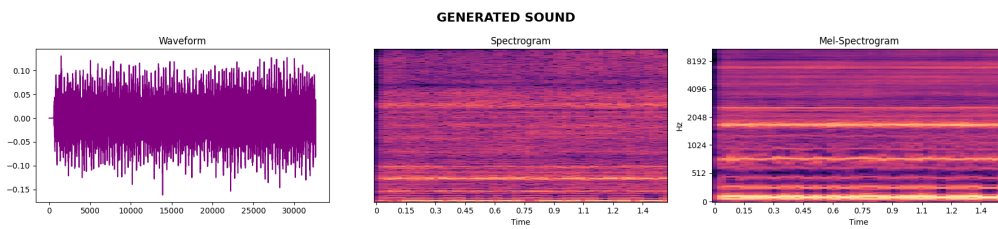


Figure 4.11: Generated - Sound 0

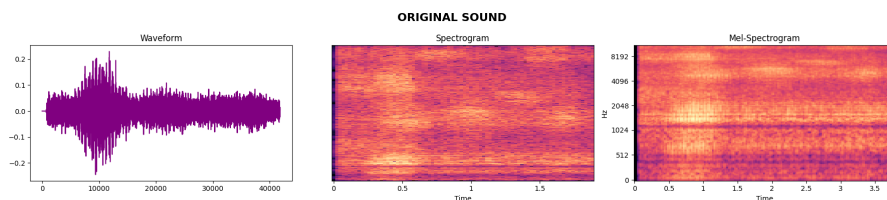


Figure 4.12: Original - Sound 10

Overall the generation of non-harmonic sounds is quite good. Perceptually, the generated sounds preserve all the main properties of the original sounds. In particular, smooth, continuous, and deep sounds are particularly recognizable, while glitchy, sizzling, and sci-fi sounds are a bit more challenging to generate. However, this study does not only focus on sound generation but also on its analysis. That's why this paragraph will provide a detailed analysis of the latent space of the reference dataset.

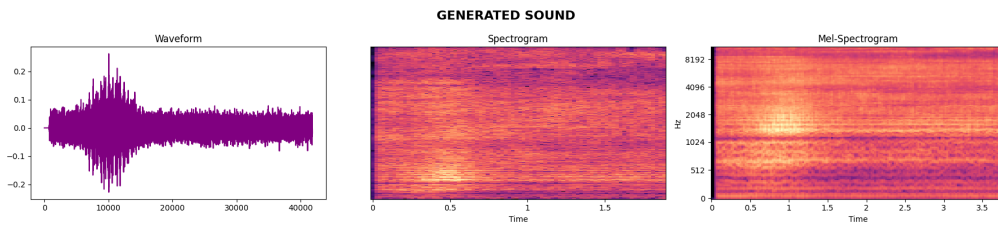


Figure 4.13: Generated - Sound 10

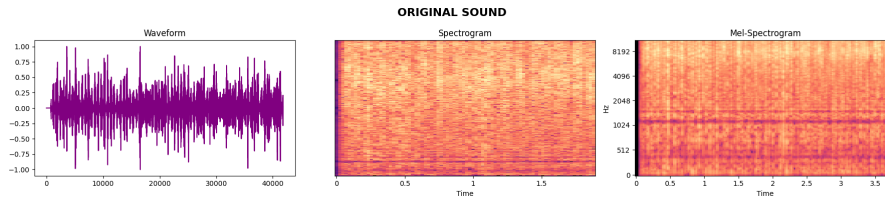


Figure 4.14: Original - Sound 16

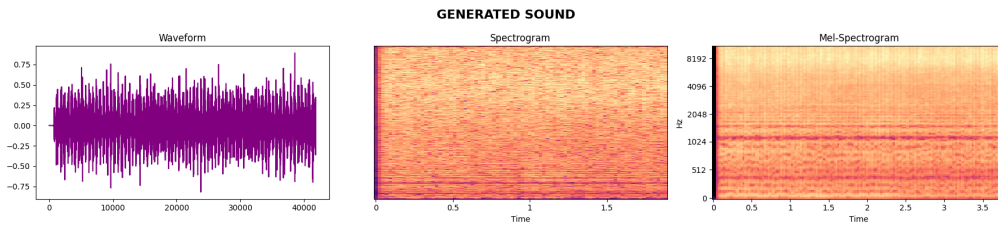


Figure 4.15: Generated - Sound 16

For constructing the projection of the latent space, the model was chosen with a latent space dimensionality of 256. Dimensionality reduction was applied using PCA and T-SNE. Specifically, for PCA, it was chosen to visualize the projection in the case of both 2 and 3 components.

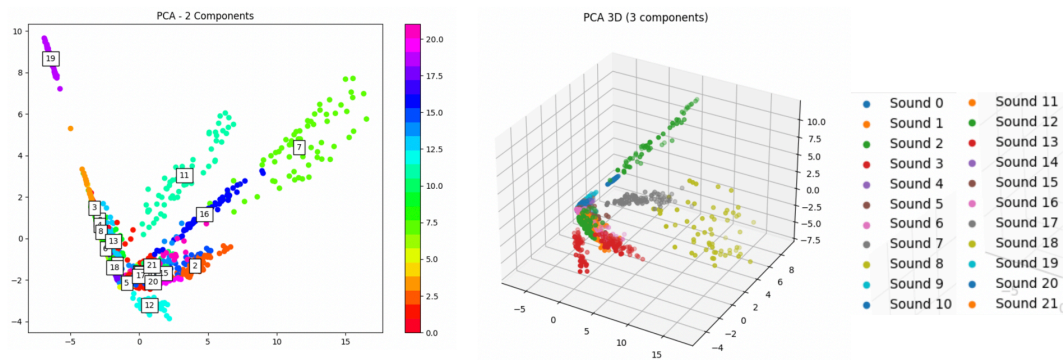


Figure 4.16: Principal Component Analysis with 2 and 3 components

Regarding T-SNE, it was chosen to visualize 2 components with a perplexity of 10, an early exaggeration of 40, and a number of iterations of 3000. The choice of these values is not random but the result of several attempts. The combination of all these values is what made such a distinctly separate visualization between sound classes possible. The space, being regularized, is taken from the various classes in a fairly uniform manner.

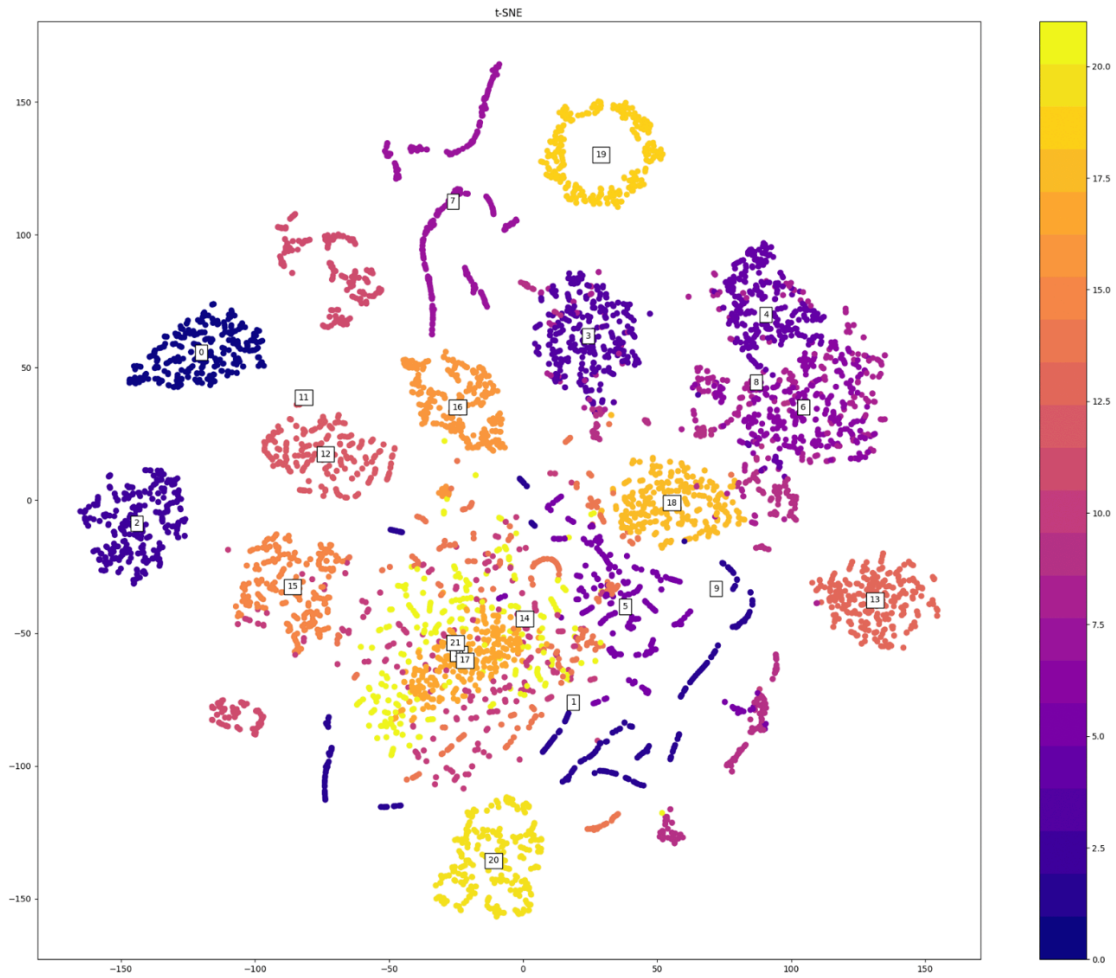


Figure 4.17: T-SNE for non-Harmonic sounds with 2 components

First of all, it should be noted that the further the classes of points are from the center, the clearer the characteristics and well-defined patterns they exhibit. Conversely, central points show less evident patterns, often due to the sound itself (which has a less pronounced pattern) but also due to the model's inability to capture all the nuances.

We can imagine the projected latent space as divided by two diagonals that describe its basic

characteristics. Specifically, there are two characteristics that best define the latent space:

- Smoothness. Continuous sounds and sounds with frequent temporal variations and discontinuities are extremely separated in space. For example, the first quadrant consists of the smoothest sounds among all, while in the third quadrant, sounds are full of discontinuities and sonic distortions.
- Frequency. The dataset is very diverse regarding frequency bands, as deep sounds have very low frequencies, up to sounds with variable and higher frequencies.

HARMONIC SOUND

For harmonic sounds, faithful timbre regeneration is of utmost importance. In this regard, VAE models can be considered an excellent solution, as the generated sounds have proven to be highly proficient in producing a coherent representation of the reference musical instrument.

For a comprehensive sound analysis, it is necessary not to limit oneself to the evaluation of musical timbre alone; there are additional aspects to consider.

Remember that the initial dataset consists of sounds that not only have different timbres but also various characteristics. The frequencies vary and change not only depending on the nature of the instrument but also among sounds of the same instrument, there are variations in pitch and frequency (see the figure below, where the black line represents the fundamental frequency range for each musical instrument, while the striped pattern represents harmonics range).

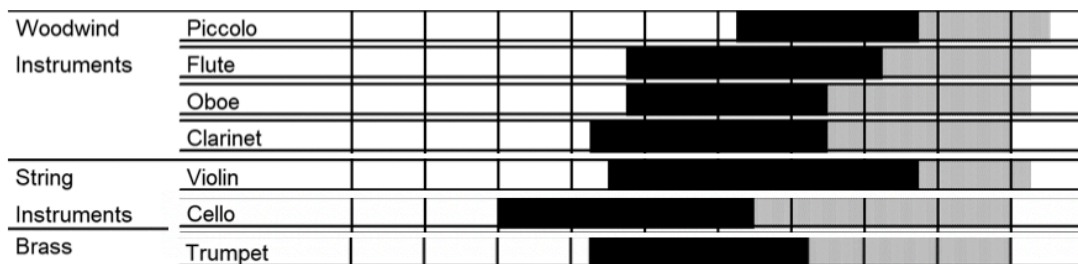


Figure 4.18: Sound Range for each musical instrument

It is important to consider that the regeneration is optimal for all types of frequencies and instrument ranges.

However, the higher the frequency emitted by the instrument, the lower its resolution and pleasantness. While in low-frequency clips (clarinet, cello...), the sound is clear and deep, in higher frequencies (such as violin and piccolo), it tends to be shrill and less natural. Nevertheless, overall sound quality is perceived as quite good.

Here is the generation of the spectrogram at the beginning and at the end of the training process.

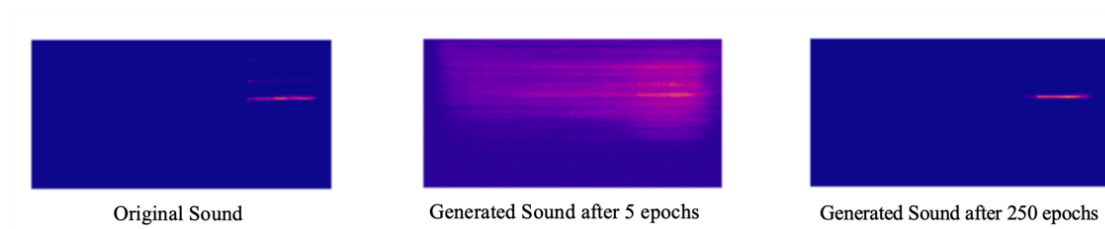


Figure 4.19: Harmonic Generation - an example

Both in terms of sound and in terms of the spectrogram, the sound is generated very well. It is possible to recognize the musical instrument and its timbre in each sound. For an overview of the excellent performance of the model, please refer to the following image.

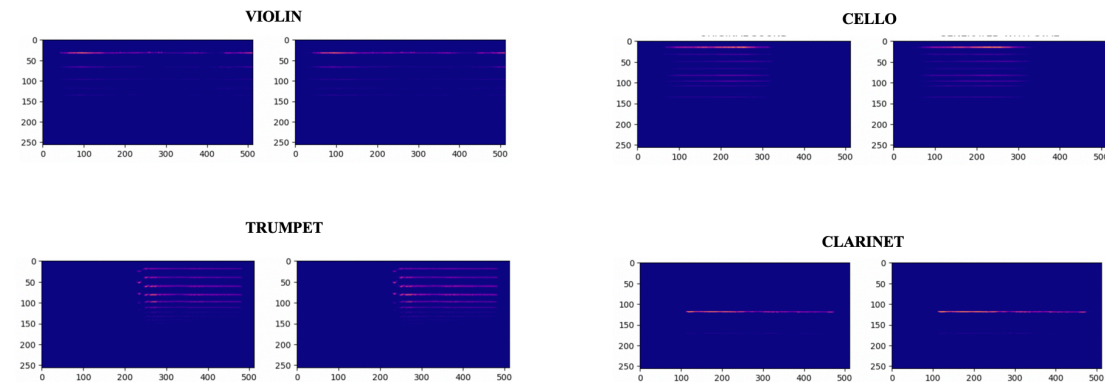


Figure 4.20: Original VS Generated Instruments

Although the model's performance is excellent, there are two inaccuracies that arise in some generations. The first inaccuracy concerns the discontinuity of the sound. It is not always regenerated too similarly to the original, as the model tends to make the discontinuous sound less discontinuous. However, this does not represent a significant issue in this research, as our goal is to generate a good characteristic sound, not an identical copy of a sound.



Figure 4.21: Original VS Generated Sound

The second inaccuracy concerns the generation of harmonics. While it is true that the fundamental frequency is always reproduced optimally, the harmonics tend to be abundant. This often happens in sounds with delayed attack or with many harmonics in their spectrogram. Here, the inaccuracy has a negative impact on sound production, as it results in the presence of disturbing frequencies when listening. They are not entirely unpleasant, but they are certainly noticeable and make the sound heavier. However, the sound still appears more or less natural.

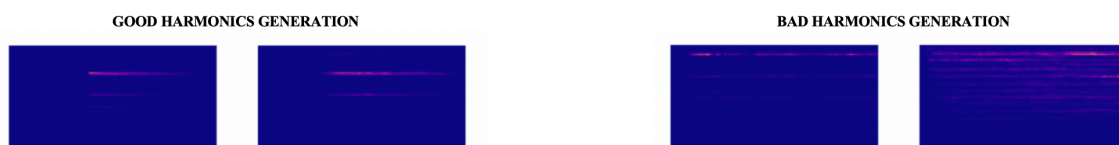


Figure 4.22: Original VS Generated Sound 2

Another fundamental element is the so-called sound envelope, which represents the variation in intensity and amplitude of a sound over time. The most common model of an envelope is the ADSR[46] (Attack, Decay, Sustain, Release), which evaluates the sound's change over time by dividing it into four phases: Attack, Decay, Sustain, Release.

In general, we can observe that the generation is quite consistent throughout the audio, except for the attack phase, whose accurate generation is closely related to the reference audio clip. In fact, the timing variability of the attack varies greatly from clip to clip. Some audio clips have the sound attacking shortly after the beginning, while others have the sound attacking towards the end. Moreover, the nature of the attack can differ; it can be sharp and sudden or softer and more gradual. Despite the significant variability, it is easy to notice that the immediate attacks are typical of clarinet and trumpet clips, while less immediate attacks (starting from the middle of the audio) are typical of oboe and piccolo. The cello stands out for having rather powerful attacks.

These observations are crucial for a thorough analysis of the generative model’s performance. The attack phase is what typically poses the most challenges in the case of delayed attacks. The phenomenon that occurs is clearly visible in the spectrogram shown below:

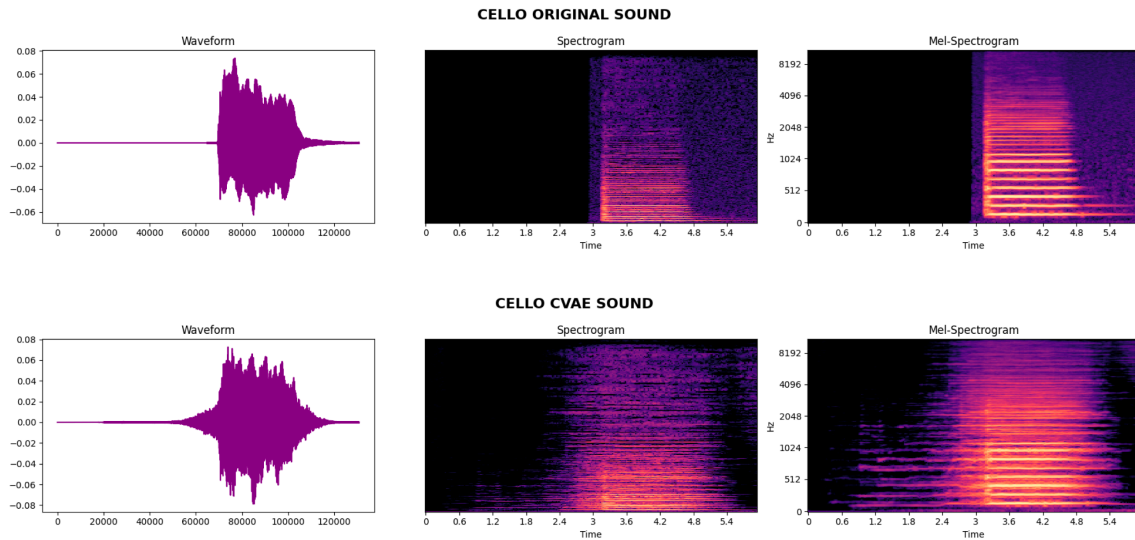


Figure 4.23: Attack Issue in CVAE Generation

While the model faithfully reproduces the original sound in immediate and soft attacks, a peculiar sonic effect occurs in sudden and delayed attacks, preceding the actual attack. It is plausible that the model, trained on temporally diverse attack data, attempts to fill the silence preceding the attack with harmonic sounds in some way. This suggests that the latent space can capture information about the timing of the attack but is less effective at capturing information about moments preceding it, which are filled with attack frequencies but at a lower intensity. The resulting effect is neither the most pleasant nor the most unpleasant. It’s as if it sonically prepares for the actual attack, which then occurs at the right moment. While this doesn’t compromise the sound generation after the attack, it’s worth noting that the attack phase is an essential part of the sound, as it clearly defines its nature from the beginning. This may be seen as one of the model’s significant limitations.

In the following table, we provide a concise summary of some sound characteristics and their evaluation in the generated sounds.

Characteristic	What is it?[47]	Generation Result
Timbre	Attribute of auditory sensation in terms of which a listener can judge that two sounds similarly presented and having the same loudness and pitch are dissimilar (ASA, 1960)	Clearly recognizable by the human ear, even by non-experts (see the following chapter for further details).
Pitch	Attribute of auditory sensation in terms of which sounds may be ordered on a musical scale (ASA, 1960)	Extremely similar to that of the original sound.
Intensity	Subjective perception of sound pressure	Extremely similar to that of the original sound.
Envelope	How the level of a sound wave changes over time	Similar to the original sound, but with issues related to the attack depending on the instrument and the reference audio clip. Some difficulties in case of tremolo or discontinuity in the sound.

Table 4.2: Harmonic Generation Evaluation

In this case as well, an analysis of the latent space through projection and dimensionality reduction can provide a clear understanding of the encoding process.

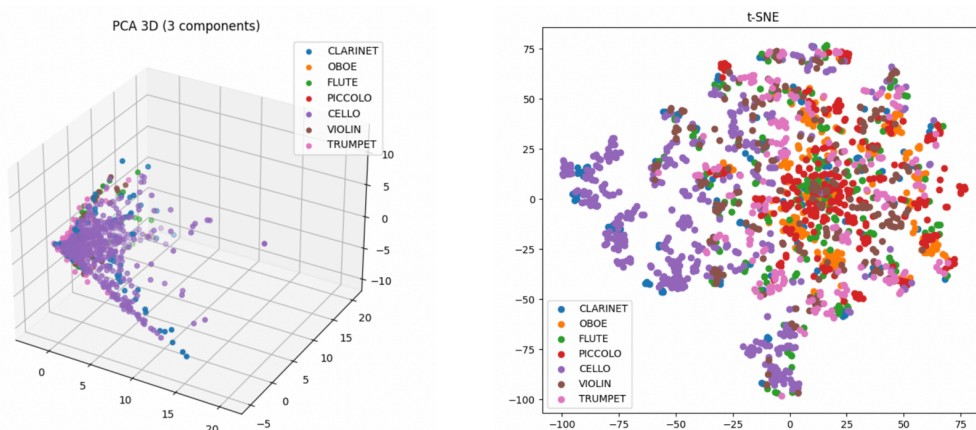


Figure 4.24: Latent Space Plots - 3D PCA and T-SNE

Unlike non-harmonic sounds, in this case, the classes are not so separable from each other. However, this should not be a problem as it does not significantly affect the generation process. While a latent space with linearly separable classes ensures excellent class classification, it is also true that we are using a variational autoencoder that regularizes the latent space, mak-

ing it more challenging to separate the classes without significantly affecting their generation. A clear example of this is seen when applying the same model with the same parameters and hyperparameters but without regularization, i.e., implementing an autoencoder without the variational component.

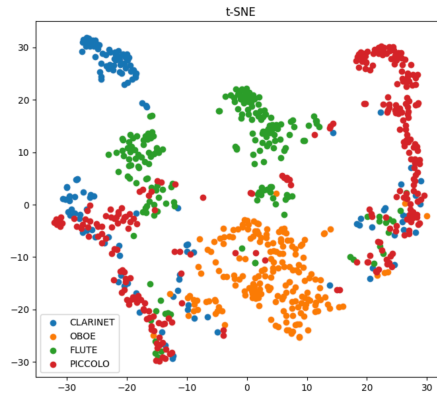


Figure 4.25: Woodwind Latent Space with AE

The reason behind this limited separation between classes may be that many instruments share frequency bands, and frequency is considered the most relevant feature for sound classification by the encoding. This is evident from the plot, as the class furthest from the others is the cello class, which has the lowest and most distinct frequency range compared to all other instruments. The closer one gets to the center, the higher the frequency. It is no coincidence that the outermost points are from the cello, clarinet, and trumpet classes, while the innermost points are from the piccolo and violin classes. I refer the reader to Figure 4.18 for an overview of the frequency range of each instrument, perfectly respected and cataloged in the acquired latent space.

However, frequency is not the only discriminator. In fact, the characteristic that truly maps the points in the latent space is the timbre. It is related to three implicit features: frequency and harmonics (that we have already discussed), attack/decay, and tremolo.

Specifically, it can be observed that there is a clear distinction between continuous and well-defined sounds (right part of the plot) and discontinuous sounds (left part). Furthermore, the area around is characterized by well-defined frequencies and harmonics with strong harmonics present. The closer one moves towards the center, the less defined the sounds become. This can also be caused by limitations in sound reproduction or a less clear timbral quality.

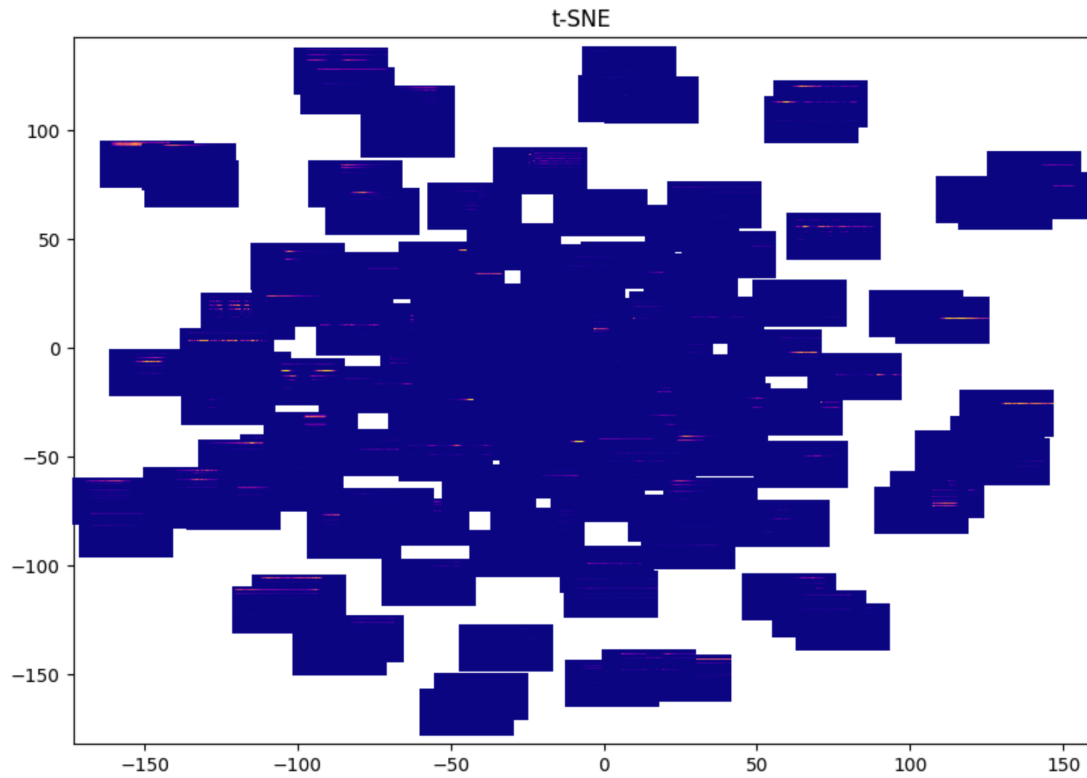


Figure 4.26: Mel-spectrograms distributed in Latent Space

It's possible to conclude this section by stating that the latent space is highly capable of capturing nuances of sound for each type of sound, and even in the case of a latent space with separable classes, sound generation continues to be optimal.

5

Experimental evaluation

In this section, I will illustrate the process that led to the creation of audio clips that smoothly blend different sounds together in a gradual and controlled manner. The goal of good sound morphing is indeed to transition from an initial sound to a final sound that may be more or less different (with perceptually different characteristics such as timbre or pitch) while traversing all characteristic combinations between the two sounds without a significant break between them. Fluidity, smoothness, and naturalness in the sound change are the ultimate objectives of this process. In the scientific literature, there are various methods to perform sound morphing[40]. In the field of sound, there has been extensive exploration of the interpolation[36] technique, which allows us to interpolate the most relevant features of a sound by controlling a single parameter, alpha, a coefficient that ranges from 0 to 1 and determines the level of interpolation. In this research, we opted for the use of the weighted average technique, which manages the metamorphosis process from one sound to another through a control parameter.

5.1 HOW TO COMBINE SOUNDS

The basic operation is very intuitive: sounds with a different degree of feature mixing are concatenated together, starting from the initial sound and ending with the final sound. The starting point is the latent space, which maps the sound through some of its characteristics. For example, in the case of harmonic sounds, the sounds are mapped for their frequency and smooth-

ness. At this point, it is necessary to choose two points in space, preferably from two classes of sounds that are more or less different from each other. Then, the weighted average formula is applied:

$$\text{Weighted Average} = \frac{\text{weight1} \cdot \text{point1} + \text{weight2} \cdot \text{point2}}{\text{weight1} + \text{weight2}} \quad (5.1)$$

This will generate different combinations of features that, depending on the weight (ranging from 0 to 1) assigned to each observation, will be more or less similar to the initial sound. The weight of 0 corresponds to the absence of characteristics from the first chosen sound and the presence of characteristics from the second observation (sound). Intermediate points, such as a weight of 0.50, indicate a perfectly equal mix between the initial and final sounds. It ends with a weight of 1, equal to 1, which means the total absence of characteristics from the second chosen sound. At this point, the decoding part of the CVAE can be applied to regenerate the sounds and obtain their respective spectrograms. It should be noted that the decoding is made independent of the reference classes, allowing for the generation of sounds of a different nature without the constraint of belonging to a specific class (which would be an obstacle to mixed generation).

The figure below illustrates the process leading to the formation of various sound combinations:

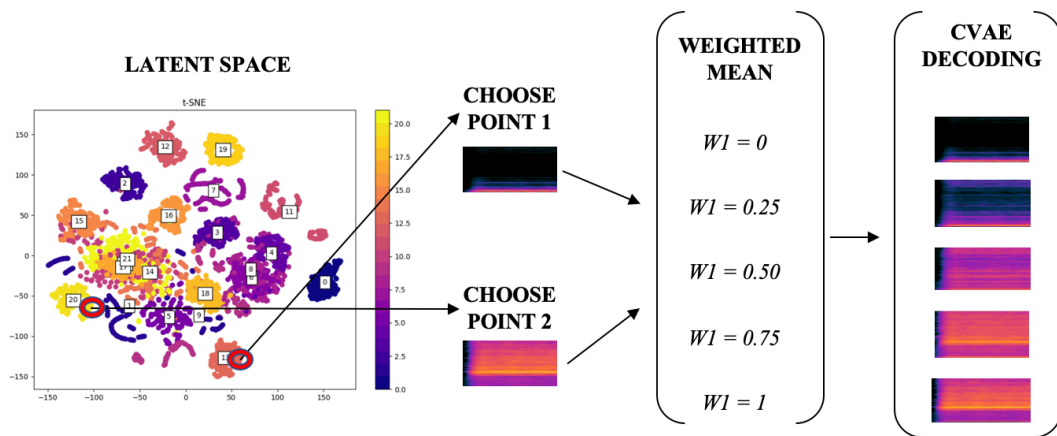


Figure 5.1: Sound morphing process - pt.1

5.2 HOW TO IMPROVE SMOOTHNESS AND VARIABILITY

While the creation of audio that blends points in the latent space is the fundamental core of the process, it is also true that there cannot be successful sound morphing without a good connection between one observation and another. The risk of concatenating observations without some level of preprocessing inevitably leads to sound distortions or clicks. Furthermore, another fundamental problem is the lack of sound variability. These issues attempt to find a solution in some operations that will be presented in this section.

FADE IN/FADE OUT AND OVERLAP

After creating various audio mixes, it is necessary to concatenate them in a thoughtful manner. To achieve a natural and smooth transition, audio clips should overlap at their respective initial and final parts. Additionally, some fading in and out at the entry and exit points of each clip is required for smoother transitions. Specific techniques are employed for this purpose.

Fade In is a gradual entrance of sound, while Fade Out is a gradual exit. They work by selecting a reference function, which can be of various types. In this research, linear, exponential, and logarithmic functions were experimented with. While the exponential function proved to be ineffective, linear and logarithmic functions are suitable for smoothing the transition between clips. For instance, in the case of harmonic sounds, a logarithmic Fade In/Out is optimal, while for non-harmonic sounds, a linear function suffices for simple sounds, and the logarithmic function is more suitable for complex sounds. There is a portion of sounds (when concatenating two complex and substantially different sounds) where the transition is not very smooth even with the use of a logarithmic function. For experiments on this concatenation, Reaper[48] was chosen, as it provides great flexibility and manipulation of these operations, designed for music production.

Once an appropriate late fade in/out is applied to the audio, overlapping is applied according to the chosen overlap coefficient, which determines the portion of sound to be matched.

Below is an explanatory plot that represents the concatenation of two violin audio clips.

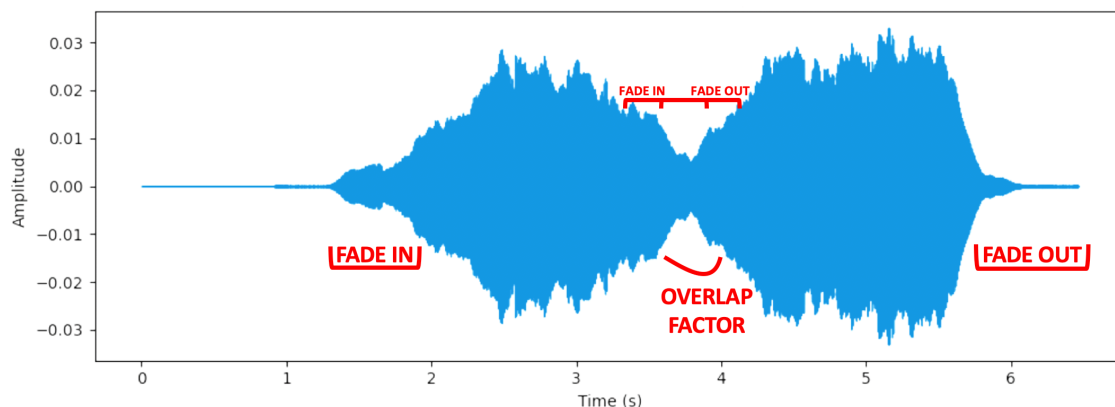


Figure 5.2: Preliminary procedures for sound morphing - fade in/fade out + overlap

PERLIN NOISE

The variability in sound morphing can be increased in two ways:

- Changing the observation by selecting one from the same class to vary the type of sound while preserving its characteristics.
- Applying Perlin Noise of different intensities during the sound morphing process.

Perlin Noise is a gradient-based noise commonly used in computer vision applications but applicable to sound as well. While adding noise to a spectrogram and, consequently, its audio waveform does alter the audio, it's important to note that too random of a noise component leads to an unpleasant and non-uniform mutation, resulting in audio more distorted than the original. Perlin Noise, on the other hand, has the characteristic of being controlled noise, which is why it is widely used for creative purposes. The parameter controlling its complexity is referred to as “octaves”. Below, you can see a non-harmonic sound from the dataset in its variations depending on the type of Perlin Noise injected. It's easy to observe that the spectrogram changes, but only slightly and in a completely controlled manner, preserving its distinctive characteristics.

In the image below, you can see how the spectrogram of a sound changes depending on the different type of Perlin noise added.

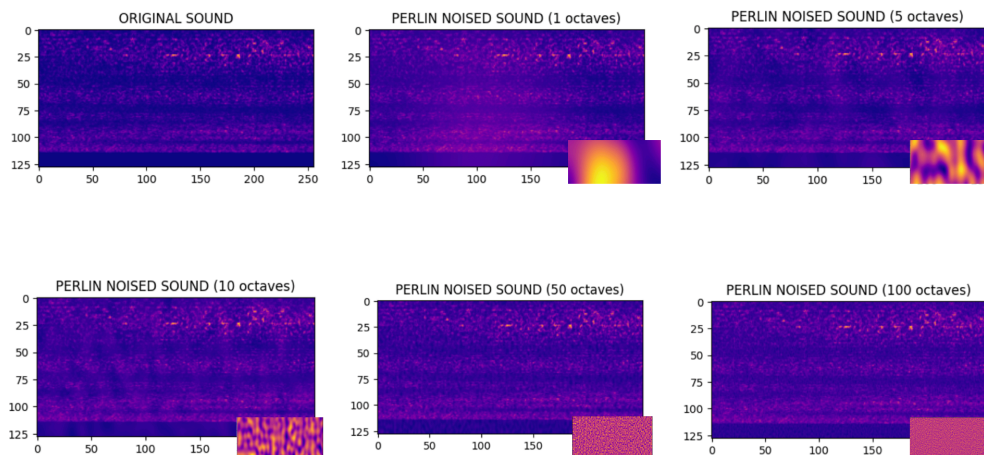


Figure 5.3: Perlin Noise Injection effect

5.2.1 GENERAL CONSIDERATIONS

In general, the sound is quite pleasant, and the transition from one sound to another is generally very smooth. For further considerations on the quality of sound morphing, please refer to the following chapter. Before delving into the details, it is appropriate to make some considerations about the operations performed for the creation of sound transformations:

ADVANTAGES:

- There is a high level of control. The ability to choose the number of observations to generate before arriving at the final observation can make the transition very fluid, especially when dealing with sounds with very different characteristics. The control over the overlap coefficient also contributes to smoothness.
- The latent space allows for combinations with great flexibility. Even with a simple calculation like weighted averaging, it is possible to obtain various results. This allows for the mixing of many different types of sounds, starting from a reasoned combination of their main characteristics. Understanding the ratio with which points are projected into the latent space enables the creation of sounds with well-defined features.

DISADVANTAGES:

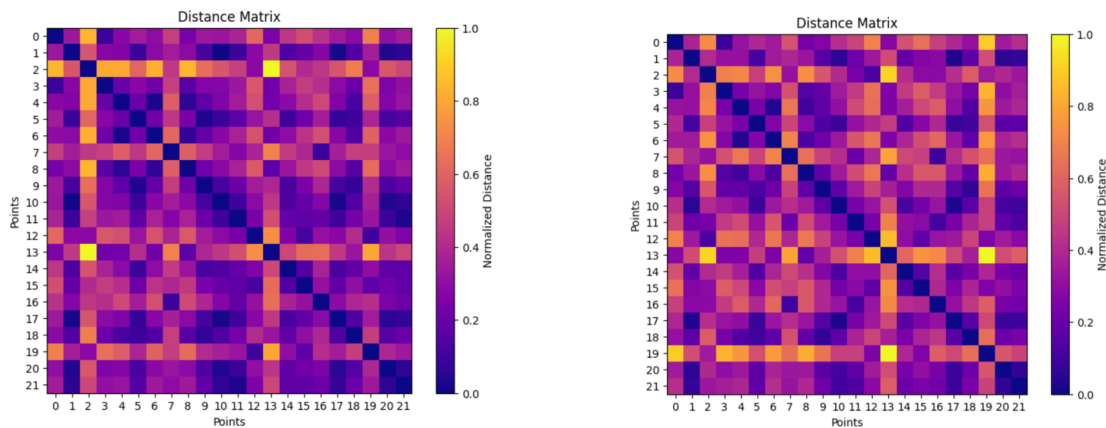
- The final result depends too much on parameters and hyperparameters. This problem is closely related to the nature of the data, as the attacks in the clips vary in time, leading to very short-duration sounds in some cases (e.g., in oboe or piccolo clips). The shorter

the actual sound duration, the less possibility there is for smooth transitions unless a programmer meticulously controls the overlap coefficient or adjusts the starting point of the sound. This is further compounded by the model's inability to reconstruct silence in the case of clips with delayed attacks, filling the silence with preparatory and inappropriate frequencies.

- The smoothness is too closely tied to the distance between points in the latent space. The greater the distance between points, the greater the difference in sound characteristics, and the more intermediate steps are required for good smoothness. The number of steps, along with other parameters, requires careful control.

NON-HARMONIC SOUNDS

For non-harmonic sounds, we want to analyze how the sound changes as the differences that most characterize the distribution of the latent space, namely smoothness and frequency, vary. To assess the distance between sound classes, in order to conduct more diverse sound morphing experiments, a distance matrix has been created, both for the centroids of each class and for the median points.



(a) Average of points for each class.

(b) Median of points for each class.

Figure 5.4: Distance Matrix - Points in Latent Space

The exploration of the sound space in the case of non-harmonic sounds has been carried out with various combinations. What we wanted to investigate the most is how the distance between classes in the latent space can influence the development of smooth sound morphing,

especially in the case of changes in the main sound-mapping characteristics (smoothness and frequency). Here are some interesting examples:

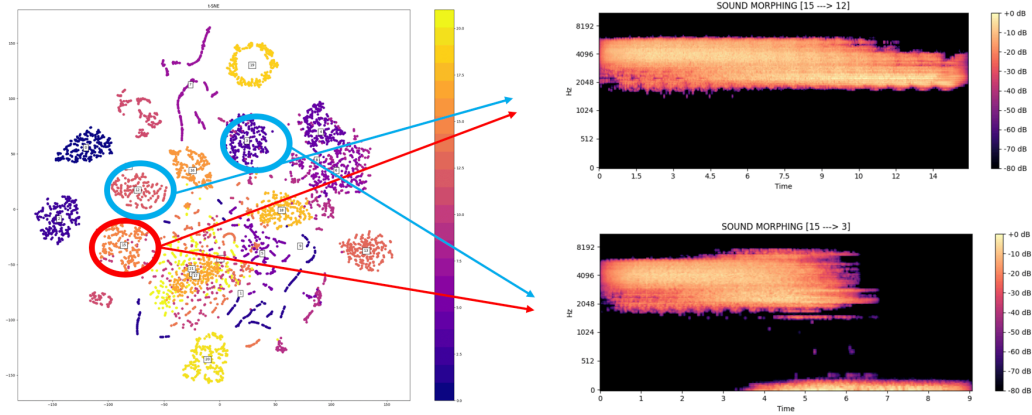


Figure 5.5: Sound Morphing according to distance (15 -> 12 and 15 -> 3)

Starting from sound 15, it is evident how the distance in the space influences the sound metamorphosis. Specifically, note in the transition to sound number 12 how the reference frequency bands are similar, leading to an extremely gradual sound morphing. Instead, to reach sound 3, the spectrogram reaches lower frequencies, resulting in a more significant but still smooth sound change.

Another example of interest, especially for exploring the most distinctive and pronounced characteristics of sound, is the sound morphing carried out on extreme points in the latent space:

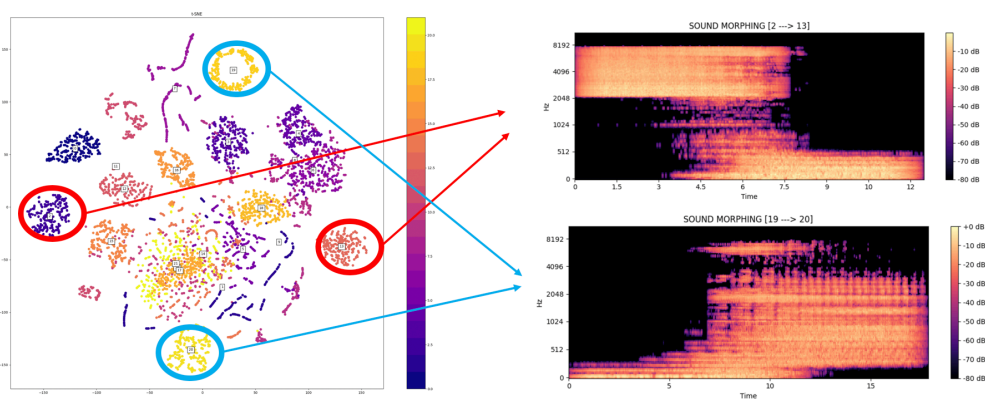


Figure 5.6: Sound Morphing - Extreme distance points (12 -> 13 and 19 -> 20)

In this case, as intended to demonstrate, the characteristics that are mutated along the sound metamorphosis are very defined. Indeed, from sound 2 to 13 (respectively, the sounds farthest to the right and left in the latent space), a significant metamorphosis from high frequencies to low frequencies can be observed. Therefore, it is expected that from sound 19 to 20 (respectively, the sounds highest and lowest in the latent space), the transition involves the smoothness property of the sounds, which happens and is evident from the spectrogram shown above. Note that, in addition to a frequency mutation, the spectrogram changes from being very smooth (with very smooth horizontal lines) to having a vertical pattern, indicating interruptions and temporal discontinuities. This is exactly what indicates the transition from a continuous sound to a discontinuous one (such as glitchy or sizzling classes in the dataset).

HARMONIC SOUNDS

In the case of harmonic sounds, the distance between one sound class and another is likely represented by the timbre and all attributed sounds (with particular attention to the fundamental frequency). Furthermore, the division into classes is already implicitly performed by the instrument class to which the instruments in the dataset belong (woodwinds - strings - brass).



Figure 5.7: Sound Morphing Mechanism for Harmonic Sounds

A first example of sound morphing was performed between cello and trumpet.

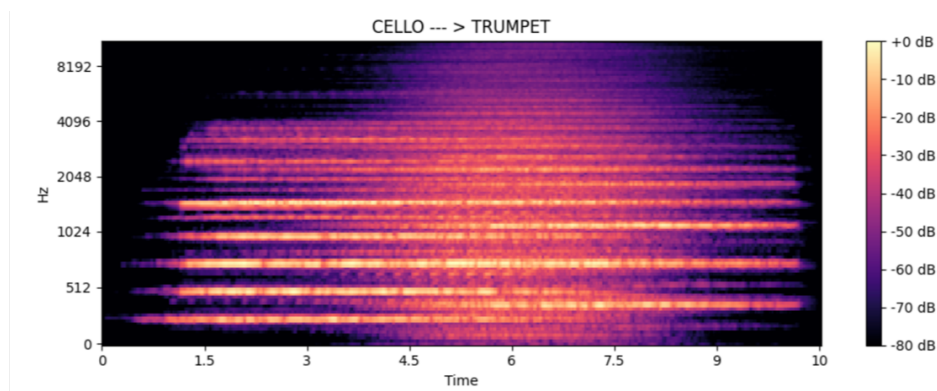


Figure 5.8: Cello - Trumpet Sound Morphing

These two instruments share a fairly close range of frequencies. What distinguishes them significantly are the harmonics and the sound development, with the trumpet being generally more discontinuous. This can be observed in the spectrogram shown below.

In a sound morphing like the one between cello and violin, the fluidity of the transition from a lower frequency to a higher one is evident. Since the sounds belong to the same category (strings), there are no significant additional changes.

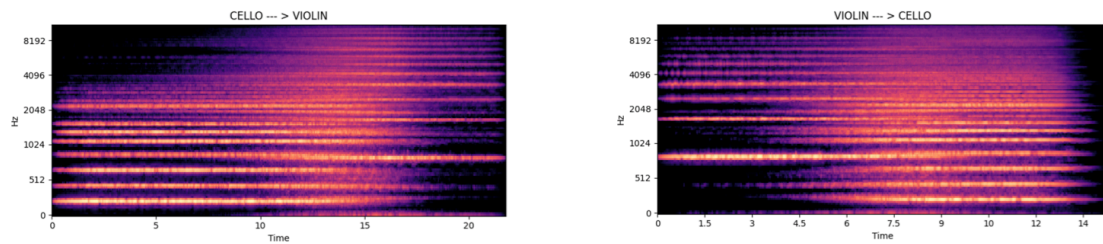


Figure 5.9: Cello - Violin Sound Morphing

Below, other attempts at sound morphing are also shown, and the final results are discussed in the following chapter.

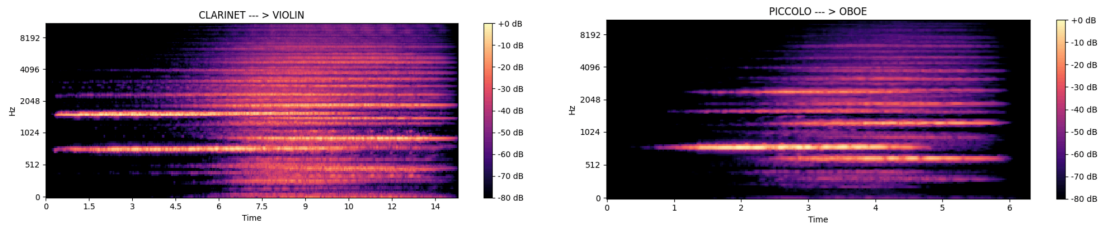


Figure 5.10: Woodwind to String - Woodwind to Woodwind

6

Final Results

6.1 SURVEY

In order to objectively evaluate some key sound characteristics, a questionnaire was administered. The PsyToolkit software was used. Its preparation involved generating 12 different sounds using some of the best experimented CVAE models. Specifically, 4 individual sounds were generated (all non-harmonic), along with 8 sound morphings (4 harmonic and 4 non-harmonic). The generation of sound morphings required careful control of certain parameters, such as the number of steps in transitioning from one sound to another (a parameter closely related to the duration of the audio clip as well).

The purpose of the survey is to investigate the following aspects:

	Non-Harmonic	Harmonic
Quality	Absence of clicks, distortions, or any auditory disturbances.	Absence of clicks, distortions, or any auditory disturbances. Perception of musical timbres.
Class	What perceptual associations are related to the sound.	Which musical instruments are involved in the generation.
Smoothness	Smoothness of the change in sound characteristics (frequency, sound continuity).	Smoothness of the timbral change (gradual and pleasant).

Table 6.1: Survey aims and general structure

In short, we want to investigate how the generated sound approaches certain perceptual and timbral characteristics of the original sound.

6.1.1 PARTICIPANTS

The survey was administered to individuals aged 18 and above. They took part in the questionnaire completely voluntarily and with a guarantee of anonymity in their responses. 45 observations were collected. Survey participants were very diverse, exhibiting a wide range of ages, genders and presence of earing problems.

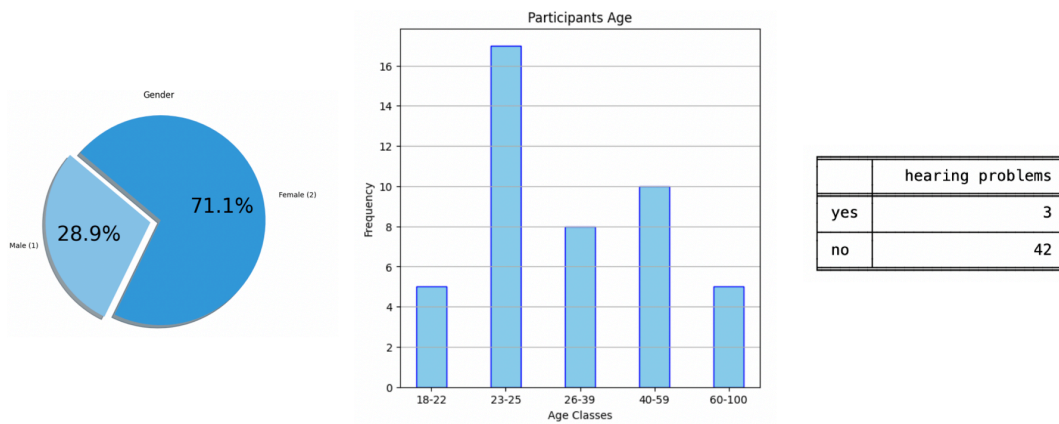


Figure 6.1: Participants plots - Age, Gender and hearing problems

It should be noted that the survey was not administered exclusively to musically competent individuals or musicians. Many of the participants had limited musical expertise, which could potentially lead to different results, particularly in the section related to the recognition of musical instruments, which inherently requires a certain level of experience with sound timbres.

6.1.2 STRUCTURE

The survey structure is as follows:

- First section: listening to 4 classes of non-harmonic sounds, each lasting 8-12 seconds. Each sound was obtained by randomly sampling 5 points from the same latent space class. Random Perlin Noise was injected into each point, which was then transformed into generated sound through a decoding process. Crossfading was applied to each sound to concatenate the 5 spectrograms smoothly. Finally, the Griffin Lim Algorithm

transformed the final spectrogram into sound. This allows the construction of a kind of sound trajectory of the same sound class. This aspect is very important because it ensures variability and dynamism in the generated sound.

- Second section: listening to 4 non-harmonic sound morphings, each lasting 10-18 seconds. The 4 audio clips explore various distances between classes in the latent space, with different levels of intermediate steps depending on the class distance.
- Third section: listening to 4 harmonic sound morphings, each lasting 10-18 seconds. The 4 audio clips investigate changes between different types of musical instruments, ranging from sounds of the same class (for example, from violin to cello) to different classes (from cello to trumpet).

HOW QUESTIONS, AUDIOS AND OPTIONS WERE CHOSEN

For questions about non-harmonic sounds, 4 quite different sounds were chosen (a deep sound, a scratchy sound, one with varying pitch, and one sizzling). The answer options were chosen according to the distance matrix in Figure 4 and the proximity of the classes from the t-sne plot. Specifically, for each question, there are four different answer options:

- The actual class to which the sound belongs
- Two sound classes close to the actual class
- One sound class far from the actual class

For questions about harmonic sound morphing, the answer options are the same for all questions and correspond to various instruments (clarinet, oboe, flute, violin, cello, trumpet). The piccolo, being essentially a higher-pitched flute, will be categorized as a flute, especially to accommodate less experienced ears. Here, it is generally not important to precisely identify the specific instrument but rather the nature of the instrument itself. Therefore, the aim is not for precise and strict instrument identification but rather for the reference class.

6.2 RESULTS

6.2.1 QUALITY

Sound quality was the first characteristic measured. By “quality”, I do not refer to the pleasantness of the sound but rather the absence of any sound distortions, various types of noise, clicks, and, more generally, any auditory phenomena that would disrupt the sound.

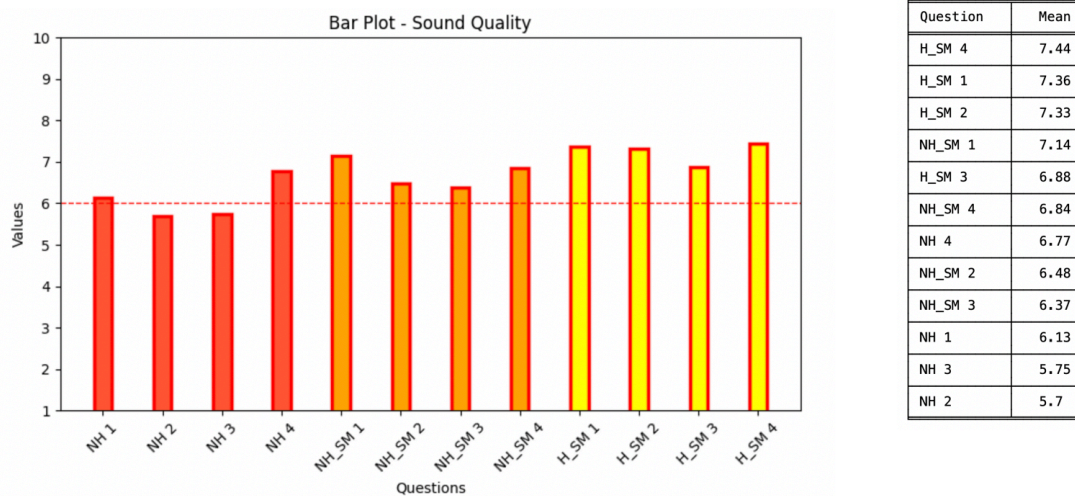


Figure 6.2: Sound Quality Evaluation

The overall evaluation of the sounds is acceptable and reasonably good, as all audio clips were rated as fully sufficient, with the exception of non-harmonic sounds number 2 and 3, which had an average score of 5.5. It is not surprising that non-harmonic sounds were rated on average lower in quality. This phenomenon could also be attributed to the type of sound being reproduced, as it might not be a coincidence that lower scores were given to glitchy and sizzling sounds with very rapid temporal variations. In fact, for somewhat smoother sounds (such as the last audio clip of non-harmonic sounds), the quality improved significantly.

Regarding harmonic sounds, the ear might be more accustomed to them, as they are pleasant to listen to. These considerations could provide a rationale for the variations among different types of sound classes. However, the most important points to consider are:

- The sound quality is good and acceptable but not optimal.
- The variance of the scores is quite high, indicating a wide range of judgments, possibly due to individual perceptions of sound quality.

- Sound morphing received higher average scores than individual sounds.
- Harmonic sounds received higher average scores than non-harmonic sounds.
- The sound with the highest quality is the sound morphing between cello and trumpet, obtained from steady, constant sound clips with clear and stable pitch.
- The sound with the lowest quality is non-harmonic sound number 0, with glitchy characteristics. It is a sound with constant and rapid temporal variations, easily confused by less experienced ears as poor or interrupted sound quality.

In order to verify if there is a significant difference between the different audio clip results, we have chosen to implement some statistical tests. Specifically, we want to test the null hypothesis that there is no significant difference between the results of the different audio clips, and that the quality of the generation is fairly consistent across all audio clips. Therefore, we can opt for either an ANOVA test[49] or a Kruskal-Wallis test[50] (nonparametric test), depending on the behavior of the means and variances of our results. As an already verified assumption, we have the independence of the audio clips. An assumption to be checked, in case of using ANOVA, is that the data should be normally distributed, and the variance between the data should be homogeneous. If the assumptions are not met, we can proceed with a Kruskal-Wallis test.

To verify these two basic conditions, I implement the Shapiro-Wilk[51] test to check for normal distribution and the Levene test[52] to check for homogeneity of variance.

Below are the results of statistics and p-values. As a reference value to accept or reject the null hypothesis, I have chosen the value 0.05.

Shapiro-Wilk	SM1	SM2	SM3	SM4	SM5	SM6	SM7
Statistic	0.92	0.91	0.94	0.93	0.94	0.95	0.95
p-value	0.003	0.003	0.027	0.007	0.023	0.046	0.039

Shapiro-Wilk	NH1	NH2	NH3	NH4	SM8
Statistic	0.96	0.96	0.97	0.97	0.95
p-value	0.163	0.135	0.267	0.267	0.1

Levene Test	Values
Statistic	1.7
p-value	0.069

Figure 6.3: Verifying ANOVA assumptions: Shapiro-Wilk and Levene Tests

It is evident from the results, particularly from the p-value, that the assumption of homogeneous variance is verified, but the assumption of normal distribution is not. In fact, only

some audio clips exhibit normally distributed results, while others show a p-value that leads to rejecting the normality assumption. Therefore, we choose to proceed with the Kruskal-Wallis test since, given the lack of basic assumptions and the small sample size, ANOVA would lack robustness and efficiency.

I proceed, therefore, with the Kruskal-Wallis test to verify significant differences among the results of various clips. Below are the results:

Kruskal-Wallis	Values
Statistic	31.9
p-value	0.0007

Figure 6.4: Kruskal-Wallis Test - Quality Scores

The obtained values make it clear about the data behavior: the null hypothesis, which assumes no significant difference between the groups, is rejected. We can conclude that there are two or more groups showing significant differences in the results. To analyze which groups might exhibit these differences, various statistical tests are available. The most common ones are the Tukey test[53] and the Dunn test[54]. Since the Tukey test is typically used with ANOVA, we opted for the use of the Dunn test, which is closely related to the Kruskal-Wallis test and is its direct consequence.

Below is the table of p-values indicating the significance of differences between the results of the various audio clips. Values below 0.05, signifying significance, are highlighted in red.

	NH1	NH2	NH3	NH4	SM1	SM2	SM3	SM4	SM5	SM6	SM7	SM8
NH1	1.000000	0.340135	0.377927	0.373341	0.024300	0.231703	0.550855	0.150868	0.017712	0.021693	0.167858	0.013878
NH2	0.340135	1.000000	0.946740	0.065951	0.001345	0.031565	0.122345	0.017079	0.000900	0.001178	0.019931	0.000652
NH3	0.377927	0.946740	1.000000	0.078055	0.001800	0.038362	0.141571	0.021151	0.001216	0.001579	0.024560	0.000889
NH4	0.373341	0.065951	0.078055	1.000000	0.177198	0.764930	0.770198	0.586995	0.140725	0.162223	0.626851	0.118423
SM1	0.024300	0.001345	0.001800	0.177198	1.000000	0.290798	0.100345	0.421862	0.895050	0.955318	0.389488	0.825294
SM2	0.231703	0.031565	0.038362	0.764930	0.290798	1.000000	0.553338	0.804733	0.237072	0.268532	0.849408	0.203671
SM3	0.550855	0.122345	0.141571	0.770198	0.100345	0.553338	1.000000	0.403542	0.077528	0.091076	0.436404	0.063805
SM4	0.150868	0.017079	0.021151	0.586995	0.421862	0.804733	0.403542	1.000000	0.352422	0.392862	0.954520	0.308568
SM5	0.017712	0.000900	0.001216	0.140725	0.895050	0.237072	0.077528	0.352422	1.000000	0.939846	0.323675	0.929614
SM6	0.021693	0.001178	0.001579	0.162223	0.955318	0.268532	0.091076	0.392862	0.939846	1.000000	0.362048	0.869894
SM7	0.167858	0.019931	0.024560	0.626851	0.389488	0.849408	0.436404	0.954520	0.323675	0.362048	1.000000	0.282257
SM8	0.013878	0.000652	0.000889	0.118423	0.825294	0.203671	0.063805	0.308568	0.929614	0.869894	0.282257	1.000000

Figure 6.5: Dunn Test for significant differences - p-values table

It can be concluded that the qualitative results of the audio clips exhibit homogeneous variance, different distributions (normal and non-normal), a significant difference between the

various results, particularly involving the difference between single sound and sound morphing results. In fact, based on the tests, it can be inferred that the quality of single sounds was considered significantly lower compared to sound transformations.

6.2.2 CLASSIFICATION

Regarding sound classification, it was approached differently for harmonic and non-harmonic sounds. This is because the aspects being investigated are different. For harmonic sounds, the goal is to determine if the reference instrument is recognizable, while for non-harmonic sounds, the aim is to investigate if the latent space is capable of perceptually mapping the sounds. In fact, the descriptions of the generated sounds are not objectively traceable to a specific sound source but rather to a kind of sensation (e.g., an artificial simulation of cricket sounds) that the sound conveys through more prominent auditory features (e.g., a deep sound).

In this perspective, below are the obtained results:

NON - HARMONIC

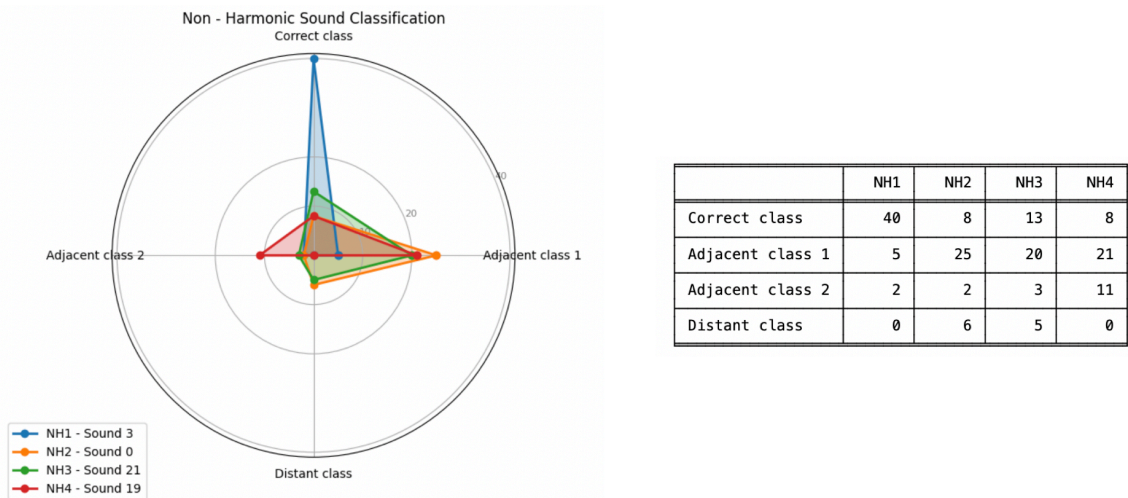


Figure 6.6: Non Harmonic Sound Classification according to CVAE Latent Space

The results of perceptual recognition for non-harmonic sounds are not excellent but certainly promising. It can be observed that:

There is always a clear predominant response, except for the last audio clip, which appears to have mixed responses. Only one observation (the first audio clip) was correctly associated

with its reference class, and with a significant majority of votes. For the other classes, the obtained responses are not optimal from a purely classification perspective. However, the highly promising aspect of this study is that the sound is consistently associated with classes adjacent to the reference class. This suggests that, perceptually, the sound is correctly linked to a certain type of sensation. Therefore, we can conclude that the results of this part of the survey are improvable but very promising from a sound perception perspective. They demonstrate a clear and practical indication that the latent space maps sound perceptions, which can likely be recreated starting from the latent space.

HARMONIC

The results of harmonic sound morphing are surprisingly good, especially considering that a significant portion of the participants had little musical expertise, presumably having an untrained ear for timbral recognition.

Each sound morphing represents the transition from one musical instrument to another. The survey required the recognition of both instruments, allowing for the selection of multiple musical instruments.

The audio clips submitted in the survey are quite diverse. Two transitions involve instruments from the same class (flute to oboe and cello to violin), while two transitions involve instruments from different classes (clarinet to violin and cello to trumpet), with varying degrees of frequency distance between them (clarinet to violin with a high-frequency distance, cello to trumpet with a reduced frequency distance).

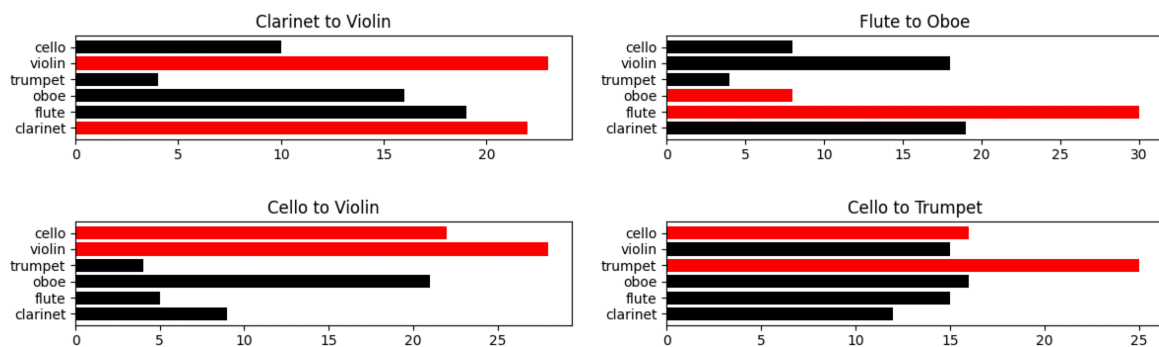


Figure 6.7: Harmonic Sound Morphing Classification Results

It is evident that:

Sound Morphing	Evidences
CLARINET TO VIOLIN	Instruments were clearly identified. There were fewer selections for instruments from similar classes (e.g., flute), while instruments from different classes (e.g., trumpet) were chosen very infrequently.
CELLO TO VIOLIN	Instruments were clearly identified. Numerous selections were made for the oboe as well.
FLUTE TO OBOE	The flute was clearly identified, but the oboe was not. This could be a direct consequence of the oboe's similarity to the clarinet, which is the second most selected instrument.
CELLO TO TRUMPET	Instruments were clearly identified. Numerous selections were made for the oboe and violin as well.

Table 6.2: Harmonic Sound Morphing Results

It can be concluded that the most clear and recognizable instrument is the violin, which received very high scores when present. The trumpet also proved to be very recognizable. The instrument with the least recognizability is the oboe, possibly due to the less experienced ear's difficulty in clearly identifying its sound and differentiating it from similar instruments like the flute or clarinet.

Furthermore, it is evident that the final sound is generally more recognizable than the initial one. This may be due to the nature of sound morphing itself, where the final sound has more potential for sound development, with a fairly clear and recognizable attack and conclusion.

For the statistical evaluation of the previous results obtained, a chi-square test was used. It can be used to determine whether there is a significant relationship between the categories of the variables involved in audio classification. It can be used to check for a significant relationship between generated sound and position in latent space (in the case of non-harmonic sounds) as well as to verify the presence of a relationship between generated sounds and mu-

sical instruments. In short, it is possible to assess the significance of the responses and results obtained.

NON-HARMONIC					HARMONIC						
	Sound 1	Sound 2	Sound 3	Sound 4		Clarinet	Flute	Oboe	Trumpet	Violin	Cello
Adjacent class 1	5	25	20	21	Sound 1	15	10	5	8	12	6
Correct class	40	8	13	8	Sound 2	12	8	6	9	10	4
Adjacent class 2	2	2	3	11	Sound 3	10	6	4	7	8	3
Distant class	0	6	5	0	Sound 4	8	5	3	5	6	2

Figure 6.8: Chi Square - Contingency Tables for Sound Morphing items in the survey

After creating the contingency tables, everything is set for the implementation of the chi-square test.

	Statistic Test	p-value	Degrees of Freedom	Interpretation
Harmonic	80.032	6.88e-11	9	Significant
Non Harmonic	74.268	2.2e-12	15	Significant

Figure 6.9: Chi Square Test - Values and Interpretations

From the values obtained through the chi-square test, we can easily conclude that, both in the case of harmonic and non-harmonic sounds, the results are very good. There is a clear significance of the results ($p\text{-value} < 0.01$), and the obtained results can be considered reliable and indicative of a relationship between sound and assigned class.

6.2.3 SMOOTHNESS

Sound smoothness is an extremely important element in sound morphing because it provides a clear and unequivocal measure of how smooth, gradual, unforced, or unpleasant the sound change is.

Below is the image showing the response distributions for each item (Figure 6.10).

I proceed, as with the sound quality assessment, to use statistical tests for result validation. Specifically, we want to verify if there is a significant difference among the various audio clips.

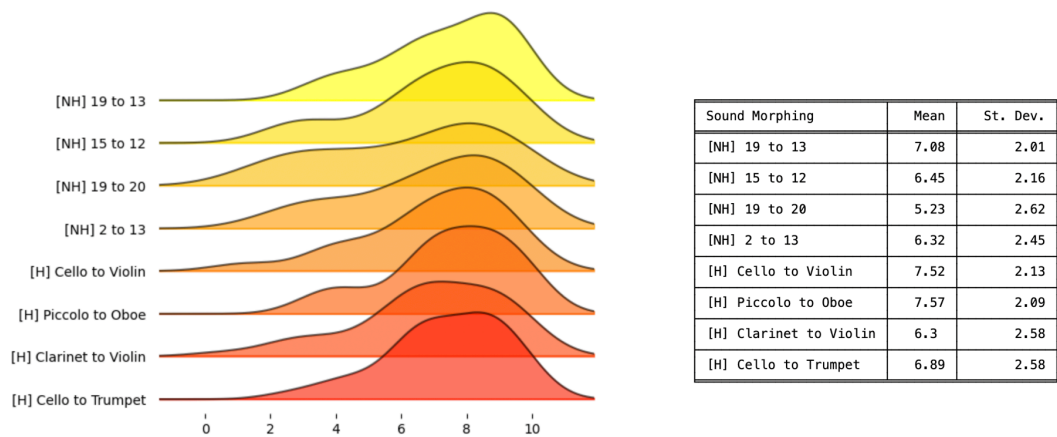


Figure 6.10: Smoothness Evaluation

Therefore, as before, I conduct the Shapiro-Wilk and Levene tests.

Shapiro-Wilk	nh_SM1	nh_SM2	nh_SM3	nh_SM4	h_SM5	h_SM6	h_SM7	h_SM8	Levene Test	Values
Statistic	0.93	0.92	0.93	0.92	0.95	0.92	0.93	0.93	Statistic	1.55
p-value	0.008	0.005	0.011	0.009	0.043	0.004	0.008	0.009	p-value	0.15

Figure 6.11: Verifying ANOVA assumptions: Shapiro-Wilk and Levene Tests

Here, the assumption of homoscedasticity is confirmed, but the assumption of normal distribution of all data is not. We, therefore, proceed with the non-parametric Kruskal-Wallis test.

Kruskal-Wallis	Values
Statistic	6.28
p-value	0.507

Figure 6.12: Kruskal-Wallis Test - Smoothness Scores

It is evident that there is no significant difference among the smoothness results of the various audio clips. It is, therefore, possible to conclude that the smoothness of sound morphing is considered quite good and homogeneous between harmonic and non-harmonic sounds.

Several considerations can be made regarding this:

- On average, sound morphing appears to be smoother in the case of harmonic sounds. This may also be connected to the inherently smoother nature of harmonic sounds, which are generally smoother compared to some of the non-harmonic sounds under examination.

- There is noticeable variability in the responses, with a distribution skewed to the left. The variability is not excessive but remains consistent for all the questions asked, with a standard deviation range from 2.01 to 2.62.

- The worst sound morphing result is for sound 19 to 20, with an average score below passing (5.23). This is not surprising as it represents sound morphing between the sounds that are farthest from each other. This is evident both from the latent space (they are the extreme sounds in the space) and from the perception of sound, which transitions from being very deep and smooth to having significant sound variations over time. To achieve this sound morphing, multiple steps were used, which provided a degree of smoothness but not enough to reach the levels of other sounds that are on average less distant from each other.

- The best sound morphing overall is between cello and violin, followed by piccolo and oboe. This is not coincidental, as they are all harmonic sounds (perceived more suggestively) from the same class (string and woodwind, respectively). Multiple steps were used for this sound morphing, which proved to be sufficient for a smooth transition between instruments.

- The best non-harmonic sound morphing is between sounds 19 and 13. They share a crucial point of continuity and depth in sound. This is what allows this audio clip to be so pleasant, despite only a few steps being used. It can be intuitively inferred that smoothness is widely achieved for all types of sounds.

- The transition between sounds is gradual and pleasant, without any sense of force. The achieved gradualness depends closely on the number of steps used, but especially on the quantity and quality of shared characteristics between the two connected sounds. In the case of sounds perceived as distant from each other, it is still possible to increase the number of steps and the duration of the audio clip to allow for a more pleasant transition.

7

Conclusion

7.1 DISCUSSION

The conclusions of this research address two very different questions:

- Is it possible to generate sound using Variational Autoencoders?
- Can the sound space be explored using Variational Autoencoders?

Regarding the capacity and efficiency of variational autoencoders, there is no doubt that they can be used for sound generation. Even with simple models, without the necessary use of convolutional layers or too many layers with numerous units, sound can be correctly generated. In fact, similarities between generated sound and original sound are evident both analytically and perceptually.

For both harmonic and non-harmonic sounds, sound generation is optimal. However, it depends on the model and, more importantly, on other fundamental aspects: the dimension of the latent space and the complexity of the original sound. Both aspects are connected to the complexity of generation, which requires a strong trade-off between ineffective generation and overly accurate generation, almost tending towards sound distortion. An excessively large latent space, for example, does not always lead to optimal sound production but rather to mechanization, making the sound less fluid and natural.

Among the best models, a model with three hidden layers of 512 hidden units each and ReLU activation function stood out. As for hyperparameters, a learning rate of 0.0005, 500 epochs, and a batch size of 64 proved to be excellent. For non-harmonic sounds, a different latent space dimensionality is preferred depending on the complexity of the sound. For deep and continuous sounds, a dimensionality of 64 dimensions may suffice, while for complex sounds with frequent temporal variations, a dimensionality of 128 or 256 is more suitable.

However, there is little experimentation with more complex models, which could involve the use of more convolutional layers or simply deeper networks. The choice to focus more on sound exploration was made for several reasons: first, the desire to delve deeper into sound exploration; once good generations were achieved, the modeling research was terminated. Second, computational costs played a role. Even for simpler models, training time exceeded several hours for epochs greater than 250. Therefore, it was preferred to steer the study towards the timbral and analytical aspects rather than the algorithmic ones.

From the survey administered, it is possible to affirm that fundamental features for generating good sound, such as sound quality (understood not as sound pleasantness but as the absence of sound disturbances in the audio clip), sound classification (objective for harmonic sounds, with timbres originating from a clear sound source, subjective and perceptual for non-harmonic sounds), and the smoothness and fluidity of sound changes (during sound space exploration in the latent space) were generally well-rated by participants. The results are not excellent but very promising.

From a sound perspective, a significant analysis of the latent space of sounds has been carried out. For both harmonic and non-harmonic sounds, the latent space is a precious source of sound information, from which it is possible to generate sound not only mechanically but, above all, to manipulate it. The strength of this approach lies in its highly controllable sound manipulation capacity. It has been possible to autonomously design well-defined trajectories, both within the same sound class and between different classes. The trajectory generated between different types of sounds (sound morphing) is smooth and pleasant, allowing for sound synthesis that touches many nuances of sound with very simple operations in the latent space, such as weighted averaging.

However, among the disadvantages of this approach, there is limited sound variability, connected to excessive individual control. In fact, creating a specific sound trajectory without considerable hyperparameter control can be unpleasant and static. Finally, the generated sounds themselves do not exhibit great variability but simply faithfully reproduce the characteristics of the reference dataset. In this sense, the model is very faithful and efficient but not very creative

if creativity is understood as the ability to generate something unique, different, and original.

In conclusion, the Variational Autoencoder is an excellent model for music generation, especially due to the level of control and manipulation it allows for sound. Furthermore, the study has shown very interesting aspects from the perspective of sound exploration and timbral musical experiments. We anticipate that this research can contribute to the study of generative AI for various types and complexities of sounds and can serve as a basis for future work, where the timbral element and its exploration are the focus of sound research.

7.2 FUTURE DIRECTIONS

Advanced studies are already underway on the use of Variational Autoencoders for AI sound generation. One such example is the Riffusion model[55], which, given a written prompt, can generate suitable and interesting audio clips. The improvement of implementations for interpolating between different clips is also ongoing, utilizing the latent space of the model, similar to what was done in this study. In this regard, Riffusion could be easily employed for timbral studies like the one conducted in this research.

However, there are several potential future applications of this study, which is expected to contribute to the future developments in sound exploration:

REAL-TIME INFINITE AND CONTROLLED MUSIC GENERATION

It is possible to create an infinite sound trajectory that explores specific regions of the chosen latent space. For example, a generative system could take a prompt with words describing a sonic characteristic (e.g., dark or deep) as input and create a trajectory that traverses the sound's latent space with those characteristics.

AI TOOL FOR MUSICIANS AND SOUND DESIGNERS

Timbral exploration can be made much simpler when starting from precise characteristics. It could be interesting and useful for sound creators to have easy access to a wide range of artificial sound exploration (It refers, for example, to RAVE[55], a tool based on the Riffusion model that utilizes VAE for sound manipulation). By expanding the dataset to include various sound types, it is genuinely possible to create countless artificial musical timbres that can be manipulated with minimal operations, quickly, efficiently, and accessible to all.

OPTIMIZING MUSIC GENERATION SYSTEM FOR EXPERIENCES AND PERCEPTIONS

This development is related to the power of music to influence perceptions. It is well-established that from a neuroscientific perspective, music has significant benefits, particularly in accompanying a specific experience or altering a particular mental state. By connecting a type of sound to a defined sensation, it is possible to generate endless sound trajectories designed to do just that. I refer, in particular, to the SoundFood project, an interesting research project currently ongoing at the Centro di Sonologia Computazionale (CSC) at the Università di Padova, which aims to generate sounds aimed at enhancing the eater's experience. Indeed, thanks to a blend of musical and neuroscientific knowledge, it is possible to create the perfect sound mix to optimize the taste perception of what is being consumed. This is a clear example of how this research can be employed to generate music for experiential purposes.

References

- [1] “Music definition.” [Online]. Available: <https://www.merriam-webster.com/dictionary/music>
- [2] I. Godt, “Music: A practical definition,” *The Musical Times*, vol. 146, no. 1890, pp. 83–88, 2005. [Online]. Available: <http://www.jstor.org/stable/30044071>
- [3] E. Varèse and C. Wen-chung, “The liberation of sound,” *Perspectives of New Music*, vol. 5, no. 1, pp. 11–19, 1966. [Online]. Available: <http://www.jstor.org/stable/832385>
- [4] “Musical composition | Definition, History, Structure, Types, & Facts — britannica.com,” <https://www.britannica.com/art/musical-composition>, [Accessed 09-11-2023].
- [5] P. Doornbusch, “Computer sound synthesis in 1951: The music of csirac,” *Computer Music Journal*, vol. 28, no. 1, pp. 10–25, 2004.
- [6] C. Roads and M. Mathews, “Interview with max mathews,” *Computer Music Journal*, vol. 4, no. 4, pp. 15–22, 1980.
- [7] M. V. Mathews and F. R. Moore, “Groove—a program to compose, store, and edit functions of time,” *Communications of the ACM*, vol. 13, no. 12, pp. 715–721, 1970.
- [8] D. Ellis and M. Beecher, “Fairlight cmi review (emm jun 1981),” *Electronics & Music Maker*, no. Jun 1981, pp. 56–59, 1981.
- [9] P. Westergaard and L. A. Hiller, *Journal of Music Theory*, vol. 3, no. 2, pp. 302–306, 1959. [Online]. Available: <http://www.jstor.org/stable/842857>
- [10] C. Hernandez-Olivan and J. R. Beltran, “Music composition with deep learning: A review,” 2021.

- [11] G. Hadjeres, F. Pachet, and F. Nielsen, “DeepBach: a steerable model for Bach chorales generation,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 1362–1371. [Online]. Available: <https://proceedings.mlr.press/v70/hadjeres17a.html>
- [12] S. Dadman, B. A. Bremdal, B. Bang, and R. Dalmo, “Toward interactive music generation: A position paper,” *IEEE Access*, vol. 10, pp. 125 679–125 695, 2022.
- [13] J.-P. Briot, “From artificial neural networks to deep learning for music generation – history, concepts and trends,” 2020.
- [14] A. Natsiou and S. O’Leary, “Audio representations for deep learning in sound synthesis: A review,” 2022.
- [15] S. Ji, X. Yang, and J. Luo, “A survey on deep learning for symbolic music generation: Representations, algorithms, evaluations, and challenges,” *ACM Computing Surveys*, vol. 56, 05 2023.
- [16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [17] F. Rosenblatt, “The perceptron: a probabilistic model for information storage and organization in the brain.” *Psychological review*, vol. 65, no. 6, p. 386, 1958.
- [18] D. Eck and J. Schmidhuber, “A first look at music composition using lstm recurrent neural networks,” *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale*, vol. 103, no. 4, pp. 48–56, 2002.
- [19] M. Dua, R. Yadav, D. Mamgai, and S. Brodiya, “An improved rnn-lstm based novel approach for sheet music generation,” *Procedia Computer Science*, vol. 171, pp. 465–474, 01 2020.
- [20] M. Li, “A tutorial on backward propagation through time (bptt) in the gated recurrent unit (gru) rnn,” *Dept. Comput. Sci., Univ. British Columbia, Vancouver, BC, Canada, Tech. Rep*, 2016.
- [21] R. Grosse, “Lecture 15: Exploding and vanishing gradients,” *University of Toronto Computer Science*, 2017.

- [22] S. Shahriar, “GAN computers generate arts? A survey on visual arts, music, and literary text generation using generative adversarial network,” *CoRR*, vol. abs/2108.03857, 2021. [Online]. Available: <https://arxiv.org/abs/2108.03857>
- [23] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, “Deep learning techniques for music generation – a survey,” 2019.
- [24] J.-P. Briot and F. Pachet, “Deep learning for music generation: challenges and directions,” *Neural Computing and Applications*, vol. 32, no. 4, pp. 981–993, oct 2018. [Online]. Available: <https://doi.org/10.1007%2F500521-018-3813-6>
- [25] J. A. Sloboda, *Generative processes in music: The psychology of performance, improvisation, and composition*. Clarendon Press/Oxford University Press, 1988.
- [26] R. Roozendaal, “Psychological analysis of musical composition: Composition as design,” *Contemporary music review*, vol. 9, no. 1-2, pp. 311–324, 1993.
- [27] M. Civit, J. Civit-Masot, F. Cuadrado, and M. J. Escalona, “A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends,” *Expert Systems with Applications*, vol. 209, p. 118190, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417422013537>
- [28] G. Loy and C. Abbott, “Programming languages for computer music synthesis, performance, and composition.” *ACM Comput. Surv.*, vol. 17, pp. 235–265, 06 1985.
- [29] C. Ames, “Automated composition in retrospect: 1956–1986,” *Leonardo*, vol. 20, pp. 169 – 185, 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:58264446>
- [30] [Online]. Available: <https://doi.org/10.1613%2Fjair.3908>
- [31] S. Ji, J. Luo, and X. Yang, “A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions,” 2020.
- [32] H. Zhang, L. Xie, and K. Qi, “Implement music generation with gan: A systematic review,” in *2021 International Conference on Computer Engineering and Application (ICCEA)*, 2021, pp. 352–355.

- [33] S. Mangal, R. Modak, and P. Joshi, "LSTM based music generation system," *IARJSET*, vol. 6, no. 5, pp. 47–54, may 2019. [Online]. Available: <https://doi.org/10.17148%2Fiarjset.2019.6508>
- [34] D. Herremans, C.-H. Chuan, and E. Chew, "A functional taxonomy of music generation systems," *ACM Computing Surveys*, vol. 50, no. 5, pp. 1–30, sep 2017. [Online]. Available: <https://doi.org/10.1145%2F3108242>
- [35] F. Roche, T. Hueber, S. Limier, and L. Girin, "Autoencoders for music sound modeling: a comparison of linear, shallow, deep, recurrent and variational models," 2019.
- [36] K. Tatar, D. Bisig, and P. Pasquier, "Latent timbre synthesis: Audio-based variational auto-encoders for music composition and sound design applications," *Neural Computing and Applications*, vol. 33, no. 1, p. 67–84, Oct. 2020. [Online]. Available: <http://dx.doi.org/10.1007/s00521-020-05424-2>
- [37] O. Romani Picas, H. Parra Rodriguez, D. Dabiri, and X. Serra, "Good-sounds dataset," Jun. 2017. [Online]. Available: <https://doi.org/10.5281/zenodo.820937>
- [38] S. McAdams, "Musical timbre perception," *The psychology of music*, pp. 35–67, 2013.
- [39] "Variational Autoencoders - Notes on AI — notesonai.com," <https://notesonai.com/Variational+Autoencoders>, [Accessed 10-11-2023].
- [40] P. Esling, A. Chemla-Romeu-Santos, and A. Bitton, "Generative timbre spaces: regularizing variational auto-encoders with perceptual metrics," 2018.
- [41] S. Hussain, "Understanding the Reparameterization Trick in Variational Autoencoders — snawarhussain.com," <https://snawarhussain.com/blog/genrative%20models/python/vae/tutorial/machine%20learning/Reparameterization-trick-in-VAEs-explained/>, [Accessed 10-11-2023].
- [42] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 11 2008.
- [43] J. Zhai, S. Zhang, J. Chen, and Q. He, "Autoencoder and its various variants," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2018, pp. 415–419.

- [44] “GitHub - ChiaraDelu/CVAE-for-Timbre-Manipulation-and-Sound-Generation: Using Convolutional Variational Autoencoder for the generation and manipulation of both harmonic and non-harmonic timbres. — github.com,” <https://github.com/ChiaraDelu/CVAE-for-Timbre-Manipulation-and-Sound-Generation>, [Accessed 26-11-2023].
- [45] “Papers with Code - Griffin-Lim Algorithm Explained — paperswithcode.com,” <https://paperswithcode.com/method/griffin-lim-algorithm>, [Accessed 15-11-2023].
- [46] F. Chen, M. Paul, M. Lo, and T. Hornbeck, “Music synthesizer,” 2008.
- [47] B. C. J. Moore, *Loudness, Pitch and Timbre*. John Wiley Sons, Ltd, ch. 13, pp. 408–436. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470753477.ch13>
- [48] “REAPER | Audio Production Without Limits — reaper.fm,” <https://www.reaper.fm/>, [Accessed 28-11-2023].
- [49] L. St, S. Wold *et al.*, “Analysis of variance (anova),” *Chemometrics and intelligent laboratory systems*, vol. 6, no. 4, pp. 259–272, 1989.
- [50] P. E. McKight and J. Najab, “Kruskal-wallis test,” *The corsini encyclopedia of psychology*, pp. 1–1, 2010.
- [51] S. S. Shapiro and M. B. Wilk, “An analysis of variance test for normality (complete samples),” *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965. [Online]. Available: <http://www.jstor.org/stable/2333709>
- [52] M. E. O’Neill and K. L. Mathews, “Levene tests of homogeneity of variance for general block and treatment designs,” *Biometrics*, vol. 58, no. 1, pp. 216–224, 2002.
- [53] W. C. Driscoll, “Robustness of the anova and tukey-kramer statistical tests,” *Computers & Industrial Engineering*, vol. 31, no. 1-2, pp. 265–268, 1996.
- [54] A. Dinno, “Nonparametric pairwise multiple comparisons in independent groups using dunn’s test,” *The Stata Journal*, vol. 15, no. 1, pp. 292–300, 2015.
- [55] “GitHub - acids-ircam/RAVE: Official implementation of the RAVE model: a Real-time Audio Variational autoEncoder — github.com,” <https://github.com/acids-ircam/RAVE>, [Accessed 25-11-2023].

Acknowledgments

I would like to thank my Supervisor and Co-Supervisor, Professor Sergio Canazza and Professor Antonio Rodà, for all the knowledge, mentorship, and support they have provided me during the research process and the writing of this thesis. Their patience and availability have guided me serenely throughout this journey, and their passion has greatly inspired me in learning and research, which I have approached with great curiosity and passion.

I would also like to express my gratitude to Filippo Carnovalini and Alessandro Fiordelmondo for their constant presence and dedication. The constructive and stimulating discussions with them have undoubtedly contributed to my academic growth.

Lastly, I sincerely and deeply thank the Centro di Sonologia Computazionale and all the people who are part of it or whom I have encountered there. This thesis came to life at the CSC, a place that has continuously provided me with support, encouragement, and a wealth of knowledge and enthusiasm during these months. I will always treasure the curiosity and passion for music that this place and its people have nurtured within me.