Master Thesis on Sound and Music Computing

Universitat Pompeu Fabra

Dualization Of Rhythm Patterns

Błażej Kotowski

Supervisor: Sergi Jordá

Co-supervisor: Behzad Haki

July 2020

Universitat
Pompeu Fabra
Barcelona

Master Thesis on Sound and Music Computing

Universitat Pompeu Fabra

Dualization Of Rhythm Patterns

Błażej Kotowski

Supervisor: Sergi Jordá

Co-supervisor: Behzad Haki

July 2020

Universitat
Pompeu Fabra
Barcelona

Table of Contents

Acknowledgments

First of all, I would like to thank Sergi Jorda, my supervisor, and the author of the idea of rhythm dualization. Thank you for inspiring me to do the research and for motivating me at all the points of my work.

I would like to thank Xavier Serra, the head of the MTG and Sound and Music Computing master. Thank you for doing a great job being the driving force and a leader of this group of extraordinary people and thank you for giving me the opportunity to study among them.

Thank you to Andrés Lewin-Richter, the first person that has introduced me to the Music Technology Group and encouraged me to consider taking the course that led to writing this thesis.

Also, thank you to Behzad Haki, the co-supervisor of the thesis. Without you, my Machine Learning efforts could have been much more of a struggle.

Not less importantly I would like to thank all the people that I have met in the context of the MTG and SMC master. I am extremely grateful for all the new amazing friendships that I have made and that will remain for the rest of my life. Thanks to you this master was fun and you have also contributed to all the good memories of Barcelona I have made.

Thank you to my parents that always support me in my decisions and that has made taking this course possible.

Thank you, Daphne.

Abstract

This dissertation is a summary of the research on the task of the dualization of rhythm patterns. Rhythm pattern dualization is a transformation of a multi-instrumental rhythm pattern to another pattern composed of maximum two instruments while maintaining coherence and the perceptual essence of the original rhythm. It is a novel task, so comprehensive literature research marrying many disciplines is conducted first. The problem is approached in a multidisciplinary way. Drawing from neurology, cognitive science, and psychology, we assemble solid foundations for tackling the task. We propose two machine learning models built upon the, recently reported by Google Magenta, GrooVAE model for rhythm humanization [34]. The GrooVAE network topology is a combination of Sequence To Sequence Learning and Variational Autoencoder architectures. We treat the task of dualization as a variation of the dimensionality reduction problem. Thus, we intend to achieve the dualized version of rhythm by modifying the network's architecture in a way, that creates a reduced intermediary representation in the process of autoencoding. We propose two models achieving rhythm compression in different ways. In the first, Autoencoders model, we first reduce the dimensionality of the original GrooVAE network, next we collect hidden state vector values from the first layer of the decoding network. Then, we train a cluster of autoencoders to find a latent, two-dimensional representation of these h-vectors, which we treat as a dualized version of the input pattern. In the second, Bottleneck model, we create a two-dimensional bottleneck layer in between the two original layers of the decoder network. We treat this two-dimensional bottleneck representation as a dualized version of the input pattern. Finally, we evaluate our models with listening experiments and report the results.

Keywords: rhythm dualization; dimensionality reduction; rhythm analysis; sequence to sequence; autoencoder; LSTM; latent model;

1. Introduction

1.1. Motivation

Music is a phenomenon that is inherent to human culture. The history of music is as old as the history of humanity itself. Most of us would agree that a world without music would not be the same place. Why? The answer to this question cannot be answered easily, however it appears clear that there must be a reason why it is an unquestionably necessary component of what is called mankind. If one's ambition is to gain a better understanding of the human mind's nature, then the study of such an integral ingredient of it cannot be omitted. Indeed, researchers seem to acknowledge this fact quite well. It was Darwin, in his great dissertation on human nature [35], to first propose the idea that the natural selection for music might be an important factor in the evolution of the human mind. His proposition has quickly become popular and a lot of research was built on its basis since then.

Dominantly apparent across several branches of research on human cognition, the theory of temporal attending, claims an inherently rhythmic nature of perception. Jones and Boltz in [36] propose that mechanisms like attention or memory are subjects to temporal nature and depend on concepts of anticipation and regularity. They build their explanations of the most basic mechanisms of human consciousness on top of ideas well defined in the musical rhythm domain. Mechanisms like grouping, temporal expectancy, or event duration.

All around the world, most of the cultures known to the scientific world have developed some form of music putting a strong emphasis on rhythm, a repetitive, sometimes trance-inducing pulse, marking a temporal organization of musical pieces. The intuition suggests that there has to be some quality that is unique and inherently human in the way we perceive rhythm. Throughout the centuries, science developed a multitude of methods of analyzing rhythm, from different points of view.

What is the essence of rhythm and how do we perceive? There is no simple answer to that. However, every effort on getting us closer, at the same time brings near the comprehension of our minds. We are now living the computational power revolution. Tasks that we didn't imagine resolving in the close past, are now possible to address. Deep neural networks, with just enough data and computational power, are capable of learning abstract concepts and most importantly, teaching us, augmenting our intelligence. This gives hope to gain an understanding of the fields that were hardly accessible before. One such field is human rhythm perception and in this work we will attempt to learn about it, building upon the knowledge gathered so far, but with big help of machine learning models.

1.2. Rhythm pattern dualization

The task of rhythm dualization is defined as a transformation of a multi-instrumental rhythm pattern to another pattern composed of a maximum two instruments while maintaining coherence and the perceptual essence of the original rhythm.

Every multi-instrumental rhythm pattern at its core has its essence. The same rhythm can be played on different sets of instruments, by various players, with distinct expressivity. If we compare multiple performances having the same rhythmic essence at their cores, no matter how they are played, the shared substance will always be conveyed and perceived by the listener. What is this essence then? What is the minimal representation of the rhythm pattern? If a multi-instrumental pattern gets flattened into a monophonic stream of its onsets, some part of the essence, related to its horizontal, temporal structure remains, however it loses the vertical quality related to the interaction in between different voices. The monophonic pattern then fails to convey the essence of any multi-instrumental rhythm. If the lacking quality is related to the interaction or tension between instruments, the way to maintain it could be adding one more voice to the pattern. Therefore, a two-voice pattern might be enough to represent both vertical and horizontal qualities of any rhythm pattern. The justification for such an approach doesn't only come from this simple thought experiment. In [1], Patel argues the importance of motor areas of our brains for building rhythm perception. The roots of rhythmic music reach back to the tradition of drumming, the act of hitting the drum with the use of two hands. Jaki Liebezeit, the founding member of Can, a drummer most known for his virtuosity, at some point in his career has parted with American drum kit played also with legs in favor of a simpler setup where two hands are enough to play [16]. Liebezeit has also invented a system, called E-T, where only two symbols are enough to represent the rhythm accommodating any musical situation.

There is a well-known phenomenon in auditory perception, called subjective rhythmization [37]. It emerges when we are subjected to monotonous auditory stimuli, like the ticking of a clock. Instead of hearing 'tick- tick – tick -tick', we perceive 'tick – tock - tick – tock'. Our brains transform a monophonic sequence into a sequence composed of two parts, so it sounds more interesting, more rhythmic to us.

Witek et al. in [11] explore the idea of rhythm streams, a concept related to the theory of auditory streaming. They find out that adding a single instrumental part to a monophonic rhythm pattern can significantly affect how the rhythm is perceived, whilst adding one more part to a two-instrument pattern only affects the perception in particular instrumentation settings.

All mentioned clues help us reinforce the initial intuition that dualized simplifications of multi-instrumental rhythm patterns could be sufficient to convey both vertical and horizontal qualities of rhythm.

1.3. Objectives

In this dissertation, we mainly want to test the aforementioned idea that the dualized version of the rhythm pattern is sufficient to convey what we perceive as its essence. We will approach the problem of rhythm dualization from different perspectives, looking for the most promising approach of explaining it with a formula capable of executing the job in practice. We will then formulate the approach and implement it, to finally subject it to examination through the listening tests. In the end, we will conclude our journey and formulate the lessons that were learned leaving the topic open for further investigation.

2. State Of The Art

The subject of rhythm dualization is a complex task, not very widespread in the literature, marrying thoughts from multiple disciplines of science. In this work, it was first approached from several different perspectives in order to find a set of methods that would serve as the most promising approach to tackle the problem. We have reviewed ideas from fields of computer science, computational musicology, cognitive science, and neuroscience. In this chapter, we will summarize our findings on the most fundamental concepts that lay the foundations for understanding the task of dualization, along with selected methods that pointed to the direction we have finally decided to take to attempt to resolve the discussed problem.

2.1. Rhythm perception models

2.1.1. Neuronal entrainment

The human brain itself does spontaneously generate activity that can be described as repetitive and rhythmic. When a cluster of neurons, in response to a stimulus, fires its potential, it can be seen as an evoked active state of such a cluster. In other cases, when the same cluster remains idle, its state is described as inactive. There is a tendency for the brain to work in the way that such clusters of neurons switch between these two states in a regular, repetitive way. Neural oscillations are the patterns of this alternation in the central nervous system.

It was shown by a number of studies in neurophysiology, that if a regular stimulus is presented to the patient, their neuronal oscillations in relevant areas of the brain will most likely entrain to the presented signal, which means that they will synchronize their oscillation frequency to the frequency of the signal. In the realm of rhythm, there is an approach to explaining beat and meter perception called frequency tagging. It claims that entrainment of sets of neuronal clusters induces the feeling of beat and grouping of presented rhythmic pattern. In [3] authors presented a 2.4 Hz rhythmic pattern to the listeners and asked them to group events in either binary or ternary way. They asked them to imagine the selected grouping and the underlying beat of the presented music and they have registered the EEG results of their brain activity. The results have succeeded to prove that the beat elicits a sustained periodic EEG response that aligns well with the presented beat frequency. More importantly, however, they have succeeded in showing that there is one more entrained neural oscillation, tuned to the frequency of the imagined meter. The meter frequency, that is not present in the stimulus may still be emphasized by neural oscillations, which suggests that the neural entrainment may reflect the internal representation of the perceived rhythm in the brain (Fig. 1)

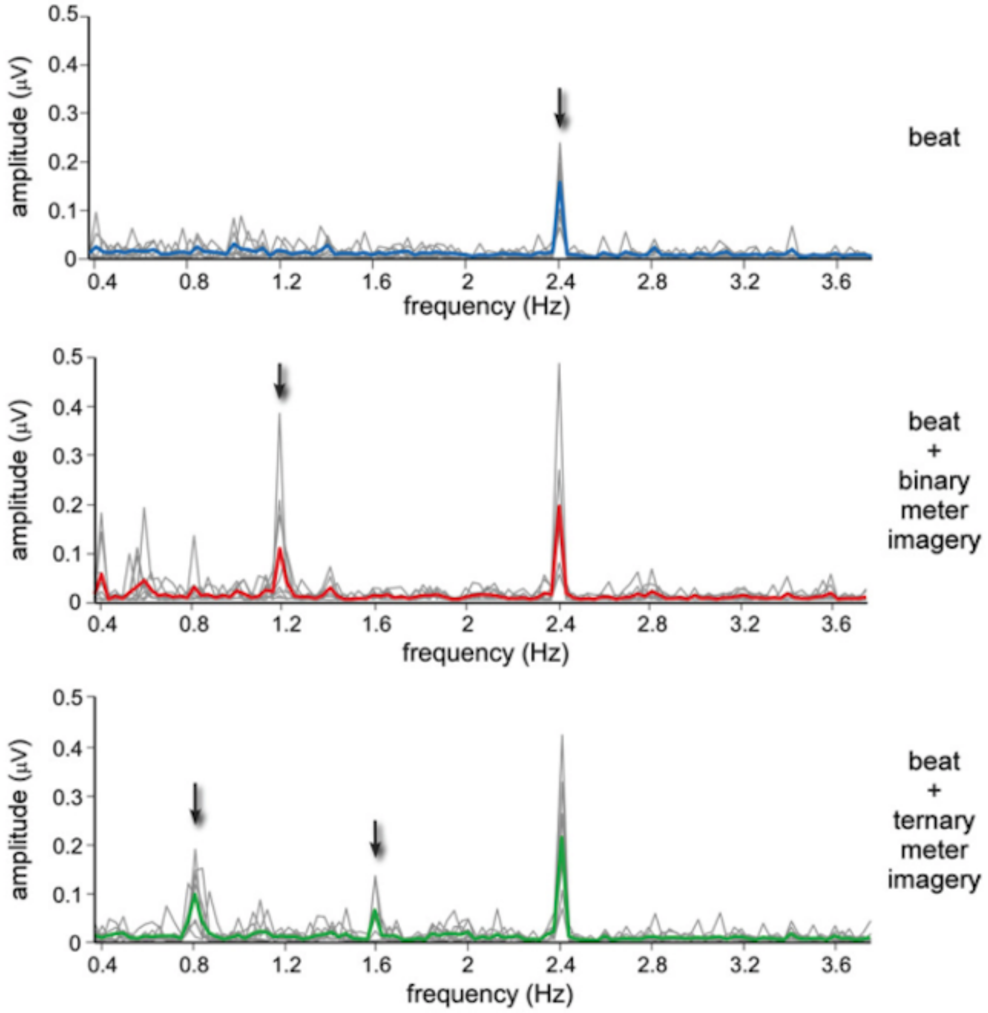


Figure 1: A periodic rhythm elicited a steady-state evoked potential (SS-EP) at the stimulus repetition frequency, and meter imagery elicited subharmonic resonances corresponding to the metric interpretation of this periodic rhythm [4].

Following the aforementioned research, the authors have next experimented with syncopated rhythms, so the kind of rhythms with less energy in the positions related to the predicted beat. They found out that in the EEG results, peaks tuned to the frequency of the absent auditory beat could still be observed. Moreover, peaks present at both beat-related and meter-related frequencies, although not necessarily predominant regarding the acoustic energy, were selectively enhanced by the brain, which was visible in the EEG results [2]. In another study, focused on the level of beat syncopation, authors have experimented with different levels of syncopation to test if/whether the acoustic energy on the frequency of predicted beat is necessary for the participants to entrain to the rhythm. They have shown that even if there is no acoustic event on the predicted beat frequency (as in the case of highly syncopated rhythms), the EEG would still register the peaks at the predicted beat frequency. Authors call this phenomenon “missing pulse” and they compare it to the well-known from the pitch domain phenomenon of “missing fundamental” [4]. This observation suggests that the beat and meter-related peaks in the EEG results are not just a mere reflection of the auditory stimulus, but they rather mirror a complex, internally synthesized.

2.1.2. Action Simulation Auditory Prediction (ASAP)

While the neuronal entrainment theory for the beat perception in the human brain is a very appealing proposition, some researchers argue that such a mechanistic approach cannot be sufficient for a full understanding of the cognitive mechanisms behind it. One of the reasons justifying the doubts regarding the neural synchronization theory comes from the idea of the “pure perception”, which is a situation of listening to the presented piece in the absence of overt movement [1]. Such kind of listening still strongly involves the motor areas of the brain (Fig. 2). This suggests that there must be a significant link between the auditory system and the motor system in the task of building beat perception. Any cognitive theory dealing with the understanding of the mechanisms behind beat perception should apply to the explaining of this connection.

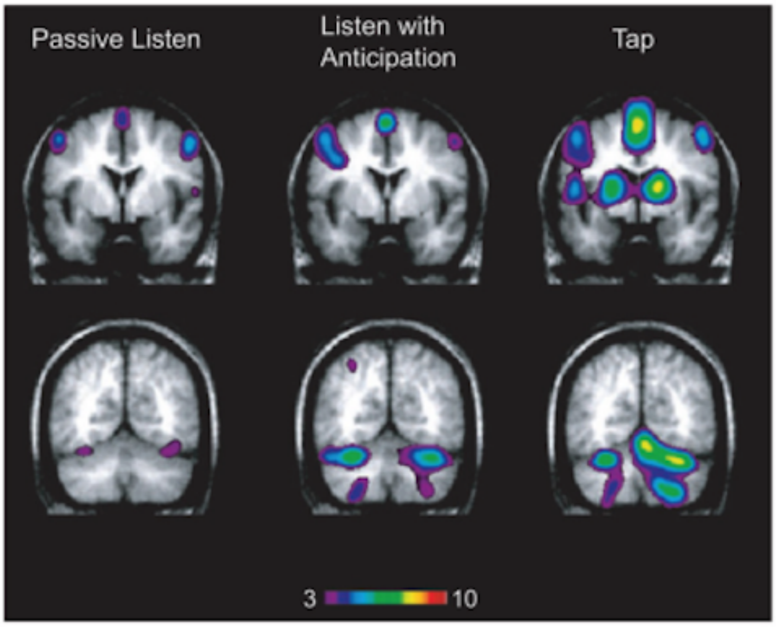


Figure 2: fMRI showed that listening to musical rhythms recruits both auditory and motor areas of the brain even in perception tasks without a motor component (image from [4]).

An important cue on this path is a research on the Rhesus Monkeys' beat perception [5]. Neural entrainment theory would suggest that any brain capable of the synchronization of its neural activity to the rhythmic stimulus would be able to create an internal representation of the beat. However, the aforementioned research showed that monkeys, although capable of neural entrainment, do not show the ability to follow the musical beat. Their brains have exposed similar behavior to the humans', but still, they were not able to follow more than 2 clicks of the metronome. First of all, training for this simple task has taken as much as a whole year, but more importantly, even if they exhibited quite a good ability to synchronize with the mentioned 2 clicks, their tapping response would occur around 100 ms after the stimulus. This is very different from how humans synchronize to the music, as their reaction normally occurs much more precisely, or before the auditory stimulus related to the beat frequency. That would suggest that humans expose a unique ability to anticipate the coming stimulus, instead of merely reacting to it quite in time, like in the case of the monkeys. These results confirm that the entrainment of neural activity is not enough to explain the perhaps much more complex system of beat perception in humans.

A function of heavily rhythm-oriented music is often to elicit a highly synchronized, regular motor response referred to as a dance in the listeners. A number of studies prove that motor areas of the brain play a significant role in beat perception and synchronization (BPS). In [1] Patel and Iversen argue that the motor system is an active contributor to the construction of beat perception in our brains. According to the researchers, in the task of building the beat perception, the human brain depends on the communication between auditory regions and motor planning regions of our brain. We predict the auditory stimuli thanks to the simulation of a periodical body movement. The question here is why would the motor system be involved in the task of auditory stimuli prediction. Authors argue that the motor system is a very well suited mechanism for the generation of neuronal periodicities necessary for perceiving and synchronizing to rhythms, as humans frequently make periodic motions in a very similar time scale to musical one (like while laying steps or swinging arms while walking).

2.1.3. Other beat perception models

Apart from the presented models explaining the synchronization to the musical beat, there are other approaches like the perhaps more outdated error-correction [6][7] or Bayesian model [8][9] approach, which were not analyzed in this dissertation. As the rhythm perception appears to be a very complex issue, which can be studied on a multi-disciplinary level, it is worth acknowledging other possible explanations about it. What is worth noting though, is the common denominator of a significant body of work in the field which is the coupling in between motor planning regions with auditory regions in the efforts of building rhythm perception. This coupling is one of the inspirations behind the belief that rhythm dualization is a significant and justified task.

2.2. Rhythm analysis

2.2.1. Hierarchical interaction

Filter bank segregation

In the audio signal domain, Scheirer proposes passing the audio input through a bank of six bandpass filters, dividing the signal into six bands [10]. In the implemented solution, each of the bands is roughly covering a range of one octave. Next, for each of the sub-bands, the amplitude envelopes are calculated. After that, the corresponding bandpass filters are used to filter noise source and such filtered noise is modulated by previously extracted amplitude envelopes and added to the output signal. Scheirer argues that such output audio is sufficient information for humans to analyze the rhythmic structure of the recording. The filter bank approach and its variations are broadly applied by researchers working on any kind of rhythm-related tasks where the input is an audio waveform. It is comfortable because the error-prone transcription step might be omitted, while still reducing the amount of data significantly. What is worth noting, Scheirer's goal was to build an effective beat tracking / pulse detection algorithm and the proposed psychoacoustic simplification was a step on the way of achieving it. However, beat tracking might be a less complex task than rhythmic pattern dualization, therefore good results achieved with the method in beat tracking task don't it being effective for the dualization.

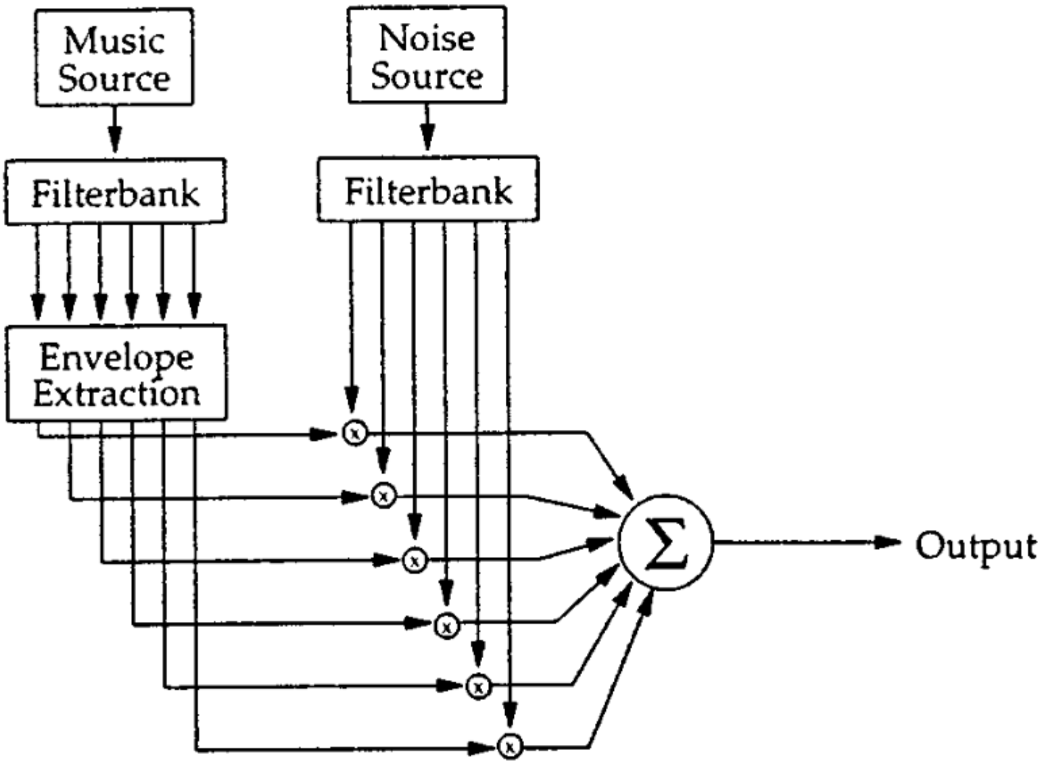


Figure 3: Schematics of Scheirer's beat tracking algorithm, picturing the process of segregating the complex rhythm into six separate streams depending on the frequency (image from [10]).

Rhythm streams

Witek, et al have explored the phenomenon of syncopation in polyphonic drum patterns in [11]. They measured the effect of interaction between monophonic percussive patterns on both perceived rhythm stability and ease of tapping along it. They construct the patterns out of one, two, or three "streams", where the streams are simply monophonic patterns of either bass drum, snare drum, or hi-hat. In the case of single-stream (monophonic) patterns, unsurprisingly the more syncopated patterns were generally considered less stable than the ones where the percussive hits would fall on metrically strong positions. Adding one more percussive track (raising the number of streams to two), would increase the perceived stability of rhythm, but only if the additional stream was a pattern of drums falling on metrically strong, tactus positions, in contrast to the more syncopated, initial pattern. Adding a third stream would significantly increase perceived rhythmic stability only in case when the two initial streams were syncopated bass drum track with hi-hat stream on tactus positions and the added third stream was snare-drum on the tactus positions, reinforcing weaker hi-hat pattern. The result of this experiment may lead to valuable insight, that the timbre of an instrument might have a big effect on the significance it plays in the polyphonic rhythmic pattern. To be more precise, what matters is the relation between the timbres existing in the pattern. Lower frequency instruments like bass drum have a bigger influence on the rhythm perception in comparison to higher-frequency instruments like hi-hats. The confirmation for this theory could also be sought in Hove's, et al. work [12], where authors find out that for the lower frequency sounds, human timing error tolerance is significantly lower than for higher frequencies. These observations suggest that distinct frequency bands shouldn't be treated with the same impact they have on the pattern perception. What is a significant takeaway from Witek's work is the insight that the notion of rhythmic stability often improves when switching from a single stream to a dual-stream pattern. However, when moving from two-stream to three-stream pattern (by adding one more monophonic pattern), the notion of stability improves only in some special cases that have to do with instrument timbres. This lays more foundation for the hypothesis that two streams are sufficient to build the most fundamental representation of rhythm, conveying the syncopation effect.

Voice leading rules

Voice leading is an art of combining individual melodic lines over time. David Huron has built a comprehensive guide on voice leading from a psychological standpoint. In [14] he compiles the most prominent voice-leading rules into an essential list that may be seen in a similar way to the set of axioms in a formal system. Through the experimentation, he has extracted the principles of polyphonic composition, that depend on both temporal and non-temporal sound features. Although Huron's work focuses mostly on melodic composition and not on percussive patterns, the auditory perception domain, which is a foundation on which he builds the rules serves as a common ground for both categories.

2.2.2. Pattern finding methods

Geometry of rhythms

In his book [15], Toussaint models a novel, comprehensive approach to rhythm analysis based on drawing rhythm patters on geometrical figures. This geometry-founded approach makes room for many methods only available when seeing rhythmic patterns from this unusual point of view. He addresses several aspects of rhythm analysis like meter, grouping, syncopation, or rhythm similarity with new analytical tools, based in the language of geometry. He offers a number of perspectives on rhythmic patterns and most of them can be adapted as rules for pattern-finding algorithms or templates for template-matching systems.

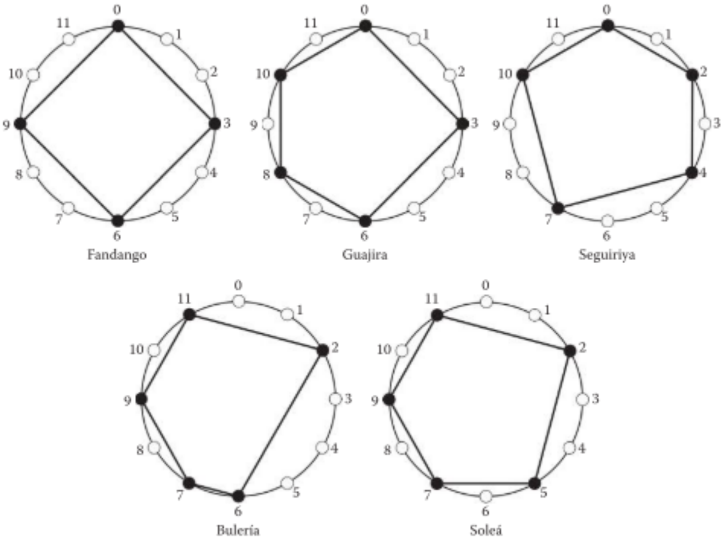


Figure 4: Toussaint proposes a novel approach to rhythm analysis based on drawing rhythmic patterns as geometric figures. The figure pictures five popular rhythms drawn on the plane of the circle – the foundation of Toussaint's method of rhythm analysis.

E-T system

Jaki Liebezeit, the drummer of a legendary rock band and a rhythmic innovator has come up with his own, novel approach to rhythmic generation and analysis as well. What is notable in his E-T system, explained in [16], the rhythms are built with the use of only two, semiotically simple symbols: dash and dot, and the interpretations of such rhythms are played on two drums, with the use of two hands. Liebezeit was an advocate of a view that such a configuration is sufficient for addressing any musical situation and although his views are not strongly backed by science, his experience and novel approach is worth considering. Similarly, as in the case of Toussaint's theory, Libezeit's system serves as a set of rules for pattern-finding and a useful, creative tool for rhythmic analysis.

2.3. Rhythm computing

2.3.1. Pattern representation

In order to process a rhythmic pattern as a computational object, it has to be represented within a data structure, capable of holding all the information necessary to reproduce it in its original form. There are mostly two widely spread ways to approach rhythm in the context of data computation: audio and MIDI representation.

The advantages of representing rhythmic pattern as an audio wave come mostly as a consequence of the sound file's main characteristic: it is a faithful, nearly exact reproduction. Direct recordings contain all the information about the analyzed rhythm, including both temporal information like onset / offset timing or transients and spectral information like pitch or timbre. However, working with audio representation also has its drawbacks. Sound recordings contain not only all the information about the rhythm but also a big part of information noise, that makes it harder to access the relevant data. This leads to the necessity of pre-processing the data to extract the bits of information that are relevant for the methodology to be applied in a specific task. Therefore, working with audio waves may quickly become quite cumbersome when the specificity of a tackled problem calls for a less exact representation of rhythms.

The MIDI representation serves a generalization of a pattern, encapsulating essential information about the event timing and the interaction in between instrumental parts, while omitting the information like details of the timbre, metric division, or tempo of reproduction, effectively leaving these parameters as variables, adjusted depending on the application. Such representation makes working on symbolic information more straightforward as it doesn't require any additional processing to access the rhythmic event-related information.

Both sound and rhythm can emerge only within a time framework and through the repetition, and therefore they both share the sequential nature, however of a different time granularity. When it comes to rhythmic analysis, one is normally interested in the time information on a much bigger time scale than it is provided by audio representation. Therefore in the task of rhythmic dualization, we are opting for the MIDI representation.

2.3.1. Dimensionality reduction

Dimensionality reduction is a process of reducing the amount of data in a dataset by reducing the number of attributes under consideration [18]. Most commonly it is used in data mining simply for compressing the size of big datasets. Through the dimensionality reduction, the data is transformed in a way that it is represented in a much compact form, while closely maintaining the original integrity. Interpreting the task of rhythm dualization, through the prism of dimensionality reduction, what we effectively tackle is a problem of going from a multi-instrumental rhythmic pattern, represented by a bigger number of individual onset tracks, to a two-instrument rhythmic pattern represented but a smaller number (two) of individual onset tracks. Transforming the data represented by multiple tracks to a smaller amount of two tracks fulfills the requirements of this class of problems.

Principal Component Analysis

PCA (also called the Karhunen-Loeve, or K-L), is an orthogonal linear transformation of data that projects the data onto a new coordinate system, where the coordinates are ordered by the significance of data that lies on them. Effectively, on the first coordinate – the first principal component - we find the data of the highest variance, on the second principal component, the data of the second-highest variance, and so on [19]. In the information theory interpretation, this characteristic implies that most of the information about the given dataset after the PCA transformation is contained in the first principal component.

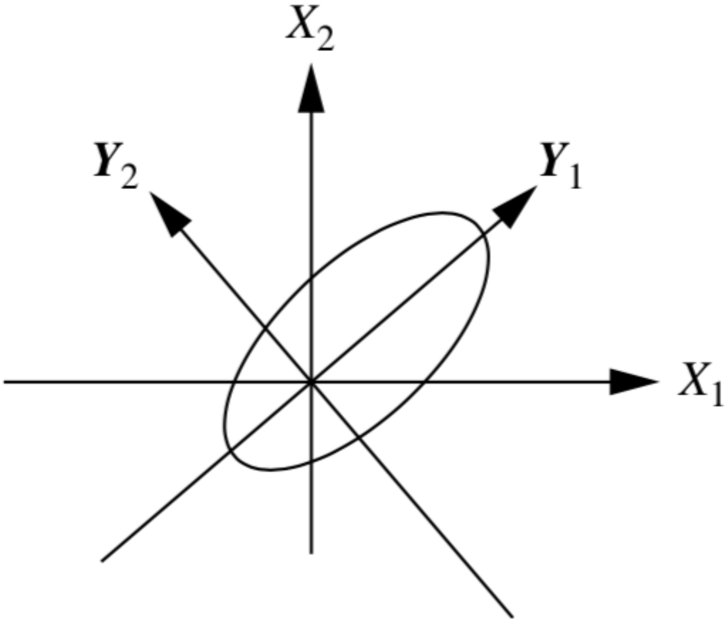


Figure 5: First two principal components Y1 and Y2 for the original data projected as X1 and X2. The new projection helps identify groups or patterns within data [20].

In practice, for a dataset described by n attributes, PCA searches for k n-dimensional orthogonal vectors that best represent the data, where $k \leq n$. Therefore, the n-dimensional data is projected on a smaller k-dimensional space, effectively applying dimensionality reduction. An important characteristic of such projection is the way in which the attributes in the new lower rank space are composed. They are not simply selected from the original data, instead they are linear combinations of original attributes. In this way, PCA is not only an effective tool for reducing dimensionality but also a very informative tool for finding some inter-attribute relationships that may lead to interpretations that have not been thought of before [20].

The ability of PCA to find the most significant bits of data-enabled it to be an algorithm successfully used for tasks from the domain of image processing like image denoising [21][22], or image compression [23] (Fig. 6)

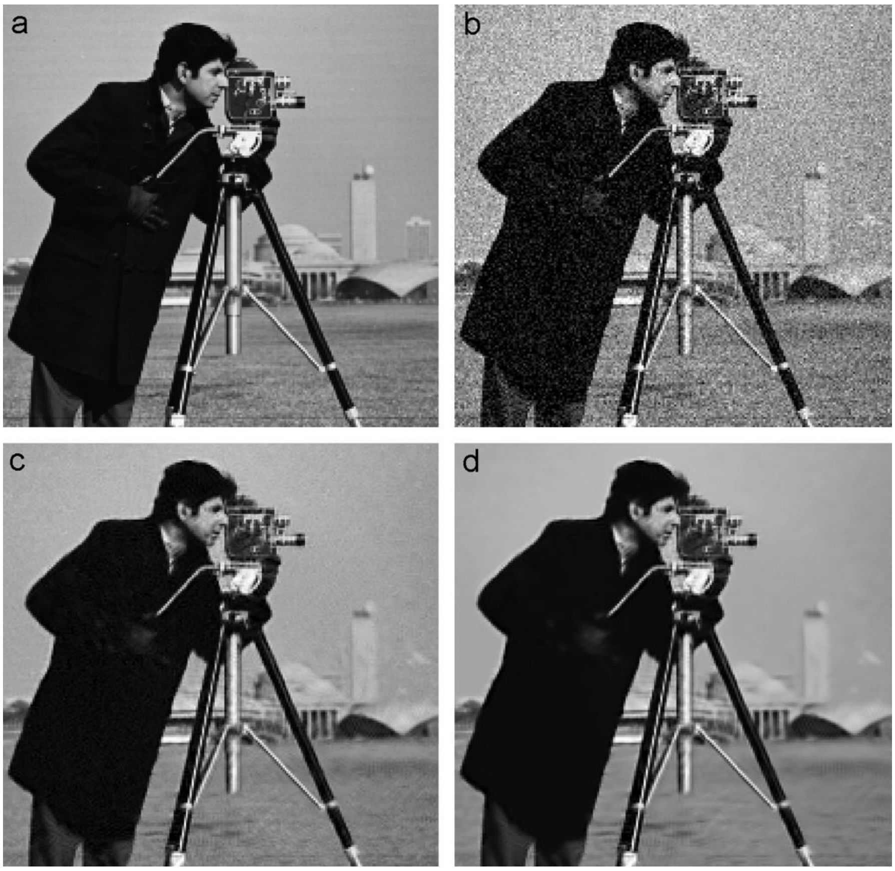


Figure 6: PCA in image denoising task [22]. (a) Original image Cameraman; (b) noisy image (c) denoised image after the first stage of the proposed PCA-based method and (d) denoised image after the second stage.

Autoencoder

The autoencoder (AE) is a neural network architecture designed to learn the identity function of data. The identity function means that through the transformation inside the network, on its output it returns the data in the same form as on its input. The network is designed with a layer in the middle called bottleneck, which has a lower dimensionality than the input data. It can also be seen as two separate networks: encoder and decoder networks coupled together into one autoencoder [25]:

- Encoder network - during training, in this part of the network, it learns the non-linear transformation compressing the data to the bottleneck of lower dimensionality. Such low-dimensional representation produced in the middle layer is called the latent space/representation
- Decoder network – in this part the network decompresses the latent data and transforms it back to its original form.

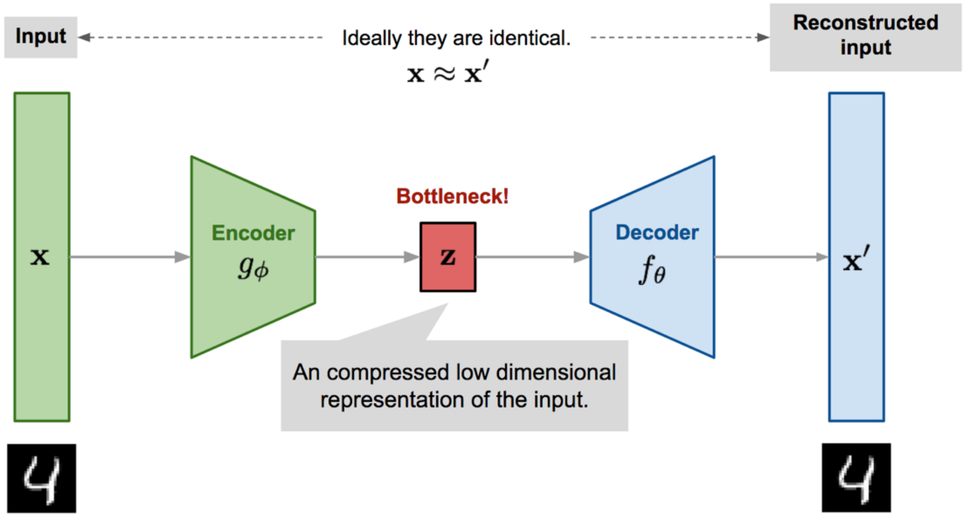

Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
$\mathbf{x}$
Encoder
g_ϕ
Bottleneck!
$\mathbf{z}$
Decoder
f_θ
$\mathbf{x}'$
An compressed low dimensional representation of the input.

Figure 7: Illustration of autoencoder architecture [24].

Autoencoders are not only good for dimensionality reduction, but they are also optimized for data reconstruction. This means that not only do they find more compact, latent representation, but they are also pretty good at reconstructing this compressed version back to its original form.

As in the case of PCA, autoencoders are also suitable for tasks like data compression [25], denoising [26], or feature extraction [27].

2.3.2. Neural networks for sequential data processing

Recurrent Neural Networks (RNN)

Music rhythms build upon the temporal relations between events happening in time, across different voices present in the pattern. Our understanding of music in general is founded in the same temporal nature. In the world of computation, a sequence of events is a term that aligns well with the aforementioned definition of music and therefore can be applied to music sequences (more generally), and the rhythm patterns (more specifically).

A recurrent neural network is a type of architecture that can deal with sequential data for two reasons in particular [28]: it can operate on dynamically-sized input/output data and it can keep track of the past inputs (what it has already seen before).

In most neural net architectures, the input and output dimensionality is fixed to an arbitrary size. In the case of RNNs, there is no restriction on the permanence of input/output dimensionality, as it is designed to work on sequences and those can come in different lengths.

The building block of RNN is a cell that is fed back to itself. As in the feed-forward networks, cells from consecutive layers are connected in the forward direction, in the case of RNN the cells are connected to themselves creating a feedback loop, and for this reason, they can be represented either as a single-layer net with a feedback loop or as multiple layers of regular FFNs with the dynamic number of layers depending on the number of vectors in the sequence.

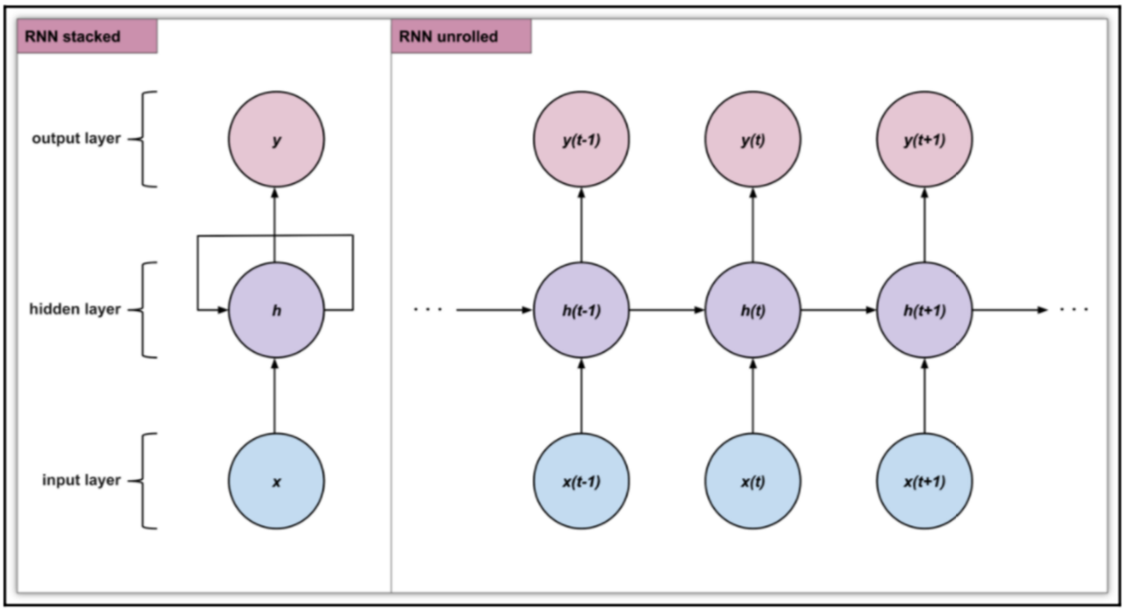


Figure 8: Recurrent Neural Network represented in two forms. On the left: single stacked layer, on the right: unrolled into multiple layers [28].

The most distinct structural characteristic of the RNN network, the feedback connection of RNN cells implies, that by always combining the previous output with the new input, the network holds the information about the previous data that was passed to it. This is what makes this network suitable for working with sequences.

Long-Short Term Memory cells

There is a big body of work addressing various compositional tasks with the means of recurrent neural networks (RNNs), within a part of the research using is focused on automatic music composition. In most of the recent work, the regular neuron units used for the nets are substituted with LSTM units, capable of maintaining significant bits of memory for a longer time-span than classic RNN units suffering from the problem of vanishing gradients.

Research on using RNN networks based on LSTM units is conducted by the Magenta team at Google Brain [29]. They have developed a series of different variations of RNN networks for generating melodies e.g. (Melody RNN) or drums (Drum RNN). These networks, for every given input note, generate a probability distribution of the next possible notes. The style of generated melodies depends on the dataset on which the network was trained. The results are coherent and convincing. Authors have proven that provided RNN networks are capable of learning musically-significant knowledge on the temporal relationships between notes, just by analyzing the provided dataset.

Variational Autoencoder

Another project coming from the Magenta team is MusicVAE [31], a model making use of RNN networks stacked into the autoencoder architecture, capable of understanding latent features of melody. One significant contribution of the work is the use of a special type of autoencoder for sequential data.

The difference between the Variational Autoencoder (VAE) and a regular Autoencoder (AE) lies in the latent space representation. AEs are doing their job really well for finding the lower-dimensional representation of provided data, however, they suffer the continuity in the latent space. This means that if a latent vector that was not generated by the encoder part is provided to the decoder, the network will generate output that is not coherent and musically significant. This is because there are no constraints on the latent space representation. VAEs address the issue by using so-called variational loss. Instead of projecting the input data into a fixed-dimensionality latent representation, variational autoencoders map the input into a distribution [24]. In the case of MusicVAE, it is a unit Gaussian distribution parameterized by μ (mean) and σ (standard variation). To generate the output, all we have to do is to sample a regular latent vector z from the distribution, giving the parameters and pass it to the decoder.

The advantage of using such a representation of latent space is its relative continuity. It means that we can more easily generate new meaningful sequences by randomly sampling latent vectors, as well as perform semantically meaningful tasks like an interpolation in between latent spaces or latent space arithmetic.

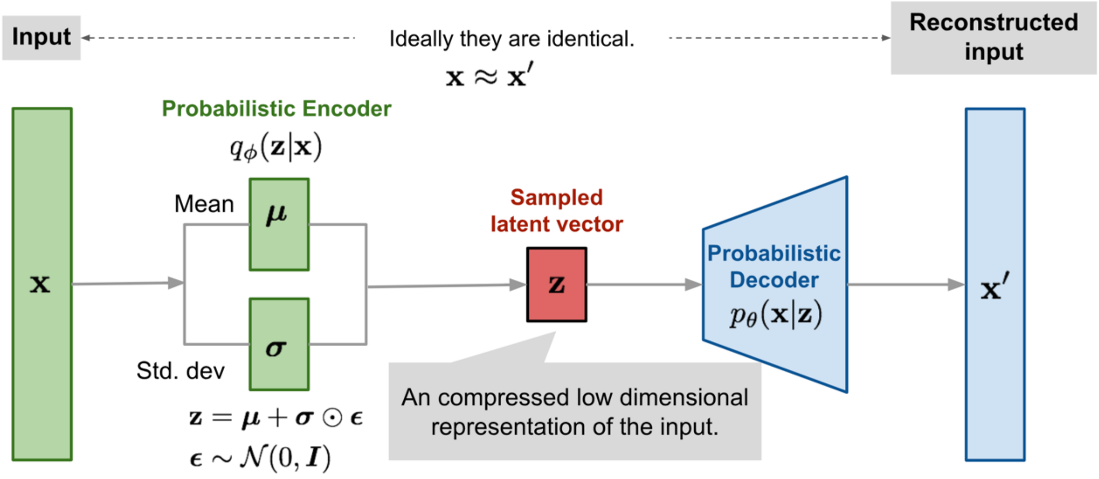

Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
Probabilistic Encoder
$q_\phi(\mathbf{z}|\mathbf{x})$
$\mathbf{x}$
Mean
$\boldsymbol{\mu}$
$\boldsymbol{\sigma}$
Std. dev
$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$
$\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$
Sampled latent vector
$\mathbf{z}$
An compressed low dimensional representation of the input.
Probabilistic Decoder
$p_\theta(\mathbf{x}|\mathbf{z})$
$\mathbf{x}'$

Figure 9: Variational Autoencoder architecture. The latent vector is sampled from a distribution parameterized by mean and standard deviation [24].

Sequence To Sequence Learning

An interesting machine learning architecture coming from the realm of natural language processing is Sequence to Sequence (seq2seq). It builds upon a design similar to the autoencoder. The seq2seq network, first proposed in [32] consists of an encoder network translating the sequential data into a single latent vector (code), that is next decoded by a decoder network to a different sequence, as pictured on Fig. 10.

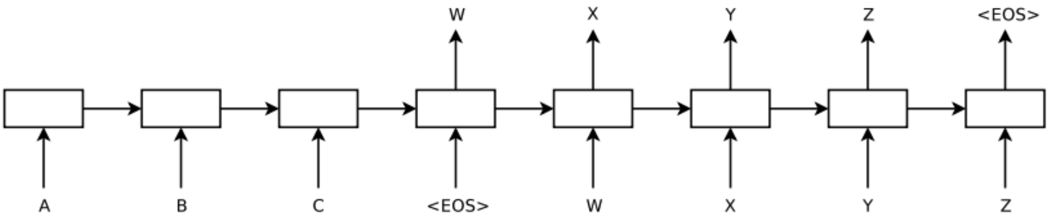


Figure 10: Sequence to sequence (seq2seq) architecture. The sequence A, B, C is translated to W, X, Y, Z.

A big advantage of seq2seq models is the capability of outputting sequences of a different length than these on the input. This characteristic, very useful and commonly applied in natural language translation problems, was proven to be well suitable also for music generation purposes.

Hutchings, in [33], motivated by the idea of separate drum tracks in multi-instrumental drum patterns, “speaking different languages but saying the same thing at the same time” has managed to successfully train a seq2seq network generating a full drum kit recordings being given only the monophonic kick drum track on the input.

In GrooVAE [34], Gillick et al. implement effectively a seq2seq architecture learning to encode the rhythm pattern into a single vector latent representation. Then the latent vector is passed to the decoder that autoregressively decodes it into the output space. They apply the same variational loss term to the model as in the case of MusicVAE turning the whole architecture into a marriage in between Sequence to Sequence with VAE. Gillick et al. work with a dataset of MIDI drum patterns captured in real-time from professional drummers playing MIDI drums interface. This way they have access to patterns containing exact information about both stroke timing and velocity.

Authors train the model to perform three distinct tasks:

- Humanization: given a two-bar, quantized sequence, with no velocity, and no microtiming information, generate a good expressive performance (containing both velocity and microtiming information).
- Infilling: provided a drum performance, missing one pre-selected instrument, add the missing instrument to the midi track.
- Tap2Drum: given only the exact timing information (with microtiming), but no velocity and no multiple tracks, generate a full drum kit expressive performance.

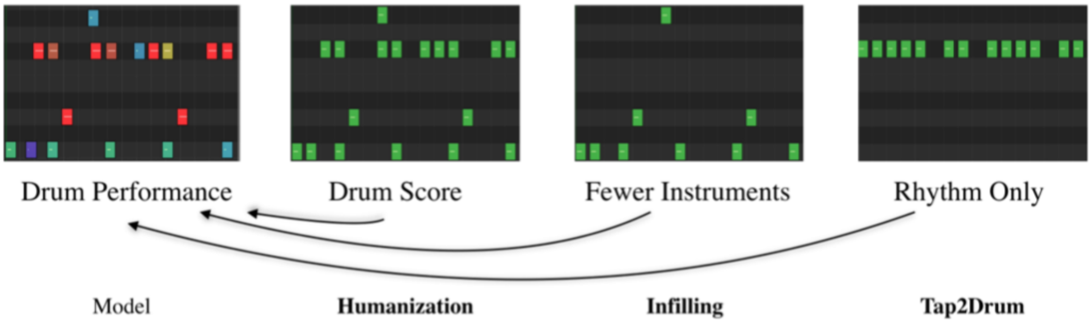


Figure 11: GrooVAE model for "learning to groove" on three distinct sequence transformation tasks. Figure from [34].

In each case, they train the network by modifying the input vectors by omitting bits of the information and conditioning the output.

Then they evaluate the results through the listening tests and they succeed to show that in all the cases the network generates results comparable to expert performances, in some cases even outperforming the original data quality.

3. Methodology

3.1. Introduction

This dissertation aims at finding a practical method for transforming any percussive rhythm pattern into its dualized form. As presented in previous chapters, there are several different approaches that could be utilized in building a final "dualizer". Some of them build upon the knowledge of human cognition and brain function. Others, on the other hand, draw from the domain of data science, or more specifically machine learning. While it is impossible to not take the field of human cognition as a necessary background when tackling a task dealing with musical rhythms, it would be a reckless effort to propose a methodology built in a rather speculative manner, solely centered around the aforementioned field. Instead, we have decided to go with a more pragmatic approach drawing from an existing, well-reviewed body of research in the field of music technology and data science, while keeping in mind the gathered knowledge about human rhythm processing.

The task of dualization can be interpreted as a variation on dimensionality reduction. What is conceptually being done in the process of transformation from a multi-instrumental to two-instrument rhythm pattern is the reduction of its complexity, while maintaining the essence or recognizability.

In the GrooVAE paper [34], authors manage to train an ML model that successfully learns perceptually significant information about the groove of musical rhythms. The architecture of the model draws from concepts of variational autoencoder (VAE) and seq2seq learning, both briefly discussed in the previous chapter. Apart from serving as an architecture successful for generative purposes, VAEs are good at extracting the latent features of the data and effectively in dimensionality reduction. The fact that GrooVAE has the VAE network at its core, along with the fact that it deals with 2-bar midi rhythms and that it is quite successful at its tasks, prompted us to develop our research on top of this model. Our approach, however, is to focus on the internal information of the architecture. Assuming that the model is successful in its original form, we decided to change it as little as possible, to be able to "extract" the dualized rhythm from the data that flows inside of the neural nets.

3.2. Generalized model

For the sake of consistency, the technical details of the tackled task shall be specified clearly at this point. First, we will present the more general view on the designed algorithm, specifying the input data and data processing, along with the algorithm's expected output.

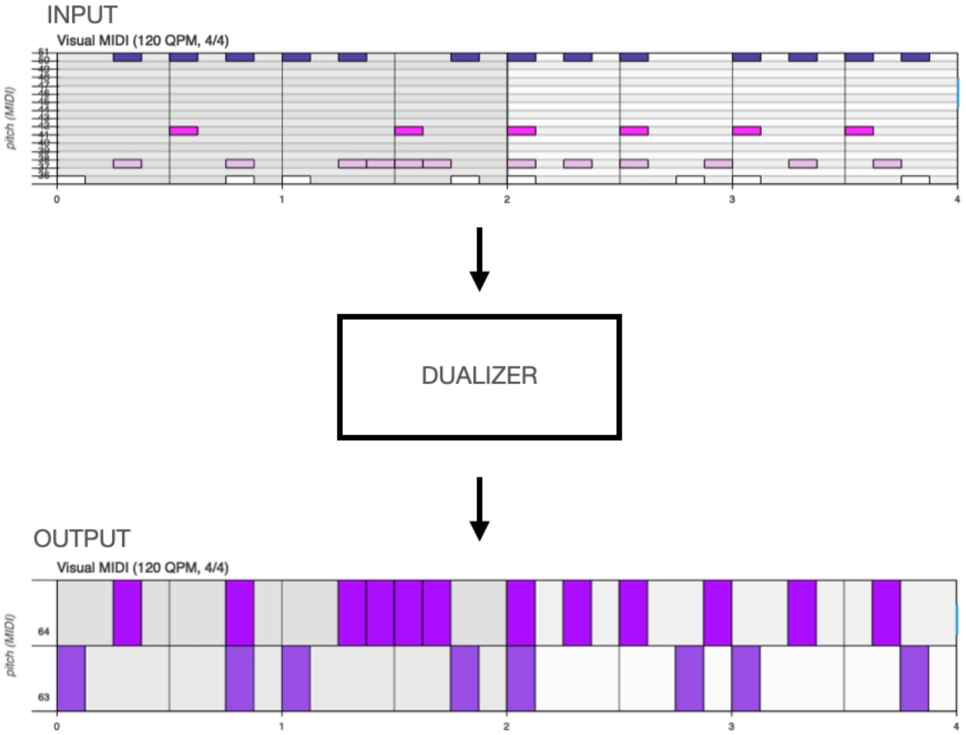
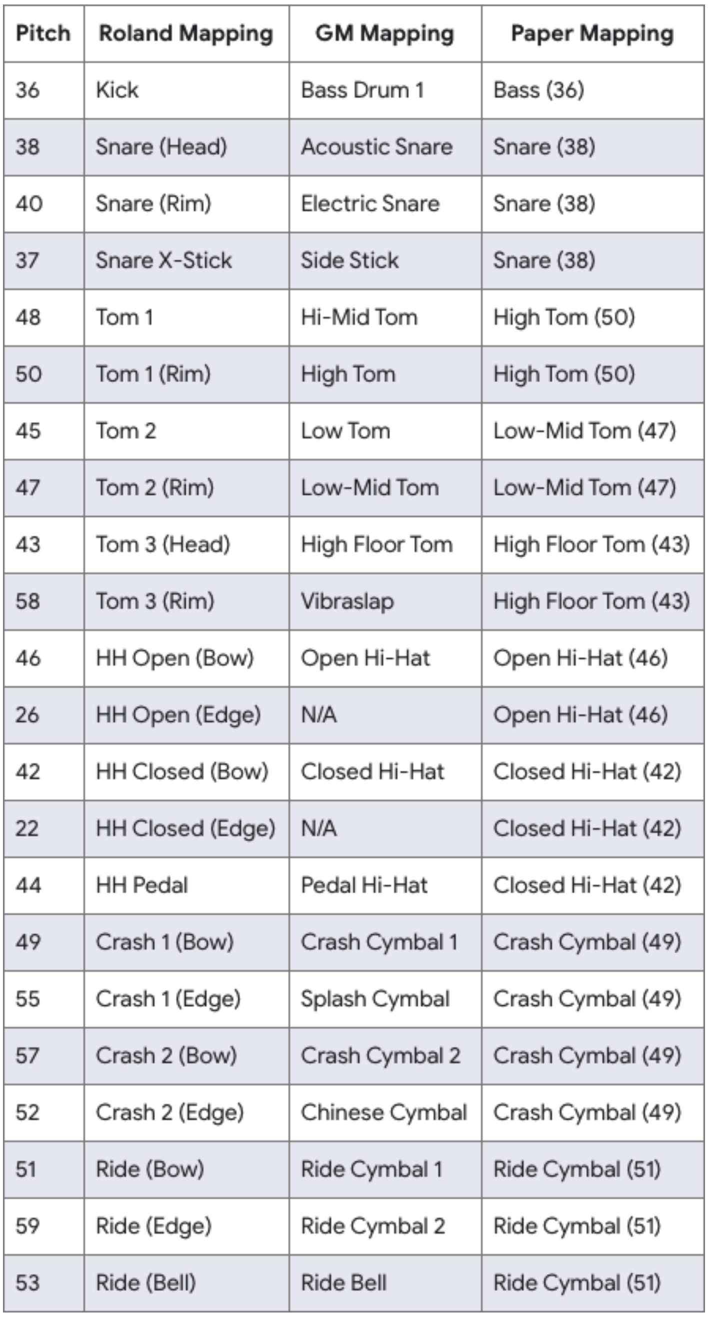


Figure 12. Generalization of the dualization process. The multi-instrumental pattern (INPUT) is transformed with the DUALIZER algorithm to the two-voices dualized pattern (OUTPUT).

Input data and processing

Our models were built with the intention of training on the Groove MIDI Dataset – a dataset of short rhythm patterns generated as an additional output of research on GrooVAE conducted in [34]. Therefore, the constraints on the input of our models were implied by the format of the dataset we decided to use.

Hence, the algorithm is provided with 2-bar rhythm patterns in the MIDI format. The pattern may contain an undefined number of distinct notes, however, all of them should occupy MIDI channel #9, traditionally reserved for the drum tracks. Although an undefined number of distinct MIDI notes is allowed on the input, in the processing stage they are mapped to 9 categories (Fig.12), clustering different timbres of the same instruments together.



Pitch	Roland Mapping	GM Mapping	Paper Mapping
36	Kick	Bass Drum 1	Bass (36)
38	Snare (Head)	Acoustic Snare	Snare (38)
40	Snare (Rim)	Electric Snare	Snare (38)
37	Snare X-Stick	Side Stick	Snare (38)
48	Tom 1	Hi-Mid Tom	High Tom (50)
50	Tom 1 (Rim)	High Tom	High Tom (50)
45	Tom 2	Low Tom	Low-Mid Tom (47)
47	Tom 2 (Rim)	Low-Mid Tom	Low-Mid Tom (47)
43	Tom 3 (Head)	High Floor Tom	High Floor Tom (43)
58	Tom 3 (Rim)	Vibraslap	High Floor Tom (43)
46	HH Open (Bow)	Open Hi-Hat	Open Hi-Hat (46)
26	HH Open (Edge)	N/A	Open Hi-Hat (46)
42	HH Closed (Bow)	Closed Hi-Hat	Closed Hi-Hat (42)
22	HH Closed (Edge)	N/A	Closed Hi-Hat (42)
44	HH Pedal	Pedal Hi-Hat	Closed Hi-Hat (42)
49	Crash 1 (Bow)	Crash Cymbal 1	Crash Cymbal (49)
55	Crash 1 (Edge)	Splash Cymbal	Crash Cymbal (49)
57	Crash 2 (Bow)	Crash Cymbal 2	Crash Cymbal (49)
52	Crash 2 (Edge)	Chinese Cymbal	Crash Cymbal (49)
51	Ride (Bow)	Ride Cymbal 1	Ride Cymbal (51)
59	Ride (Edge)	Ride Cymbal 2	Ride Cymbal (51)
53	Ride (Bell)	Ride Bell	Ride Cymbal (51)

Figure 13. Different timbres of the same instrument are mapped to 9 distinct categories on the processing stage. Mapping from [34].

The pattern may come unquantized, as the MIDI format allows to lay notes freely, out of the grid. However, in the processing stage, the pattern gets quantized to 32, 16th note steps.

Output

The algorithm is expected to generate a quantized, 32-step, 16th note pattern constructed of a maximum of 2 distinct notes. Such a pattern should best fulfill the assumptions of good rhythm pattern dualization, so it should convey the essence of the original, multi-instrumental pattern.

3.3. Dualizing method

As the aforementioned GrooVAE serves as a foundation for all the methods we have developed, it is necessary to bring near the - essential for our research - parts of the model's architecture. We have experimented with various entry points of the architecture to extract the dualized version of the pattern. Fig. 14 visualizes generalized architecture with its original hyper-parameter values. It emphasizes the sections that are important for understanding our research while omitting some of the technical details that this dissertation does not deal with.

The flow of the information in the network is a standard seq2seq data flow. First, the input rhythm is processed and represented in the form of thirty-two tensors (one tensor for each single 16th note step of input rhythm pattern).

$$X = [x_1, x_2, x_3, x_4, \dots, x_{32}]$$

Equation 1. The input is a tensor of 32 single-step vectors.

Each input tensor has 27 dimensions built of 3 concatenated 9-dimensional tensors (one dimension for each possible instrument [see Fig. 13]), representing consecutively the instrument hits (I), the velocity (V), and the offsets (O) of the relevant step.

$$x_i = [I_{i,1}, I_{i,2}, \dots, I_{i,9}, \quad V_{i,1}, V_{i,2}, \dots, V_{i,9}, \quad O_{i,1}, O_{i,2}, \dots, O_{i,9}]$$

Equation 2. Format of GrooVAE single step input tensor.

The referential GrooVAE's task we have chosen to work with is the task of *humanization*. In this task, the network on the input receives only the information about the hits and it is expected to learn how to reproduce the "humanized" pattern containing information about the microtiming (offsets) and velocity on the output. To achieve that, the last 18 values of the step input tensor are set to 0, so at the input, a fully quantized pattern is fed to the network.

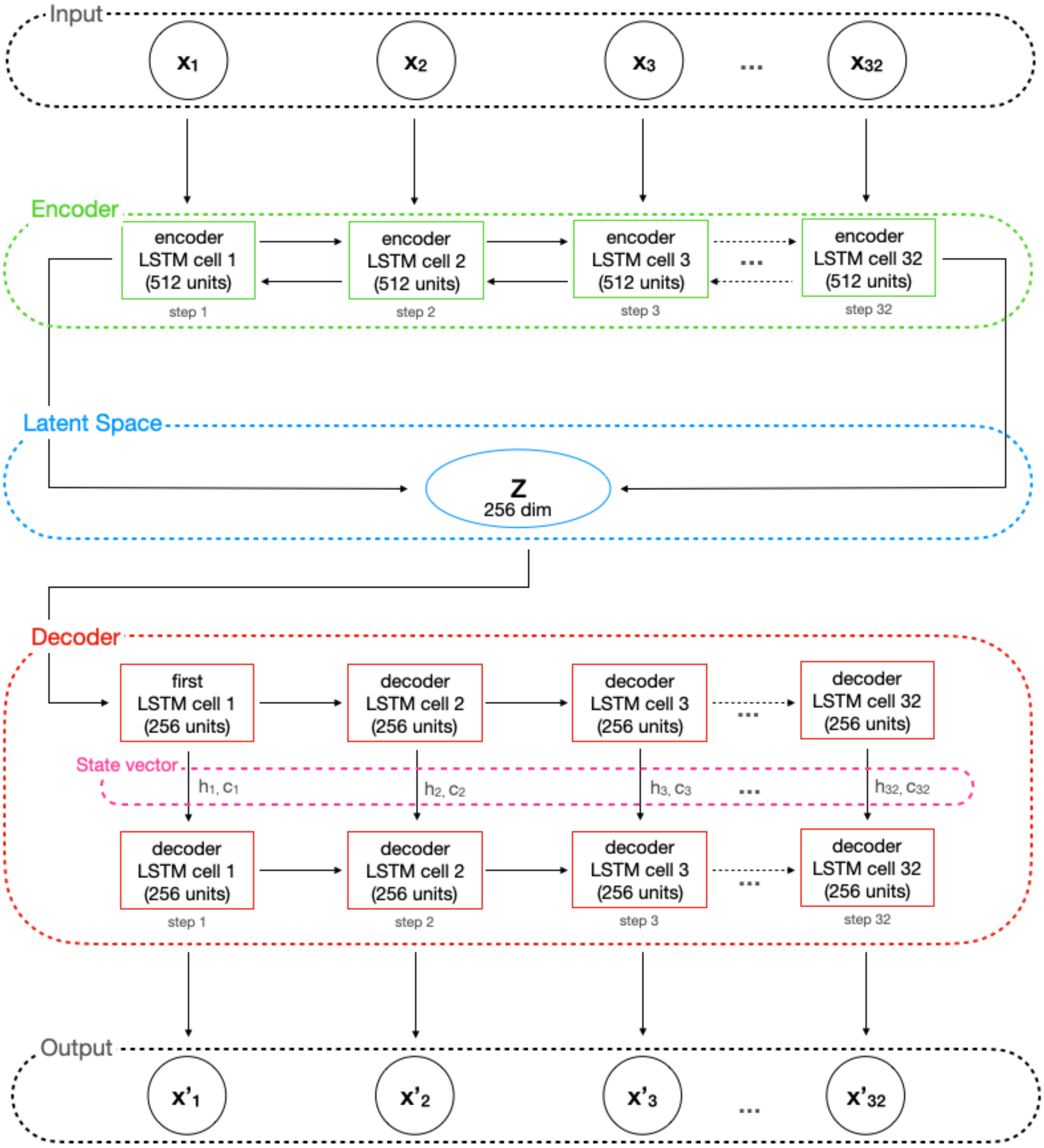

Input
x1
x2
x3
...
x32
Encoder
encoder LSTM cell 1 (512 units)
step 1
encoder LSTM cell 2 (512 units)
step 2
encoder LSTM cell 3 (512 units)
step 3
encoder LSTM cell 32 (512 units)
step 32
Latent Space
Z
256 dim
Decoder
first LSTM cell 1 (256 units)
decoder LSTM cell 2 (256 units)
decoder LSTM cell 3 (256 units)
decoder LSTM cell 32 (256 units)
State vector
h1, c1
h2, c2
h3, c3
h32, c32
decoder LSTM cell 1 (256 units)
step 1
decoder LSTM cell 2 (256 units)
step 2
decoder LSTM cell 3 (256 units)
step 3
decoder LSTM cell 32 (256 units)
step 32
Output
x'1
x'2
x'3
x'32

Figure 14. Visualization of the GrooVAE architecture.

The input tensors are passed, one by one to the bidirectional encoder LSTM network. After all the input is processed, the encoder outputs the latent representation, which is originally 256-dimensional. What is worth emphasizing is that this latent representation contains temporal information about all the pattern. Only such a compact representation is then passed to the decoder part of the architecture.

The decoder consists of two layers of LSTM cells: each 256-dimensional. The latent space serves as the initial state of the first LSTM layer. The first unit updates its state and passes the new state to the second layer of LSTM units. The second (final) layer processes the state vector to output a step of the output pattern. It is then recurrently fed back to itself so it reproduces the pattern step by step until the full pattern is produced (so 32 times in our case). The *state tensor* in the case of the LSTM unit consists of its hidden state *h* and its cell memory *c.* The recurrent nature of LSTM units implies that the hidden state holds the information about all the previously processed inputs, so the hidden state of the last LSTM cell holds the information about all the steps in the pattern. This is also how the latent space is encoded on the encoder side of the network: it is simply the hidden state of the last LSTM cell.

To extract the dualized version of our original pattern, we have decided to experiment with the two most promising parts of the network: its latent space *(z)* and the hidden states of the first LSTM layer on the decoder side. This approach has resulted in three different dualizing methods that are described in the following sections.

3.3.1. Singular Vector Decomposition on the latent space

The first approach we have tested was focused on the latent space (*z*) of the original model. The latent representation in this model is a single, 256-dimensional vector of real values in the range of *[-1, 1].* One of the characteristics of Variational Autoencoder networks is the continuity of the latent space. It means that the network is capable of decoding latent vectors that it has not seen before. In the case of images, it is a well-documented characteristic of autoencoders, that compression of the latent space may lead to perceptually relevant compression of the original image. These two cues have prompted us to conduct the simple experiment of compressing the latent representation of rhythm patterns using the Singular Vector Decomposition (SVD) method.

SVD is one of the ways to conduct a Principal Component Analysis (PCA), described more in detail in the Dimensionality Reduction part of the State of The Art chapter. Simply put, SVD allows us to find the most valuable components of the data matrix. Using this transformation, we can find out the most essential bits of information in the dataset. We have therefore collected a dataset of latent representations of all the rhythm patterns in the Groove MIDI dataset. This dataset was then transformed using SVD. Next, we have gradually reduced the number of first, most informative data components taken into account, while observing the patterns decoded from such compressed latent representations.

We have come to the conclusion that the compression of latent space in the case of the GrooVAE model does not lead to perceptual compression of the pattern. The fewer components we would use for the compression, the further the output pattern would fall from the original. All the patterns during compression tended to converge to a similar "minimal" form which seemed to convey the "average" of all the patterns in the dataset, rather than a more essential version of a particular pattern.

3.3.2. Autoencoding the h-vectors

As a follow-up improvement of the SVD method described in the previous section we have decided to focus on the h-vectors returned on the output of the first layer of the decoding LSTM network. The state vector of the first layer of the decoder holds the information about the output pattern on each time-step separately. Therefore, it can be treated as a kind of intermediary latent representation of the full pattern. If the previous approach of analyzing the single latent space vector has suffered from the lack of information about the temporal structure of the pattern, looking at hidden state vectors could address this issue.

An *h*-vector at *i*-th time step contains compressed information about all the hits, velocities, and offsets at a time point, as well as all the information about the hits, velocities, and offsets about all the previous time steps necessary for deciding on the current output. The dimensionality of this information is defined by the number of LSTM units in a cell of the decoder, and this number is an adjustable parameter.

If the network successfully produces the momentary output tensor depending on the current state of the LSTM unit, it means that this state contains enough information to be able to produce a single output step. With this assumption, we have decided to treat these vectors separately and to build the pattern looking at them step by step. In order to achieve a two-instrument output track, we have implemented a simple autoencoder network responsible for representing the hidden state vectors in a more compact, 2-dimensional form (Fig. 15).

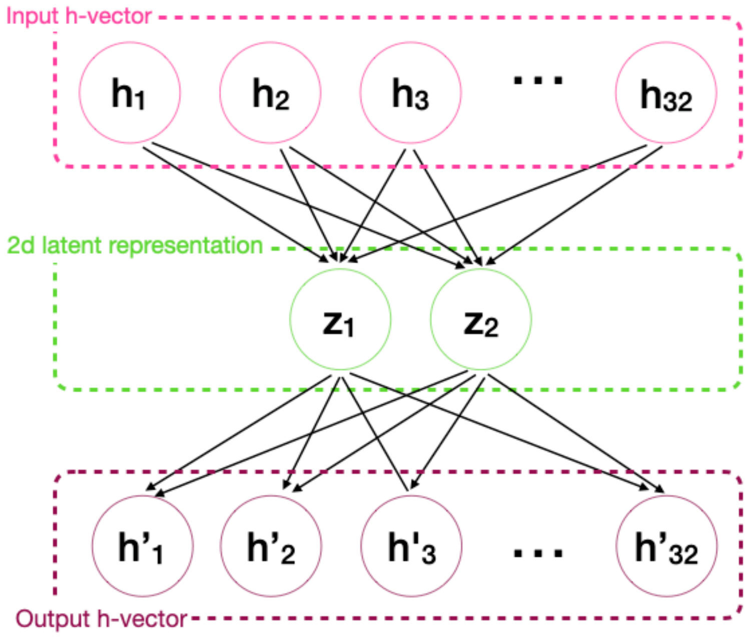


Figure 15. Single-step H-vector autoencoder architecture. The middle layer (latent representation) serves as a dualized representation at a time-step.

Adjusting dimensionality of GrooVAE

As the bottleneck in the designed network architecture is very narrow, the dimensionality of the input should be low enough for the autoencoder to be capable of reconstructing the vector without losing the essential information. For that reason, the dimensions of the GrooVAE needed to be adjusted and the model was trained again using the author's original procedure [34]. We have followed a grid search-like approach to find the acceptable, minimal dimensionality of the decoder's first LSTM layer while maintaining a similar effectivity of the network. We were reducing the decoder's dimensionality along with the latent space and encoder's dimensionality while observing the network's validation loss value. We have found that we could downsize the network quite significantly without losing much on the validation loss. Finally, we have maintained 512 dimensions for the encoder's LSTM units, while reducing the latent representation down to 32 dimensions, along with 32 dimensions for the first layer of the decoder and 256 dimensions for the second layer. With these parameters, the decoder would output 32-dimensional state vectors, which we have used for the next step.

Training autoencoders network

With such reparameterized and newly trained GrooVAE, we have generated the dataset of H-vectors in respect to the time-step. We have used 80% of the whole Groove MIDI dataset. We have passed all the rhythm patterns through the network to collect the H-vectors. Next, we have trained 32 separate instances of AE networks (Fig. 16).

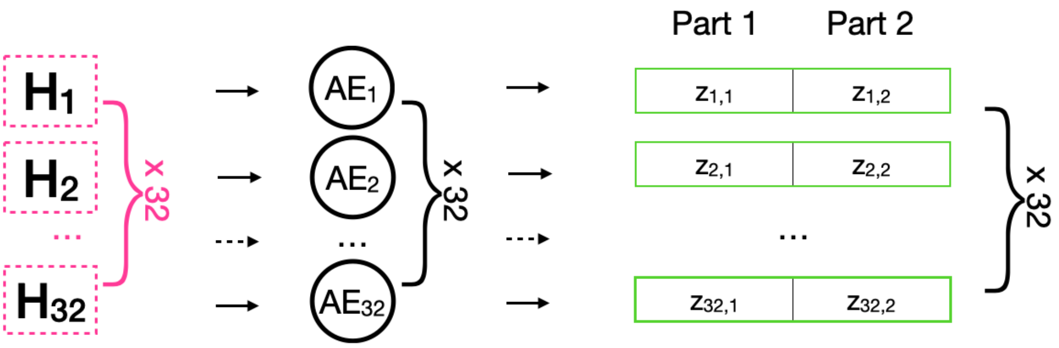


Figure 16. H-vectors autoencoders network.

Interpreting the results

To generate the dualized version of the rhythm pattern, the original pattern is encoded with GrooVAE architecture, while H-vectors from the first decoder's layer are collected. Then they are encoded with the autoencoders network. The bottleneck layers of the parallel autoencoders network are put together to form a tensor *D* of *[32, 2]* dimensions, where 32 is the number of steps and 2 is the number of output tracks (one track for each instrument).

$$D = \left(z_{1,1} \cdots z_{32,1}\, z_{1,2} \cdots z_{32,2} \right),\ z_{i,j} \in R : 0 \leq z_{i,j} \leq 1$$

Equation 3. Dualization is composed of autoencoders' bottleneck values.

The activation function of the bottleneck layer in the autoencoder network is set to be sigmoid, therefore the values that it generates fall in the range of *[0, 1]*. This allows us to interpret them as a confidence of a hit presence at a time step. The most direct way to translate this representation to the rhythm track is to map the range of *[0, 1]* to *[0, 127]* which represents the range of velocity in the MIDI standard.

3.3.3. Creating a bottleneck in the GrooVAE decoder

The autoencoding approach has proven to not be very effective for a couple of reasons discussed more in detail in the Discussion section. One of them is both networks being trained in separation. This characteristic causes the weights of the original GrooVAE network not to be adjusted according to the reproduction error of the autoencoder network. Effectively, the network does not learn that the output of the decoder’s first layer should be prepared for the autoencoding step and the autoencoder ends up not being capable to learn the 2-dimensional representation very well.

One possible solution to this problem is to plug the autoencoding step into the original network. The most straightforward way to do it is to create a symmetric bottleneck after the first layer of the decoder (Fig. 17). Effectively we end up with a 4-layer LSTM decoder. The first and last layers are the original layers (downsized following the procedure from the previous section). The layers in between are the 2-dimensional bottleneck right after the first layer and layer of the same size as the first one right after the bottleneck, to create a standard autoencoder symmetry.

Such a modified GrooVAE model was trained again using the original training procedure.

Interpreting the results

To read the dualized pattern, the original pattern must be processed by the modified GrooVAE architecture. The state vectors are collected in the process of decoding the layer. In this model, we have access to both hidden state vectors and cell memory vectors. Both of them can be used to construct the two-instrument rhythm. The same as in the case of the autoencoders model, when we put the states together, we end up with a tensor of size *[32, 2]*, where *32* is the number of steps and *2* is the dimensionality of the bottleneck.

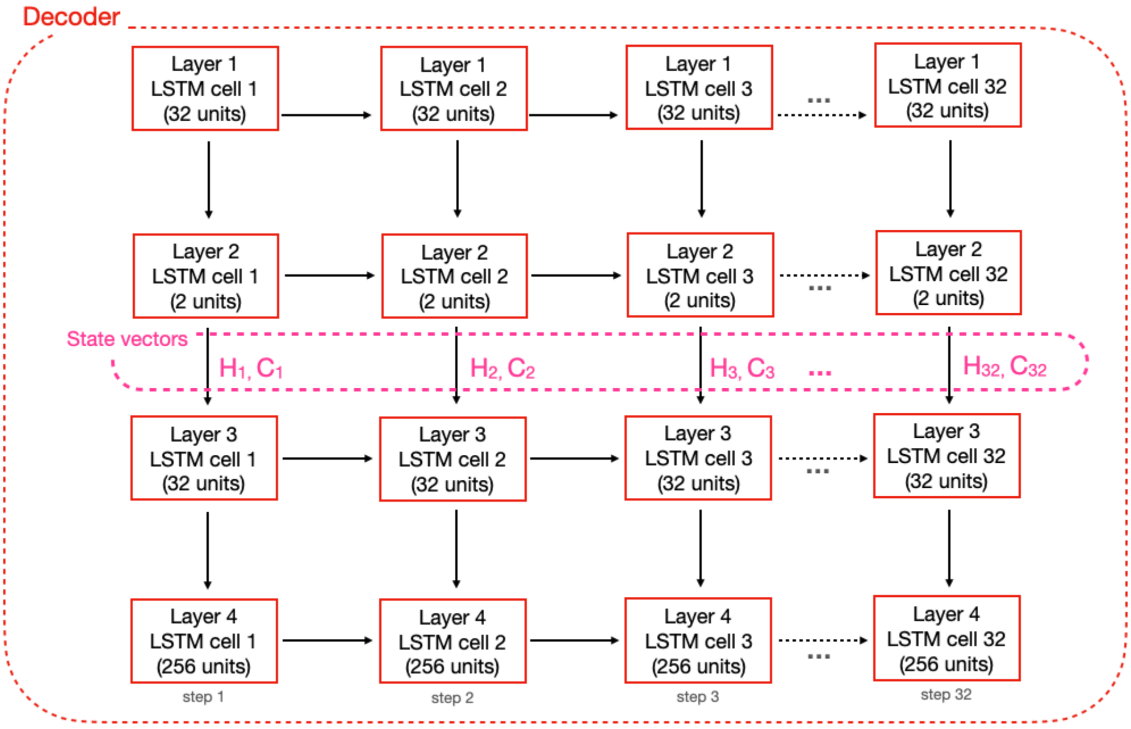

Decoder
Layer 1 LSTM cell 1 (32 units)
Layer 1 LSTM cell 2 (32 units)
Layer 1 LSTM cell 3 (32 units)
...
Layer 1 LSTM cell 32 (32 units)
Layer 2 LSTM cell 1 (2 units)
Layer 2 LSTM cell 2 (2 units)
Layer 2 LSTM cell 3 (2 units)
...
Layer 2 LSTM cell 32 (2 units)
State vectors
H1, C1
H2, C2
H3, C3
...
H32, C32
Layer 3 LSTM cell 1 (32 units)
Layer 3 LSTM cell 2 (32 units)
Layer 3 LSTM cell 3 (32 units)
...
Layer 3 LSTM cell 32 (32 units)
Layer 4 LSTM cell 1 (256 units)
Layer 4 LSTM cell 2 (256 units)
Layer 4 LSTM cell 3 (256 units)
...
Layer 4 LSTM cell 32 (256 units)
step 1
step 2
step 3
step 32

Figure 17. The 2-dimensional bottleneck in the decoder part of the GrooVAE

The C (cell memory) values are activated by *sigmoid* function, the same as in the case of the latent space in autoencoders architecture from the previous section. Therefore, the same process as in the previous section is used for reading these values. In the case of h-vectors, they are activated within LSTM cells with *tanh* function, therefore they're range differs (See Equation 4).

$$h_{i,j} \in R: \ -1 \leq h_{i,j} \leq 1$$

Equation 4. The range of h-vector values within the GrooVAE decoder network.

To be able to follow the same procedure for reading the h-vectors, the normalization must be applied to the values so that they fall into the range of [0,1], like in the previous cases. A simple normalization transformation is applied to all the values in the tensor (Equation 5). After that, they are read as hit presence confidence, like in the case of the previously described dualizer method.

$$h_{i,j} = \frac{h_{i,j} - min\ (h)}{(h)\ \ - min\ (h)}$$

Equation 5. H-vector values normalization.

4. Evaluation

In this chapter, we will discuss the design of the listening experiments we used to evaluate the outcome of dualizer models.

We have decided to evaluate the two last models:

- The Autoencoders model described in section 3.3.2, addressed in the rest of the work as **AE**.
- Decoder's bottleneck described in section 3.3.3, addressed in the rest of the work as **Bottleneck**.

The SVD model has failed the preliminary tests, therefore evaluating it through the listening experiments would not bring any additional value to the work.

We have conducted two independent listening experiments with separate participants and different types of core questions. Hence, this chapter will be split into the two main sections reporting on them. Both experiments were addressing both models but addressing their performance but in different aspects.

In each of the sections we will elaborate on the participants of the experiment, then we will bring closer the design, to finally conclude with the results and a brief commentary.

Although the core listening exercise was different for each experiment, in both cases we have presented the same questions to learn some relevant facts about our audience. Before starting the experiment, they were asked to answer a set of questions about their age, and musical experience. The results of these questions will be shared in the Participants section. Participants were also presented with a short explanation of the concept of dualization task.

4.1. Choose your preferred dualization (CPD)

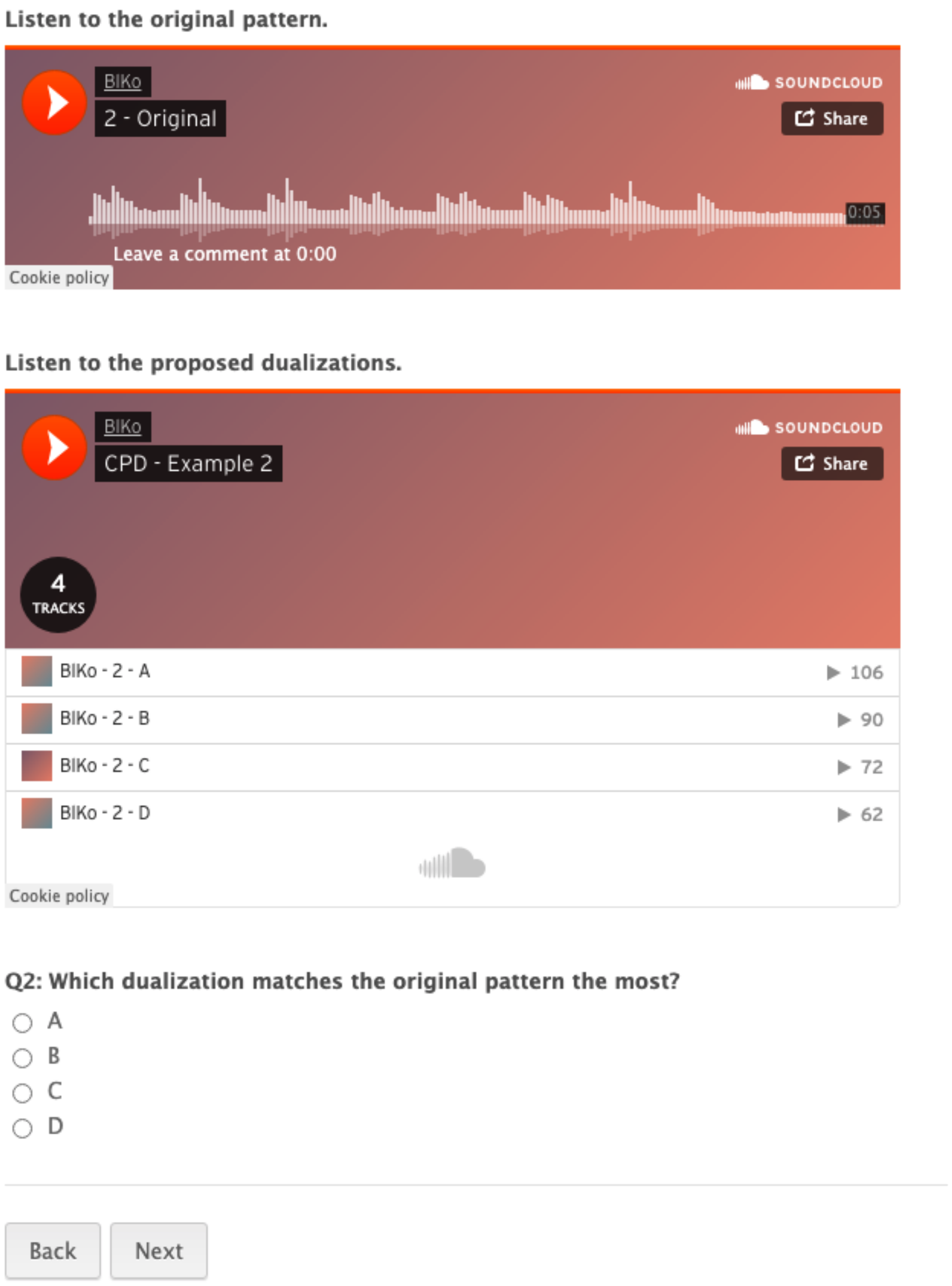
In this experiment we intended to learn about listeners’ preference between the outcome dualizer models, along with a simple Kick & Hihat baseline model and a random dualizer, both described below.

4.1.1. Experiment Design

The participants had to listen to a single original pattern. Next, they were presented 4 possible dualizations: two of them generated by our model, one generated by a simple baseline algorithm, and one random. They had to choose the pattern that they find the closest to the original one.

The experiment was prepared in the following steps:

1. We have randomly selected 20 two-bar patterns from the Groove MIDI Dataset.
2. Next, we have used the two aforementioned dualizing models: AE and Bottleneck to process the selected patterns in order to generate their dualized versions.
3. We have also used two simple baseline models
   a. Kick & Hihat, generating the dualization by selecting only the kick and hi-hat tracks from the original pattern.
   b. Random model generating the dualized pattern randomly: for each of 32 steps, for each of two tracks, it would randomly select a number in between [0, 1] that would directly translate to the velocity of the played note.
4. Finally, we have presented the original pattern, along with all 4 dualized patterns to the participants, asking them to select a single dualization, that they find the closest to the original one (Fig. 18).



Listen to the original pattern.

BIKo
2 - Original

Listen to the proposed dualizations.

BIKo
CPD - Example 2

BIKo - 2 - A ▶ 106
BIKo - 2 - B ▶ 90
BIKo - 2 - C ▶ 72
BIKo - 2 - D ▶ 62

Q2: Which dualization matches the original pattern the most?

○ A
○ B
○ C
○ D

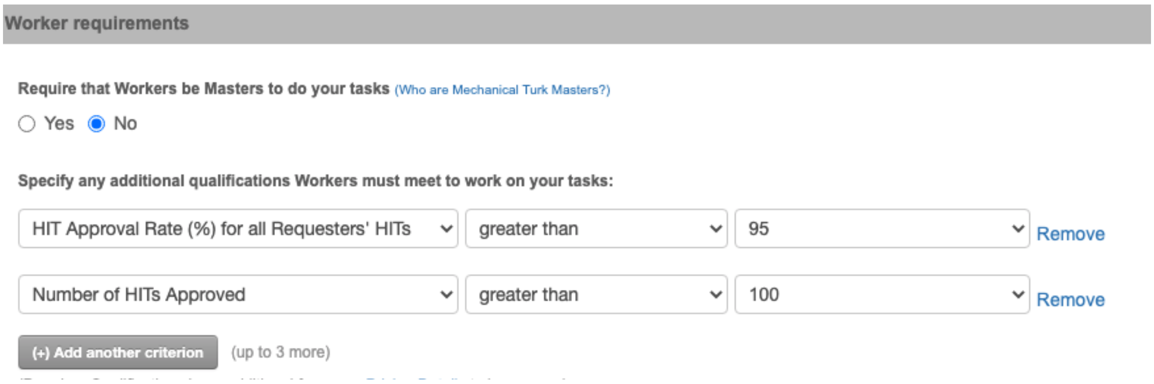
Back Next

Figure 18. Online questionnaire for the CPD experiment. [1]

[1] Available at https://form.jotform.com/202303133301028.

4.1.2. Participants

We have used Amazon Mechanical Turk (MTurk[2]) platform to find the participants for the questionnaire. The platform allows setting the requirements for the participants. We have set the filters to only allow relatively experienced "workers" that have participated on the platform before and fulfilled the tasks with success (See Fig. 19).



Worker requirements

Require that Workers be Masters to do your tasks (Who are Mechanical Turk Masters?)

○ Yes ● No

Specify any additional qualifications Workers must meet to work on your tasks:

HIT Approval Rate (%) for all Requesters' HITs | greater than | 95 | Remove

Number of HITs Approved | greater than | 100 | Remove

Figure 19. MTurk worker requirements.

We have collected the questionnaires from 52 participants.

Below we present the plot picturing the age distribution of the participants (see Fig. 20). Along with it, we show the distribution of the participants that play some instrument or specifically the drums (see Fig. 21). Finally, we asked how much time per week do they spend on playing the instrument in case they have answered that the instrument they play is drums (see Fig. 22).

[2] Available at https://www.mturk.com/

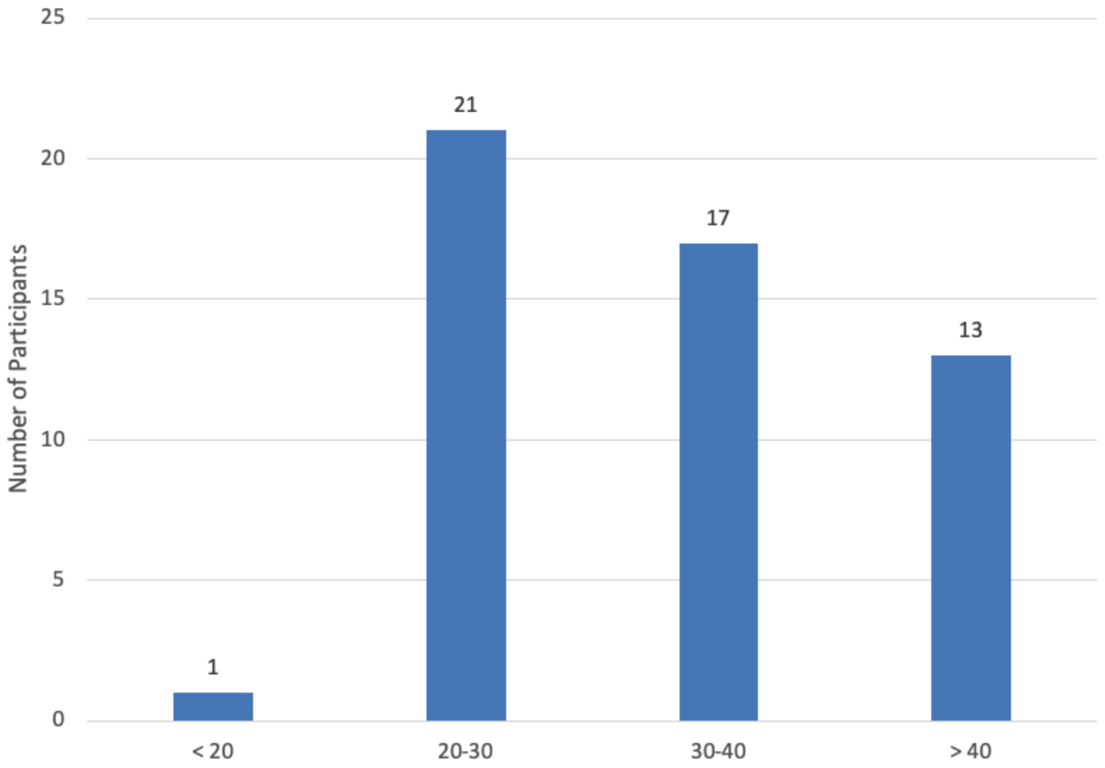


Figure 20. CPD – Age distribution of the participants

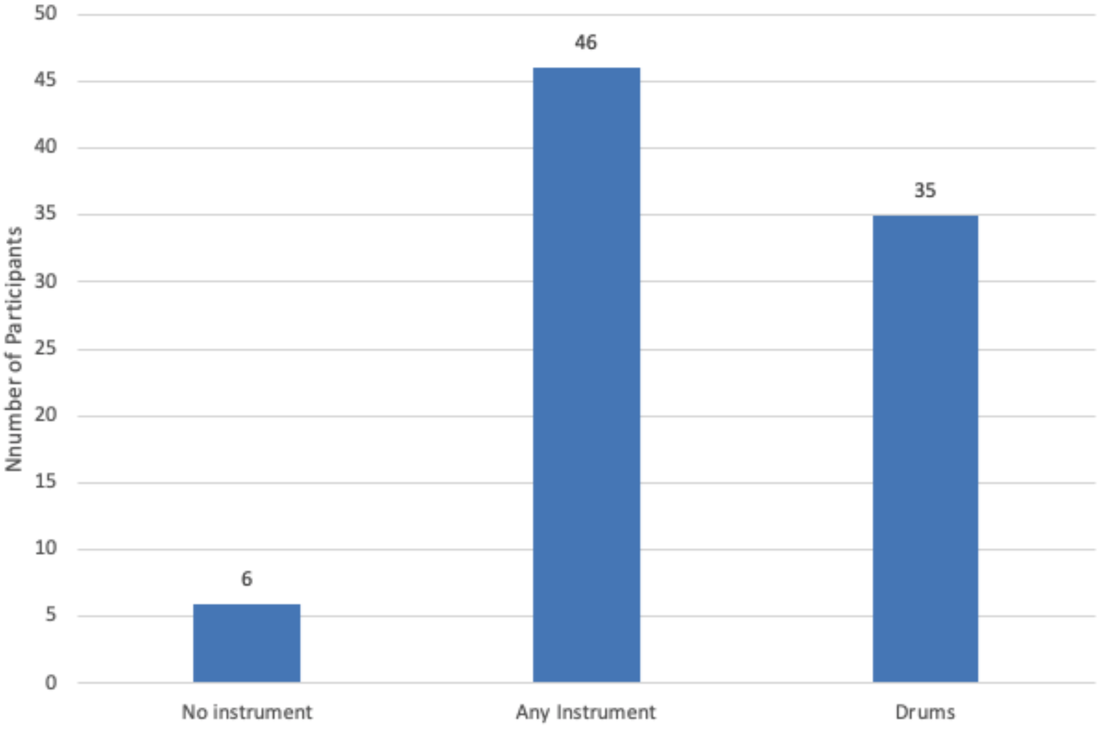


Figure 21. CPD – Do you play any instrument?

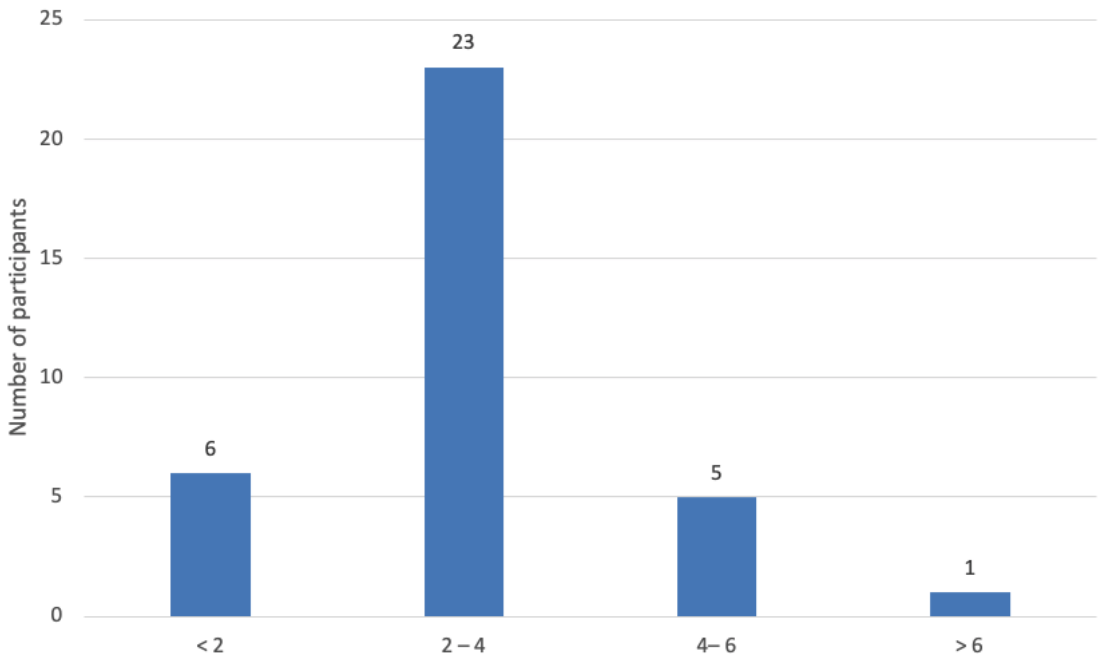


Figure 22. CPD - Time in hours per week, spent on playing drums.

4.2.3. Experiment Results

First, the number of votes for each dualization from all the experiments were summed up together. The *adjusted Wald confidence intervals* were calculated in order to evaluate the statistical significance of the difference between the choices. Figure. 23 pictures the results of the analysis.

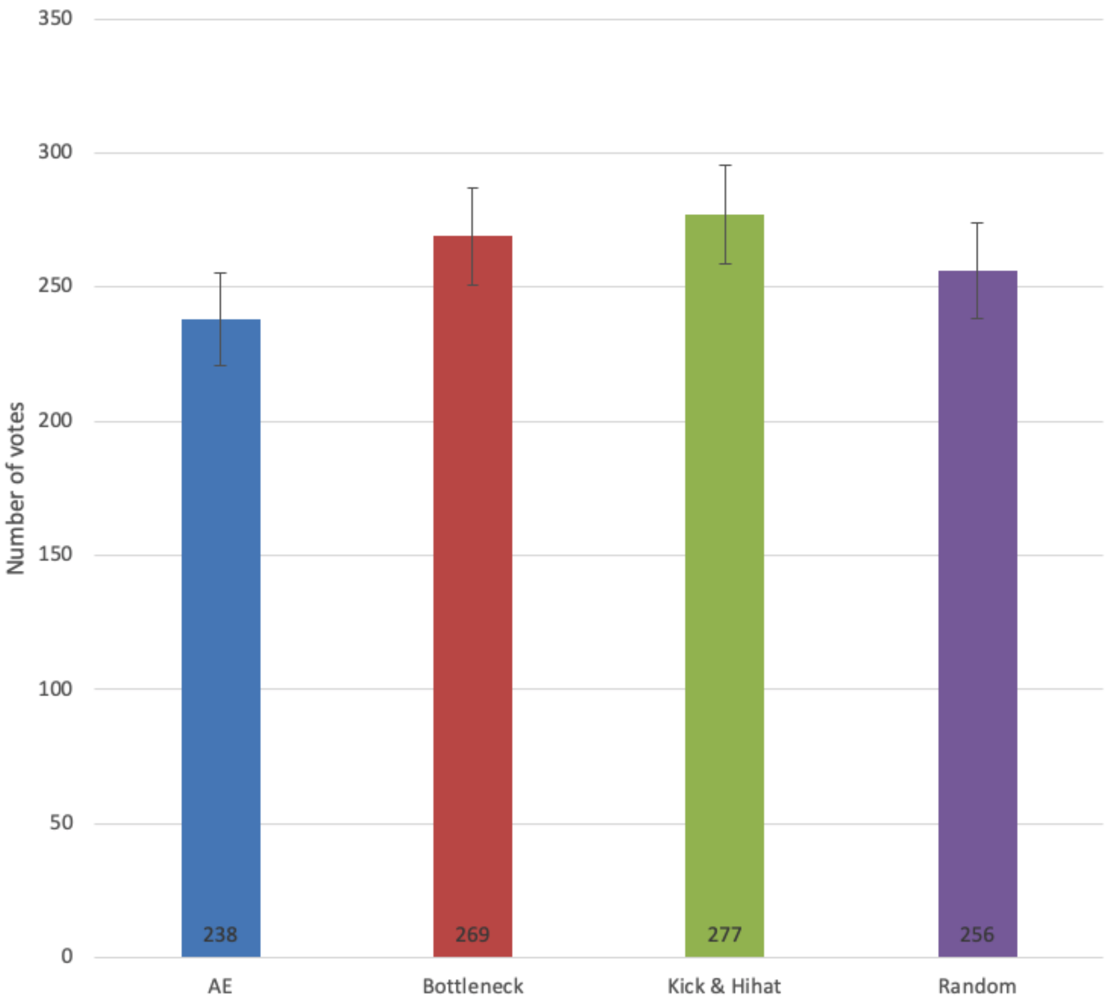


Figure 23. CPD - Summary distribution of participants' preferred dualizers. Confidence intervals were calculated using the adjusted Wald method.

The analysis shows that when the answers are analyzed cumulatively, the confidence intervals are overlapping in between all the groups. Therefore, the choices cannot be distinguished from the random.

In order to find the statistical significance of the results separately for each of the 20 examples, the *p-value* from a two-tailed binomial test was calculated for the winning dualizations. Table 1 summarizes the results.

Example	1	2	3	4	5	6	7	8	9	10
p-value	0.2	0.2	0.04	0.08	0.08	0.02	0.2	0.02	0.52	0.52

Example	11	12	13	14	15	16	17	18	19	20
p-value	0.01	0.2	0.52	0.52	0.11	0.11	0.34	0.34	0.11	0.11

Table 1. CPD - Two-tailed binomial p-values for the winning choices in respect to the example.

Only the examples where the *p-value* calculated for the most chosen option was below 0.1 were selected for further analysis. In the rest of the cases, the *p-value* would indicate that there is at least a 10% probability of the option being chosen at random. The examples fulfilling the condition were examples: *3, 4, 5, 6, 8, 11.*

For this subset of examples, the adjusted Wald confidence intervals were calculated in order to learn about the statistical significance of differences between the selected options. Figure 24 pictures the results of the analysis.

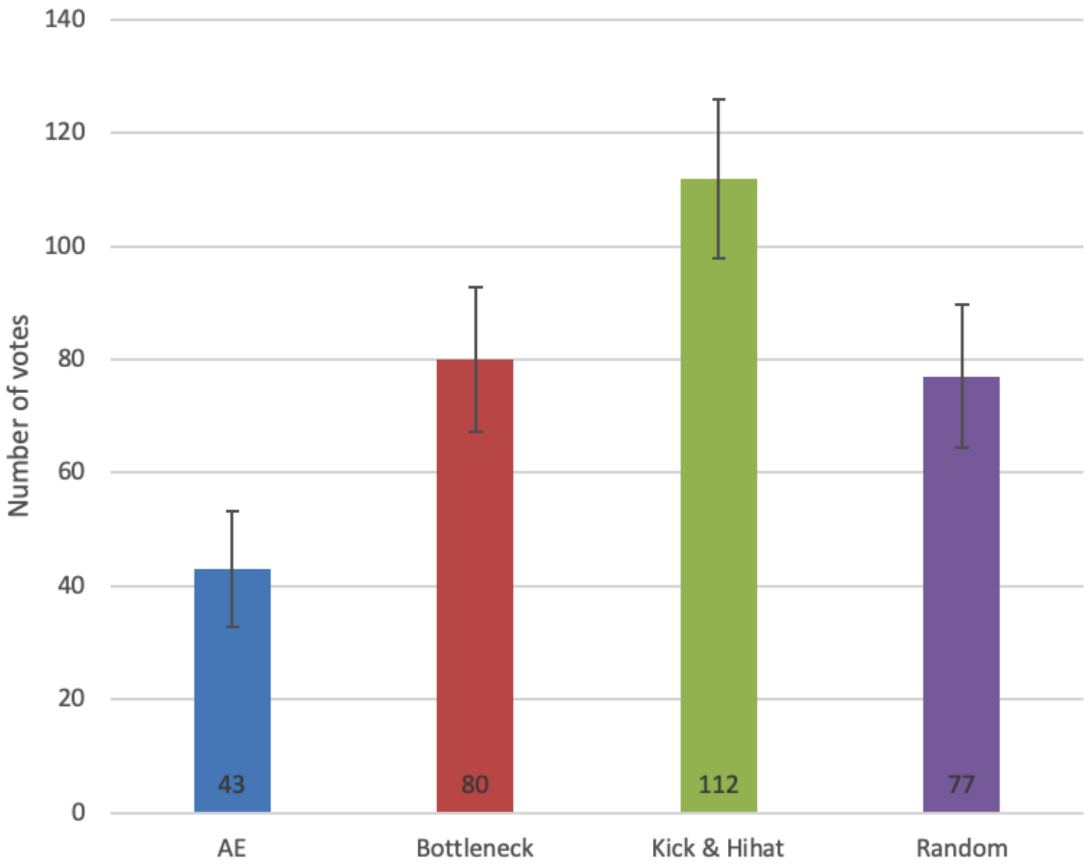


Figure 24. CPD – Summary distribution of participants' votes for the preferred dualization for the examples where two-tailed binomial p-value was smaller than 0.1. The confidence intervals are calculated following the adjusted Wald method.

The adjusted Wald confidence analysis for the examples where the winning option was selected with statistical significance leads to the following conclusions:

- Kick & Hihat baseline was selected as the best matching dualization, with statistical significance.
- AE dualization significantly loses in comparison with all the rest of the possible options.
- Although the Bottleneck dualization was chosen more often than the Random one, statistically they cannot be distinguished from each other.

4.2. Match dualization with the original pattern (MDO)

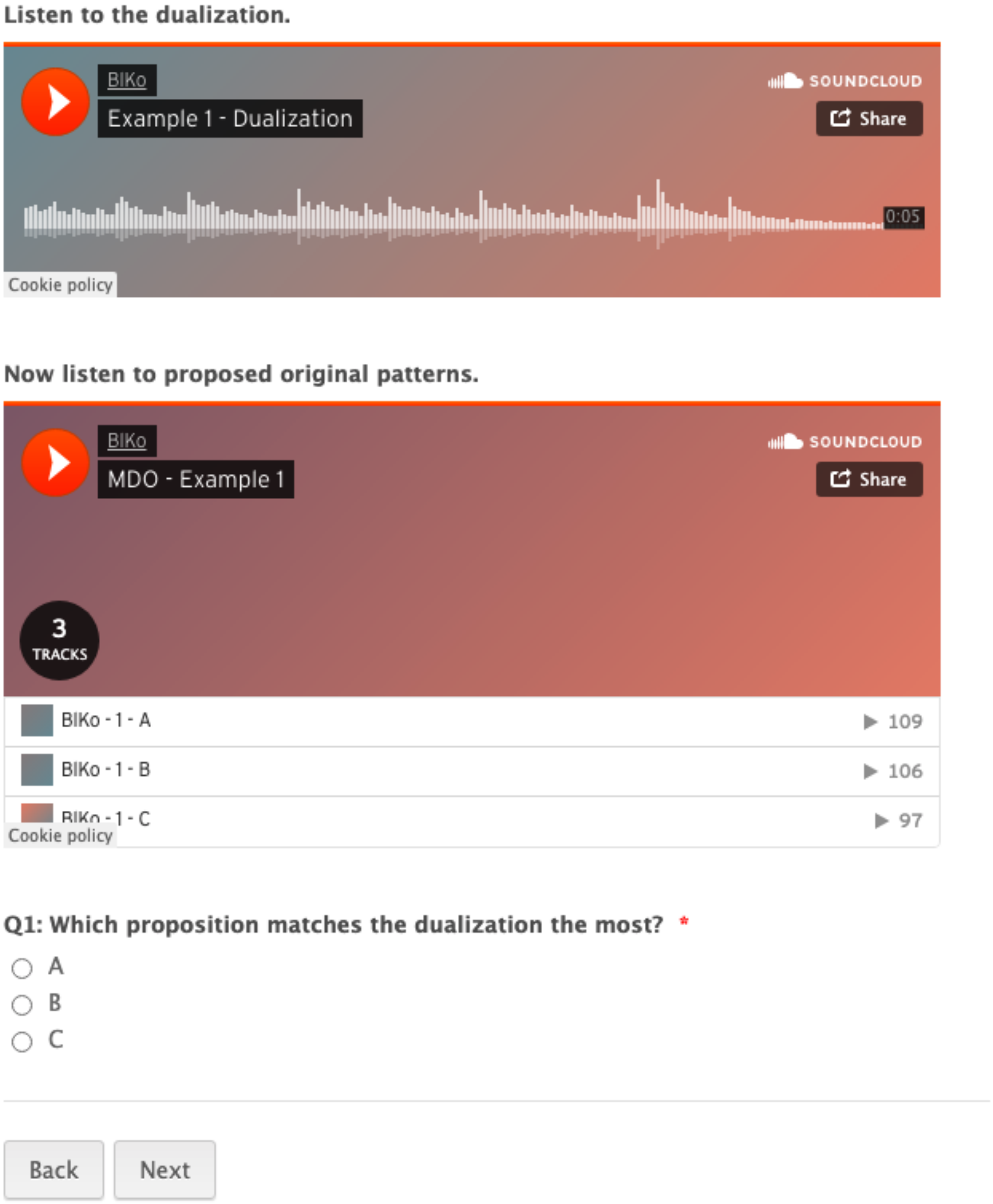
In this experiment, we intended to learn if the proposed models succeed to convey the essence of the original pattern. We have tested if the original pattern can be recognized in its dualized form.

4.2.1. Experiment design

The participants first had to listen to a single dualization, either Bottleneck or Autoencoder. Next, they were presented 3 possible patterns, that the dualization was derived from. They had to choose the one, that was most likely the original one.

The experiment was prepared in the following steps:

1. From the Groove MIDI Dataset we have randomly selected:
    a. 10 unique patterns for testing the AE model (group A),
    b. 10 unique patterns for testing the Bottleneck model (group B),
    c. 40 unique patterns for pairing with the original ones (group T)
2. We have used two aforementioned dualizing models: AE and Bottleneck to process the selected patterns from groups A and B in order to generate their dualized versions.
3. For each dualized version of the pattern, we have created a group of 3 patterns consisting of 1 original pattern (the one the dualized version was derived from), and 2 randomly picked, unique patterns from the group T.
4. We have presented 20 sets of the dualized pattern along with the aforementioned group of 3 patterns to the participants, asking them to choose the one that they think should be the original one (see Fig. 25).



Listen to the dualization.

Now listen to proposed original patterns.

Q1: Which proposition matches the dualization the most? *

○ A

○ B

○ C

Figure 25. Online questionnaire for the MDO experiment.[3]

[3] Available at https://form.jotform.com/202335390045346.

4.2.2. Participants

The same as in the case of the CPD experiment, we have outsourced the task of participating in the survey through the MTurk platform (See 4.1.2).

We have collected the questionnaires from 74 participants.

Below, we present the plot picturing the age distribution of the participants (see Fig. 26). Along with it, we show the distribution of the participants that play some instrument or specifically drums (see Fig. 27). Finally, we asked how much time per week do they spend on playing the instrument in case they have answered that the instrument they play is drums (see Fig. 28).

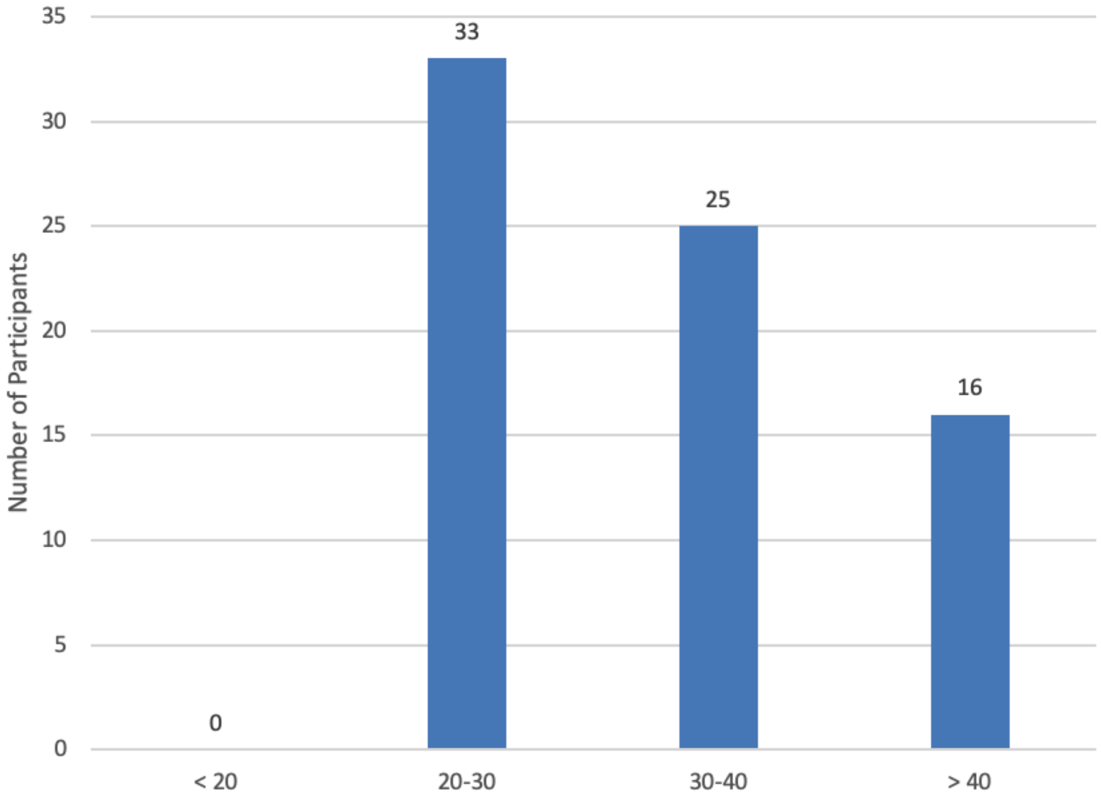


Figure 26. MDO - Age distribution of the participants.

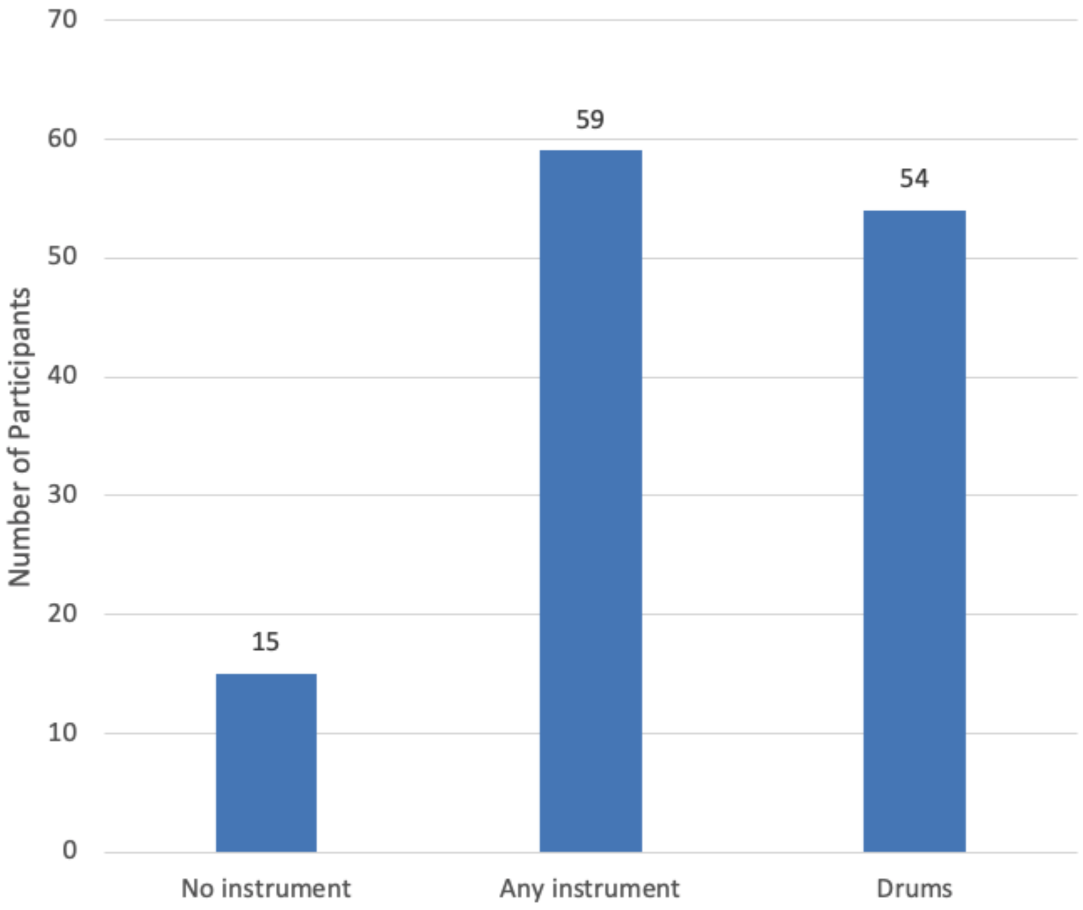


Figure 27. MDO – Do you play any instrument?

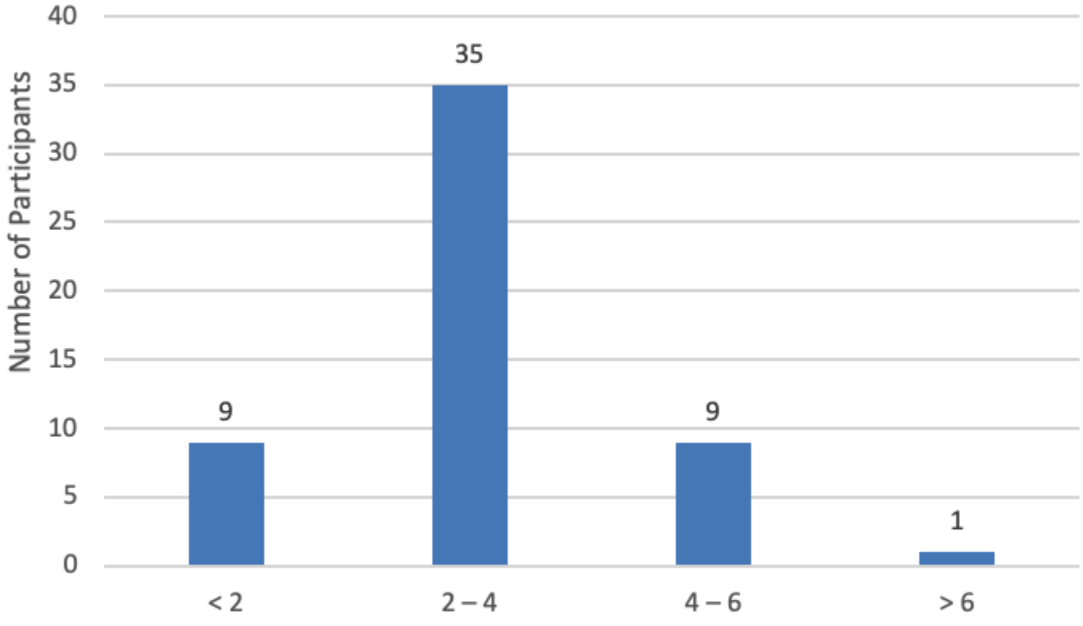


Figure 28. MDO - Time in hours per week, spent on playing drums.

4.2.3. Experiment Results

First, the number of votes for each chosen most fitting pattern from all the experiments were summed up together. The *adjusted Wald confidence intervals* were calculated in order to evaluate the statistical significance of the difference between the choices. Figure. 29 pictures the results of the analysis.

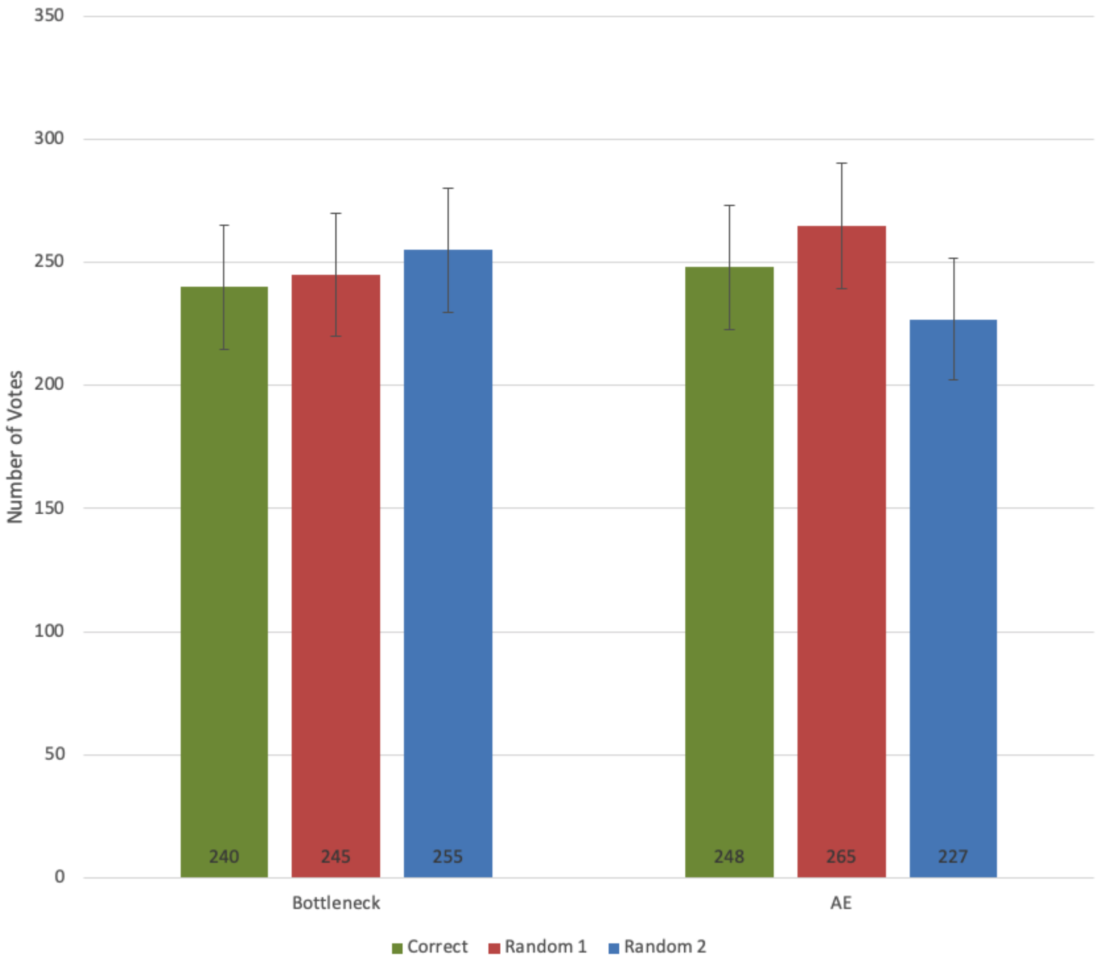


Figure 29. MDO - Summary distribution of participants chosen most fitting pattern. Confidence intervals were calculated using the adjusted Wald method.

The plot shows the confidence intervals overlapping in between all the tested groups. The distribution cannot be distinguished from the random distribution.

Similarly, with the CPD experiment (4.2.3), we calculated the *p-value* from a two-tailed binomial test for the winning options. Table 2 summarizes the results.

Example	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
p-value	0.001	0.06	0.06	0.49	0.05	0.01	0.18	0.54	0.39	0.05

Example	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
p-value	0.71	0	0.27	0.9	0.05	0.03	0.1	0.03	0.18	0.18

Only the examples where the *p-value* calculated for the most chosen option was below 0.1 were selected for further analysis. In the rest of the cases, the *p-value* would indicate that there is at least a 10% probability of the option being chosen at random. The examples fulfilling the condition were examples: *1, 2, 3, 5, 6, 10, 12, 14, 15, 18.*

For this subset of examples, the adjusted Wald confidence intervals were calculated in order to learn about the statistical significance of differences between the selected options. Figure 24 pictures the results of the analysis.

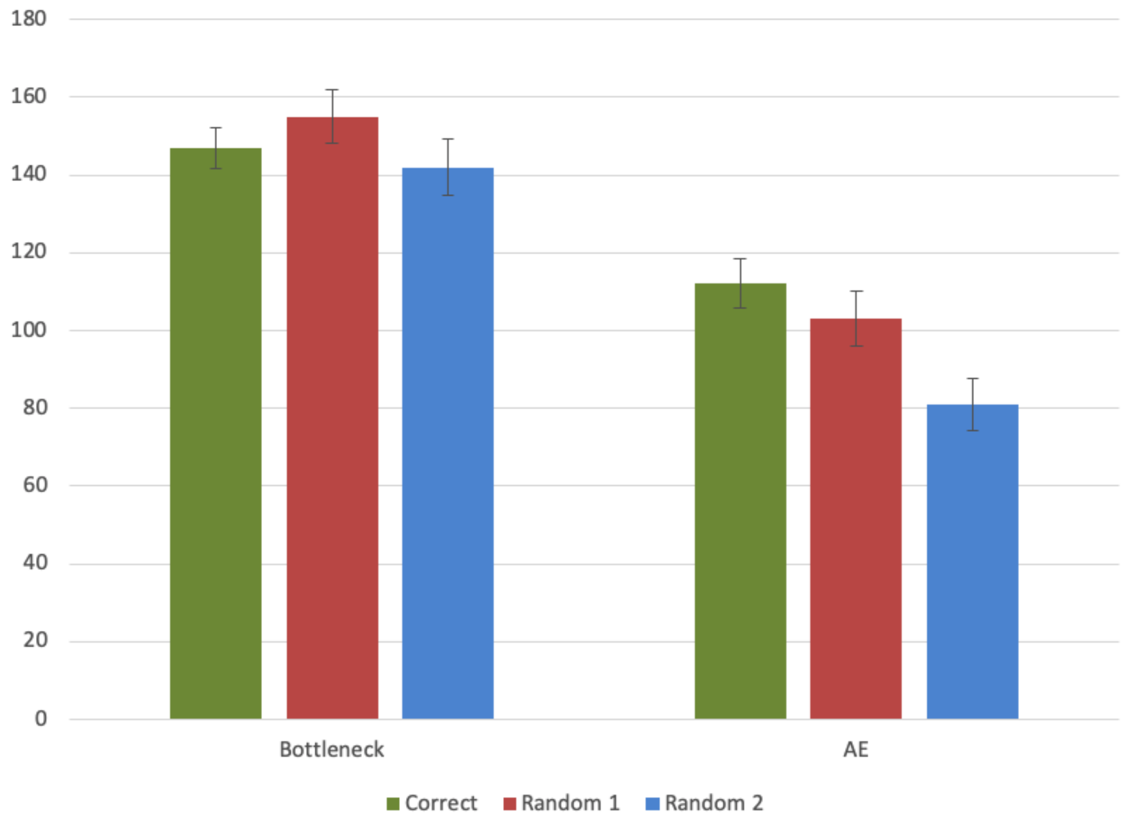


Figure 30. MDO – Summary distribution of participants' votes for the best matching original pattern for the examples where two-tailed binomial p-value was smaller than 0.1. The confidence intervals are calculated following the adjusted Wald method.

The adjusted Wald confidence analysis for the examples where the winning option was selected with statistical significance leads to the following conclusions:

- In the case of the Bottleneck model, all the best matching chosen options' confidence intervals overlap, therefore the differences cannot be distinguished from the random distribution.
- In the case of the AE model, the correct option was chosen significantly more often than Random 2. However, it cannot be statistically distinguished from the Random 2 option, therefore it has no statistical significance.

5. Discussion

In this chapter, we will provide a discussion on the problematics of the task of dualization, the methodology we have finally chosen, our approach to the evaluation, and the results of the thesis. We will conclude on the outcome of the dissertation. Furthermore, in the future work section, we will discuss the possible direction this work could be taken upon.

5.1. General discussion

Our exploration of the topic of rhythm dualization was a long and very illuminating adventure. We have consciously decided to take a risk of working on a new task that was not previously defined and not a lot of research would tackle it in its direct form before. We have decided to follow the intuition that we have all shared, that the task can be addressed. We aimed at paving a path at least and putting some signs that would point towards the right direction.

Working with new concepts requires maintaining the heads open wide. Therefore the first steps we were putting quite bravely, exploring wide areas all at once, searching for an inspiration drawing from the concepts coming from a broad spectrum of disciplines. We have intended to find strong foundations for our work that could efficiently support our methodology. We have considered resolving the problem in a purely analytical way, just by understanding the rules of interaction between different parts of polyphonic rhythms.

After a while, we have realized that the topic is very complex, and finding a good analytical approach would require organizing a multitude of experiments involving the listening participants. This kind of approach would require much more time than we have had in the limited framework of writing a master thesis. For this reason, we needed to re-evaluate our initial idea of going the analytical way and to start leaning towards the alternative that we have kept in mind since before starting our efforts.

The alternative was the ever-surprising area of machine learning. We have seen numerous times how the ML models would find sense in the data and build the concepts on their own. Inspired by the examples from both fields of audio and image, we have decided to give it a try and take a look into neural nets guts hoping that the internal representations will make perceptual sense like in other examples.

5.1.1. Methodology

We have decided to build upon a model that was already positively assessed and tested which happened to be the GrooVAE model. We have tried looking at the internal, intermediary representations of the processed patterns at different stages. Finally, we have proposed to test if by forcing the net to represent the pattern at each time step as a two-dimensional hidden state vector, the net would learn to represent them in a perceptually relevant way.

Accessing the h-vector values

The issue with this approach is that we are not sure how to interpret the values kept in the hidden state vectors. Even if they hold the perceptual relevance, it might be obscured by a data transformation that is a by-product of the networks' training process and we don't know what data transformation it is. For example, even if the values on each time point express the energy in the rhythm track, the values could be inverted, the neural net could deal with that inversion on further layers and we wouldn't necessarily know about it.

We have decided to go with an assumption that the values kept in the h-vectors can be read as velocity. We have thought about different possible data transformations, like thresholding or inversion, but every such transformation would be an arbitrary choice and we did not want to play a guessing game.

To know how the data should be interpreted, we should first know in what form the data is kept. Some constraints could be put on the way how the h-vectors are constructed. One way of conditioning the construction of hidden state vectors would be modifying the loss function in such a way, that it would penalize the intermediary representations that do not keep the imposed structure. The internal representations of the network would then be a subject of training, the same way as the output of the network is.

One concern with such a direct way of reading the h-vectors comes from the fact that the GrooVAE net we were training learns to represent the information not only about the presence and velocity of the hits but also about their microtiming. If the information kept in the state would consist only of velocity and hit, reading such state as velocity only would be better justified, because both these data bits represent the energy at a time point. However, with the additional bit of information about the microtiming, perhaps the information should be transformed first so that the microtiming is not contained in it before it can be interpreted as the information about the energy at a time point.

Another interesting cue suggesting that constraining the structure of the h-vectors is the fact that even in case of quite sparse original patterns, the dualizations returned by both tested models remain active, not reflecting the nature of the original rhythm very well. Such an example is presented in Fig. 31.

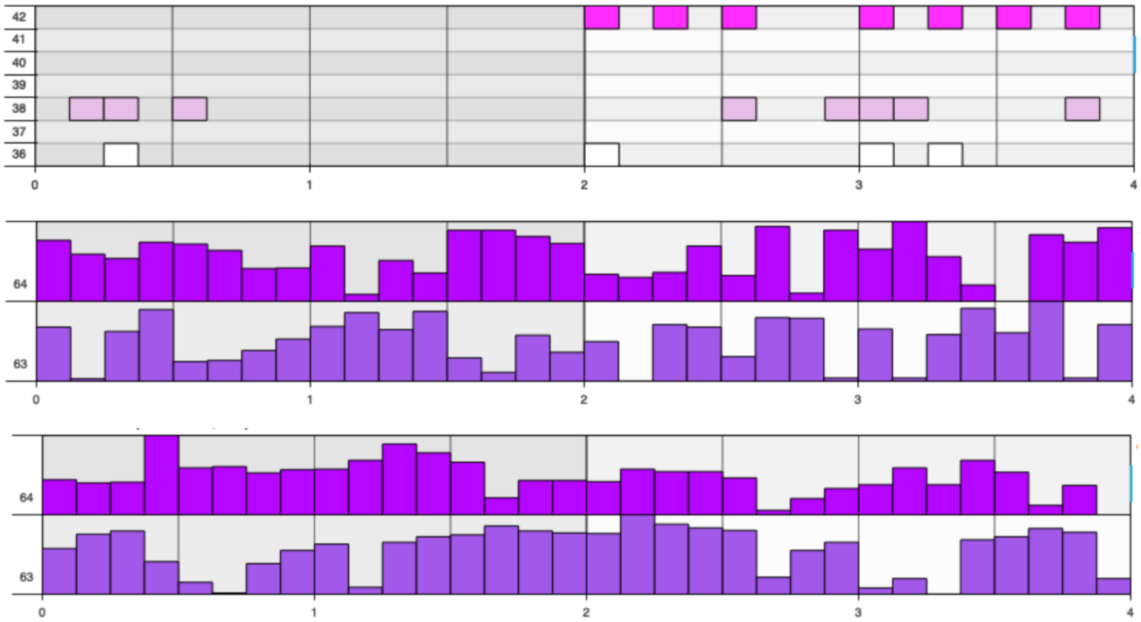


Figure 31. The sparse, original pattern (top), AE dualization (middle), and Bottleneck dualization (bottom).

Dataset

We have used the entire Groove MIDI Dataset for training our models. The dataset, however, lacks some descriptive characteristics of the rhythms that it contains. Looking closer at the original patterns, along with their dualizations reveals a couple of interesting facts about the dualizing models.

For example, the more loopy rhythm patterns sometimes effect in similar loopy characteristics of the output dualizations, at least in the case of the Bottleneck model (See Fig. 32).

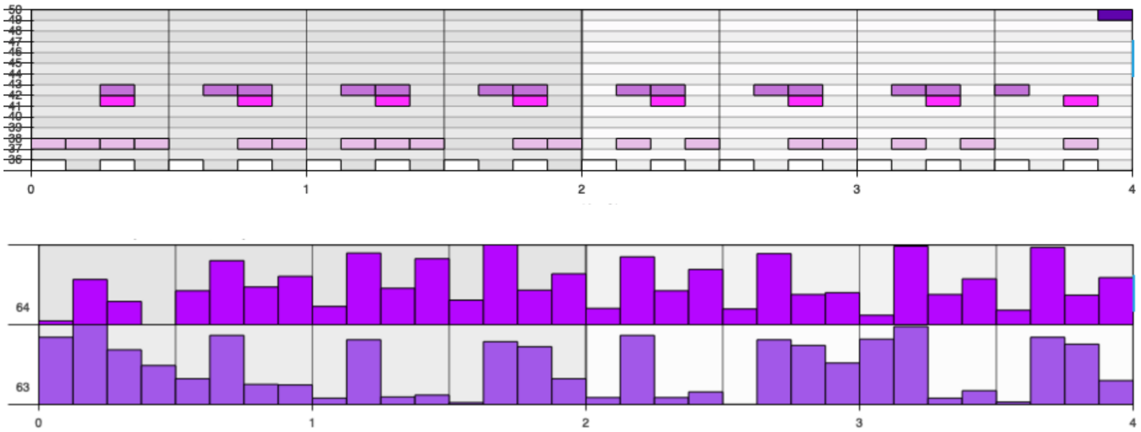


Figure 32. Looped rhythm pattern (upper part) and its Bottleneck dualization (lower part).

In this example, you can see that the first track of the dualization exposes similar bar-wise repetitiveness as the original pattern.

This is not a behavior that relates to all the patterns, however, without good descriptors in the dataset, we were not able to explore this kind of phenomena more in detail.

5.1.2. Evaluation

For the evaluation part of the dissertation, we wanted to test two aspects of the outcome of our research. Firstly, we wanted to learn if the models we have developed, generate the pattern transformations that are perceptually similar to the input patterns. Secondly, in case the perceptual similarity would already be in place, we wanted to evaluate how well would they perform in comparison to a very simple baseline algorithm.

Both conducted experiments suffered from a similar issue: the lack of statistical significance. After we have prepared the questionnaires we have already understood that the task of matching several rhythm patterns together using the notion of “similarity” or “recognizability” based fully on one’s intuition is not a simple task, and it might be especially difficult if one does not have a lot of musical experience.

We have observed some commonality between the patterns where the two-tailed binomial p-value for the winning option was low, which means that the preferred choices were more emphasized and effectively statistically more significant. If we look at the symbolic, MIDI representation we can see that the character of such patterns is rather very repetitive, loopy (Fig. 33, Fig. 34)

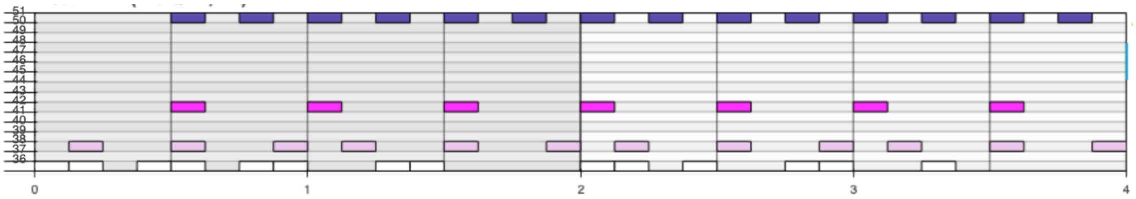


Figure 33. CDO - Experiment 11, p-value of winning option: 0,02.

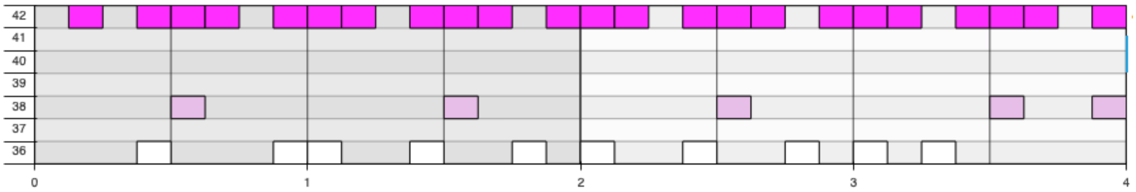


Figure 34. CDO - Experiment 6, p-value of winning option: 0,01.

Especially when compared to patterns whose p-value for winning option was very high, effectively meaning that the winning choice distribution would be possibly random. These patterns (Fig. 35, Fig. 36) share a higher level of temporal complexity, as compared to the previously presented ones.

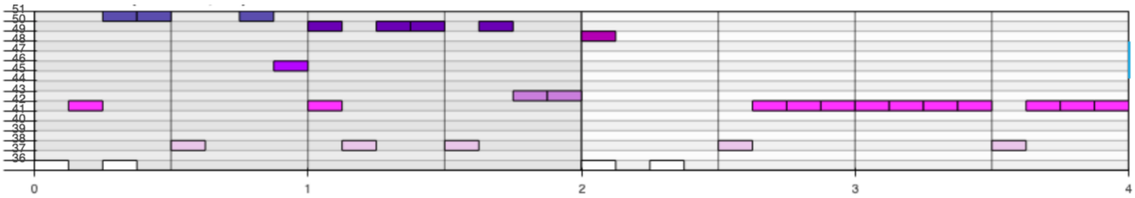


Figure 35. CDO - Experiment 9, p-value of the winning option: 0,52.

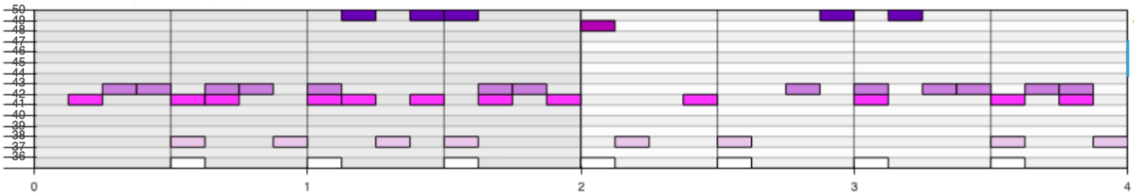


Figure 36. CDO - Experiment 13, p-value of the winning option: 0,52.

This shows that perhaps a better curation of evaluated patterns could lead to more informative results. The comparative analysis between groups of patterns of different complexity could be conducted to better understand the influence of the pattern complexity on the model performance.

5.2. Conclusions

This work has dealt with a complex, novel concept of rhythm pattern dualization. We have reviewed the state of the art in several fields of science dealing topics like neural rhythm representation, rhythmic pattern cognition, rhythm analysis, drum performance, sequence processing, or machine learning. The results of this analysis are available in the State Of The Art.

The most straight-forward goal of the thesis was to find a computational method for transforming any polyphonic rhythm pattern into its dualized form. In Methodology, we present our efforts on the development of two machine learning models based on the state of the art research on Variational Autoencoder (VAE) networks. We have looked into the h-vectors of the VAE network and observed that the rhythm structure is somehow reflected in the dualizations. However, more work has to be done in order to develop reliable, perceptually relevant models.

In Evaluation we have presented our efforts on conducting online listening experiments, presenting two types of tasks to evaluate the developed dualizers. The results of the listening experiments showed that there is more work to be done to propose a successful solution to the presented task. The listening experiments proved that developed models are not good enough and require more work in the search of perceptual significance.

Finally, in the Discussion, we have elaborated more on the nature of the rhythm dualization. We have then discussed the advantages and the drawbacks of our decision, regarding both methodology and evaluation methodology.

Although the developed dualizer models proved to not be addressing the task very well, we believe that this work still holds value. Perhaps the biggest contribution we have made is the strong theoretical fundamentals that we have put for the new task in the field of music analysis, the task of rhythm dualization.

We have left the work at a point from where it could be picked up pretty easily. There is a well-defined room for improvements that we are suggesting in the next section.

5.3. Future work

In the future, most importantly the research on adjusting the loss function of our models could be conducted. As described in the Discussion section, the loss function could constrain the structure of the hidden state vector that we build the dualizations upon. With a better-designed loss function, we would know the transformation function necessary to cast the dualized representations kept in h-vectors into the more perceptually relevant dualizations.

With better curation of the dataset, the algorithms could be tested more in detail. The only patterns we have tested were real-life rhythms played by professional drummers. To better understand the representation of data kept in the hidden state, one could generate a dataset of goal-oriented MIDI patterns. Some examples of such patterns would be: a subset of patterns when one instrument is active on every step; a subset of patterns, where one bar is looped over the whole pattern; a subset of patterns where one half of the pattern differs a lot from another half. Then, side to side comparison between pattern and the dualization could be conducted for a different subset of patterns. That would definitely be a valuable insight into the nature of h-vector representations.

One technical improvement that would enable further exploration would be parting from the GrooVAE model. Now, that we have conducted preliminary research based on this model, it does not seem to be necessary to continue developing the research upon it. Developing a more dedicated technical framework would make introducing further model improvements it much more straightforward and independent from the efforts of the Magenta team at Google.

6. Bibliography

1. Patel, A. D., & Iversen, J. R. (2014). The evolutionary neuroscience of musical beat perception: the Action Simulation for Auditory Prediction (ASAP) hypothesis. Frontiers in systems neuroscience, 8, 57. https://doi.org/10.3389/fnsys.2014.00057

2. Nozaradan, S., Peretz, I., & Mouraux, A. (2012). Selective Neuronal Entrainment to the Beat and Meter Embedded in a Musical Rhythm. The Journal of Neuroscience, 32, 17572 - 17581.

3. Nozaradan, S., Peretz, I., Missal, M., & Mouraux, A. (2011). Tagging the Neuronal Entrainment to Beat and Meter. The Journal of Neuroscience, 31, 10234 - 10240.

4. Large, E.W., Herrera, J.A., & Velasco, M.J. (2015). Neural Networks for Beat Perception in Musical Rhythm. Frontiers in Systems Neuroscience, 9.

5. Honing, H., Merchant, H., Háden, G.P., Prado, L., & Bartolo, R. (2012). Rhesus Monkeys (Macaca mulatta) Detect Rhythmic Groups in Music, but Not the Beat. PLoS ONE, 7.

6. Mates, J. (2004). A model of synchronization of motor acts to a stimulus sequence. Biological Cybernetics, 71, 186.

7. Hary, D., & Moore, G.P. (1987). Synchronizing human movement with an external clock source. Biological Cybernetics, 56, 305-311.

8. Temperley, D. (2009). A Unified Probabilistic Model for Polyphonic Music Analysis. Journal of New Music Research, 38, 18 - 3.

9. Temperley, D. (2004). An Evaluation System for Metrical Models. Computer Music Journal, 28, 28-44.

10. Scheirer, E. D. (1998). Tempo and beat analysis of acoustic musical signals. The Journal of the Acoustical Society of America. https://doi.org/10.1121/1.421129

11. Witek, M. A. G., Clarke, E. F., Kringelbach, M. L., & Vuust, P. (2014). Effects of polyphonic context, instrumentation, and metrical location on syncopation in music. Music Perception, 32(2), 201–217. https://doi.org/10.1525/MP.2014.32.2.201

12. Hove, M. J., Marie, C., Bruce, I. C., & Trainor, L. J. (2014). Superior time perception for lower musical pitch explains why bass-ranged instruments lay down musical rhythms. Proceedings of

the National Academy of Sciences of the United States of America. https://doi.org/10.1073/pnas.1402039111

14. Huron, D. (2001). Tone and Voice: A Derivation of the Rules of Voice-Leading from Perceptual Principles. Music Perception. https://doi.org/10.1525/mp.2001.19.1.1

15. Toussaint, G. T. (2016). The geometry of musical rhythm: What makes a “Good” rhythm good? In The Geometry of Musical Rhythm: What Makes a “Good” Rhythm Good? https://doi.org/10.1080/17513472.2014.906116

16. Podmore, J. (2020). Jaki Liebezeit: The Life, Theory And Practice Of A Master Drummer. Unbound.

17. Moore, B. C. J., & Gockel, H. E. (2012). Properties of auditory stream formation. Philosophical Transactions of the Royal Society B: Biological Sciences. https://doi.org/10.1098/rstb.2011.0355

18. Han, J., Kamber, M., & Pei, J. (2006). Data Mining: Concepts and Techniques, 3rd edition.

19. Jolliffe I.T. Principal Component Analysis, Series: Springer Series in Statistics, 2nd ed., Springer, NY, 2002, XXIX, 487 p. 28 illus. ISBN 978-0-387-95442-4

20. Perdomo, Wilmar & Méndez, Alfredo. (2018). Application of Principal Component Analysis to Image Compression. 10.5772/intechopen.75007.

21. Murali, Y., & Babu, M. (2012). PCA based image denoising. Signal & Image Processing : An International Journal, 3, 236-244.

22. Zhang, L., Dong, W., Zhang, D., & Shi, G. (2010). Two-stage image denoising by principal component analysis with local pixel grouping. Pattern Recognit., 43, 1531-1549.

23. Hernandez, W., & Méndez, A. (2018). Application of Principal Component Analysis to Image Compression.

24. https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

25. Hinton, G.E., & Salakhutdinov, R. (2006). Reducing the Dimensionality of Data with Neural Networks. Science, 313, 504 - 507.

26. Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. (2008). Extracting and composing robust features with denoising autoencoders. ICML '08.

27. http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/

28. DuBreuil, A. (2020), Hands-On Music Generation with Magenta, p. 46

29. https://magenta.tensorflow.org/2016/07/15/lookback-rnn-attention-rnn

30. Choi, K., Fazekas, G., & Sandler, M.B. (2016). Text-based LSTM networks for Automatic Music Composition. ArXiv, abs/1604.05358.

31. Roberts, A., Engel, J., Raffel, C., Hawthorne, C., & Eck, D. (2018). A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music. ArXiv, abs/1803.05428.

32. Sutskever, I., Vinyals, O., & Le, Q.V. (2014). Sequence to Sequence Learning with Neural Networks. ArXiv, abs/1409.3215.

33. Hutchings, P. (2017). Talking Drums: Generating drum grooves with neural networks. ArXiv, abs/1706.09558.

34. Gillick, J., Roberts, A., Engel, J., Eck, D., & Bamman, D. (2019). Learning to Groove with Inverse Sequence Transformations. ICML.

35. Darwin, C. (1871). The descent of man: And selection in relation to sex. London: J. Murray.

36. Jones, M.R., & Boltz, M.G. (1989). Dynamic attending and responses to time. Psychological review, 96 3, 459-91.

37. Bååth, R. (2015). Subjective rhythmization: A replication and an assessment of two theoretical explanations. Music Perception, 33, 244-254.