

Master in Sound and Music Computing
Universitat Pompeu Fabra

Real-time Generation of Percussive Rhythms Using Descriptors

Alexandre Vilanova

Supervisor: Daniel Gómez

Co-Supervisor: Sergi Jordà

August 2025



Contents

1	Introduction	1
2	State of the Art	3
2.1	Rhythm Perception	3
2.2	Rhythm Spaces	5
2.3	Variational Autoencoders in Generative Music	6
3	Methods	8
3.1	Problem Setup	8
3.2	Variational Autoencoder Model	9
3.2.1	Model Architecture	9
3.2.2	Training Procedure	10
3.2.3	Latent Space Sampling	11
3.3	Descriptor-based Model	11
3.3.1	Rhythm Descriptors	11
3.3.2	Discrete descriptor precision	15
3.3.3	Neural Network Implementation	16
3.3.4	Dimensionality Reduction	17
3.3.5	Model Evaluation	18
3.4	Comparing VAE and descriptor-based approaches	19
3.5	Smoothness Experiment	21
3.5.1	Experiment Design	22
3.5.2	Implementation	24

3.6	User Experience Experiment	24
3.6.1	Implementation	25
3.6.2	Structure	25
3.6.3	Data Collection and Storage	26
4	Results	31
4.1	Dimensionality Reduction	31
4.2	Smoothness Experiment	34
4.3	User Experience Experiment	36
5	Discussion	40
5.1	VAE vs. Descriptor-Based Approaches	40
5.2	Descriptor Selection	41
5.3	Smoothness Analysis	41
5.4	User Experience Experiment	42
5.4.1	Validation of the Descriptor-Based Approach	42
5.4.2	User Background and Performance	43
5.4.3	Descriptor Interpretability	43
5.5	Qualitative User Feedback Analysis	44
5.6	Methodological Considerations and Limitations	45
6	Conclusions	46
7	Future Work	48
	List of Figures	50
	List of Tables	51
A	Source code and demo	55

Acknowledgement

I would like to express my sincere gratitude to my supervisor, Daniel Gómez, for his invaluable guidance, unwavering support, and contagious enthusiasm throughout this work. His thoughtful insights and continuous encouragement have played a crucial role in guiding the direction and enhancing the quality of this research, and his mentorship has made this journey both rewarding and inspiring.

Abstract

A fundamental challenge in computational music generation lies in developing control interfaces that provide intuitive, musically meaningful interactions with generative systems. This thesis addresses this challenge specifically for rhythmic generation, focusing on the development of a system capable of generating 16-step monophonic rhythmic patterns in real time using musically intuitive controls.

Our method uses perceptually grounded rhythmic descriptors as an expressive, intuitive control space. A neural network is trained on all possible binary 16-step monophonic patterns, learning to map from descriptor space back to rhythmic patterns. We compare this descriptor-based approach to a variational autoencoder model and find the former more effective for usability and expressive control. An interactive interface is developed for exploration and testing, followed by quantitative and qualitative experiments evaluating the smoothness and user intuitiveness of the system.

Findings show that the descriptor-based model aligns well with listener perception, balancing usability with expressive flexibility. While limited to monophonic rhythms, the system establishes descriptors as a strong foundation for extending interactive rhythm generation to polyphonic and more complex domains.

Keywords: rhythm generation, descriptor engineering, variational autoencoders, generative music, symbolic music, real-time interaction.

Chapter 1

Introduction

One of the central challenges in computational music generation is designing control interfaces that allow users to interact with generative systems in ways that feel both intuitive and musically meaningful. This thesis focuses on this challenge for rhythm, developing a system capable of producing 16-step monophonic rhythmic patterns in real time, guided by controls that are easy for musicians to understand and use.

Our main objective is to develop a musically meaningful method for generating 16-step monophonic rhythms in real time. To achieve this goal, we investigate two contrasting approaches: one based on abstract latent space representations learned through variational autoencoders (VAEs), and another grounded in explicit rhythm descriptors corresponding to established concepts in music perception theory.

The first approach employs a VAE architecture that learns compact latent representations from rhythmic patterns without incorporating explicit musical knowledge. Following established work in VAE-based symbolic music generation (Brunner et al. 2018; Roberts et al. 2018; Vigliensoni et al. 2022), this method provides smooth interpolation capabilities through continuous latent spaces.

The second approach directly maps rhythm descriptors to monophonic rhythm patterns using a feedforward neural network. These descriptors, grounded in music

perception research, encompass quantitative measures of rhythmic properties such as onset density (Milne, Dean, and Bulger 2021; Milne and Herff 2020), syncopation (Gómez-Marín, Jordà, and Herrera 2015; Lerdahl and Jackendoff 1983), evenness (Milne and Dean 2016), and balance (Milne and Herff 2020), providing transparent control over specific rhythmic aspects and enabling users to manipulate well-defined musical features.

A preliminary exploration indicates that the descriptor-based method offers superior interpretability and musical control. Consequently, this thesis focuses primarily on the descriptor-based approach, investigating its effectiveness through smoothness experiments and user interaction studies to evaluate how well the system supports musical control in practice.

The contributions of this research include:

1. Implementation and comparison of VAE and descriptor-based approaches.
2. Development of a real-time interactive interface for testing both the VAE and the descriptor-based models.
3. Evaluation of the descriptor-based model through smoothness analysis and user interaction experiments, assessing performance and musical usability.

In this thesis, we focus on the complete space of 16-step monophonic patterns—comprising 65,536 possible binary sequences—to ensure that the developed methods can represent and manipulate any combination. Although this exhaustive scope may initially seem excessive, it is designed to keep the model genre-agnostic and to serve as an exploratory tool for generating and experimenting with new patterns.

Future work may extend these techniques to real-time transformation from monophonic to polyphonic structures, using the monophonic generation model as a baseline for constructing more complex, layered rhythms. This approach could allow the system to retain the expressive control of the existing model while exploring interactions between multiple rhythmic voices, potentially bridging the gap between monophonic input and polyphonic generation highlighted by Clark (2023).

Chapter 2

State of the Art

In this chapter, we cover several topics: rhythm perception, rhythm spaces, and the usage of VAEs in symbolic music generation. These are relevant to the context of the thesis and, more specifically, serve as fundamental pillars for our methodology.

Rhythm perception refers to the ability to detect, interpret, and respond to patterns of sound and silence over time. It is an elementary aspect of music processing and is essential for understanding the structure, phrasing, and emotional content of musical compositions.

Rhythm spaces are multidimensional interactive maps designed for the visualization, retrieval, and generation of drum patterns. These spaces organize drum patterns based on their perceptive similarity.

Variational autoencoders (VAEs) are a class of generative models that allow learning complex data distributions in an unsupervised manner. Unlike traditional autoencoders, VAEs are designed to encode input data into a latent space and decode it back, while also learning the underlying probability distribution of the data.

2.1 Rhythm Perception

Rhythm is a fundamental component of music, derived from the identification and comparison of recurring or varying events over time. Research in music cognition

indicates that the perception of rhythm is not only based on objective data, but also includes interpretative processes such as synchronization, anticipation, and hypothesis evaluation (Clark 2013). When individuals engage with rhythmic information, they frequently employ pattern comparison as a strategic approach to identify essential elements and allocate attention effectively (Palmer and Krumhansl 1990).

When humans hear a sequence of sounds, they naturally organize these sounds into a structure of strong and weak accented beats, known as the meter. The consistent strong positions within a meter are referred to as the pulse and they typically correspond to the main beat that is perceived and sometimes tapped while listening to music (London 2012).

In certain rhythmic patterns, the emphasis intentionally moves away from the strong pulses to the weak offbeats or even to unexpected places within the meter. This rhythmic variation from the regular pulse is known as syncopation and introduces depth and dynamism to the music, challenging the listener’s expectations and enhancing the rhythmic tension and release (Fitch and Rosenfeld 2007; Huron 2006; Longuet-Higgins and Lee 1984; Toussaint 2013).

The concept of density in rhythm refers to the number of onsets in a sequence. The interaction between onset density and their positions within the meter influences rhythm perception. As a sequence becomes more complex and dense, the individual rhythmic value of each sound diminishes (Milne, Dean, and Bulger 2021). Conversely, in sparser sequences, each sound carries more rhythmic significance, making it easier to discern the meter.

Within the literature, a variety of methods have been proposed to describe and quantify some of the characteristics of rhythm. We call these rhythm descriptors and they provide systematic ways to capture structural and perceptual properties, like onset density (Milne and Herff 2020), syncopation (Gómez-Marín, Jordà, and Herrera 2015; Lerdahl and Jackendoff 1983), evenness (Milne and Herff 2020; Milne and Dean 2016), and balance (Milne and Herff 2020).

While some rhythm descriptors operate in the audio domain, this thesis focuses

on symbolic representations, where onsets are explicitly encoded as discrete events. In the monophonic case, these symbolic descriptors are particularly effective, capturing both the temporal distribution of onsets and their alignment with metrical structures. Although many descriptors can be extended or adapted to polyphonic rhythms—such as the set of symbolic descriptors introduced by Gómez-Marín, Jordà, and Herrera (2020), which account for density and syncopation across multiple voices—this work concentrates exclusively on monophonic sequences, where the measures are more straightforward to compute and interpret.

2.2 Rhythm Spaces

Understanding and modeling rhythm similarity is a central topic in music cognition and computational music analysis. One influential approach to this problem has been the concept of rhythm spaces, which provide a structured, often low-dimensional representation of rhythmic patterns based on perceptual or computational criteria.

Early work by Gabrielsson (1973) explored several procedures to quantify polyphonic rhythm similarity. A key outcome of this research was the creation of conceptual rhythm spaces, in which rhythmic patterns were mapped to lower-dimensional representations, such as two-dimensional XY maps or three-dimensional scenes, enabling visualization and comparison of rhythmic structures.

Subsequent studies have extended and refined these ideas. Gómez-Marín, Jordà, and Herrera (2016, 2018, 2020) developed symbolic rhythm descriptors and rhythm spaces that capture both monophonic and polyphonic characteristics, grounded in principles of human rhythmic perception. More recently, Clark (2023) focused on bridging monophonic rhythm input with polyphonic drum pattern generation, highlighting methods to translate tapped rhythms into more complex, multi-voice patterns while preserving perceptual consistency.

2.3 Variational Autoencoders in Generative Music

Variational Autoencoders (VAEs) are a class of deep generative models designed to learn continuous latent representations of complex data distributions. In music generation, VAEs have been successfully applied to symbolic and audio representations alike, enabling the creation of novel musical sequences by sampling from the learned latent space.

In the symbolic domain, VAEs are particularly appealing for rhythm and melody generation because they can encode high-dimensional musical patterns—such as note sequences or drum hits—into a compact, continuous latent space that captures underlying musical structure and stylistic characteristics. Sampling from this latent space allows models to generate coherent variations, interpolate between musical patterns, and explore new combinations that maintain musical plausibility.

Several works have demonstrated the effectiveness of VAEs for generative symbolic music. For instance, Roberts et al. (2018) proposed hierarchical VAEs to capture long-term musical structure in symbolic sequences, while Brunner et al. (2018) used VAEs for modeling and generating polyphonic MIDI sequences.

Additionally, research has explored the integration of perceptual and timbral information into generative models. Esling, Chemla-Romeu-Santos, and Bitton (2018) introduced perceptually-regularized VAEs that align the latent space with human judgments of timbre similarity, enabling descriptor-based synthesis and controlled interpolation between sounds. Similarly, Kim et al. (2018) proposed a model for flexible timbre control in neural music synthesis, using learned instrument embeddings to allow continuous morphing between timbres while preserving musical expressiveness.

Other approaches have focused specifically on rhythmic generation, showing that VAEs can learn latent representations that reflect rhythmic style, density, and complexity, enabling both reconstruction of existing patterns and the creation of new, stylistically consistent rhythms. For instance, Vigliensoni et al. (2022) introduced

R-VAE, which enables real-time exploration of rhythmic patterns through a tactile interface, facilitating both reconstruction of existing rhythms and creation of new, stylistically consistent ones.

The core VAE architecture in this context consists of an encoder, which maps symbolic sequences to a latent distribution (often Gaussian), and a decoder, which reconstructs sequences from sampled latent vectors. Training optimizes a combination of reconstruction loss and a regularization term (the Kullback-Leibler divergence) to structure the latent space, allowing for smooth interpolation and meaningful exploration of musical variations.

Chapter 3

Methods

3.1 Problem Setup

We frame the rhythm generation task as a mapping from a low-dimensional control space to 16-step binary monophonic rhythmic patterns. Each pattern consists of a sequence of binary values indicating silence (0) or onset (1), resulting in a total of $2^{16} = 65,536$ possible combinations.

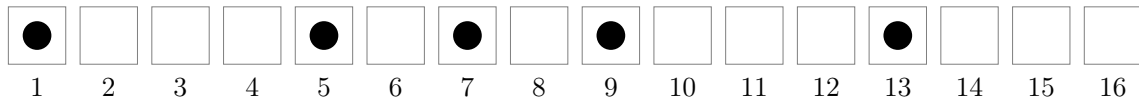


Figure 1: Visualization of a 16-step rhythm pattern.

In Figure 1, you can see an example of how a 16-step monophonic binary rhythm pattern looks like, onsets are represented as circles at steps 1, 5, 7, 9, and 13.

At the core of our approach lies the question: how can we design a control space that is both compact and musically meaningful? To explore this, we investigate two contrasting strategies.

The first strategy takes a purely data-driven approach: a VAE learns an abstract latent space from rhythmic patterns without incorporating any explicit musical priors. This latent space is compact and supports smooth interpolation between patterns, making it attractive for generative applications. However, because its dimensions are

not explicitly tied to perceptual or musical concepts, it is initially unclear whether this space will support intuitive or controllable user interaction.

The second strategy leverages a set of perceptually grounded rhythm descriptors—such as syncopation, onset density, and balance—as an interpretable control space. Informed by rhythm perception literature, these descriptors allow users to directly influence musically relevant attributes during generation.

By comparing the data-driven latent space learned by the VAE with the perceptually grounded descriptor-based control space, we aim to evaluate the trade-offs between generative expressiveness, interpretability, and user control in interactive rhythm generation.

3.2 Variational Autoencoder Model

We train a VAE model to learn a compact latent space from binary rhythm patterns without relying on predefined musical features. The model aims to discover a low-dimensional, continuous representation that enables smooth interpolation and generation of rhythm patterns.

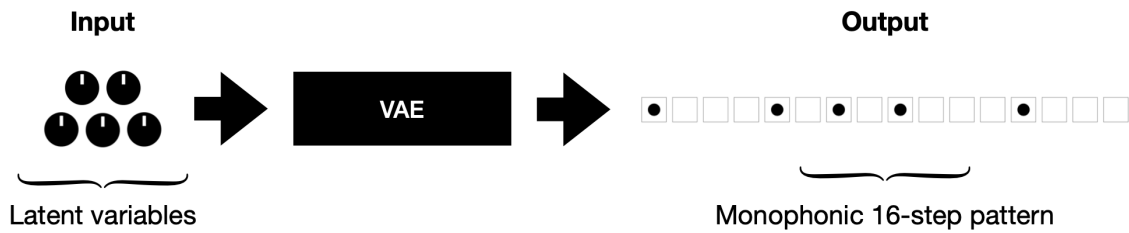


Figure 2: Overview of the Variational Autoencoder model pipeline.

3.2.1 Model Architecture

The autoencoder consists of two main components: an encoder, which maps the 16-step binary rhythm patterns into a compact latent space that captures their essential features, and a decoder, which reconstructs the original patterns from this latent representation while preserving the key rhythmic structure. The following sections provide details on how these two layers have been configured:

Encoder

- Fully connected layer with 64 neurons and ReLU activation.
- Fully connected layer with d latent neurons (no activation).

Decoder

- Fully connected layer with 64 neurons and ReLU activation.
- Fully connected layer with 16 output neurons followed by a sigmoid activation to map outputs into $[0, 1]$ range.

For our experiments, we set the latent space dimensionality to $d = 5$. This value is chosen with a knob-based interface in mind, as five knobs are considered an appropriate number to provide sufficient expressive control while maintaining a minimal and intuitive interface.

3.2.2 Training Procedure

We train the autoencoder using all possible 16-step binary rhythm patterns ($2^{16} = 65,536$ patterns), split into training (70%) and test (30%) sets. The model is optimized using the Binary Cross-Entropy (BCE) loss function, which is suitable for binary data reconstruction. The Adam optimizer is employed with a learning rate of 0.001.

- **Loss function:** Binary Cross-Entropy Loss.
- **Optimizer:** Adam.
- **Epochs:** 400.
- **Batch size:** 32.

We monitor both reconstruction loss and binary reconstruction accuracy, where accuracy is computed as the proportion of correctly reconstructed onset positions after applying a 0.5 threshold on the decoder outputs.

3.2.3 Latent Space Sampling

To assess the generative capability of the model, we sample random points from the latent space and decode them into rhythm patterns. While binary patterns are obtained by applying a threshold of 0.5 to the decoder outputs, we can also experiment with directly using the continuous outputs in the range $[0, 1]$ to represent note velocities. This approach allows the model to generate rhythms with dynamic intensity, capturing expressive variations beyond simple binary events and providing a richer representation of rhythmic nuances.

3.3 Descriptor-based Model

We train a neural network model that maps perceptual rhythm descriptors to 16-step monophonic rhythm patterns. Each descriptor represents a specific structural or perceptual property of rhythm, based on findings from rhythm perception literature.

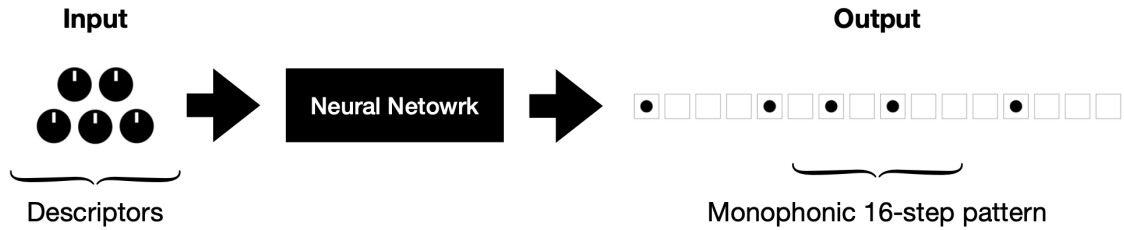


Figure 3: Overview of the Descriptor-based model pipeline.

3.3.1 Rhythm Descriptors

A rhythm descriptor is a function that takes a monophonic rhythm pattern as input and returns a numerical value quantifying a specific structural or perceptual property of that rhythm. Examples include measures of onset density, syncopation, evenness, and balance. To ensure consistency and comparability across features, all descriptors are normalized to lie in the range $[0, 1]$, with 0 representing the minimum expression of a feature and 1 the maximum. This normalization allows the descriptors to be directly used as inputs to computational models, such as neural networks, without the need for additional scaling and ensures that each feature contributes proportionally during learning.

Onset count

Number of active steps in a rhythmic pattern, represented by the number of onsets in a binary sequence. It provides a basic sense of rhythmic density or how many beats are played within the fixed number of steps.

$$\text{onsets} = \{i \in \{1, 2, \dots, 16\} \mid \text{pattern}_i > 0\} \quad \text{nOnsets} = |\text{onsets}|$$

$$\text{dOnsetCount} = \frac{1}{16} \cdot \text{nOnsets}$$

Start

Position of the first onset in the rhythm. It serves as a reference point for timing and can influence the perceived groove or alignment of the rhythm within a measure.

$$\text{dStart} = \frac{1}{16} \cdot \min\{i \mid \text{pattern}_i > 0\}$$

Center

Center of mass of the rhythm across 16 steps. It shows how the weight of the rhythm is distributed in time, helping to identify whether the rhythm feels front-heavy, back-heavy, or balanced.

$$\text{dCenter} = \frac{1}{16} \cdot \frac{1}{\text{nOnsets}} \cdot \sum_{i=1}^{16} i \cdot (\text{pattern}_i > 0)$$

Syncopation

Measures how much a rhythm deviates from a regular metrical pattern. It quantifies the displacement of accents to weaker beats, providing insight into rhythmic complexity and tension (Huron 2006).

$$w = [5, 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1]$$

$$s_i = \max(0, (v_i - v_{(i+1) \bmod 16}) \cdot (w_{(i+1) \bmod 16} - w_i))$$

$$\mathbf{dSyncopation} = \frac{1}{30} \sum_{i=0}^{15} s_i + 15$$

This formula builds on the metrical hierarchy weights proposed by Lerdahl and Jackendoff (1983), later summarized in Toussaint (2013). In this hierarchy, beats at different positions within a 16-step pattern are assigned weights: strong downbeats (5), mid-points (4), half-beats (3), quarter subdivisions (2), and off-beats (1). This hierarchy is reflected in the weight vector w above.

The syncopation score s_i follows the principle that rhythmic tension arises when an onset precedes silence or a weaker onset, but leads into a metrically stronger position. This is conceptually related to Longuet-Higgins and Lee (1984) and its later formalization by Fitch and Rosenfeld (2007), where syncopation is quantified as the difference in metrical weight between an onset and a subsequent rest at a stronger beat. The normalization factor ($\frac{1}{30}$) and offset (+15) ensure that syncopation values fall within a consistent range across 16-step rhythmic patterns.

Syncopation awareness

Refines the basic syncopation metric by factoring in perceptual salience. Each onset is weighted according to its perceived importance or noticeability to human listeners, yielding a measure that aligns more closely with musical perception (Gómez-Marín, Jordà, and Herrera 2015).

$$a = [8, 8, 8, 8, 1, 1, 1, 1, 4, 4, 4, 4, 2, 2, 2, 2]$$

$$\mathbf{dSyncopationAwareness} = \frac{1}{115} \sum_{i=0}^{15} s_i \cdot a_i + 65$$

Evenness

As described in Milne and Herff (2020), and related to the concept of density, the evenness of a rhythm reflects the regularity of interonset intervals¹. Rhythms with lower variance in interonset spacing are considered more even, while higher variance indicates irregularity. Our formulation adapts the geometric approach to rhythmic evenness introduced by Milne and Dean (2016), projecting onset positions onto the unit circle and comparing them against an ideal uniform distribution.

$$\mathbf{dEvenness} = \frac{1}{\mathbf{nOnsets}} \sum_{k=0}^{\mathbf{nOnsets}-1} \left| \cos \left(\frac{2\pi k}{\mathbf{nOnsets}} - \frac{2\pi \cdot \mathbf{onsets}_k}{16} + \frac{2\pi \cdot \mathbf{onsets}_0}{16} \right) \right|$$

Balance

Complementary to evenness, balance measures the symmetry of onset distribution. It is defined as the proximity of the rhythm's center of mass to the center of the unit circle². A rhythm with high balance is spread more symmetrically around the cycle, contributing to greater perceptual stability. Together, evenness and balance describe complementary aspects of rhythmic distribution: one quantifies uniformity of spacing, the other symmetry around the cycle.

$$\mathbf{dBalance} = 1 - \frac{1}{\mathbf{nOnsets}} \sqrt{\left(\sum_{k=0}^{\mathbf{nOnsets}-1} \cos \frac{2\pi \mathbf{onsets}_k}{16} \right)^2 + \left(\sum_{k=0}^{\mathbf{nOnsets}-1} \sin \frac{2\pi \mathbf{onsets}_k}{16} \right)^2}$$

Syness

Combined metric that incorporates both syncopation and the number of onsets. It captures the interplay between rhythmic complexity and density, offering a nuanced view of groove and structure.

¹An interonset interval is the time between two consecutive pulses (Milne and Herff 2020).

²The unit circle is a visualization of a rhythmic cycle where steps are placed around a circle, like positions on a clock (Milne and Herff 2020).

$$\mathbf{dSyness} = \frac{1}{0.633} \cdot \frac{\mathbf{dSyncopationAwareness}}{\mathbf{nOnsets}}$$

The normalization constant $\frac{1}{0.633}$ corresponds to the maximum attainable value of the measure across all 16-step patterns, ensuring that $\mathbf{dSyness} \in [0, 1]$

3.3.2 Discrete descriptor precision

When using analog controls such as knobs, it is possible to obtain highly continuous input values. However, in the case of a digital user interface, we cannot assume access to continuous input signals. Moreover, in the context of our project, maintaining MIDI compatibility is an important requirement.

To account for this, when generating the descriptor dataset we restricted values to the precision of the MIDI range (0–127). This inevitably introduces some redundancy in the data (e.g., multiple descriptor inputs may correspond to the same pattern).

Input Type	Dimensions	Repeated	Unique (%)
MIDI	8	292	99.6
	5	9742	85.1
Float	8	41	99.9
	5	3855	94.1

Table 1: Precision using MIDI and floats to define descriptors.

Table 1 summarizes the impact of using discrete MIDI values versus continuous floating-point values for defining descriptors. When restricting inputs to the MIDI resolution (0–127), a higher number of repeated patterns emerges, particularly in the 5-dimensional case, where over 9,000 repetitions occur and only 85.1% of patterns are uniquely identified. In contrast, using floating-point values greatly reduces redundancy, with nearly all patterns uniquely identified in both the 5 and 8-dimensional cases. These results highlight the trade-off between maintaining MIDI compatibility and achieving finer precision in the descriptor space. In the context of our experiment, we use the dataset in the MIDI range.

3.3.3 Neural Network Implementation

The model is a feedforward neural network trained to regress from rhythm descriptors to rhythm patterns. The input layer takes n descriptor values, and the output layer produces a 16-dimensional vector of logits, which are passed through a sigmoid function to obtain onset probabilities for each of the 16 steps in the rhythm pattern. Again, these probabilities can be interpreted in two ways: (1) by applying a threshold (e.g. 0.5) to produce binary onset predictions indicating the presence or absence of an onset, or (2) by directly mapping the probabilities to velocity values, yielding a continuous representation of onset strength at each step.

Architecture

- **Input layer:** n descriptor values.
- **Hidden layers:** Four fully connected layers with 16, 32, 64, and 32 neurons respectively, each followed by a ReLU activation function.
- **Output layer:** 16 neurons (no activation; sigmoid is applied externally to interpret the outputs as probabilities).

The model is trained using the `BCEWithLogitsLoss` loss function, and optimized with the Adam optimizer. Training is performed for 200 epochs with mini-batches of size 32.

Evaluation Methods

To assess the quality of the model’s output, we employ two distinct evaluation strategies that reflect different aspects of rhythmic similarity:

- **Pattern-based accuracy:** the predicted output is thresholded at 0.5 to obtain a binary pattern, which is then directly compared to the ground-truth binary rhythm pattern. Accuracy is computed as the proportion of correctly predicted onset positions across all steps and examples.

$$\text{Score} = \frac{1}{16} \sum_{i=1}^{16} \mathbf{1}[\hat{p}_i = p_i]$$

- **Descriptor-based accuracy:** the predicted binary rhythm patterns are post-processed to compute their descriptors (using the same feature extraction method as the input). Accuracy is computed based on the similarity (e.g., inverse normalized error) between the predicted descriptors and the original input descriptors. This version prioritizes perceptual or structural similarity over exact pattern match.

$$\text{Score} = 1 - \frac{1}{8} \sum_{j=1}^8 |\hat{d}_i - d_i|$$

Further experiments will be conducted to determine which of the two evaluation approaches is more suitable for our use case.

3.3.4 Dimensionality Reduction

To investigate the effectiveness of a reduced descriptor space for rhythm pattern generation, we conduct experiments using dimensionality reduction by systematically removing features from the full descriptor set. The motivation behind this process is to enable a more usable and compact control space—particularly relevant in interactive or hardware-based systems, such as knob-based interfaces, where having too many dimensions can hinder intuitive control. A smaller, well-chosen set of descriptors would allow for more expressive yet manageable manipulation of rhythm generation, enhancing both user experience and creative flexibility. This experiment is conducted using only the pattern-based accuracy.

Specifically, we employ a leave- k -out approach, where $k \in \{1, 2, 3\}$. In each case, k descriptors are excluded from the input feature set, and the model is retrained and evaluated using the remaining $n - k$ descriptors. This process allows us to evaluate how much each descriptor (or group of descriptors) contributes to the performance of the model, and whether a smaller subset of descriptors can still preserve predictive

quality.

For each value of k , we test all possible combinations of descriptor subsets:

- **Leave-One-Out (L1O)**: We train and evaluate n models, each omitting a different single descriptor.
- **Leave-Two-Out (L2O)**: We train and evaluate $\binom{n}{2}$ models, each omitting a pair of descriptors.
- **Leave-Three-Out (L3O)**: We train and evaluate $\binom{n}{3}$ models, each omitting a triplet of descriptors.

This exhaustive evaluation provides insight into redundancy, complementarity, and relative importance of the rhythm descriptors. The results from these experiments are used to inform feature selection strategies and to identify minimal yet effective descriptor subsets for interpretable and efficient rhythm generation.

3.3.5 Model Evaluation

We evaluate the model using both pattern-based and descriptor-based accuracy.

Using all eight rhythm descriptors, the model achieves a **pattern-based accuracy** of **78.35%**, meaning that the predicted binary rhythm patterns match the ground-truth patterns at this rate when applying a 0.5 threshold to the output probabilities.

For **descriptor-based accuracy**, where we compared the descriptors of the predicted rhythms to the original input descriptors, the model reached a much higher accuracy of **93.80%**. This indicates that even when exact rhythm patterns are not perfectly reconstructed, the structural and perceptual properties of the rhythms are well preserved.

Following the dimensionality reduction experiments (leave-one-out, leave-two-out, and leave-three-out), we identify a minimal subset of five descriptors that preserve high predictive performance while reducing model complexity. The selected descriptors are:

```
onset_count  start  center  syncopation  balance
```

Using this reduced set of five descriptors, the model achieves a **pattern-based accuracy** of **74.05%** and a **descriptor-based accuracy** of **92.24%**. These results demonstrate that the smaller descriptor set retains sufficient information for effective rhythm generation, maintaining high performance while offering a more compact and interpretable control space.

From now on, when we refer to the descriptor-based model, we are referring to this specific five-descriptor configuration.

3.4 Comparing VAE and descriptor-based approaches

To enable real-time interaction with our models, we develop an interactive Pure Data patch in conjunction with a Python server capable of handling model requests. Communication between the two components is established using the Open Sound Control (OSC) protocol.

The Pure Data patch includes a sequencer that plays the onsets of a given pattern and supports features such as toggling velocity and enabling a metronome. It also provides interactive controls for all descriptors or latent variables, depending on the currently selected model. Finally, a Bézier curve visualization is provided to represent the onset distribution of the pattern.

In Figure 4 we can see a part of the Pure Data patch used to interact for the models—in this case we see sliders that control the descriptor-based model, but this patch is also capable of using the VAE with the 5 latent variables as well. On the bottom, we see a preview of the pattern generated: we can see the probability for each step as well as a Bézier curve visualization of the pattern.

The terminal screenshot in Figure 5 shows some logs in the Python server that handles the OSC requests between the Pure Data patch and the PyTorch models.

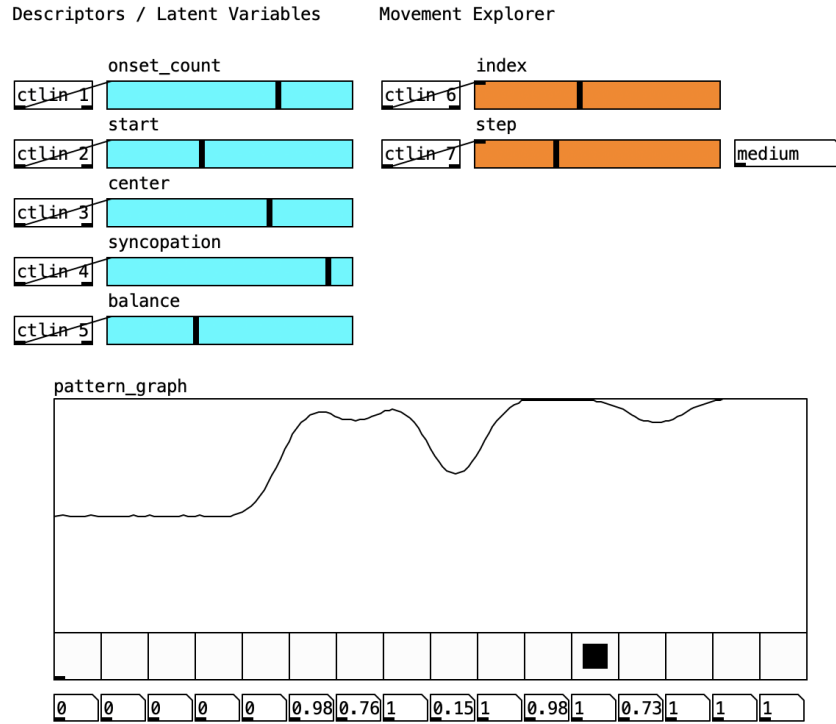


Figure 4: Screenshot of the interactive Pure Data patch.

```

Received descriptors: [0.27, 0.01, 0.51, 0.13, 0.09] from /genpattern
Sending response: [1.0, 0.0, 0.99, 0.13, 0.61, 0.0, 0.01, 0.0, 0.06, 0.0, 0.17, 0.04, 0.93, 0.07, 0.8, 0.35]
Received request for interpolation 1415, step 15 from /interp
Sending interpolation step: [33.95, 1.51, 66.57, 15.23, 11.1] with distance category: small
Received descriptors: [0.27, 0.01, 0.51, 0.13, 0.09] from /genpattern
Sending response: [1.0, 0.0, 0.99, 0.13, 0.61, 0.0, 0.01, 0.0, 0.06, 0.0, 0.17, 0.04, 0.93, 0.07, 0.8, 0.35]

```

Figure 5: Screenshot of some logs in the Python server terminal.

Through hands-on interaction with both the descriptor-based model and the VAE model using this interface, we observed significant differences in usability and intuitiveness.

While the VAE-based model offers smooth transitions in the rhythmic patterns due to its continuous latent space, it proves difficult to control in a musically meaningful way. The latent variables lack clear interpretability, making it challenging for us to predict or intentionally shape the resulting rhythms when adjusting the controls. In contrast, the descriptor-based model provides a much more intuitive interaction. Each descriptor corresponds to a well-defined musical property, allowing us to directly influence specific aspects of the rhythm (e.g., onset_count, syncopation, balance).

This observation is consistent with previous studies on VAEs for generative music applications, where the abstract nature of the latent space often limits direct user control and hinders intuitive interaction. For instance, in the domain of timbre modeling, Esling, Chemla-Romeu-Santos, and Bitton 2018 show that VAEs can indeed construct continuous timbre spaces that enable smooth interpolation between instruments. However, these latent spaces remain difficult to interpret and do not directly correspond to perceptually meaningful properties. To address this, they propose incorporating audio descriptors to regularize the latent space, allowing for descriptor-based synthesis and a closer alignment with perceptual ratings.

This parallel reinforces our findings: while VAEs provide smoothness and continuity, they lack sufficient interpretability for end users to effectively guide the generation process. In contrast, descriptor-based models make the mapping between controls and perceptual properties explicit, enabling purposeful exploration and more musically meaningful interaction.

Based on these results, we decide to focus exclusively on the descriptor-based model for the remainder of our study and applications, as it offers a better balance between control, expressiveness, and usability.

3.5 Smoothness Experiment

Having established that descriptor-based control is more interpretable and musically meaningful, we now turn to evaluating which of the possible descriptor-based models produces the smoothest transitions in the generated rhythmic patterns.

Specifically, we analyze how gradual changes in the descriptor values translate into perceptually smooth changes in the rhythms. To quantify this, we test two variants of descriptor-based models: one trained using descriptor-based accuracy and another trained using pattern-based accuracy. The outputs are evaluated using two complementary error metrics: the binary pattern distance and the Kullback-Leibler (KL) divergence between consecutive rhythmic patterns. These metrics provide a quantitative measure of the smoothness of the model’s generated transitions.

We use this smoothness metric as the primary criterion for selecting the better model—specifically, we compare the model trained using **descriptor-based accuracy** to the one trained using **pattern-based accuracy**. The model that yields smoother transitions is preferred, as it better supports continuous and intuitive control over the generated rhythms.

3.5.1 Experiment Design

We investigate the smoothness of the mapping from descriptor space to rhythmic patterns by defining a large set of continuous *movements* within the 5-dimensional descriptor space. Each movement corresponds to a linear interpolation between two points:

- **Start point** (x_{start}): A randomly sampled descriptor vector within the valid range $[0, 1]^5$, respecting any domain-specific constraints.
- **End point** (x_{end}): A point located at a fixed Euclidean distance from the start point, where only *two* of the five descriptors are varied. This constraint reflects typical user interaction scenarios where at most two control knobs are adjusted simultaneously, making movements more interpretable and practically relevant.

We categorize movements into three groups based on the magnitude of the Euclidean distance d between start and end points:

- **Small movement:** $d = \sqrt{2} \times 0.25$.
- **Medium movement:** $d = \sqrt{2} \times 0.5$.
- **Large movement:** $d = \sqrt{2} \times 0.75$.

For each movement, we generate N intermediate descriptor vectors by linear interpolation:

$$x_t = (1 - \alpha_t) \cdot x_{\text{start}} + \alpha_t \cdot x_{\text{end}} \quad \text{with} \quad \alpha_t = \frac{t}{N-1} \quad \text{for} \quad t = 0, \dots, N-1.$$

At each interpolation step x_t , the trained model generates a corresponding 16-step rhythmic pattern.

To quantitatively assess smoothness along each movement, we analyze consecutive pairs of generated patterns using two complementary metrics:

- **Pattern distance (KL divergence):** we compute the Kullback-Leibler (KL) divergence between the probabilistic onset distributions of consecutive patterns, capturing differences in their rhythmic structure beyond binary similarity.
- **Descriptor distance (Euclidean distance):** we compute the Euclidean distance between the descriptor vectors estimated from consecutive patterns, measuring how much the underlying control parameters change in the model's output space.

By tracking these metrics across the $N - 1$ transitions in each movement, we obtain a detailed profile of how smoothly small control changes translate into pattern variations. This analysis allows us to compare the behavior of different descriptor-based models under controlled, interpretable manipulations.

In Figure 6, we show a graphical representation of a movement in the descriptor space, as defined in the smoothness experiment. In this example, the changing descriptors are `onset_count` and `syncopation`. The distance of this movement can be expressed as:

$$d = \sqrt{(oc_A - oc_B)^2 + (sy_A - sy_B)^2}.$$

This corresponds to the Euclidean distance between the initial pattern (A) and the final pattern (B).

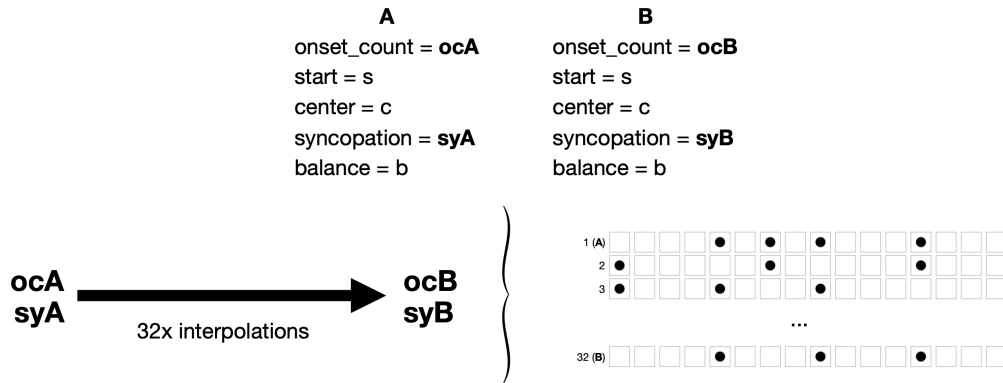


Figure 6: Visualization of a movement in the descriptor space.

3.5.2 Implementation

The experiment is implemented in a Python notebook that automatically generates a large number of random movements across the three distance categories, computes the above metrics, and saves the generated pattern sequences.

We extend the Pure Data patch to support the simulation of these movements in the descriptor space. As shown in Figure 4, the two orange sliders allow users to navigate through the set of pre-generated movements and explore the interpolations interactively. The *index* slider selects the specific movement to be simulated, while the *step* slider controls the interpolation position along the movement. As the step slider is adjusted, the corresponding descriptor values smoothly interpolate from the start to the end point, and users can listen in real time to the gradual transformation of the generated rhythmic pattern.

3.6 User Experience Experiment

To evaluate the usability and effectiveness of the system, we develop a fully web-based study where participants attempt to replicate reference rhythmic patterns using descriptor sliders.

3.6.1 Implementation

A web-based format is chosen to maximize accessibility and scalability, allowing a broader and more diverse pool of participants to take part in the study remotely. This approach also simplifies deployment and reduces setup time, enabling efficient collection of larger amounts of data across different devices and environments.

Since the original interactive system is developed as a Pure Data patch, we re-implement it for the web. The webapp is built using the following stack:

- **React:** for building the interactive user interface.³
- **Tone.js:** for sequencing and playing samples.⁴
- **ONNX Runtime:** to execute our final PyTorch model within the browser.⁵

This architecture ensures that pattern generation and audio playback are executed locally within the browser, minimizing latency and ensuring a smooth user experience.

3.6.2 Structure

The experiment consists of the following stages:

- **Background questionnaire:** participants provide demographic information, including age range and musical experience (years of study, performance, and percussion-specific experience).
- **Eight exercises:** in each exercise, participants listen to a target rhythmic pattern (*Pattern A*) and attempt to replicate it using the provided sliders, which control rhythmic descriptors such as `onset_count`, `start`, `center`, `syncopation`, and `balance`. After submitting their solution (*Pattern B*), they rate

³<https://react.dev>

⁴<https://tonejs.github.io>

⁵<https://onnxruntime.ai>

the perceived similarity between the two patterns. The two patterns are randomized but they are always within a medium distance (as defined in the smoothness experiment).

- **Final feedback:** participants offer their overall impressions regarding the user interface, the clarity of the task, and the difficulty of controlling each descriptor, along with open-ended comments about confusing aspects, liked features and suggestions for improvement.

3.6.3 Data Collection and Storage

Throughout the study, a range of objective and subjective data points are collected for each participant and stored privately in Google Sheets using the App Scripts API. Some of these fields are directly submitted by the participants, while others derive from processing the experimental data (e.g., objective and parametric similarity metrics or elapsed time). No personal or identifiable information is collected, and all stored data is processed with the informed consent of the participants.

The collected data includes:

- **Background information:**
 - Participant ID
 - Age range (18-25, 26-35, 36-45, 46-55, 56+)
 - Years of musical study (0-4+ scale)
 - Years of musical performance (0-4+ scale)
 - Years spent performing percussion (0-4+ scale)
- **Exercise data:**
 - Exercise number (1-8)
 - Target pattern (16-step pattern)
 - Submitted pattern (16-step pattern)

- Target descriptors (5 descriptor values)
 - Initial descriptors (5 descriptor values)
 - Final descriptors (5 descriptor values)
 - Elapsed time (trial duration in seconds)
 - Subjective similarity (0-5 scale)
 - Objective similarity (euclidean distance between patterns)⁶
 - Parametric similarity (euclidean distance between descriptors)⁷
- **Final feedback:**
- Ratings of interface intuitiveness and difficulty per descriptor (0–5 scale).
 - Confusing aspects of the interface (free text)
 - Liked aspects of the interface (free text)
 - Feedback, suggestions, or ideas (free text)

Figures 7 to 11 illustrate key stages of the user study, including the introduction, background questionnaire, exercise interface, and feedback questions.

⁶Defined by the euclidean distance between the presented pattern and the one submitted by the user.

⁷Computed with the euclidean distance between the initial set of descriptors and the final one defined by the user.

Welcome! In this experiment, you'll explore a rhythm generation tool powered by machine learning. Your task is to learn how it works, complete challenges, and share feedback. The experiment will take around 15 minutes and it is important that you give it your full attention and focus.

Consent and Data Use Notice

By participating in this experiment, you agree that your interaction data (e.g., slider values, patterns you create, and timing data) may be anonymously recorded and analyzed for academic research purposes. No personally identifiable information is collected. All data will be stored securely and used solely for the purpose of understanding user interaction with rhythm generation tools. Participation is voluntary, and you may stop at any time by closing the browser window.

For questions, contact alex@vilanova.dev

☒ I agree to participate in this study.

Age range:

☒ 18-25 ☐ 26-35 ☐ 36-45 ☐ 46-55 ☐ 56+

Number of years spent studying music:

☐ 0 ☒ 1 ☐ 2 ☐ 3 ☐ 4+

Number of years spent performing music:

☐ 0 ☐ 1 ☐ 2 ☒ 3 ☐ 4+

Number of years of percussion experience:

☐ 0 ☐ 1 ☒ 2 ☐ 3 ☐ 4+

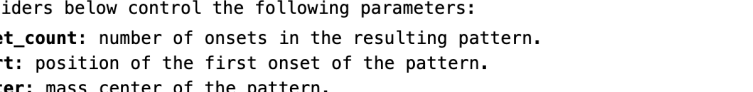
Next

Figure 7: User experience experiment: intro and background questionnaire.

#0 Play around with the sliders, get familiar with the interface.

The sliders below control the following parameters:

- **onset_count**: number of onsets in the resulting pattern.
- **start**: position of the first onset of the pattern.
- **center**: mass center of the pattern.
- **syncopation**: how syncopated are the onsets in the pattern.
- **balance**: how well distributed are the onsets in the pattern.



onset_count start center syncopation balance

tempo metronome velocity


stop

Next

Figure 8: User experience experiment: familiarity step.


[1/8]

#1 Listen to this pattern.



#2 Use the sliders below to transform the next pattern into the one you just listened to (#1). Try to get as close as possible, but don't worry if they're not identical. Click "Done" when you're ready.

onset_count start center syncopation balance



play

Done

Figure 9: User experience experiment: exercise interaction interface.

1/8

#1 Listen to this pattern.

play

#2 Use the sliders below to transform the next pattern into the one you just listened to (#1). Try to get as close as possible, but don't worry if they're not identical. Click "Done" when you're ready.

play

How similar do you think the two patterns are?

☐ 0
☐ 1
☐ 2
☒ 3
☐ 4
☐ 5

Next

Figure 10: User experience experiment: exercise feedback form.

#17 How intuitive was the interface?

☐ 0 ☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

#18 How difficult was it to use the sliders?

onset_count	<input checked="" type="radio"/> 0	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5
start	<input type="radio"/> 0	<input checked="" type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5
center	<input type="radio"/> 0	<input checked="" type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5
syncopation	<input type="radio"/> 0	<input type="radio"/> 1	<input type="radio"/> 2	<input checked="" type="radio"/> 3	<input type="radio"/> 4	<input type="radio"/> 5
balance	<input type="radio"/> 0	<input type="radio"/> 1	<input type="radio"/> 2	<input type="radio"/> 3	<input checked="" type="radio"/> 4	<input type="radio"/> 5

#19 What was the most confusing part of the interface?

...

#20 What did you like about the interface?

...

#21 Do you have any feedback, suggestions, or ideas?

...

Finish

Figure 11: User experience experiment: final feedback questionnaire.

Chapter 4

Results

4.1 Dimensionality Reduction

We systematically performed a series of leave- k -out experiments, where we removed one, two, or three descriptors at a time and measured the classification accuracy. Tables 2, 3, and 4 summarize the results for the leave-one-out (L1O), leave-two-out (L2O), and leave-three-out (L3O) tests, respectively. Each row in these tables corresponds to one experimental configuration, listing the descriptors used and the resulting classification accuracy.

Id	Acc. (%)	Descriptors
1	79.13	onset_count, start, center, syncopation, syncopation_awareness, evenness, balance
2	77.68	onset_count, start, center, syncopation, syncopation_awareness, evenness, syness
3	78.83	onset_count, start, center, syncopation, syncopation_awareness, balance, syness
4	78.38	onset_count, start, center, syncopation, evenness, balance, syness
5	74.92	onset_count, start, center, syncopation_awareness, evenness, balance, syness
6	76.22	onset_count, start, syncopation, syncopation_awareness, evenness, balance, syness
7	76.06	onset_count, center, syncopation, syncopation_awareness, evenness, balance, syness
8	78.94	start, center, syncopation, syncopation_awareness, evenness, balance, syness

Table 2: Leave-one-out (L1O) experiment results.

Id	Acc. (%)	Descriptors
1	77.66	onset_count, start, center, syncopation, syncopation_awareness, evenness
2	79.16	onset_count, start, center, syncopation, syncopation_awareness, balance
3	77.14	onset_count, start, center, syncopation, syncopation_awareness, synness
4	74.29	onset_count, start, center, syncopation, evenness, balance
5	77.05	onset_count, start, center, syncopation, evenness, synness
6	78.20	onset_count, start, center, syncopation, balance, synness
7	74.68	onset_count, start, center, syncopation_awareness, evenness, balance
8	73.59	onset_count, start, center, syncopation_awareness, evenness, synness
9	74.66	onset_count, start, center, syncopation_awareness, balance, synness
10	74.67	onset_count, start, center, evenness, balance, synness
11	76.11	onset_count, start, syncopation, syncopation_awareness, evenness, balance
12	74.87	onset_count, start, syncopation, syncopation_awareness, evenness, synness
13	74.42	onset_count, start, syncopation, syncopation_awareness, balance, synness
14	75.54	onset_count, start, syncopation, evenness, balance, synness
15	71.90	onset_count, start, syncopation_awareness, evenness, balance, synness
16	76.55	onset_count, center, syncopation, syncopation_awareness, evenness, balance
17	74.31	onset_count, center, syncopation, syncopation_awareness, evenness, synness
18	73.03	onset_count, center, syncopation, syncopation_awareness, balance, synness
19	75.48	onset_count, center, syncopation, evenness, balance, synness
20	72.46	onset_count, center, syncopation_awareness, evenness, balance, synness
21	70.21	onset_count, syncopation, syncopation_awareness, evenness, balance, synness
22	77.04	start, center, syncopation, syncopation_awareness, evenness, balance
23	77.68	start, center, syncopation, syncopation_awareness, evenness, synness
24	78.47	start, center, syncopation, syncopation_awareness, balance, synness
25	75.00	start, center, syncopation, evenness, balance, synness
26	74.79	start, center, syncopation_awareness, evenness, balance, synness
27	76.27	start, syncopation, syncopation_awareness, evenness, balance, synness
28	76.15	center, syncopation, syncopation_awareness, evenness, balance, synness

Table 3: Leave-two-out (L2O) experiment results.

Id	Acc. (%)	Descriptors
1	77.32	onset_count, start, center, syncopation, syncopation_awareness
2	73.02	onset_count, start, center, syncopation, evenness
3	74.05	onset_count, start, center, syncopation, balance
4	76.58	onset_count, start, center, syncopation, synness
5	73.56	onset_count, start, center, syncopation_awareness, evenness
6	74.58	onset_count, start, center, syncopation_awareness, balance
7	72.91	onset_count, start, center, syncopation_awareness, synness
8	71.16	onset_count, start, center, evenness, balance
9	73.58	onset_count, start, center, evenness, synness
10	74.47	onset_count, start, center, balance, synness
11	74.91	onset_count, start, syncopation, syncopation_awareness, evenness
12	74.31	onset_count, start, syncopation, syncopation_awareness, balance
13	73.65	onset_count, start, syncopation, syncopation_awareness, synness
14	70.78	onset_count, start, syncopation, evenness, balance
15	74.30	onset_count, start, syncopation, evenness, synness
16	73.62	onset_count, start, syncopation, balance, synness
17	71.92	onset_count, start, syncopation_awareness, evenness, balance
18	70.57	onset_count, start, syncopation_awareness, evenness, synness
19	69.94	onset_count, start, syncopation_awareness, balance, synness
20	71.77	onset_count, start, evenness, balance, synness
21	74.77	onset_count, center, syncopation, syncopation_awareness, evenness
22	73.06	onset_count, center, syncopation, syncopation_awareness, balance
23	71.99	onset_count, center, syncopation, syncopation_awareness, synness
24	71.77	onset_count, center, syncopation, evenness, balance
25	74.19	onset_count, center, syncopation, evenness, synness
26	72.63	onset_count, center, syncopation, balance, synness
27	72.59	onset_count, center, syncopation_awareness, evenness, balance
28	70.39	onset_count, center, syncopation_awareness, evenness, synness
29	68.58	onset_count, center, syncopation_awareness, balance, synness
30	72.28	onset_count, center, evenness, balance, synness
31	70.11	onset_count, syncopation, syncopation_awareness, evenness, balance
32	69.34	onset_count, syncopation, syncopation_awareness, evenness, synness
33	68.40	onset_count, syncopation, syncopation_awareness, balance, synness
34	69.74	onset_count, syncopation, evenness, balance, synness
35	65.77	onset_count, syncopation_awareness, evenness, balance, synness
36	75.22	start, center, syncopation, syncopation_awareness, evenness
37	75.36	start, center, syncopation, syncopation_awareness, balance
38	77.15	start, center, syncopation, syncopation_awareness, synness
39	72.73	start, center, syncopation, evenness, balance
40	73.57	start, center, syncopation, evenness, synness
41	74.08	start, center, syncopation, balance, synness
42	73.14	start, center, syncopation_awareness, evenness, balance
43	73.56	start, center, syncopation_awareness, evenness, synness
44	74.48	start, center, syncopation_awareness, balance, synness
45	72.84	start, center, evenness, balance, synness
46	72.86	start, syncopation, syncopation_awareness, evenness, balance
47	74.77	start, syncopation, syncopation_awareness, evenness, synness
48	74.26	start, syncopation, syncopation_awareness, balance, synness
49	71.08	start, syncopation, evenness, balance, synness
50	71.99	start, syncopation_awareness, evenness, balance, synness
51	73.75	center, syncopation, syncopation_awareness, evenness, balance
52	74.48	center, syncopation, syncopation_awareness, evenness, synness
53	72.84	center, syncopation, syncopation_awareness, balance, synness
54	72.33	center, syncopation, evenness, balance, synness
55	72.25	center, syncopation_awareness, evenness, balance, synness
56	70.04	syncopation, syncopation_awareness, evenness, balance, synness

Table 4: Leave-three-out (L3O) experiment results.

4.2 Smoothness Experiment

Figure 12 presents the results of the smoothness experiment. We show both the accumulated KL divergence and the KL divergence distribution for the two models—one using pattern distance as the error metric and the other using distances in the descriptor space—using a scatter plot and a box plot, respectively. Similarly, we display the accumulated Euclidean distance and the Euclidean distance distribution for both models, again using scatter and box plots, respectively.

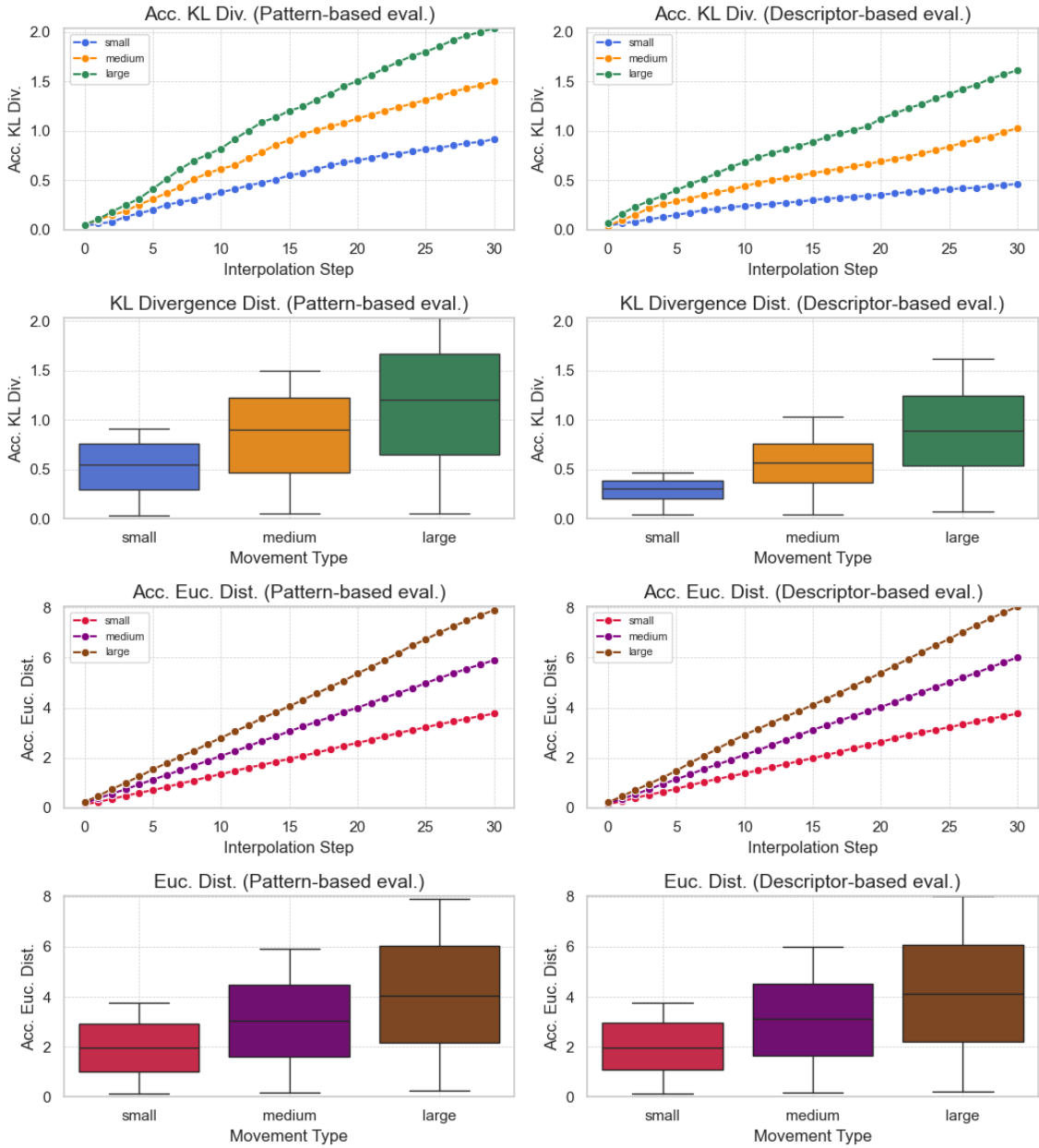


Figure 12: Smoothness experiment plots.

Tables 5 and 6 summarize the statistical analysis of differences between the two models across the three different kinds of movements. Table 5 presents the results of an ANOVA test for the two distance metrics—KL Divergence and Euclidean Distance—showing the corresponding p-values for each model. All p-values are highly significant, indicating that there are statistically significant differences among the groups for both pattern and descriptor distances. Table 6 provides a post-hoc analysis using Tukey’s HSD test to determine which specific group pairs differ significantly. The table lists the mean differences, p-values, and significance indicators for all pairwise comparisons among the *small*, *medium*, and *large* groups. Overall, the results show that most group comparisons are significant, particularly for KL Divergence, while a few Euclidean Distance comparisons are not statistically significant.

Model	Metric	p-value	Significant
Pattern distance	KL Divergence	0.00000903	✓
	Euclidean Distance	0.0000683	✓
Descriptor distance	KL Divergence	0.000000000150	✓
	Euclidean Distance	0.0000719	✓

Table 5: ANOVA p-values for KL Divergence and Euclidean Distance.

Model	Metric	Group 1	Group 2	Mean Diff.	p-value	Significant
Pattern distance	KL Divergence	large	medium	-0.2954	0.0412	✓
		large	small	-0.6174	0.0000	✓
		medium	small	-0.3220	0.0233	✓
Pattern distance	Euclidean Distance	large	medium	-1.0466	0.0652	
		large	small	-2.1315	0.0000	✓
		medium	small	-1.0849	0.0537	
Descriptor distance	KL Divergence	large	medium	-0.3255	0.0002	✓
		large	small	-0.6026	0.0000	✓
		medium	small	-0.2771	0.0019	✓
Descriptor distance	Euclidean Distance	large	medium	-1.0516	0.0663	
		large	small	-2.1416	0.0000	✓
		medium	small	-1.0900	0.0546	

Table 6: Tukey’s HSD test for KL Divergence and Euclidean Distance.

4.3 User Experience Experiment

Figures 13 and 14 show the distributions of participants' background and similarity data, respectively. Figure 13 presents, from left to right, the distributions of age ranges and musical experience (years spent studying, performing music, and specifically performing percussion). Figure 14 provides an overview of the three similarity metrics (subjective, objective, and parametric) across all exercise takes from every participant.

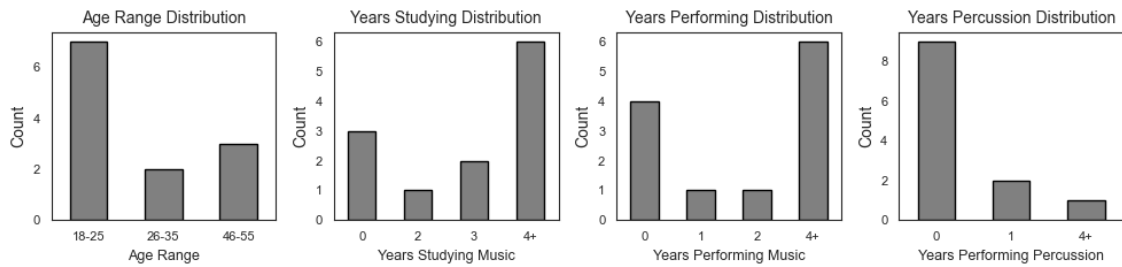


Figure 13: Age range and music experience distributions.

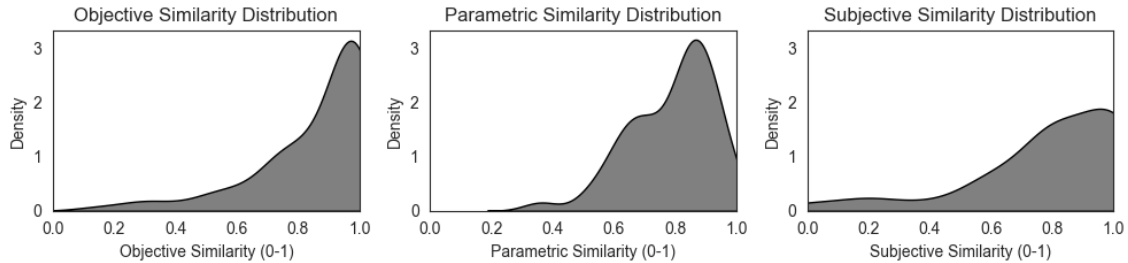


Figure 14: Pattern similarity distributions.

Figure 14 visualizes the correlations between the different types of similarity. The gray dots represent the data points, with a red regression line overlaid. In the top-left corner of each plot, a legend reports the correlation coefficient and p-value. We used Spearman's correlation for the first two cases and Pearson's correlation for the last one, since subjective similarity is an integer variable (0–5), while both objective and parametric similarities are non-categorical. Statistical significance was determined using the threshold of $p < 0.05$. All correlations were strong and statistically significant, especially between Subjective and Objective similarities.

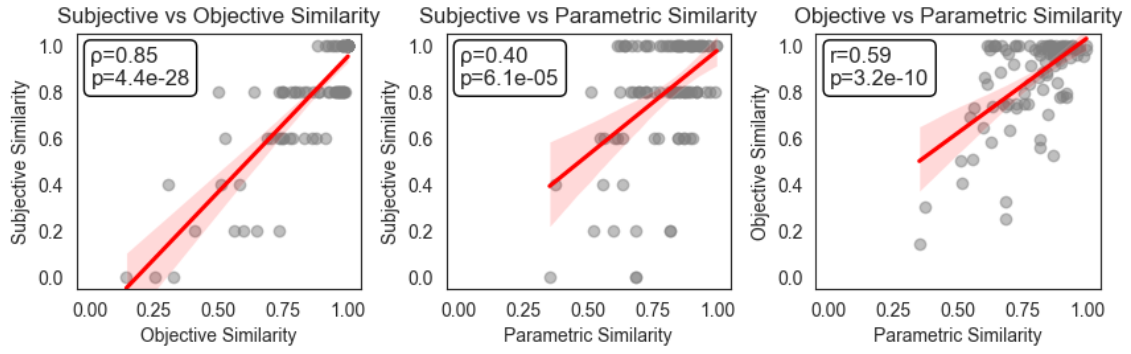


Figure 15: Pattern similarity correlations.

Table 7 summarizes the correlations between participant variables (elapsed time, age range, music study, music performance, and percussion performance) and three types of similarity measures. For each combination, the table reports the correlation coefficient, the associated p-value, and whether the correlation is statistically significant. Spearman’s correlation was used in all cases, except for Elapsed Time vs. Objective and Parametric similarities, where Pearson’s correlation was applied since the data is non-categorical. Statistical significance was again determined using a threshold of $p < 0.05$.

Variable	Similarity	Correlation	p-value	Significant
Elapsed Time	Subjective	-0.19	0.060	
	Objective	-0.22	0.031	✓
	Parametric	-0.14	0.190	
Age Range	Subjective	-0.04	0.690	
	Objective	-0.05	0.650	
	Parametric	-0.10	0.320	
Music Study	Subjective	-0.20	0.047	✓
	Objective	-0.18	0.076	
	Parametric	-0.04	0.720	
Music Performance	Subjective	-0.14	0.160	
	Objective	-0.04	0.690	
	Parametric	-0.09	0.380	
Percussion Performance	Subjective	0.14	0.160	
	Objective	0.12	0.260	
	Parametric	0.05	0.660	

Table 7: Correlation results between various variables and similarity measures.

We also asked participants to rate the perceived difficulty of each descriptor. Figures 16 and 17 present insights from this data. Figure 16 shows the overall distribution of relative difficulty across all descriptors, normalized to a 0–100% scale on the Y-axis. Figure 17 displays the distributions of difficulty ratings for each descriptor individually.

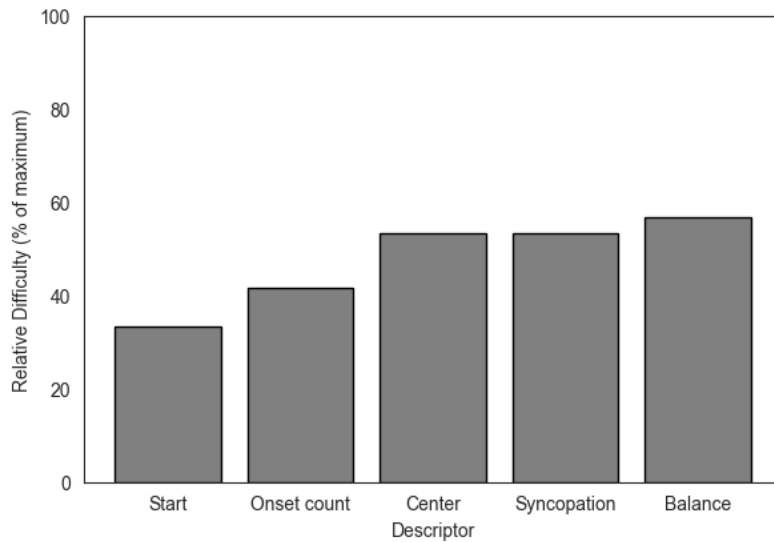


Figure 16: Distribution of the perceived difficulty for each descriptor.

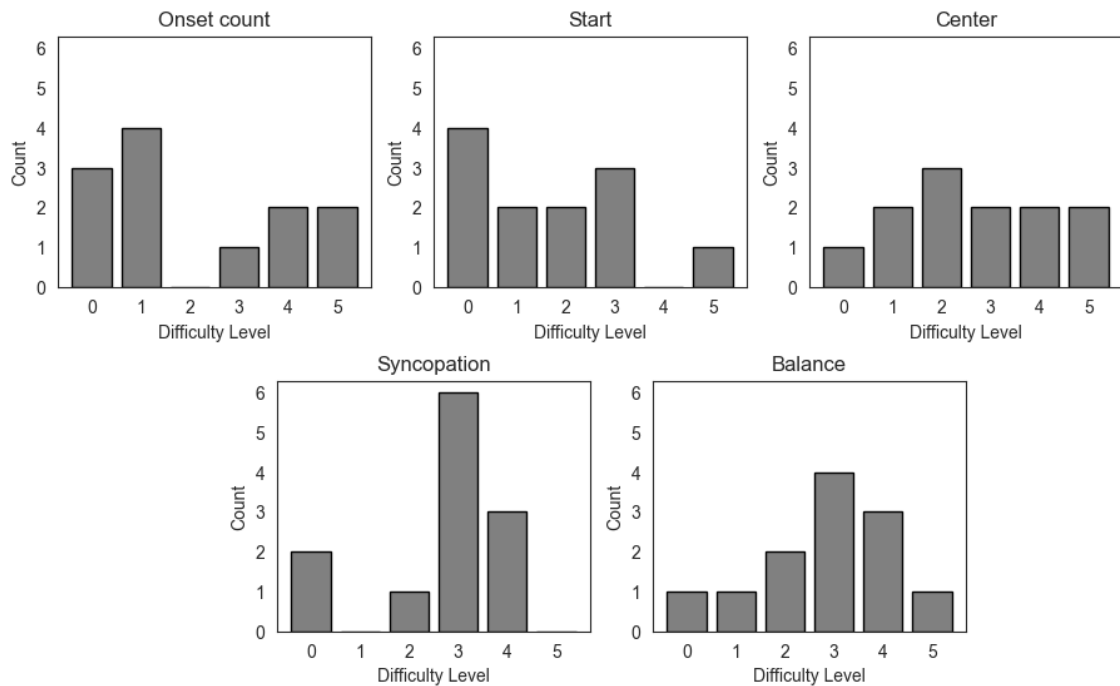


Figure 17: Descriptors ranked by relative difficulty.

In addition to quantitative ratings, we conducted a qualitative analysis of user feedback to capture broader insights into their experience. Tables 8, 9, and 10 summarize these findings. Table 8 outlines aspects of the system that participants particularly appreciated, as well as elements they found problematic. Table 9 distills recurring usability findings into key observations. Finally, Table 10 highlights specific feature requests that emerged, reflecting users’ ideas for improving functionality and support.

Aspect	Liked	Disliked
Interface Design	Simplicity, clarity, visual feedback; clean, intuitive appearance; fun and engaging to use	Complex parameter labels; lack of explanatory text
Interaction	Real-time auditory feedback; immediate visual response; exploratory learning through experimentation	Difficulty understanding slider effects; confusing parameter relationships; unclear mapping between controls and output
Technical Parameters	Start parameter and onset_count intuitive	Syncopation slider confusion; balance parameter unclear
Learning Experience	Hands-on discovery process	Lack of tutorials or guidance; missing tooltips or help system

Table 8: Summary of liked and disliked aspects from user feedback.

Usability Finding	Observation
Slider complexity	New users experienced difficulty with technical parameters
Immediate feedback	Visual and auditory feedback enhanced user engagement
Exploratory learning	Users learned through hands-on experimentation
Interface clarity	Simplicity and clean design facilitated usage
Educational support needs	Users requested guidance tools for complex features

Table 9: Key usability observations from user testing.

Suggested Feature	User Request
Tooltips/hover explanations	Contextual help for slider parameters
Tutorial system	Interactive guidance and onboarding
Visual parameter mapping	Connection between slider movements and rhythm changes
Adaptive interface	Different complexity levels for different user types
Additional controls	Swing parameter, instrument selection, sound textures

Table 10: Feature requests from user feedback.

Chapter 5

Discussion

In this section, we analyze and interpret the data collected in the results, examining the performance, usability, and perceptual relevance of the different rhythm generation approaches. By discussing both quantitative and qualitative findings, we aim to highlight the strengths, limitations, and practical implications of our descriptor-based rhythm generation approach.

5.1 VAE vs. Descriptor-Based Approaches

The preliminary comparison between VAE and descriptor-based models validates the central hypothesis of this research. While the VAE approach demonstrated smooth latent space interpolation, it failed to provide meaningful user control due to abstract latent dimensions that made it impossible to predict or intentionally influence specific rhythmic properties.

The descriptor-based model, enabled purposeful exploration and predictable outcomes. This trade-off between perfect smoothness and interpretable control favors interactive applications prioritizing user agency. The success reinforces the value of incorporating domain knowledge from music perception research rather than relying solely on data-driven feature learning.

5.2 Descriptor Selection

The leave-k-out experiments (Tables 2, 3, and 4) revealed important insights about descriptor redundancy and the minimal feature set required for effective rhythm generation. The results demonstrate that a reduced set of five descriptors (onset_count, start, center, syncopation and balance) maintains strong predictive performance while significantly simplifying the control space.

The transition from eight to five descriptors resulted in only a modest decrease in pattern-based accuracy (from 78.35% to 74.05%, as seen in Table 4) while maintaining high descriptor-based accuracy (92.24%). This finding is particularly significant for practical applications, as it suggests that three descriptors (syncopation awareness, evenness, and synness) contribute relatively little unique information beyond what is captured by the core five features.

The superior performance of descriptor-based accuracy (92.24%) compared to pattern-based accuracy (74.05%) indicates that the model successfully learns to preserve the perceptual and structural properties of rhythms even when exact pattern reconstruction is imperfect. This aligns with music cognition research suggesting that human rhythm perception is more tolerant of surface-level variations when underlying structural relationships are maintained.

5.3 Smoothness Analysis

The smoothness experiment (Figure 12) provided evidence for model selection between the pattern-based and descriptor-based training approaches. The ANOVA results in Table 5 confirm that the descriptor-based error metric produces significantly smoother transitions across all movement categories. The post-hoc Tukey's HSD analysis results in Table 6 reveal a clear hierarchy in smoothness across movement magnitudes. Small movements consistently show the smoothest behavior, followed by medium movements, with large movements exhibiting the most variability. This hierarchy suggests that users can expect more predictable rhythmic transitions when making subtle descriptor adjustments compared to dramatic changes.

The choice of KL divergence and Euclidean distance as complementary smoothness metrics proves valuable. KL divergence captures probabilistic differences in the model’s output distributions, reflecting the uncertainty and gradation in rhythm generation, while Euclidean distance in descriptor space measures how well the model preserves the intended control relationships. The concordance between these metrics strengthens confidence in the smoothness findings.

5.4 User Experience Experiment

5.4.1 Validation of the Descriptor-Based Approach

The correlation between objective and subjective similarity measures (Figures 14 and 15) validates the use of Euclidean distance for rhythm pattern comparison and confirms that our computational measures align with human judgment. This alignment demonstrates that participants’ perceptual assessments of rhythmic similarity correspond meaningfully with algorithmic distance calculations, supporting the fundamental assumption that geometric relationships in descriptor space reflect musical relationships as perceived by listeners.

Beyond validating our similarity metric, this correlation provides evidence for the effectiveness of our chosen descriptors as a control space. The fact that users’ subjective evaluations consistently relate to computational measures suggests that the five-dimensional descriptor space captures perceptually relevant aspects of rhythmic structure. This perceptual grounding distinguishes our approach from abstract latent representations, where such alignment between computational and human similarity judgments cannot be assumed.

The statistical significance of this relationship establishes a foundation for automated evaluation of rhythm generation quality, enabling future systems to optimize for human-perceived similarity. Moreover, it validates the descriptor-based methodology as a bridge between computational representation and musical cognition, supporting the integration of music perception research with machine learning approaches for interactive music systems.

5.4.2 User Background and Performance

Table 7 summarizes the correlations between participant background, elapsed time doing the exercises, and the three types of similarity. The results reveal a weak but statistically significant negative correlation between elapsed time and objective similarity, suggesting that participants who spent more time on each exercise tended to perform slightly worse. This may indicate that longer interaction did not consistently support learning, possibly due to ineffective exploration, task difficulty, or frustration.

Musical study experience also showed a weak negative correlation with subjective similarity ratings, revealing a paradox: more musically trained participants rated their reproductions as less similar to the targets, despite no significant difference in objective performance. This may reflect higher critical standards among experienced musicians or suggest that musical training shapes expectations about rhythm control that differ from those supported by the descriptor-based interface.

By contrast, no significant correlations were found between age range, music performance experience, percussion performance experience, and any of the similarity measures. This suggests that the descriptor-based interface may be equally accessible to both specialists and non-specialists across age groups and experiential backgrounds. Nonetheless, a larger and more diverse sample of participants is warranted to obtain more robust insights.

5.4.3 Descriptor Interpretability

User feedback revealed varying levels of descriptor intuitiveness (Figures 16 and 17): start and onset_count were most accessible, while syncopation and balance proved challenging. Despite syncopation’s theoretical grounding in music theory, users found it difficult to control in practice, indicating a gap between theoretical validity and practical usability.

Balance, defined as symmetry of onset distribution around the unit circle, received low intuitiveness ratings. While mathematically well-defined, this geometric mea-

sure may be too abstract for immediate musical comprehension without additional interface support or training.

5.5 Qualitative User Feedback Analysis

In addition to quantitative measures, a qualitative analysis of user feedback provided rich insights into participants' experiences with the descriptor-based rhythm interface. Tables 8, 9, and 10 summarize these findings.

Table 8 highlights aspects of the system that users appreciated, such as the clarity of the interface, immediate auditory and visual feedback, and the enjoyment of exploratory learning. Participants generally found the start and onset_count parameters intuitive. Conversely, they reported difficulties with less tangible descriptors like syncopation and balance, reflecting a gap between theoretical relevance and practical interpretability. Participants also expressed a need for explanatory text or guidance, suggesting that some aspects of the interface were initially opaque.

Table 9 distills recurring patterns in usability. Notably, users valued immediate feedback and hands-on experimentation, which facilitated engagement and learning. However, the complexity of certain sliders and parameter interactions sometimes hindered understanding, highlighting the importance of interface design that balances expressive control with accessibility.

Finally, Table 10 presents specific feature requests. Users proposed contextual tooltips, tutorial systems, and visual aids to better map slider movements to rhythmic outcomes. Additional suggestions included adaptive interfaces tailored to user expertise and expanded control over musical parameters such as swing, instrument selection, and sound textures. These requests indicate directions for improving both learnability and creative flexibility in future systems.

5.6 Methodological Considerations and Limitations

This research operates within several methodological constraints that shape both its contributions and applicability. The focus on 16-step monophonic patterns provides computational tractability and enables systematic analysis of the complete pattern space, but necessarily limits direct application to multiple genres of music that feature variable time signatures. However, this constraint serves the research goals of establishing fundamental principles for descriptor-based control that can inform more complex systems.

The user study with 12 participants provides initial insights into system usability, but the sample size and participant composition limit the generalizability of findings. Notably, few participants had extensive percussion experience (Figure 13), which may have influenced interpretability ratings for rhythm-specific descriptors.

The pattern replication task enables quantitative evaluation and objective performance metrics. While this may not fully reveal creative potential, it establishes baseline performance characteristics that inform system design, prioritizing rigorous evaluation of core principles over immediate practical deployment.

Chapter 6

Conclusions

The central goal of this research was to create a rhythm generation system where users can intuitively shape rhythmic patterns by controlling musically relevant properties. This chapter summarizes how we achieved this objective through systematic comparison of generation approaches, optimization of the control interface, and validation of the system’s effectiveness for musical interaction.

We approached this problem by comparing two generation methods: variational autoencoders (VAEs) and descriptor-based neural networks. While the VAE approach is mathematically elegant and offers smooth latent space interpolation, it proved unsuitable for musical interaction due to its abstract, unintuitive control dimensions. In contrast, the descriptor-based approach—leveraging rhythm features grounded in music perception research—provided a more musically meaningful and engaging interface for users.

Initially, we trained the neural network using 8 rhythm descriptors, however, in order to optimize the descriptor-based system for live use, we systematically reduced the control space from eight to five descriptors through leave-k-out experiments. This minimal set (onset_count, start position, center, syncopation, and balance) maintained a decent pattern accuracy while significantly simplifying the interface.

We defined a smoothness metric to evaluate model performance, quantifying how

gradual changes in descriptor values correspond to smooth rhythmic transitions. The metric combined KL divergence and Euclidean distance in descriptor space to capture both probabilistic and geometric aspects of rhythm change. To systematically assess behavior, we introduced the concept of a *movement*, categorized into three types—small, medium, and large—based on the magnitude of descriptor adjustments. Using this metric across movement types, we identified the model that produced the most continuous and predictable rhythmic transformations.

The user experience experiment, conducted with 12 participants, validated the system’s usability and effectiveness. Participants were able to successfully manipulate rhythmic properties using the descriptor sliders, demonstrating intuitive control over the generated rhythms. Moreover, the observed correlation between objective similarity metrics and participants’ subjective similarity ratings confirmed that our computational approach aligns closely with human rhythm perception, supporting the perceptual relevance of the chosen descriptors.

While constrained to 16-step monophonic patterns and limited by our participant sample, this work demonstrates that perceptually grounded descriptors can bridge computational rhythm generation with intuitive musical control, establishing a foundation for real-time interactive rhythm systems.

Chapter 7

Future Work

A major direction for future research is the development of a pipeline that extends descriptor-controlled monophonic generation into polyphonic drum patterns. This next step would build on previous research, such as the tapping studies by Clark (2023)—which demonstrated how monophonic rhythmic input can be mapped to polyphonic outputs while preserving perceptual structure—and the dualized drum patterns dataset by Haki et al. (2023). By integrating our descriptor-based monophonic generator with polyphonic expansion techniques, users could design simple rhythmic skeletons through intuitive descriptor control and transform them into rich, multi-voice drum arrangements in real time. This approach would require further investigation into how descriptor relationships—such as syncopation and balance—translate to polyphonic structures and whether additional descriptors would be needed.

Current experiments in this thesis relied on the full 16-step binary pattern dataset, which, while comprehensive, can introduce redundancy and noise in the resulting patterns. Future work could explore intelligent dataset reduction strategies to improve efficiency while preserving musical diversity, such as clustering rhythmically similar patterns or identifying archetypal rhythm families. Additionally, it would be interesting to explore extending the system to generate variable-length patterns beyond the 16-step constraint—which would require developing descriptor formula-

tions that generalize across different time signatures and pattern lengths.

Finally, it would be valuable to replicate the user experience experiment with a larger and more diverse participant pool to obtain more robust insights into usability and perceptual alignment. Future studies could also consider an audio-only version of the experiment, removing the visual representation of patterns, to assess how well participants can manipulate and perceive rhythmic structures based purely on sound. Additionally, given that users in the current study were generally able to match target patterns effectively, presenting exercises with larger descriptor distances could help evaluate the system’s behavior over a broader range of rhythmic variations and further challenge participants’ control and perceptual sensitivity.

List of Figures

1	Visualization of a 16-step rhythm pattern.	8
2	Overview of the Variational Autoencoder model pipeline.	9
3	Overview of the Descriptor-based model pipeline.	11
4	Screenshot of the interactive Pure Data patch.	20
5	Screenshot of some logs in the Python server terminal.	20
6	Visualization of a movement in the descriptor space.	24
7	User experience experiment: intro and background questionnaire. . .	28
8	User experience experiment: familiarity step.	28
9	User experience experiment: exercise interaction interface.	29
10	User experience experiment: exercise feedback form.	29
11	User experience experiment: final feedback questionnaire.	30
12	Smoothness experiment plots.	34
13	Age range and music experience distributions.	36
14	Pattern similarity distributions.	36
15	Pattern similarity correlations.	37
16	Distribution of the perceived difficulty for each descriptor.	38
17	Descriptors ranked by relative difficulty.	38

List of Tables

1	Precision using MIDI and floats to define descriptors.	15
2	Leave-one-out (L1O) experiment results.	31
3	Leave-two-out (L2O) experiment results.	32
4	Leave-three-out (L3O) experiment results.	33
5	ANOVA p-values for KL Divergence and Euclidean Distance.	35
6	Tukey’s HSD test for KL Divergence and Euclidean Distance.	35
7	Correlation results between various variables and similarity measures.	37
8	Summary of liked and disliked aspects from user feedback.	39
9	Key usability observations from user testing.	39
10	Feature requests from user feedback.	39

Bibliography

- Brunner, Gino, Yating Wang, Rafael Wattenhofer, and Sitao Zhao (2018). “MIDI-VAE: Modeling dynamics and instrumentation of music with applications to style transfer”. In: *19th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 430–437.
- Clark, A. (2013). “Whatever next? Predictive brains, situated agents, and the future of cognitive science”. In: *The Behavioral and brain sciences* 36.3, pp. 181–204. DOI: 10.1017/S0140525X12000477.
- Clark, Peter (2023). *Tap to Drums: Extending Monophonically Tapped Rhythms to Polyphonic Drum Pattern Generation*. DOI: 10.5281/zenodo.8381068.
- Esling, Philippe, Axel Chemla-Romeu-Santos, and Adrien Bitton (2018). “Generative timbre spaces with variational audio synthesis”. In: *CoRR* abs/1805.08501. arXiv: 1805.08501.
- Fitch, W. and Andrew Rosenfeld (Sept. 2007). “Perception and Production of Syncopated Rhythms”. In: *Music Perception* 25, pp. 43–58. DOI: 10.1525/mp.2007.25.1.43.
- Gabrielsson, A. (1973). “Similarity ratings and dimension analyses of auditory rhythm patterns. I & II”. In: *Scandinavian journal of psychology* 14, pp. 138–176. DOI: 10.1111/j.1467-9450.1973.tb00105.x.
- Gómez-Marín, D., S. Jordà, and P. Herrera (June 2016). “Rhythm spaces”. In: *Proceedings of the 4th International Workshop on Musical Metacreation (MUME 2016)*. *Seventh International Conference on Computational Creativity, ICC 2016*. Paris, France: International Workshop on Musical Metacreation, p. 5.

- (Sept. 2018). “Drum rhythm spaces: from global models to style-specific maps”. In: *Music technology with swing. 13th International Symposium on Computer Music Multidisciplinary Research CMMR 2017*. Ed. by M. Aramaki, M. Davies, R. Kronland-Martinet, and S. Ystad. LNCS 11265. Matosinhos, Portugal: Springer, pp. 123–134. DOI: 10.1007/978-3-030-01692-0_9.
- (Aug. 2020). “Drum rhythm spaces: from polyphonic similarity to generative maps”. In: *J New Music Res* 49.5, pp. 438–456. DOI: 10.1080/09298215.2020.1806887.
- Gómez-Marín, Daniel, Sergi Jordà, and Perfecto Herrera (2015). “PAD and SAD: Two Awareness-Weighted Rhythmic Similarity Distances”. In: *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 573–579. URL: <https://citeseerx.ist.psu.edu/document?doi=64019901897692dc59c9541c089c6e891cef7ac9>.
- Haki, Behzad, Błażej Kotowski, Cheuk Lee, and Sergi Jorda (Nov. 2023). “Tap-TamDrum: A Dataset for Dualized Drum Patterns”. In: *Proceedings of the 24th International Society for Music Information Retrieval Conference. ISMIR*.
- Huron, David (Jan. 2006). *Sweet Anticipation: Music and the Psychology of Expectation*. Vol. 1. The MIT Press. ISBN: 9780262275965. DOI: 10.7551/mitpress/6575.001.0001.
- Kim, Jong Wook, Rachel M. Bittner, Aparna Kumar, and Juan Pablo Bello (2018). “Neural Music Synthesis for Flexible Timbre Control”. In: *CoRR* abs/1811.00223. arXiv: 1811.00223. URL: <http://arxiv.org/abs/1811.00223>.
- Lerdahl, Fred and Ray Jackendoff (1983). *A Generative Theory of Tonal Music*. Cambridge, MA: MIT Press.
- London, Justin (May 2012). *Hearing in Time: Psychological Aspects of Musical Meter*. Oxford University Press. ISBN: 9780199744374. DOI: 10.1093/acprof:oso/9780199744374.001.0001.
- Longuet-Higgins, H. C. and C. S. Lee (1984). “The Rhythmic Interpretation of Monophonic Music”. In: *Music Perception: An Interdisciplinary Journal* 1.4, pp. 424–441. ISSN: 07307829, 15338312. URL: <http://www.jstor.org/stable/40285271> (visited on 08/21/2025).

- Milne, Andrew, Roger Dean, and David Bulger (Jan. 2021). *Tapping to unfamiliar and highly syncopated rhythms: Modelling behaviour and cognitive mechanisms*. DOI: 10.31234/osf.io/qaek6.
- Milne, Andrew and Steffen Herff (June 2020). *The perceptual relevance of balance, evenness, and entropy in musical rhythms*. DOI: 10.31234/osf.io/5ue9a.
- Milne, Andrew J. and Roger T. Dean (Mar. 2016). “Computational Creation and Morphing of Multilevel Rhythms by Control of Evenness”. In: *Computer Music Journal* 40.1, pp. 35–53. ISSN: 0148-9267. DOI: 10.1162/COMJ_a_00343.
- Palmer, C. and C. L. Krumhansl (1990). “Mental representations for musical meter”. In: *Journal of Experimental Psychology: Human Perception and Performance* 16.4, pp. 728–741. DOI: 10.1037/0096-1523.16.4.728.
- Roberts, Adam, Jesse Engel, Colin Raffel, Curtis Hawthorne, and Douglas Eck (2018). “A hierarchical latent vector model for learning long-term structure in music”. In: *International Conference on Machine Learning*. PMLR, pp. 4364–4373.
- Toussaint, Godfried T. (2013). *The Geometry of Musical Rhythm: What Makes a “Good” Rhythm Good?* Boca Raton, FL: CRC Press.
- Vigliensoni, Gabriel, Liam McCallum, Eduardo Maestre, and Robin Fiebrink (2022). “R-VAE: Live latent space drum rhythm generation from minimal-size datasets”. In: *Journal of Creative Music Systems* 1.1.

Appendix A

Source code and demo

This appendix provides additional resources related to the work presented in this thesis. The following links give access to a demo of the user experience experiment and the full source code repository:

- **UX Experiment:** https://alexvilanovab.github.io/d2p_experiment
- **Code Repository:** https://github.com/alexvilanovab/master_thesis

Code Structure

The code repository is organized into several folders, each containing specific components of the project:

`d2p_model` — Implementation of the descriptor model.
`interactive_pd_patch` — Interactive Pure Data patch for experimentation.
`lxo_experiment` — Scripts for the Leave-X-Out experiment.
`smoothness_experiment` — Analysis code for the smoothness experiment.
`ux_experiment` — Analysis code for the user experience experiment.
`vae_model` — Implementation of the Variational Autoencoder model.

These resources allow for both reproduction of results and further exploration of the methods described in this thesis.