
Faculty of Science, Technology and Communication

# Generating 3D Dances From Music Using Deep Neural Networks

Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master in Information and Computer Sciences

*Author:*
Elona Dupont

*Supervisor:*
Prof. Djamila Aouada

*Reviewer:*
Prof. Leon Van der Torre

*Advisor:*
Dr. Renato Baptista

*Co-Advisor:*
Dr. Anis Kacem

June 2021

# Abstract

This thesis focuses on the generation of original and unique 3D dances given a music using deep neural networks. A state of the art model (Dance Revolution) was adapted to take as input 3D data. Then it was trained using the recently published AIST++ dataset. At the generation phase, the model is able to generate credible dances. This was achieved by introducing a novel audio data augmentation technique that modifies the harmonic content of a song without changing the rhythmic content. This method allowed for an increase in the number of training epochs before the LSTM network converges to a static pose. Additionally, a novel method to evaluate the coherence of the generated dances with respect to the style of music is proposed. The comparison is based on key dance moves that are identified using the matrix profile. Using this method to evaluate the dances, it was found that the model generate coherent dances with respect to the dominant styles of music in the dataset.

# Acknowlegements

Working on this project has been a lot of fun but it also happens during times of turmoil. I am therefore extremely grateful to all the Computer Vision, Imaging and Machine Intelligence ($CVI^2$) team for their continuous support. I would like to first thank Prof. Djamila Aouada for believing in me and proposing a research topic that suited my interests and personality. I would also like to thank Dr. Renato Baptista for his support and guidance throughout the process and Dr. Anis Kacem for the always interesting and challenging conversations.

I would also like to thank Nesryne Mejri for her friendship and support in difficult times. And finally I would like to thank Hatsune Miku who has managed to make so many random appearances during this project as if it was written in the stars.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Over the last years, there has been a growing interest from the computer science community to use Artificial Intelligence (AI) to create art pieces. One of the most popular examples is probably the use of Generative Adversarial Network (GAN) [Goodfellow et al., 2014] to create images that are of similar quality as the ones drawn by humans. The hype around artificially generated art is such that in 2018 a painting produced using a GAN was sold for $432,500 [Christie's, 2018]. AIs are not limited to the generation of creative images as they can also be programmed to generate music. For example, MuseNet [Payne, 2019] can generate 4-minute songs made of up to ten different instruments. Nevertheless it can be noted that one artistic area remains under explored by the AI research community, namely dance.

The investigation of this thesis focuses on the generation of unique dance choreographies given a song as input. In this work, we propose a deep neural network framework that produces coherent and credible 3D skeleton dance movements.

## 1.1 Motivation

Generating dances from music is an interesting problem not only from an artistic point of view but also from a practical and commercial perspective. One possible practical application is the generation of a dance from a music track in order to produce live dances for virtual pop idols. Even though virtual pop idols are mostly known to youth subcultures in the Western world, virtual idols such as Hatsune Miku [pia, 2021] are extremely popular in Asia, in particular in Japan. Hatsune Miku has had live performances as a

hologram in some world famous music festivals [Cirone and Decepticon, 2020]. The main limitation of live performances is that the hologram has to be programmed in advance and there is no space for the musician to improvise. To address this, unique AI-based dances could be generated on the fly as the musicians are performing. Further applications could be found in video games and animation. In fact this would support small independent studios that might not have the funds to hire and record real life dancers to quickly generate original dance choreographies.

## 1.2 Recent Developments in AI

Nowadays, the majority of AI-based generative methods rely on deep neural networks (DNN). Neural networks have proven to be extremely successful at numerous tasks such as scene labelling, face recognition, action recognition, text classification, translation etc [Bhandare et al., 2016]. The training of DNNs requires large amounts of labelled data that can act as ground truth. For example, for a neural network to learn to identify whether a cat is present in a picture, a large set of labeled images of cats as well as images without cats with the corresponding label is needed. In the same fashion, in order to generate dances from music, a dataset of music and corresponding dance should be used. One challenge when creating a dance dataset is that there does not exist a unique ground truth dance for a given music. One can easily imagine a variety of equally "correct" dances for the same music. This is due to the fact both music and dances are creative processes and consequently a subjective aspect.

Despite those challenges, there has been a growing interest for music conditioned dance generation in the past few years. However, most works focus on 2D dances [Fan et al., 2011] [Lee et al., 2019a] [Huang et al., 2021]. Few approaches have considered 3D dances. However they remain either non-reproduceable as neither the code nor the data has been published [Li et al., 2020a] or the models are yet to be published at the time of writing [Li et al., 2021].

## 1.3 Contributions

The main contributions of this thesis are:

- extending the 2D Dance Revolution Network [Huang et al., 2021] to 3D,

- using the newly released AIST++ dance dataset [Li et al., 2021] for training,
- combining a sliding window method and a novel audio preprocessing method that allows for longer training and therefore dances with a larger variety of moves,
- evaluating the coherence of the generated dances with respect to the style of music using a novel criterion based on the concept of motifs or key movements.

## 1.4 Outline

The structure of the thesis is as follows:

- In Chapter 2, a literature review of cross-modal learning and dance generation from audio is presented.
- Chapter 3 focuses on providing the relevant background information related to digital audio and matrix profile that is used to analyse time series. The problem tackled in this thesis is also formalised.
- Chapter 4 presents the overall proposed approach. The approach consists of three main components; (1) a modification of the original Dance Revolution Network to generate 3D dances, (2) a novel approach to preprocess the input data using a sliding window method with audio data augmentation, (3) a new criterion to evaluate dances using motifs.
- In Chapter 5, the experimental setup is described in details. First a thorough description of the AIST++ dataset is presented, then the different strategies investigated to deal with dance data of different duration as well as the audio data augmentation method are explained in great depth. The training set-up is also outlined.
- Chapter 6 presents a thorough analysis of the results generated by the model. The analysis is divided into two parts; a qualitative analysis and a quantitative analysis using our novel dance evaluation criterion based on key movements.
- Finally, in Chapter 7, the conclusion of the thesis is presented as well as possible directions for future work.

# Chapter 2

# Related Work

In this chapter, we present recent research findings in the fields related to cross-modal learning and music based dance generation.

## 2.1 Cross-Modal Learning

Cross-modal learning refers to learning models that transforms one type of information into a different type [Skocaj et al., 2012]. Generating dances from music is a task of transferring an audio sequence to a sequence of 2D or 3D images. Cross-modal learning is most commonly centered around computer vision and natural language processing. One example is image captioning, which consists of predicting a sequence of word from a single image. The results obtained are often very impressive [Lu et al., 2017]. This is due, in part, to the availability of large public datasets for this task, often containing millions of annotated images [Li et al., 2020b]. Other sequence-to-sequence cross-modal methods include text-to-speech [Valle et al., 2020], text and notes to audio singing [Lee et al., 2019b], speech-to-hand gestures [Ferstl et al., 2020] and video-to-music [Gan et al., 2020]. However, music to dance generation is still a field that remains relatively unexplored mainly due to the difficulty of obtaining reliable datasets.

## 2.2 Dance Generation from Audio

Generating unique dances from music is a topic that has recently been receiving more attention. This can be explained by the recent advances in learning-based methods. One of the earliest work focused on finding a mapping between styles of music and key moves using an AdaBoost method [Fan et al., 2011]. [Lee et al., 2019a] proposed a

decomposition-to-composition framework. The decomposition phase aims at extracting movement beats from a dancing sequence using a kinematic beat detector. For that purpose a variational auto-encoder maps the dance units into an initial pose space. In the composition phase, a music-to-movement GAN is used to generate a sequence of movements conditioned on the input music. The model proposed is complex and the dataset used is of low quality and only contains 2D dance data extracted from YouTube videos. Recent developments in transformers and attention mechanisms have led to improvements of dance generation methods (see Section 3.4.2 for a more detailed description of transformers). [Li et al., 2020a] proposed a model using transformers. However, neither the data nor the source code for the model were made available, making the results not reproducible. Another promising work using transformers [Li et al., 2021] has also not released their model (at the time of writing). They proposed a new dataset, called AIST++, that provides high quality 3D dances of skeleton data points for a wide range of dance styles (see Section 5.1 for more details). A limitation of this model is that it requires a 2 second dance seed to generate a dance of arbitrary length.

A key component in this thesis is the model presented by [Huang et al., 2021] as it proposes a novel model composed of a transformer to embed music features, followed by a Long Short Term Memory (LSTM) network to generate dances. The model proposed was trained using 2D data extracted from YouTube videos at 15 frame per seconds. Notwithstanding, the data is not sufficiently accurate leading to the generation dances containing many artifacts. For example the presence of body limbs growing out of proportion for short periods of time can be observed. The main contribution of this thesis is therefore to modify this network to 3D. Hence we can take advantage of the AIST++ dataset for generating high quality 3D dances at 60 frames per second.

# Chapter 3

# Background and Problem Formulation

In this section, the necessary background information related to the representation of music and dance music is recalled. Then, the required tools for the analysis of time series as well as a basic description of the neural network architectures is explained. Finally, we present the formulation of the problem of generating dances from music.

## 3.1 Music as a Digital Signal

In this section, concepts related to sound and music are introduced. These concepts are particularly relevant to understanding how the raw audio data is modified before used as input to the DNN.

### 3.1.1 Sound

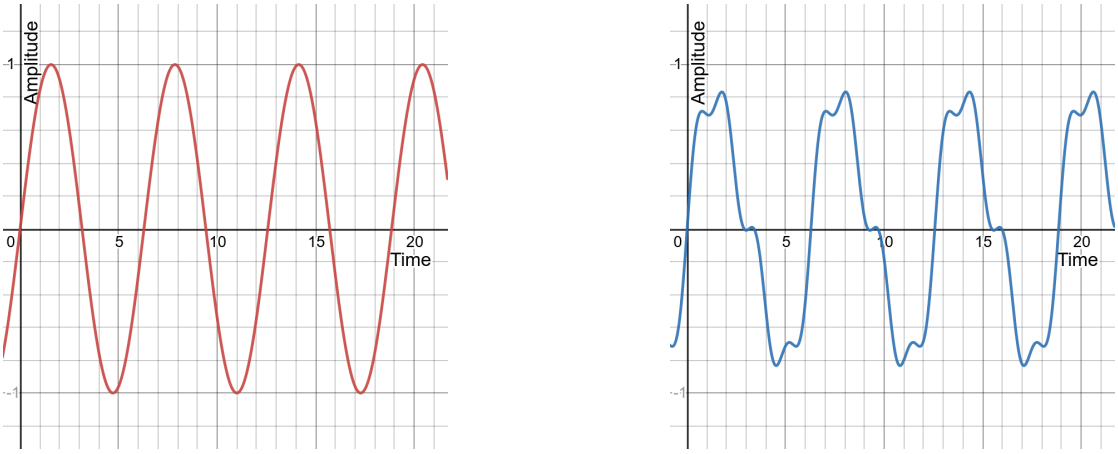
A music is perceived by humans through the hearing of sound. Thus, to produce a model that generates dances from music, it is important to have some basic understanding of sound and music.

A sound is the result of the oscillation of a medium (such as air particles). The frequency of oscillation (the number of oscillations per second) is related to pitch perception and the amplitude (the displacement from equilibrium position) is related to the perceived loudness of a sound. Both pitch and loudness human perception are logarithmic in scale [Smith, 1997]. Humans can generally hear sounds of frequencies within the range of 20Hz to 20kHz but are more sensitive to sounds between 1 kHz and 4 kHz. The

sensitivity to changes in frequencies varies at different frequency ranges. For example, it is easy to distinguish two tones if their frequencies differ by more than about 0.3% at 3kHz, whereas at 100Hz a 3% difference is required.

### 3.1.2 Music

Beyond pitch and loudness, there are also two other main components to take into account when discussing sound as music, namely timbre and tempo. Pitch, loudness, timbre and tempo can be thought as the main components of a music that a dancer will intuitively consider to decide how to move their bodies at a point in time. Timbre is a term used to describe the harmonic content of a sound. For example, the same note played on a piano and a guitar will sound different as the timbre of the instruments are different [Blatter, 2017]. This is illustrated in Figure 3.1.




(a) Pure sine tone

(b) Sine tone with added harmonic

Figure 3.1: The perceived pitch (fundamental frequency) for the two tones would be the same. 3.1b has more harmonics and therefore has a different timbre from 3.1a

Tempo, on the other hand, is a measure of the speed of the music. It is usually measured in beats per minute (bpm). In many Western music, notes are generally played on the beat, on the half, quarter, eighth or sixteenth beat [Blatter, 2017]. Tempo is a crucial component when dancing as it directly influences the speed at which the body moves.

### 3.1.3 Digital Audio and Mel Frequency Cepstral Coefficient

As previously mentioned, sound is the result of the oscillation of air particles. A sound at a point in space can be modelled by a continuous function $x(t)$ such that $x$ is the displacement of the particle from the equilibrium position at an instant of time $t$. In order to store a sound information in a computer, it is convenient to quantize $x$ into a digital signal (a discrete series) $x[n]$ where $n$ is a time step. The number of samples per second is called the sampling rate and it is common in digital audio to have a sampling rate of approximately 44kHz [Smith, 1997]. This signal is a convenient representation of the voltages to be sent to a speaker to create a sound.

To obtain a representation of sound as it is perceived by humans, it can be useful to map the original digital audio signal from the temporal domain to the frequency domain. The Mel Frequency Cepstral Coefficients (MFCC) is a popular method to obtain a more representative signal with respect to the perception of sound [Huang et al., 2005].

Given a digital signal $x[n]$, the first step to find the MFCCs is to apply the Discrete Fourier Transform (DFT) to $x[n]$. We obtain, the signal $S[l]$ expressed in the frequency domain.

$$S[l] = \sum_{k=0}^{N-1} x[k] e^{-i2\pi klN}, 0 \leq k < N. \tag{3.1}$$

Then, a triangular filterbank $H_m[k]$ with $M$ filters ($m = 1, 2, ..., M$) is defined as:

$$H_m[k] = \begin{cases} 0, & k < f[m-1] \\ \frac{k - f[m-1]}{f[m] - f[m-1]}, & f[m-1] \leq k \leq f[m] \\ \frac{f[m+1] - k}{f[m+1] - f[m]}, & f[m] < k \leq f[m+1] \\ 0, & k > f[m+1] \end{cases} \tag{3.2}$$

Figure 3.2 illustrates the filterbank. It can be noted that the width of the filters increases as the frequency increases. This is to match the human frequency perception as discussed in Section 3.1.1. The frequency boundary points $f[m]$ are the range of frequencies covered by the triangular filters. To match human frequency perception, the boundary points are equally spaced in the mel-scale and $f[m]$ is defined as:

$$f[m] = \frac{N}{F_s} B^{-1}(B(f_l) + m \frac{B(f_h) - B(f_l)}{M+1}), \tag{3.3}$$

where $F_s$ is the sampling frequency, $f_l$ and $f_h$ the highest and lowest frequencies of the

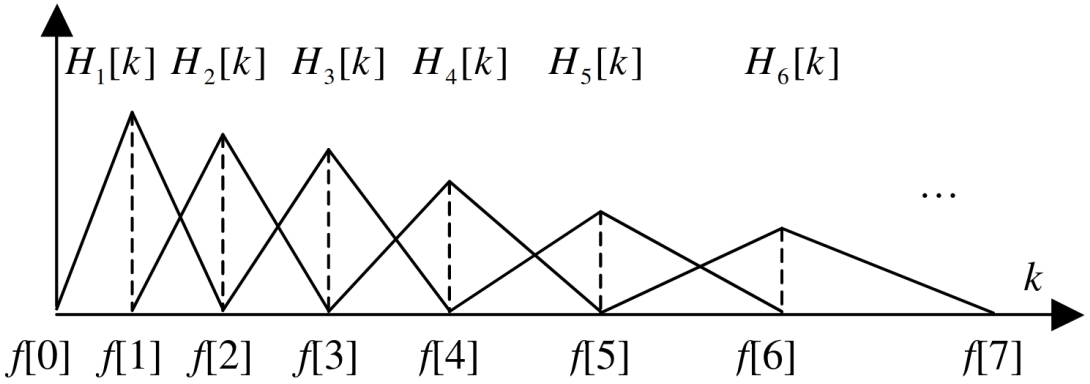



Figure 3.2: Filterbank used to compute the MFCC [Huang et al., 2005]

filterbank, $N$ the size of the FFT and $B$ the mel-scale with:

$$\begin{aligned} B(f) &= 1125\ln(1 + f/700) \\ B^{-1}(b) &= 700(e^{b/1125} - 1) \end{aligned} \tag{3.4}$$

Then the log-energy of the output $L[m]$ of each filter is computed as follows:

$$L[m] = \ln \sum_{k=0}^{N-1} \mid S[k] \mid^2 H_m[k], \qquad 0 \leq m < M. \tag{3.5}$$

Finally, the Mel Frequency Cepstrum coefficients $c[n]$ are computed using the discrete cosine transform of $L[m]$ such that:

$$c[n] = \sum_{m=0}^{M-1} L[m] cos(\pi n(m + 0.5)/M), \qquad 0 \leq n < M. \tag{3.6}$$

## 3.2 3D Skeleton and Dance Representation

As the goal of the thesis is to generate dances from an input music, it is essential to have a mathematical representation of a human body incorporating the temporal aspect.

In computer vision, it is common to represent the motion of the human body as a times series of body joint coordinate positions, which mathematically can be expressed as a sequence of vectors [Xing and Zhu, 2021]. Formally, the representation of an $d_j$-joint 3D skeleton, $\mathbf{Y}_t$, at an instant of time $t$ is,

$$\mathbf{Y}_t = [\mathbf{j}_1, ..., \mathbf{j}_{d_j}], \tag{3.7}$$

where $\mathbf{j}_i \in \mathbb{R}^3$ is the position of $i^{th}$ joint.

A dance is then represented as a temporal sequence of 3D skeletons and can be denoted as $\mathbf{Y} = \{\mathbf{Y}_1, ..., \mathbf{Y}_T\}$.

In the AIST++ dataset, the 3D representation of the skeleton at a given time (a pose) consists of 17 points corresponding to key body points, mostly joints (see Figure 3.3 and Table 3.1). A dance is therefore represented as a sequence of poses of dimension $17 \times 3 \times d_y$ where $d_y$ is number of frames in the dance video. In the dataset the dances are captured at 60 fps that is to say one second of dance consists of 60 poses.

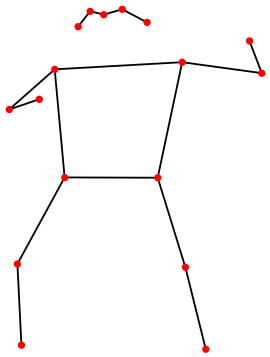


Figure 3.3: Example of skeleton joint points (shown in red)

| Joints | | | | |
|---|---|---|---|---|
| Nose | Left eye | Right eye | Left ear | Right ear |
| Left shoulder | Right shoulder | Left elbow | Right elbow | Left wrist |
| Right wrist | Left hip | Right hip | Left knee | Right knee |
| Left ankle | Right ankle | | | |

Table 3.1: List of body key point joints

## 3.3 Time series analysis and the matrix profile

The previous sections introduced the representations of music and dance. In this part, the required tools to quantify whether the generated dances are coherent with respect to the style of the input music will be presented.

In order to evaluate the dances, identifying similar dance moves of a given duration (window length) from the 3D skeleton time series representation will be required. Finding the nearest neighbour of each window of a time series is known as the all-pairs-similarity-search and can be solved efficiently using the matrix profile algorithm [Yeh et al., 2016].

### 3.3.1 Matrix Profile

The matrix profile algorithm is a fast similarity search algorithm using the z-normalized Euclidean distance and leveraging the overlap between subsequences and the Fast Fourier Transform. Given a time series and a subsequence length, the matrix profile consists of two sequences of the same length of the original time series. The first sequence is a score to the nearest neighbour subsequence (the matrix profile) and the second sequence consists of the corresponding indices of the start of the closest nearest neighbour subsequence (see Figure 3.4).

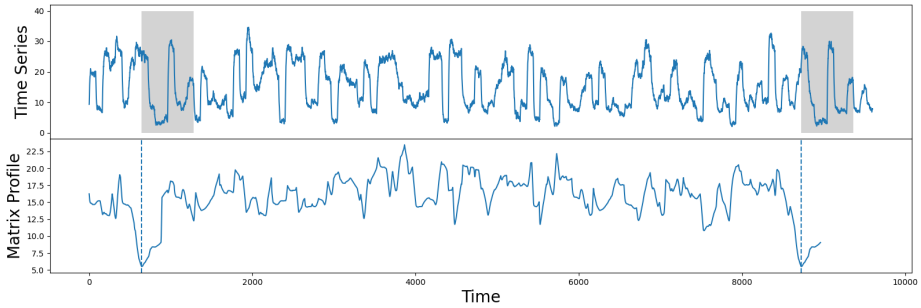



Figure 3.4: Example of a time series and its corresponding Matrix Profile. The two time steps at which the Matrix Profile is the lowest correspond to the most similar windows(adapted from [Law, 2019])

Subsequences with lowest distance scores are the most similar patterns in the series and can be considered as motifs of the time series. One of the key element of the evaluation of the dances is to identify the most significant motif for different window length.

### 3.3.2 Multidimensional Matrix Profile

The matrix profile algorithm can be adapted for multi-dimensional time series [Yeh et al., 2017] using the mSTAMP algorithm. This is essential to identify motifs in the dances as the 3D skeleton representation is multi-dimensional and a key dance move should be defined by the motion of more than one body joint. Figure 3.5 shows an example of a 60 frame long motif identified using the 3D coordinates of a subset of joints.

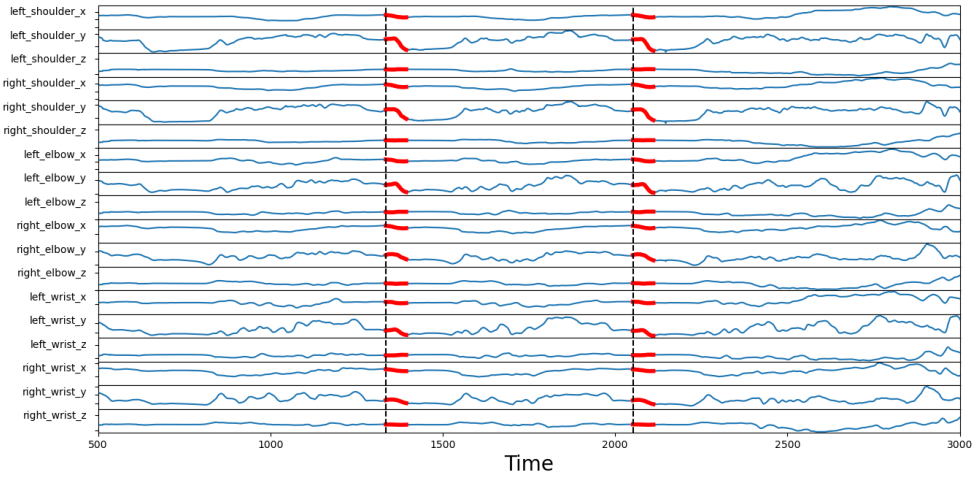



Figure 3.5: Mutlidimensional Matrix Profile with the two closest 60-frame windows highlighted in red

### 3.3.3 Time series chain

Using the matrix profile, it is possible to identify time series chains [Zhu et al., 2017]. A time series chain is constructed by considering the right and left nearest neighbors of each subsequence in the series. A time series chain is an ordered set of subsequences such that for each subsequence the right matrix profile index to an element has a left matrix profile index that points back to the original subsequence. To illustrate this, let us consider the following toy example time series (adapted from [Law, 2019]):

$$47, \quad 32, \quad 1, \quad 22, \quad 2, \quad 58, \quad 3, \quad 36, \quad 4, \quad -5, \quad 5, \quad 40 \tag{3.8}$$

For the purpose of illustrating the concept of a time series chain, we consider a window size of 1 so that the matrix profile value between two elements corresponds to the difference between them.

Figure 3.6 shows the closest right neighbor for each element (arrows above) and the closest left neighbor (arrows below). The longest chain forming a closed loop is from the sequence 1, 2, 3, 4 and 5 as shown in Figure 3.7.

Time series chains will be used to identify the most relevant key move in a dance and evaluate the coherence of a generated dance with respect to the style of the input music. The time series chain with the lowest matrix profile distance will be considered for different subsequence lengths.

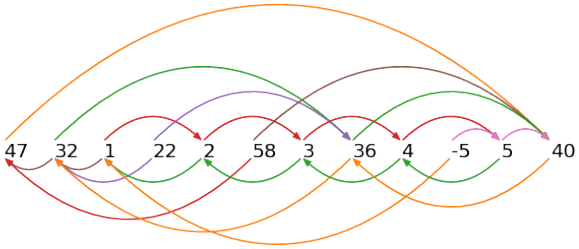



Figure 3.6: Illustration of the right and left nearest neighbors of each element in the time series [Law, 2019]


Figure 3.7: Illustration of the longest time series chain [Law, 2019]

## 3.4 Neural Network Architectures

In this section the structure and roles of the main neural network components of the Dance Revolution model are outlined.

### 3.4.1 Long Term Short Memory

One of the earliest form of neural network specifically designed for generating predictions of time series is the Recurrent Neural Network (RNN) [Rumelhart et al., 1985]. Compared to simple feedforward neural network, RNNs make use of a so-called latent state (or hidden state) that depends on the input of the current time step and of the latent state from the previous time step. It was found that RNNs fail to capture long term dependencies due to the vanishing gradient problem. To solve this issue the LSTM architecture was introduced [Hochreiter and Schmidhuber, 1997]. LSTMs share the same idea as the vanilla RNN architecture but make use of three different types of gates (input gates, forget gates, and output gates) at each time step [Zhang et al., 2020]. The input (update) gate decides when to read data into memory cell and captures long-term dependencies in sequences. The forget gate is a mechanism for resetting the content of

the memory cell and helps capturing short-term dependencies in sequences. The output gate combines the entries from the memory cell to produce a single output. Figure 3.8 shows a representation of an LSTM memory cell.

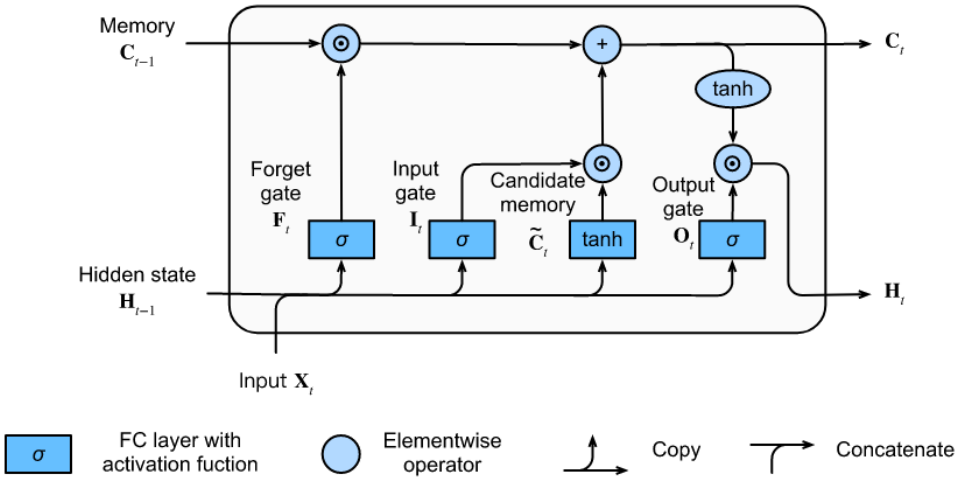



Figure 3.8: Illustration of the mathematical flow of a LSTM memory cell [Zhang et al., 2020]

In this investigation, an LSTM architecture is used to take as input the music embedding at each time step and to output the corresponding pose, this is the dance decoder.

### 3.4.2 Transformer

The transformer architecture is an alternative neural network model to solve problems with a temporal component. It was designed to rely solely on the attention mechanism and avoid using convolutional or recurrent type layers [Vaswani et al., 2017]. A transformer consists of several multi-head attention layers.

#### 3.4.2.1 Attention

Given a query $Q$ and a set of key($K$)-value($V$) pairs with $Q$, $K$ and $V$ vectors, an attention function is a mapping of $Q$, $K$ and $V$ to an output vector. The dimensions of $Q$ and $K$ are $d_k$ and the dimension of $V$ is $d_v$. The attention function can be thought of as a weighted sum of the values where the weights are found by computing the similarity between the key and the query using the softmax ($\sigma$) of the dot product of the vectors (see eq. (3.9)).

$$\text{Attention(Q,K,V)} = \sigma\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{3.9}$$

#### 3.4.2.2 Multi-Head Attention

A transformer is made of multi-head attention layers. That is, at each layer, multiple attention functions are computed in parallel to learn different representations of the subspace. The output of a multi-head attention consisting of $h$ heads can be computed as follows:

$$\begin{aligned} \text{Multihead}(Q, K, V) &= \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O \\ \text{with} \quad \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{3.10}$$

with parameter matrices $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}, W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}, W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ and $d_{\text{model}}$ the dimension of the output.

In this work, a multi-head transformer is used to take as input the preprocessed music features at each time step and output the corresponding embedding; this is the music encoder.

## 3.5 Problem Formalization

This thesis aims at producing a model that can generate coherent and relevant 3D dances given a music track. The model is to be trained using examples of actual dances annotated with their corresponding music.

Let $\mathcal{D} = \{(\mathbf{X}^{(i)}, \mathbf{Y}^{(i)})\}_{i=1...N}$ be a set of $N$ data points with $\mathbf{X}^{(i)}$ as music in a digital audio signal form and $\mathbf{Y}^{(i)}$ its corresponding desired target (e.g. 3D skeleton dance). The goal of the investigation is to estimate a generative model $\mathcal{G}$ such that $\mathcal{G}$ maps a music, $\mathbf{X}^{(i)}$, into a dance, $\mathbf{Y}^{(i)}$. The music vector, $\mathbf{X}^{(i)}$, is a set of discrete audio samples through time in $\mathbb{R}^{d_x}$ where $d_x$ is number of samples in the sound file. The dance vector, $\mathbf{Y}^{(i)}$, is a representation of skeleton in 3D through time in $\mathbb{R}^{d_y \times 3 \times d_j}$ where $d_y$ is the number of frames in the dance video and $d_j$ the number of joints in the skeleton. Therefore, the generative model is such that $\mathcal{G} : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y \times 3 \times d_j}$.

# Chapter 4

# Proposed Work

In this chapter, the general approach taken to train a model to generate 3D dances given music will be outlined.

## 4.1 Extension of the Dance Revolution Network Architecture to 3D

In this section, the Dance Revolution Network architecture is described in details. As mentioned in Section 2.2, the Dance Revolution Network consists of two main components a music encoder and a dance decoder (see Figure 4.1).

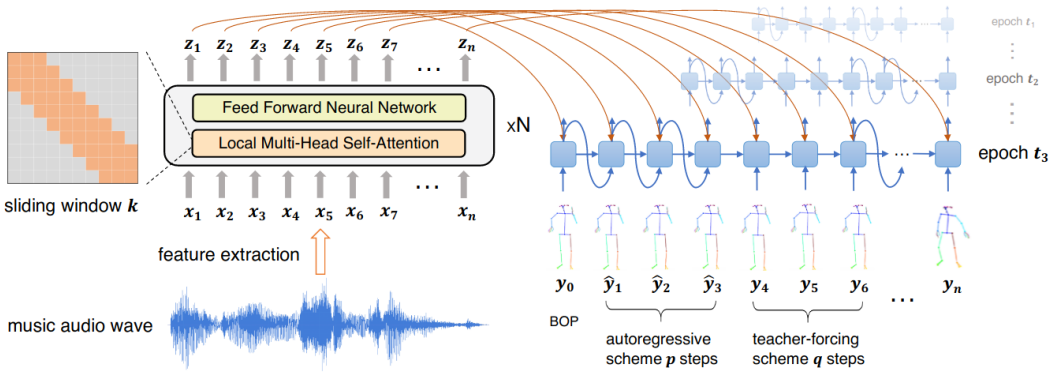



Figure 4.1: Illustration of the Dance Revolution Model [Huang et al., 2021]

The Dance Revolution Network imposes some constraints on the structure of the training data. The first one is that all the dances used for training must be of identical duration and thus consist of the same number of frames. More details on how we achieve this will be presented in Section 4.2 and Section 5.2. Another constraint is that the

LSTM must be fed an initial begin of pose (BOP), $\mathbf{Y}^0$. This is chosen to be a random skeleton pose.

### 4.1.1 Music Encoder

The role of the music encoder is to identify an appropriate transformation of the raw digital audio into a set of relevant embeddings to generate a suitable dance.

Instead of using the raw digital signal as input to the transformer, some preprocessing steps are performed on the audio. This is done to obtain a representation of the audio that is closer to the one of human perception (see Section 3.1.2). In fact, it is important to provide the network data that is similar to the one perceived by the dancer.

The features extracted in the preprocessing step can be grouped into three categories: pitch, loudness and rhythm. The pitch characteristic features are the MFCC, the MFCC delta and the Constant-Q chromagram (similar to a discrete Fourier transform but with 1/24th octave filter bank [Brown, 1991]). The loudness or strength features are the tempogram and the onset strength. The onset strength is converted to a one-hot vector to represent the beat. A total of 438 features are extracted per frame. The audio is sampled at a rate of 30,720Hz and hop size 1024, therefore providing 60 frames per second to match the dance frame rate. The features are extracted using the Librosa library [Mcfee et al., 2015].

The structure of the transformer is not modified. The transformer consists of a stack of 2 identical layers. Each layer consists of a local self-attention sublayer with 8 heads such that $d_k = d_v = 64$ and a fully connected feed-forward sublayer with 1024 hidden units.

### 4.1.2 Dance Decoder

The role of the dance decoder is to transform the music embeddings into a pose sequence. The dance decoder consists of 3 LSTM layers with an input size of 1024 (corresponding to the size of the output of the music encoder) and the pose vector as dimension 51 corresponding to 17 3D joints. The latter had to be modified from the original architecture as 25 2D joints were used.

## 4.2 Robustness Against Data Length Variation

One challenge that is often faced when training neural networks with time series is that the input of the network should have a fixed pre-defined duration. However, dances are based on music and songs that might have different lengths. Hence, the dances and their corresponding musics must be adapted to fit this fixed duration requirement.

As the dataset originally used with the Dance Revolution model was extracted from videos, it was possible to use dances of any given length that is no more than the shortest video (one minute in that case). The transformer based AIST++ model uses clips of 4 seconds to get a 2 seconds seed and a 2 second prediction. However, this would most likely fail to identify long-term patterns in dances. Another disadvantage of chopping the dance into smaller pieces is that the same music would be used several times making the model less robust.

To overcome this, we propose a sliding window method with a novel data augmentation strategy that allows for the training of a more robust model.

## 4.3 Proposed Evaluation Metric

Evaluating the quality of the generated dances is not a simple task as the relation between the music and the dance is not a one-to-on and onto mapping. There is no scientific consensus on a criterion that quantifies the quality of a generated dance. The most common criteria attempt to quantify how close in time music beats are to kinetic beats, quantify how close the generated dance is to an actual dance and quantify the quality of the dance with a user study [Fan et al., 2011] [Huang et al., 2021] [Lee et al., 2019a] [Li et al., 2020a] [Li et al., 2021].

However, we argue that these methods are not scientifically sound. Attempting to match kinetic beats to music beats is not a trivial process. The kinetic beat is defined as a sudden drop in the velocity of a joint. Thus, the claim that a synchronized dance must have kinetic beats close to musical beats is fair but it is not a necessity. Indeed, sharp movements could happen not only on a beat but also on a quarter or half beat. For instance, if the music has a bpm of 100bpm, then the duration of a beat is 0.6s and the duration of a quarter beat is 0.15s. Using quarter beats as the resolution is very small compared to the resolution of the generated dances to provide a significant insight. Moreover, a dance with smoother movements, such as ballet would, has generally

a smaller amount of sharp movements making this method hardly applicable.

As mentioned previously, dance is a creative process and two dances generated from the same music could be perfectly valid and yet be very different. Comparing a generated dance to an actual dance is therefore not an appropriate as it does not take into account the creative aspect of dances.

Finally, using a user survey to evaluate dances is at best subjective and consequently unreliable as the evaluation of the dances would highly depend on the dance knowledge of the participant. Moreover, an expert in one style of dance is not necessary expert in another style. For example, [Huang et al., 2021] based their survey on a sample of 10 amateur dancers, which is clearly too small to be statistically significant.

We propose a novel method to evaluate dances that focuses on quantifying the coherence of dances with respect to the style of music used as input. The main assumption in our method is that for each style of music there should be similar key dance moves. These moves should be therefore be identified in generated dances.

The evaluation method consists of first identifying the key dance moves or motifs for each style of music using the matrix profile. Then how close those moves are in dances of different styles is measured. A good dance generator should generate dance moves that can be found in similar styles of music.

# Chapter 5

# Experiment

In this chapter, the different experiments used to train the model to learn to generate dances from music is detailed. First, a short description of the AIST++ dataset is presented. Then, different preprocessing methods for adapting the data are discussed. Also, a novel technique to augment the audio data is depicted. Finally, the experimental setup used for training is presented.

## 5.1 AIST++ Dataset

As discussed in Section 2.2, most previous works in music conditioned dance generation use 2D data that can be extracted from 2D videos. The extraction of the dance relies on human pose estimation methods that tend to produce noisy data. Also it can be difficult to project these 2D dance data into 3D. In this section, a large and recent 3D dataset used in this work is described.

The AIST++ dataset ([Tsuchida et al., 2019]) is to the best of our knowledge the largest existing dance dataset with 3D human keypoint annotations. The dataset is composed of 1408 sequences of 3D human dance motion but only 60 musics distributed equally among the dance genres. The same music is therefore used for different dances. It is important to note that even though there are 10 different dance genres (refer to Table 5.1) the number of music genres is much smaller as the same music style can be used for different dances. Most of the dance genres could be described as modern and urban. By listening to different musics in the data, they can be classified into four main music genres: Hip-Hop, Ballet, Pop/Funk and Techno. The duration of the choreographies varies between 7.4 seconds to 48 seconds.

| Dance Genre | Music Tempo | Number of choreographies | Dance duration (seconds) | | Total duration (seconds) |
|---|---|---|---|---|---|
| | | | Basic (85%) | Advanced (15%) | |
| ballet jazz | 80 - 130 | 141 | 7.4 - 12.0 | 29.5 - 48.0 | 1910.8 |
| street jazz | 80 - 130 | 141 | 7.4 - 12.0 | 14.9 - 48.0 | 1875.3 |
| krump | 80 - 130 | 141 | 7.4 - 12.0 | 29.5 - 48.0 | 1904.3 |
| house | 110 - 135 | 141 | 7.1 - 8.7 | 28.4 - 34.9 | 1607.6 |
| LA-style hip-hop | 80 - 130 | 141 | 7.4 - 12.0 | 29.5 - 48.0 | 1935.8 |
| middle hip-hop | 80 - 130 | 141 | 7.4 - 12.0 | 29.5 - 48.0 | 1934.0 |
| waack | 80 - 130 | 140 | 7.4 - 12.0 | 29.5 - 48.0 | 1897.1 |
| lock | 80 - 130 | 141 | 7.4 - 12.0 | 29.5 - 48.0 | 1898.5 |
| pop | 80 - 130 | 141 | 7.4 - 12.0 | 29.5 - 48.0 | 1872.9 |
| break | 80 - 130 | 140 | 7.4 - 12.0 | 23.8 - 48.0 | 1858.3 |
| Total | | 1408 | | | 18694.6 |

Table 5.1: Summary of the AIST++ dataset dance genres and their characteristics

## 5.2 AIST++ Data Preprocessing

As mentioned earlier in Section 4.1, one of the challenge resulting from the use of the AIST++ dataset is that the dances have different duration whereas the network requires inputs of identical length. In this section, different possible solutions to address this issue are presented.

### 5.2.1 Interpolation

Ideally, the input dance would be as long as possible allowing the network to learn long-term dependencies. Therefore, one possible approach is to upsample short dances to make their duration equal to the longest available dance. This is a common practice in the computer vision field [Yao et al., 2020] [Ghorbel et al., 2015]. Interpolation to stretch the duration of the dances has been investigated using spline interpolation. However, this leads to different issues.

First of all, not only the duration of the motion needs to be extended but also the duration of the corresponding music. While it is possible to stretch an audio over time, this often results in a drastic change in pitch. This would also change the original bpm to a bpm that is much lower than the one expected in a given style of music. We can be solve that by sampling the audio at a higher rate than shorter videos. As long as the required sampling rate is below the original sample rate of the audio file then this method works.

The main concern with interpolating the motion is that a method such as spline interpolation tends to smooth out the curves. This is usually desirable as it reduces the noise in the data. Nonetheless, in the context of dance motion, the smoothing tends to take away sharp movements that are important features for characterizing dances. As shown in Figure 5.1, the sharp movements are smoothed out (wider peaks). We can also observe a slight temporal translation which is not desirable since the temporal synchronisation between the dance movements and music is a key information for the training of the network. When training the model with interpolated data (using cubic splines), it has been noticed that the dancer looked as if they were underwater. This might be explained by the fact that the generated dance movements are lacking sudden changes in velocity.

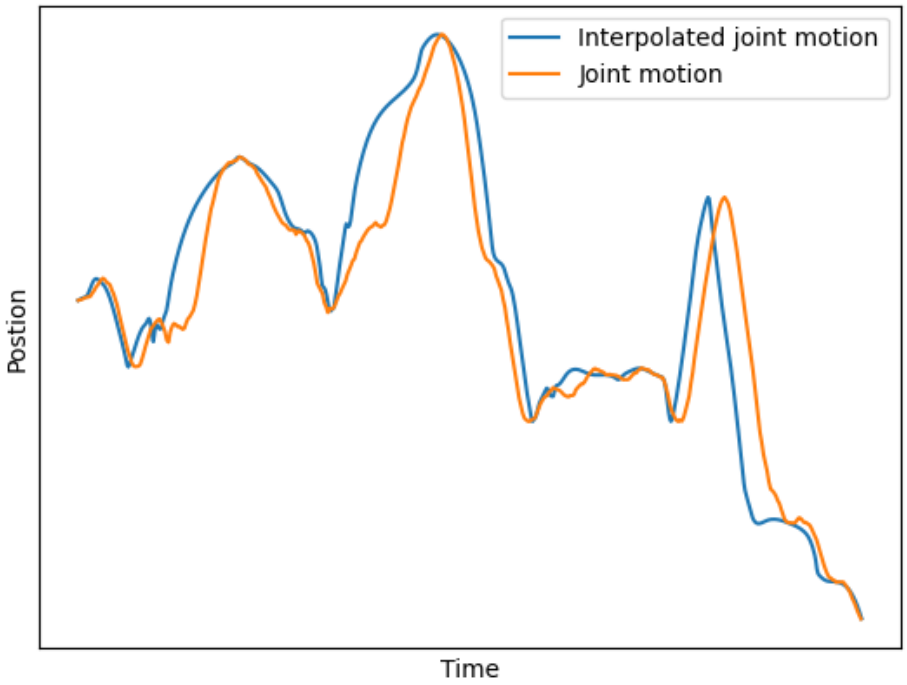



Figure 5.1: Example of the position of a body point in one dimension as a function of time. The original joint motion consists of 443 frames and the interpolated joint motion consist of 720 frames.

### 5.2.2 Zero-Padding

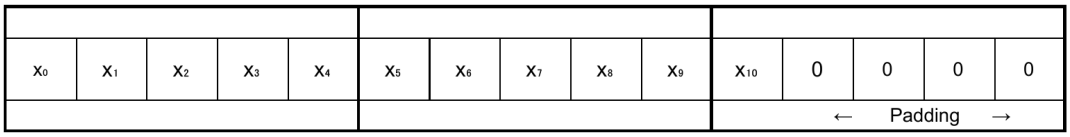
Another possible solution is to chop the dances in equal duration segments, e.g. the duration corresponding to the shortest duration. If the dance duration is not an exact multiple of the duration of the shortest dance, then the last segment can be padded with zeroes. However, it was noticed through the experiments, that if many examples have long zero padding, then the model fails to generate coherent dances. It was also observed that the model quickly converges towards producing a static pose of an upside down skeleton. Another disadvantage is that the chopping moment in the videos is arbitrary. Since it is based on the shortest video, the cutting point could happen in the middle of a move and the information would be lost. To address this issue, a sliding window method can be used.

### 5.2.3 Sliding Window

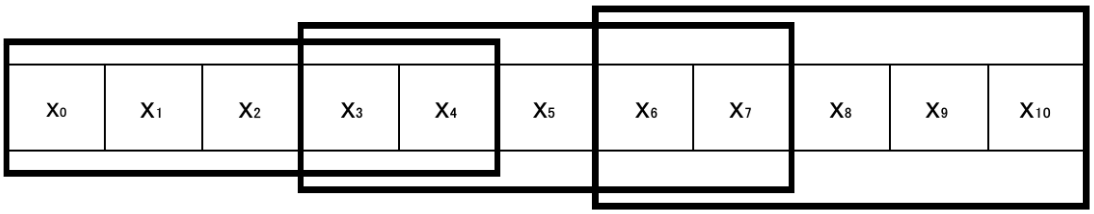
One alternative method to divide the time series in equal lengths is to use a sliding window method. This consists in taking segments of the dances of a given length (equal to the shortest dance) at an interval apart that is smaller than the window length. This is illustrated in Figure 5.2. In the experiments, the sliding window length used was equal to 420 frames (corresponding to 7 seconds) at 60 frames intervals. This method ensures that all the features of the dances are being used in the training. This also leads to a large number of training samples. This can be seen as beneficial as it provides more examples to learn from. However, it is also a disadvantage as it increases the training time and provides many examples with the same music.

## 5.3 Audio data augmentation

Data augmentation is a common practice in computer vision when training neural networks. It aims at providing a more robust model that is able to generalize effectively under different variations [Simard et al., 2003] [Ciregan et al., 2012] [Wan et al., 2013]. Popular techniques in data augmentation include flipping, rotating images, adding noise and changing contrast or saturation. When augmenting the audio data, it is primordial to avoid changing the rhythmic structure of the music, as it is a crucial element that defines a dance. However, the harmonic content or timbre of the instruments (see Section 3.1.2) can be modified. Changing the harmonic content of a song is analogous to changing the




(a) Zero padding




(b) Sliding window

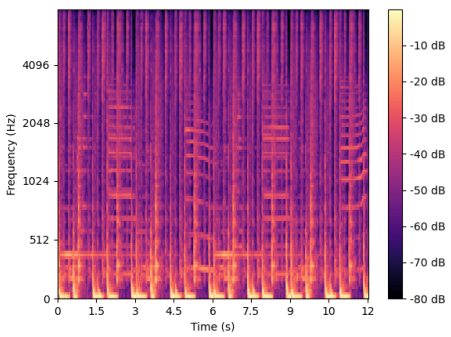
Figure 5.2: Diagram illustrating how to divide a time series with ten samples using zero padding (5.2a) and the sliding window method (5.2b) with a window length of five and an spacing interval of 3 samples. Both methods produce three windows, however the last window when using zero padding is mostly composed of 0's.

color of an image. For that purpose, the harmonic and percussive parts of the music are first extracted. Then the harmonic signal is shifted in pitch by a random amount of quarter-tone between 2 lower octaves and 2 higher octaves. Then, a random number of harmonics between 1 and 7 are added to the signal. The pitch of the percussive content is also shifted by a random amount of quarter tones between one lower octave and one higher octave. This allows the audio to remain within the hearing range. The two signals are then recombined. This leads to a music containing the same rhythmic structure but with a slightly different tonal content. This can be seen in Figure 5.3, the spectra of the processed augmented is only transformed in frequency (vertically) but there is no change in the temporal placement of the notes (no horizontal shift).

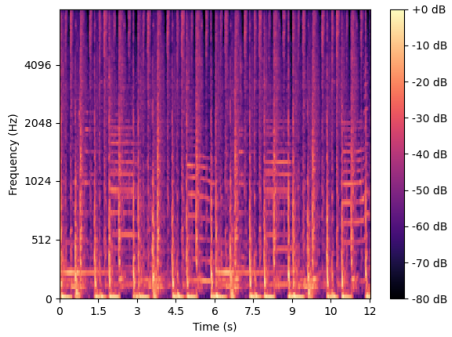
## 5.4 Training Set-Up

Using the sliding window method the initial choreographies of varying lengths have been transformed to 7510 training files of 420 frames or 7 seconds each, corresponding to the shortest choreography in the dataset.

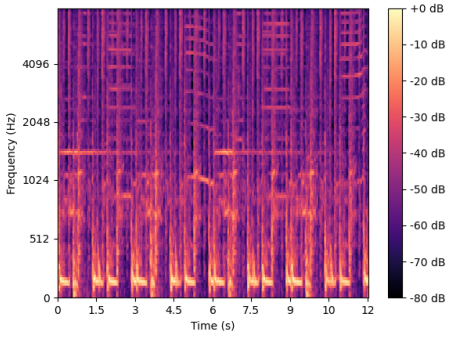
The training of the network is conducted on an Intel Xeon E5-2640-v4 CPU and an NVIDIA Titan V GPU for 6000 epochs. Each epoch takes about 3min20s, making the



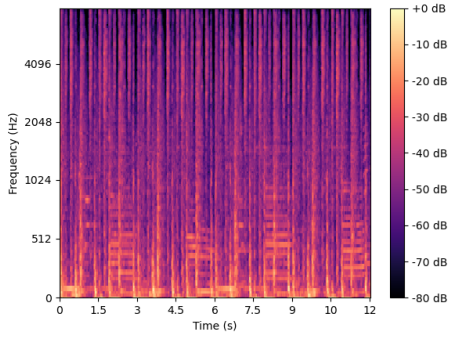

(a) MFCC of original audio




(b) MFCC of processed audio




(c) MFCC of processed audio




(d) MFCC of processed audio

Figure 5.3: Graphs showing the effect of the preprocessing method on the frequency domain of the audio files

total training time slightly under 14 days.

# Chapter 6

# Evaluation

In this section, the dances generated by the model are thoroughly evaluated. First a qualitative evaluation is conducted. Then a quantitative evaluation using the motif based approach is reported allowing the assessment of the style coherence of the generated dances.

## 6.1 Qualitative dance evaluation

In this section, the dances obtained using different preprocessing methods are compared.

### 6.1.1 Results using zero padding

As mentioned in the previous section, chopping longer dances in equal sequences and padding the last frames with zeroes can have a negative impact by possibly cutting a dance at a point of time that could be relevant; thus resulting in a loss of information. By looking at the dances generated on unseen music, another weakness of this method has been identified. Some dances contained some jittery movements. The jitter is mainly due to the fact that the model generates three dances at the same time and considering frames with an interval of 3 frames produced a smooth and coherent dance. The jittering motion appears mostly at the beginning of the dance or when there is a drastic variation in the music. It can be hypothesised that the jittery motion is the result of the model trying to settle for one style of dance. On some occasions the jittering motion did not disappear for the entire generated video, thus making the dance totally incoherent and unrealistic. Even with a training of 10,000 epochs, it was not possible to completely remove this issue.

### 6.1.2 Results Using Sliding Window

#### 6.1.2.1 Without audio data augmentation

The main improvement that the sliding window method brought is that the jittery motion considerably decreased even if it was still present at the beginning of the generated dances for a few seconds after 3000 epochs. However, this came at a cost. The dances generated from the model after a training of 4000 epochs were just a static skeleton for all the tested musics. Convergence of RNN based models towards a static pose is a well known phenomena that is due to the difference in distributions of the data and the model [Aksan et al., 2020]. One common method to address this issue is to add noise to the data [Jain et al., 2016]. The audio data augmentation technique is a novel method that aims at addressing this issue. The rest of this chapter will be dedicated to the analysis of the results obtained using the sliding window and the audio augmentation methods.

#### 6.1.2.2 With audio data augmentation

The main observable advantage of using the audio data augmentation method described in Section 5.3 is that it allows for longer training. It is possible to train the model for up to 6000 epochs and still obtain coherent generated dances. At 7000 epochs, the dances tend to have slower moves with some static poses between moves. This indicates that the model tends to slowly converge towards a static pose. As a result the training is stopped at that point and only results for up to 6000 epochs are considered.

By inspecting the generated dances at every epoch, it appears that the longer the model trains the more dance moves the model learns. After 1000 or 2000 epochs of training, it seems that most generated dances are mainly made of the same few dance moves but as the training reaches about 5000 epochs, or more, the generated dances appears to contain a larger variety of dance moves. Some generated dance moves can be seen in Figure 6.1.

For dance non-experts, it is difficult to judge whether the generated dances that are around 3 minutes long are a patchwork of moves or a complete choreography. This contrasts with the results of the original Dance Revolution network [Huang et al., 2021]. As a remainder, the original Dance Revolution model is trained with 2D data with dance moves of 1 minute. It is therefore possible for the model to learn long-term dependencies in dances as the training data contained complete choreographies. On the other hand, the AIST++ dataset used in this investigation is made of short dances (from 7 to 40 seconds

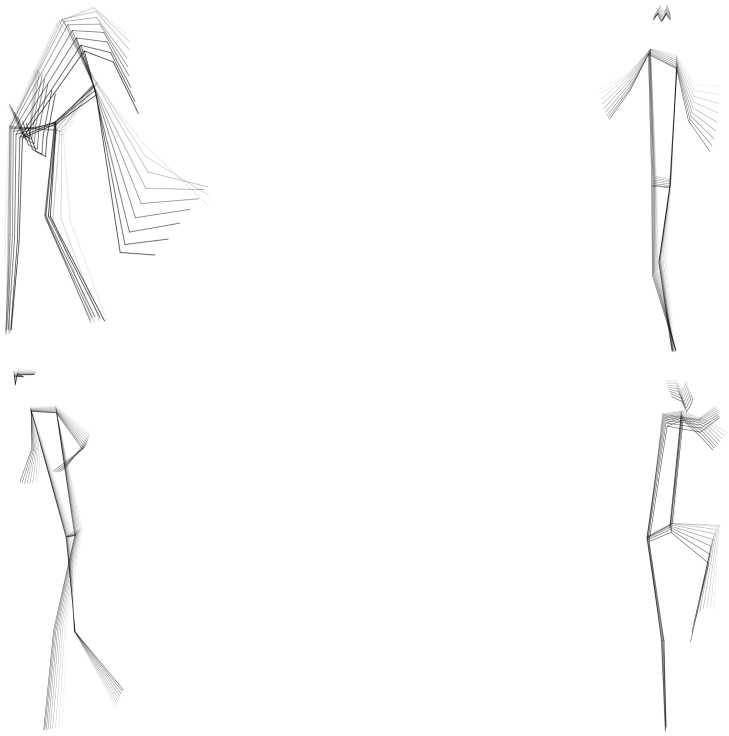


Figure 6.1: Examples of dance moves generated

long) and most of the basic dance videos are just one dance move repeated over and over. Therefore, the training data does not contain sufficient information for the model to learn how to compose coherent and creative dance choreography from the learned moves. As mentioned earlier, [Li et al., 2021] also developed a model that is trained on the AIST++ dataset using only transformers and no LTSM. Unfortunately, the model as well as the code are not published at the time of writing so it is not possible to compare them with our results. However, some short videos of the results have been published. Interestingly, the published generated dances are relatively short (16 to 22 seconds) and the majority seems to contain the same dance move repeated. From these observations it can be hypothesized that their model was not able to generate long coherent and creative compositions.

The audio data augmentation has clearly improved the generated dances. One disadvantage of the technique is that due to the large amount of data, it is a computationally heavy task. As a result, the preprocessing of the audio is carried out once before the training starts. It would be interesting to try to re-process the audio every 500 or 1000 epochs to expose the model to slightly different data. This might further improve the robustness of the model.

## 6.2 Quantitative Dance Evaluation

In this section, a novel criterion for quantitatively assessing the ability of the model to predict appropriate dances for a given music style is presented and used to evaluate our model.

### 6.2.1 Motif-Based Evaluation

As mentioned in Section 3.3 and Section 4.3, the proposed model is evaluated on its ability to generate dance moves that are appropriate for a given style of music. This is achieved by identifying similar dance moves (motifs) that are repeated within the same generated dance and trying to find the closest match in other dances.

For this purpose, 4 styles of music are considered: ballet, hiphop, pop-funk and techno. Those styles of music were chosen as, even though the AIST++ dataset contains 10 different styles of dances, all the musics fall under one of these categories. It is very difficult to differentiate the music used for dance sub-genres such as waack and lock for example.

For each style of music, 8 one-minute long song samples were used making a total of 32 minutes of music. The musics chosen are not part of the AIST++ dataset and were therefore not used during the training.

For each of the music segments, the most significant motifs of length 30 frames (0.5 seconds) and 120 frames (2 seconds) are identified using the matrix profile and time series chains (see Section 3.3.3). An example of motifs can be seen in Figure 6.2, in which it can be noted that the dancer's joint positions are very similar in the two instances of the motif found but there are some small differences in joint positions such as in the left arm on the third frame and fourth frame.

Once the key motifs for each generated dance are identified, the remaining tasks are to find the closest match of these motifs in each dance and then to quantify how close the match is to the motif.

To identify the most similar segment of a generated dance to a motif, the motif is first concatenated to the dance. Then, the matrix profile is run and the closest segment is identified as the index pointer of the matrix profile corresponding to the first frame of the motif.

Finally, the Euclidean distance between the motif and the closest identified match is calculated and averaged over the number of frames in the motif (30 or 120).

Figure 6.2: Example of a key motif found twice in the same generated dance. The two dance moves are clearly very similar but there are small apparent differences in the position of the left arm on the third and fourth frame.

The key idea is that dances generated from the same style of music should have key dance moves that are more similar than the ones generated from other styles of music. In the next section, the results will be analysed for different number of epochs.

### 6.2.2 Results

Table 6.1 shows the results obtained using the metric described in the previous Section 4.3. The average scores are computed at different stages of the training (every 1000 epochs). If the motifs of a given style have similar matching moves within a generated dance then the score should be low. Therefore, a high score means that on average the generated dances are quite different from the motif. It is expected that a well performing model will have as lowest score the same style of music for the motif and the compared generated dances. For example the motifs extracted from the hiphop music are expected to score lower when compared against dances generated using hiphop music than from the ones generated from other styles of music.

The first point to be made from the table is that the overall average score increases as the number of epochs increases. This can be explained by the fact that the longer the model is trained for the larger the variety of dance moves are in the generated dance. As mentioned in the qualitative analysis, early in the training, the model generates dances that are mostly made of a few identifiable dance moves.

Another observation that can be made is that for the motifs of style pop-funk, the

average score is consistently the lowest for the generated dances of style pop-funk. This suggests that the model is able to generate characteristics moves for that style of music and a dance generated from this style of music is coherent with respect to the music.

For the motifs of hiphop style, the lowest scores through out the training are either for the generated dances of style hiphop or pop-funk. Since the chosen pop-funk songs tend to have a musical beat that is similar to a disco beat, it is therefore not very different from hiphop from a rhythmical perspective. However, the melodies and harmonic contents of the two styles are quite different. It can be noted that for the hiphop motifs if the lowest score is not the one from the hiphop generated dances, the hiphop generated dances score has the second lowest value. It can therefore be concluded that there is some coherence for the dances generated from a hiphop music with respect to the style of music.

Similarly, the motifs of techno style the lowest score or second lowest score is for dances of techno style. After a training of 4000 and 6000 epochs, the lowest score obtained is for hiphop and pop-funk respectively. Those style of musics are quite different from techno. Therefore, it seems that there is less coherence in the dances generated from a techno music than in the ones resulting from hiphop or popfunk.

Finally, the last style of motifs to consider is ballet. It seems that it was more difficult for the model to generate key moves that were distinctive for this style of music as the lowest score is often from different style of generated dances. Even though at 6000 epochs the lowest score is for ballet, it can be noted that the second lowest score is not that different. This can be due to the fact that ballet is very different from the other styles of music and comparatively there is less examples of ballet style music examples in the training set. One observation that could also explain this result is that in many of the generated dances some movements that could be clearly identified as ballet moves (such as a jump with rotation) could be found in many generated dances of different styles.

As a result, it can be concluded that the model is able to generate dances that are overall coherent with respect to the style of music. This is especially true when dealing with styles of music that are well represented in the dataset such as hiphop, pop-funk and techno. It is possible that a more balanced dataset with respect to the styles of music and not just with respect to the styles of dances would lead to more coherent and consistent results.

| Motifs music style | Generated dance music style | Average score | | | | | |
|---|---|---|---|---|---|---|---|
| | | 1000 | 2000 | 3000 | 4000 | 5000 | 6000 |
| ballet | ballet | **10.8** | 17.6 | 27.5 | 30.6 | **22.0** | **32.8** |
| | hiphop | 19.2 | 17.4 | 25.2 | **29.2** | 26.6 | 36.5 |
| | pop-funk | 19.5 | **16.8** | **22.7** | 30.1 | 30.9 | 33.3 |
| | techno | 19.8 | 20.0 | 23.4 | 33.3 | 30.9 | 35.4 |
| hiphop | ballet | 20.5 | 20.8 | 26.2 | 28.0 | 36.5 | 32.8 |
| | hiphop | 12.0 | **12.8** | **17.4** | 16.5 | **28.9** | 30.8 |
| | pop-funk | **11.7** | 13.5 | 18.4 | **15.8** | 29.7 | **28.2** |
| | techno | 13.5 | 15.2 | 17.9 | 25.5 | 29.7 | 32.2 |
| pop-funk | ballet | 20.4 | 20.3 | 26.6 | 32.2 | 36.6 | 34.5 |
| | hiphop | 14.8 | 12.2 | 21.1 | 22.6 | 34.8 | 32.3 |
| | pop-funk | **11.7** | **10.4** | **16.9** | **17.6** | **24.8** | **24.2** |
| | techno | 15.9 | 12.5 | 20.8 | 30.3 | 34.1 | 27.2 |
| techno | ballet | 19.0 | 22.2 | 27.5 | 35.7 | 41.5 | 30.0 |
| | hiphop | 13.8 | 13.4 | 19.7 | **27.9** | 36.7 | 30.9 |
| | pop-funk | 12.7 | 13.5 | 17.9 | 31.0 | 35.5 | **23.1** |
| | techno | **11.6** | **12.4** | **16.2** | 30.7 | **31.7** | 24.8 |

Table 6.1: Average scores for the comparison between key motifs of different styles and the generated music for different training epochs. A coherent model would generate scores that are lowest for matching style of motif and generated dance.

# Chapter 7

# Conclusion

In this chapter, a summary of the thesis contributions is presented. Then, some possible ideas for future work are proposed.

## 7.1 Summary

In this thesis, a method to adapt an existing state-of-the-art model (2D Dance Revolution) is presented in order to generate 3D dances from music. The model is trained using the recently published AIST++ dataset. It is noted that the model is able to generate coherent and credible 3D dances for different styles of music.

The first part of the thesis put the investigation into context by presenting recent research in cross-modal learning and audio based dance generation.

Then the problem was presented in a formal manner and the relevant theoretical information related to digital representation of music and dance were outlined. The theoretical information related to the matrix profile was described in details as it is a key element in the evaluation of the dances. The relevant background on the different neural network architectures was also briefly outlined.

The main contributions of this thesis are adapting a 2D model to generate 3D dances, a novel audio data augmentation technique and a new approach for evaluating the generated dances.

Then the experiments conducted were described in details with the main component being using 3D dance data of different duration for a network that takes as input dances of fixed duration.

Finally, the results produced by the model were evaluated. It was found that aug-

menting the audio data by changing the main pitch and harmonic content is without any doubt the most successful contribution of this thesis. It was found that it allowed for longer training of the model. LSTMs, when trained for a large number of epochs tend to produce a static output which in the case of this investigation is a static pose. Using the data augmentation, the model could be trained for twice as long before starting to converge towards outputting a static pose.

A novel method to evaluate the dances is also proposed using the concept of key dance moves or motifs. The key motifs for four different styles are identified then used to quantify how well a generated dance belongs to a certain class. Using this metric, it has been found that our model is able to generate dances that are coherent with respect to the most represented styles of music in the dataset (hiphop, popfunk and techno), but at the same time failed when considering under-represented style of music such as ballet.

## 7.2 Future Directions

One direction that might need further improvements is the method used to evaluate the dances. It is very difficult to qualitatively evaluate the dances without being a dance expert. It is even more difficult to define what qualities a dance expert are. This is because dancing is not just a creative process but it is also very cultural dependant. The AIST++ dataset is only made of modern Western dances and it is would be interesting to make effort to include other kind of dances. It is also extremely challenging to quantify the quality of a dance.

Beyond the evaluation of dances, the question of the assessment of creativity of AIs can be expected to take a larger part within the scientific community. However, defining and quantifying creativity is extremely challenging. A paradigm shift in mathematical representations might be required for not only producing truly creative AIs but to also assess creativity.

# Bibliography

[pia, 2021] (2021). Hatsune miku official website. https://piapro.net/intl/en.html. Accessed: 2021-6-1.

[Aksan et al., 2020] Aksan, E., Cao, P., Kaufmann, M., and Hilliges, O. (2020). A spatio-temporal transformer for 3d human motion prediction.

[Bhandare et al., 2016] Bhandare, A., Bhide, M., Gokhale, P., and Chandavarkar, R. (2016). Applications of convolutional neural networks. *nternational Journal of Computer Science and Information Technologies,*, 7(5):2206–2215.

[Blatter, 2017] Blatter, A. (2017). *Revisiting music theory: basic principles*. Routledge.

[Brown, 1991] Brown, J. C. (1991). Calculation of a constant q spectral transform. *The Journal of the Acoustical Society of America*, 89(1):425–434.

[Christie's, 2018] Christie's (2018). Is artificial intelligence set to become art's next medium? https://www.christies.com/features/A-collaboration-between-two-artists-one-human-one-a-machine-9332-1.aspx.

[Ciregan et al., 2012] Ciregan, D., Meier, U., and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649.

[Cirone and Decepticon, 2020] Cirone, D. and Decepticon (2020). Kyary pamyu and pamyu and hatsune miku to turn coachella "kawaii" in 2020 – j-generation.

[Fan et al., 2011] Fan, R., Xu, S., and Geng, W. (2011). Example-based automatic music-driven conventional dance motion synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 18(3):501–515.

[Ferstl et al., 2020] Ferstl, Y., Neff, M., and McDonnell, R. (2020). Adversarial gesture generation with realistic gesture phasing. *Computers & Graphics*, 89:117–130.

[Gan et al., 2020] Gan, C., Huang, D., Chen, P., Tenenbaum, J. B., and Torralba, A. (2020). Foley music: Learning to generate music from videos. *Computer Vision – ECCV 2020 Lecture Notes in Computer Science*, page 758–775.

[Ghorbel et al., 2015] Ghorbel, E., Boutteau, R., Boonaert, J., Savatier, X., and Lecoeuche, S. (2015). 3d real-time human action recognition using a spline interpolation approach. *2015 International Conference on Image Processing Theory, Tools and Applications (IPTA)*.

[Goodfellow et al., 2014] Goodfellow, I. J., Pouget-abadie, J., Mirza, M., Xu, B., Warde-farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *In NIPS*.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

[Huang et al., 2021] Huang, R., Hu, H., Wu, W., Sawada, K., Zhang, M., and Jiang, D. (2021). Dance revolution: Long-term dance generation with music via curriculum learning. In *International Conference on Learning Representations*.

[Huang et al., 2005] Huang, X., Acero, A., and Hon, H.-W. (2005). *Spoken language processing: a guide to theory, algorithm, and system development*. Prentice Education Taiwan.

[Jain et al., 2016] Jain, A., Zamir, A. R., Savarese, S., and Saxena, A. (2016). Structural-rnn: Deep learning on spatio-temporal graphs. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Law, 2019] Law, S. M. (2019). STUMPY: A Powerful and Scalable Python Library for Time Series Data Mining. *The Journal of Open Source Software*, 4(39):1504.

[Lee et al., 2019a] Lee, H.-Y., Yang, X., Liu, M.-Y., Wang, T.-C., Lu, Y.-D., Yang, M.-H., and Kautz, J. (2019a). Dancing to music.

[Lee et al., 2019b] Lee, J., Choi, H.-S., Jeon, C.-B., Koo, J., and Lee, K. (2019b). Adversarially trained end-to-end korean singing voice synthesis system. *Interspeech 2019*.

[Li et al., 2020a] Li, J., Yin, Y., Chu, H., Zhou, Y., Wang, T., Fidler, S., and Li, H. (2020a). Learning to generate diverse dance motions with transformer.

[Li et al., 2021] Li, R., Yang, S., Ross, D. A., and Kanazawa, A. (2021). Learn to dance with aist++: Music conditioned 3d dance generation.

[Li et al., 2020b] Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., and et al. (2020b). Oscar: Object-semantics aligned pre-training for vision-language tasks. *Computer Vision – ECCV 2020 Lecture Notes in Computer Science*, page 121–137.

[Lu et al., 2017] Lu, J., Xiong, C., Parikh, D., and Socher, R. (2017). Knowing when to look: Adaptive attention via a visual sentinel for image captioning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Mcfee et al., 2015] Mcfee, B., Raffel, C., Liang, D., Ellis, D., Mcvicar, M., Battenberg, E., and Nieto, O. (2015). librosa: Audio and music signal analysis in python. *Proceedings of the 14th Python in Science Conference*.

[Payne, 2019] Payne, C. (2019). Musenet. https://openai.com/blog/musenet.

[Rumelhart et al., 1985] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation.

[Simard et al., 2003] Simard, P., Steinkraus, D., and Platt, J. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 958–963.

[Skocaj et al., 2012] Skocaj, D., Leonardis, A., and Kruijff, G.-J. M. (2012). *Cross-Modal Learning*, pages 861–864. Springer US, Boston, MA.

[Smith, 1997] Smith, S. W. (1997). *The scientist and engineers guide to digital signal processing*. California Technical Pub.

[Tsuchida et al., 2019] Tsuchida, S., Fukayama, S., Hamasaki, M., and Goto, M. (2019). Aist dance video database: Multi-genre, multi-dancer, and multi-camera database for dance information processing. In *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019*, pages 501–510, Delft, Netherlands.

[Valle et al., 2020] Valle, R., Shih, K., Prenger, R., and Catanzaro, B. (2020). Flowtron: an autoregressive flow-based generative network for text-to-speech synthesis.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Wan et al., 2013] Wan, L., Zeiler, M., Zhang, S., Lecun, Y., and Fergus, R. (2013). Regularization of neural networks using dropconnect. In *In Proceedings of the International Conference on Machine learning*.

[Xing and Zhu, 2021] Xing, Y. and Zhu, J. (2021). Deep learning-based action recognition with 3d skeleton: A survey. *CAAI transactions on intelligence technology*, 6(1):80–92.

[Yao et al., 2020] Yao, L., Yang, W., and Huang, W. (2020). A data augmentation method for human action recognition using dense joint motion images. *Applied Soft Computing*, 97:106713.

[Yeh et al., 2017] Yeh, C.-C. M., Kavantzas, N., and Keogh, E. (2017). Matrix profile vi: Meaningful multidimensional motif discovery. *2017 IEEE International Conference on Data Mining (ICDM)*.

[Yeh et al., 2016] Yeh, C.-C. M., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, H. A., Silva, D. F., Mueen, A., and Keogh, E. (2016). Matrix profile i: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. *2016 IEEE 16th International Conference on Data Mining (ICDM)*.

[Zhang et al., 2020] Zhang, A., Lipton, Z. C., Li, M., and Smola, A. J. (2020). *Dive into Deep Learning*. https://d2l.ai.

[Zhu et al., 2017] Zhu, Y., Imamura, M., Nikovski, D., and Keogh, E. (2017). Matrix profile vii: Time series chains: A new primitive for time series data mining (best student paper award). *2017 IEEE International Conference on Data Mining (ICDM)*.